

IBM WebSphere InterChange Server



**WebSphere InterChange Server
インプリメンテーション・ガイド**

V4.3.0

IBM WebSphere InterChange Server



**WebSphere InterChange Server
インプリメンテーション・ガイド**

V4.3.0

お願い

本書および本書で紹介する製品をご使用になる前に、371 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM WebSphere InterChange Server (5724-178)、バージョン 4.3.0 および IBM WebSphere Business Integration Toolset (5724-177)、バージョン 4.3.0 および IBM WebSphere Business Intergration Toolset (5724-177)、バージョン 2.6 に適用されます。

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原 典： IBM WebSphere InterChange Server
Implementation Guide for WebSphere InterChange Server
V4.3.0

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2004.10

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、
平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2001, 2004. All rights reserved.

© Copyright IBM Japan 2004

目次

本書について	vii
対象読者	vii
本書の前提条件	vii
関連文書	vii
表記上の規則	viii
本書の使用	viii
このリリースの新機能	ix
リリース 4.3 の新機能	ix
リリース 4.2.2 の新機能	ix
リリース 4.2.1 の新機能	x
リリース 4.2.0 の新機能	x
リリース 4.1.1 の新機能	xi
リリース 4.1.0 の新機能	xii
リリース 4.0.0 の新機能	xii
第 1 章 IBM WebSphere InterChange Server の開発モデルとプロセス	1
WebSphere InterChange Server Business Integration システム	1
WebSphere InterChange Server 開発モデル	3
実装の段階	8
第 2 章 環境の準備	15
個々の開発環境の準備	15
開発統合環境の準備	22
ソース・コード制御の使用	23
第 3 章 IBM WebSphere InterChange Server ツールおよび環境	27
System Manager、統合テスト環境、および Collaboration Debugger について	27
System Manager の使用	30
InterChange Server インスタンスの処理	35
統合コンポーネント・ライブラリーの処理	44
ユーザー・プロジェクトの処理	47
統合コンポーネント・ライブラリー内のコンポーネントの処理	52
ソリューションの処理	66
System Manager を使用したパッケージへのコンポーネントのエクスポート	68
サーバーへのコンポーネントの展開	70
InterChange Server リポジトリ内のコンポーネントの処理	83
依存関係および参照	94
複数のワークベンチ・リソースで使用可能な標準の操作	96
Eclipse ベースのワークベンチの使用	98
第 4 章 ビジネス・プロセス・インターフェースの開発	109
全体的な開発フロー	109
開発作業の順序	111
第 5 章 WebSphere InterChange Server の構成	119
InterChange Server 構成ウィザードの使用	119
ウィザードを使用したセキュリティー・プロパティの構成	124
System Manager の使用	125
System Manager を使用したセキュリティーの構成	137

第 6 章 repos_copy の使用	143
Repos_copy の構文	143
repos_copy の使用シナリオ	150
repos_copy ファイルのロケール	157
第 7 章 コネクターの構成	159
Connector Configurator の使用	159
コネクターの標準プロパティ	180
第 8 章 データベース接続プールの構成	207
データベース接続プールを使用する状況	207
データベース接続プールを使用しない状況	210
データベース接続プールとデータベース接続の作成	211
データベース接続プールの検証	214
データベース接続プールの変更	215
コラボレーションおよびマップでのデータベース接続プールの使用	216
トランザクション・ブラケットの構成	217
第 9 章 コラボレーション・オブジェクトの構成	219
コラボレーション・オブジェクトとそのグループ	219
コラボレーション・オブジェクトの作成	220
コラボレーション・オブジェクトの変更	225
第 10 章 Relationship Manager の使用	241
Relationship Manager の始動	243
サーバーへの接続および接続の切断	243
関係の処理	244
関係データの処理	252
第 11 章 Test Connector の使用	261
推奨されるテスト手順	261
Test Connector の始動	262
Test Connector のシャットダウン	263
コネクタ・プロファイルの作成および編集	263
コネクタのエミュレート	264
ビジネス・オブジェクトの処理	265
第 12 章 統合テスト環境の使用	273
テスト・サーバーとしての InterChange Server の登録	273
統合テスト環境の始動	274
統合テスト環境インターフェース	274
サーバー構成の選択	277
設計モードで始動するための InterChange Server の構成	278
テスト・モードでのサーバーの開始	279
テスト・プロジェクトおよびテスト・ユニットの使用	280
アウトライン・ビューの使用	286
タスク・マネージャー・ビューの使用	293
統合テスト環境コンソール・ビューおよび InterChange Server コンソール・ビューの使用	301
テスト・ユニット・ビューの使用	303
Client Simulator ビューの使用	309
BO インспекター・ビューの使用	325
統合テスト環境を使用したテストの実行	333
第 13 章 Collaboration Debugger の使用	337
Collaboration Debugger の始動	337
Collaboration Debugger インターフェース	339

Collaboration Debugger の接続および切り離し	345
コラボレーション・テンプレート・エディターの使用	347
イベントの処理	350
ブレークポイントの処理	351
デバッグ操作の実行	355
変数の処理	358
コラボレーションのデバッグ	359
第 14 章 パフォーマンスの調整	361
イベントにより起動されるフローの並行処理の実装	361
コネクター・エージェントによる要求の並行処理の実装	363
コネクター・エージェントの配布	366
静的関係のキャッシング	367
データベース接続プールの使用	367
メモリー・チェッカー・スレッドの使用	367
特記事項	371
プログラミング・インターフェース情報	372
商標	373
索引	375

本書について

IBM^(R) WebSphere^(R) InterChange Server およびそれに関連する Toolset は、IBM WebSphere Business Integration adapters とともに使用され、先進の e-business テクノロジーとエンタープライズ・アプリケーションとのビジネス・プロセス統合および接続性をもたらします。

本書では、WebSphere InterChange Server にビジネス・インテグレーション・システムを実装するための作業の概要について説明します。

対象読者

本書は、コンサルタントおよび WebSphere InterChange Server 統合コンポーネントおよびシステムの開発を担当する開発者を対象にしています。

本書の前提条件

本書を使用する前に、「テクニカル入門 (IBM WebSphere InterChange Server)」をお読みください。

関連文書

この製品に付属の資料の完全セットは、すべての WebSphere InterChange Server インストールに共通の機能およびコンポーネントを説明するとともに、特定のコンポーネントに関する参照資料についても記載しています。

資料は、以下のサイトからインストールできます。

- InterChange Server の資料:

<http://www.ibm.com/websphere/integration/wicserver/infocenter>

- コラボレーションの資料:

<http://www.ibm.com/websphere/integration/wbicollaborations/infocenter>

- WebSphere Business Integration Adapters の資料:

<http://www.ibm.com/websphere/integration/wbiadapters/infocenter>

上記のサイトには資料のダウンロード、インストール、および表示に関する簡単な説明が記載されています。

注: 本書の発行後に公開されたテクニカル・サポートの技術情報や速報に、本書の対象製品に関する重要な情報が記載されている場合があります。これらの技術情報や速報は、WebSphere Business Integration のサポート Web サイト (<http://www.ibm.com/software/integration/websphere/support/>) で参照できます。適切なコンポーネント領域を選択し、「Technotes (技術情報)」セクションと「Flashes (速報)」セクションを参照してください。

表記上の規則

この資料は下記の規則に従って編集されています。

Courier フォント	コマンド名、ファイル名、入力情報、システムが画面に出力した情報など、記述されたとおりの値を示します。
太字	初出語を示します。
イタリック、イタリック	変数名または相互参照を示します。
青い文字	オンラインで表示したときのみ見られる青の部分は、相互参照用のハイパーリンクです。アウトラインの内側をクリックすると、参照先オブジェクトにジャンプします。
{ }	構文の記述行の場合、中括弧 ({ }) で囲まれた部分は選択対象のオプションです。1 つのオプションのみを選択する必要があります。
[]	構文の記述行の場合、大括弧 ([]) で囲まれた部分はオプションのパラメーターです。
...	構文の記述行の場合、省略符号 (...) は直前のパラメーターが繰り返されることを示します。例えば、option[,...] は、複数のオプションをコンマで区切って入力できることを示します。
< >	命名規則では、不等号括弧は名前の個々の要素を囲み、各要素を区別します。 (例: <server_name><connector_name>tmp.log)
/, ¥	本書では、ディレクトリー・パスの規則として円記号 (¥) を使用します。UNIX システムの場合には、円記号をスラッシュ (/) に置き換えてください。すべての IBM WebSphere InterChange Server 製品のパス名は WebSphere InterChange Server 製品の使用システムへのインストール先ディレクトリーに対する相対パスになっています。
UNIX:/Windows:	これらのいずれかで始まるパラグラフは、オペレーティング・システム間の相違を列挙した注記です。
%text% と \$text	% 記号で囲まれたテキストは、Windows の text システム変数またはユーザー変数の値を示します。UNIX 環境での同等の表記は \$text です。これは、text UNIX 環境変数の値を示します。

本書の使用

本書は次の目的で構成しています。

- 顧客要件の評価からライブ・システムの配置まで、WebSphere InterChange Server の実装に必要な上位レベルの段階について説明します。
- WebSphere InterChange Server ソリューションのコンポーネントおよびインターフェースの開発時や構成時に開発者が実行する作業について順次説明します。
- 特定の開発作業に適切な開発資料またはシステム資料にナビゲートできるハブ資料を提供します。

本書には、新しい情報と、「システム管理ガイド」に以前から記載されていた構成情報の一部の両方が含まれています。WebSphere InterChange Server コンポーネントの開発者およびライブで配置済みのシステムの管理者の両方が実行できる作業については、「システム管理ガイド」を参照してください。

このリリースの新機能

リリース 4.3 の新機能

このリリースでは、製品に対して次の変更が行われました。

- Web サービスが使用可能になりました。詳細については、55 ページの『統合コンポーネントとしての Web サービスの指定』を参照してください。
- 配置プロパティを配置時に構成できるようになりました。詳細については、72 ページの『配置時プロパティの構成』を参照してください。
- セキュリティーのために、役割ベースのアクセス制御が提供されています。詳細については、137 ページの『System Manager を使用したセキュリティーの構成』を参照してください。
- エンドツーエンド・プライバシーを使用してシステムを構成できるようになりました。詳細については、139 ページの『System Manager を使用したプライバシーの構成』を参照してください。
- repos_copy コマンドには新しい 3 つのオプションがあります。これらのオプションの詳細については、143 ページの『第 6 章 repos_copy の使用』を参照してください。
- 統合テスト環境に変更が加えられています。詳細については、39 ページの『InterChange Server のモード』および 273 ページの『第 12 章 統合テスト環境の使用』を参照してください。

リリース 4.2.2 の新機能

このリリースでは、製品に対して次の変更が行われました。

- コンポーネント展開用のワークフローが変更されました。詳細については、77 ページの『配置ウィザードを使用するコンポーネントの展開』を参照してください。
- 今回のリリースでは、コンポーネントの展開は、System Manager でドラッグ・アンド・ドロップを使用して行うことができます。詳細については、83 ページの『ドラッグ・アンド・ドロップを使用するコンポーネントの配置』を参照してください。
- Collaboration Debugger という名前の新しいパースペクティブを使用すると、コラボレーション・オブジェクトの実行を一時停止したり、フローの処理に従ってフローのデータを表示できます。詳細については、337 ページの『第 13 章 Collaboration Debugger の使用』を参照してください。
- 統合テスト環境には、重大な変更が加えられました。詳細については、273 ページの『第 12 章 統合テスト環境の使用』を参照してください。
- 幾つかの System Manager インターフェースには、マイナー変更があります。System Manager の使用については、27 ページの『第 3 章 IBM WebSphere InterChange Server ツールおよび環境』を参照してください。
- 前のリリースで Web ベースの System Monitor と見なされていたツールが、今回は System Monitor と見なされています。前のリリースで Windows ベースの

System Monitor と見なされていたツールの機能が、InterChange Server コンポーネント管理という名前の System Manager のビューに取り込まれました。

リリース 4.2.1 の新機能

このリリースでは、製品に対して次の変更が行われました。

- System Manager は、メモリー・チェッカー・スレッドを構成するためのインターフェースを備えました。詳細については、367 ページの『メモリー・チェッカー・スレッドの使用』を参照してください。
- パッケージを展開する前に、System Manager で検証できるようになりました。詳細については、75 ページの『System Manager を使用したパッケージの検証』を参照してください。
- データベースのデッドロックが発生した場合に、InterChange Server は失敗するのではなくデッドロックしたトランザクションを再試行するようになりました。再試行の回数および各再試行間の間隔を設定できます。詳細については、127 ページの『System Manager を使用したデータベース・プロパティの構成』を参照してください。
- リリース 4.2.0 以前のテキスト・ファイルをインポートする際のエンコード方式を指定するために、新規オプション `-nc` が `repos_copy` に追加されました。詳細については、143 ページの『第 6 章 repos_copy の使用』を参照してください。

リリース 4.2.0 の新機能

このリリースでは、製品に対して次の変更が行われました。

- CrossWorlds System Manager は、Eclipse ベースのツール・フレームワークに置き換えられ、その機能はフレームワーク内の System Manager というプラグインから提供されます。詳細については、27 ページの『第 3 章 IBM WebSphere InterChange Server ツールおよび環境』を参照してください。
- 統合テスト環境という新しいツール・プラグインにより、統合インターフェース全体をテストする機能が提供されます。
- InterChange Server 開発モデルが、前のリリースから大きく変更されました。新しい開発モデルの概要については、1 ページの『第 1 章 IBM WebSphere InterChange Server の開発モデルとプロセス』を参照してください。

このリリースでは、「WebSphere InterChange Server システム・インプリメンテーション・ガイド」に対して次の変更が行われました。

- 環境のセットアップ方法に対する推奨事項を説明するため、15 ページの『第 2 章 環境の準備』が追加されました。
- System Manager および配置手順について説明するため、27 ページの『第 3 章 IBM WebSphere InterChange Server ツールおよび環境』が追加されました。
- InterChange Server の構成に使用可能な 2 つのツールについて説明するため、119 ページの『第 5 章 WebSphere InterChange Server の構成』が追加されました。
- `repos_copy` およびいくつかの共通使用シナリオについて説明するため、143 ページの『第 6 章 repos_copy の使用』が追加されました。

- Visual Test Connector の使用について説明するため、261 ページの『第 11 章 Test Connector の使用』が追加されました。
- パフォーマンスに影響を及ぼす、WebSphere InterChange Server の機能および構成可能なプロパティの一部についての資料をまとめるため、361 ページの『第 14 章 パフォーマンスの調整』が追加されました。これは包括的な参照ではありませんが、今後の情報のための基礎となるよう設計されています。

リリース 4.1.1 の新機能

IBM CrossWorlds 4.1.1 リリースでは、InterChange Server、付属ツール、および関連付けられた API の国際化が行われています。国際化 (I18N) によって、IBM CrossWorlds は、特定の国、地域、および言語特有の国/地域別情報と文字エンコード方式からなるロケールのサポートを提供できるようになりました。これにより IBM CrossWorlds Components は、特定のロケールおよびネイティブ・エンコード方式をサポートするようにローカライズすることができます。

本書では、次の国際化済みのコンポーネントについての説明を提供します。

- InterChange Server (ICS)

システム構成ファイルによって、InterChange Server により使用されるロケールを指定するための新規パラメーター `LOCALE` が提供されます。`LOCALE` が設定されていない場合、InterChange Server では、オペレーティング・システムのロケールが使用されます。

英語以外のシステム構成ファイルについては、翻訳済みのシステム構成ファイルに次のような名前がつけられています。InterchangeSystem_<locale_name>.cfg (ここで、locale_name は適切なロケール名です)

InterChange Server では、ロケールの文字エンコード方式を使用してエラー・メッセージ、トレース・メッセージ、および状況メッセージが生成されます。英語以外のシステム・メッセージ・ファイルについては、翻訳済みのシステム・メッセージ・ファイルに次のような名前がつけられています。

InterchangeSystem_<locale_name>.txt (ここで、locale_name は適切なロケール名です)

- Connector Designer

新しいコネクタ構成プロパティである `Locale` によって、コネクタ・フレームワークで使用されるロケールが指定されます。`Locale` が設定されていない場合、コネクタ・フレームワークでは、オペレーティング・システムのロケールが使用されます。

`CharacterEncoding` コネクタ構成プロパティでは、コネクタ・フレームワークによって使用される文字エンコード方式が指定されます。

現在、多数のコネクタ構成プロパティによって、プロパティ値のネイティブ・エンコード方式がサポートされています。ただし、コネクタ構成プロパティの名前では、英語 (ASCII) 文字のみが使用されます。

リリース 4.1.0 の新機能

4.0.0 リリース以来、「*WebSphere InterChange Server* システム・インプリメンテーション・ガイド」には編集上の変更のみが加えられています。

リリース 4.0.0 の新機能

CrossWorlds 4.0.0 の「*WebSphere InterChange Server* システム・インプリメンテーション・ガイド」はこの資料の最初のリリースです。

第 1 章 IBM WebSphere InterChange Server の開発モデルとプロセス

WebSphere InterChange Server システムには、コア・インフラストラクチャー (InterChanger Server と Business Integration Toolset) と、アプリケーションを統合しビジネス・プロセスを自動化するために連携する構成可能なモジュラー要素が含まれます。これらのモジュラー要素には、本書で**統合コンポーネント**と呼ばれるコネクタ、コラボレーション・テンプレート、コラボレーション・オブジェクト、ビジネス・オブジェクト、データベース接続プール、マップ、関係などが含まれます。

注: InterChange Server および WebSphere 統合ツール・セットは、IBM Tivoli License Manager (ITLM) に対応しており、モニター対象ノードで実行中のソフトウェアを検出できます。ITLM の詳細については、「テクニカル入門」を参照してください。

この章では、WebSphere InterChange Server システム実装の概要について説明します。次のセクションが含まれます。

- 『WebSphere InterChange Server Business Integration システム』
- 3 ページの『WebSphere InterChange Server 開発モデル』
- 8 ページの『実装の段階』

WebSphere InterChange Server Business Integration システム

このセクションでは、以降のセクションで使用する統合概念の一部を定義します。

統合コンポーネント

IBM WebSphere InterChange Server Business Integration システムには、構成可能なモジュラー**統合コンポーネント**の開発および実行をサポートする、コア・インフラストラクチャー (InterChange Server および Toolset) が含まれます。統合コンポーネントはさまざまな役割を果たすので、ユーザーはそれらがともに機能するよう開発します。例えば、アプリケーションと通信するコネクタ、データを保持するビジネス・オブジェクト、ビジネス・プロセス・ロジックを定義するコラボレーション・テンプレートなどです。

インターフェース

通常、ユーザーは**インターフェース**のコンテキスト内で統合コンポーネントを開発します。インターフェースとは、ビジネス・プロセスを自動化するためにともに機能する、統合コンポーネントの集合です。例えば、あるインターフェースは、PeopleSoft と SAP 間の従業員レコードを同期します。別のインターフェースは、Siebel と Oracle 間の注文処理を自動化します。さらに、別のインターフェースは、Siebel と SAP 間の顧客レコードを同期します。通常、インターフェースは、コラボレーション・テンプレートのインスタンスであるコラボレーション・オブジェクト

トを中心として置かれます。このとき、コラボレーション・オブジェクトのポートは、そのインターフェースに対応するコンポーネントにバインドされます。

ソリューション

ユーザーは、ソリューションのコンテキスト内で統合コンポーネントを開発する場合もあります。ソリューションとは、通常は業界全体（小売業や自動車業など）にわたる、ビジネス・プロセスの広範なニーズに応えるよう設計されているコンポーネントの集合であり、インターフェースの集合であることもあります。

統合コンポーネント・ライブラリー

インターフェースとソリューションは、統合コンポーネントの概念的なグループ化です。つまり、コンポーネントは、インターフェースまたはソリューションのコンテキスト内で特定の目的のために機能するという意味でグループ化されます。

一方、統合コンポーネント・ライブラリーは、ユーザーの開発環境におけるコンポーネントのグループ化です。System Manager でライブラリーを定義すると、ライブラリーを表すファイル・システム内にディレクトリーが作成されます。ライブラリー・ディレクトリー内には、統合コンポーネントのタイプごとの多数のサブディレクトリーがあります。統合コンポーネントを作成する場合、それはファイルまたはファイルのグループとして作成され、ライブラリー・ディレクトリー内の適切なサブディレクトリーに格納されます。

ライブラリーには、機能的に作業中のインターフェースに属さないコンポーネントが含まれることがありますが、それらはビジネス・インテグレーション・システム内で別の開発者が作業中のインターフェースに属しています。ライブラリーによって、お客様のサイトで開発中のすべてのインターフェースに含まれるすべてのコンポーネント定義を、ビジネス・インテグレーション・システムの一部として保管できます。作業することのないサイトのシステムに対応するライブラリーを所有することもできます。ライブラリーの作成方法に制約事項はありませんが、統合コンポーネント・ライブラリーと InterChange Server インスタンスが 1 対 1 で関連するようにすることをお勧めします。

ユーザー・プロジェクト

ユーザー・プロジェクトとは、WebSphere InterChange Server Toolset のインターフェースおよびソリューションをサポートする、ソフトウェア実装です。インターフェースは概念的な構造です。ユーザーは自身の設計に合わせて、ビジネス・プロセスを自動化するために関連するコンポーネントが必要なため、コンポーネントをインターフェースの概念にグループ化します。統合コンポーネント・ライブラリーは概念的な構造ではありません (Toolset 内で作成および作業します) が、特に規則正しくなくても構いません。つまり、『統合コンポーネント・ライブラリー』で説明したように、ライブラリーには、ユーザーが作業しているインターフェースに属さないコンポーネントの定義を含めることができます。

ユーザー・プロジェクトは、複数の統合コンポーネントをまとめて編成するニーズをサポートするよう設計されているので、それらが 1 つのインターフェースに属しているように表示されます。ユーザー・プロジェクトは、1 つ以上のライブラリー内の統合コンポーネントへのショートカットの集合なので、各インターフェースごとに 1 つのユーザー・プロジェクトを作成できます。インターフェースはコンポー

ネットを共用することが多いので、各ユーザー・プロジェクトでは、1つのライブラリー内の同一コンポーネントへの複数のショートカットが含まれる場合があります。例えば、SAP アプリケーションに関連する顧客同期インターフェースと注文処理インターフェースには、どちらも SAP 用アダプターが必要なため、これらのインターフェースに対応するユーザー・プロジェクトには、それぞれ SAP コネクターの定義へのショートカットが含まれます。

ビジネス・インテグレーション・システム

インターフェースとソリューションは、それ自体でもビジネス要件を満たすことができますが、一般的には**ビジネス・インテグレーション・システム**のコンテキスト内で開発されます。ビジネス・インテグレーション・システムは、お客様のサイトのビジネス・インテグレーションおよび自動化の要件を満たすために必要な、インターフェースとソリューションの集合です。ビジネス・インテグレーション・システムの開発が発生するのは、より差し迫った要件を満たすように、インターフェースが優先順位別にグループ化されたり優先順位に沿って開発および生産されると同時に、実際の目標やスケジュールを維持する場合です。

InterChange Server インスタンス

WebSphere InterChange Server は、ビジネス・プロセスの統合および自動化を実現するソフトウェア・インフラストラクチャーです。コンピューターに InterChange Server をインストールしてサーバーを始動する場合、サーバー・**インスタンス**を開始することになります。サーバー・インスタンスは、お客様の統合要件を満たすインターフェースすべてから構成される**ビジネス・インテグレーション・システム**をホストします。『WebSphere InterChange Server 開発モデル』で説明するように、プロジェクトにかかわる開発者は、自身の InterChange Server のローカル・インスタンスを保持する必要があります。また、少なくとも、ビジネス・インテグレーション・システム全体をテストしたり、システムの生産リリースをホスティングしたりする専用の InterChange Server のインスタンスを持つ必要があります。

WebSphere InterChange Server 開発モデル

このセクションでは、ビジネス・インテグレーション・システムを開発するために従う、開発モデルについて説明します。

インターフェースのライフ・サイクル

多くの場合、1人の人がインターフェース全体を開発することは理にかなった方法です。8 ページで説明する実装の段階に従って、個々の統合コンポーネントが連動してインターフェース要件を満たすように開発してください。

ローカルでのインターフェースの開発

統合コンポーネントを開発する場合、WebSphere InterChange Server Toolset を使用します。

まず始めに、開発環境を準備します。これには、WebSphere InterChange Server をローカル・システムにインストールすることが含まれます。詳細については、15 ページの『第 2 章 環境の準備』を参照してください。

次に、コンポーネント定義を保管する統合コンポーネント・ライブラリーと、インターフェースを表すユーザー・プロジェクトを定義します。インターフェース用の統合コンポーネントをライブラリーに作成および保管し、必要に応じて、それらのショートカットをユーザー・プロジェクトに追加します。詳細については、44 ページの『統合コンポーネント・ライブラリーの処理』および 47 ページの『ユーザー・プロジェクトの処理』を参照してください。

ローカルでのインターフェースの配置

統合コンポーネントを開発している間、それらはローカル・ファイル・システム内に存在しているだけにすぎません。インターフェースが完了したら、コンポーネント定義がリポジトリに保管されているローカルの InterChange Server インスタンスに、ユーザー・プロジェクトを配置します。リリース 4.2.0 より前の InterChange Server では、リポジトリ内でコンポーネント定義を直接処理しましたが、本リリースでは、ローカル・ファイル・システム内でコンポーネント定義を排他的に処理してから、それらをサーバーに配置するようになりました。この開発モデルは、「静的メタデータ管理」といいます。これは J2EE 開発モデルを反映しており、開発アクティビティーを実行時アクティビティーから分離します。ユーザー・プロジェクトの配置の詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

ローカルでのインターフェースのテスト

インターフェースをローカルの InterChange Server インスタンスに配置したら、インターフェースをテストして、お客様の要件に一致するかどうか検証する必要があります。テストの詳細については、261 ページの『第 11 章 Test Connector の使用』および 273 ページの『第 12 章 統合テスト環境の使用』を参照してください。

ローカルの InterChange Server インスタンスでインターフェースの健全性を検証したら、プロジェクトを担当する他の開発者と協調して、そのインターフェースを他の開発者のインターフェースにビジネス・インテグレーション・システムとして結合します。詳細については、『ビジネス・インテグレーション・システムのライフ・サイクル』を参照してください。

ビジネス・インテグレーション・システムのライフ・サイクル

3 ページの『インターフェースのライフ・サイクル』では、ローカル・システムでインターフェースを開発する方法について説明しました。このセクションでは、他の開発者と協力してビジネス・インテグレーション・システムを送達する方法について説明します。

開発統合環境への配置

ローカル・システムでインターフェースの開発およびテストが終了したら、プロジェクト内でインターフェースを使用する他の開発者と協調して、ビジネス・インテグレーション・システムを作成する必要があります。どの開発者も、自身のインターフェースを共通環境 (開発統合環境ともいわれます) に配置する必要があります。

これを実行するには、開発統合環境で統合コンポーネント・ライブラリーを作成し、次に各開発者が自身のインターフェース内のコンポーネントが含まれるパッケージをエクスポートし、そのパッケージを新規ライブラリーにインポートします。

すべての冗長度を解決し、変更を行ったら、インターフェースの集合を表すコンポーネントを、開発統合環境内のローカル・サーバー・インスタンスに配置します。

統合テストの実行

ビジネス・インテグレーション・システム (またはシステム実装の特定の段階) のインターフェースを結合したら、再度テストする必要があります。これによって、各インターフェースを他のインターフェースと一緒に実行するときに、お客様が定義したビジネス要件を満たしているか確認します。ローカル開発環境から統合開発環境への移行において問題や見落としがあった場合 (共用コンポーネントを適切に更新できないなど)、この段階の厳密なテストによって識別されます。

統合テストは、サイトに応じて、開発統合サーバーまたは専用テスト環境内で実行されます。個別のテスト環境を保持している場合は、開発統合環境内のインターフェースをすべて結合し、その整合性を検証してから、テスト環境を変更することができます。このようにして、テスト環境で既存のインターフェースをテストすると同時に、開発統合環境で新規のインターフェースを作成できます。ただし、この方法では、お客様が別の環境をサポートするために必要な装置を購入し、環境を構成する必要があります。

パフォーマンス・テストの実行

統合テストを実行してインターフェースの集合が機能要件を満たしていることを検証したら、負荷テストを実行して、ビジネス・インテグレーション・システムがお客様のパフォーマンス要件に一致するか確認できます。

実稼働環境に等しいパフォーマンス・テスト環境を構成している場合、シミュレートされるスループットは、実動パフォーマンスにより近い予測になります。ただし、実稼働環境にコストがかさみ、実動に近づけてミラーリングするパフォーマンス・テスト環境の作成に、お客様が投資できない場合があります。

ビジネス・インテグレーション・システムをパフォーマンス・テスト環境に移行するには、開発統合環境に移行したときと同じ手順を実行します。

- ビジネス・インテグレーション・システムのコンポーネントが含まれるパッケージをエクスポートします。
- 負荷テスト環境に統合コンポーネント・ライブラリーを作成します。
- パッケージを新規ライブラリーにインポートします。
- 環境固有のプロパティの必要に応じて、コンポーネント定義を更新します。
- 変更済みコンポーネントのパッケージを、負荷テスト・サーバーのリポジトリに配置します。

パフォーマンスが十分でない場合、一部のコンポーネントを変更または再構成する必要があります。パフォーマンスに影響する構成および技法の詳細については、361 ページの『第 14 章 パフォーマンスの調整』を参照してください。

実動の開始

ビジネス・インテグレーション・システムが機能要件およびパフォーマンス要件を満たすことを検証したら、インターフェースの集合を実稼働環境に移行します。

ビジネス・インテグレーション・システムを実稼働環境に移行するには、前の環境に移行したときと同じ手順を実行します。

- ビジネス・インテグレーション・システムのコンポーネントが含まれるパッケージをエクスポートします。
- 実稼働環境に統合コンポーネント・ライブラリーを作成します。
- パッケージを新規ライブラリーにインポートします。
- 環境固有のプロパティの必要に応じて、コンポーネント定義を更新します。
- 変更済みコンポーネントのパッケージを、実動サーバー・リポジトリに配置します。

7 ページの図 1 に、一般的な環境における統合コンポーネントの、統合コンポーネント・ライブラリーからユーザー・プロジェクト、サーバー・リポジトリ、およびエクスポート済みパッケージへのフローを示します。

開発チームは、サイト管理者がインターフェースの集合を管理するために必要なスキルや情報を取得するよう、知識を伝達する必要があります。これで統合実装を完了するか、または次の段階の一部として新規インターフェースの開発を開始します。

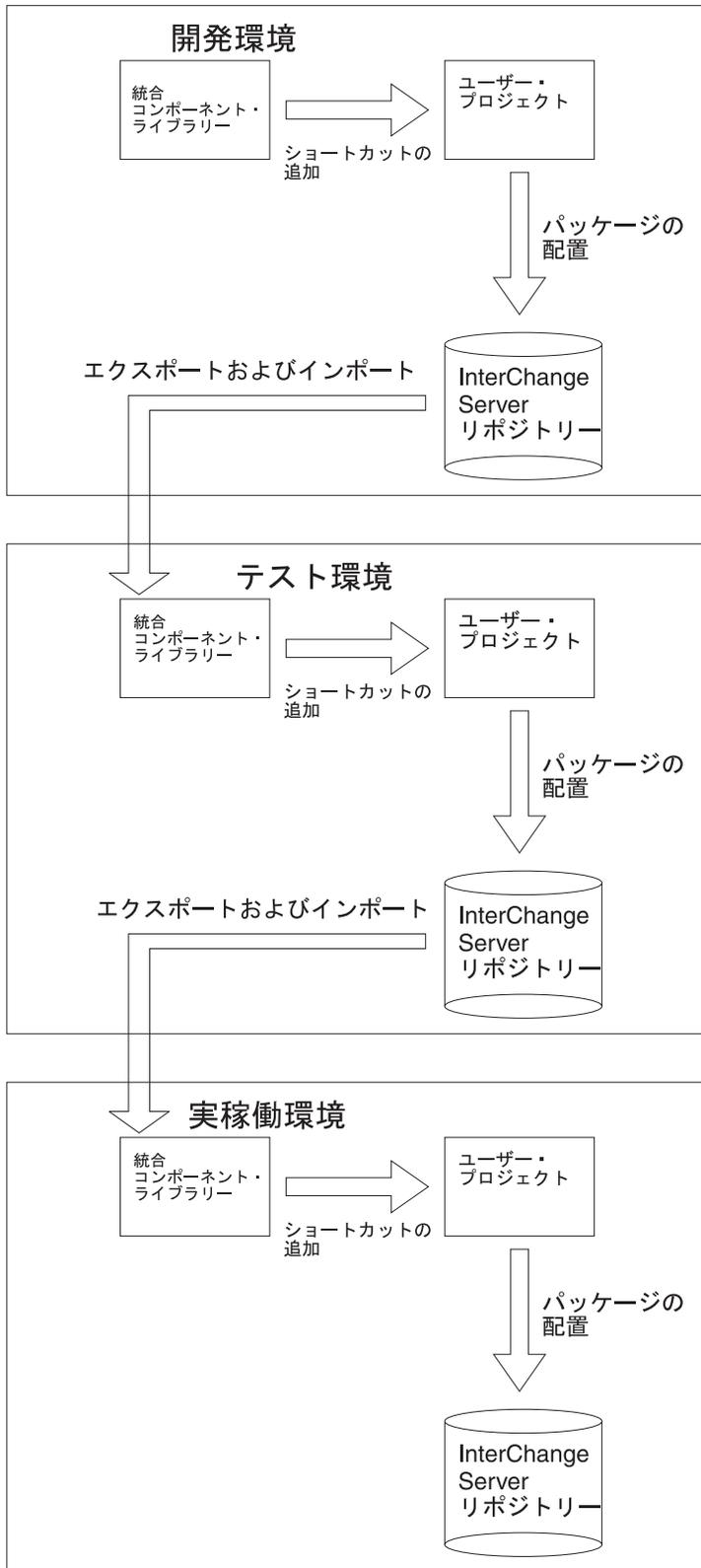


図 1. WebSphere InterChange Server 開発モデル

実装の段階

ビジネス・インテグレーション・システムの開発は、段階ごとに実行することをお勧めします。各段階で生成される成果物の厳密な詳細、性質、およびタイミングは、実装を実行する組織によって異なる場合があります。ただし、通常、ビジネス・インテグレーション・システムの開発には、次の段階が含まれます。

- 『要件の発見および評価』
- 10 ページの 『環境の準備』
- 10 ページの 『設計』
- 10 ページの 『開発および構成』
- 11 ページの 『検証』
- 12 ページの 『配置』

要件の発見および評価

この段階では、プロジェクトのビジネス・ゴール、システム要件、および開発作業の全体的な範囲を識別することにより、実装プロセスが開始されます。

発見段階は、上位レベルから開始され、その後詳細な下位レベルに移行します。この段階は、次の上位レベルの問題を検討することで開始する必要があります。

- お客様にとって解決する必要がある具体的なビジネス上の問題は何か。

この問題への回答は重要です。これによって、テスト段階におけるインターフェースの機能要件が確立されるためです。テストによってビジネス上の問題が解決されていないことが判明する場合、インターフェースが正しく設計または開発されなかったこととなります。

- ビジネス上の問題に対処するために統合または自動化する必要がある、企業レベルのビジネス・プロセスは何か。

次の質問および関係があると思われるその他の質問をします。

- 統合が必要なアプリケーションの名前と特定のバージョンは何か。
- ソース・アプリケーションはどれですか。
- 宛先アプリケーションはどれですか。
- 記録システムのアプリケーションはどれですか。
- アプリケーションは稼働中ですか、それとも実装中ですか。
- アプリケーションはソフトウェア・ベンダーが開発したものですか、それとも社内の開発者が開発したものですか。アプリケーションがお客様のサイトの開発者によるカスタム製品である場合、それらの開発者を投入できますか。
- アプリケーション、データベース、API など、ビジネス・プロセスを統合させる必要がある技術的な環境は何か。

技術的な環境の特徴を判別します。次の各項目を調べてください。

- データベースのベンダーおよびバージョン
- プラットフォーム、オペレーティング・システム、およびバージョン
- アプリケーションに対して存在する API

- すべてのアプリケーション・クライアントとサーバー・プラットフォームの場所
- ネットワーク環境
- 予想されるトランザクションのボリューム

実装に必要なインターフェースと使用するコンポーネントを識別するために詳細な下位レベルの情報を調査して、実装する特定のビジネス・プロセス、必要なビジネス・ロジックと必要なデータ変換、およびインターフェースが対話するアプリケーションやデータベースの詳細を識別し、これらについて説明します。調査の際、次の情報収集作業を行う場合があります。

- ビジネス上の問題を解決するために統合または自動化する必要があるビジネス・プロセスを識別し、説明します。その際、次の質問をします。
 - ビジネス・プロセスに含まれる、自動タスクおよび手動タスクの現行シリーズは何ですか。
 - どのイベントがプロセスを開始しますか。
 - ビジネス・プロセスに関与する人は誰ですか。
 - ビジネス・プロセスに影響を受ける人 (エンド・ユーザー) は誰ですか。
 - 入力データと出力データは何ですか。
 - データに前提条件や依存関係は存在しますか。
 - フィルター要件が存在しますか。
 - 複数の宛先アプリケーションがある場合、処理されるエンティティの経路指定方法を決定するのは何ですか。
 - インターフェースは両方向ですか。
 - プロセスが発生する頻度はどれくらいですか。
 - トランザクション・プロセスを完了させる必要がある時間フレームは存在しますか。その他のプロセスは、時間フレームによって処理されるエンティティおよび正常終了に依存しますか。
 - データのボリュームはどのくらいですか。
 - ビジネス・プロセスはリアルタイムで発生しますか、それともバッチ単位で発生しますか。
 - インターフェースは同期 (開始プログラムが応答を要求する) ですか、それとも非同期ですか。
 - エラーへの応答の手順はどのようなものがよいですか。
- インターフェースで処理するエンティティの構造を記述します。
- ソースと宛先のアプリケーション・エンティティ構造間で実行する必要があるデータ形式変更を識別します。
- Holosofx などのツールを使用して、フロー・チャートでビジネス・プロセス・フローを図示します。フロー・チャートを使用すると、実行する必要がある機能の分析が可能になり、後で既存のコラボレーション・テンプレートと比較する際に役立ちます。これにより、使用するコラボレーションおよび必要な修正レベルの決定に役立ちます。

この段階は、IBM WebSphere InterChange Server の資料で広範には説明しません。このセクションにはいくつかの質問が提示されていますが、要件を導き出すプロセ

スはコンサルティング組織によってかなり異なります。適切な方法にはすべて従ってください。他の段階は、特定のコンサルティング手法にはそれほど依存せず、ソフトウェアにより依存します。

環境の準備

この段階では、お客様にビジネス・プロセス統合ソリューションを提供するために必要な環境を準備します。

通常、各開発者は、インターフェースを開発するために独自の開発環境を準備します。

システム設計者は、開発統合環境、負荷テスト環境、実稼働環境を準備および管理することをお勧めします。

環境の準備の詳細については、15ページの『第2章 環境の準備』を参照してください。

設計

評価および設計段階は、発見段階で収集した詳細な情報に依存しています。

インターフェースおよびそれを構成する統合コンポーネントの詳細な要件を決定すると、既存のコンポーネントがニーズを満たすかどうかを評価する準備ができます。一部の要件に対してはコンポーネントがすでに存在し、そのまま使用できる場合もありますが、要件によって、既存のコンポーネントを拡張する（ニーズに合わせて変更する）必要がある場合や、新規の（カスタム）統合コンポーネントを作成する必要がある場合もあります。

各コンポーネントは、個々に評価すると同時に、全体のインターフェース内での他のコンポーネントとのかかわり方を見て評価します。あるコンポーネントがインターフェース内で対話する別のコンポーネントの設計を開始するまでは、そのコンポーネントの設計を完了することはできません。

コンポーネントの設計に関する詳細な情報については、次のガイドを参照してください。

- 「コネクタ開発ガイド (C++ 用)」
- 「コネクタ開発ガイド (Java 用)」
- 「ビジネス・オブジェクト開発ガイド」
- 「マップ開発ガイド」
- 「コラボレーション開発ガイド」

開発および構成

この段階では、『設計』段階で派生した仕様に基づいて統合コンポーネントを開発します。

インターフェースを開発するには、使用可能な既存のコンポーネント（コラボレーション・テンプレートやビジネス・オブジェクトなど）を変更し、環境固有の新規のコンポーネント（マップなど）を作成します。

開発および構成の段階は反復性があるので、進捗に合わせてコンポーネントを再開発したり構成を変更したりする必要がある場合があります。

コンポーネントを開発するときは、単体テストを行って、設計したとおりにインターフェース内で役割を果たすかどうか確認することをお勧めします。インターフェースですべてのコンポーネントを開発および単体テストした後で、ストリング・テストを実行し、インターフェース全体が設計したとおりに動作するかどうか検証します (少なくともローカル開発環境で実行します)。

109 ページの『第 4 章 ビジネス・プロセス・インターフェースの開発』で説明するように、開発および構成は通常、規定の手順に準じて実行します。

コンポーネントの開発に関する詳細な情報については、次のガイドを参照してください。

- 109 ページの『第 4 章 ビジネス・プロセス・インターフェースの開発』
- WebSphere Business Integration Adapters ソフトウェア・リリースのアダプター・ガイド
- 「コネクタ開発ガイド (C++ 用)」
- 「コネクタ開発ガイド (Java 用)」
- 「ビジネス・オブジェクト開発ガイド」
- 「マップ開発ガイド」
- 「コラボレーション開発ガイド」
- 159 ページの『第 7 章 コネクタの構成』
- 219 ページの『第 9 章 コラボレーション・オブジェクトの構成』

検証

検証段階では制御されたテスト環境でシステム・テストを実行し、すべてのインターフェースにおいて、要件の発見および評価段階で作成された要件が、開発時のように確実に満たされるようにします。必要に応じて、機能、パフォーマンス、回帰のテストを実行します。

• 機能テスト

機能テストは、ビジネス・インテグレーション・システムのインターフェースが、プロジェクトにおけるビジネス・プロセスの自動化および統合の目標を達成しているか確認するために実行します。

• パフォーマンス・テスト

パフォーマンス・テストは、スループット、応答時間、および待ち時間がお客様の要求を満たすか確認するために実行します。

• 回帰テスト

機能テストまたはパフォーマンス・テストの結果としてコンポーネントを変更する場合 (例えば、ビジネス要件を満たさないコラボレーション・テンプレートの変更や、適切に動作しないコネクタの変更など)、回帰テストを実行して、行った変更によってコンポーネントが要件を満たさない結果にならないよう確認する必要があります。

検証段階では、次の操作を実行します。

- すべての機能要件およびパフォーマンス要件を示す、要件マトリックスを作成します。
- システム・テスト環境を準備します。
- 機能要件およびパフォーマンス要件に対処する、一連のテスト・ケースを識別および開発します。
- ビジネス・インテグレーション・システムを、開発統合環境からテスト環境に移行します。
- システム・テスト計画および手順を開発、調整し、クライアントからの同意を得ます。各テスト・マトリックス項目に対して特定のテストを割り当てます。
- テスト・マトリックスに記述されている機能テストおよびパフォーマンス・テストを実行します。
- テストによって明らかになった問題や不備な点を識別、文書化、および解決します。
- 必要に応じて、回帰テストを実行します。
- テストおよびそれに伴う構成や再開発についてのレポートを作成および公開します。
- クライアントから、開発したシステムに対する承諾を得ます。

コンポーネントおよびインターフェースのテストの詳細については、次を参照してください。

- 261 ページの『第 11 章 Test Connector の使用』
- 「マップ開発ガイド」

配置

配置段階の目的は、ビジネス・インテグレーション・システムを実稼働環境に移行し、実稼働 InterChange Server インスタンスを開始することです。

配置段階では、次の操作を実行します。

- 全体的な配置計画を開発して、クライアントから承諾を得ます。既存のシステムおよびプロセスを使用して中断せずに業務を継続できるようにします。
- 開発済み、テスト済みのシステムをテスト環境から実稼働環境に移行します。
- クライアント固有のビジネス・プロセスおよびシステム統合に必要なデータ変換要件を取得して文書化します。
- 固有の実稼働環境要件 (E メール・アドレス、アプリケーション・データベース・リソース、パスワードなど) をサポートするために必要な変更を開発、実装およびテストします。
- ビジネス・インテグレーション・システムを実稼働環境にインストールおよびテストします。
- コンポーネントおよび実稼働環境を変更して、特定の環境に合わせてパフォーマンスを最適化します。
- クライアントから、配置した実動システムに対する承諾を得ます。

実稼働への移行の詳細については、次を参照してください。

- 27 ページの『第 3 章 IBM WebSphere InterChange Server ツールおよび環境』
- 「システム管理ガイド」

第 2 章 環境の準備

この章では、開発、テスト、および実動の各段階でインターフェースをとるために必要になるさまざまな環境の準備方法について説明します。

この章では、この後に記載されている章を参照しています。それらの章では、環境の準備に使用する必要があるツールについて説明しています。実装のライフ・サイクルを本書に反映させるため、サイトでの環境の準備は初期設計の後に、開発は開始する前に行うことから、この章はここに置かれています。この章は、一度本書を通読してから再度参照してください。

この章を構成するセクションは次のとおりです。

- 『個々の開発環境の準備』
- 22 ページの『開発統合環境の準備』

個々の開発環境の準備

3 ページの『インターフェースのライフ・サイクル』で説明しているように、通常、1 人が開発の責任を負うビジネス・プロセス・インターフェースは 1 つであり、各個人は専用のラップトップなどのコンピューターに準備した開発環境で作業します。コンピューターに必要なソフトウェアをインストールし、構成するために必要な大部分の情報は「システム・インストール・ガイド (Windows 版)」に記載されていますが、このセクションでは、効率的かつ効果的な開発のための環境の変更に関するヒントについて説明します。

必要なソフトウェアのインストール

IBM WebSphere InterChange Server は統合インフラストラクチャーを提供するために多くのエンタープライズ・クラス・ソフトウェアを必要としますが、システムから特に多くのトランザクションを処理しない限り、通常は必要なソフトウェアをすべてワークステーションまたはラップトップ型コンピューターで実行できます。コンポーネントの開発および単体テストをサポートするにはこのようなコンピューターが最適ですが、大量のフローをサポートすることはできません。以下のリストに、いくつかの必要なソフトウェアを示します。

- IBM WebSphere InterChange Server

InterChange Server はローカル・システムにインストールし、インターフェースを開発できるようにする必要があります。

- IBM WebSphere Business Integration Adapters

インターフェースでアプリケーションと通信するために必要なアダプターをインストールし、コネクターの構成の検査およびアプリケーションへの接続の検証が実行できるようにすることをお勧めします。一部のアダプターでは、アプリケーション・クライアントもインストールする必要があります。その他のインストール要件については、インターフェースに必要なアダプターのガイドを参照してください。

- IBM Java オブジェクト・リクエスト・ブローカー (ORB)

IBM ORB をインストールし、ツールやコネクタなどのクライアントが InterChange Server と通信できるようにする必要があります。

- WebSphere Workbench (IBM WebSphere InterChange Server に付属の Eclipse ベースのフレームワーク。InterChange Server のインストール時に Toolset のインストールを選択するとインストールされます。)

WSWB または WSADIE を使用すると、System Manager および統合テスト環境プラグインをサポートできます。WSWB はプラグインに必要なサポートをすべて提供し、WSADIE ほど高価なリソースではないため、WSWB の使用をお勧めします。

- WebSphere Application Server または Jakarta Tomcat

System Monitor は主に実動システムの管理者のために設計されていますが、コンポーネントの状態を管理する機能が豊富であり、開発段階でも役立ちます。通常、Tomcat の方がシステム・リソースに対する要件が低くなります。

- データベース・サーバー

プロジェクトの各開発者は、通常専用の開発環境を持ち、専用のリポジトリ・データベースを必要とします。サイトの単一のデータベース・サーバー・ホストに開発者ごとにデータベースを作成することもできますが、データベース・サーバーを各コンピューターにインストールしてそのコンピューターでリポジトリに対するホスト・サービスを提供することをお勧めします。これにより、サイトのデータベース・サーバーにアクセスできない場合でも開発を継続できます。

コンポーネントはすべてローカル・ファイル・システムで開発するため、サーバー・リポジトリにアクセスできない場合でも引き続き開発できます。ただし、テストは頻繁に行うことをお勧めしますが、リポジトリがなければコンポーネントをテストできないため、リポジトリを使用可能にしておくことをお勧めします。

- WebSphere MQ

WebSphere MQ はデータの効率的なパーシスタンスを提供するため、実稼働環境では WebSphere MQ をお勧めしますが、開発環境では必須ではなく、システム・リソースの消費が多くなる場合があります。個々の開発環境では、WebSphere MQ の代わりに IIOP を使用することをお勧めします。これを行うには、Connector Configurator を使用し、コネクタの **DeliveryTransport** プロパティを値 IDL に設定します。Connector Configurator およびコネクタの標準プロパティについては、159 ページの『第 7 章 コネクタの構成』を参照してください。

インストールの推奨事項

必要なソフトウェアをインストールする場合は、以下の事項を考慮してください。

- IBM WebSphere InterChange Server をインストールするときには、サーバー・インスタンスの名前を指定します。サーバーはクライアント (ツールおよびコネクタなど) が通信する必要がある名前付き CORBA オブジェクトとして公開されるため、サーバー・インスタンスはネットワーク内で固有である必要があります。同じ名前のオブジェクトがネットワークに複数存在する場合は、クライアン

トは適切なサーバーを検出できません。ローカル・マシンにインストールするときにサイトの開発者全員がデフォルトのサーバー名を受け入れた場合は、多くのサーバー名が重複してしまいます。

コンピューターの名前もネットワーク内で固有でなければならないため、インストール時にはコンピューターの名前を `InterChange Server` インスタンスの名前として指定することをお勧めします。

さらに、サーバー名に他の情報も含めておくと便利です。例えば、`InterChange Server` および `WebSphere MQ Integrator Broker` タイプのプロローカーの両方を定期的に処理する場合は、サーバー名に `WICS` を含めておくと、プロローカーのタイプを識別できます。システムの異なるリリースから複数のサーバー・インスタンスをインストールする場合は、数値 `420` を含めてリリース・バージョンを示すことができます。`DEV` を含めておくと、サーバーがテスト環境や実稼働環境のサーバーではなく、開発サーバーであることを示すことができます。

- `InterChange Server` インスタンスに指定した名前と同じ名前のディレクトリーに製品をインストールすると便利です。このようにすると、同じシステムに製品を数回インストールする場合にインストールを相互に区別できます。
- インストール時に「スタート」メニューに作成されるプログラム・グループの名前に `InterChange Server` インスタンスの名前を含めるのも有用です。
- インストールが必要なソフトウェアの多くは、`PATH` および `CLASSPATH` 環境変数にエントリーを追加します。Windows システムではこれらの変数に 512 文字の制限があるため、文字スペースを使い尽くしてしまうと、プログラムを始動できなくなる可能性があります。この制限を克服するには、アプリケーションのフォルダーを共用し、ネットワーク・ドライブをマップし、元のインストール・ディレクトリーに対する `PATH` および `CLASSPATH` エントリーをマップされたネットワーク・ドライブで置き換えます。

例えば、以下のディレクトリーに `InterChange Server` をインストールするとします。

```
D:¥ProgramFiles¥IBM¥WICSInstallations¥WebSphereICS420DEV
```

`WICSInstallations` ディレクトリーを共用し、`F:` などのネットワーク・ドライブをマップし、`PATH` および `CLASSPATH` エントリーで `F:¥WebSphereICS420DEV` を使用できます。

設計モードでの `InterChange Server` の処理

ローカル `InterChange Server` は設計モードで開始することをお勧めします。これにより、統合コンポーネントを段階的に組み立てることができます。実動モードで実行した場合はこのような組み立ては不可能です。詳細については、39 ページの

『`InterChange Server` のモード』を参照してください。

バッチ・ファイルの変更

ローカル開発環境で `start_server.bat` バッチ・ファイルを以下のように変更してください。

- バッチ・ファイルの終わりにある `endlocal` コマンドの上にあるブランク行に、`pause` コマンドを追加します。

デフォルトでは、start_server.bat ファイルは、InterChange Server の始動時にエラーが発生するとすぐに終了するように作成されます。多くの場合、サーバーのエラーは最初の始動時に発生しますが、問題を示すエラー・メッセージがコンソール・ウィンドウに表示されても、そのメッセージを読む前にコンソール・ウィンドウが終了してしまいます。バッチ・ファイルに pause コマンドを追加すると、InterChange Server を始動できないエラーが発生した後、コンソール・ウィンドウにエラー・メッセージが表示された状態になります。変更を行うと、ファイルの終わりは下の例のようになります。

```
%CWJAVA% -Djava.ext.dirs=%JRE_EXT_DIRS%;"%MQ_LIB%";"%DB2_LIB%"
-Duser.home="%CROSSWORLDS%" -mx%CW_MEM_HEAP%m -DTEAgent=I200
-DCW_MEMORY_MAX=%CW_MEM_HEAP% %ORB_PROPERTY% -classpath %JCLASSES%
ServerWrapper -s%SERVERNAME% %2 %3
```

```
pause
```

```
endlocal
```

- クライアントがサーバーと通信するためには、データベース・サーバー、WebSphere MQ、IBM ORB などの前提条件ソフトウェア・アプリケーションが実行されている必要がありますが、必ずしもサーバーを実行しておく必要はありません。これらの前提条件ソフトウェア・アプリケーションが不要なときに実行しておくとは非効率的になる場合があります。サード・パーティー・ソフトウェアのサービスは手動サービスとして構成し、対応する net start コマンドをサーバーのバッチ・ファイルに追加することをお勧めします。例えば、start_server.bat バッチ・ファイルの最初の setlocal コマンドの後に以下のコマンドを追加すると、バッチ・ファイルの残りの部分を実行する前に、WebSphere MQ および DB2 サービスを開始できます。

```
setlocal
net start "IBM MQSeries"
net start "DB2 - DB2-0"
net start "DB2 License Server"
net start "DB2 Remote Command Server"
net start "DB2 Security Server"
net start "DB2DAS - DB2DAS00"
```

WebSphere Application Server を System Monitor のホストとする場合は、そのサービスおよび必要なサービスを開始し、モニター対象のアプリケーションの URL を自動的にブラウザで開くバッチ・ファイルを作成できます。このようなバッチ・ファイルの例を以下に示します。

```
net start "DB2 - DB2-0"
net start "DB2 License Server"
net start "DB2 Remote Command Server"
net start "DB2 Security Server"
net start "DB2DAS - DB2DAS00"
net start "IBM HTTP Administration"
net start "IBM HTTP Server"
net start "IBM WS AdminServer 4.0"
start iexplore http://devserver/ICSMonitor
```

注: サービス名や URL は環境に適した設定で置き換えてください。

開発環境でのロギングの構成

以下の理由により、サーバーのロギング出力は、コンソールと開発環境のファイルの両方に送信することをお勧めします。

- 出力をコンソールに送信すると、マウスでクリックすることによってコンソールに「マークを付ける」ことができるため、便利です。これにより、プロセスは一時停止します。これを使用すると、システムからフローを送信し、コンソールでエラーの兆候を監視し、コンソールにマークを付けることができるため、ロギング・ステートメントによって画面からエラー・テキストが消えてしまう前にプロセスを一時停止できます。

注: エラーの調査が終わったら、必ずコンソールの「マークを解除」してください。インターフェースをデバッグし、コンソールにマークを付けてエラーを検討するのは通常の作業ですが、忘れずにマークを解除してください。

InterChange Server プロセスはマークが解除されるまで一時停止したままになるため、このままではリポジトリでのコンポーネントのコンパイルや展開アクションの実行が不可能です。このようにプロセスが停止している場合、他に原因が考えられないときは、InterChange Server コンソールにマークが付いていないかどうか確認してください。

- コンソールの他、ログ・ファイルにも出力を送信すると、テクニカル・サポートに問い合わせる必要がある場合に関連するエラー情報が得られるため、便利です。エラー・テキストがコンソールのみ存在し、異常終了した場合は、テクニカル・サポートが問題のトラブルシューティングを行うために必要とする情報が残りません。

InterChange Server のロギングおよびトレースの構成方法については、119 ページの『第 5 章 WebSphere InterChange Server の構成』を参照してください。

環境のカスタマイズ

ほとんどの開発者は、以下のセクションで説明するように環境をカスタマイズすると作業が効率的になります。

ショートカットの編成

ほとんどの開発者にとっては、頻繁に使用するショートカットを編成し、プログラム・メニューを使用する頻度を少なくすると便利です。デスクトップにショートカット用のフォルダーを作成するか、Windows のタスクバーにカスタム・ツールバーを作成できます。InterChange Server、必要なコネクター、WSWB (System Manager)、データベース・サーバー管理ツール、WebSphere MQ コンソール、System Monitor、テキスト・エディターなどへのショートカットを組み込むと便利です。

ショートカットのプロパティの変更

デフォルトでは、InterChange Server およびコネクターは、サイズおよび画面バッファが小さく、読みにくいカラー・スキームのコンソール・ウィンドウで実行されます。それらのショートカットを変更するには、以下の手順を実行します。

1. InterChange Server またはコネクターのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティ」を選択します。
2. 「レイアウト」タブをクリックし、以下のように変更します。
 - 「画面バッファ・サイズ (Screen buffer size)」ペインの「高さ (Height)」フィールドを値 9999 に設定します。

これにより、コンソールに多くの情報がキャッシュされるため、システムがイベントを処理しているときにエラー・メッセージが置換されるのが遅くなります。

- 「ウィンドウ・サイズ (Window size)」ペインで「幅 (Width)」フィールドを値 120 に設定し、「高さ (Height)」フィールドを値 25 に設定します。

これにより、横方向に表示される情報が多くなり、縦方向のスペースが小さくなるようにコンソールのサイズが設定されます。通常、高さが 25 のコンソールは 1 画面に 3 つ配置できます。これにより、通常インターフェースでテストに必要な InterChange Server および 2 つのコネクターのコンソール・ウィンドウを容易に並べて表示できます。

3. 「カラー」タブをクリックし、以下のように変更します。
 - 「画面の背景」ラジオ・ボタンを選択し、色を選択します。
 - 「画面の文字」ラジオ・ボタンを選択し、色を選択します。

注: 画面の背景に暗い色を使用し、画面のテキストに明るい色を使用するのが最も有用です。InterChange Server とコネクター・コンソールに別々のカラー・スキームを使用すると、それぞれを容易に識別できます。

4. 「OK」をクリックします。

ツール設定の構成

98 ページの『Eclipse ベースのワークベンチの使用』で説明しているように、ツール設定を構成する必要があります。

ワークスペースの設定

ワークスペース (ワークベンチで作成したプロジェクトがデフォルトで格納されるディレクトリー) は、ニーズに最適なロケーションに設定することをお勧めします。ワークスペース・ディレクトリーを指定するには、以下の手順を実行します。

1. 製品ディレクトリーの bin ディレクトリーにナビゲートします。
2. **startesm.bat** という名前のファイルを編集します。
3. **-data** オプションの後の値を、デフォルト・ワークスペースにするディレクトリーに設定します。例えば、WebSphere InterChange Server 製品ディレクトリーの **Projects** という名前のディレクトリーにワークスペースを設定するには、ファイルは以下のようになります。

```
setlocal
set WSWB_EXECUTABLE=%1
"%WSWB_EXECUTABLE%" -data "%CROSSWORLDS%/Projects -vmargs
-Xbootclasspath/p:"%CROSSWORLDS%¥lib¥vbjorb.jar
-Dorg.omg.CORBA.ORBClass=com.inprise.vbroker.orb.ORB
-Dorg.omg.CORBA.ORBSingletonClass=com.inprise.vbroker.orb.ORBSingleton
-DCWTools.home="%CROSSWORLDS%¥bin
```

4. ファイルの保管して閉じます。

統合コンポーネント・ライブラリーの作成

以下の統合コンポーネント・ライブラリーを作成することをお勧めします。

- 対話する必要がある InterChange Server インスタンスごとに 1 つ。ライブラリーとサーバー・インスタンスの間には 1 対 1 の相関関係が必要なため、ローカル開発環境は対象のリポジトリをミラーリングします。

このタイプのライブラリーには、対応するサーバーに関連する方法で名前を付けることをお勧めします。例えば、サーバー名が WICS420DEV の場合は、ライブラリーに WICS420DEVICL と名前を付けます。同様の規則は、『ユーザー・プロジェクトの作成』のユーザー・プロジェクトに対してもお勧めします。

- 「ステージング」ライブラリー。このライブラリーは、サーバー環境に明示的に属さないコンポーネントのコンテナとして使用できます。この内容には、サンプル、デモンストレーション用データ、部分的に開発されたコンポーネント、複数のライブラリーにわたって頻繁に再利用するコンポーネントなどを含めることができます。

ユーザー・プロジェクトの作成

以下の統合コンポーネント・ライブラリーを作成することをお勧めします。

- 作成した統合コンポーネント・ライブラリーごとに 1 つ。ライブラリーとサーバーの間には 1 対 1 の相関関係を保ち、ライブラリーからサーバーにコンポーネントを展開するためのユーザー・プロジェクトを用意することをお勧めします。これにより、各ユーザー・プロジェクトは各ライブラリー専用となり、必要な場合はライブラリー全体をサーバーに展開できます。

このタイプのユーザー・プロジェクトには、対応するサーバーに関連する方法で名前を付けることをお勧めします。例えば、サーバー名が WICS420DEV の場合は、ユーザー・プロジェクトに WICS420DEVUP と名前を付けます。同様の規則は、20 ページの『統合コンポーネント・ライブラリーの作成』の統合コンポーネント・ライブラリーに対してもお勧めします。

- 開発中のインターフェースごとに 1 つのライブラリー。これはユーザー・プロジェクトの最も便利な使用法です。インターフェースに必要なコンポーネントのみに対するショートカットをユーザー・プロジェクトに作成すると、ライブラリーにあるコンポーネントのサブセットのビューを使用するとき非常に便利です。

コネクターの構成

ローカル開発環境にコネクターをインストールして実行し、インターフェースの基本的なテストの一部を実行できます。このようにコネクターを活用するには、このセクションに記載されている推奨事項に従ってください。

トランスポート・プロトコルとしての IDL の使用

15 ページの『必要なソフトウェアのインストール』で説明しているように、ローカル開発環境で WebSphere MQ を実行しておく必要はありません (実行しておくパフォーマンスが低下する場合があります)。代わりに、コネクター定義の **DeliveryTransport** プロパティを値 IDL に設定し、IIOP を使用します。

キー・ポーリングの使用

イベント通知に対して責任を持つコネクターがある場合は、コネクターがイベント・テーブルをポーリングして新規イベントを検出する必要があります。ポーリングは必要ですが、コネクターのコンソール・ウィンドウに書き込まれたメッセージ

によって重要なトラブルシューティング情報が画面から消えてしまう可能性があります。キー・ポーリングを使用すると、コネクタは指示があった場合にのみポーリング呼び出しを発行します。キー・ポーリングおよびコネクタの構成については、159 ページの『第 7 章 コネクタの構成』を参照してください。

開発統合環境の準備

4 ページの『開発統合環境への配置』で説明しているように、開発するインターフェースは、プロジェクトの他の開発者とともに集中環境に展開します。この環境は、使用可能なリソースおよびシステムの複雑さに応じて、インターフェースの調整、相互参照依存関係がすべて解決されていることの保証、および共用コンポーネントの変更を行うための完全に専用の環境とすることも、テスト環境とすることもできます。

1 人の開発者をテクニカル・マネージャーに任命し、開発統合環境のすべてのインターフェースを結合するプロセスを任せるとをお勧めします。通常、マップ定義およびコラボレーション・オブジェクトは単一のインターフェースに固有ですが、多くの場合、ビジネス・オブジェクト、コネクタ、関係、およびコラボレーション・テンプレートは複数のインターフェースによって共有されます。テクニカル・リーダーは、インターフェースを相互に結合するときに、すべての共用コンポーネントがすべてのインターフェースで必要とされる動作を行っていることを確認する必要があります。ビジネス・オブジェクトおよびコラボレーション・テンプレートの場合、プロジェクトのすべての開発者が各工程で相互に緊密に連絡をとらない限り、これを実現するのは困難です。

あるインターフェースで共用コンポーネントを変更する必要がある場合は、変更する必要があるインターフェースの開発者が変更を行う必要があります。インターフェースがコンポーネントを共有するその他の開発者は、変更されたバージョンをライブラリーにインポートしてから各開発者のインターフェースを完成させます。必ずコンポーネントおよび影響を受けるインターフェースのプロジェクト・ドキュメンテーションを更新し、変更内容を反映させてください。

インターフェースを開発統合環境で結合する場合は、一般に、コネクタや関係などの一部のコンポーネントを変更する必要があります。例えば、インテグレーション・システムに顧客同期インターフェースと注文処理インターフェースがあり、両方とも SAP を必要とするとしてします。通常、インターフェースは別々の開発者がそれぞれの環境で開発します。各開発者の環境にある SAP コネクタ定義には、その開発者のインターフェースをサポートするために必要な構成のみが含まれています。注文処理インターフェースに必要なビジネス・オブジェクトなどのコンポーネントはまだ開発中であるため、顧客同期インターフェースの開発者はこれらのコンポーネントを持ちません。開発者がそれぞれのインターフェースを開発統合環境に展開する場合は、SAP 用アダプターを変更し、両方のインターフェースのビジネス・オブジェクトおよびマップをサポートさせる必要があります。テクニカル・リーダーは、必要に応じて共用コンポーネントが適切に更新されていることを確認します。

必ずプロジェクト・ドキュメンテーションを正確な状態に保ち、新規インターフェースを環境に展開する場合に変更が必要なコンポーネントを追跡できるようにしてください。

ソース・コード制御の使用

ソース・コード制御下で統合コンポーネントを処理できる、ワークベンチ・ツールで使用可能なプラグインがあります。ソース・コード制御を実装すると、プロジェクトの開発者全員が、他の開発者がコンポーネントを変更する危険性を考慮しなくても統合コンポーネント・ライブラリー (またはユーザー・プロジェクト) を各自のワークスペースに追加し、ライブラリーのコンポーネントの作業を行うことができます。

注: ソース・コード制御によって複数の開発者が同じコンポーネント定義を変更するのを回避できますが、複数の開発者がサーバーにコンポーネントを展開するのを回避することはできません。複数の開発者がサーバーのコンポーネントを上書きしないようにするには、1 人を展開担当者とし、InterChange Server パスワードを変更します。

Rational ClearCase の使用

このセクションでは、ClearCase プラグインを使用可能にし、ClearCase 接続用に System Manager を使用可能にする方法について説明します。また、ClearCase の使用方法についても一部説明しています。ただし、ClearCase プラグインの詳細については ClearCase の資料を参照してください。また、このセクションでは、ClearCase クライアントのインストール、ビューの作成などの前提となるステップについては説明していません。

プラグインがインストール済みかつ使用可能であることの確認

必要なプラグインがインストールされており、ワークベンチで使用可能なことを確認するには、以下の手順を実行します。

1. Eclipse ベースのツール・フレームワーク (WSWB や WSADIE など) の `plugins` ディレクトリーにナビゲートします。
2. `com.ibm.btools.csm.ui.cc` という名前のディレクトリーを探し、存在する場合はその内容を表示させます。
3. `plugin.xml` という名前のファイルを探します。この名前のファイルが存在しないが `plugin.xml.off` という名前のファイルが存在する場合は、`plugin.xml.off` を `plugin.xml` に名前変更します。
4. 以下のプラグインについてもステップ 2 から 3 を繰り返します。
 - `com.rational.clearcase`
 - `com.rational.clearcase.help`

これらの必要なディレクトリーが見つからない場合、または `plugin.xml` という名前のファイルが存在しない場合は、ClearCase とともに System Manager を使用することはできません。

ClearCase 接続用の System Manager の構成

ファイル・システムでプラグインが使用可能になっている場合は、以下の手順を実行して ClearCase 接続用に System Manager を構成する必要があります。

1. ワークベンチを開始します。
2. メニュー・バーから「パースペクティブ」>「すべて閉じる」を選択します。
3. 「System Manager」パースペクティブを開きます。

4. メニュー・バーから「パースペクティブ」>「カスタマイズ (Customize)」を選択します。
5. 「その他」ノードを展開します。
6. 「ClearCase」チェック・ボックスを有効にします。
7. 「OK」をクリックします。

「ClearCase」メニューがメニュー・バーに追加されます。

8. WebSphere Business Integration System Manager ビューのタイトル・バーにある黒い下向き矢印をクリックし、コンテキスト・メニューから「ClearCase 装飾を表示します」を選択します。

ワークベンチのリソースに、ClearCase というコンテキスト・メニューが表示されます。このコンテキスト・メニューはソース・コード制御操作に使用でき、チェックアウトされているかどうかを示すアイコンなどが表示されます。

ClearCase への接続

System Manager を ClearCase に接続するには、メニュー・バーから「ClearCase」>「Rational ClearCase へ接続」を選択します。

ClearCase へのプロジェクトの移動

複数の開発者が同じコンポーネントにアクセスできるようにするには、最初にシステムに統合コンポーネント・ライブラリーを作成し、ClearCase に移動する必要があります。ClearCase にライブラリーを移動するには、以下の手順を実行します。

1. 統合コンポーネント・ライブラリーを右マウス・ボタンでクリックし、コンテキスト・メニューから「ClearCase」>「プロジェクトを ClearCase に移動」を選択します。
2. ビューで、統合コンポーネント・ライブラリーを移動するディレクトリーにナビゲートし、「OK」をクリックします。

「進行状況情報 (Progress Information)」ダイアログが表示され、プロジェクトがソース・コード制御に移動されます。

3. 「要素をソース・コントロールに追加 (Add Element(s) to Source Control)」ウィンドウで、ライブラリーに追加する統合コンポーネントを選択し、「OK」をクリックします。

ワークスペースへのプロジェクトの追加

複数の開発者によって共用される統合コンポーネント・ライブラリーがある開発者によって ClearCase に移動されると、他の開発者は、以下の手順を実行して専用のワークスペースにプロジェクトを追加できます。

1. メニュー・バーから「ClearCase」>「Rational ClearCase へ接続」を選択し、ClearCase に接続します。
2. メニュー・バーから「ClearCase」>「プロジェクトをワークスペースに追加」を選択します。
3. 「ClearCase プロジェクト・ロケーション (ClearCase Project Location)」ダイアログで、ソース・コード制御下に移動した統合コンポーネント・ライブラリーのディレクトリーにナビゲートして選択します。
4. 「OK」をクリックします。

5. 「要素をソース・コントロールに追加 (Add Elements(s) to Source Control)」ダイアログで「OK」をクリックします。

プロジェクト・フォルダーがワークスペースに追加されます。コンポーネントをチェックインまたはチェックアウトするには、コンポーネントを右マウス・ボタンでクリックして「ClearCase」コンテキスト・メニューを使用します。

注: プロジェクト・フォルダーを表示させるには、ワークベンチを再始動する必要があります。

VCM の使用

System Manager は VCM とともに使用することもできます。これを実行する方法については、VCM および VCM プラグインの資料を参照してください。VCM 接続を使用可能にするには、23 ページの『プラグインがインストール済みかつ使用可能であることの確認』に記載されているステップを以下のプラグイン・ディレクトリに適用します。

- com.ibm.btools.csm.ui.vcm
- org.eclipse.vcm.core
- org.eclipse.vcm.ui

第 3 章 IBM WebSphere InterChange Server ツールおよび環境

この章では、ビジネス・プロセス統合インターフェースの開発に使用する WebSphere InterChange Server Toolset について説明します。この章を構成するセクションは次のとおりです。

- 『System Manager、統合テスト環境、および Collaboration Debugger について』
- 30 ページの『System Manager の使用』
- 35 ページの『InterChange Server インスタンスの処理』
- 44 ページの『統合コンポーネント・ライブラリーの処理』
- 47 ページの『ユーザー・プロジェクトの処理』
- 52 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』
- 66 ページの『ソリューションの処理』
- 68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』
- 70 ページの『サーバーへのコンポーネントの展開』
- 83 ページの『InterChange Server リポジトリ内のコンポーネントの処理』
- 94 ページの『依存関係および参照』
- 96 ページの『複数のワークベンチ・リソースで使用可能な標準の操作』
- 98 ページの『Eclipse ベースのワークベンチの使用』

System Manager、統合テスト環境、および Collaboration Debugger について

System Manager、統合テスト環境、および Collaboration Debugger は、WebSphere Studio Workbench および WebSphere Studio Application Developer Integration Edition という名前の Eclipse ベースのツール・フレームワーク内で動作するプラグインです。このセクションでは、Eclipse フレームワーク、WSWB、WSADIE、および IBM WebSphere InterChange Server ツール・プラグインの概要について説明します。

Eclipse Platform について

Eclipse Platform は、ツールの作成用のオープン・ソースの統合開発環境 (IDE) です。これによって提供される開発キットおよびランタイムにより、ツールの開発者は、ユーザーが特定のタイプのリソースを処理できるプラグインを作成できます。

IBM は、WebSphere Studio WorkBench (WSWB) および WebSphere Studio Application Developer Integration Edition (WSADIE) の 2 つの商標で Eclipse プラットフォームを提供しています。

プラグイン

プラグインは、Eclipse ベースのワークベンチに機能を追加するためにソフトウェア・ベンダーが開発するモジュール式の拡張です。プラグインは、ワークベンチのユーザーが特定のタイプのリソースを処理できるパースペクティブ、エディター、およびビューをカプセル化します。

例えば、あるプラグインはテキスト・エディターの機能を提供します。別のプラグインは、HTML エディターの機能を提供します。WebSphere InterChange Server プラグインは、統合コンポーネントと連携するための機能を提供します。このプラグイン・モデルにより、ユーザーはリソースのタイプごとの専用ツールを使用するのではなく、さまざまなタイプのリソースとの連携が可能な単一ツールを持つことになります。

プラグインをインストールするには、プラグインをまとめた 1 つ以上の圧縮したアーカイブを、ワークベンチの製品ディレクトリーにある `plugins` ディレクトリーに解凍します。WICS プラグインは、インストーラーによって `plugins` ディレクトリーに解凍されます。

IBM は、ビジネス・インテグレーション・リソースを処理するために、System Manager、統合テスト環境、および Collaboration Debugger といったプラグインに WebSphere InterChange Server を提供しています。これらのプラグインは、`com.ibm.btools` ネーム・スペースにあるワークベンチの `plugins` ディレクトリーの多くの未圧縮ディレクトリーにまとめられています。例えば、統合コンポーネントの作成に使用するほとんどの基本インターフェース (System Manager) は、`com.ibm.btools.csm` プラグイン・ディレクトリーに入っています。

ワークベンチ

ワークベンチは、Eclipse ベースのツール・フレームワークでアクティブなパースペクティブ、エディター、およびビューの集合です。さらに、このフレームワークは、インストール済みの使用可能なプラグインの集合の影響を受けます。インターフェースは使用方法に応じて変わりますが、この用語は、それとは無関係に作業する Eclipse ベースのインターフェースを指す一般的な用語です。

ワークスペース

ワークスペースはプロジェクトのコンテナです。ワークスペースはファイル・システムのディレクトリーであり、デフォルトではここにプロジェクトを格納するようプロンプトが出されています。

プロジェクト

プロジェクトは、ユーザー定義のリソースのグループであり、最終的にはファイル・システムのディレクトリーとして実現されます。

ビジネス・プロセス・インターフェースを開発するときの最初の作業は、統合コンポーネント・ライブラリー (開発するコンポーネントを入れるプロジェクト) を定義することです。統合コンポーネント・ライブラリーを作成するときには、格納先ファイル・システムの場所を指定します (デフォルトではワークスペースのディレクトリーです)。その場所には、統合コンポーネント・ライブラリーに指定した名前で

フォルダーが作成されます。ライブラリー・フォルダーには、統合コンポーネントのタイプごとに多数のフォルダー (Maps、BusinessObjects、Connectors など) が作成されます。

また、**ユーザー・プロジェクト**というプロジェクトも作成します。ユーザー・プロジェクトは、統合コンポーネントを参照するショートカットの集合です。コンポーネントを InterChange Server インスタンスに展開するには、統合コンポーネントを統合コンポーネント・ライブラリーからユーザー・プロジェクトに追加する必要があります。ユーザー・プロジェクトは、コンポーネントをサーバーに展開する他に、コンポーネントを機能の観点からグループ化できるように設計されています。統合コンポーネント・ライブラリーが処理対象となる可能性があるすべてのコンポーネントの集合であるのに対し、ユーザー・プロジェクトは特定のインターフェースで作業するコンポーネントをグループ化するために設計されています。

リソース

リソースは、ワークベンチで処理するプロジェクト、ファイル、およびフォルダーです。

統合コンポーネントを作成すると、統合コンポーネント・ライブラリー・プロジェクトの適切なフォルダーにファイルとして格納されます。統合コンポーネントはそれぞれ異なる拡張子で格納されます (例えば、マップは `.cwm` の拡張子で格納され、コラボレーション・テンプレートは `.cwt` の拡張子で格納されます) が、格納形式はすべて XML です。

マップやコラボレーション・テンプレートなどの一部のコンポーネントは、それらの定義ファイルのほかに Java ソース・ファイルを持ちます。

パースペクティブ

パースペクティブは、特定のユーザー役割に必要な機能を提供するためにエディターおよびビューをグループ化したものです。

例えば、System Manager パースペクティブは、InterChange Server インスタンスで作業するためのビューと、コラボレーション・オブジェクト定義およびデータベース接続プールの統合コンポーネント・ライブラリーとエディターを提供します。統合テスト環境パースペクティブには、クライアント・エミュレーション・インターフェースおよびビジネス・オブジェクトのテスト・データに対するビューと、テスト・ユニット用のエディターがあります。

エディター

エディターにより、ワークベンチでリソースのオープン、保管、およびクローズを行うことができます。

例えば、System Manager のエディターではコラボレーション・オブジェクトを変更できます。また、InterChange Server を構成できるエディターもあります。

ビュー

ビューは、ワークベンチで処理するリソースに関する情報を提供します。

例えば、System Manager の WebSphere Business Integration System Manager ビューは、統合コンポーネント・ライブラリーおよびユーザー・プロジェクトに対するビューです。また、InterChange Server ビューでは、登録した InterChange Server インスタンスを処理できます。

WSWB および WSADIE について

WebSphere Studio Workbench (WSWB) は、IBM 製品としてリリースされる Eclipse プラットフォームです。IBM では WebSphere InterChange Server とともに WSWB を配布しているため、コア・インフラストラクチャーとともにインストールできます。WSWB は、WebSphere InterChange Server 統合コンポーネントの開発に必要なすべてのプラグインを実行できます。

WebSphere Studio Application Developer Integration Edition (WSADIE) は、WSWB と同様に IBM 製品としてリリースされる Eclipse プラットフォームですが、新規プラグインの開発にも使用できます。新規プラグインを作成する機能は統合コンポーネントの開発には不要なため、WSADIE は WebSphere InterChange Server には付属していません。ただし、インストールした場合は、必要な System Manager および統合テスト環境プラグインを実行するために使用できます。WebSphere InterChange Server をインストールするときには、インストール済みの WSADIE を更新し、必要なプラグインをインストールできます。

System Manager について

System Manager は、WebSphere InterChange Server Business Integration システムで統合コンポーネントおよびサーバー・インスタンスを処理するパースペクティブです。System Manager は、主に以下の作業に使用します。

- WebSphere InterChange Server Business Integration Toolset での他のツールの起動
- 一部の統合コンポーネントの開発および構成
- InterChange Server インスタンスの処理
- リポジトリへの統合コンポーネントの展開

統合テスト環境について

統合テスト環境は、開発したビジネス・インテグレーション・インターフェースをテストできるパースペクティブです。提供されるグラフィック・インターフェースにより、コネクターのエミュレート、必要なコンポーネントの始動、およびビジネス・オブジェクト・データの検証が可能です。このパースペクティブの詳細については、273 ページの『第 12 章 統合テスト環境の使用』を参照してください。

Collaboration Debugger について

Collaboration Debugger は、コラボレーション・ロジックのトラブルシューティングを簡単にするパースペクティブです。詳細については、337 ページの『第 13 章 Collaboration Debugger の使用』を参照してください。

System Manager の使用

このセクションでは、System Manager パースペクティブの始動方法および使用方法について説明します。

System Manager の始動

IBM WebSphere InterChange Server をインストールするときには、WebSphere Studio Workbench とともに InterChange Server との対話に必要なプラグインをインストールするか、既存のインストール済み WebSphere Studio Application Developer Integration Edition にプラグインをインストールするかを選択できます。

いくつかの Java 仮想マシン引き数を指定してワークベンチを起動して、フレームワークが CORBA によって InterChange Server と通信できるようにする必要があります。これを行うために、startcsm.bat という名前のバッチ・ファイルが WebSphere InterChange Server 製品ディレクトリーの bin ディレクトリーに作成され、そのバッチ・ファイルへのショートカットが「プログラム」グループに作成されます。

WebSphere InterChange Server 環境でワークベンチを機能させるには、ワークベンチ専用のプログラム・グループのショートカットを起動したりワークベンチの実行可能ファイルを直接起動したりする代わりに、カスタム・ショートカットを起動する必要があります。

System Manager を始動するには、以下の手順を実行します。

1. 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「管理」>「System Manager」を選択します。
2. メニュー・バーから「ウィンドウ」>「パースペクティブを開く」>「その他」を選択します。
3. パースペクティブのリストから「System Manager」を選択し、「OK」をクリックします。

WebSphere Studio Workbench が始動し、表示されます。32 ページの図 2 に System Manager のパースペクティブを示します。このインターフェースおよびエレメントについては、『System Manager インターフェース』で説明します。

System Manager インターフェース

System Manager パースペクティブを開くときのデフォルト構成には、いくつかのビューおよびエディターがあります。32 ページの図 2 に System Manager パースペクティブを示します。

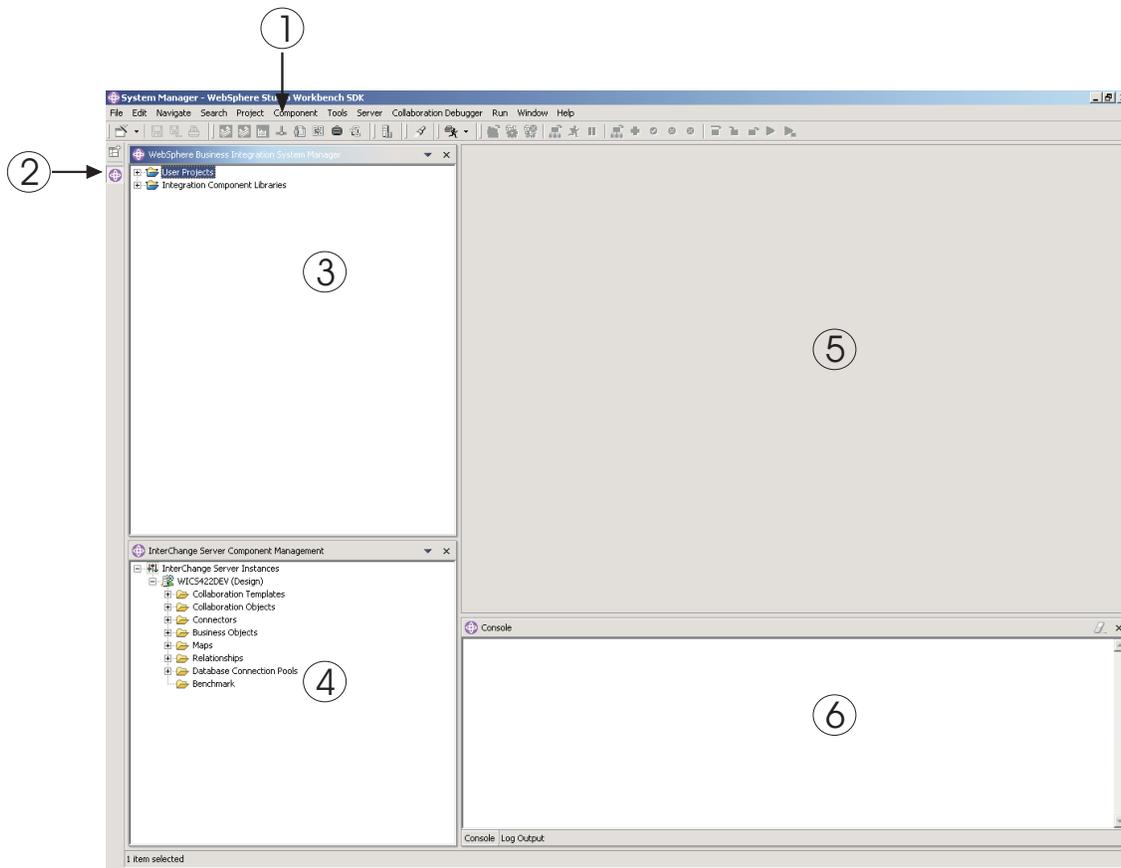


図 2. System Manager パースペクティブ

表 1 に、System Manager パースペクティブのインターフェース・エレメントを示します。番号は 32 ページの図 2 と対応しています。

表 1. System Manager パースペクティブのインターフェース・エレメント

インターフェース・エレメント番号	インターフェース・エレメント名
1	『メニュー・バーおよびツールバー』
2	34 ページの『パースペクティブのショートカット・バー』
3	34 ページの『WebSphere Business Integration System Manager ビュー』
4	34 ページの『InterChange Server コンポーネント管理ビュー』
5	35 ページの『エディター・ビュー』
6	35 ページの『コンソール・ビュー』

以下のセクションでは、System Manager のインターフェース・エレメントの詳細について説明します。

メニュー・バーおよびツールバー

メニュー・バーおよびツールバーを使用して、Eclipse ベースのツール・フレームワークの処理および WebSphere InterChange Server コンポーネントの処理を行うこと

ができます。多くのメニュー・バー項目にはツールバーにも対応する機能があるため、以下のセクションでは、メニュー・バーとメニュー・バー項目についてのみ説明します。

「ファイル」メニュー：これは Eclipse 標準のメニューであり、リソースを処理するために使用します。主に、新規統合コンポーネント・ライブラリーおよびユーザー・プロジェクトの作成に使用します。

統合コンポーネント・ライブラリーおよびユーザー・プロジェクトの作成方法については、44 ページの『統合コンポーネント・ライブラリーの処理』および 47 ページの『ユーザー・プロジェクトの処理』を参照してください。

「編集」メニュー：これは Eclipse 標準のメニューであり、切り取り、コピー、貼り付けなどの標準的な項目が多数あります。

作成するコンポーネントの切り取り、コピー、貼り付け、および削除については、52 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。

「ナビゲート」メニュー：これは Eclipse 標準のメニューであり、ワークベンチ内のリソース間をナビゲートするために使用できます。このメニューの詳細については、ワークベンチの文書を参照してください。

「検索」メニュー：これは Eclipse 標準のメニューであり、リソースの検索とリソース内の検索を行うために使用できます。このメニューの詳細については、ワークベンチの文書を参照してください。

「プロジェクト」メニュー：これは Eclipse 標準のメニューであり、ワークベンチ内のプロジェクト・リソースを処理するためのメニュー項目があります。このメニュー項目は、System Manager パースペクティブの処理中には使用されません。このメニューの詳細については、ワークベンチの文書を参照してください。

「コンポーネント」メニュー：このメニューは System Manager パースペクティブで提供されます。このメニューは、作成する統合コンポーネントの処理に役立ちます。このメニューの項目は、このガイド全体および特定のタスクを説明するセクションにおいて記述されています。

「ツール」メニュー：このメニューは System Manager パースペクティブで提供されます。このメニューは、統合コンポーネントの作成に使用するツールの起動に使用します。

詳細については、52 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。

「サーバー」メニュー：このメニューには「サーバーを登録」項目があります。このメニュー項目を使用すると、InterChange Server インスタンスを登録できます。

詳細については、35 ページの『InterChange Server インスタンスの登録』を参照してください。

「実行」メニュー: このメニューには、外部プログラム、バッチ・ファイル、およびビルド・スクリプトを実行するため、外部ツールを構成する項目があります。詳細については、ワークベンチの文書を参照してください。

「ウィンドウ」メニュー: このメニューには、パースペクティブ、ビュー、エディター、および設定を操作するための項目があります。

これらのメニュー項目は、ワークベンチの文書およびこのガイドのさまざまなセクションで説明されています。

「ヘルプ」メニュー: このメニューには、ワークベンチ文書を紹介する項目や、ワークベンチとパースペクティブのバージョン情報を取得できる項目があります。

パースペクティブのショートカット・バー

パースペクティブのショートカット・バーは、パースペクティブ間をナビゲートするのに便利です。例えば、System Manager、統合テスト環境、および Java パースペクティブを開いている場合は、パースペクティブ・ショートカット・ツールバーの各ワークスペース・アイコンをクリックしてパースペクティブを切り替えることができます。

また、「ウィンドウ」メニューを使用して他のパースペクティブにアクセスすることもできます。

- 現在表示中のパースペクティブのアイコンより上位のパースペクティブ・ショートカット・バーにあるアイコンによって表されるパースペクティブにアクセスするには、キーボード・ショートカット **Alt + 上矢印**を使用します。
- 現在表示中のパースペクティブのアイコンより下位のパースペクティブ・ショートカット・バーにあるアイコンによって表されるパースペクティブにアクセスするには、メニュー・バーから「パースペクティブ」>「次の」を選択するか、キーボード・ショートカット **Alt + 下矢印**を使用します。

WebSphere Business Integration System Manager ビュー

このビューには、ユーザー・プロジェクト・ノードおよび統合コンポーネント・ライブラリー・ノードがあります。これらは WebSphere InterChange Server プロジェクトの一種です。

これらのタイプのプロジェクトの処理方法については、44 ページの『統合コンポーネント・ライブラリーの処理』および 47 ページの『ユーザー・プロジェクトの処理』を参照してください。

InterChange Server コンポーネント管理ビュー

このビューを使用して、InterChange Server インスタンスおよびその内部コンポーネントを操作します。InterChange Server インスタンスを登録すると、行と記入項目が、サーバーごとに、「サーバー・インスタンス」ノードの下に作成されます。ノードを展開すると、そのインスタンスのリポジトリ内のコンポーネントを表示でき、右マウス・ボタン・クリックで表示されるメニュー項目を使用して、それらのコンポーネントを操作できます。

InterChange Server インスタンスの登録および処理については、35 ページの『InterChange Server インスタンスの処理』を参照してください。

エディター・ビュー

これは、ファイルおよび統合コンポーネント定義などのフレームワークの各リソースを処理するビューです。処理対象のリソースのタイプに応じて異なるエディターが開きます。例えば、テキスト・ファイルはテキスト・エディターで開きますが、コラボレーション・オブジェクト定義は WebSphere InterChange Server 固有のエディターで開きます。

コンソール・ビュー

このビューには、「コンソール」および「ログ出力」の 2 つのタブがあります。System Manager でマップまたはコラボレーション・テンプレートをコンパイルすると、各コンポーネントのコンパイルが正常に完了したかどうかを示すメッセージが「コンソール」タブに表示され、「ログ出力」タブに検出されたエラーまたは警告が表示されます。

InterChange Server インスタンスの処理

このセクションでは、InterChange Server を処理するために System Manager で行う作業について説明します。

InterChange Server インスタンスの登録

InterChange Server インスタンスを処理するには、System Manager に登録する必要があります。以下の手順を実行します。

要確認: InterChange Server インスタンスに接続するには、InterChange Server が実行されている必要があります。InterChange Server の始動については、「システム管理ガイド」を参照してください。実行されていないサーバーも登録できますが、登録しても System Manager は接続しません。

1. 以下のいずれかの操作を実行し、「サーバーを登録」ダイアログを表示させます。
 - メニュー・バーから「サーバー」>「サーバーを登録」を選択します。
 - InterChange Server コンポーネント管理ビューで、「**InterChange Server インスタンス**」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「サーバーを登録」を選択します。

注: InterChange Server ビューが参照できない場合は、100 ページの『表示およびクローズ』の手順に従って使用可能にしてください。

2. 以下のいずれかの操作を実行し、「サーバー名」フィールドにサーバー名を入力します。
 - InterChange Server インスタンスの名前を「サーバー名」フィールドに入力します。

要確認: InterChange Server インスタンス名は、大文字小文字を区別するので、名前は正確に指定してください。

- 以下の手順を実行し、ネットワークの InterChange Server インスタンスを参照します。
 - a. 「参照」をクリックします。

System Manager が、ネットワーク上のアクティブなサーバーを検出し、「サーバーを検索」ダイアログに表示します。ネットワークの規模、速度、構成によっては時間がかかる場合があります。

注: サーバーの名前を入力することによって実行されていないサーバーを登録できますが、実行されていないサーバーは参照できません。

- b. 登録するサーバー・インスタンスを選択し、「OK」をクリックします。
3. 「ユーザー名」フィールドに InterChange Server インスタンスと対話するユーザーの名前を入力します。

デフォルトのユーザー名は admin です。

4. 「パスワード」フィールドに、ステップ 3 で指定したユーザー名のパスワードを入力します。

デフォルト・ユーザー名 admin に対するデフォルト・パスワードは null です。

5. System Manager において、InterChange Server インスタンスに接続する必要があるたびに、ユーザー名とパスワードを指定しないで済むようにするには、「ユーザー ID/パスワードを保管」チェック・ボックスを有効にします。

要確認: この方法でユーザー名およびパスワードをキャッシュする場合は、必ずセキュリティについて考慮してください。一部のコンポーネント定義では、統合するアプリケーションにログインするために有効なユーザー名およびパスワードなどの機密情報が必要です。このような情報によって、これらのアプリケーションに保管されているレコード、およびそれらのレコードに保管されている重要な情報 (クレジット情報や給与計算情報など) に他のユーザーがアクセスできる場合があります。この方法でユーザー名およびパスワードをキャッシュするのは、System Manager からアクセス可能な情報が実動情報でない場合に限定してください。

6. ローカル・テスト・サーバーとして InterChange Server インスタンスを登録するには、「ローカル・テスト・サーバー」チェック・ボックスを使用可能にし、「ローカル・サーバーのインストール・パス」フィールドに WebSphere InterChange Server 製品ディレクトリーの絶対パスを入力するか、「参照」ボタンを使用して製品ディレクトリーまで進みます。

このチェック・ボックスは、インターフェースをテストする必要がある場合、またはサーバーが System Manager と同じマシンにある場合のみ有効にしてください。コンポーネントを開発する場合または実動サーバーで作業する場合は、必ず使用不可にしてください。

インターフェース・テストの詳細については、273 ページの『第 12 章 統合テスト環境の使用』 および 337 ページの『第 13 章 Collaboration Debugger の使用』を参照してください。

7. 「OK」をクリックします。

System Manager が、InterChange Server インスタンスを登録して、(InterChange Server インスタンスの名前、ユーザー名、およびパスワードがすべて正確に指定され、サーバーと IBM Java オブジェクト・リクエスト・ブローカー (ORB) が

動作している場合) その登録したインスタンスに接続し、InterChange Server コンポーネント管理ビューに、登録したインスタンスに対する入力項目を表示します。

その後、InterChange Server コンポーネント管理ビューで、InterChange Server インスタンスを右マウス・ボタンでクリックすると、InterChange Server インスタンスを処理できるようになります。

図 3 に、「新規サーバーを登録」ダイアログを示します。ここでは、ユーザー名とパスワードが入力され、キャッシュ化されています。

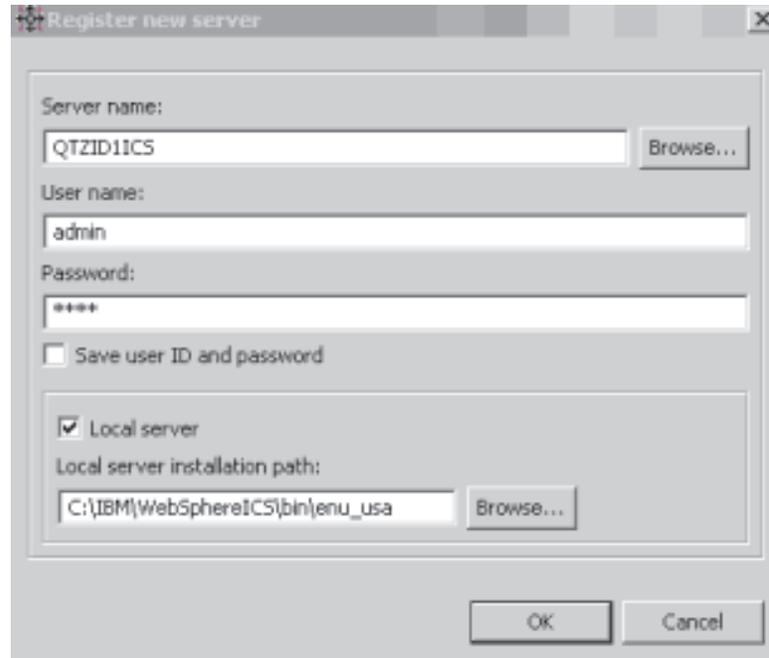


図 3. InterChange Server インスタンスの登録

InterChange Server への接続

System Manager で InterChange Server インスタンスを登録するときに、インスタンス名、ユーザー名、およびパスワードがすべて正確であり、サーバーおよび IBM ORB が動作している場合、System Manager は自動的に登録されたインスタンスに接続されます。

インスタンスをシャットダウンする必要がある場合、または System Manager を終了する場合は、System Manager をインスタンスに再接続する必要があります。ユーザー名およびパスワードをキャッシュしているかどうかに応じて作業が少し異なるため、以下の該当するセクションのステップに従ってください。

キャッシュしたユーザー名およびパスワードでの接続

サーバーの初期登録時に、ユーザー名とパスワードをキャッシュに入れることにした場合は、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、InterChange Server インスタンスの記入項目を右マウス・ボタンでクリックし、コンテキスト・メニューから「接続」を選択します。

「ログイン」ダイアログが表示され、キャッシュに入っているログイン情報が示されます。図 4 にダイアログを示します。



図 4. 「サーバー・ユーザー ID およびパスワード」ダイアログ・ボックスおよびキャッシュされた情報

2. 「OK」をクリックします。

注: System Manager がサーバーに接続できなかった場合は、「*Problem Determination Guide*」を参照してください。

ユーザー名およびパスワードをキャッシュしていない場合の接続

最初に InterChange Server インスタンスを登録するときにユーザー名およびパスワードのキャッシュを選択しなかった場合は、以下の手順を実行する必要があります。

1. InterChange Server コンポーネント管理ビューで、InterChange Server インスタンスの記入項目を右マウス・ボタンでクリックし、コンテキスト・メニューから「接続」を選択します。
2. 「ユーザー名」フィールドに InterChange Server インスタンスと対話するユーザーの名前を入力します。

デフォルトのユーザー名は admin です。

3. ステップ 2 で指定したユーザー名に対するパスワードを「パスワード」フィールドに入力します。

デフォルト・ユーザー名 admin に対するデフォルト・パスワードは null です。

4. System Manager の InterChange Server インスタンスに接続するたびにユーザー名およびパスワード指定の必要がない場合は、「ユーザー ID/パスワードを保管」チェック・ボックスを有効にします。

要確認: この方法でユーザー名およびパスワードをキャッシュする場合は、必ずセキュリティについて考慮してください。一部のコンポーネント定義

では、統合するアプリケーションにログインするために有効なユーザー名およびパスワードなどの機密情報が必要です。このような情報によって、これらのアプリケーションに保管されているレコード、およびそれらのレコードに保管されている重要な情報（クレジット情報や給与計算情報など）に他のユーザーがアクセスできる場合があります。この方法でユーザー名およびパスワードをキャッシュするのは、System Manager からアクセス可能な情報が実動情報でない場合に限定してください。

5. 「OK」をクリックします。

InterChange Server のモード

InterChange Server は、実装サイクルの各段階に最も適したモードで実行できます。

設計モード

設計モードでは、InterChange Server は、リポジトリが整合していない状態を許可します。このため、依存するコンポーネントが存在しない状態でもリポジトリにコンポーネントを追加できます。例えば、子オブジェクトを持つビジネス・オブジェクト定義をリポジトリにインポートするときに子オブジェクトがまだ存在しない場合、実動モードの InterChange Server ではインポートに失敗し、リポジトリの整合性が確保されます。しかし、設計モードの InterChange Server では先に進むことができるため、開発アプローチに最も適した方法で統合コンポーネントを組み立てることができます。

さらに、設計モードのサーバーにパッケージを展開するときに、必ずしもマップおよびコラボレーション・テンプレートをコンパイルする必要はありません。実動モードでは、サーバーはマップおよびコラボレーション・テンプレートをすべて自動的にコンパイルします。

設計モードは、別の環境からコンポーネントをインポートするときに特に有用です。依存関係をすべて把握していない場合は、未解決の依存関係が存在してもインポート操作が失敗せずに段階的にコンポーネントをインポートできる機能は重要です。

設計モードで InterChange Server を始動するには、サーバーの始動時に `-design` パラメーターを使用します。40 ページの図 5 に、Windows システムにおける InterChange Server のショートカットを示します。「ターゲット」フィールドに `-design` パラメーターが表示されています。UNIX システムでは、`ics_manager` スクリプトの実行時に `-design` パラメーターを使用します。

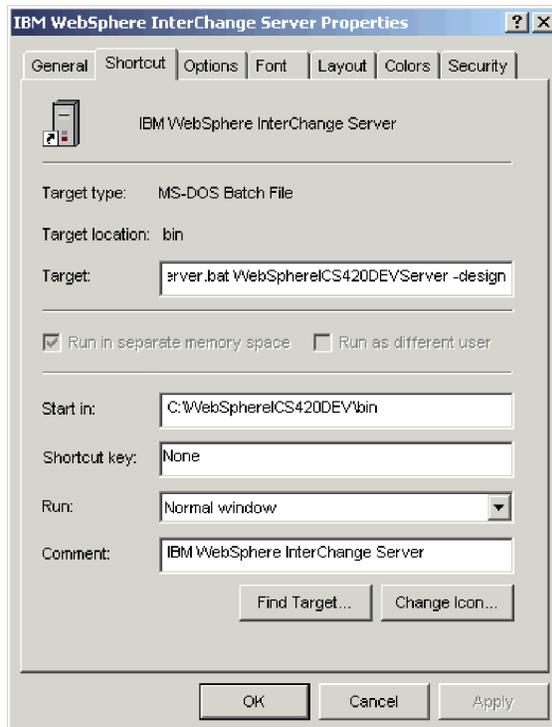


図 5. InterChange Server ショートカットでの設計モードの指定

テスト・モード

InterChange Server を登録すると、サーバーをテスト・モードで始動できます。テスト・モードで InterChange Server を始動するには、サーバーの始動時に `-test` パラメーターを使用します。

注: サーバーをテスト・モードで実行するために、テスト役割が、始動、シャットダウン、配置などのタスクを実行する操作許可を持っていることを確認してください。

統合テスト環境 (ITE) では、ローカル・サーバーおよびリモート・サーバーがサポートされます。ローカル・サーバーとは、統合テスト環境のタスク・マネージャーから始動される InterChange Server のことです。タスク・マネージャーからサーバーを起動するには、InterChange Server が登録パネルにローカル・サーバーとして登録されている必要があります。リモート・サーバーとは、ITE 外から起動される InterChange Server のことです。したがって、ITE と同じマシン上で始動されるサーバーでも、ITE 外から起動されればリモート・サーバーになる場合があります。

ローカル・サーバー・スクリプトを変更して、サーバーがテスト・サーバーとして自動的に始動するようにすることができます。リモート・サーバーにこの方法で始動する機能がありません。したがって登録されたサーバーがリモート・サーバーの場合、サーバー起動オプションは使用不可になります。

実動モード

実動モードでは、InterChange Server はリポジトリの整合性を保証するように設計されています。未解決の依存関係が存在するパッケージはリポジトリに展開できず、展開パッケージのマップおよびコラボレーション・テンプレートはすべて自動的にコンパイルされます。これらの制限により、サーバー環境でコンポーネントを適切に実行できることが保証されます。未解決の依存関係を持つコンポーネントまたは未コンパイルのコンポーネントが実行時にサーバー環境に存在すると、それらのコンポーネントを必要とするトランザクションが失敗します。開発環境ではまだ必要なコンポーネントを作成中であると考えられるため、この事態は許容されますが、実稼動環境で許容されるとは考えられません。したがって、これらの制限により、安全な展開手順が強制的に適用されます。

実動モードは、InterChange Server のデフォルトのモードであるため、実動モードで始動するために必要な構成ステップはありません。ただし、実動モードで始動する場合は、設計モードで始動するためのステップを実行しなかったことを確認してください。また、System Manager の InterChange Server コンポーネント管理ビューで、InterChange Server のモードも確認してください。

InterChange Server のパスワードの変更

InterChange Server に接続するために使用されるユーザー・アカウントのパスワードを変更できます。パスワードを変更するには、以下の手順を実行します。

1. InterChange Server ビューの InterChange Server インスタンスのエントリーを右マウス・ボタンでクリックし、コンテキスト・メニューから「パスワードを変更」を選択します。

「InterChange Server パスワードの変更」ダイアログが表示されます。これを図 6 に示します。

2. 「旧パスワード」フィールドに現在のパスワードを入力します。
3. 「新パスワード」フィールドに新規パスワードを入力します。
4. 「確認パスワード」フィールドに再度新規パスワードを入力します。
5. 「OK」をクリックします。



図 6. InterChange Server のパスワードの変更

InterChange Server の最新表示

コンポーネントを InterChange Server インスタンスに展開したら、System Manager でインスタンスを最新表示し、サーバーにコンポーネントが正確に表示されるようにする必要があります。例えば、コンポーネントをサーバーに展開し、新規統合ライブラリーを作成し、サーバーからライブラリーにコンポーネントを追加する場合は、サーバーを最新表示しないと、System Manager は新しく展開されたコンポーネントをリストしません。

サーバー・インスタンスの表示を最新にするには、InterChange Server コンポーネント管理ビューでインスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「最新表示」を選択します。

InterChange Server からの切断

InterChange Server インスタンスと System Manager の接続を切断するには、InterChange Server コンポーネント管理ビューで、切断する InterChange Server インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「切断」を選択します。

InterChange Server のシャットダウン

InterChange Server インスタンスをシャットダウンするには、InterChange Server コンポーネント管理ビューで、シャットダウンする InterChange Server インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「切断」を選択してから、インスタンスのシャットダウンの方法により、サブメニューから「正常」または「即時に」のいずれかを選択します。

シャットダウンのタイプとして「即時に」を選択した場合は、InterChange Server はただちにシャットダウンし、その時点で処理中であったフローは失敗します。失敗したフローは、後で Flow Manager を使用して解決できます。このタイプのシャットダウンは、システムのフローを考慮する必要がない開発環境およびテスト環境の場合、または失敗したフローをサブミットしても混乱しない実稼働環境で使用してください。

シャットダウンのタイプとして「正常」を選択した場合は、InterChange Server 統合コンポーネントがその時点のフローの処理を完了してからサーバーがシャットダウンします。このタイプのシャットダウンは、フローの故障によって混乱が発生する可能性がある実稼働環境で使用します。

注: InterChange Server のシャットダウンには、実行しているコンソール・ウィンドウを閉じる方法もあります。これは即時シャットダウンとなるため、その時点で処理中であったフローは失敗します。開発環境ではデータの保全本性は重要でないため、この方法も許容できますが、実稼働環境では行わないでください。開発環境の場合でも、System Manager を使用して InterChange Server をシャットダウンする利点は、InterChange Server コンポーネント管理ビューにサーバーの状況が表示されることです。System Manager を使用しない他の方法でシャットダウンした場合、System Manager は、サーバー状態の変化を検出して報告することができません。

ビューからの InterChange Server インスタンスの削除

System Manager で、InterChange Server コンポーネント管理ビューから、InterChange Server インスタンスを除去することが必要な場合があります。例えば、多くのサーバーが登録されている場合に、使用しなくなったサーバーがリストに含まれていると不便です。InterChange Server インスタンスを除去するには、以下の手順を実行します。

1. 除去対象の InterChange Server インスタンスに、System Manager が接続されていないことを確認します。これは、InterChange Server インスタンスから System Manager を切断するか、InterChange Server インスタンスをシャットダウンして行うことができます。

InterChange Server インスタンスから System Manager を切断する方法の詳細については、42 ページの『InterChange Server からの切断』を参照してください。

InterChange Server インスタンスをシャットダウンする方法の詳細については、42 ページの『InterChange Server のシャットダウン』を参照してください。

2. InterChange Server コンポーネント管理ビューで、除去対象の InterChange Server インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「削除」を選択します。
3. 「サーバーの削除の確認」プロンプトで「OK」をクリックします。

サーバー・インスタンスがビューから除去されます。

その他の InterChange Server 用 System Manager コマンド

InterChange Server インスタンスを右マウス・ボタンでクリックしたときに表示されるコンテキスト・メニューには他のメニュー項目もありますが、これらのメニュー項目については他の資料を参照してください。

表 2. その他の InterChange Server 用 System Manager コマンド

InterChange Server メニュー項目	
構成を編集	このメニュー項目については、119 ページの『第 5 章 WebSphere InterChange Server の構成』を参照してください。
統計	このメニュー項目については、「システム管理ガイド」を参照してください。
システム表示	このメニュー項目については、「システム管理ガイド」を参照してください。
サーバー・オブジェクトの削除	このメニュー項目については、92 ページの『サーバー・オブジェクトの削除ウィザードを使用したコンポーネントの削除』を参照してください。
リポジトリを削除	91 ページの『リポジトリ全体の削除』。
モニター定義ウィザード	このメニュー項目については、「システム管理ガイド」を参照してください。

統合コンポーネント・ライブラリーの処理

統合コンポーネント・ライブラリーは、開発するコンポーネントを格納する際に使用します。このセクションでは、新規統合コンポーネント・ライブラリーの作成方法について説明します。

統合コンポーネント・ライブラリーを作成したときには、通常、以下の作業も行います。

- ライブラリーにコンポーネントをインポートします。いくつかのインポート方法については、52 ページの『統合コンポーネント・ライブラリー内のコンポーネントの処理』を参照してください。
- ユーザー・プロジェクトのコンポーネントに対するショートカットを作成します。この詳細については、47 ページの『ユーザー・プロジェクトの処理』を参照してください。
- InterChange Server インスタンスにコンポーネントを展開します。詳細については、70 ページの『サーバーへのコンポーネントの展開』を参照してください。
- コンポーネントをパッケージにエクスポートし、サーバーや他のライブラリーにインポートしたり、開発をバックアップしたりします。詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

統合コンポーネント・ライブラリーの概念的な情報については、2 ページの『統合コンポーネント・ライブラリー』を参照してください。

統合コンポーネント・ライブラリーの作成

ウィザードを使用して System Manager で新規統合コンポーネント・ライブラリーを作成するには、以下の手順を実行します。

1. 以下のいずれかの操作を実行し、「新規統合コンポーネント・ライブラリー」ウィザードを開始します。
 - メニュー・バーから「ファイル」>「新規」>「新規統合コンポーネント・ライブラリー」を選択します。
 - WebSphere Business Integration System Manager ビューで「統合コンポーネント・ライブラリー」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規統合コンポーネント・ライブラリー」を選択します。
 - ツールバーの「新規ウィザードを開く (Open The New Wizard)」ボタンをクリックし、メニューから「新規統合コンポーネント・ライブラリー」を選択します。

45 ページの図 7 に、「新規統合コンポーネント・ライブラリー」ウィザードを示します。

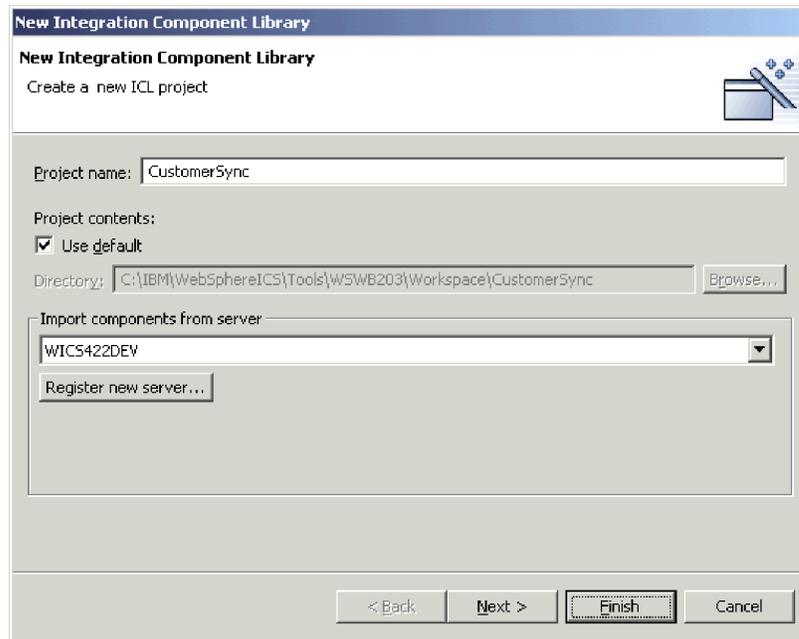


図7. 統合コンポーネント・ライブラリーの作成

2. 「プロジェクト名」フィールドに統合コンポーネント・ライブラリーの名前を入力します。

プロジェクト名には英数字および下線のみが使用でき、英語で指定します。

推奨される統合コンポーネント・ライブラリーの命名方法については、20ページの『統合コンポーネント・ライブラリーの作成』を参照してください。

3. デフォルトのロケーション (ワークスペース) にライブラリー用フォルダーを作成し、ライブラリーに指定した名前と同じ名前を付けるには、「デフォルト・ロケーションの使用」チェック・ボックスを有効にしておきます。

ライブラリー・フォルダーの名前およびロケーションを指定する場合は、以下の手順を実行します。

- a. 「デフォルト・ロケーションの使用」チェック・ボックスをクリアします。
- b. ライブラリーに使用するディレクトリーの絶対パスおよび名前を「ロケーション」フィールドに入力するか、「ブラウズ」をクリックして既存のディレクトリーを選択します。

注: 「デフォルト・ロケーションの使用」チェック・ボックスを使用して System Manager で作成する以外に、ワークスペースのパスにライブラリー用フォルダーを作成する方法はありません。

4. この時点でサーバーのリポジトリーからライブラリーにコンポーネントをインポートしない場合は、ステップ 5 (46 ページ) に進みます。

実行中のサーバーのリポジトリーからライブラリーにコンポーネントをインポートする場合は、以下の手順を実行します。

- a. 以下のいずれかを実行して、コンポーネントのインポート元のサーバーを指定します。

- 「サーバーからコンポーネントをインポート」ドロップダウン・メニューでサーバーを選択します。

サーバーが実行されており、System Manager がサーバーに接続されていなければ、サーバーはドロップダウン・メニューに表示されません。

- 「新規サーバーを登録」をクリックし、ウィザードを使用してリストにサーバーを追加します。詳細については、35 ページの『InterChange Server インスタンスの登録』を参照してください。
- b. 「次へ」をクリックします。

図 8 に、サーバーからコンポーネントをインポートする場合の画面を示します。

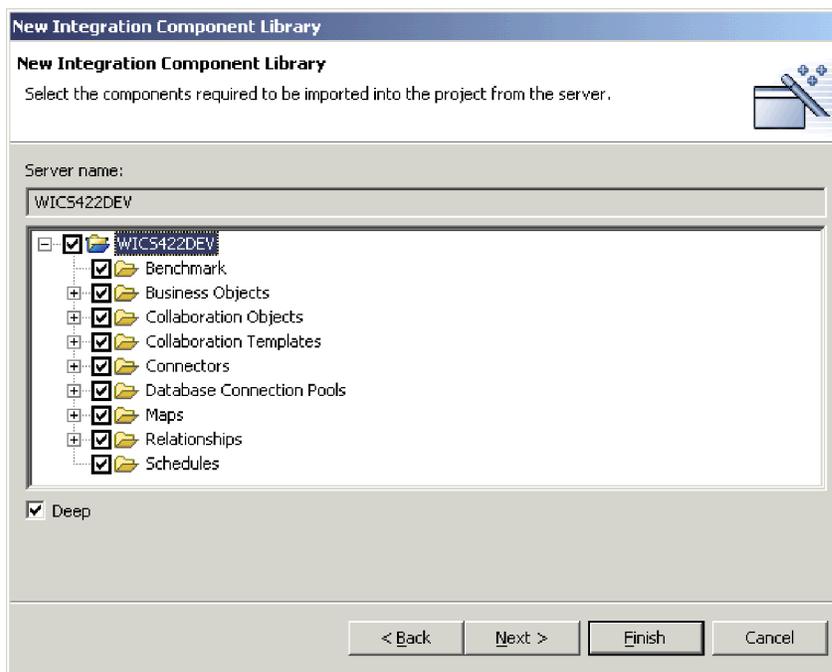


図 8. サーバーからライブラリーへのコンポーネントのインポート

- c. ウィザードの次の画面で、サーバー・インスタンスの横にあるチェック・ボックスを有効にしてリポジトリーにあるコンポーネントをすべて追加するか、サーバー・フォルダーを展開してコンポーネント・グループの横にあるチェック・ボックスを有効にするか、グループのフォルダーを展開して個々のコンポーネントのチェック・ボックスを有効にします。
 - d. 「ディープ」チェック・ボックスを有効にし、選択したコンポーネントの依存関係をすべて追加します。依存関係については、94 ページの『依存関係および参照』を参照してください。
5. 「完了」をクリックして、ウィザードを終了します。

System Manager が、「統合コンポーネント・ライブラリー」フォルダーの下に指定の名前でフォルダーを作成します。

ユーザー・プロジェクトの処理

1 つ以上のライブラリーを処理するには、統合コンポーネントへのショートカットをユーザー・プロジェクトに作成します。ユーザー・プロジェクトにより、コンポーネントのビューをインターフェースとして編成できます。System Manager からサーバーにコンポーネントを展開するには、コンポーネントのショートカットをユーザー・プロジェクトに追加する必要があります。

このセクションを構成するセクションは次のとおりです。

- 『ユーザー・プロジェクトの作成』
- 49 ページの『ユーザー・プロジェクトへのショートカットの追加』
- 66 ページの『ソリューションのエクスポート』

ユーザー・プロジェクトの概念的な情報については、2 ページの『ユーザー・プロジェクト』を参照してください。

ユーザー・プロジェクトの統合ブローカー設定の構成

複数の統合ブローカーに対して、System Manager を使用してユーザー・プロジェクトを作成できます。異なる統合ブローカーに対して、System Manager を使用可能にするには、以下のいずれかを実行します。

1. テキスト・エディターで、*ProductDir*\bin 内の *cwtools.cfg* という名前のファイルを開きます。
2. WebSphere InterChange Server プロジェクトに対して、System Manager を使用可能にするには、ICS_PROJECT セクション内の「インストール済み (Installed)」プロパティーに、値 *true* を設定します。

WebSphere Application Server プロジェクトに対して、System Manager を使用可能にするには、WAS_PROJECT セクション内の「インストール済み (Installed)」プロパティーに、値 *true* を設定します。WebSphere Application Server プロジェクトの操作の詳細については、「アダプター実装ガイド (*WebSphere Application Server*)」を参照してください。

WebSphere MQ Integrator Broker プロジェクトに対して、System Manager を使用可能にするには、WMQI_PROJECT セクション内の「インストール済み (Installed)」プロパティーに、値 *true* を設定します。WebSphere MQ Integrator Broker プロジェクトの処理の詳細については、「*WebSphere Message Brokers* 使用アダプター・インプリメンテーション・ガイド」を参照してください。

3. ファイルの保管して閉じます。

ユーザー・プロジェクトの作成

ウィザードを使用して System Manager で新規ユーザー・プロジェクトを作成するには、以下の手順を実行します。

1. 以下のいずれかの操作を実行し、「新規ユーザー・プロジェクト」ウィザードを開始します。
 - メニュー・バーから「ファイル」>「新規」>「ユーザー・プロジェクト」を選択します。

- WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規ユーザー・プロジェクト」を選択し、「新規 ICS プロジェクト」を選択します。
 - WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、「InterChange Server プロジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規 ICS プロジェクト」を選択します。
 - ツールバーの「新規ウィザードを開く (Open The New Wizard)」ボタンをクリックし、メニューから「新規ユーザー・プロジェクト」を選択します。
2. 「プロジェクト名」フィールドにユーザー・プロジェクトの名前を入力します。

プロジェクト名には英数字および下線のみが使用でき、英語で指定します。

推奨されるユーザー・プロジェクトの命名方法については、21 ページの『ユーザー・プロジェクトの作成』を参照してください。

3. デフォルトのロケーション (ワークスペース) にユーザー・プロジェクト用フォルダーを作成し、ユーザー・プロジェクトに指定した名前と同じ名前を付けるには、「プロジェクト内容」ペインの「デフォルトの使用」チェック・ボックスを有効にしておきます。

ユーザー・プロジェクト・フォルダーの名前およびロケーションを指定する場合は、以下の手順を実行します。

- a. 「プロジェクト内容」ペインの「デフォルトの使用」チェック・ボックスをクリアします。
- b. ユーザー・プロジェクトに使用するディレクトリーの絶対パスおよび名前を「ディレクトリー」フィールドに入力するか、「ブラウズ」をクリックして既存のディレクトリーを選択します。

注: 「デフォルト・ロケーションの使用」チェック・ボックスを使用して System Manager で作成する以外に、ワークスペースのパスにユーザー・プロジェクト用フォルダーを作成する方法はありません。

4. この時点で既存の統合コンポーネントへのショートカットを作成しない場合は、ステップ 5 (49 ページ) に進みます。

既存の統合コンポーネントへのショートカットを作成する場合は、統合コンポーネント・ライブラリーの横にあるチェック・ボックスを有効にしてその中にあるすべてのコンポーネントへのショートカットを作成するか、統合コンポーネント・ライブラリー・フォルダーを展開してコンポーネント・グループの横にあるチェック・ボックスを有効にするか、グループのフォルダーを展開して個々のコンポーネントのチェック・ボックスを有効にします。

注: 複数の統合コンポーネント・ライブラリーから同じ名前のコンポーネントを選択する場合は、選択内容に重複する参照が存在することを通知するプロンプトは出されません。重複するコンポーネントを選択した場合は、ウィザードで選択したときにライブラリーの最下部にあった統合コンポーネント・ライブラリーのコンポーネントに対してショートカットが作成されます。

図9に、「新規ユーザー・プロジェクト」ウィザードを示します。

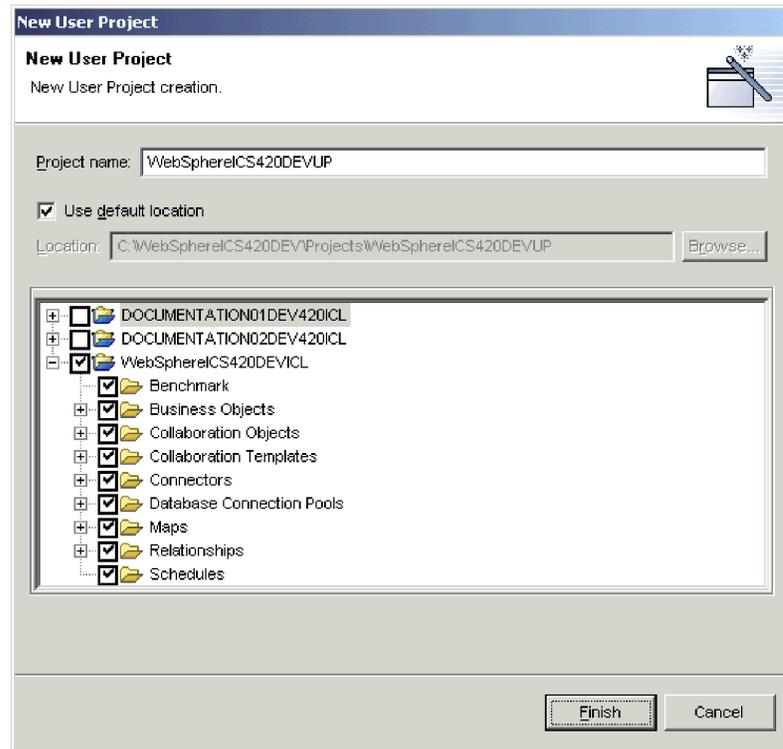


図9. ユーザー・プロジェクトの作成

5. 「完了」をクリックして、ウィザードを終了します。

ユーザーが指定した名前の付いたフォルダーが、System Manager によって、「ユーザー・プロジェクト」フォルダー内の「InterChange Server プロジェクト」フォルダーに作成されます。

ユーザー・プロジェクトへのショートカットの追加

作業しているインターフェースを表示できるようにするには、ユーザー・プロジェクトにショートカットを追加します。ユーザー・プロジェクトにショートカットを追加する方法はいくつかあります。以下のセクションではこれらの方法について説明します。

- 『依存関係ツリーの使用』
- 51 ページの『「プロジェクトを更新」ウィザードの使用』
- 51 ページの『コンポーネントのドラッグ・アンド・ドロップ』
- 67 ページの『ソリューションのインポート』
- 67 ページの『ソリューションのインポート』

依存関係ツリーの使用

「依存関係ツリー」ウィザードは、コンポーネント・ショートカットをユーザー・プロジェクトに追加するための最も便利なインターフェースです。ユーザー・プロジェクトは、主に、インターフェースを表現する目的で設計され、インターフェースは、一般に、コラボレーション・オブジェクトを中心に置かれます。さらに、ユ

ユーザー・プロジェクトは一般にコラボレーション・オブジェクトを中心に置きます。通常、ユーザー・プロジェクトに必要なショートカットを作成するには、コラボレーション・オブジェクトの依存関係を見つけます。

依存関係の概念的な情報については、94 ページの『依存関係および参照』を参照してください。

「依存関係ツリー」ウィザードを使用してユーザー・プロジェクトにショートカットを追加するには、以下の手順を実行します。

1. ライブラリーにあるコラボレーション・オブジェクトなどの統合コンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**依存関係の表示**」を選択します。
2. 「**プロジェクトに追加**」ドロップダウン・メニューから、ショートカットの追加先ユーザー・プロジェクトを選択します。
3. ウィザードの左側のペインで、ショートカットを作成するコンポーネントを選択します。

キーボード・ショートカットを使用すると作業しやすくなります。例えば、オブジェクトの範囲を選択するには **Shift** を押しながら操作し、不連続な単一オブジェクトを選択するには **Ctrl** を押しながら操作します。

4. 右向きの矢印をクリックし、ウィザードの右側のペインにコンポーネントを追加します。

図 10 に、「依存関係ツリー」ウィザードを示します。

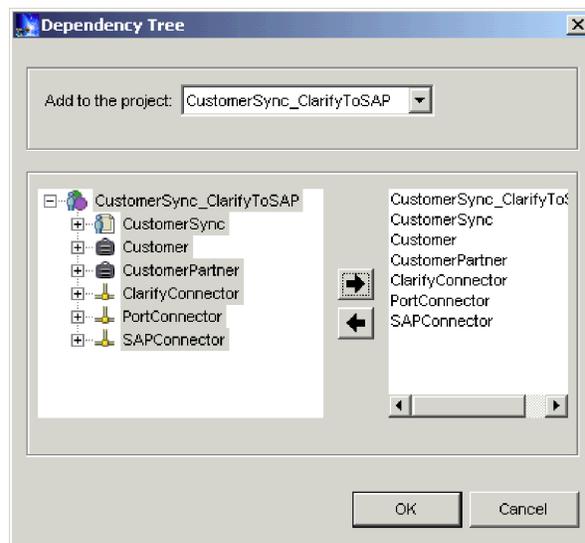


図 10. 「依存関係ツリー」を使用したユーザー・プロジェクトへのショートカットの追加

5. 「**OK**」をクリックします。

System Manager が、指定のユーザー・プロジェクトに選択されたコンポーネントへのショートカットを作成します。

「プロジェクトを更新」ウィザードの使用

「プロジェクトを更新」ウィザードが提供するインターフェースにより、最初にユーザー・プロジェクトを作成するときに表示されたユーザー・プロジェクトに類似したユーザー・プロジェクトへのショートカットを追加できます。「プロジェクトを更新」ウィザードを使用するには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで任意のユーザー・プロジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「**プロジェクトを更新**」を選択します。
2. 統合コンポーネント・ライブラリーの横にあるチェック・ボックスを有効にしてその中にあるすべてのコンポーネントへのショートカットを作成するか、統合コンポーネント・ライブラリー・フォルダーを展開してコンポーネント・グループの横にあるチェック・ボックスを有効にするか、グループのフォルダーを展開して個々のコンポーネントのチェック・ボックスを有効にします。
3. 「完了」をクリックします。

ウィザードで選択したユーザー・プロジェクトと同じ名前のコンポーネントへのショートカットがユーザー・プロジェクトに存在する場合は、プロンプトが表示され、以下の操作が可能になります。

- 表示されたコンポーネントを上書きします。
- 重複するコンポーネントをすべて上書きします。
- コンポーネントを上書きしません。
- 更新操作をキャンセルします。

ユーザー・プロジェクトにショートカットが存在するコンポーネントと同じ名前のコンポーネントを選択しなかった場合は、プロジェクトにショートカットが追加され、ウィザードが終了します。

コンポーネントのドラッグ・アンド・ドロップ

統合コンポーネント・ライブラリー・フォルダーからコンポーネントを選択し、ユーザー・プロジェクトにドラッグ・アンド・ドロップすると、そのユーザー・プロジェクトにコンポーネントへのショートカットを追加できます。

ユーザー・プロジェクトのフォルダーにまだショートカットが存在しない場合は、フォルダー自体にコンポーネントをドラッグ・アンド・ドロップしてください。フォルダーにコンポーネントをドラッグ・アンド・ドロップし、マウス・ポインターの下に四角形が表示されたらマウス・ボタンを放します。

ユーザー・プロジェクトのフォルダーにすでにショートカットが存在する場合は、フォルダー自体にコンポーネントをドラッグ・アンド・ドロップすることはできません。線が表示されるまでフォルダーの既存のショートカットの間でコンポーネントをドラッグ・アンド・ドロップし、マウス・ボタンを放します。

統合コンポーネント・ライブラリー内のコンポーネントの処理

WebSphere InterChange Server インテグレーション・システムを実装する場合、大半の時間が統合コンポーネントの処理に費やされます。本書では個々のコンポーネントの開発に関する詳細については説明しません。このセクションでは、Designer ツールの起動方法、新規コンポーネントの作成方法、既存のコンポーネントの変更方法、および System Manager で開発する一部のコンポーネントの処理方法について説明します。

統合コンポーネントの開発方法について詳しくは、次のガイドを参照してください。

- 「コラボレーション開発ガイド」
- 「マップ開発ガイド」
- 「ビジネス・オブジェクト開発ガイド」
- 「コネクタ開発ガイド (C++ 用)」
- 「コネクタ開発ガイド (Java 用)」
- 「アクセス開発ガイド」

開発して InterChange Server インスタンスに配置したコンポーネントの処理の詳細については、83 ページの『InterChange Server リポジトリ内のコンポーネントの処理』を参照してください。

Designer ツールの起動

このセクションでは、各 Designer ツールの起動方法について説明します。Designer ツールを使用すると、新規コンポーネントの作成や、既存のコンポーネントのオープンおよび変更を行うことができます。

注: Designer ツールのいずれかを起動中に、クラスが見つからないというエラーが発生した場合、System Manager を起動して、その Designer ツールをもう一度起動する必要があります。ただし、ツールが起動した後は、System Manager を稼働したままにする必要はありません。

Business Object Designer

Business Object Designer を起動するには、次のいずれかの方法を使用します。

- 「WebSphere Business Integration」システム表示で「ビジネス・オブジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規ビジネス・オブジェクトの作成」を選択します。
- WebSphere Business Integration System Manager ビューで、任意のフォルダーを選択し、以下の操作のいずれかを実行します。
 - メニュー・バーから「ツール」>「Business Object Designer」を選択します。
 - 「Business Object Designer」ツールバー・ボタンをクリックします。
 - キーボード・ショートカット **Ctrl+4** を使用します。
- 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「開発」>「Business Object Designer」を選択します。

Business Object Designer の詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

Connector Configurator

Connector Configurator を起動するには、以下のいずれかの操作を実行します。

- 「WebSphere Business Integration」システム表示で「コネクタ」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規コネクタの作成」を選択します。
- WebSphere Business Integration System Manager ビューで、任意のフォルダーを選択し、以下の操作のいずれかを実行します。
 - メニュー・バーから「ツール」>「Connector Configurator」を選択します。
 - 「Connector Configurator」ツールバー・ボタンをクリックします。
 - キーボード・ショートカット **Ctrl+1** を使用します。
- 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「開発」>「Connector Configurator」を選択します。

Connector Configurator については、159 ページの『第 7 章 コネクタの構成』および、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

Map Designer

Map Designer を起動するには、次のいずれかの方法を使用します。

- 「WebSphere Business Integration」システム表示で「マップ」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規マップの作成」を選択します。
- WebSphere Business Integration System Manager ビューで、任意のタイプの統合コンポーネントに対するフォルダーを選択し、以下のいずれかの操作を実行します。
 - メニュー・バーから「ツール」>「Map Designer」を選択します。
 - 「Map Designer」ツールバー・ボタンをクリックします。
 - キーボード・ショートカット **Ctrl+3** を使用します。
- 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「開発」>「Map Designer」を選択します。

Map Designer には、「ツール」メニューの中に「自動マッピング」という機能があります。この機能を使用すると、類似の属性を持つ 2 つのビジネス・オブジェクト間に自動的にマッピングを作成できます。Map Designer および自動マッピングの詳細については、「マップ開発ガイド」を参照してください。

Relationship Designer

Relationship Designer を起動するには、以下のいずれかの操作を実行します。

- WebSphere Business Integration System Manager ビューで、「関係」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「Relationship Designer」を選択します。

- 「WebSphere Business Integration」システム表示で任意のタイプの統合コンポーネントのフォルダーを選択し、以下のいずれかの操作を実行します。
 - メニュー・バーから「ツール」>「**Relationship Designer**」を選択します。
 - 「**Relationship Designer**」ツールバー・ボタンをクリックします。
 - キーボード・ショートカット **Ctrl+5** を使用します。
- 「スタート」>「プログラム」>「**IBM WebSphere InterChange Server**」>「**IBM WebSphere Business Integration Toolset**」>「開発」>「**Relationship Designer**」を選択します。

Relationship Designer については、「マップ開発ガイド」を参照してください。

Process Designer

Process Designer を起動するには、以下のいずれかの操作を実行します。

- 「WebSphere Business Integration」システム表示で「コラボレーション・テンプレート」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規コラボレーション・テンプレートの作成」を選択します。
- 「WebSphere Business Integration」システム表示で任意のタイプの統合コンポーネントのフォルダーを選択し、以下のいずれかの操作を実行します。
 - メニュー・バーから「ツール」>「**Process Designer**」を選択します。
 - 「**Process Designer**」ツールバー・ボタンをクリックします。
 - キーボード・ショートカット **Ctrl+2** を使用します。
- 「スタート」>「プログラム」>「**IBM WebSphere InterChange Server**」>「**IBM WebSphere Business Integration Toolset**」>「開発」>「ツール」>「**Process Designer**」を選択します。

Process Designer の詳細については、「コラボレーション開発ガイド」を参照してください。

新規コンポーネントの作成

以下のコンポーネントの場合、52 ページの『Designer ツールの起動』の説明に従って、各コンポーネントの Designer ツールを起動すると、そのタイプの新規コンポーネントを作成できます。

- ビジネス・オブジェクト
- コネクター
- マップ
- 関係
- コラボレーション・テンプレート

コンポーネントによっては専用の Designer ツールがないものもあります。この場合は、System Manager によって表示されるインターフェースで作成します。新しいベンチマーク、コラボレーション・オブジェクト、データベース接続プール、およびスケジュールを作成するには、55 ページの表 3 を参照してください。

表 3. Designer ツールを使用しない新規統合コンポーネントの作成方法

コンポーネント	方法	詳細情報の参照先
ベンチマーク	<p>「WebSphere Business Integration」システム表示で「ベンチマーク」フォルダーを右マウス・ボタンでクリックし、「ベンチマークのセットアップ」コンテキスト・メニューから以下のいずれかを選択します。</p> <ul style="list-style-type: none"> • 新規ベンチマークを構成 • 既存のベンチマークを編集 • ワークロードをファイルに生成 	「Benchmarking Guide」
コラボレーション・オブジェクト	<p>「WebSphere Business Integration」システム表示で「コラボレーション・オブジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規コラボレーション・オブジェクトの作成」を選択します。</p>	219 ページの『第 9 章 コラボレーション・オブジェクトの構成』
データベース接続プール	<p>「WebSphere Business Integration」システム表示で「データベース接続プール」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規データベース接続の作成」を選択します。</p>	207 ページの『第 8 章 データベース接続プールの構成』
スケジュール	<ul style="list-style-type: none"> • 「WebSphere Business Integration」システム表示で「スケジュール」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンポーネントのスケジュールを編集」を選択します。 • ライブラリーでスケジュール可能な任意のコンポーネント、またはユーザー・プロジェクトでスケジュール可能な任意のコンポーネントへのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンポーネントのスケジュールを編集」を選択します。 	「システム管理ガイド」

統合コンポーネントとしての Web サービスの指定

このセクションでは、System Manager で Web サービスを統合コンポーネントとして登録、起動する方法について説明します。以下のセクションから構成されています。

- 『概要』
- 56 ページの『Web サービスの登録』
- 57 ページの『登録済み Web サービスの表示』
- 58 ページの『登録済み Web サービスのテスト』

概要

Web サービスはネットワーク全体に配布されるアプリケーションで、ビジネス・プロセスの一部になります。このサービスは、InterChange Server またコラボレーシ

ョンの一部ではありません。ただし、アクセス・フレームワークを使用して InterChanger Server と通信します。このサービスは、InterChange Server 管理下のデータベースを更新します。したがって、Web サービスはビジネス・プロセス内のデータを同期するため、情報について InterChange Server から連絡を受ける必要があります。

Web サービスには、ローカル、分散、または Web ベースのものがあります。このサービスは、TCP/IP、HTTP、Java、HTML、および XML などのオープン・スタンダードの上に構築されます。Web サービスでは、メッセージング用の SOAP (Simple Object Access Protocol)、パブリッシュおよびディスカバリー用の UDDI (Universal Description, Discovery and Integration) および WSDL (Web Services Description Language) などの新しい標準テクノロジーを使用します。

System Manager を使用すると、統合コンポーネント・ライブラリー (ICL) のコンポーネントとして、Web サービスを登録することができます。登録プロセスにより、Web サービスが統合コンポーネント・ライブラリーの下にリストされ、Web サービスとの通信のために InterChange Server が使用するビジネス・オブジェクトが自動的に生成されます。Web サービスをサーバーに配置する前に、System Manager を使用して登録済み Web サービスをテストすることもできます。

Web サービスの登録

System Manager では、「新規 Web サービスを登録」ウィザードを使用して、Web サービスを登録します。

1. ウィザードを開始するには、WebSphere Business Integration システム・ビューで「統合コンポーネント・ライブラリー」を展開し、「Web サービス」フォルダーを右クリックし、「新規 Web サービスの登録」を選択します。「Web サービスを登録する方法の選択」ウィンドウが表示されます。
2. 次のいずれかのオプションをクリックします。

オプション	説明
WSDL ファイルの URL の入力	登録する Web サービスを正確に指定することができます。
Web サービスの UDDI レジストリーの検索	UDDI レジストリーを指定し、そこから Web サービスを検索することができます。

3. 「次へ」をクリックします。
4. ステップ 2 で「WSDL ファイルの URL の入力」を選択した場合は、「WSDL URL を入力してください」ウィンドウが表示されます。「WSDL URL」フィールドに登録するサービスの URL を指定し、「次へ」をクリックしてステップ 8 (57 ページ) に進みます。
5. ステップ 2 で「Web サービスの UDDI レジストリーの検索」をクリックした場合は、「Web サービスの検索」ウィンドウが表示されます。以下の手順を実行します。
 - a. 「検索条件」フィールドに、検索する Web サービスの検索条件を入力します。
 - b. 以下のサポートされる UDDI レジストリーを 1 つ以上選択します。
 - IBM UDDI レジストリー

- Microsoft UDDI レジストリー
 - XMethods UDDI レジストリー
- c. 「**検索**」をクリックします。「**検索結果**」タブに、結果が見つかったレジストリーの名前が表示されます。選択したレジストリーが使用可能でない場合は、エラー・メッセージが表示されます。
 6. リストのそれぞれのレジストリーごとに、以下を実行します。
 - レジストリー名を選択して、そのレジストリーの検索結果の数を表示します。
 - レジストリー・リストを展開して、そのレジストリーで見つかったサービスの名前を表示します。
 - サービスの名前を選択して、Web サービスの詳細を表示します。
 7. 登録したい Web サービスの名前を選択し、「**次へ**」をクリックしてウィザードの次のページに進みます。「**新規 Web サービスが見つかりました**」ウィンドウが表示されます。
 8. 「**新規 Web サービスが見つかりました**」ページでは、以下を実行できます。
 - 「**使用可能なメソッド**」タブをクリックして、Web サービスの呼び出し可能なサービスのリストを表示します。基本的に、これらのサービスは、クライアントが呼び出すことのできるメソッドです。
 - 「**WSDL**」タブをクリックして、Web サービスを記述している XML コードを表示します。
 9. 「**使用可能なメソッド**」をクリックすると、詳細情報が、(有効であれば) 右方のペインに表示されます。Fault(s) パラメーターは、サービスを呼び出したときに発生する可能性のあるエラーまたは例外を保管するために使用されることに注意してください。
 10. 「**完了**」をクリックして、Web サービスを登録します。

登録済み Web サービスの表示

Web サービスを登録すると、System Manager で表示できます。

1. 「**統合コンポーネント・ライブラリー**」>「**Web サービス**」の下で、Web サービスの名前をダブルクリックします。「**概要**」タブが表示されます。
2. 「**概要**」タブをクリックして、「**使用可能な操作**」ウィンドウを表示します。このウィンドウには、その Web サービスに関連するサービス、つまり、メソッドのリストがあります。「**選択可能な操作**」ページの右側に表示される「**Web サービス・メソッド**」パネルにも、メソッドのリストが表示されます。

Web サービスのすべてのメソッドは、これらを使用する意図があるかどうかに関わらず、登録されて「**選択可能な操作**」ページの「**操作**」列に表示されます。

Web サービスを登録したとき (56 ページの『Web サービスの登録』を参照) に各メソッドのために作成されたビジネス・オブジェクトが、「**入力ビジネス・オブジェクト**」列と「**出力ビジネス・オブジェクト**」列の下にリストされます。

3. ビジネス・オブジェクトを再作成するには、「**ビジネス・オブジェクト定義の作成 (Create Business Objects Definitions)**」をクリックします。 .

4. 「Web サービス・メソッド」パネルで、1つのメソッドを選択します。これにより、そのメソッドのパラメーターが「メソッド・パラメーター」パネルに表示されます。パラメーターは、「入力」、「出力」、および「障害」の各カテゴリーの下にリストされます。

登録済み Web サービスのテスト

登録済み Web サービスを統合サーバーに配置する前に、この Web サービスをテストすることができます。

1. 「統合コンポーネント・ライブラリー」 > 「Web サービス」の下で、Web サービスの名前をダブルクリックします。「概要」タブが、「Web サービス・メソッド」パネルおよび「メソッド・パラメーター」パネルとともに表示されます。
2. 「Web サービス・メソッド」パネルで、1つのメソッドを選択します。
3. 「概要」ページで、メソッド名をダブルクリックするか、メソッド名を右クリックして「テスト」を選択し、メソッドをテストします。選択したメソッドのテスト・ページが、メソッドの入力ビジネス・オブジェクトおよび出力ビジネス・オブジェクトの名前と共に表示されます。このページを使用して、選択したメソッドをテストします。
4. ビジネス・オブジェクトをテストするには、以下の手順を実行します。
 - 「新規要求の作成 (Create New Request)」をクリックします。左側のペインに、要求ビジネス・オブジェクトが作成されます。入力パラメーターを提供します。属性 SOAPConfigMO、 ProtocolConfigMO、 ObjectEventID は無視することができます。
 - 「Invoke Web Services」をクリックして、Web サービスを呼び出します。Web サービスからの応答が、「応答ビジネス・オブジェクト」パネルに表示されます。
5. 「概要」ページで各メソッドをダブルクリックして、テスト・ページでそのメソッドをテストします。

既存のコンポーネントの変更

ビジネス・オブジェクト、コネクタ、マップ、関係、およびコラボレーション・テンプレートを変更するには、以下の操作を実行します。

- ライブラリーのコンポーネントまたはユーザー・プロジェクトのコンポーネントへのショートカットをダブルクリックします。
- ライブラリーのコンポーネントまたはユーザー・プロジェクトのコンポーネントへのショートカットを選択し、以下のいずれかの操作を実行します。
 - 対応する Designer ツールを起動します。詳細については、52 ページの『Designer ツールの起動』を参照してください。
 - **Enter** を押します。
 - **Ctrl+E** を押します。
 - メニュー・バーから「コンポーネント」 > 「定義を編集」を選択します。
- ライブラリー内のコンポーネントまたはユーザー・プロジェクト内のコンポーネントへのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「定義を編集」を選択します。

- コンポーネントの Designer ツールを起動し (52 ページの『Designer ツールの起動』を参照)、ツールが起動したらコンポーネントを開きます。

ベンチマーク、コラボレーション・オブジェクト、データベース接続プール、スケジュールのコンポーネントの変更方法については、表 4 を参照してください。

表 4. Designer ツールを使用しない統合コンポーネントの変更方法

コンポーネント	方法	詳細情報の参照先
ベンチマーク	<p>「WebSphere Business Integration」システム表示で「ベンチマーク」フォルダーを右マウス・ボタンでクリックし、「ベンチマークのセットアップ」コンテキスト・メニューから以下のいずれかを選択します。</p> <ul style="list-style-type: none"> • 新規ベンチマークを構成 • 既存のベンチマークを編集 • ワークロードをファイルに生成 	「Benchmarking Guide」
コラボレーション・オブジェクト	WebSphere Business Integration System ビューで、「コラボレーション・オブジェクト」フォルダー内のコラボレーション・オブジェクトをダブルクリックします。	219 ページの『第 9 章 コラボレーション・オブジェクトの構成』
データベース接続プール	データベース接続プールは変更できません。プール・コンポーネントの一部のプロパティは変更できますが、プールが接続するデータベースなどの定義エレメントは変更できません。	207 ページの『第 8 章 データベース接続プールの構成』
スケジュール	<ul style="list-style-type: none"> • 「WebSphere Business Integration」システム表示で「スケジュール」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンポーネントのスケジュールを編集」を選択します。 • ライブラリーでスケジュール可能な任意のコンポーネント、またはユーザー・プロジェクトでスケジュール可能な任意のコンポーネントへのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンポーネントのスケジュールを編集」を選択します。 <p>「スケジュール」インターフェースが表示され、定義済みのスケジュールを変更できるようになります。</p>	「システム管理ガイド」

インポート・ウィザードを使用するサーバーからライブラリーへのコンポーネントのインポート

InterChange Server リポジトリからライブラリーに統合コンポーネントをインポートできます。この機能は、開発チームに参加したときに既存のコンポーネントがある個々の開発環境を更新する必要がある場合に特に役立ちます。

コンポーネントは、ほかの開発者のライブラリーからインポートできますが、コンポーネントが適切に保守されていない場合は、既知の品質のコンポーネントのみを使用することになります。通常、テスト環境および実稼働環境は適切に保守されており、一般に個々の開発環境よりも信頼性が高いと考えられます。

InterChange Server リポジトリから統合コンポーネント・ライブラリーにコンポーネントをインポートするには、以下の手順を実行します。

注: InterChange Server インスタンスからライブラリーにコンポーネントをインポートするには、サーバーが実行されている必要があります。「コンポーネントをインポート」ウィザードを始動したときに System Manager がサーバーに接続されていない場合は、インポートできるコンポーネントは表示されません。

1. System Manager を InterChange Server に接続します。詳細については、37 ページの『InterChange Server への接続』を参照してください。
2. WebSphere Business Integration System Manager ビューで、コンポーネントのインポート先ライブラリーを右マウス・ボタンでクリックし、コンテキスト・メニューから「サーバーからコンポーネントをインポート」を選択し、以下のいずれかの操作を実行して「コンポーネントをインポート」ウィザードを始動します。
 - サブメニューから「コンポーネント」を選択し、サーバーにあるすべてのタイプのコンポーネントを表示させます。
 - サブメニューからコンポーネント・タイプを選択し、そのタイプのコンポーネントのみを表示させます。

System Manager によって、図 11 に示すような「サーバーから追加のコンポーネントをインポート」ウィザードが表示されます。

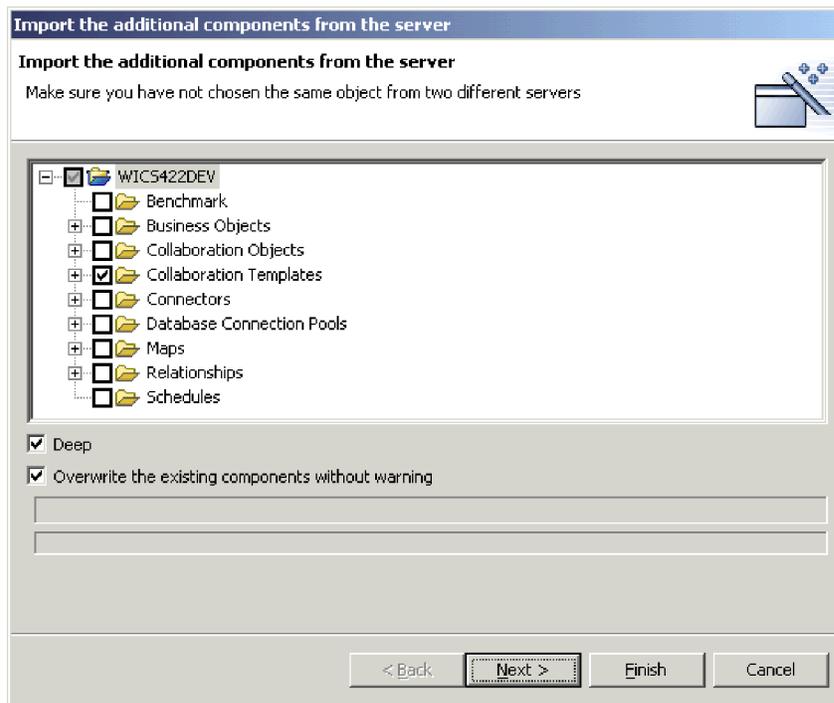


図 11. サーバーからのコンポーネントのインポート

3. サーバー、コンポーネント・グループ、またはサーバーからインポートする個々のコンポーネントの横にあるチェック・ボックスを有効にします。
4. コンポーネントの依存関係もインポートする場合は、「**ディープ**」チェック・ボックスを有効にします。依存関係については、94 ページの『依存関係および参照』を参照してください。
5. サーバーからインポートするコンポーネントと同じ名前のコンポーネントがライブラリーに存在するときにそのコンポーネントを上書きする場合は、「**警告せず既存のコンポーネントを上書きします**」チェック・ボックスを有効にします。

マップまたはコラボレーション・テンプレートを選択しなかった場合は、ステップ 7 (62 ページ) に進みます。

マップまたはコラボレーション・テンプレートのインポートを選択した場合は、以下のいずれかを実行します。

- 4.2.2 よりも前のサーバー・インスタンスからマップおよびコラボレーション・テンプレートをインポートする場合、およびこの時点でマップおよびコラボレーション・テンプレートを 4.2.2 のフォーマットにアップグレードしたい場合は、ステップ 6 に進んでウィザードの次の画面に進みます。
 - インポートするマップおよびコラボレーション・テンプレートがすでに 4.2.2 のフォーマットである場合、または この時点でのアップグレードを望まない場合は、ステップ 7 (62 ページ) に進んでウィザードを完了します。
6. 「コンポーネントをインポート」ウィザードの 2 番目の画面で、「**マップ**」および「**コラボレーション・テンプレート**」の両チェック・ボックスを有効にし、選択したコンポーネントの間でマップやコラボレーション・テンプレートを、System Manager に 4.2.2 形式に変換させます。

repos_copy を使用して、4.2.2 より前のリリースからリリース 4.2.2 のサーバー・リポジトリにコンポーネントを展開した場合は、そこに格納されたマップやコラボレーション・テンプレートは、アップグレードされません。それらのコンポーネントを 4.2.2 形式にアップグレードするには、System Manager または Map Designer と Process Designer の両ツールを使用する必要があります。

「コンポーネントをインポート」ウィザードの 2 番目の画面では、「**マップ**」および「**コラボレーション・テンプレート**」チェック・ボックスを有効にしてこの時点で変換させることも、これらのチェック・ボックスを無効にして変換を据え置き、後から Designer ツールで変換することもできます。これらのタイプのコンポーネントが多いときは特に変換に時間がかかる場合があるため、変換を据え置くこともできます。

62 ページの図 12 に、「コンポーネントをインポート」ウィザードの 2 番目の画面を示します。

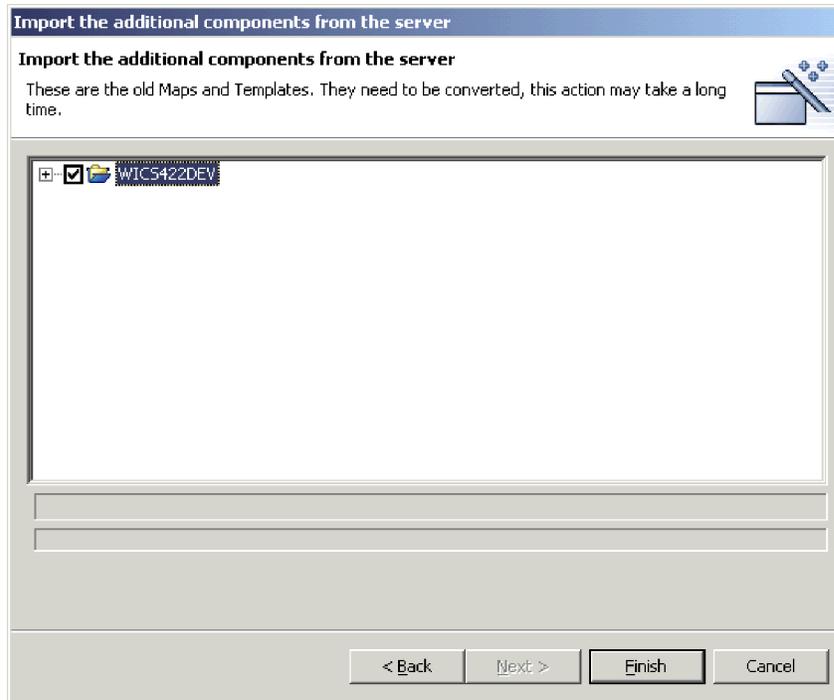


図 12. サーバーからインポートするときのマップおよびコラボレーション・テンプレートの変換

7. 「完了」をクリックします。

ドラッグ・アンド・ドロップを使用するサーバーからライブラリーへのコンポーネントのインポート

System Manager ビューで、ドラッグ・アンド・ドロップ操作を使用して、統合コンポーネント・ライブラリーにコンポーネントをインポートするには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、以下の操作を行って、展開するコンポーネントを選択します。
 - ユーザー・プロジェクトまたは統合コンポーネント・ライブラリー内のフォルダーを選択して、同じタイプのコンポーネントをすべて選択します。
 - コンポーネントのフォルダーを展開し、コンポーネントを個別に選択して、コンポーネント・フォルダーを選択します。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。
 - **Shift** を押したまま、連続する項目を選択します。
 - **Ctrl** を押したまま、連続しない項目を選択します。
2. WebSphere Business Integration System Manager ビューで、コンポーネントをインポートする先の統合コンポーネント・ライブラリー上に選択したリソースをドラッグ・アンド・ドロップします。

System Manager が、指定した統合コンポーネント・ライブラリーにコンポーネントをインポートしようとします。メッセージとエラーは、コンソール・ビューに表示されます。

パッケージからライブラリーへのコンポーネントのインポート

68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』の説明に従って、統合コンポーネントを .jar パッケージ・ファイルにエクスポートします。これにより、コンポーネントの環境間での移行、他の開発者との共用、およびテクニカル・サポートへのサブミットが容易になります。

パッケージから統合コンポーネント・ライブラリーにコンポーネントをインポートするには、以下の手順を実行します。

要確認: インポートするパッケージのコンポーネントと同じ名前のコンポーネントが存在する場合、System Manager は警告を出さずに既存のコンポーネントを上書きします。

1. 統合コンポーネント・ライブラリーを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリ・ファイルのインポート」を選択します。

System Manager によって、図 13 に示すような「リポジトリ・ファイルのインポート」ウィザードが表示されます。

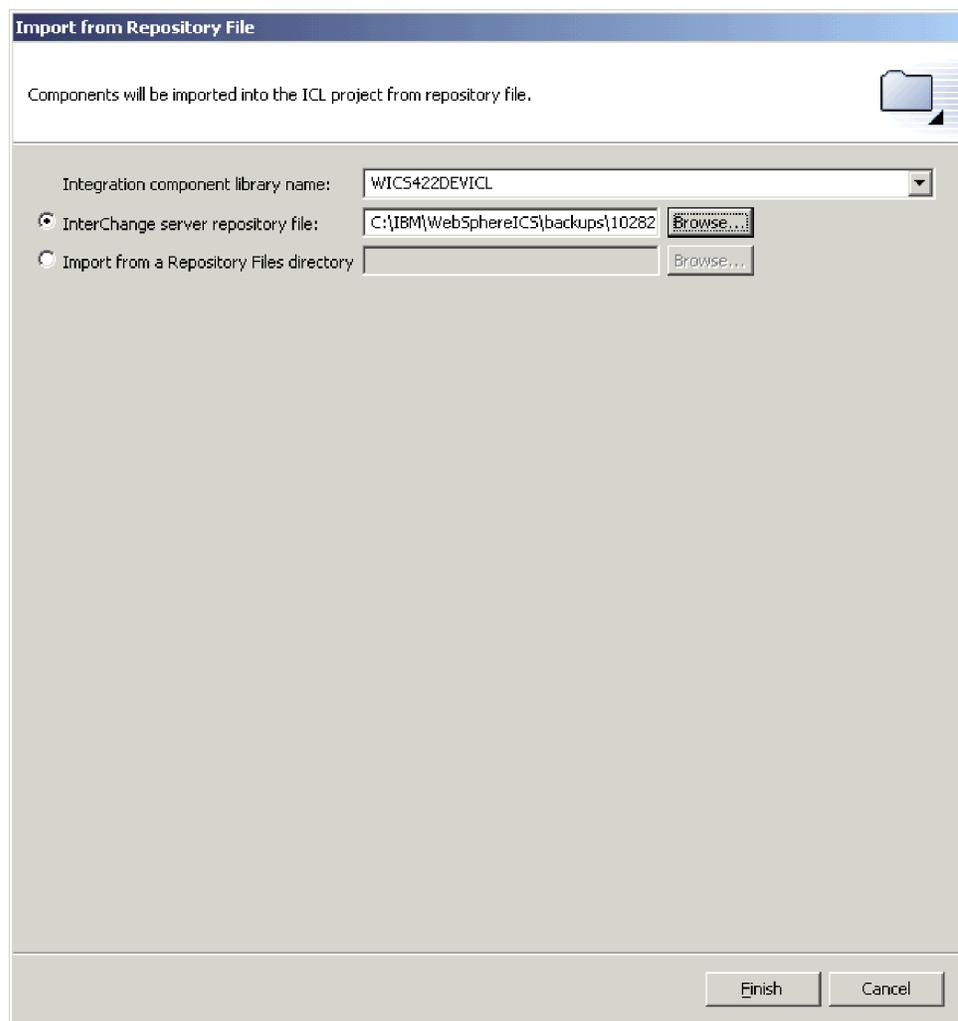


図 13. パッケージのインポート

2. 「リポジトリ・ファイルのインポート」画面で、「統合コンポーネント・ライブラリー名」ドロップダウン・メニューに、コンポーネントをインポートする先ライブラリーの名前が含まれていることを確認します。

コンポーネントをインポートする先とは異なるライブラリーから「リポジトリ・ファイルのインポート」ウィザードを起動した場合、ウィザードを終了してから、再起動することなく、以下の方法でインポート先を変更できます。

3. インポートするコンポーネントを指定するには、以下のいずれかを実行します。
 - 単一パッケージ・ファイルをインポートするには、「**InterChange Server リポジトリ・ファイル**」フィールドに、インポートする .jar ファイルの絶対パスと名前を入力するか、「参照」をクリックしてファイルを選択します。
 - パッケージ・ファイルを格納しているディレクトリ全体をインポートするには、「**リポジトリ・ファイル・ディレクトリからインポートします**」フィールドにディレクトリの絶対パスを入力するか、「参照」をクリックしてファイルを選択します。
4. 「完了」をクリックします。

注: ワークベンチで「ファイル」>「インポート」メニュー項目を使用してパッケージ・ファイルをインポートしないでください。「Zip ファイル」ウィザードは、.jar 拡張子を持つアーカイブを処理するもので、InterChange Server パッケージ・ファイルには .jar 拡張子が付いていますが、「Zip ファイル」ウィザードは、パッケージ・ファイルに対しては正常に機能しません。

統合コンポーネント・ライブラリーにおけるマップとコラボレーション・オブジェクト・プロパティーの変更

マップおよびコラボレーション・オブジェクトには変更可能なプロパティーがあります。これらのプロパティーを変更すると、コンポーネントの振る舞いを変更できます。コンポーネントのプロパティーを変更するには、統合コンポーネント・ライブラリーのコンポーネントまたはユーザー・プロジェクトのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティー」を選択します。System Manager に、コンポーネントのプロパティーを構成するためのダイアログが表示されます。

コラボレーション・オブジェクト・プロパティーについては、219 ページの『第 9 章 コラボレーション・オブジェクトの構成』を参照してください。

Map Designer における重要な機能の 1 つは、同じ名前のソースおよび宛先属性を持つビジネス・オブジェクト間のマップを自動的に生成できることです。マップ・プロパティーの詳細については、「マップ開発ガイド」を参照してください。

データベース接続プールの検証

データベース接続プール・コンポーネントには「接続を検証」というコンテキスト・メニュー項目があります。このメニュー項目については、214 ページの『データベース接続プールの検証』を参照してください。

マップおよびコラボレーション・テンプレートのコンパイル

マップおよびコラボレーション・テンプレートは System Manager でコンパイルできます。コンポーネントを展開したりパッケージにエクスポートしたりするにはコンパイルする必要があるため、この機能は非常に便利です。System Manager では、非同期コンパイルを使用します。これは、1 つのコンポーネントのコンパイル中に、System Manager が別のタスクを実行できることを意味します。

コンポーネントのコンパイル時には、項目に基づいた進行レポートと進行状況表示バーを使用します。この情報は、サーバーのキャッシュ・テーブルに保持されません。進行状況は、各成果物で操作が完了すると更新されます。

注: System Manager は非同期コンパイルを使用しますが、一度に 1 つのコンパイルのみを実行できます。

コンパイルするコンポーネントが、IBM で提供していないライブラリー (ユーザー自身が作成したライブラリー、別のベンダーが提供しているライブラリーなど) を使用する場合は、System Manager がそれらのライブラリーを参照するように構成する必要があります。「コンパイラー」設定インターフェースについて詳しくは、4 (104 ページ) を参照してください。

マップまたはコラボレーション・テンプレートをコンパイルするには、以下のいずれかの操作を実行します。

- 統合コンポーネント・ライブラリーのマップまたはコラボレーション・テンプレート、またはユーザー・プロジェクトにおけるショートカットを選択し、以下のいずれかの操作を実行します。
 - **Ctrl+F7** を押します。
 - メニュー・バーから「**コンポーネント**」>「**コンパイル**」を選択します。
- 統合コンポーネント・ライブラリーのマップまたはコラボレーション・テンプレート、またはユーザー・プロジェクトにおけるショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「**コンパイル**」を選択します。

統合コンポーネント・ライブラリーのマップまたはコラボレーション・テンプレートをすべてコンパイルするには、以下のいずれかの操作を実行します。

- 統合コンポーネント・ライブラリーの「**マップ**」または「**コラボレーション・テンプレート**」フォルダーを選択し、以下のいずれかの操作を実行します。
 - **F7** を押します。
 - メニュー・バーから「**コンポーネント**」>「**すべてコンパイル**」を選択します。
- 統合コンポーネント・ライブラリーの「**マップ**」または「**コラボレーション・テンプレート**」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「**すべてコンパイル**」を選択します。

マップをサブマップとともにコンパイルするには、統合コンポーネント・ライブラリーのマップ、またはユーザー・プロジェクトのマップへのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サブマップでコンパイル**」を選択します。

ソリューションの処理

ユーザー・プロジェクトはソリューションとしてエクスポートできます。このアクションは、ショートカットの他、統合コンポーネント・ライブラリーでショートカットが参照するコンポーネント定義もユーザー・プロジェクトからコピーします。これにより、ある環境から別の環境に全インターフェースまたはビジネス・インテグレーション・システムを容易に移行できます。

ソリューションのエクスポート

参照するユーザー・プロジェクトおよび統合コンポーネントをソリューションとしてエクスポートするには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、「**InterChange Server プロジェクト**」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ソリューションをエクスポート**」を選択します。

System Manager によって、図 14 に示すような「ソリューションのエクスポート」ウィザードが表示されます。

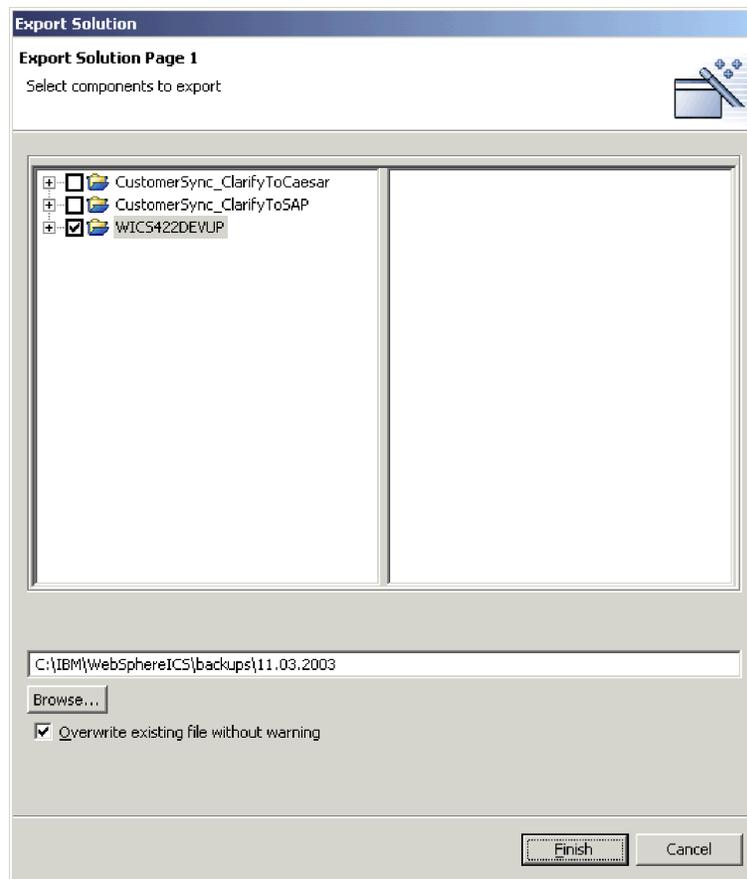


図 14. ソリューションのエクスポート

2. 以下の方法で、エクスポートするコンポーネントを選択します。

- ユーザー・プロジェクトの横のチェック・ボックスを有効にして、プロジェクト内のすべてのコンポーネントを選択します。
 - コンポーネント・グループの横のチェック・ボックスを有効にして、グループ内のすべてのコンポーネントを選択します。
 - コンポーネント・グループを強調表示し、右側のペインで個々のコンポーネントの横にあるチェック・ボックスを有効にして、それらのコンポーネントを選択します。
3. ウィザード画面の下部にあるテキスト・フィールドにソリューションのエクスポート先ディレクトリーの絶対パスおよび名前を入力するか、「参照」をクリックして所定のディレクトリーにナビゲートします。
 4. 「完了」をクリックします。

System Manager が以下の処理を実行し、ステップ 3 で指定したディレクトリーにソリューションをエクスポートします。

- ソリューションのエクスポート時に選択したユーザー・プロジェクトに、ショートカットを含む User ディレクトリーを作成します。
 - ソリューションのエクスポート時に選択したユーザー・プロジェクトに、ショートカットによって参照される統合コンポーネント・ライブラリーのディレクトリーを含む System ディレクトリーを作成します。
5. エクスポート操作が正常に完了したことがプロンプトで表示されたら、「OK」をクリックします。

ソリューションのインポート

ソリューションをインポートするには、以下の手順を実行します。

1. WebSphere Business Integration System Manager ビューで、「ユーザー・プロジェクト」フォルダーを展開し、「InterChange Server プロジェクト」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「ソリューションをインポート」を選択します。

System Manager によって、68 ページの図 15 に示すような「ソリューションのインポート」ウィザードが表示されます。

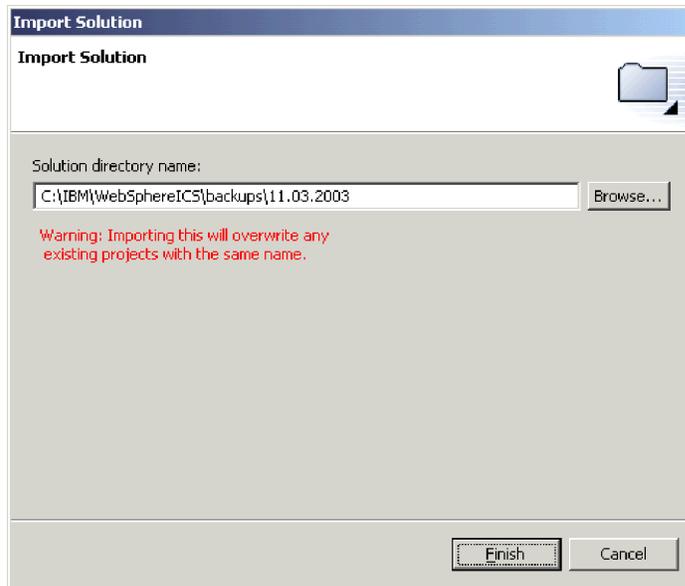


図 15. ソリューションのインポート

2. エクスポートされたソリューションが存在するディレクトリーの絶対パスとディレクトリー名を「ソリューション・ディレクトリー名」フィールドに入力するか、「参照」をクリックして目的のディレクトリーまでナビゲートします。
3. 「完了」をクリックします。

System Manager が、エクスポートされたソリューションで定義されている統合コンポーネント・ライブラリーおよびユーザー・プロジェクトを環境に作成します。

System Manager を使用したパッケージへのコンポーネントのエクスポート

統合コンポーネントはパッケージ・ファイルにエクスポートできます。統合コンポーネントはリソースであり、最終的には、ファイル・システムに格納されるファイルとなります。詳細については、29 ページの『リソース』を参照してください。System Manager は、コンポーネントをパッケージにエクスポートするときに以下のリソースを .jar (Java アーカイブ) ファイルに圧縮します。

- 定義ファイル (コンポーネント・タイプに応じた拡張子で XML 形式で格納します)
- マップおよびコラボレーション・テンプレートの Java ソース・ファイル
- メッセージ・ファイル

コンポーネントをパッケージにエクスポートするには、以下の手順を実行します。

要確認: マップまたはコラボレーション・テンプレートをエクスポートするには、コンパイルする必要があります。コンパイルされていないマップまたはコラボレーション・テンプレートをエクスポートしようとすると、System Manager は先にコンパイルするように要求します。

1. エクスポート対象のコンポーネントを含んでいる統合コンポーネント・ライブラリーまたはユーザー・プロジェクトのいずれかを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリ・ファイルとしてエクスポート」を選択します。

System Manager によって、図 16 に示すような「リポジトリ・ファイルのエクスポート」ウィザードが表示されます。

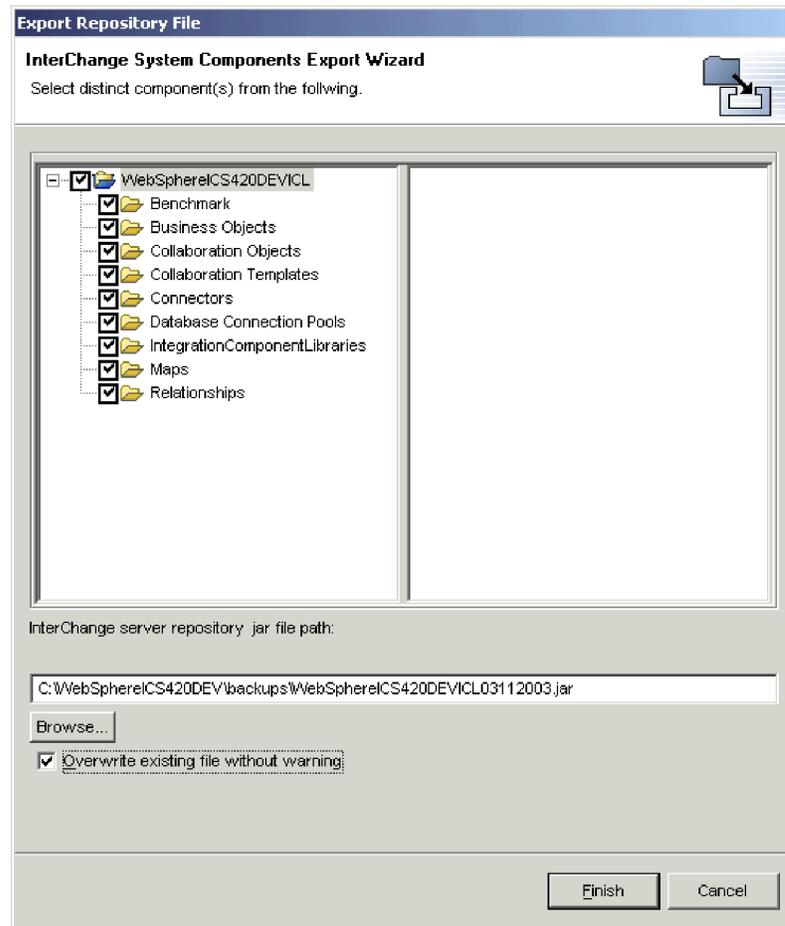


図 16. パッケージのエクスポート

2. 以下の方法で、エクスポートするコンポーネントを選択します。
 - 統合コンポーネント・ライブラリーまたはユーザー・プロジェクトの横のチェック・ボックスを有効にして、ライブラリーまたはプロジェクト内のすべてのコンポーネントを選択します。
 - コンポーネント・グループの横のチェック・ボックスを有効にして、グループ内のすべてのコンポーネントを選択します。
 - コンポーネント・グループを強調表示し、右側のペインで個々のコンポーネントの横にあるチェック・ボックスを有効にして、それらのコンポーネントを選択します。
3. コンポーネントのエクスポート先として既存の .jar ファイルを指定し、かつプロンプトを表示せずにファイルを上書きする場合は、「警告せずに既存のファイルを上書き」チェック・ボックスを有効にします。

注: 「警告せずに既存のファイルを上書き」チェック・ボックスの効果を機能させるには、ステップ 4 で説明するように、使用するファイルの指定の前にこのチェック・ボックスを有効にする必要があります。System Manager では、既存ファイルがすでに指定されていることを検出すると、ウィザードの完了を待たずに、すぐにファイルの上書きを確認するプロンプトが表示されるため、チェック・ボックスの効果を機能させるには、あらかじめこのオプションを有効にしておく必要があります。

4. コンポーネントのエクスポート先の .jar ファイルの名前およびパスを「InterChange サーバー・リポジトリ JAR ファイル・パス」フィールドに入力するか、「参照」をクリックし、上書きするファイルを選択するか、またはディレクトリにナビゲートしてファイル名を指定します。

既存ファイルの名前およびパスを指定するときに「警告せずに既存ファイルを上書き」チェック・ボックスを有効にしなかった場合、既存のファイルを上書きするには、プロンプトが表示されたときに「はい」をクリックします。

注: ファイルの名前およびパスをフィールドに入力する場合は、.jar 拡張子を含めなければ「完了」ボタンは使用可能になりません。

5. 「完了」をクリックして、ウィザードを終了します。

サーバーへのコンポーネントの展開

3 ページの『インターフェースのライフ・サイクル』で説明しているように、統合コンポーネントはローカル・ファイル・システムのライブラリーに作成し、InterChange Server インスタンスに展開して実行可能にします。サーバーにコンポーネントが配置されると、サーバーのランタイムが更新されて、これらのコンポーネントが直ちに使用できるようになります。

System Manager のグラフィカル・インターフェースまたは repos_copy コマンド行インターフェースのいずれかを使用して、統合コンポーネントのパッケージを配置できます。System Manager の使用については、77 ページの『配置ウィザードを使用するコンポーネントの展開』を参照してください。この作業に repos_copy コマンド行インターフェースを使用するには、最初にパッケージ・ファイルを作成する必要があります。パッケージ・ファイルの作成の詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。各インターフェースの利点および欠点については、『System Manager を使用した展開または repos_copy を使用した展開の決定』を参照してください。

System Manager を使用した展開または repos_copy を使用した展開の決定

71 ページの表 5 に、展開に System Manager を使用する場合と repos_copy を使用する場合の利点および欠点を示します。2 つのインターフェースを評価し、ニーズに適したインターフェースを使用してください。

表 5. 展開における System Manager および repos_copy の利点および欠点

インターフェース	利点	欠点
System Manager	<ul style="list-style-type: none"> • System Manager はグラフィカル・インターフェースを使用します。 • 展開するユーザー・プロジェクトのコンポーネントを選択できます。 • 展開時にサーバー・リポジトリに関係スキーマを作成するか作成しないかを選択できます。 • 同時に複数のユーザー・プロジェクトを展開できます。 • ドラッグ・アンド・ドロップ展開方法を使用すると、ユーザー・プロジェクト、統合コンポーネント・ライブラリー、または個々のコンポーネントも簡単に展開できます。 	<ul style="list-style-type: none"> • System Manager は Windows でのみ動作します。
repos_copy	<ul style="list-style-type: none"> • コンポーネントをグループ化する必要がありません (System Manager で展開する場合はユーザー・プロジェクトに入れる必要があります)。 • repos_copy はプラットフォームに依存しません。 	<ul style="list-style-type: none"> • repos_copy はコマンド行インターフェースを使用します。 • 最初に System Manager と repos_copy のいずれかを使用して、展開するコンポーネントをパッケージ・ファイルにエクスポートしておく必要があります。 • パッケージ・ファイルのコンポーネントを選択して展開することはできません。 • 一度に展開できるパッケージ・ファイルは 1 つのみです。

展開用のコンポーネントの初期状態の設定

状態を持つコンポーネント (コネクタ、コラボレーション・オブジェクト、マップ、関係、データベース接続プールなど) の場合、展開した後に、サーバーのリブート時にコンポーネントが初期化される状態を設定できます。コンポーネントの展開後の初期状態を設定するには、以下の手順を実行します。

1. ユーザー・プロジェクトに設定する初期状態を持つコンポーネントのショートカットを選択します。

複数のコンポーネントを選択する場合、以下のような Windows の標準的な選択方法を使用できます。

- **Shift** を押したまま、連続する項目を選択します。
- **Ctrl** を押したまま、連続しない項目を選択します。

2. ステップで選択した項目を右マウス・ボタンでクリックし、以下のいずれかを実行します。
 - 「開始」または「すべて開始」を選択して、選択したコンポーネントを「実行」状態で初期化します。
 - 「停止」または「すべて停止」を選択して、選択したコンポーネントを「停止」状態で初期化します。

配置時プロパティの構成

配置時プロパティは、設計時ではなく、配置時に構成される、成果物のプロパティです。System Manager を使用すると、サーバーに対して固有の配置時プロパティを指定できます。

サーバーに成果物が配置されると、System Manager では、配置する前にその成果物の配置時プロパティを適用します。成果物のプロパティを複数のサーバーにコピーすることもできます。これにより、複数の異なるサーバーの複数の異なるコンポーネントのすべてのプロパティを追跡して記録する必要がなくなります。複数の異なるサーバーに対して異なる値を持つプロパティを識別し、これらのプロパティを配置構成の一部として追加することができます。その後、サーバーに応じて、これらのプロパティの値を構成できます。

サーバーにオブジェクトが配置されると、System Manager では、配置する前にそれぞれのサーバー用に構成された値を自動的に適用します。配置時のプロパティを構成するには、統合コンポーネント・ライブラリー・プロジェクトを右マウス・ボタンでクリックし、「配置記述子の編集」を選択します。73 ページの図 17 が表示されます。

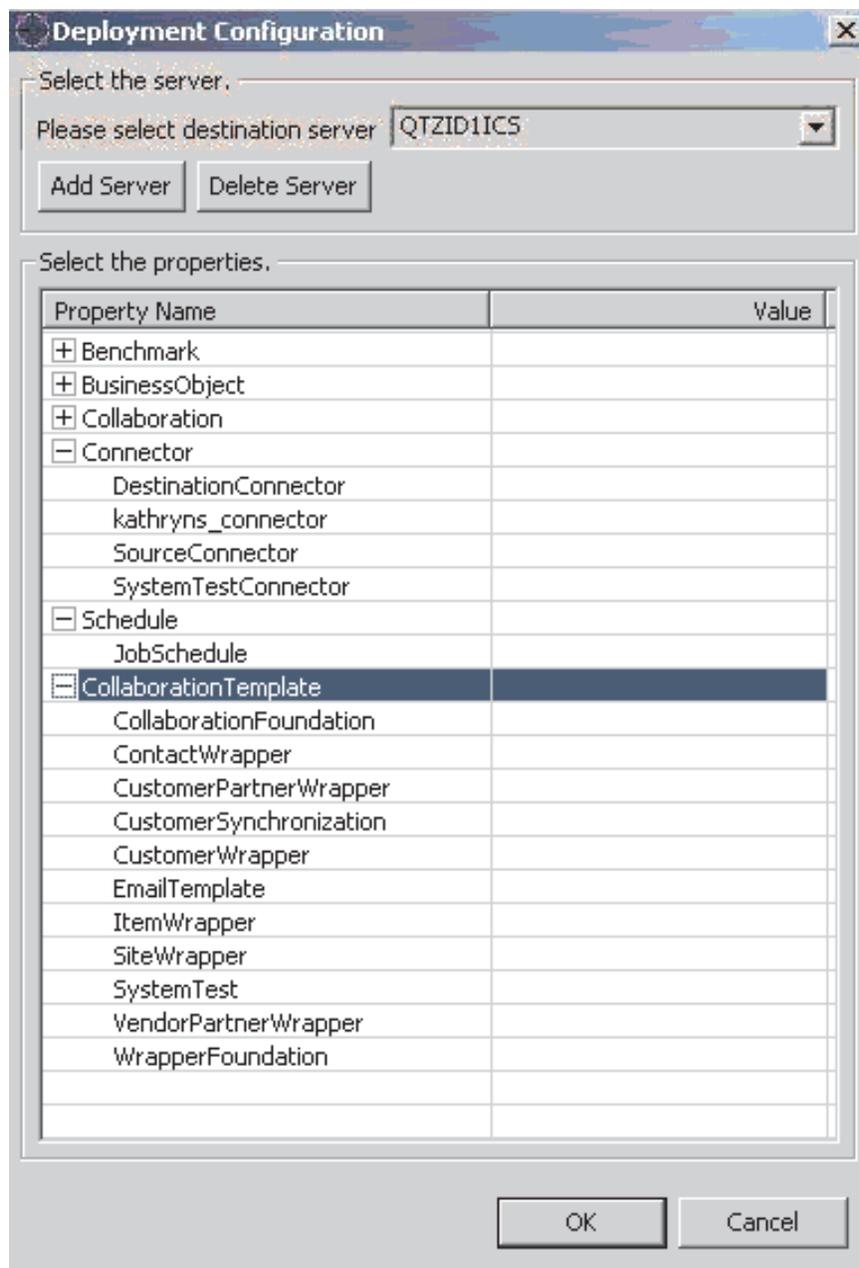


図 17. 「配置構成」 ウィンドウ

「配置構成」 ウィンドウで、以下を実行することができます。

1. サーバーの追加。サーバーを追加するには「サーバーの追加」をクリックします。ダイアログが表示されます。ここで、宛先サーバーの名前を入力します。「OK」をクリックします。

注: 「構成」 ダイアログですでにコンポーネントおよびプロパティを構成している場合、これらの値は、構成に追加されるすべてのサーバーに適用されません。

2. サーバーの削除。サーバーを削除するにはリストから削除するサーバーを選択して、「サーバーを削除」をクリックします。

3. プロパティの構成。「プロパティの選択」リストでは、以下を実行することができます。

- コンポーネントの追加。「配置構成」リストにコンポーネントを追加するには、空のセルを右マウス・ボタンでクリックし、追加するコンポーネントを「**コンポーネントの選択**」ダイアログから選択します。
- プロパティの変更。プロパティを変更するには、変更するプロパティのセルをダブルクリックし、「**値**」列に適切な設定値を入力します。
- プロパティの追加。プロパティを追加するには、プロパティを追加するコンポーネントを右マウス・ボタンでクリックし、表示されたコンテキスト・メニューから「**プロパティを追加**」を選択します。「プロパティの選択」ダイアログに表示されたリストから、追加するプロパティを選択し、「**OK**」をクリックします。
- プロパティの削除。特定のサーバー用の個々の配置プロパティを削除するには、サーバーを選択し、削除するプロパティを右マウス・ボタンでクリックして、コンテキスト・メニューから「**削除**」を選択します。

注: あるサーバー構成からプロパティを削除しても、他のサーバー構成からはそのプロパティは削除されません。

サーバー間でのプロパティのコピー

構成管理の利点の 1 つは、サーバー間でプロパティをコピーできることです。プロパティをコピーするには、プロパティを右クリックし、コンテキスト・メニューから「**コピー**」を選択します。プロパティを貼り付けるには、コンポーネントを右クリックし、「**貼り付け**」を選択します。1 つのサーバーから複数のプロパティをコピーして、別のサーバーで貼り付けることができます。プロパティをコピーすると、そのプロパティに関連付けられている値もコピーされます。

注: 貼り付けようとしているプロパティがすでにサーバー構成に存在する場合、貼り付け操作によって既存の値が上書きされます。

配置時プロパティのインポートとエクスポート

「配置構成」ダイアログでは、配置構成をファイルにエクスポートしたり、ファイルからインポートすることができます。プロパティをエクスポートするには、プロパティを右クリックして、メニューから「**エクスポート**」を選択します。「プロパティ記述子のエクスポート」ダイアログが表示されます。ここで、「**選択済みプロパティ**」または「**すべてのプロパティ**」を選択します。プロパティをエクスポートするファイルの名前を入力するか、「**参照**」を選択してファイルを探し出します。「**OK**」をクリックします。

プロパティをインポートするには、プロパティを右クリックし、コンテキスト・メニューから「**インポート**」を選択します。「プロパティのインポート」ダイアログが表示されます。ご使用のプロパティを上書きするか、マージするかを選択できます。「**プロパティ記述子ファイルを選択してください**」フィールドにファイルの名前を入力するか、「**ブラウズ**」を選択します。「**OK**」をクリックします。

repos_copy による配置プロパティの構成

System Manager から repos_copy ファイルを作成した場合、このファイルには、配置時プロパティが自動的に組み込まれます。これらのプロパティは、サーバーに送信する前に成果物に適用されます。repos_copy およびコマンド行オプションの使用については、143 ページの『第 6 章 repos_copy の使用』を参照してください。

System Manager を使用したパッケージの検証

展開を実行する前に、展開するコンポーネントから構成されるパッケージを検証して、展開を確実に成功させることができます。コンポーネントのパッケージを検証するには、以下の手順を実行します。

1. 検証する対象のコンポーネントのショートカットをユーザー・プロジェクトに追加します。

ユーザー・プロジェクトの作成については、47 ページの『ユーザー・プロジェクトの作成』を参照してください。ユーザー・プロジェクトへのショートカットの追加については、49 ページの『ユーザー・プロジェクトへのショートカットの追加』を参照してください。

注: 一度に複数のユーザー・プロジェクトを検証できるため、必要なすべてのコンポーネントのショートカットを適切なユーザー・プロジェクトに追加します。

2. WebSphere Business Integration System Manager ビューで、検証する対象のユーザー・プロジェクトを選択し、選択したユーザー・プロジェクトのいずれかを右マウス・ボタンでクリックし、コンテキスト・メニューから「**Validate user project**」を選択します。

System Manager によって、76 ページの図 18 に示すような「プロジェクトを検証」ウィザードが表示されます。

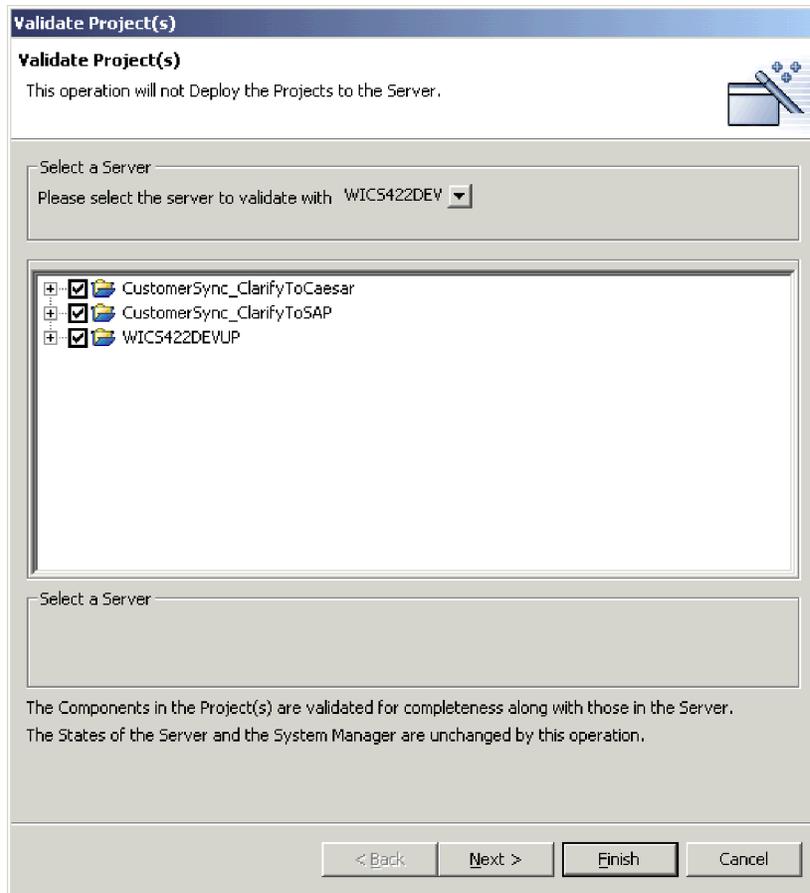


図 18. 検証するコンポーネントの選択

3. 「検証するサーバーを選択してください。」ドロップダウン・メニューから、コンポーネントを検証する比較相手のサーバーを選択します。
4. 以下の方法で、検証するコンポーネントを選択します。

- 統合コンポーネント・ライブラリーまたはユーザー・プロジェクトの横のチェック・ボックスを有効にして、ライブラリーまたはプロジェクト内のすべてのコンポーネントを選択します。
- コンポーネント・グループの横のチェック・ボックスを有効にして、グループ内のすべてのコンポーネントを選択します。
- コンポーネント・グループを展開し、個々のコンポーネントの横にあるチェック・ボックスを有効にして、それらのコンポーネントを選択します。

注: 複数のユーザー・プロジェクトに同じ名前が存在するコンポーネントを選択した場合は、ウィザードの次の画面で重複するコンポーネントからいずれか 1 つを選択するよう要求されます。

重複するコンポーネントを選択しなかった場合は、ステップ 6(77 ページ)に進みます。

重複するコンポーネントを選択した場合は、ステップ 5 に進みます。

5. 「次へ」をクリックします。

System Manager によって、図 19 に示すような「ローカル複写」画面が表示されます。

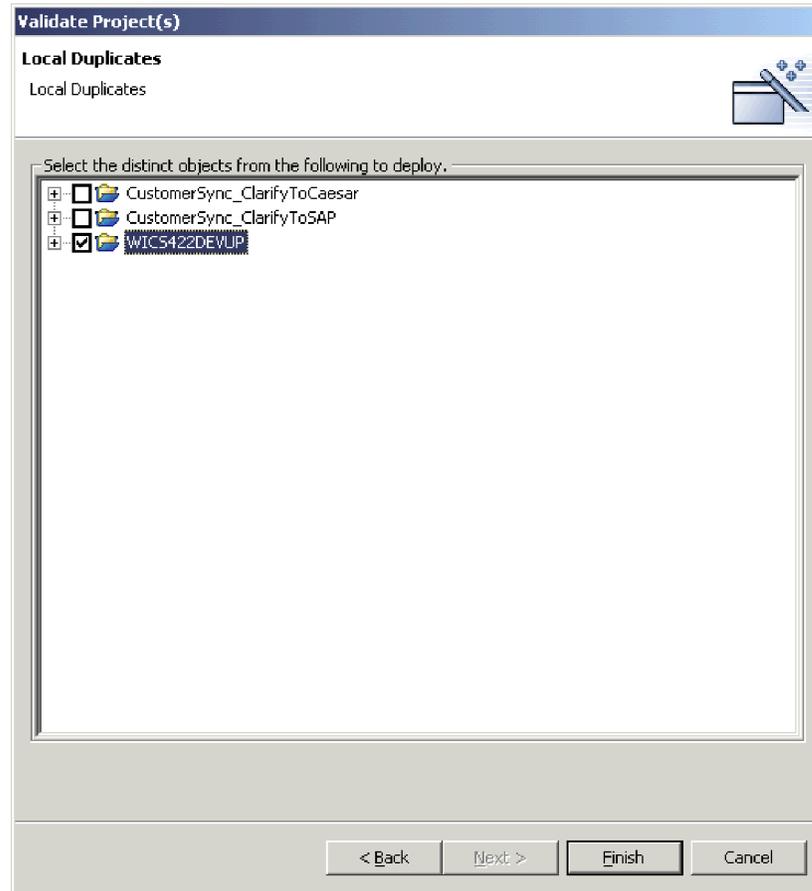


図 19. 検証するユーザー・プロジェクトからの個別のコンポーネントの選択

重複するコンポーネントから個別のコンポーネントを選択します。

6. 「完了」をクリックします。

System Manager は、選択されたコンポーネントを含むパッケージを作成し、それをサーバー・リポジトリに対して検証します。検証が成功したかどうかを示すメッセージが表示されます。

System Manager を使用するコンポーネントの展開

コンポーネントは、以下のセクションで説明されている 2 つの方法のいずれかで、InterChange Server インスタンスに展開できます。

- 『配置ウィザードを使用するコンポーネントの展開』
- 83 ページの『ドラッグ・アンド・ドロップを使用するコンポーネントの配置』

配置ウィザードを使用するコンポーネントの展開

配置ウィザードを使用して、InterChange Server インスタンスにコンポーネントのパッケージを展開するには、以下の手順を実行します。

1. 配置するコンポーネントがサーバー・リポジトリに存在する場合は (マップでない場合)、配置前にコンポーネントを停止する必要があります。

パッケージを展開する前に、InterChange Server コンポーネント管理ビューまたは System Monitor を使用し、サーバー内の重複するコンポーネントを停止します。詳細については、85 ページの『リポジトリ内のコンポーネントの状態の管理』を参照してください。

2. 展開するコンポーネントの初期状態を設定することをお勧めします。詳細については、71 ページの『展開用のコンポーネントの初期状態の設定』を参照してください。
3. 展開するコンポーネントのショートカットをユーザー・プロジェクトに追加します。

ユーザー・プロジェクトの作成については、47 ページの『ユーザー・プロジェクトの作成』を参照してください。ユーザー・プロジェクトへのショートカットの追加については、49 ページの『ユーザー・プロジェクトへのショートカットの追加』を参照してください。

注: 一度に複数のユーザー・プロジェクトを展開できるため、必要なすべてのコンポーネントのショートカットを適切なユーザー・プロジェクトに追加します。

4. 「WebSphere Business Integration System Manager」ビューで、サーバー・インスタンスに展開する各ユーザー・プロジェクトを選択します。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。
 - **Shift** を押したまま、連続する項目を選択します。
 - **Ctrl** を押したまま、連続しない項目を選択します。
5. 選択したユーザー・プロジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「ユーザー・プロジェクトを配置」を選択します。

System Manager によって、79 ページの図 20 に示すような「配置ウィザード (1 ページ)」画面が表示されます。

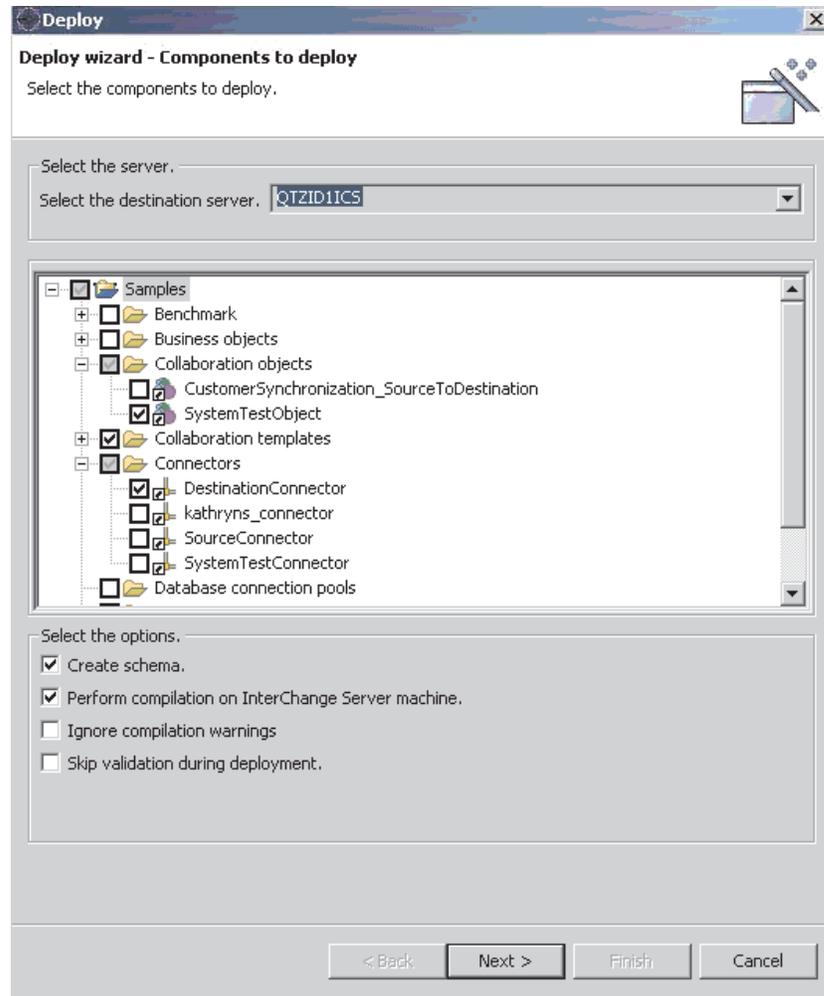


図 20. 展開可能コンポーネントおよび展開オプションの選択

6. 「配置可能なコンポーネント (Deployable Components)」画面で、以下の手順を実行します。
 - a. コンポーネントを配置する先のサーバー・インスタンスを、「宛先サーバーを選択してください」ドロップダウン・メニューから選択します。
 - b. 以下の方法を使用して、配置するコンポーネントを選択します。
 - プロジェクト内のすべてのコンポーネントを配置するには、ユーザー・プロジェクトの横のチェック・ボックスを有効にします。
 - ユーザー・プロジェクト・ノードを展開して、コンポーネント・グループ全体の横のチェック・ボックスを有効にして、そのタイプのコンポーネントをすべて配置します。
 - ユーザー・プロジェクト・ノード、次に、コンポーネント・グループ・ノードを展開して、個々のコンポーネントの横のチェック・ボックスを有効にして、有効にしたコンポーネントのみを配置します。

注: 別のユーザー・プロジェクトに同じ名前が存在するコンポーネントを選択した場合は、ウィザードの次の画面にプロンプトが表示され、重複しているコンポーネントのうちどのコンポーネントを配置するかを選択するように要求されます。

- c. 配置処理中に、データベース接続プールと関係スキーマを作成する場合は、「スキーマを作成」チェック・ボックスを有効にします。

関係スキーマとデータベース接続プールは、それらに参照することで機能するインターフェースに対して作成する必要があります。実行する目的でインターフェースを展開する場合は、インターフェースが使用するコンポーネントすべてに対して、このオプションを使用可能にする必要があります。

注: このオプションを有効にするのは、データベース接続プールおよび関係の設定がコンポーネントの展開先サーバーの適切な接続情報を参照する場合に限定してください。データベース設定が正しい環境からデータベース設定が無効になった環境にコンポーネントを移行する場合は、展開中にこれらのコンポーネントのスキーマを作成しないでください。

- d. InterChange Server にコンポーネントをコンパイルする場合は、「InterChange Server マシン上でコンパイルを実行 (Perform compilation on InterChange Server machine)」チェック・ボックスを有効にします。
- e. 使用すべきでないメソッドなど、コンパイル中に生成される警告に関するプロンプトを表示させないようにする場合は、「コンパイル警告を無視」チェック・ボックスを有効にします。
- f. 配置するオブジェクトすでに検証済みの場合は、「配置中の検証をスキップ」チェック・ボックスを有効にします。配置するオブジェクトを検証しようとしているため、通常このチェック・ボックスは選択しません。
- g. 選択内容に応じて、以下のいずれかを実行します。
- 選択したユーザー・プロジェクト全体にわたって重複するコンポーネントも、サーバー・リポジトリの既存のコンポーネントも選択しなかった場合は、この時点で展開を開始することが可能なので、ステップ 9 (82 ページ) に進んでください。
 - 選択したユーザー・プロジェクト全体にわたって重複するコンポーネントを選択しなかったが、サーバー・リポジトリの既存のコンポーネントを選択した場合は、「サーバー重複オブジェクト」画面が表示されます。ステップ 8 (81 ページ) に進んでください。
 - ユーザー・プロジェクト内で重複するコンポーネントを選択した場合は、ステップ 7 に進みます。

7. 「次へ」をクリックします。

ユーザー・プロジェクト内で重複するコンポーネントを選択した場合は、81 ページの図 21 に示すように、「配置ウィザード (2 ページ)」画面が表示されず。

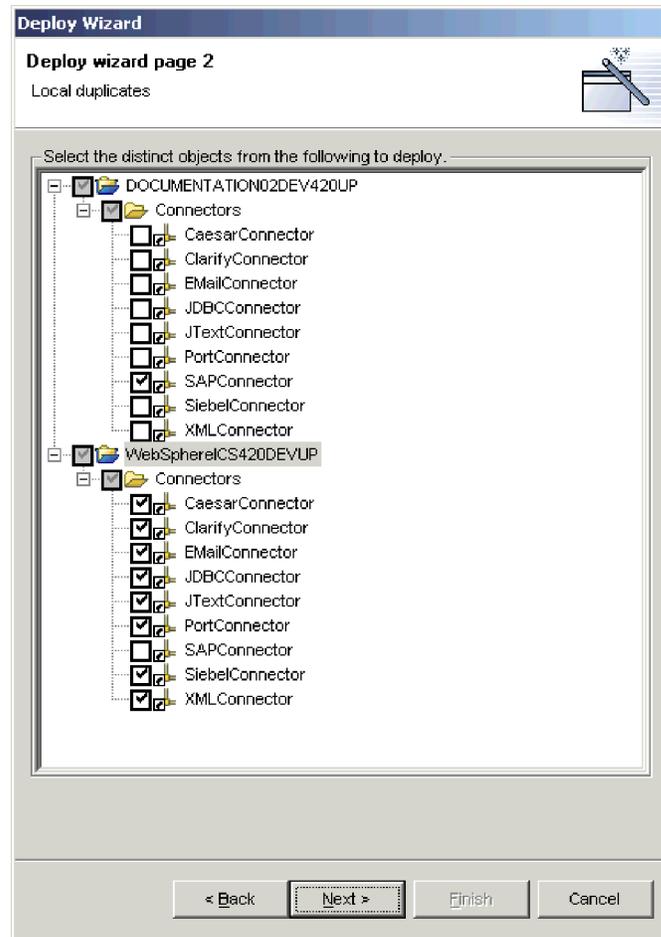


図 21. 展開時のローカルな重複の選択

表示されたユーザー・プロジェクトのフォルダーを展開し、重複しているローカル・コンポーネントのうち、配置する方のコンポーネントのチェック・ボックスを有効にします。

選択内容に応じて、以下のいずれかを実行します。

- サーバー上の既存のコンポーネントを配置するように選択しなかった場合は、この時点で配置を開始することが可能です。ステップ 9 (82 ページ) に進んでください。
- サーバー上の既存のコンポーネントを配置するように選択した場合は、ステップ 8 に進みます。

8. 「次へ」をクリックします。

サーバー上の既存のコンポーネントを配置するように選択した場合は、82 ページの図 22 に示すような「サーバー重複オブジェクト」画面が表示されます。

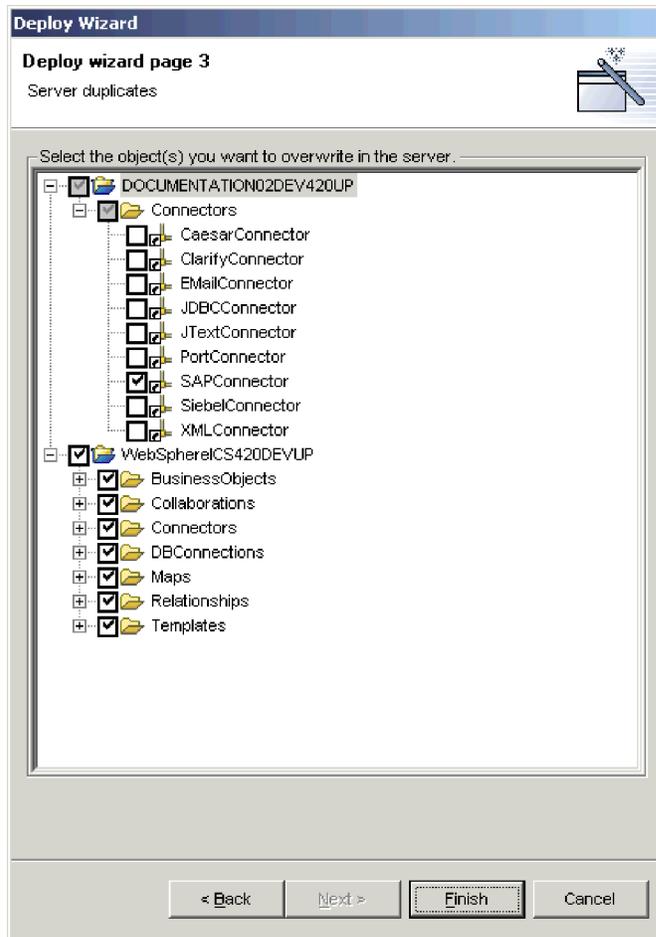


図 22. サーバーで上書きする重複オブジェクトの選択

表示されたユーザー・プロジェクトのフォルダーを展開し、サーバー・リポジトリで上書きする重複コンポーネントのチェック・ボックスを有効にします。

9. 「完了」をクリックします。

System Manager が選択されたコンポーネントをサーバー・リポジトリに展開します。サーバー・ログ出力を参照すると、展開セッションが開始され、コンポーネントがリポジトリに追加されたときに記録されたメッセージが表示されます。

展開セッションが完了すると、System Manager に、展開の正常終了を示す情報プロンプトが表示されるか、展開の失敗を示すエラー・プロンプトが表示されます。

10. プロンプトで表示された情報で重要なものをすべてメモに取ります。例えば、プロンプトによって、一部の配置されたコンポーネントをアクティブにするため、サーバーをリポートするように要求される場合、アクティブなコンポーネントの上書きができないため、配置が正常終了しなかったことが通知される場合もあります。「OK」をクリックして、プロンプトで指定された操作をすべて行います。

ドラッグ・アンド・ドロップを使用するコンポーネントの配置

System Manager ビューで、ドラッグ・アンド・ドロップ操作を使用して、コンポーネントを配置するには、以下の手順を実行します。

1. 配置するコンポーネントがサーバー・リポジトリに存在する場合は、配置前にコンポーネントを停止する必要があります。

パッケージを展開する前に、InterChange Server コンポーネント管理ビューまたは System Monitor を使用し、サーバー内の重複するコンポーネントを停止します。詳細については、85 ページの『リポジトリ内のコンポーネントの状態の管理』を参照してください。

2. WebSphere Business Integration System Manager ビューで、以下の操作を行って、配置するコンポーネントを選択します。
 - ユーザー・プロジェクトまたは統合コンポーネント・ライブラリーを選択して、その中のすべてのコンポーネントを選択します。
 - ユーザー・プロジェクトまたは統合コンポーネント・ライブラリー内のフォルダーを選択して、同じタイプのコンポーネントをすべて選択します。
 - コンポーネントのフォルダーを展開し、コンポーネントを個別に選択して、コンポーネント・フォルダーを選択します。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。
 - **Shift** を押したまま、連続する項目を選択します。
 - **Ctrl** を押したまま、連続しない項目を選択します。
3. InterChange Server コンポーネント管理ビューで、コンポーネントを配置する先の InterChange Server インスタンス上に選択したリソースをドラッグ・アンド・ドロップします。

System Manager が、指定した InterChange Server インスタンスにコンポーネントを配置しようとしています。メッセージとエラーは、コンソール・ビューに表示されます。

注: WebSphere Business Integration System Manager ビューで、重複したコンポーネントを選択した場合は、コンポーネントの配置を試みるときに、System Manager によりエラーが表示されます。選択を絞り込み、固有のコンポーネントのみが含まれるようにし、再度配置を試みます。

InterChange Server リポジトリ内のコンポーネントの処理

InterChange Server コンポーネント管理ビューを使用すると、登録した InterChange Server のリポジトリを管理したり、それらのリポジトリ内のコンポーネントを管理することができます。

リポジトリ内のコンポーネントの検証

39 ページの『InterChange Server のモード』で説明しているように、InterChange Server におけるリポジトリの整合性に関する制限は、始動したモードに応じて異なります。設計モードで稼働しているサーバーではリポジトリの不整合も許容されるため、コンポーネントは使用可能になるごとに追加できますが、実動モードで稼働しているサーバーではコンポーネント間の参照および依存関係がすべて解決さ

れる必要があります。設計モードのサーバーから実動モードのサーバーに移行する場合は、リポジトリを事前に検証し、リポジトリが整合した状態になること確認してください。

System Manager を使用してリポジトリを検証するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、リポジトリを検証するサーバー・インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリを検証」を選択します。
2. リポジトリが整合した状態にある場合は、情報プロンプトが表示されます。

「OK」をクリックして「リポジトリを検証」プロンプトを閉じます。

未解決の依存関係がリポジトリに存在する場合は、「欠落している依存関係」ダイアログが表示されます。これを 図 23 に示します。「完了」をクリックし、「欠落している依存関係」ダイアログを閉じます。

注: 欠落している依存関係をすべて解決しなければ、サーバーを実動モードで始動できません。実動モードについては、41 ページの『実動モード』を参照してください。

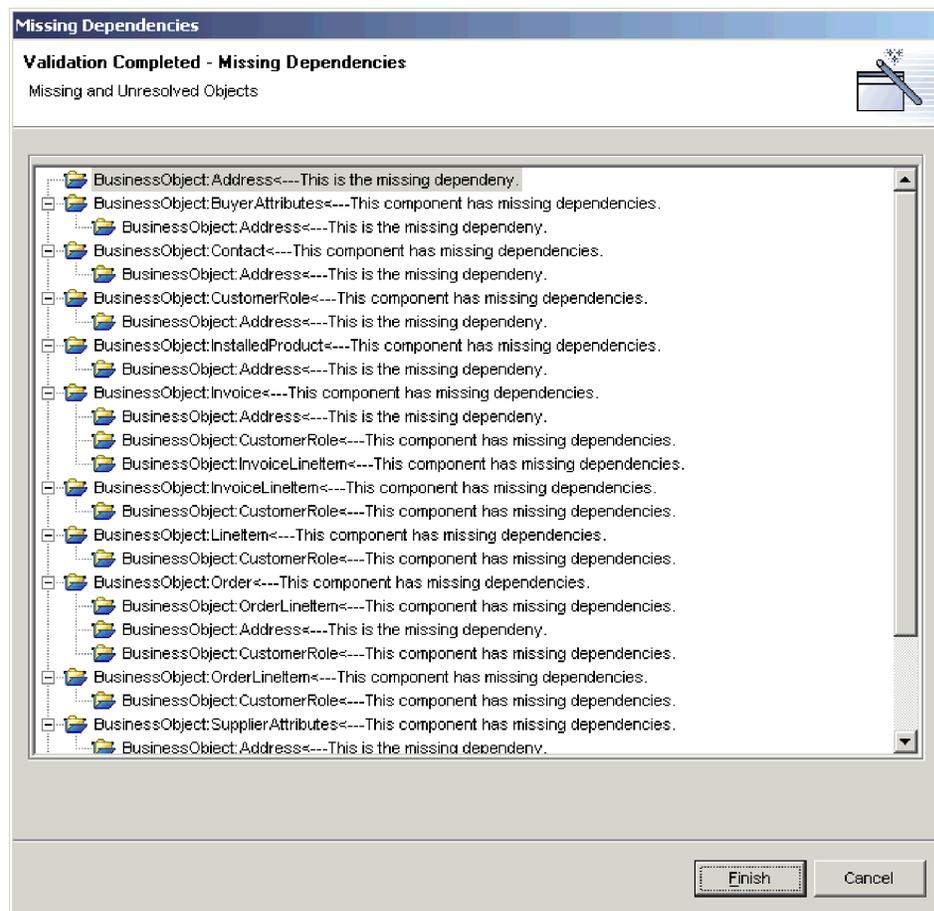


図 23. リポジトリにおける未解決の依存関係

データベース接続プールの検証

データベース接続プールを検証するには、InterChange Server コンポーネント管理ビューでプールを右マウス・ボタンでクリックし、コンテキスト・メニューから「データベース接続を検証」を選択します。

データベース接続プールの詳細については、207 ページの『第 8 章 データベース接続プールの構成』を参照してください。

リポジトリ内のコンポーネントの状態の管理

InterChange Server コンポーネント管理ビューを使用すると、InterChange Server インスタンスのリポジトリ内にあるコンポーネントの状態を管理できます。

インターフェースの開発時には開発するコンポーネントをテストする必要があるため、このビューはとても便利です。頻繁にコンポーネントをテストして、変更が必要なことを認識します。次に、変更したコンポーネントをサーバーに展開して、もう一度テストします。ただし、コンポーネントをサーバーに展開するには非アクティブ状態である必要があるため、展開する前にコンポーネントを停止します。インターフェースの開発、展開、およびテストをすべてワークベンチ内で行うことができるので、同様にワークベンチからコンポーネントの状態を管理できることはとても便利です。

InterChange Server コンポーネント管理ビューでコンポーネントの状態を変更するには、ナビゲーション・ツリーのコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから適切な状態を選択します。86 ページの表 6 に、ランタイム状態の各コンポーネント・タイプに適用できる状態操作をリストします。コンポーネントが異なる状態である場合のシステムの振る舞いについては、「システム管理ガイド」を参照してください。

表 6. System Manager におけるコンポーネントに選択可能な状態操作

コンポーネント・タイプ	状態操作
コラボレーション・オブジェクト	<ul style="list-style-type: none"> • 開始 • 停止 • 一時停止 • シャットダウン
コネクター	<ul style="list-style-type: none"> • 開始 • 停止 • 一時停止 • シャットダウン • ブート
マップ	<ul style="list-style-type: none"> • 開始 • 停止
関係	<ul style="list-style-type: none"> • 開始 • 停止

また、複数のコンポーネントを選択し、その中の 1 つを右マウス・ボタンでクリックし、状態変更を一度にすべてのコンポーネントに適用することもできます。この操作は、各コンポーネントに対して、状態管理操作を別々に行う必要がなくなるため、複数のコンポーネントを同時に配置する必要のある場合に非常に有効です。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。

- **Shift** を押したまま、連続する項目を選択します。
- **Ctrl** を押したまま、連続しない項目を選択します。

注: 選択したコンポーネントのグループに対して、状態管理操作を実行しようとしたときに、状態が選択されていないコンポーネントが存在する場合は、コンテキスト・センシティブ・メニューには、状態管理オプションは表示されません。例えば、複数のコネクターを停止しようとしていて、間違えて、ビジネス・オブジェクト定義も選択してしまった場合、状態管理操作は、コンテキスト・センシティブ・メニューには公開されません。

特定のタイプのコンポーネントすべての状態を変更するには、InterChange Server コンポーネント管理ビューで、そのコンポーネント・タイプのフォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべて開始」または「すべて停止」を選択します。

製品インストール内のコンポーネント管理には、System Monitor の使用をお勧めします。System Monitor は、高度な柔軟性を備えた専用の管理ツールです。System Monitor の使用については、「システム管理ガイド」を参照してください。

リポジトリ内のコンポーネント・プロパティの変更

マップ、コネクタ、およびコラボレーション・オブジェクトには、ランタイム・プロパティがあります。これらのプロパティを変更すると、コンポーネントの振る舞いを変更できます。以下のセクションの説明に従って、特定のコンポーネントのプロパティを変更します。

- 『コラボレーション・オブジェクトのプロパティの変更』
- 88 ページの『マップのプロパティの変更』
- 89 ページの『コネクタのプロパティの変更』

コラボレーション・オブジェクトのプロパティの変更

コラボレーション・オブジェクトのプロパティを変更するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、そのプロパティを変更するコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティ」を選択します。
2. 「コラボレーションの一般プロパティ」と「プロパティ」の両方のタブで、プロパティに目的の値を設定します。

コラボレーション・オブジェクト・プロパティについては、219 ページの『第9章 コラボレーション・オブジェクトの構成』を参照してください。

88 ページの図 24 に、コラボレーション・オブジェクトの「プロパティ」ダイアログを示します。

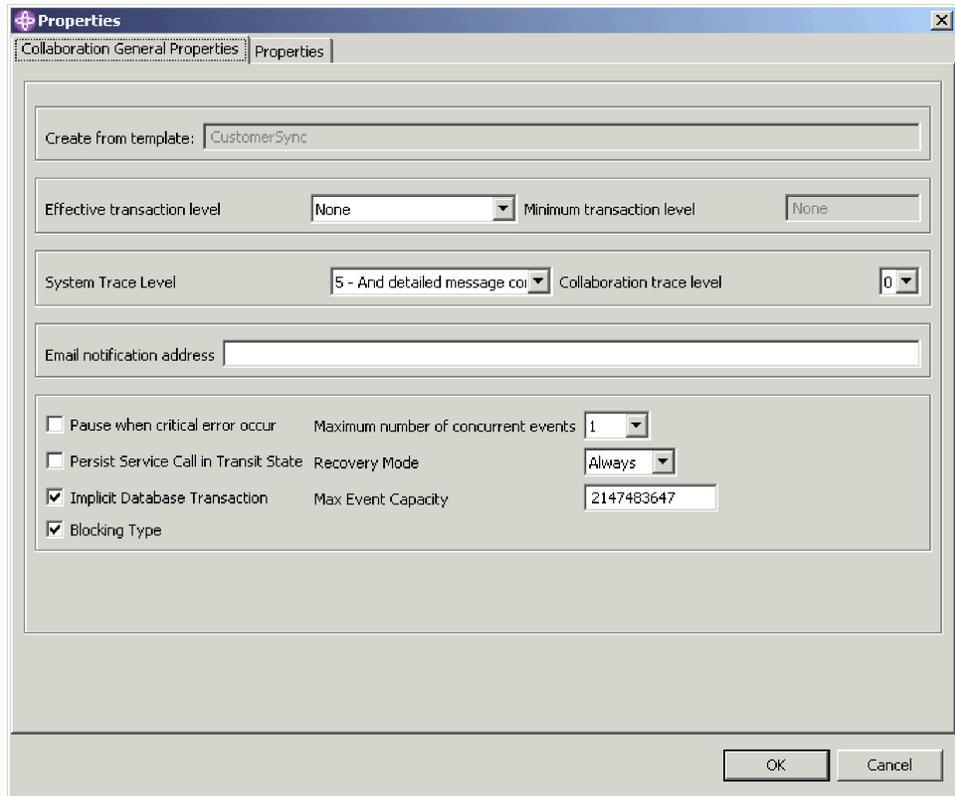


図 24. コラボレーション・オブジェクトのプロパティーの変更

3. 「OK」をクリックします。

マップのプロパティーの変更

マップのプロパティーを変更するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、そのプロパティーを変更するマップを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティー」を選択します。
2. 「マップ・プロパティー・ページ」ダイアログで、プロパティーに目的の値を設定します。

マップ・プロパティーの詳細については、「マップ開発ガイド」を参照してください。

89 ページの図 25 に、「マップ・プロパティー・ページ」ダイアログを示します。

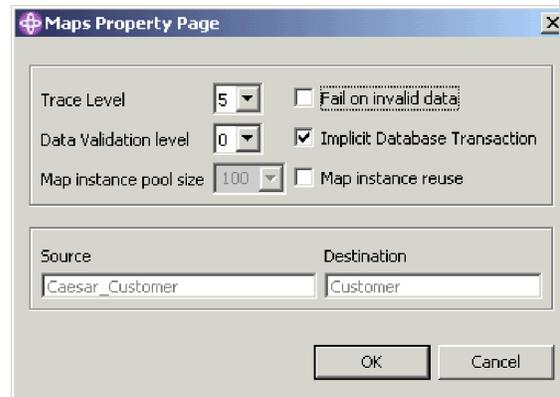


図 25. マップのプロパティーの変更

3. 「OK」をクリックします。

コネクターのプロパティーの変更

コネクターのプロパティーを変更するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、そのプロパティーを変更するコネクターを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティー」を選択します。
2. 「コネクター標準プロパティー」、「関連付けられたマップ」、および「リソース」の各タブでプロパティーに目的の値を設定します。コネクター・プロパティーの詳細については、159 ページの『第 7 章 コネクターの構成』を参照してください。

90 ページの図 26 に、コネクターの「プロパティー」ダイアログを示します。

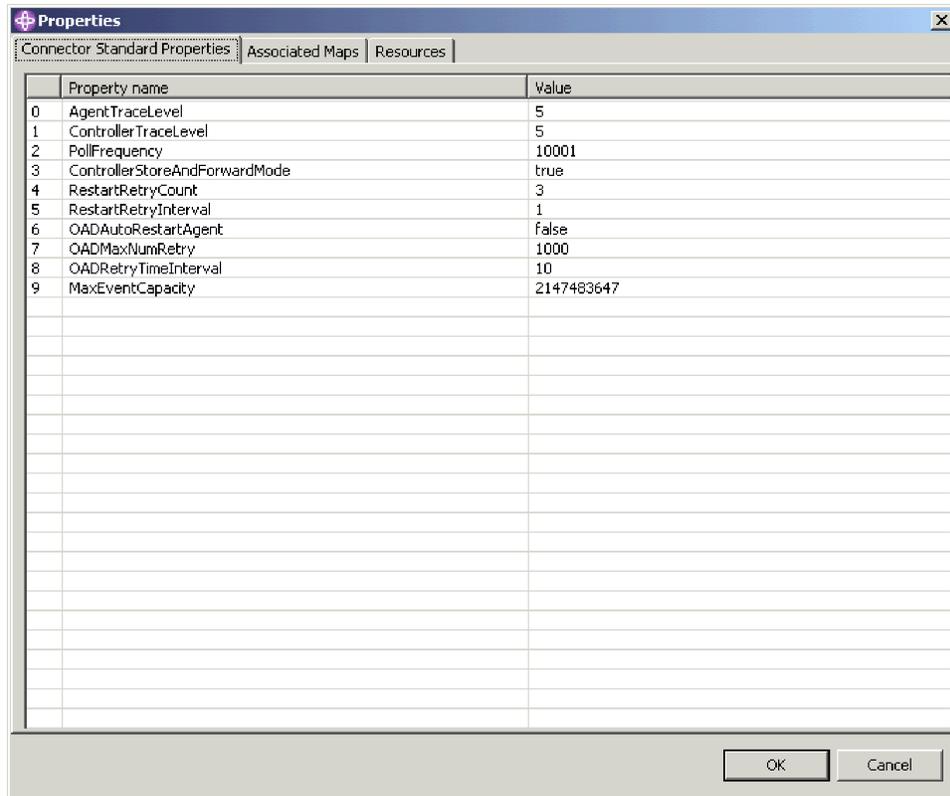


図 26. コネクターのプロパティの変更

3. 「OK」をクリックします。

リポジトリ内のコンポーネントのコンパイル

コラボレーション・テンプレートは、それに基づくコラボレーション・オブジェクトが動作するためには、コンパイルする必要があります。マップも動作するためにはコンパイルする必要があります。設計モードで動作する InterChange Server には、コンパイルすることなく、コラボレーション・テンプレートとマップを展開できますが、テストする場合には必ずコンパイルする必要があります。

InterChange Server で、コラボレーション・テンプレートをコンパイルするには、InterChange Server コンポーネント管理ビューで、コラボレーション・テンプレートを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンパイル」を選択します。

InterChange Server で、すべてのコラボレーション・テンプレートをコンパイルするには、InterChange Server コンポーネント管理ビューで、「コラボレーション・テンプレート」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべてコンパイル」を選択します。

InterChange Server で、マップをコンパイルするには、InterChange Server コンポーネント管理ビューで、マップを右マウス・ボタンでクリックし、コンテキスト・メニューから「コンパイル」を選択します。

InterChange Server で、すべてのマップをコンパイルするには、InterChange Server コンポーネント管理ビューで、「マップ」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべてコンパイル」を選択します。

マップとそのすべてのサブマップをコンパイルするには、InterChange Server コンポーネント管理ビューでマップを右マウス・ボタンでクリックし、コンテキスト・メニューから「サブマップでコンパイル」を選択します。

「コンソール」ビューに、コンパイルの成功または失敗を示すメッセージが表示されます。

リポジトリからのコンポーネントの削除

ビジネス・インテグレーション・システムを開発するときには、システム的设计は頻繁に変化します。一般に、新しい要件や環境の違いによってインターフェースの再設計が必要になります。現在の設計に関するビジネス上の問題を解決する必要があるコンポーネントのみがリポジトリに含まれることを確認する必要があります。この結果、変更された設計に属するコンポーネントを削除する必要がある場合もあります。

リポジトリ全体の削除

System Manager を使用してサーバー・リポジトリのコンポーネントをすべて削除するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、リポジトリを削除するサーバー・インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「リポジトリを削除」を選択します。
2. 「リポジトリの削除の確認」プロンプトで「OK」をクリックします。
3. 次の「リポジトリの削除の確認」プロンプトで「OK」をクリックします。

コンポーネント・ブラウザーを使用したコンポーネントの削除

コンポーネント・ブラウザーを使用して InterChange Server リポジトリからコンポーネントを削除するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、ブラウザー内で削除するコンポーネントを選択します。

Windows での標準技法を使用して複数の項目を選択できます。例えば、連続する複数の項目を選択するには **Shift** を、連続しない複数の項目を選択するには **Ctrl** を押します。

2. コンポーネントを削除するには、まずそれが非アクティブ状態であるか確認します。コンポーネントの削除を試行する前に、それらを停止して確実に非アクティブ状態にします。

System Manager で InterChange Server コンポーネント管理ビューを使用してコンポーネントを停止する場合は、85 ページの『リポジトリ内のコンポーネントの状態の管理』を参照してください。

System Monitor を使用してコンポーネントを停止する場合は、「システム管理ガイド」を参照してください。

3. 以下のいずれかを実行して、選択したコンポーネントを削除します。

- 「削除」を押します
- 選択したコンポーネントのいずれかを右マウス・ボタンでクリックし、コンテキスト・メニューから「削除」を選択します。
- メニュー・バーから、「編集」>「削除」を選択します。

図 27 に、コンポーネント・ブラウザを使用してコンポーネントを削除する場合の System Manager を示します。

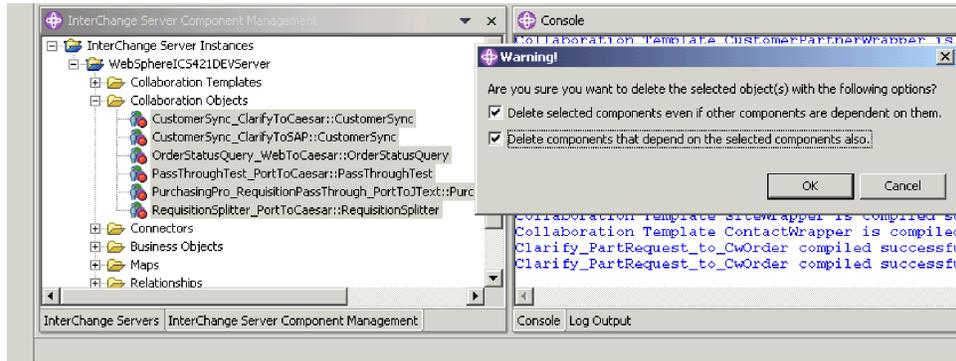


図 27. コンポーネント・ブラウザを使用したコンポーネントの削除

4. ダイアログが表示されたら、以下の手順を実行します。
 - 他のコンポーネントから参照されている場合でもコンポーネントを削除するには、「他のコンポーネントが依存していても、選択されたコンポーネントを削除します。」チェック・ボックスを有効にします。

例えば、ビジネス・オブジェクトが他のビジネス・オブジェクトを含む場合は、それらのビジネス・オブジェクトに依存関係があるため、このオプションを有効にしない限り、従属コンポーネントの削除を選択してもビジネス・オブジェクトは削除できません。

 - 選択したコンポーネントの依存関係も削除するには、「選択されたコンポーネントに依存するコンポーネントも削除します。」チェック・ボックスを有効にします。依存関係については、94 ページの『依存関係および参照』を参照してください。
5. 「完了」をクリックします。

削除操作が正常に終了すると、ウィザードは終了します。

削除操作に失敗した場合は、「OK」をクリックしてプロンプトを閉じ、問題を解決してから操作を再試行します。プロンプトのエラー情報を使用してトラブルシューティングを行うことが困難な場合は、サーバーのログ出力を使用して失敗の原因を判別してください。

サーバー・オブジェクトの削除ウィザードを使用したコンポーネントの削除

サーバー・オブジェクトの削除ウィザードを使用して InterChange Server リポジトリからコンポーネントを削除するのに System Manager を使用するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、リポジトリからコンポーネントを削除するサーバー・インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「サーバー・オブジェクトの削除」を選択します。
2. コンポーネントを削除するには、まずそれが非アクティブ状態であるか確認します。コンポーネントの削除を試行する前に、それらを停止して確実に非アクティブ状態にします。

System Manager で InterChange Server コンポーネント管理ビューを使用してコンポーネントを停止する場合は、85 ページの『リポジトリ内のコンポーネントの状態の管理』を参照してください。

System Monitor を使用してコンポーネントを停止する場合は、「システム管理ガイド」を参照してください。

3. 以下の操作を実行し、削除するコンポーネントを選択します。
 - サーバー・インスタンスのチェック・ボックスを有効にし、リポジトリのコンポーネントをすべて削除します。
 - サーバー・インスタンスのフォルダーを展開し、特定のコンポーネント・タイプのチェック・ボックスを有効にし、そのタイプのオブジェクトをすべて削除します。
 - 特定のコンポーネント・タイプのフォルダーを選択し、リポジトリのそのタイプのコンポーネントのリストを右側のペインに表示させ、個々のコンポーネントのチェック・ボックスを有効にします。
4. 他のコンポーネントから参照されている場合でもコンポーネントを削除するには、「他のオブジェクトがこのオブジェクトに依存している場合でも強制的に削除されます」チェック・ボックスを有効にします。

例えば、ビジネス・オブジェクトが他のビジネス・オブジェクトを含む場合は、それらのビジネス・オブジェクトに依存関係があるため、このオプションを有効にしない限り、従属コンポーネントの削除を選択してもビジネス・オブジェクトは削除できません。

5. 選択したコンポーネントの依存関係も削除するには、「従属オブジェクトも削除」チェック・ボックスを有効にします。依存関係については、94 ページの『依存関係および参照』を参照してください。

94 ページの図 28 に、「サーバー・オブジェクト削除」ウィザードを示します。

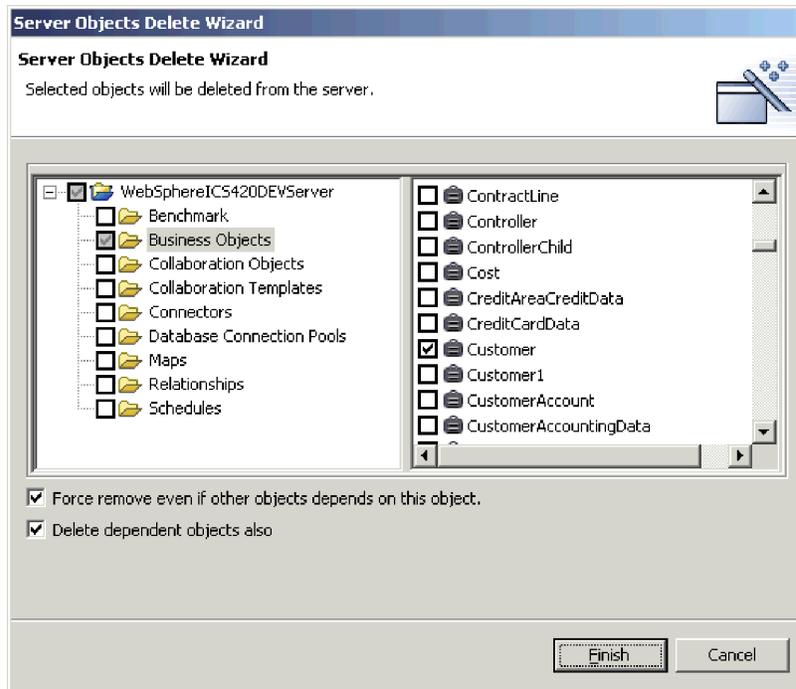


図 28. サーバーからのオブジェクトの削除

6. 「完了」をクリックします。

削除操作が正常に終了すると、ウィザードは終了します。

削除操作に失敗した場合は、「OK」をクリックしてプロンプトを閉じ、問題を解決してから操作を再試行します。プロンプトのエラー情報を使用してトラブルシューティングを行うことが困難な場合は、サーバーのログ出力を使用して失敗の原因を判別してください。

依存関係および参照

統合コンポーネントは相互に依存し合ってビジネス・インテグレーション・システムでの役割を果たします。例えば、ビジネス・オブジェクト定義は、他のビジネス・オブジェクト定義を子として含むことができ、コラボレーション・テンプレートは、そのポートと関連付けられているビジネス・オブジェクト定義を持ち、コネクタは、そのサポートしているビジネス・オブジェクトと関連付けられているマップを持ちます。システムが正常に機能するためには、これらの依存関係が満たされている必要があり、実動モードで起動されると、InterChange Server は、すべての依存関係と参照が解決されていることを確認します。未解決の依存関係または参照を検出すると、InterChange Server は実動モードでは始動しません。

コンポーネント間の関係について説明するときには、状況に応じて**依存関係**および**参照**という用語を使用します。例えば、コネクタ定義が InterChange Server とデータを交換するには、サポートするビジネス・オブジェクト定義およびそれらのビジネス・オブジェクト定義に関連付けられたマップが存在する必要があります。この場合には、ビジネス・オブジェクト定義およびマップはコネクタ定義の**依存関係**です。ビジネス・オブジェクト定義とコネクタ定義の間に同じ関係がある場合でも、ビジネス・オブジェクトの文脈では、コネクタはビジネス・オブジ

エクト定義の参照の 1 つであり、そのビジネス・オブジェクトを子として含むほかのビジネス・オブジェクト、そのビジネス・オブジェクトを変換するマップ、ポート定義用にビジネス・オブジェクトをサポートするコラボレーション・テンプレートなどを持ちます。

表 7 に、依存関係および参照となるコンポーネントをコンポーネント・タイプごとに示します。

表 7. 統合コンポーネントの依存関係および参照

コンポーネント	依存関係	参照
ビジネス・オブジェクト	<ul style="list-style-type: none"> ビジネス・オブジェクト 	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ コネクタ コラボレーション・テンプレート コラボレーション・オブジェクト
関係	<ul style="list-style-type: none"> ビジネス・オブジェクト 	なし
コネクタ	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ 	<ul style="list-style-type: none"> コラボレーション・オブジェクト
マップ	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ 	<ul style="list-style-type: none"> コネクタ コラボレーション・オブジェクト
コラボレーション・テンプレート	<ul style="list-style-type: none"> ビジネス・オブジェクト 	なし
コラボレーション・オブジェクト	<ul style="list-style-type: none"> ビジネス・オブジェクト マップ コネクタ コラボレーション・テンプレート コラボレーション・オブジェクト 	<ul style="list-style-type: none"> コラボレーション・オブジェクト

依存関係および参照の表示

System Manager を使用すると、統合コンポーネントの依存関係および参照を表示させることができます。

コンポーネントの依存関係を表示させるには、System Manager でコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**依存関係を表示**」を選択します。「依存関係ツリー」ウィザードが表示されます。詳細については、49 ページの『依存関係ツリーの使用』を参照してください。

コンポーネントの参照を表示させるには、System Manager でコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**参照の表示**」を選択します。「オブジェクト参照」ウィンドウが表示されます。

システムで検出できない依存関係

依存関係および参照によっては、システムで自動的に検出および実行できないものがあります。表 8 に、システムが依存関係および参照を判別できないコンポーネントを示します。

システムでこれらの依存関係および参照を検出することはできませんが、システムを正常に動作させるには、これらが満たされている必要があります。必ずプロジェクト・ドキュメンテーションを正確に保ち、環境から環境にインターフェースを展開するときにはこれらの依存関係も含めてください。

表 8. システムが依存関係および参照を判別できないコンポーネント

コンポーネント	説明
関係	マップおよびコラボレーション・テンプレートは関係を使用できますが、これは Java コードの API から行います。システムは、すべてのマップおよびコラボレーション・テンプレートのすべてのコードを構文解析して関係の使用を検出することができません。
データベース接続プール	マップおよびコラボレーション・テンプレートはデータベース接続プールを使用できますが、これは Java コードの API から行います。システムは、すべてのマップおよびコラボレーション・テンプレートのすべてのコードを構文解析してデータベース接続プールの使用を検出することができません。
データ・ハンドラー	コネクタも、他のコンポーネントと同様にデータ・ハンドラーを使用してデータをフォーマットできます。データ・ハンドラーはシステムによって一切管理されないため、検出できません。
コード内のみで使用されるコンポーネント	ビジネス・オブジェクト定義をソースまたは宛先としてマップにドラッグ・アンド・ドロップした場合は、システムはマップがビジネス・オブジェクトに依存するかどうかを判別できますが、カスタム Java コードでのみ行われるコンポーネントの使用は検出できません。新規ビジネス・オブジェクトをインスタンス化するか、マップまたはコラボレーションで手動でマップを呼び出す場合は、システムはその依存関係を検出できません。
ベンチマーク	現在、システムはベンチマークの依存関係および参照の検出をサポートしていません。
スケジュール	現在、システムはスケジュールの依存関係および参照の検出をサポートしていません。

複数のワークベンチ・リソースで使用可能な標準の操作

ワークベンチで行う多くの作業は処理している特定のリソースまたは作業している状況に依存しますが、すべてのリソースに同じように影響を及ぼす多くの操作も用意されています。このセクションでは、ワークベンチで実行可能な作業のうち、すべてのリソースに対して同様に機能するものについて説明します。

ソース・コード制御からワークスペースへのプロジェクトの追加

Rational ClearCase などのソース・コード制御システムから統合コンポーネント・ライブラリーおよびユーザー・プロジェクトをワークスペースに追加できます。

追加方法については、ソース・コード制御システム・プラグインの資料を参照してください。

ClearCase を使用して行う例については、24 ページの『ワークスペースへのプロジェクトの追加』を参照してください。

リソースの切り取り、コピー、および貼り付け

リソースの切り取り、コピー、および貼り付けは、System Manager とファイル・システムの両方で可能です。

System Manager でユーザー・プロジェクト、統合コンポーネント・ライブラリー、統合コンポーネント、ショートカット、またはフォルダーの切り取り、コピー、または貼り付けを行うには、リソースを右マウス・ボタンでクリックし、所定のメニュー項目を選択します。統合コンポーネントをコピーする場合は、同じライブラリーに貼り付ける (例えば、ビジネス・オブジェクト定義をコピーして同じライブラリーに貼り付け、別の名前を指定してテンプレートとして使用する) ことはできません。ただし、Designer ツールでコンポーネント定義を開き、「別名保管」操作を実行して新しい名前と同じライブラリーに保管することはできます。

ファイル・システムで統合コンポーネントまたはショートカットの切り取り、コピー、および貼り付けを行うには、Windows のエクスプローラーを起動し、プロジェクト・ディレクトリーの該当するサブディレクトリーにナビゲートし、コンポーネントの名前を共用するファイルをコピーし、宛先プロジェクト・ディレクトリーを含む適切なサブディレクトリーに貼り付けます。

ファイル・システムで切り取り、コピー、および貼り付けを行う場合は、必ず System Manager で統合コンポーネント・ライブラリーまたはユーザー・プロジェクトを最新表示させ、新しく追加したリソースが表示されるようにしてください。詳細については、『リソースの最新表示』を参照してください。

存在するユーザー・プロジェクトおよびライブラリーを指定するメタデータ参照がワークベンチで管理されているため、ユーザー・プロジェクト全体または統合コンポーネント・ライブラリー全体の切り取り、コピー、および貼り付けを行うことはできません。ワークスペース・ディレクトリーにフォルダーをコピーしても、メタデータ参照は更新されません。ただし、System Manager でメタデータ・エントリーを満たすように新規ユーザー・プロジェクトまたはライブラリーを作成してから、新規ライブラリーまたはユーザー・プロジェクトに対応するディレクトリーにコンポーネント定義のフォルダーを貼り付けることはできます。

リソースの最新表示

ファイル・システムでの貼り付けの切り取り、コピー、および貼り付けによってコンポーネント定義をライブラリーに追加したりショートカットをユーザー・プロジェクトに追加したりする場合は (64 ページの『統合コンポーネント・ライブラリーにおけるマップとコラボレーション・オブジェクト・プロパティーの変更』を参照)、必ず System Manager でライブラリーを最新表示させ、変更内容を反映させてください。

統合コンポーネント・ライブラリーまたはユーザー・プロジェクトを最新表示させるには、System Manager で右マウス・ボタンをクリックし、コンテキスト・メニューから「最新表示」を選択します。

リソースの削除

ワークベンチ・リソースを削除するには、以下の手順を実行します。

1. System Manager でリソースを右マウス・ボタンをクリックしてコンテキスト・メニューから「削除」を選択するか、System Manager でリソースを選択して **Delete** キーを押します。
2. 「コンポーネントの削除」ダイアログが表示された場合は、「**OK**」をクリックします。

注: コンポーネントに依存関係が存在する場合は削除できません。

Eclipse ベースのワークベンチの使用

このセクションでは、ツールを効率的に使用するためにツール・フレームワークで実行する一部の操作について説明します。

パースペクティブのオープンおよびクローズ

このセクションでは、パースペクティブのオープン方法およびクローズ方法について説明します。

パースペクティブのオープン

ワークベンチでパースペクティブを開くには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「**ウィンドウ**」>「**パースペクティブを開く**」>「**その他**」を選択します。
2. 「パースペクティブの選択」ダイアログで開くパースペクティブを選択し、「**OK**」をクリックします。

WebSphere InterChange Server 統合コンポーネントを処理するために使用する、「System Manager」、「統合テスト環境」、および「コラボレーション・デバッガー」の各パースペクティブに加えて、以下のセクションで説明するパースペクティブを調べることが必要になる場合があります。

「**リソース**」パースペクティブ: このパースペクティブにより、統合コンポーネント・ライブラリーなどのプロジェクト・データを表すメタデータ・ファイルを直接処理できます。99 ページの図 29 に、「リソース」パースペクティブが開かれた状態を示します。このパースペクティブでは、ContactSync コラボレーション・テンプレートを表す .cwt ファイルをナビゲーター・ビューから開き、その内容をエディター・ビューに示しています。

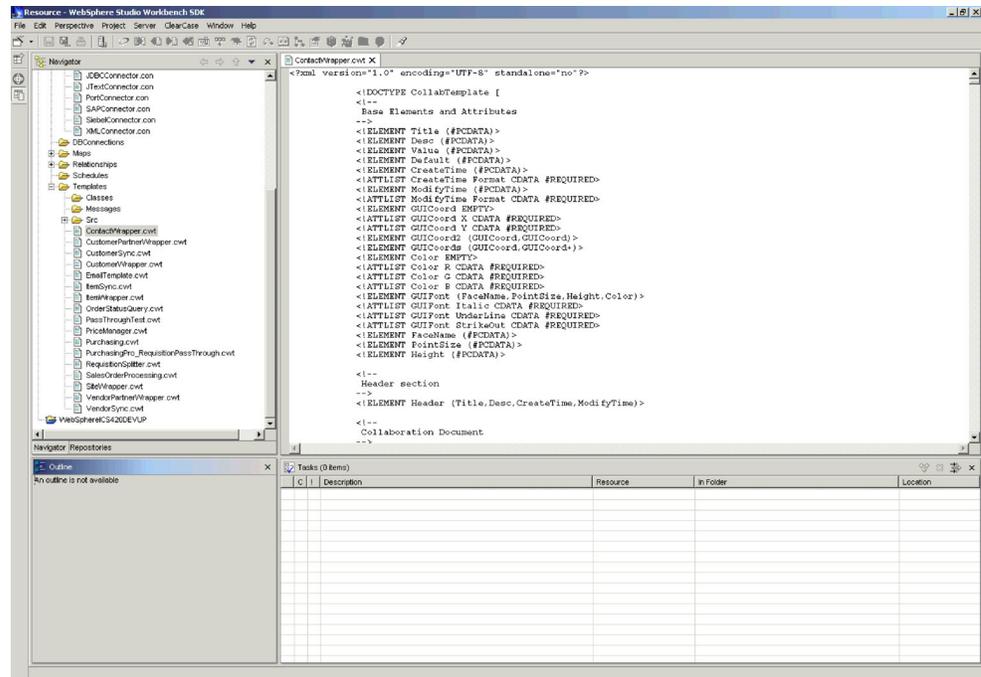


図 29. 「リソース」 パースペクティブ

要確認: 「リソース」 パースペクティブから公開されるメタデータ・ファイルは、WebSphere InterChange Server 統合コンポーネントを定義します。System Manager パースペクティブには、これらのコンポーネントをインターフェース経由で安全に処理するための方法が用意されています。直接メタデータ・ファイルを操作すると、コンポーネント定義を破壊してしまう危険があります。コンポーネント・メタデータ・ファイルを操作するのは、構造を熟知している場合、または定義のトラブルシューティングを行うためにテクニカル・サポートに連絡し、指示があった場合に限定してください。

Java パースペクティブ: このパースペクティブには、Java ファイルのオーサリングを支援するエディターやビューが備わっています。Java プログラミングは、ほとんどの場合、デザイナー・ツールで実行しますが、データ・ハンドラーなどの外部コンポーネントやユーティリティ・クラスを作成することが必要になる場合があります。そのような場合、この Java パースペクティブが、非常に有用な場合があります。図 29 に、Java パースペクティブを示します。

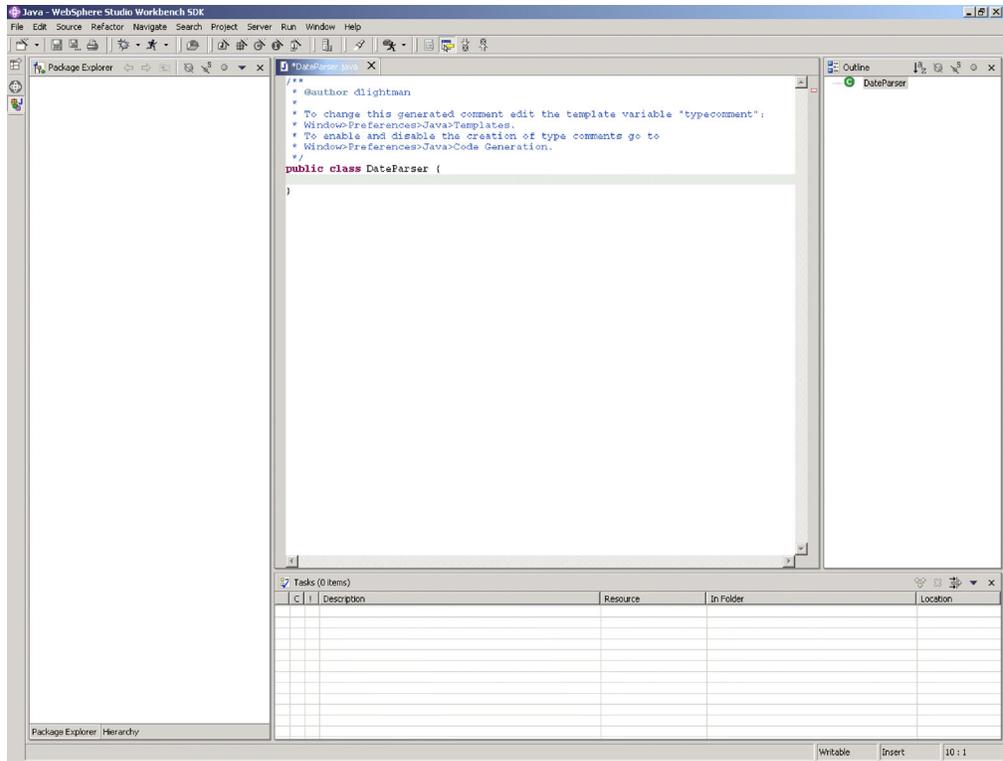


図 30. Java パースペクティブ

パースペクティブのクローズ

パースペクティブを閉じるには、以下の操作を実行します。

- ・ 「ウィンドウ」 > 「パースペクティブを閉じる」を選択し、現在アクティブなパースペクティブを閉じます。
- ・ 「ウィンドウ」 > 「すべてのパースペクティブを閉じる」を選択し、現在開かれているパースペクティブをすべて閉じます。
- ・ パースペクティブ・ショートカット・バーにあるパースペクティブのアイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「閉じる」を選択し、そのパースペクティブを閉じます。
- ・ パースペクティブ・ショートカット・バーにあるパースペクティブのアイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべて閉じる」を選択し、開かれているパースペクティブをすべて閉じます。

表示およびクローズ

WebSphere WorkBench パースペクティブおよび WebSphere Studio Application Developer Integration Edition パースペクティブに表示されるペインを制御できます。

ビューを表示する

ビューを表示するには、以下の手順を実行します。

1. 「ウィンドウ」 > 「ビューの表示」 > 「その他」を選択します。

2. **ICS Control View** などのビュー・グループのフォルダーを展開します。
3. **InterChange Server コンポーネント管理**ビューなどの特定のビューを選択します。
4. 「**OK**」をクリックします。

ビューをクローズする

ビューをクローズするには、以下のいずれかの操作を実行します。

- ビューのタイトル・バーを右マウス・ボタンでクリックし、コンテキスト・メニューから「**閉じる**」を選択します。
- ビューのタイトル・バーにある「**閉じる**」ボタンをクリックします。

パースペクティブのカスタマイズ

パースペクティブをカスタマイズし、必要なパースペクティブ、ビュー、ウィザード、およびプラグイン・インターフェースを組み込むことにより、頻繁に使用するエレメントを開いたり不要なエレメントを閉じたりする回数を最小限に抑えることができます。現在アクティブなパースペクティブをカスタマイズするには、以下の手順を実行します。

1. メニュー・バーから「**ウィンドウ**」>「**パースペクティブのカスタマイズ**」を選択します。
2. カスタマイズするノードをクリックして展開します。
3. ノード・エレメントのチェック・ボックスを有効または無効にします。表 9 に、カスタマイズ可能なパースペクティブ・ノードおよびそのノードを有効にした場合の効果を示します。

表 9. カスタマイズ可能なパースペクティブ・ノード

パースペクティブ・ノードのカスタマイズ	結果
「 ファイル 」>「 新規 」	「 ファイル 」>「 新規 」メニューから項目を追加または除去します。
「 ウィンドウ 」>「 パースペクティブを開く 」	「 ウィンドウ 」>「 パースペクティブを開く 」メニューからパースペクティブを追加または除去します。
「 ウィンドウ 」>「 ビューの表示 」	「 パースペクティブ 」>「 ビューの表示 」メニューからビューを追加または除去します。
その他	メニュー・バーおよびツールバーから追加または除去します。例えば、「 ClearCase 」メニューを表示させるには、「 ClearCase 」チェック・ボックスを有効にする必要があります。

パースペクティブの保管

パースペクティブ構成を保管し、カスタマイズ内容を保存できます。パースペクティブを保管するには、以下の手順を実行します。

1. ツール・フレームワークのメニュー・バーから「**ウィンドウ**」>「**パースペクティブの別名保管**」を選択します。
2. 「**名前**」フィールドにパースペクティブの名前を入力します。
3. 「**OK**」をクリックします。

デフォルトでのパースペクティブの設定

デフォルトでは、ツール・フレームワークは「リソース」パースペクティブを開きます。主にツール・フレームワークを使用して統合コンポーネントを処理する場合は、いずれかの IBM WebSphere InterChange Server パースペクティブをデフォルトにすることができます。これを行うには、以下の手順を実行します。

1. ツール・フレームワークのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「ワークベンチ」ノードを展開します。
3. 「ワークベンチ」ノードの下にある「パースペクティブ」ノードを選択します。
4. 「使用可能なパースペクティブ (Available Perspectives)」リストから必要なパースペクティブ (System Manager など) を選択します。
5. 「デフォルトを作成 (Make Default)」をクリックします。
6. 「OK」をクリックします。

System Manager の設定の構成

System Manager の設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「System Manager 設定」を選択し、以下の操作を実行し使用可能な設定オプションを構成します。
 - 統合コンポーネント・ライブラリーからコンポーネントを削除するときにプロンプトが表示されないようにするには、「オブジェクトの削除について確認しない」ペインでコンポーネント・タイプに対応するチェック・ボックスを有効にします。

注: 「未解決のフロー」チェック・ボックスを有効にしても、Flow Manager からフローを削除するときにプロンプトが表示されるかどうかには影響しません。Flow Manager およびその設定の構成方法については、「システム管理ガイド」を参照してください。

- コピー操作時にコンポーネントとともにコンポーネントの依存関係もコピーする場合は、「ディープ・コピー」チェック・ボックスを有効にします。

「ディープ・コピー」を使用可能にすると、ライブラリー間でビジネス・オブジェクト定義をコピーする場合、例えば、定義に含まれている子ビジネス・オブジェクトもすべてコピーされます。しかし、「ディープ・コピー」を無効にすると、ライブラリー間でビジネス・オブジェクト定義をコピーする場合に、そのビジネス・オブジェクト自体しかコピーされません。

依存関係については、94 ページの『依存関係および参照』を参照してください。

- ファイルの名前およびパスを「ログ・ファイル」フィールドに入力するか、「参照」をクリックしてファイルを選択します。System Manager でエラーが発生した場合は、指定のファイルにエラー情報が書き込まれます。メガバイト単位でログ・ファイルの最大サイズを指定するには、「最大サイズ」フィールドに数値を入力します。
- 設定エレメントをデフォルト値に設定するには、「デフォルトの復元」をクリックします。

図 31 に、System Manager の設定インターフェースを示します。

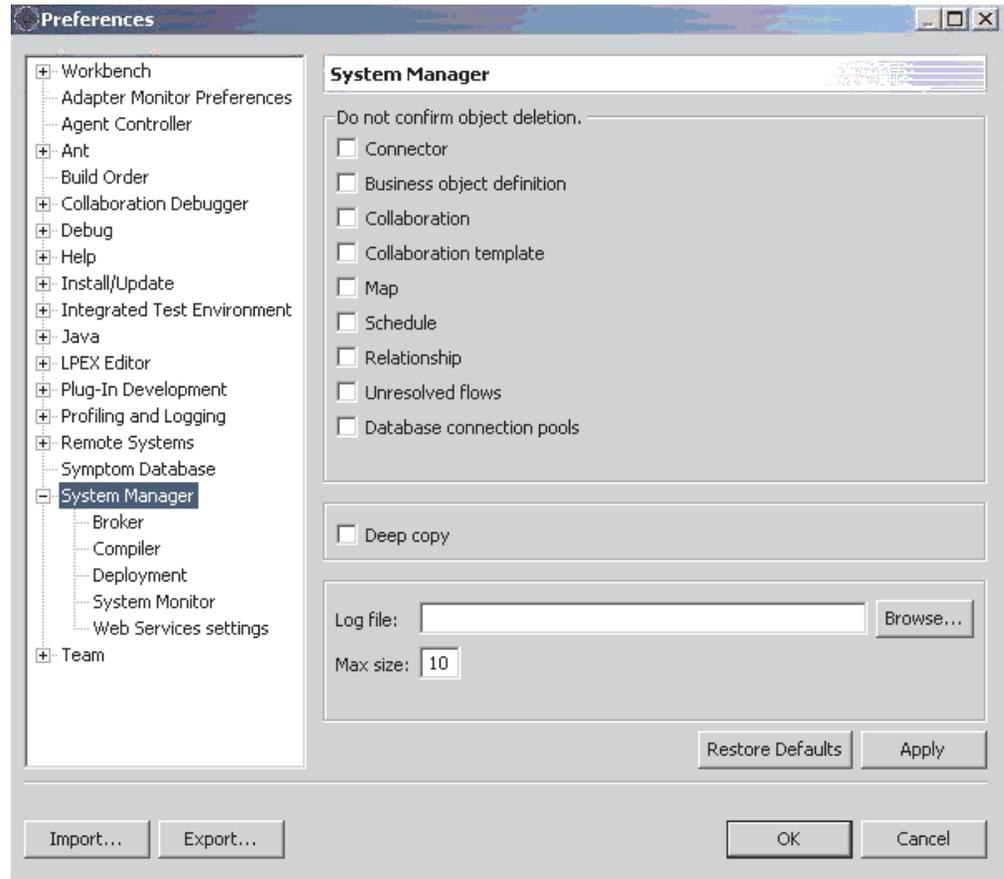


図 31. System Manager の設定

3. 「ブローカー設定」インターフェースを使用すると、サポートされる WebSphere Message Broker と連動するように System Manager を構成できます。

ブローカー設定をセットするには、以下の手順を実行します。

- a. 「System Manager 設定」を展開して、「ブローカー設定」を選択します。
- b. 「Integrator Broker のインポーター・パスを指定」フィールドに、WebSphere MQ Integrator Broker インポーター・ユーティリティの絶対パスを入力するか、「参照」をクリックして、ディレクトリーを選択します。
- c. 「Message Broker のインポーター・パスを指定」フィールドに、WebSphere BI Message Broker インポーター・ユーティリティの絶対パスを入力するか、「参照」をクリックして、ディレクトリーを選択します。
- d. 「Message Broker のワークスペース・ディレクトリーを指定」フィールドに、WebSphere BI Message Broker の Eclipse ワークスペース・ディレクトリーの絶対パスを入力するか、「参照」をクリックして、ディレクトリーを選択します。

104 ページの図 32 に、「ブローカー設定」インターフェースを示します。

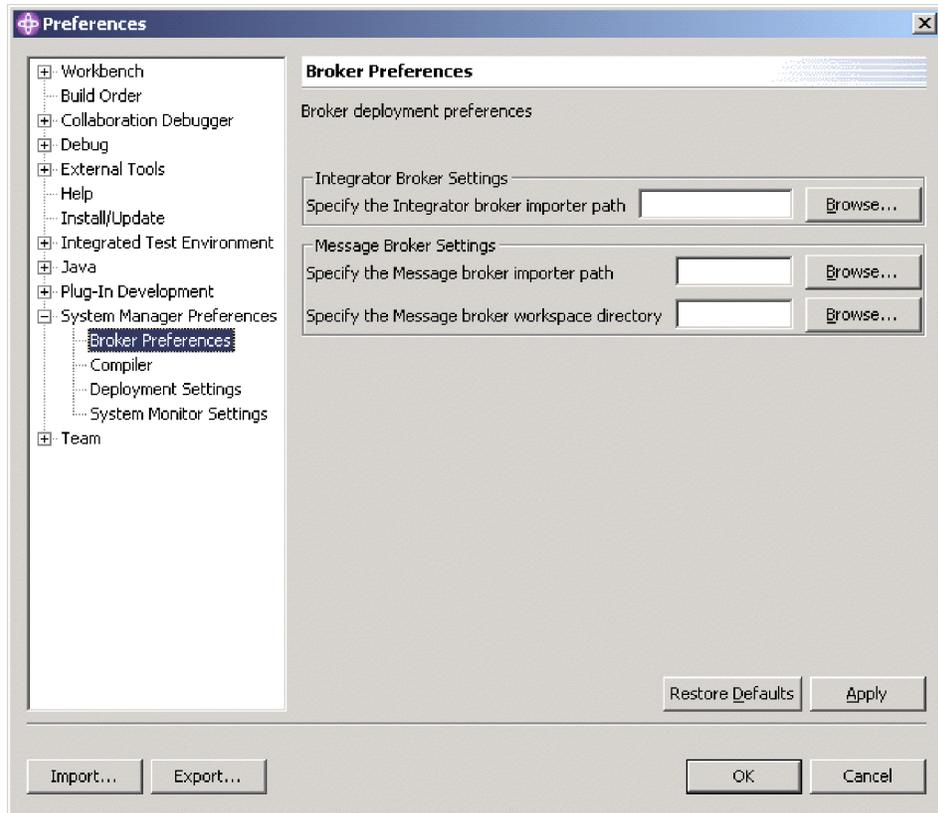


図 32. ブローカー設定

4. 「コンパイラー」設定インターフェースを使用すると、マップやコラボレーションの開発で必要とされるライブラリーのロケーションを指定できるため、コンパイラーによるライブラリーの配置が可能になります。

コンパイラー設定をセットするには、以下の手順を実行します。

- a. 「**System Manager 設定**」を展開して、「**コンパイラー**」を選択します。
- b. 「**新規**」をクリックします。
- c. 「**クラスパスを追加**」ダイアログで、コンパイラー・クラスパスに追加するライブラリーにナビゲートし、ライブラリーを選択してから、「**開く**」をクリックします。

105 ページの図 33 に「コンパイラー」設定インターフェースを示します。

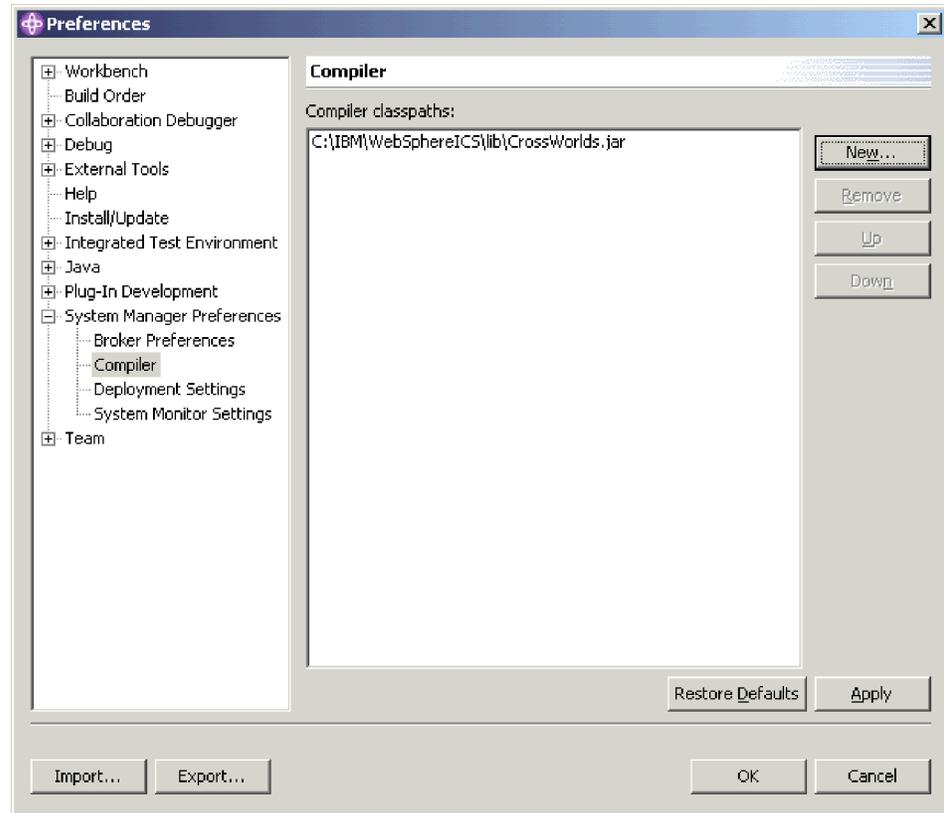


図 33. Compiler 設定

5. 「配置設定」インターフェースを使用すると、System Manager からサーバーへコンポーネントを配置する際の関連オプションを設定できます。

展開設定をセットするには、以下の手順を実行します。

- a. 「System Manager 設定」を展開して、「配置設定」を選択します。
- b. 「ドラッグ・アンド・ドロップによる配置の際にサーバーのコンポーネントを上書き」チェック・ボックスを有効にします。

106 ページの図 34 に「配置設定」インターフェースを示します。

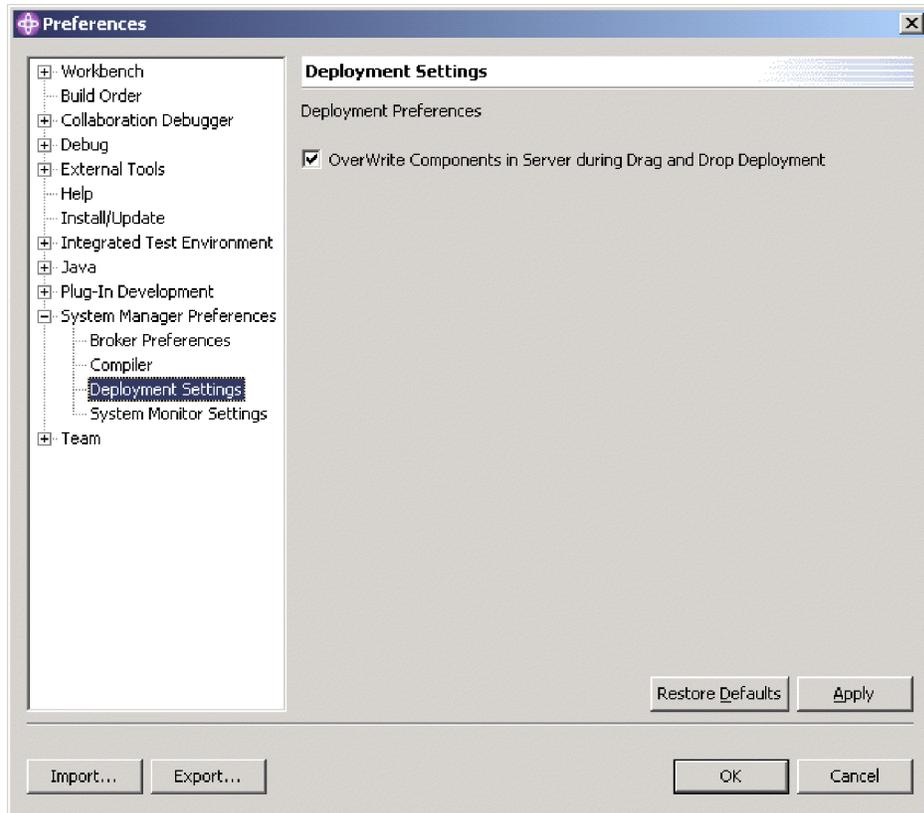


図 34. 「配置設定」 インターフェース

6. 「System Monitor 設定」 インターフェースを使用すると、InterChange Server コンポーネント管理ビューのモニター機能の振る舞いを構成できます。

System Monitor 設定をセットするには、以下の手順を実行します。

- a. 「System Manager 設定」を展開して、「System Monitor 設定」を選択します。
- b. 「システム表示」ビューに情報を表示するためにシステム状態をチェックしている間、System Manager を待機させる時間を「ポーリング間隔」フィールドに秒単位で入力します。

107 ページの図 35 に、「System Monitor 設定」インターフェースを示します。

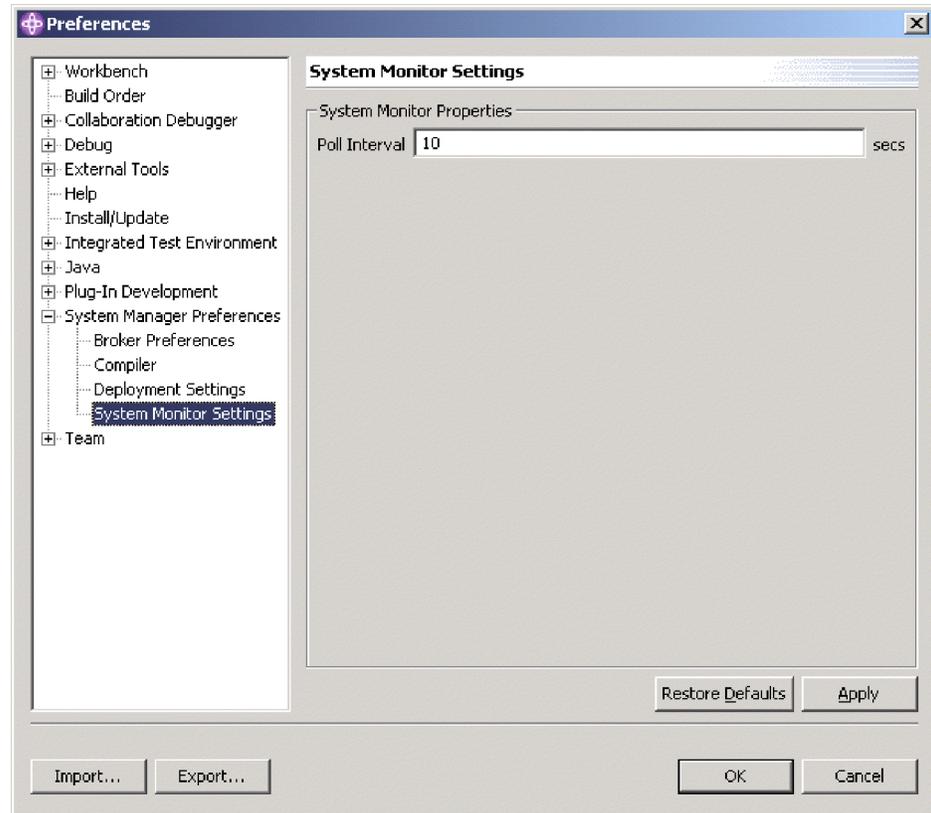


図 35. System Monitor 設定

システム表示ビューについては、「システム管理ガイド」を参照してください。

7. 「適用」をクリックして設定を保管し、「設定」ダイアログでの作業を継続するか、「OK」をクリックして設定を保管し、ダイアログを終了します。

第 4 章 ビジネス・プロセス・インターフェースの開発

この章では、InterChange Server を伴う WebSphere Business Integration システムで使用するビジネス・プロセス・インターフェースを開発する際に、推奨される方法について説明します。サイトの具体的な環境により異なりますが、並行して複数のインターフェースを開発したり、開発サイクルのさまざまな時点で、さまざまなインターフェースを開発することができます。

大抵のインターフェースを開発する際には、ここで説明する手順に従う必要があります。この手順は全体的な開発作業に対するフレームワークを提供しますが、実際の実装でこの手順を実行する場合には、それぞれの作業でより多くの反復が必要となることがあります。

この章を構成するセクションは次のとおりです。

- 『全体的な開発フロー』
- 111 ページの『開発作業の順序』

全体的な開発フロー

統合コンポーネントを開発する際は、次の一般規則に従うことをお勧めします。

• コネクタおよびデータ・ハンドラー

コネクタとアプリケーション固有のビジネス・オブジェクトは、統合されるアプリケーションと直接通信する統合コンポーネントです。

- カスタム・コネクタを開発する必要がある場合、その開発作業は他の開発作業と並行して進めることができます。コネクタの開発に必要なスキル・セットは、他のコンポーネントの開発に必要なスキル・セットとは異なるので、このタスクが必要な場合は、1 人の開発者にコネクタ開発のみを担当させるのが一般的です。この方法を使用すると、1 人の開発者がコネクタで作業し、同時に開発チームの残りの開発者が他のコンポーネントで作業することができます。
- コラボレーションおよびマップは汎用オブジェクトに依存し、汎用オブジェクトはアプリケーション固有のビジネス・オブジェクトに依存します。アプリケーション固有のビジネス・オブジェクトは、それを処理するコネクタがメタデータをどのように使用するかをユーザーが知らない限り開発できません。そのため、まず最初にコネクタを選択して、実装の進んだ時点で従属コンポーネントを再設計および開発する恐れを少なくします。さらに、多くのアダプターは Object Discovery Agent (ODA) を機能させています。これにより、アプリケーション固有のビジネス・オブジェクトを非常に効率的に開発できるようになります。

• アプリケーション固有のビジネス・オブジェクト

アプリケーション固有のビジネス・オブジェクトは、コネクタの後に開発する必要があります。これは、コネクタ用のビジネス・オブジェクトを開発するためにコネクタを理解する必要があるからです。アプリケーション固有のビジネ

ス・オブジェクトは、汎用ビジネス・オブジェクトの後に開発する必要があります。これは、汎用ビジネス・オブジェクトは通常、あるインターフェースのアプリケーション固有のビジネス・オブジェクトのスーパーセットを表すからです。

アプリケーション固有のビジネス・オブジェクトは、反復して作成することをお勧めします。アプリケーション固有のビジネス・オブジェクトは、まず比較的単純な構造で作成してからそれをテストして、コネクターがそのビジネス・オブジェクト構造を使用してアプリケーションと正常にデータを交換できることを確認します。次に複雑なレイヤーを追加し、再度ビジネス・オブジェクトをテストして、変更してもインターフェースが動作することを確認します。アプリケーション固有のビジネス・オブジェクトがインターフェースの要件を満たすくらい大規模および複雑になるまで、このプロセスを繰り返します。

• 汎用ビジネス・オブジェクト

インターフェースに必要なアプリケーション固有のビジネス・オブジェクトをすべて開発した場合、インターフェースのアプリケーション固有のビジネス・オブジェクトのスーパーセットとして、汎用ビジネス・オブジェクトを開発します。

インストーラーまたは業界ソリューションの一部として提供された既存の汎用ビジネス・オブジェクトを調べて、不要なカスタム開発を避けます。すでに作成されているものに時間を費やすべきではありません。既存の内容を使用できるか評価する場合、そのサイトに関係のあるライセンス交付条件を調べて、ライセンスがそのオブジェクトの使用を許諾しているかどうか検証してください。

最終的に、次を含む汎用ビジネス・オブジェクトを開発します。

- インターフェースの現行設計に合わせるために必要なすべての属性
- 予測される将来のインターフェースへの変更または追加に合わせるために必要なすべての属性
- インターフェースの現行要件または将来の要件を満たさない不必要な属性

• マップと関係

インターフェース用のビジネス・オブジェクトを識別または開発した場合、アプリケーション固有のオブジェクトを汎用オブジェクトに、汎用オブジェクトをアプリケーション固有のオブジェクトに変換するマップを開発する必要があります。Map Designer ツールを使用してマップを作成し、変換規則を定義し、汎用ビジネス・オブジェクトまたはアプリケーション固有のビジネス・オブジェクトをサンプルとして入力してマップの単体テストを行います。

単体テストを行ってマップ変換を検証したら、インターフェースに必要な関係を作成し、適切なマップに相互参照を実行します。マップに関連するインターフェースが実行時に実行される順序で、マップを単体テストします。

詳しくは、「マップ開発ガイド」を参照してください。

• コラボレーション・テンプレート

コラボレーション・テンプレートは、インターフェースのビジネス・ロジックを定義します。最も単純なコラボレーションは、コネクター間でビジネス・オブジェクトを経路指定するだけです。その他のコラボレーションには、トランザクション、長期継続ビジネス・プロセス、他のコラボレーション・オブジェクトへの

処理の委任などの複雑な対話も含まれる場合があります。いずれ場合も、コラボレーションは、インターフェース内のアプリケーション固有のビジネス・オブジェクトのスーパーセットを表す、汎用ビジネス・オブジェクトを中心として置かれます。

コラボレーションの開発および単体テスト中には、アプリケーション固有のビジネス・オブジェクト、マップ、またはコネクターを使用する必要はありません。コラボレーションが実行時に汎用ビジネス・オブジェクトのデータを大幅に変更しない場合や、汎用ビジネス・オブジェクトの構造に明示的に依存する操作を実行する場合には、コラボレーション・テンプレートを開発する前に汎用ビジネス・オブジェクトを開発する必要もありません。ビジネス・プロセスがしっかりと定義されていれば、汎用ビジネス・オブジェクトが全体的に開発されるまで、単純化された汎用ビジネス・オブジェクトを使用してコラボレーション・テンプレートを開発およびテストできます。

コラボレーション・テンプレートへのニーズを判断する場合は、すでに使用可能なテンプレートがどれかを調べ、使用可能で適切なコラボレーション・テンプレートの使用がサイトのライセンス交付条件によって許可されるかどうか判断します。適切な既存のコラボレーション・テンプレートがない場合は、カスタム・テンプレートを開発する必要があります。

詳しくは、「コラボレーション開発ガイド」、および使用するコラボレーションに固有の資料を参照してください。

• コラボレーション・オブジェクト

コラボレーション・テンプレートおよびそれに付随するすべてのコンポーネントを開発したら、そのテンプレートに基づいてコラボレーション・オブジェクトを作成する必要があります。

コラボレーション・オブジェクトの単体テストを行うときには、コネクター・エージェントを実行せずに、統合テスト環境または Test Connector を使用します。これにより、インターフェースの健全性をテストするときに、接続性の問題を考慮したりマップやコラボレーション・ロジックに関連するエラーを除外したりせずに済みます。その後、コネクター・エージェントを実行した状態でインターフェース全体をテストします。

開発作業の順序

インターフェース内の多くの統合コンポーネントは、別の統合コンポーネントに依存し、参照するので、各コンポーネントを開発する作業は相互に関連し、反復性があります。

統合コンポーネントを開発する際は、次の順序で行うことをお勧めします。

1. 112 ページの『カスタム・コネクターの開発』
2. 112 ページの『アプリケーション固有のビジネス・オブジェクトの開発または変更』
3. 112 ページの『コネクターの構成』
4. 113 ページの『接続性およびアプリケーション固有のビジネス・オブジェクトのテスト』

5. 115 ページの『汎用ビジネス・オブジェクトの開発または変更』
6. 115 ページの『関係の構成』
7. 115 ページの『データベース接続プールの構成』
8. 115 ページの『マップの開発およびテスト』
9. 116 ページの『コラボレーション・テンプレートの開発およびテスト』
10. 117 ページの『ビジネス・オブジェクト・サポート用コネクタおよび関連するマップの構成』
11. 117 ページの『コラボレーション・オブジェクトの作成および構成』
12. 117 ページの『インターフェースの配置』

カスタム・コネクタの開発

プロジェクトが始まったらすぐに、必要なスキル・セットを持つ開発者は、必要なカスタム・コネクタの開発を開始する必要があります。

カスタム・コネクタの開発の詳細については、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

アプリケーション固有のビジネス・オブジェクトの開発または変更

このトピックでは、アプリケーション固有のビジネス・オブジェクトの開発の簡単な概要について説明します。アプリケーション固有のビジネス・オブジェクト定義の処理の詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

アプリケーション固有のビジネス・オブジェクトを作成または変更する場合、次を実行します。

- アダプター用のガイドを参照して、アプリケーション固有のビジネス・オブジェクトの生成に使用できる Object Discovery Agent (ODA) がアダプターに備わっているか判別します。ODA によって、この開発段階は非常に効率的になります。
- アダプターに ODA が備わっていない場合は、アプリケーション用のサンプル・オブジェクトまたはエンティティ用のアプリケーション固有のビジネス・オブジェクトを、別の顧客サイトから取得します。その際、顧客の内容について知的所有権契約に違反しないようにします。
- アダプターに ODA が備わっておらず、変更して使用するためのサンプル・ビジネス・オブジェクトや既存のビジネス・オブジェクトが存在しない場合は、Business Object Designer を使用してアプリケーション固有のビジネス・オブジェクトを開発する必要があります。

ビジネス・オブジェクト定義を作成するときには、ソース・アプリケーション・トリガーやその他のイベント検出機構も必要に応じて作成する必要があります。

アプリケーション固有のビジネス・オブジェクト定義の設計および開発の詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

コネクタの構成

アプリケーション固有のビジネス・オブジェクトの単体テストを行う目的でコネクタを構成する際、そのコネクタが実装において役割を果たすために必要なビジ

ネス・オブジェクト定義およびマップを、まだすべては開発していない場合があります。しかし、これらのコンポーネントがなくても、テストに必要なビジネス・オブジェクト定義のサポートを追加したり、正常にテストしたりすることはできます。他のコンポーネントの開発が終了したら、コネクタ定義を再構成して、ビジネス・オブジェクト定義のサポートを追加し、必要なマップを関連付ける必要があります。

アダプターのアプリケーション固有のプロパティの詳細については、アダプターのガイドを参照してください。コネクタの標準プロパティおよび Connector Configurator の使用方法の詳細については、159 ページの『第 7 章 コネクタの構成』を参照してください。

接続性およびアプリケーション固有のビジネス・オブジェクトのテスト

アプリケーション固有のビジネス・オブジェクトを開発して、そのサポートをコネクタ定義に追加したら、そのビジネス・オブジェクトを単体テストして、コネクタがそれを使用してアプリケーションと正常にデータを交換できることを確認する必要があります。このテストの実行には、インターフェースで最終的に使用される汎用オブジェクト、マップ、コラボレーション・テンプレートは必要ありません。アプリケーション固有のビジネス・オブジェクトを単体テストするには、以下の手順を実行します。

1. 次のように設計されたパススルー・コラボレーション・テンプレートを作成します。
 - コラボレーション・テンプレートは、From と To という 2 つのポートを持つ。
 - From ポートと To ポートはいずれも、アプリケーション固有のビジネス・オブジェクト定義をサポートする。
 - コラボレーション・テンプレートは、Main という単一シナリオを持つ。
 - From ポートは、ビジネス・オブジェクト定義に属するすべての動詞のためのトリガー・ポートとなるよう Main シナリオで構成される。
 - Main シナリオには、次の要素が含まれる。
 - 遷移リンクによってアクション・ノードに接続された開始ノード
 - アクション・ノードに接続され、次の特徴を持つサービス呼び出しノード
 - 「通常のサービス呼び出し」ペインの「**ポート**」ドロップダウン・メニューが「To」ポートに設定されている。
 - 「通常のサービス呼び出し」ペインの「**動詞**」ドロップダウン・メニューが「作成」動詞（つまり該当ビジネス・オブジェクトの主動詞）に設定されている。
 - 「通常サービス呼び出し」ペインの「**BO 変数**」フィールドが値 triggeringBusObj に設定されている。この変数は、コラボレーションを起動するビジネス・オブジェクトを参照するために使用される。
 - 遷移リンクによってアクション・ノードに接続された終了成功ノード

コラボレーション・テンプレートの作成の詳細については、「コラボレーション開発ガイド」を参照してください。

2. コネクタ定義のプロパティを構成し、ビジネス・オブジェクト定義のサポートを追加し、そのビジネス・オブジェクト定義の「エージェント・サポート」チェック・ボックスにチェックマークを付けます。
3. アプリケーション固有のビジネス・オブジェクトのサポートを、別の「ダミー」コネクタ定義に追加します。このコネクタ定義はテスト・ツールを使用してエミュレートするのみなので、完全に構成する必要はありません。ビジネス・オブジェクト定義の「エージェント・サポート」チェック・ボックスが使用可能になっているか確認します。
4. コネクタがイベント通知機能を担う場合は、以下の手順を実行します。

- パススルー・テンプレートに基づいてコラボレーション・オブジェクトを作成し、テストするコネクタを From ポートに、「ダミー」コネクタを To ポートにバインドします。
- コネクタ、コラボレーション・テンプレート、コラボレーション・オブジェクト、およびビジネス・オブジェクト定義をサーバーに配置します。

コンポーネントの配置の詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

- テストするコネクタのコネクタ・エージェントを開始します。

コネクタの開始の詳細については、ご使用のアダプターのガイドおよび「システム管理ガイド」を参照してください。

- Test Connector を開始し、ビジネス・オブジェクト定義のサポートを追加した他のコネクタのコネクタ定義を開いて、Test Connector をエージェントに接続します。

Test Connector の詳細については、261 ページの『第 11 章 Test Connector の使用』を参照してください。

- ソース・アプリケーションでイベントを起動します。
- テストが成功すると、コネクタはイベントを表すビジネス・オブジェクトをコラボレーションに送信し、それが Test Connector に要求として表示されます。

5. コネクタが要求処理機能を担う場合は、以下の手順を実行します。

- パススルー・テンプレートに基づいてコラボレーション・オブジェクトを作成し、テストするコネクタを To ポートに、「ダミー」コネクタを From ポートにバインドします。

コラボレーション・オブジェクトの作成の詳細については、219 ページの『第 9 章 コラボレーション・オブジェクトの構成』を参照してください。

- コネクタ、コラボレーション・テンプレート、コラボレーション・オブジェクト、およびビジネス・オブジェクト定義をサーバーに配置します。

コンポーネントの配置の詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

- テストするコネクタのコネクタ・エージェントを開始します。

コネクタの開始の詳細については、ご使用のアダプターのガイドおよび「システム管理ガイド」を参照してください。

- Test Connector を開始し、ビジネス・オブジェクト定義のサポートを追加した他のコネクタのコネクタ定義を開いて、Test Connector をエージェントに接続します。

Test Connector の詳細については、261 ページの『第 11 章 Test Connector の使用』を参照してください。

- Test Connector からビジネス・オブジェクト要求を送信します。
- テストが成功すると、コネクタはビジネス・オブジェクト要求を受け取って処理し、アプリケーション内にそのエンティティ用のレコードを作成します。

汎用ビジネス・オブジェクトの開発または変更

ビジネス・プロセスに適切な既存の汎用オブジェクトが存在するかどうか判别します。サイトのライセンス交付条件で、そのオブジェクトの使用が許可されることを確認します。

利用できる既存の汎用ビジネス・オブジェクトが存在しない場合、既存の汎用ビジネス・オブジェクトを拡張するかまたは、新規の汎用ビジネス・オブジェクトを作成します。新規ビジネス・オブジェクトの作成の詳細については、「ビジネス・オブジェクト開発ガイド」を参照してください。

関係の構成

その時点でのあらゆる関係を構成し、後の段階でマップを開発するときに使用できるようにします。

関係の構成の詳細については、「マップ開発ガイド」を参照してください。

データベース接続プールの構成

マップに必要な関係を構成したように、データベース接続プールも構成する必要があります。事前にデータベース接続プールを構成することによって、マップを開発するときにそれらの使用をマップにコーディングすることができます。

データベース接続プールの構成の詳細については、207 ページの『第 8 章 データベース接続プールの構成』を参照してください。

マップの開発およびテスト

必要な汎用ビジネス・オブジェクト、アプリケーション固有のビジネス・オブジェクト、関係、およびデータベース接続プールを作成したら、インターフェース上でオブジェクトを変換するマップを開発できます。

マップを開発するときには、Map Designer のデバッグ機能を使用してマップを単体テストします。次の時点でマップを単体テストします。

- 単純な変換規則 (移動、分割、結合など) をすべて定義した後。
- カスタム Java コードに必要なより複雑な変換規則を開発した後。
- 関係およびデータベース接続プールの使用を取り込んだ後。

関係を使用するマップをテストする場合、インターフェースのコンテキスト内で実行される順序でマップをテストします。このようにしないと、相互参照ロジックが正しく実行されません。

マップの開発およびテストの詳細については、「マップ開発ガイド」を参照してください。

コラボレーション・テンプレートの開発およびテスト

コラボレーション・テンプレートには、インターフェースのビジネス・プロセス・ロジックを定義します。使用可能な既存のコラボレーション・テンプレートを調べて、ビジネス・プロセスに合うものがあるかどうか判断する必要があります。プロセスにそのまま当てはまるコラボレーションがない場合、類似するものがないかどうか判断します。類似するものがある場合、サイトのライセンス交付条件で許可されればそれを変更できるので、開発プロセスが速まります。

類似するテンプレートがない場合、新規のコラボレーション・テンプレートを開発する必要があります。新規のコラボレーション・テンプレートの開発方法、または既存のテンプレートの変更方法の詳細については、「コラボレーション開発ガイド」を参照してください。

コラボレーション・テンプレートの単体テストは、マップ、アプリケーション固有のビジネス・オブジェクト、またはインターフェース用に開発しているコネクタを使用せずに行うことができます。コラボレーション・テンプレートを単体テストするには、以下の手順を実行します。

1. コラボレーション・テンプレートによって処理される汎用ビジネス・オブジェクトのサポートを、PortConnector 定義に追加します。
2. コラボレーション・テンプレートに基づいてコラボレーション・オブジェクトを作成し、コラボレーションのポートを PortConnector にバインドします。

コラボレーション・オブジェクトの作成の詳細については、219 ページの『第 9 章 コラボレーション・オブジェクトの構成』を参照してください。

3. PortConnector、コラボレーション・テンプレート、コラボレーション・オブジェクト、およびビジネス・オブジェクト定義をサーバーに配置します。

コンポーネントの配置の詳細については、68 ページの『System Manager を使用したパッケージへのコンポーネントのエクスポート』を参照してください。

4. Test Connector を開始し、PortConnector 定義を開いて、Test Connector をエージェントに接続します。

Test Connector の詳細については、261 ページの『第 11 章 Test Connector の使用』を参照してください。

5. データを含んだ汎用ビジネス・オブジェクトのインスタンスを作成し、Test Connector から要求としてイベントを送信します。

コラボレーションはイベントを受け取って処理し、各サービス呼び出しは汎用ビジネス・オブジェクトを Test Connector に送信します。このとき、その汎用ビジネス・オブジェクトを編集して、データの変更を調べることができます。

6. InterChange Server ログ出力を調べて、コラボレーション・ロジックが期待どおりに実行されたかどうか確認します。

ビジネス・オブジェクト・サポート用コネクタおよび関連するマップの構成

開発済みのすべてのコンポーネントを一緒にテストするには、コネクタ定義を再構成して、インターフェースに加わるために必要な、アプリケーション固有のオブジェクトおよび汎用オブジェクトのサポートを追加する必要があります。システムによってマップは自動的に関連付けられますが、オブジェクトを変換するマップが複数ある場合は、明示的にマップを関連付ける必要があります。詳細については、159 ページの『第 7 章 コネクタの構成』を参照してください。

コラボレーション・オブジェクトの作成および構成

必要なビジネス・オブジェクト定義のサポートをコネクタに追加したら、テンプレートに基づいてコラボレーション・オブジェクトを作成し、そのポートを適切なコンポーネントにバインドできます。

コラボレーション・オブジェクトの処理の詳細については、219 ページの『第 9 章 コラボレーション・オブジェクトの構成』を参照してください。

インターフェースの配置

インターフェースに必要なすべてのコンポーネントを作成したら、ローカル InterChange Server に配置してインターフェースをテストする必要があります。配置の詳細については、70 ページの『サーバーへのコンポーネントの展開』を参照してください。

インターフェースのテスト

インターフェースをローカルの InterChange Server インスタンスに配置したら、インターフェース全体をテストして、コンポーネントが一緒に動作するときビジネス要件を満たすかどうか確認する必要があります。インターフェースをテストするには、以下の手順を実行します。

1. 必要なすべてのコンポーネントがアクティブであるか確認します。
2. ソース・システムでイベントを起動します。
3. InterChange Server のログ出力を調べて、インターフェースのコンポーネントが正しく実行されているか確認します。
4. 宛先システムを調べて、エンティティが正常に処理されたかどうか、およびデータが正しく表示されているかを確認します。

第 5 章 WebSphere InterChange Server の構成

InterChange Server を稼働させるには、環境固有の情報で構成する必要があります。構成情報は、製品ディレクトリーにある InterchangeSystem.cfg というファイルに XML 形式で保管されています。InterChange Server の構成には、次の 2 つのツールを使用します。

- InterChange Server 構成ウィザード

このツールは、InterChange Server のオプションをすべて設定できるわけではありませんが、サーバーが稼働していなくても使用できます。

- System Manager

このツールは、InterChange Server 構成ウィザードよりも多くのオプションを設定できますが、サーバーが稼働し、System Manager がそれに接続している必要があります。

この章では、InterChange Server 構成ウィザードと System Manager を使用して Server インスタンスを構成する方法を説明します。この章は次のセクションから構成されます。

- 『InterChange Server 構成ウィザードの使用』
- 125 ページの 『System Manager の使用』

InterChange Server 構成ウィザードの使用

InterChange Server 構成ウィザードを始動して構成ファイルを更新するには、以下の手順を実行します。

1. 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere InterChange Server 構成ウィザード」を選択します。
2. 次のセクションの説明に従い、ウィザードの各タブで各フィールドを希望の値に設定します。
 - 『ウィザードを使用したサーバー・プロパティの構成』
 - 121 ページの 『ウィザードを使用して WebSphere MQ プロパティを構成』
 - 122 ページの 『ウィザードを使用したデータベース・プロパティの構成』
 - 124 ページの 『ウィザードを使用したセキュリティー・プロパティの構成』
3. 「適用」をクリックします。
4. 「変更は完了しました。」ダイアログが表示されたら、「OK」をクリックします。
5. 「終了」をクリックします。

ウィザードを使用したサーバー・プロパティの構成

構成ウィザードの「InterChange Server」タブを使用すると、サーバー自体の基本プロパティのいくつかを変更できます。InterChange Server プロパティを構成するには、以下の手順を実行します。

1. 「InterChange Server」タブをクリックします。

図 36に、「InterChange Server」タブを示します。

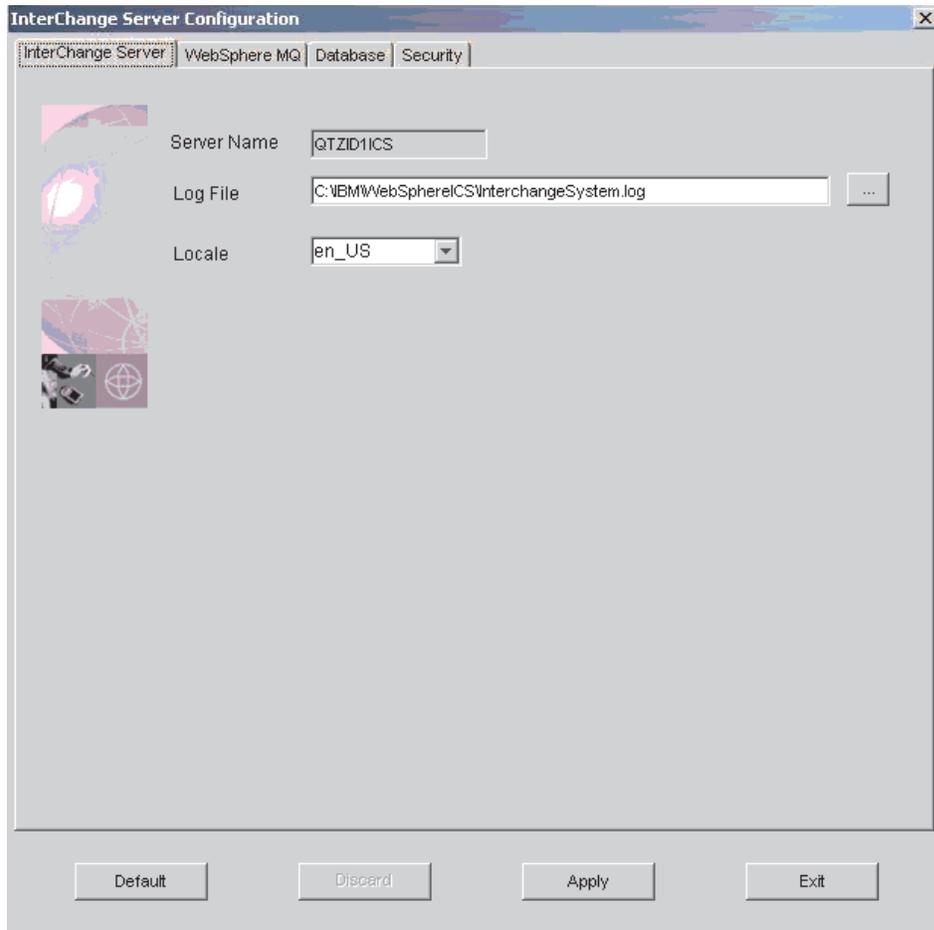


図 36. InterChange Server 構成タブ

2. InterChange Server インスタンスの名前を「**サーバー名**」フィールドに入力します。
3. 次のいずれかを実行して、「**ログ・ファイル**」フィールドの値を設定します。
 - InterChange Server ログイングをコンソールへ送る場合は、フィールドに値 **STDOUT** を入力します。
 - InterChange Server ログイングの送信先となるファイルを選択するには、フィールドにファイルの名前とパスを入力するか、「**参照**」をクリックします。

注: このプロパティを設定する際は、18 ページの『開発環境でのログイングの構成』に記載されている情報を考慮してください。
4. 「**ロケール**」ドロップダウン・メニューから目的のロケールを選択します。

ウィザードを使用して WebSphere MQ プロパティを構成

構成ウィザードの「WebSphere MQ」タブを使用すると、InterChange Server が WebSphere MQ パーシスタント・メッセージング・サーバーと通信する方法を指定するプロパティを変更できます。WebSphere MQ プロパティを構成するには、以下の手順を実行します。

1. 「WebSphere MQ」タブをクリックします。

図 37に、「WebSphere MQ」タブを示します。

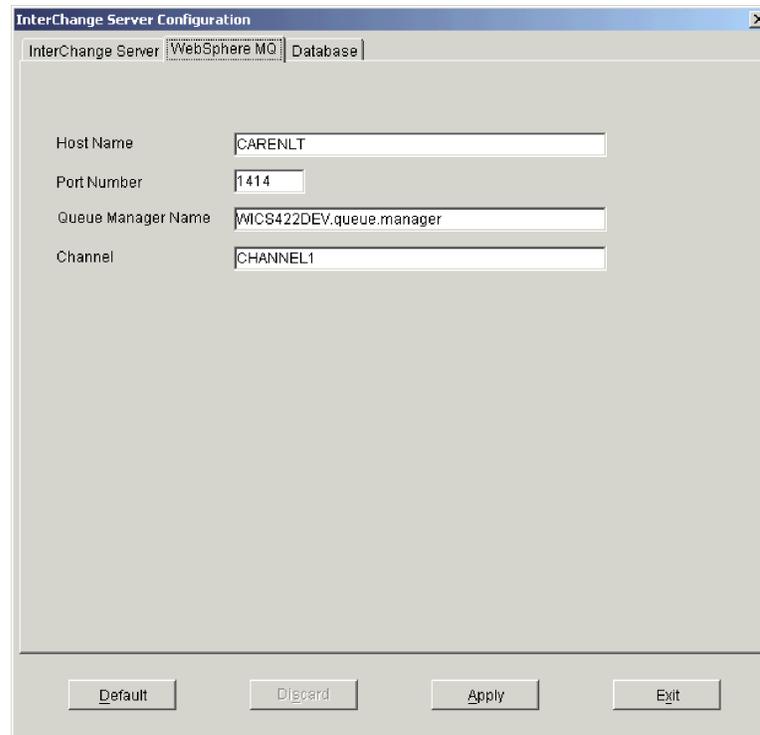


図 37. WebSphere MQ 構成タブ

2. 「ホスト名」フィールドに、WebSphere MQ サーバーがインストールされているコンピューターの名前を入力します。
3. 「ポート番号」フィールドに、WebSphere MQ でクライアントがサーバーと通信するためのポートを入力します。

デフォルト・ポートは 1414 です。

4. 「キュー・マネージャー名」フィールドに、InterChange Server が処理するフローのメッセージを永続的に保管するキューを組み込むために作成されたキュー・マネージャーの名前を入力します。
5. 「チャンネル」フィールドに、WebSphere MQ のクライアントが WebSphere MQ サーバーと通信するチャンネルの名前を入力します。デフォルト値は CHANNEL1 です。CHANNEL1 の使用が可能であるかどうかと、使用できない場合には、いずれのチャンネルを使用できるかを、サイトの WebSphere MQ 管理者に確認してください。

ウィザードを使用したデータベース・プロパティの構成

構成ウィザードの「データベース」タブを使用すると、リポジトリ、イベント管理、およびトランザクション管理データを格納するために InterChange Server が使用するデータベースを指定するプロパティを変更できます。データベース・プロパティを構成するには、以下の手順を実行します。

1. 「データベース」タブをクリックします。

図 38 に、「データベース」構成タブを示します。

The screenshot shows the 'InterChange Server Configuration' dialog box with the 'Database' tab selected. The dialog is divided into several sections:

- Database Connectivity:** Database Driver is set to 'DB2'. Max Connections is set to 'Unlimited' (checked). Max Pools is set to '10'. Idle Timeout is set to '2'.
- Event Management:** Host Name, Login (wicsadmin), Database (icsrepos), Password (masked), Max Connections (Unlimited checked), and Port Number are visible.
- Transactions:** Host Name, Login (wicsadmin), Database (icsrepos), Password (masked), Max Connections (Unlimited checked), and Port Number are visible.
- Repository:** Host Name, Login (wicsadmin), Database (icsrepos), Password (masked), Max Connections (Unlimited checked), and Port Number are visible.
- Flow Monitoring:** Host Name, Login (wicsadmin), Database (icsrepos), Schema Name, Password (masked), Port Number, and Max Connections (Unlimited checked) are visible.

At the bottom, there are buttons for 'Default', 'Discard', 'Apply', and 'Exit'.

図 38. データベース接続構成タブ

2. 「データベース・ドライバー」ドロップダウン・メニューから、データベース・ベンダーに応じて適切な値を選択します (DB2、MQ SQL Server(Type 4)、または Oracle(Type 4))。
3. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「最大接続数」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「無制限」チェック・ボックスをオンにします。
4. 「最大プール数」フィールドに、InterChange Server がキャッシュに入れるデータベース接続を組み込むために設定するプールの最大数を入力します。

5. 「アイドル・タイムアウト」フィールドに、接続オブジェクトのアイドル状態が何分続いたら再利用のためデータベース接続プールへ戻すかを示す数値を分単位で入力します。
6. 「イベント管理」ペインで次の手順を実行して、イベント管理サービス用のデータベース接続を構成します。
 - a. 「データベース・ドライバー」ドロップダウン・メニューから MQ SQL Server(Type 4) または Oracle(Type 4) を選択した場合は、データベース・サーバーが常駐するコンピューターの名前を「ホスト名」フィールドに入力します。
 - b. 「データベース」フィールドに、データベースの名前を入力します。
 - c. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「最大接続数」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「無制限」チェック・ボックスをオンにします。
 - d. 「ログイン」フィールドに、InterChange Server が指定のデータベースへのログインに使用するユーザー名を入力します。
 - e. 「パスワード」フィールドに、ステップ 6d で指定したユーザー名のパスワードを入力します。
 - f. 「データベース・ドライバー」ドロップダウン・メニューから、「MQ SQL Server(Type 4)」または「Oracle(Type 4)」を選択した場合は、クライアントがデータベース・サーバーと通信するポート番号を「ポート番号」フィールドに入力する必要があります。
7. 「トランザクション」ペインでステップ 6 を繰り返して、トランザクション管理サービスのデータベース接続を構成します。
8. 「リポジトリ」ペインでステップ 6 を繰り返して、リポジトリ・サービスのデータベース接続を構成します。
9. 「フロー・モニター」ペインで次の手順を実行して、フロー・モニター・サービス用のデータベース接続を構成します。
 - a. 「データベース・ドライバー」ドロップダウン・メニューから「Oracle(Type 4)」を選択した場合は、データベース・サーバーが常駐するコンピューターの名前を「ホスト名」フィールドに入力します。

注: Microsoft SQL Server は WebSphere Business Integration Monitor でサポートされるデータベースではありません。SQL Server は WebSphere InterChange Server が使用するその他のデータベースに使用するデータベースであり、フロー・モニターは使用できません。

- b. 「データベース」フィールドに、データベースの名前を入力します。
- c. 「スキーマ」フィールドに、WebSphere Business Integration Monitor で使用するスキーマの名前を入力します。

フロー・モニター・データベースとして Oracle を使用する場合は、スキーマの名前はユーザー・アカウントの名前と同じになります。ただし、フロー・モニター・データベースとして IBM DB2 を使用する場合は、スキーマ名はユーザー・アカウントの名前と異なる場合があるため、正しいスキーマ名を指定する必要があります。

- d. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「**最大接続数**」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「**無制限**」チェック・ボックスをオンにします。
- e. 「**ログイン**」フィールドに、InterChange Server が指定のデータベースへのログインに使用するユーザー名を入力します。
- f. 「**パスワード**」フィールドに、ステップ 9e で指定したユーザー名のパスワードを入力します。
- g. 「**データベース・ドライバー**」ドロップダウン・メニューから、「**Oracle(Type 4)**」を選択した場合は、クライアントがデータベース・サーバーと通信するポート番号を「**ポート番号**」フィールドに入力する必要があります。

フロー・モニターの詳細については、「**システム管理ガイド**」を参照してください。

データベースのパスワードの変更

InterChange Server がデータベースのいずれかにアクセスするために使用するユーザー名のパスワードを変更するには、以下の手順を実行します。

1. 「**イベント管理**」、「**トランザクション**」、「**リポジトリ**」または「**フロー・モニター**」ペインで、「**パスワード**」フィールドの横にある「**変更**」をクリックします。
2. 「**新パスワード**」フィールドに新規パスワードを入力します。
3. 「**確認パスワード**」フィールドに再度新規パスワードを入力します。
4. 「**OK**」をクリックします。

ウィザードを使用したセキュリティー・プロパティーの構成

構成ウィザードの「**セキュリティー**」タブを使用すると、InterChange Server のセキュリティー・オプションを指定するプロパティーを変更できます。セキュリティー・プロパティーを構成するには、以下の手順を実行します。

1. 「**セキュリティー**」タブをクリックします。

125 ページの図 39 に「**セキュリティー構成**」タブを示します。

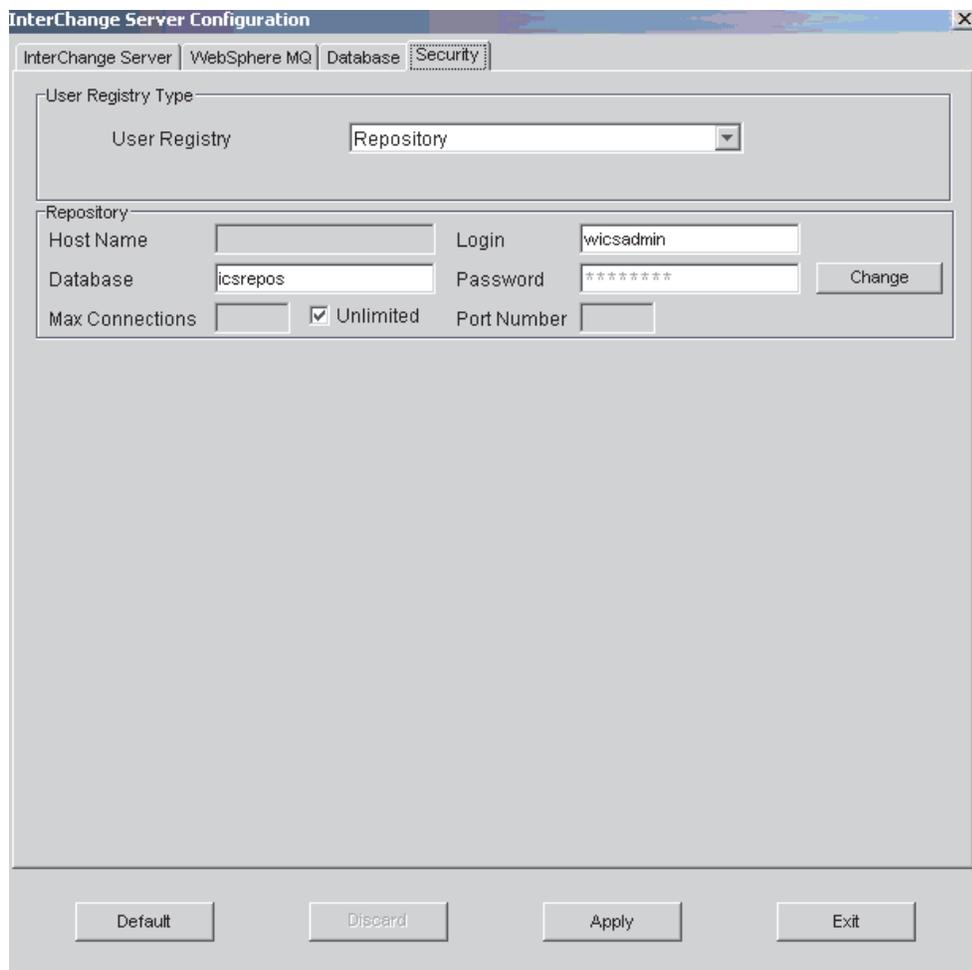


図 39. ウィザードの「セキュリティー」タブ

2. 「ユーザー・レジストリー」プルダウン・メニューで、「リポジトリー」または「LDAP」を選択します。「リポジトリー」を選択した場合は、以下の詳細を「セキュリティー」タブの「リポジトリー」領域に入力する必要があります。

- ホスト名
- データベース
- 最大接続数 (または「無制限」を選択)
- ログイン
- パスワード
- ポート番号

注: セキュリティーおよび構成の詳細については、「システム管理ガイド」を参照してください。

System Manager の使用

System Manager を使用すると、InterChange Server をより柔軟に構成できますが、サーバーが稼働し、System Manager がそれに接続している必要があります。

System Manager で InterChange Server 構成エディターを開くには、以下の手順を実行します。

1. 31 ページの『System Manager の始動』の説明に従って、System Manager を始動します。
2. 37 ページの『InterChange Server への接続』の説明に従って、System Manager を InterChange Server インスタンスに接続します。
3. InterChange Server インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「構成を編集」を選択します。
4. 次のセクションの説明に従って、各構成タブで必要な変更を加えます。
 - 『System Manager を使用して一般プロパティを表示』
 - 127 ページの『System Manager を使用したデータベース・プロパティの構成』
 - 130 ページの『System Manager を使用したトレース・レベルの構成』
 - 131 ページの『System Manager を使用したロギング・プロパティとトレース・プロパティの構成』
 - 133 ページの『System Manager を使用した E メール通知プロパティの構成』
 - 134 ページの『System Manager を使用してその他のプロパティを構成』
 - 136 ページの『System Manager を使用した環境変数の構成』
 - 136 ページの『System Manager を使用して WebSphere MQ プロパティを構成』
 - 137 ページの『System Manager を使用したセキュリティの構成』
 - 139 ページの『System Manager を使用したプライバシーの構成』
5. 完了したら、キーボード・ショートカット **Ctrl+S** を使用してファイルを保管します。

System Manager を使用して一般プロパティを表示

System Manager 構成ファイル・エディターの「一般」タブには、次の読み取り専用フィールドがあります。

- InterChange Server インスタンスの名前を示す「**サーバー名**」フィールド。
- サーバーが実行している IBM WebSphere InterChange Server ソフトウェアのバージョンを示す「**サーバーのバージョン**」フィールド。
- InterChange Server を実行しているコンピューターの RAM の容量を示す「**総メモリー**」フィールド。
- InterChange Server を実行しているコンピューターで使用可能な RAM の容量を示す「**空きメモリー**」フィールド。
- サーバー・インスタンスが始動した時刻を示す「**起動時間**」フィールド。
- System Manager を実行しているコンピューターが常駐する時間帯を示す「**現地時間帯**」フィールド。
- InterChange Server を実行しているコンピューターが常駐する時間帯を示す「**サーバーの時間帯**」フィールド。
- InterChange Server が稼働しているコンピューターのロケールを示す「**サーバーのロケール**」フィールド。

図 40 に、System Manager 構成ファイル・エディターの「一般」タブを示します。



図 40. InterChange Server 一般プロパティの構成

System Manager を使用したデータベース・プロパティの構成

構成ファイル・エディターの「データベース」タブを使用すると、リポジトリ、イベント管理、およびトランザクション管理データを格納するために InterChange Server が使用するデータベースを指定するプロパティを変更できます。このタブでは、データベースの回復力のために InterChange Server を構成することもできます。データベース・プロパティを構成するには、以下の手順を実行します。

1. 「データベース」タブをクリックします。

128 ページの図 41 に、System Manager 構成ファイル・エディターの「データベース」タブを示します。

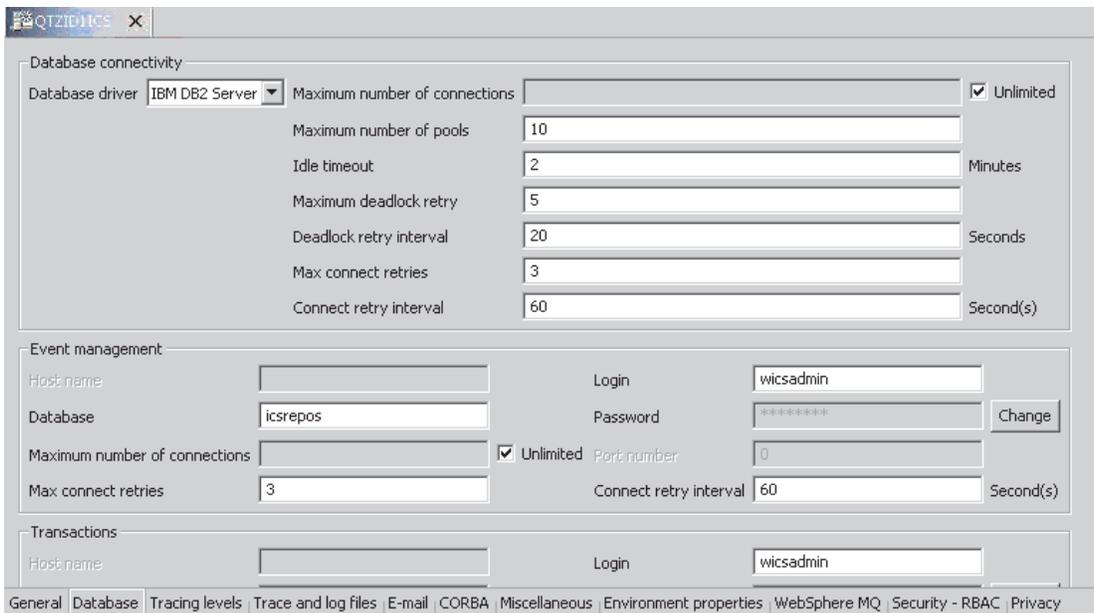


図 41. System Manager の「データベース」タブ

2. 「データベース・ドライバー」ドロップダウン・メニューから、データベース・ベンダーに応じて適切な値を選択します (IBM DB2 Server、MS SQL Server、または Oracle Server)。
3. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「**接続の最大数**」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「**無制限**」チェック・ボックスをオンにします。
4. 「**プールの最大数**」フィールドに、InterChange Server がキャッシュに入れるデータベース接続を組み込むために設定するプールの最大数を入力します。
5. 「**アイドル・タイムアウト**」フィールドに、接続オブジェクトのアイドル状態が何分続いたら再利用のためデータベース接続プールへ戻すかを示す数値を分単位で入力します。
6. 「**最大デッドロック再試行回数**」フィールドに、サーバーがデッドロックしたデータベース・トランザクションの実行を試行する回数を入力します。

このフィールドは、ステップ 7 に示す「**デッドロック再試行間隔**」フィールドに関連します。

データベース・デッドロックの際のサーバーの振る舞いについては、「システム管理ガイド」を参照してください。

7. 「**デッドロック再試行間隔**」フィールドに、サーバーがデッドロックしたデータベース・トランザクションの次の試行までに待機する秒数を入力します。

このフィールドは、ステップ 6 に示す「**最大デッドロック再試行回数**」フィールドに関連します。

データベース・デッドロックの際のサーバーの振る舞いについては、「システム管理ガイド」を参照してください。

8. データベースへの接続を失った場合 InterChange Server が再接続を試行する回数を「**最大接続再試行回数**」フィールドに入力します。

9. 「接続再試行間隔」フィールドに、データベースへの再接続を試行するまでの InterChange Server の待機時間 (秒) を入力します。
10. 「イベント管理」ペインで次の手順を実行して、イベント管理サービス用のデータベース接続を構成します。
 - a. 「データベース・ドライバ」ドロップダウン・メニューから MQ SQL Server(Type 4) または Oracle(Type 4) を選択した場合は、データベース・サーバーが常駐するコンピューターの名前を「**ホスト名**」フィールドに入力します。
 - b. 「データベース」フィールドに、データベースの名前を入力します。
 - c. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「**接続の最大数**」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「**無制限**」チェック・ボックスをオンにします。
 - d. 「**最大接続再試行回数**」フィールドに、接続を失った場合に InterChange Server が接続の回復を試行する回数を入力します。
 - e. 「**ログイン**」フィールドに、InterChange Server が指定のデータベースへのログインに使用するユーザー名を入力します。
 - f. 「**パスワード**」フィールドに、ステップ 6d (123 ページ) で指定したユーザー名のパスワードを入力します。
 - g. 「データベース・ドライバ」ドロップダウン・メニューから、「MS SQL Server(Type 4)」または「Oracle(Type 4)」を選択した場合は、クライアントがデータベース・サーバーと通信するポート番号を「**ポート番号**」フィールドに入力する必要があります。
 - h. 「**接続再試行間隔**」フィールドに、失われた接続の回復を試行後、次に試行するまでに InterChange Server が待機する時間 (秒) を入力します。
11. 「トランザクション」ペインでステップ 6 (123 ページ) を繰り返して、トランザクション管理サービスのデータベース接続を構成します。
12. 「リポジトリ」ペインでステップ 6 (123 ページ) を繰り返して、リポジトリ・サービスのデータベース接続を構成します。
13. 「フロー・モニター」ペインで次の手順を実行して、フロー・モニター・サービス用のデータベース接続を構成します。
 - a. 「データベース・ドライバ」ドロップダウン・メニューから「Oracle(Type 4)」を選択した場合は、データベース・サーバーが常駐するコンピューターの名前を「**ホスト名**」フィールドに入力します。

注: Microsoft SQL Server は WebSphere Business Integration Monitor でサポートされるデータベースではありません。SQL Server は WebSphere InterChange Server が使用するその他のデータベースに使用するデータベースであり、フロー・モニターは使用できません。

- b. 「データベース」フィールドに、データベースの名前を入力します。
- c. 「スキーマ」フィールドに、WebSphere Business Integration Monitor で使用するスキーマの名前を入力します。

フロー・モニター・データベースとして Oracle を使用する場合は、スキーマの名前はユーザー・アカウントの名前と同じになります。ただし、フロ

ー・モニター・データベースとして IBM DB2 を使用する場合は、スキーマ名はユーザー・アカウントの名前と異なる場合があるため、正しいスキーマ名を指定する必要があります。

- d. InterChange Server がデータベース・サーバーとの間に確立する接続の最大数を「**接続の最大数**」フィールドに入力するか、あるいは、InterChange Server の接続数を無制限にする場合は、「**無制限**」チェック・ボックスをオンにします。
- e. 「**最大接続再試行回数**」フィールドに、接続を失った場合に InterChange Server が接続の回復を試行する回数を入力します。
- f. 「**ログイン**」フィールドに、InterChange Server が指定のデータベースへのログインに使用するユーザー名を入力します。
- g. 「**パスワード**」フィールドに、ステップ 13f で指定したユーザー名のパスワードを入力します。
- h. 「**データベース・ドライバ**」ドロップダウン・メニューから、「Oracle(Type 4)」を選択した場合は、クライアントがデータベース・サーバーと通信するポート番号を「**ポート番号**」フィールドに入力する必要があります。
- i. 「**接続再試行間隔**」フィールドに、失われた接続の回復を試行後、次に試行するまでに InterChange Server が待機する時間 (秒) を入力します。

フロー・モニターの詳細については、「システム管理ガイド」を参照してください。

データベースのパスワードの変更

InterChange Server がデータベースのいずれかにアクセスするために使用するユーザー名のパスワードを変更するには、以下の手順を実行します。

1. 「イベント管理」、「トランザクション」、または「リポジトリ」ペインで、「パスワード」フィールドの横にある「**変更**」をクリックします。
2. 「**新パスワード**」フィールドに新規パスワードを入力します。
3. 「**確認パスワード**」フィールドに再度新規パスワードを入力します。
4. 「**OK**」をクリックします。

System Manager を使用したトレース・レベルの構成

フロー・トレースとサブシステム・トレースのプロパティを設定するには、「**トレース・レベル**」タブで以下の手順を実行します。

1. 「**トレース・レベル**」タブをクリックします。

131 ページの図 42 に、System Manager 構成ファイル・エディターの「**トレース・レベル**」タブを示します。

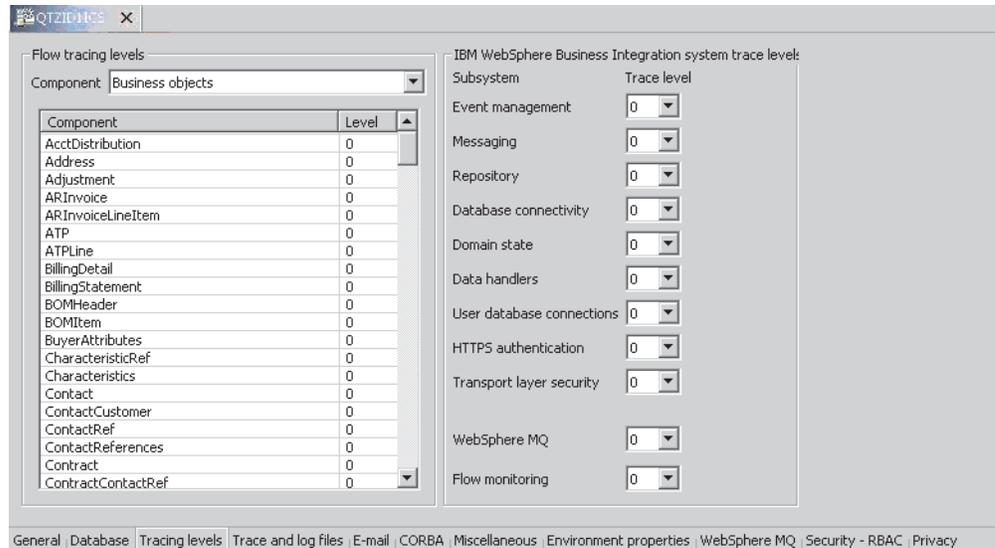


図 42. InterChange Server トレース・レベルの構成

- ビジネス・オブジェクトのフロー・トレース・レベルを設定するには、「フローのトレース・レベル」ペインで特定のビジネス・オブジェクトの「レベル」列から希望の値を選択します。

フロー・トレースの詳細については、「システム管理ガイド」を参照してください。

- InterChange Server サブシステムのトレース・レベルを設定するには、サブシステムに関連したドロップダウン・メニューから希望のトレース・レベルを選択します。

各サブシステムの各レベルでどのような情報が報告されるかについては、「システム・インストール・ガイド (Windows 版)」または「システム・インストール・ガイド (UNIX 版)」の付録『構成パラメーター』を参照してください。

System Manager を使用したロギング・プロパティとトレース・プロパティの構成

InterChange Server がランタイム情報を記録およびトレースするロケーションを構成するには、以下の手順を実行します。

- 「トレース/ログ・ファイル」タブをクリックします。

132 ページの図 43 に、System Manager 構成ファイル・エディターの「トレース/ログ・ファイル」タブを示します。

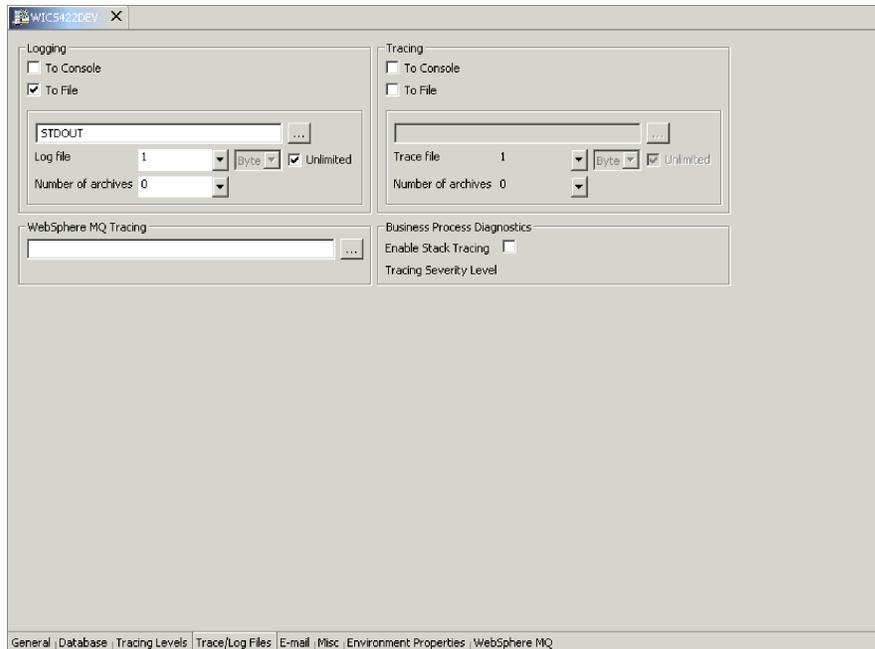


図 43. InterChange Server ログイング・プロパティとトレース・プロパティの構成

2. 「ログ」 ペインで次の手順を実行して、InterChange Server がランタイム情報をログに記録する方法を構成します。
 - InterChange Server でログ情報をコンソールへ出力したい場合は、「**コンソール**」にチェック・ボックスをオンにします。
 - InterChange Server でログ情報をファイルへ出力したい場合は、「**ファイル**」にチェック・ボックスをオンにします。このチェック・ボックスをオンにした場合は、次の手順も実行します。
 - a. 出力の記録先となるファイルの名前とパスをラベルのないテキスト・フィールドに入力するか、「参照」 ボタンをクリックしてファイルを選択します。
 - b. 次のいずれかを実行して、ログ・ファイルの管理方法を指定します。
 - ステップ 2a で指定したファイルにファイル・サイズの制限なしで InterChange Server に情報を書き込ませるには、「**無制限**」チェック・ボックスをクリックします。
- 注:** InterChange Server によってディスクがいっぱいになり、システム障害が発生する可能性もあるので、これを選択した場合はファイル管理に気をつけてください。
- ログ・ファイル・アーカイブ・アプローチを構成するには、以下の手順を実行します。
 - 1) ラベルのないドロップダウン・メニューから「**バイト**」、「**KB**」、「**MB**」、または「**GB**」を選択し、ログ・ファイルとアーカイブ・ファイルを、バイト、キロバイト、メガバイト、ギガバイトのいずれで保管するかを示します。

- 2) 「ログ・ファイル」ドロップダウン・メニューから数値を選択し、ログ・ファイルがどのくらい大きくなったらアーカイブするかを示すバイト数、キロバイト数、メガバイト数、ギガバイト数を指定します。
 - 3) 「アーカイブの数」ドロップダウン・メニューで、保持しているアーカイブ・ファイルがいくつになったらそれらを削除するかを示す数を選択します。
3. 「トレース」ペインのステップ 2 (132 ページ) を繰り返します。
 4. WebSphere MQ のトレースを有効にするには、出力の記録先となるファイルの名前とパスを、「WebSphere MQ トレース」ペイン内のラベルのないテキスト・フィールドに入力するか、「参照」ボタンをクリックしてファイルを選択します。
 5. スタック・トレースを使用するには、「スタック・トレースを使用可能にする」を選択し、「トレース重大度レベル」ドロップダウン・メニューから目的の重大度レベルを選択します。

スタック・トレースについて詳しくは、「システム管理ガイド」を参照してください。

System Manager を使用した E メール通知プロパティの構成

InterChange Server では、ビジネス・インテグレーション・システムにエラーが発生した場合に E メール通知を送信できます。デフォルトでは、サーバーは Java の E メール API を使用します。これは、多数の異なるサブシステムに E メール通知を送信できます。個々のコラボレーションでエラーが発生した場合にサーバーが E メール通知を送信できるようにするには、WebSphere Business Integration Adapter for e-Mail を使用するようにシステムを構成する必要があります。E メール通知用のシステム構成の詳細については、「システム管理ガイド」を参照してください。構成ファイルに必要な変更を加えるには (これは、「システム管理ガイド」で説明されている全タスクのうちの 1 つにすぎません)、以下の手順を実行します。

1. 「E メール」タブをクリックします。

134 ページの図 44 に、System Manager 構成ファイル・エディターの「E メール」タブを示します。

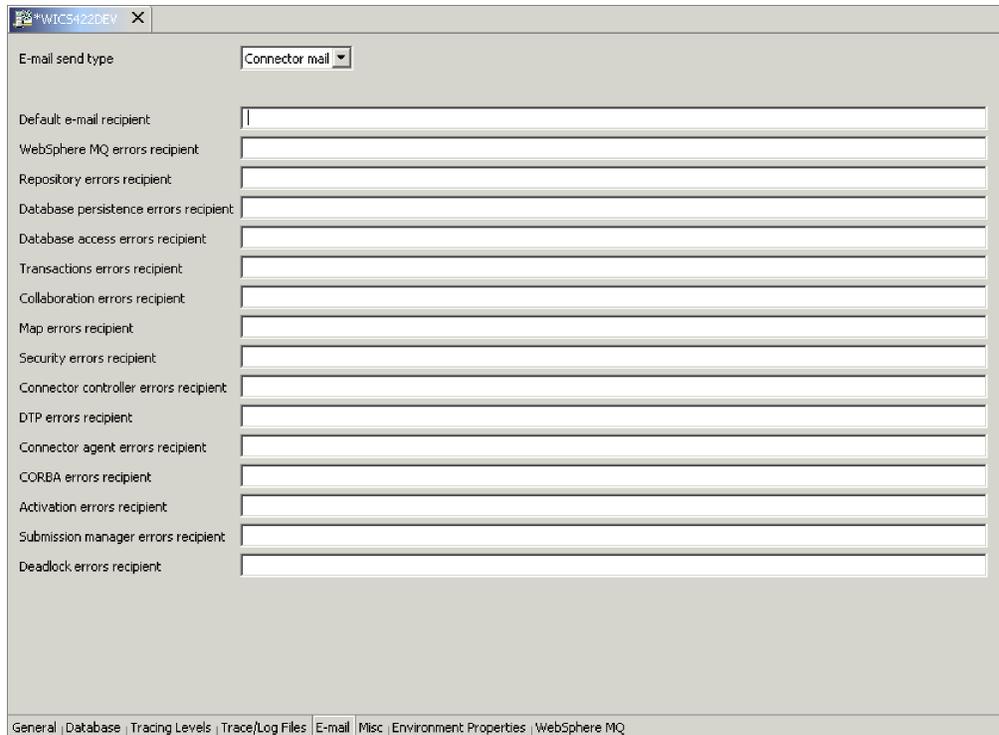


図 44. InterChange Server E メール通知プロパティの構成

2. 使用した E メール通知メカニズムに応じて、「電子メール送信タイプ」ドロップダウン・メニューから Java メールまたはコネクター・メールを選択します。

注: コラボレーションで E メール通知を送信できるのは、コネクター・メールを選択した場合のみです。

3. ステップ 2 で Java メールを選択した場合は、「SMTP メール・ホスト」フィールドに、SMTP サーバーがインストールされているコンピューターの名前を入力します。
4. 必要に応じて、各サブシステムのテキスト・フィールドに、有効な E メール・アドレスまたはコンマで区切った一連の E メール・アドレスを入力します。

System Manager を使用してその他のプロパティを構成

System Manager 構成ファイル・エディターの「各種」タブには、永続モニター、フロー制御、長期存続ビジネス・プロセスなどの機能を構成するためのペインがあります。これらの機能の設定を構成するには、以下の手順を実行します。

1. 「各種」タブをクリックします。

135 ページの図 45 に、System Manager 構成ファイル・エディターの「各種」タブを示します。

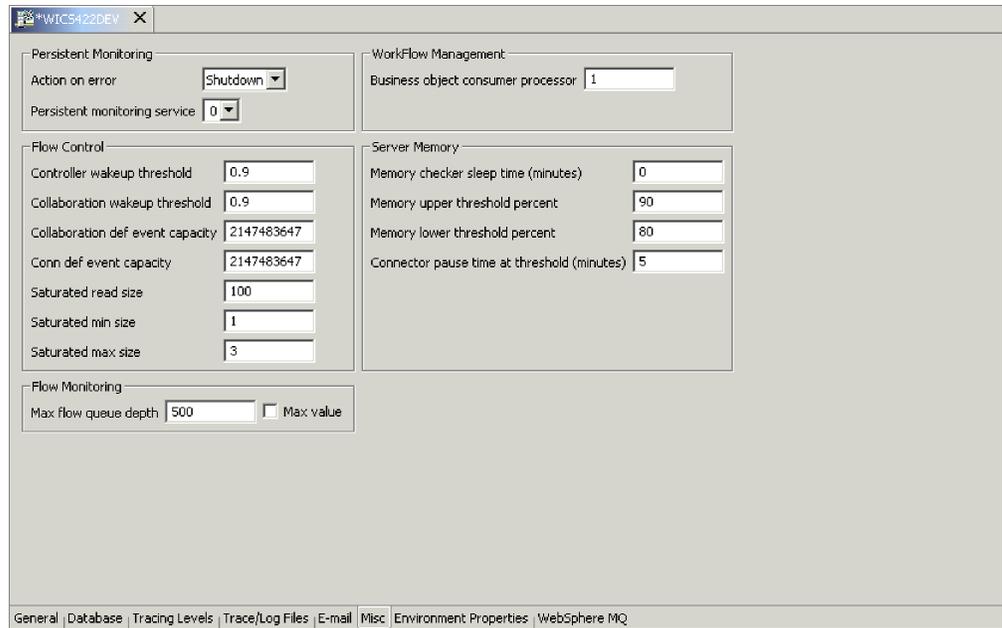


図 45. InterChange Server のその他のプロパティの構成

2. 「永続モニター」 ペインで、以下の手順を実行します。
 - a. 永続モニター・サブシステムでエラーが発生しても InterChange Server の稼働を続けたい場合は、「エラー時のアクション」ドロップダウン・メニューで継続を選択します。

サブシステムでエラーが発生した場合に InterChange Server をシャットダウンしたい場合は、「エラー時のアクション」ドロップダウン・メニューでシャットダウンを選択します。
 - b. 「永続モニター・サービス」ドロップダウン・メニューで希望のトレース・レベルを選択し、サブシステムのトレース・レベルを指定します。
3. 「フロー制御」ペイン内のインターフェース・エレメントについては、「システム管理ガイド」を参照してください。
4. 「フロー・モニター」ペイン内のインターフェース・エレメントについては、「システム管理ガイド」を参照してください。
5. 「ワークフロー管理」ペインの「ビジネス・オブジェクト・コンシューマー・プロセッサ」に、長期存続ビジネス・プロセスに関するビジネス・オブジェクトの管理用に作成するスレッドの数を入力します。

これらのスレッドは、ビジネス・オブジェクトが永続的な保管から検出された場合に、長期存続ビジネス・プロセスで処理を再開するために、コラボレーション・ランタイムから適切なコネクタ・コントローラーへビジネス・オブジェクトを送信します。長期存続ビジネス・プロセスの詳細については、「コラボレーション開発ガイド」を参照してください。

6. 「サーバーのメモリー」ペインの詳細については、367 ページの『メモリー・チェッカー・スレッドの使用』を参照してください。

System Manager を使用した環境変数の構成

「環境プロパティ」タブで、ビジネス・インテグレーション・システムに必要な任意の Java ユーザー環境プロパティを指定できます。環境によっては、こうした変数を指定する必要があります。InterChange Server を始動するスクリプトに追加する代わりに、「環境プロパティ」タブを使用して構成ファイルにそれらを追加できます。新規環境プロパティを追加するには、以下の手順を実行します。

1. 「環境プロパティ」タブをクリックします。

図 46 に、System Manager 構成ファイル・エディターの「環境プロパティ」タブを示します。

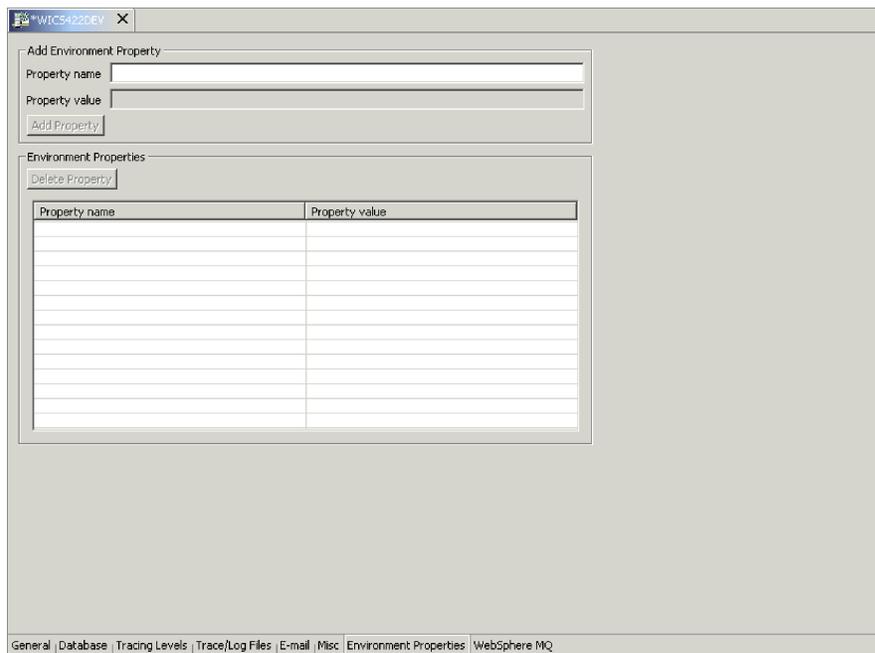


図 46. InterChange Server 環境変数の構成

2. 「プロパティ名」フィールドにプロパティの名前を入力します。
3. 「プロパティ値」フィールドにプロパティの値を入力します。
4. 「プロパティを追加」をクリックします。

プロパティを削除するには、プロパティを選択して「プロパティを削除」をクリックします。

プロパティの値を編集するには、プロパティを選択してそのプロパティの「プロパティ値」列でテキストを編集します。

System Manager を使用して WebSphere MQ プロパティを構成

InterChange Server と WebSphere MQ パーシスタント・メッセージング・ソフトウェアとの接続を構成するには、以下の手順を実行します。

1. 「WebSphere MQ」タブをクリックします。

図 47 に、System Manager 構成ファイル・エディターの「WebSphere MQ」タブを示します。

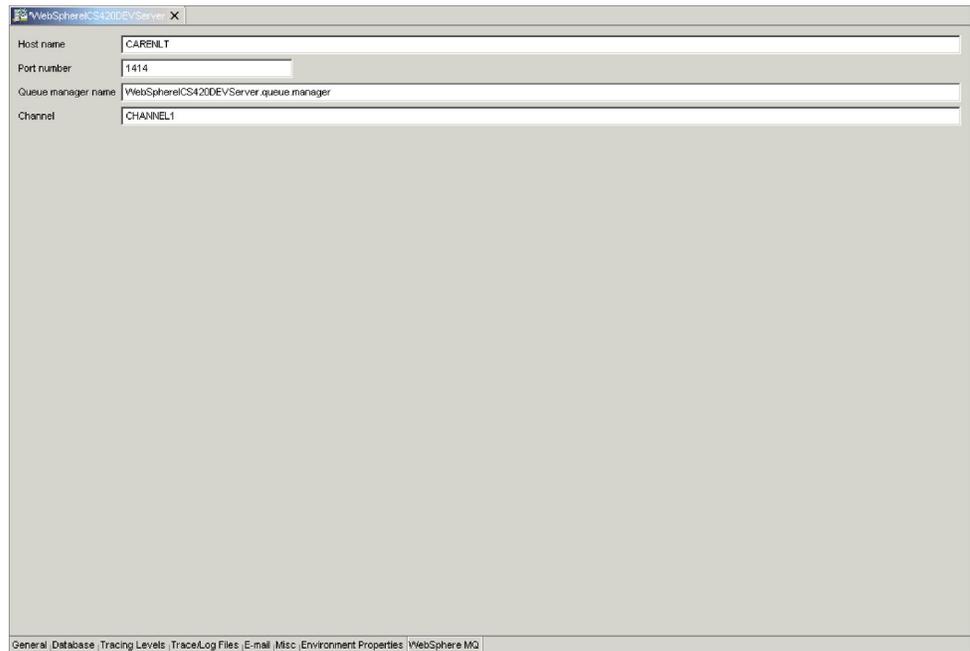


図 47. InterChange Server WebSphere MQ プロパティの構成

2. 「ホスト名」フィールドに、WebSphere MQ サーバーがインストールされているコンピューターの名前を入力します。
3. 「ポート番号」フィールドに、WebSphere MQ でクライアントがサーバーと通信するためのポートを入力します。

デフォルト・ポートは 1414 です。

4. 「キュー・マネージャー名」フィールドに、InterChange Server が処理するフローのメッセージを永続的に保管するキューを組み込むために作成されたキュー・マネージャーの名前を入力します。
5. 「チャネル」フィールドに、WebSphere MQ のクライアントが WebSphere MQ サーバーと通信するチャネルの名前を入力します。デフォルト値は CHANNEL1 です。CHANNEL1 の使用が可能であるかどうかと、使用できない場合には、いずれのチャネルを使用できるかを、サイトの WebSphere MQ 管理者に確認してください。

System Manager を使用したセキュリティの構成

InterChange Server の主要な機能の 1 つは、役割を使用して、システムにアクセスするユーザーにアクセス権を許可する機能であり、これは、役割ベースのアクセス制御 (RBAC) と呼ばれます。管理者は、簡単に役割を定義してこれをユーザーのグループに割り当てることができ、これにより、主要なコンポーネントへのアクセスを検証済みユーザーのみに制限できます。

RBAC 機能を使用すると、管理者または役割を管理する許可を持つユーザーのみが、ユーザーを作成して役割を割り当てることができることが保証されます。サー

バー上で RBAC が有効でない場合、任意のユーザーが、検証なしにユーザーや役割を作成することができます。したがって、製品のインストール後、サーバーを RBAC オンで使用する場合は、管理者に RBAC をオンにしてもらいます。これによって、ユーザーが RBAC をオンにしたり他のフィールドを編集することを回避します。

セキュリティおよび役割ベースのアクセス制御の詳細については、「システム管理ガイド」を参照してください。

注: セキュリティは、`repos_copy` コマンドで新規パラメーター `-xmsp` を使用して構成することもできます。 `repos_copy` の使用方法の詳細については、143 ページの『第 6 章 `repos_copy` の使用』を参照してください。

役割ベースのアクセス制御のために InterChange Server を構成するには、以下の手順を実行します。

1. 最初に管理者役割を持つユーザーを作成します。管理者の役割を持つユーザーがない場合、InterChange Server 構成で RBAC をオンにしてもサーバーは RBAC オフでリブートされます。
2. 「**セキュリティ: RBAC**」タブをクリックします。

図 48 に、System Manager 構成ファイル・エディターの「**セキュリティ: RBAC**」タブを示します。

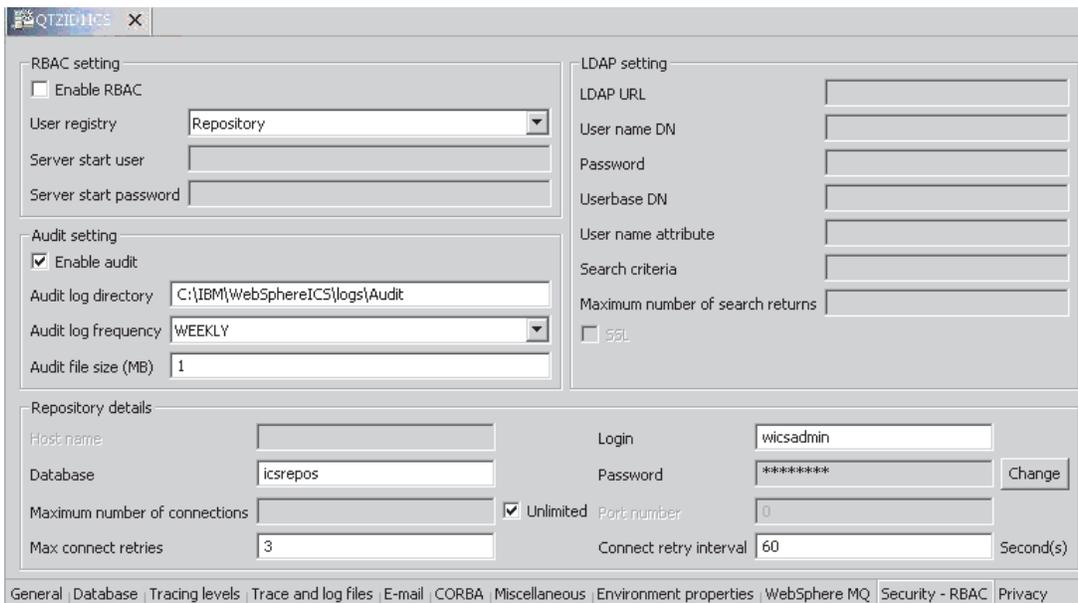


図 48. System Manager の「セキュリティ」タブ

3. 「**RBAC の使用可能化**」チェック・ボックスを選択します。
4. 「ユーザー・レジストリー」プルダウン・メニューで、「リポジトリー」または「LDAP」を選択します。

「リポジトリー」を選択した場合は、以下の詳細を「**セキュリティ: RBAC**」タブの「リポジトリーの詳細」領域に入力する必要があります。

- データベース

- 最大接続数または「**無制限**」チェック・ボックスを選択
- 最大接続再試行回数
- ログイン
- パスワード
- 接続再試行間隔

「**LDAP**」を選択した場合は、以下の詳細を「**セキュリティ: RBAC**」タブの「**LDAP 設定**」領域に入力する必要があります。

- LDAP URL
 - ユーザー名 DN
 - パスワード
 - ユーザーベース DN (基本識別名)
 - ユーザー名属性 (InterChange Server がユーザー名として使用する LDAP スキーマの属性)
 - 検索条件、LDAP ユーザーの検索に使用
 - 戻される最大検索数 (検索で戻される項目の最大数)
 - SSL、InterChange Server と LDAP 間のセキュア接続が必要かどうか示す
5. 「**サーバー開始ユーザー**」フィールドに、サーバーを開始するユーザー名を入力します。
 6. 「**サーバー開始パスワード**」フィールドに、ユーザー名に関連付けられているパスワードを入力します。
 7. 監査設定をオンにするには、「**監査の使用可能化**」チェック・ボックスを選択し、以下のフィールドに情報を入力します。
 - 監査ログ・ディレクトリー (監査ログ・ファイルのパス)
 - 監査ログの頻度 (例: 毎日、毎週、毎月)
 - 監査ファイルのサイズ (MB) (監査ファイルの最大サイズ)

InterChange Server の構成オプションの詳細については、「**システム管理ガイド**」を参照してください。

System Manager を使用したプライバシーの構成

エンドツーエンド・プライバシーは、InterChange Server の重要な機能の 1 つです。この機能を使用すると、メッセージがソース・アダプターを離れた瞬間から InterChange Server を経由して宛先アダプターに到着するまでの間、メッセージをセキュアに送信することができます。

セキュア・システムにとって重要なことは、エンドポイントの検証です。IBM WebSphere InterChange Server では、情報の流れのそれぞれのエンドポイントでセキュリティを提供することにより、端から端までの間で情報を保護します。

InterChange Server への業務通信が JMS を介して非同期でトランスポートされると、メッセージは処理を待機する間、ディスク上のキュー・マネージャーに保管されます。エンドツーエンド・プライバシーは、これらのメッセージが、このレベルにおいて保護されることを保証します。

注: エンドツーエンド・プライバシーの詳細については、「システム管理ガイド」を参照してください。

エンドツーエンド・プライバシーのために InterChange Server を構成するには、以下の手順を実行します。

1. 「プライバシー」タブをクリックします。

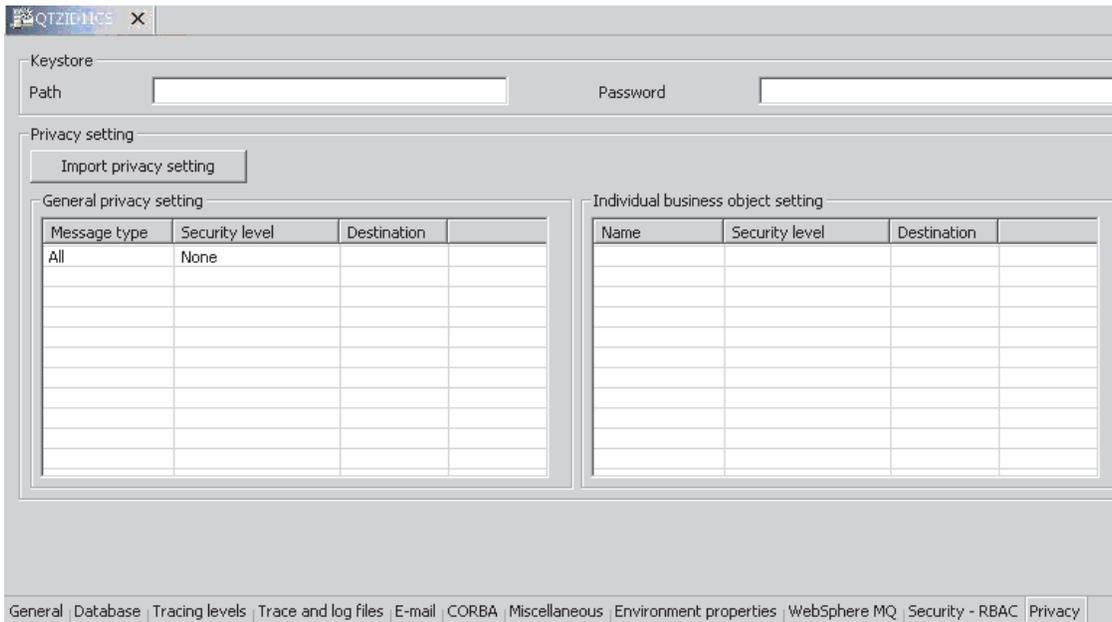


図 49. System Manager のプライバシー・タブ

2. 鍵ストアのパスと鍵ストア・パスワードを入力します。(鍵ストアの詳細については、「システム管理ガイド」を参照してください。)
3. 「プライバシー設定のインポート」をクリックし、選択可能なコネクタの中から 1 つを選択します。これにより、その特定のコネクタ用のプライバシー構成がロードされます。

以下の手順を実行して、一般プライバシー設定を定義することもできます。

- a. 「一般プライバシー設定」領域の「メッセージ・タイプ (Message type)」列で「すべて」をクリックします。ドロップダウン・リストが表示されます。メッセージ・タイプを選択します。
- b. 「セキュリティー・レベル」列で「なし」をクリックします。
- c. 「宛先」列でセルをダブルクリックして、メッセージの宛先 (例えば、System Test Connector または宛先コネクタ) を選択します。

個々のビジネス・オブジェクト用のプライバシー設定を定義するには、以下の手順を実行します。

- a. 「個々の ビジネス・オブジェクト設定」の下の「名前」列にビジネス・オブジェクトの名前を入力するか、選択可能なリストからビジネス・オブジェクトを選択します。

- b. 「**セキュリティー・レベル**」列でセルをダブルクリックし、表示されたドロップダウン・リストからオプションを選択することにより、セキュリティー・レベルを選択します。
- c. 「**宛先**」列でセルをダブルクリックして、メッセージの宛先 (例えば、System Test Connector または宛先コネクター) を選択します。

エンドツーエンド・プライバシー用に InterChange Server を構成するためのオプションの詳細については、「システム管理ガイド」を参照してください。

第 6 章 repos_copy の使用

repos_copy は、統合コンポーネントと InterChange Server リポジトリを処理するためのコマンド行インターフェースです。これを使用すると、パッケージ (統合コンポーネントの集合) をサーバー・リポジトリに配置したり、リポジトリからパッケージにコンポーネントをエクスポートすることができます。

repos_copy を実行するには、シェル・プロンプト (UNIX) または MS-DOS コマンド・プロンプト・ウィンドウ (Windows) でコマンドを入力します。インストールの結果、ユーティリティーが格納されている *ProductDir/bin* ディレクトリーがパスに指定されている必要があります。

注: repos_copy 出力ファイルには、関係およびコネクタ・アプリケーションの暗号化されたパスワードが含まれます。出力ファイルを編集してこれらのパスワードを変更しようとする、repos_copy は機能しません。

要確認: repos_copy がコンポーネントをリポジトリに配置するときには、リポジトリにのみ 配置します。メモリー内のビジネス・オブジェクト定義のテーブルには配置しません。repos_copy を使用してコンポーネントを配置する場合は、サーバーをリブートする必要はありません。ただし、repos_copy -d を使用する場合は、サーバーをリブートする必要があります。リブートしないと、メモリー内のデータ構造が更新されません。

この章を構成するセクションは次のとおりです。

- 143 ページの『Repos_copy の構文』
- 150 ページの『repos_copy の使用シナリオ』
- 157 ページの『repos_copy ファイルのロケール』

システムのバックアップの詳細については、「システム管理ガイド」を参照してください。

Repos_copy の構文

144 ページの表 10 に、repos_copy のオプションとその引き数について説明し、オプションでの大文字小文字の正しい使用法、およびオプションと引き数間にスペーシングがないことを示します。構文では、中括弧 ({}) 中のオプションは一連の必須オプションを表します。コマンド行で -u、-p、-i、-o、または -s オプションを指定しない場合、repos_copy からそれらを指定するようプロンプトが出されます。プロンプトに応じて指定しない場合、repos_copy は実行されません。大括弧 ([]) で囲まれたオプションは任意です。

注: リリース 4.2 では新規の引き数がいくつか追加され、前のリリースの一部の引き数は除去されました。これらの引き数のリストは、149 ページの『リリース 4.2 の新規引き数』および 149 ページの『リリース 4.2 で除去された引き数』を参照してください。

```

repos_copy [-sserverName] [-username] [-ppassword]
{-i [filename1] [-rrelationshipName [relationshipName2]] [[-k] [-ai] [-ar] [-arp]
[-xcompilePackage] [-vp] [-vr]]}
{-o [outfilename] [-fEntityFile] [-eEntityType:Entity1 [+EntityType:Entity2] [+...]]}
[-deep] [-summary]}
{[-d] | [-doEntityType:Entity [+EntityType:Entity2] [+...]] |
[-dfoEntityType:Entity [+EntityType:Entity2] [+...]]}
{-v}
{-vr}
{[-xCompileAll] | [-xCompileAllCollabs] | [-xCompileAllMaps] |
[-xCompileCollab:collabTemplateName [+collabTemplateName] [+...]] |
[-xCompileMap:nativeMapName [+nativeMapName] [+...]]}

```

表 10. *Repos_copy* コマンド・オプション

オプション	説明
-ai	パッケージを配置するときに見つかった、重複したオブジェクト (ビジネス・オブジェクト、マップ、関係、コラボレーション・テンプレート、コラボレーション・オブジェクト、コネクタ) を無視し、ロードしません。
-ar	パッケージを配置するときに見つかった、重複したオブジェクト (ビジネス・オブジェクト、マップ、関係、コラボレーション・テンプレート、コラボレーション・オブジェクト、コネクタ) を置換します。 注: -ar オプションは、リリース 4.2.0 以降でのみ機能します。
-arp	これは -ar オプションの対話式バージョンです。配置しているパッケージ内のコンポーネントがすでにリポジトリに存在する場合、そのコンポーネントを無視または置換するかどうか尋ねるプロンプトが <i>repos_copy</i> から出されます。 注: -arp オプションは、リリース 4.2.0 以降でのみ機能します。
-d	状態データを除く、リポジトリ内のコンポーネントを削除します。このオプションを使用すると、リポジトリ内のすべてのコンポーネントが削除されます。
-deep	従属コンポーネントをすべて組み込む場合に、-e オプションとともに使用します。-deep オプションを省略すると、-e オプションで指定されたコンポーネントのみが組み込まれます。
-dfoEntityType:Entity [+EntityType:Entity2]	このオプションは、コンポーネントに自身に依存する指示物があっても、強制的にそのコンポーネントを削除することを除けば、-do オプションと同じです。このオプションは、設計モードで稼働しているサーバーのリポジトリにのみ機能します。実動モードで稼働しているサーバーでは、未解決の依存関係および参照は許可されません。

表 10. *Repos_copy* コマンド・オプション (続き)

オプション	説明
<p><code>-doEntityType:Entity[+EntityType:Entity2]</code></p>	<p>リポジトリから削除するエンティティを指定します。エンティティ・タイプおよびキーワードのリストは、149 ページの表 11 を参照してください。オブジェクトに指示物 (そのオブジェクトに依存する他のコンポーネント) がない場合、削除が実行されます。オブジェクトに指示物がある場合、削除は失敗してメッセージが表示されます。この振る舞いは、設計モードでも実動モードでも同様です。設計モードまたは実動モードでのサーバーの開始については、「システム・インプリメンテーション・ガイド」を参照してください。</p>
<p><code>-eEntityType:Entity1[+EntityType:Entity2...]</code></p>	<p>1 つ以上の参照先最上位クラスのエンティティをエクスポートします。最上位クラスのエンティティとは、ビジネス・オブジェクト、コラボレーション・オブジェクト、コラボレーション・テンプレート、コネクタ、データベース接続プール、マップ、または関係です。149 ページの表 11 のいずれかのキーワードを指定して、ロードまたはアンロードするエンティティを識別します。</p> <p><i>EntityType</i> キーワードの後にコロン (:) とエンティティ名を続けます。複数のエンティティを指定するには「+」を使用します。<code>-o</code> オプションとともに使用すると、<code>-e</code> オプションはデータを出力ファイルにアンロードします。</p>
<p><code>-fentityFile</code></p>	<p>このオプションは、インポートするエンティティ名がファイルに保管されることを除けば、<code>-e</code> オプションと同じです。ファイルには、以下の状態でエンティティへの参照が格納されます。</p> <ul style="list-style-type: none"> • エンティティ名は、適切なエンティティ・タイプ・キーワードの後に続けます。エンティティ・タイプおよびそれらのキーワードは、149 ページの表 11 にリストされています。 • エンティティ・タイプとエンティティ名はコロンで区切ります。 • エンティティ参照ごとに改行します。 <p><code>-o</code> オプションとともに使用すると、このオプションはコンポーネントをパッケージにエクスポートします。</p>

表 10. *Repos_copy* コマンド・オプション (続き)

オプション	説明
<i>-ifilename</i>	<p>指定されたパッケージ・ファイルをリポジトリに配置します。入力ファイルの名前値を省略すると、入力ファイル名を入力するようコマンドから対話式にプロンプトが出されます。ファイルは、XML 形式のオブジェクトを含む <i>.jar</i> ファイル、またはリリース 4.2.0 より前のテキスト形式のファイルです。</p> <p><i>repos_copy</i> または System Manager により作成された <i>.jar</i> ファイルは特殊な構造を持っており、これは以降に同様のファイルを正常にインポートするために保持する必要があります。したがって、入力ファイルを手動で変更しないでください。</p>
<i>-k</i>	<p>ロード中のパッケージ・ファイルに Mercator マップが見つかったときの <i>repos_copy</i> のデフォルトの振る舞いをオーバーライドします。デフォルトでは、Mercator マップが見つかる <i>repos_copy</i> は終了します。<i>-k</i> オプションを使用すると、<i>repos_copy</i> はパッケージ・ファイル内のすべての Mercator マップをスキップし、配置プロセスを続行します。</p>
<i>-mode</i>	<p>サーバーのモードを戻します。InterChange Server モードの詳細については、「<i>WebSphere InterChange Server</i> システム・インプリメンテーション・ガイド」を参照してください。</p>
<i>-ncencoding</i>	<p>4.2.0 より前のリリースからテキスト・ベースのリポジトリ・ファイルをインポートする場合に、文字エンコード方式を指定します。</p> <p>有効な文字エンコード方式のリストは、String クラスに関する Java 資料を参照してください。</p>
<i>-outfilename</i>	<p>リポジトリ内のコンポーネントを、指定されたパッケージ・ファイルにエクスポートします。パッケージ・ファイルの名前を指定する必要があります。ファイルがすでに存在する場合、それを上書きするかどうかを尋ねるプロンプトが <i>repos_copy</i> から出されます。出力ファイルは <i>.jar</i> 形式であり、XML 形式のコンポーネント定義と、コンポーネントが持つ <i>.java</i> ソース・ファイルが含まれます。このオプションは、<i>-i</i> オプションまたは <i>-d</i> オプションとともに使用できません。また、前のリリースのようにテキスト形式でコンポーネントをエクスポートすることはできません。<i>repos_copy</i> は <i>.jar</i> 拡張子を付加しないので、出力ファイル名を指定するときにこの拡張子を指定する必要があります。</p>

表 10. Repos_copy コマンド・オプション (続き)

オプション	説明
-ppassword	-u オプションで指定したユーザー名のパスワードを指定します。パスワードには大文字小文字の区別があります。このオプションを指定しない場合、パスワードを指定するよう repos_copy からプロンプトが出されます。
-r*	このオプションは -r オプションと同様です。このオプションを使用すると、実行時スキーマを作成せずに関係定義をインポートできます。
-rrelationshipName1[:relationshipName2]	実行時スキーマを作成せずに、単一または複数の名前付きの関係定義をリポジトリにロードします。
-sserverName	repos_copy がインターフェースをとる必要がある InterChange Server インスタンスの名前を指定します。名前には大文字小文字の区別があります。サーバー名を指定しない場合、サーバー名を指定するようツールからプロンプトが出されます。
-summary	このオプションは、サーバー・リポジトリ内のコンポーネントのリストを印刷します (出力には、コンポーネントではなく「成果物」として示されます)。出力は XML 形式です。このオプションを -o オプションとともに使用すると、出力をコンソールではなくファイルに印刷することができます。
-username	InterChange Server にログインするためのユーザー名を指定します。ユーザー名を指定しない場合、ユーザー名を指定するよう repos_copy からプロンプトが出されます。
-v	repos_copy ユーティリティが実行するプログラムのバージョン番号を印刷します。
-vp	このオプションはパッケージ・ファイルを検証します。サーバーはリポジトリに照らし合わせてパッケージを検証し、パッケージ内のコンポーネント間の依存関係が解決されているか確認します。検証が失敗すると、repos_copy は欠落した依存関係のリストを印刷します。このオプションはリポジトリに対して変更を行わず、単にパッケージ・ファイルを検証します。-vp オプションを使用するときには -i オプションも使用して、検証するパッケージ・ファイルを指定する必要があります。
-vr	このオプションはリポジトリを検証します。出力メッセージには、検証が成功したか失敗したかが表示されます。検証が失敗すると、repos_copy は欠落した依存関係のリストを印刷します。

表 10. *Repos_copy* コマンド・オプション (続き)

オプション	説明
-wi	このオプションを指定すると、 <i>repos_copy</i> はコラボレーション・テンプレートまたはマップのコンパイル中に発生した警告を表示しません。コンパイル中に発生したエラーのみを表示します。これによりユーザーは、例えば使用すべきでない方法についての警告を無視できます。
-xCompileAll	リポジトリ内のすべてのコラボレーション・テンプレートおよびマップをコンパイルします。リリース 4.2 以降を使用して作成されたコラボレーション・テンプレートおよびマップのみが有効です。
-xCompileAllCollabs	リポジトリ内のすべてのコラボレーション・テンプレートをコンパイルします。リリース 4.2 以降を使用して作成されたコラボレーション・テンプレートのみが有効です。
-xCompileAllMaps	リポジトリ内のすべてのマップをコンパイルします。リリース 4.2 以降を使用して作成されたマップのみが有効です。
-xCompileCollab: <i>collabTemplateName</i> [<i>+collabTemplateName</i>]	リポジトリ内の指定されたコラボレーション・テンプレートをコンパイルします。リリース 4.2 以降を使用して作成されたコラボレーション・テンプレートのみが有効です。
-xCompileMap: <i>nativeMapName</i> [<i>+nativeMapName</i>]	リポジトリ内の指定されたマップをコンパイルします。リリース 4.2 以降を使用して作成されたマップのみが有効です。
-xCompilePackage	このオプションは、サーバーに配置されるパッケージを自動的にコンパイルします。実動モードのサーバーはすべてのパッケージを自動的にコンパイルするので、このオプションは設計モードのサーバーのみ適用します。InterChange Server のモードの詳細については、「システム・インプリメンテーション・ガイド」を参照してください。 注: このオプションは、リリース 4.2 のコンポーネントを展開する場合のみ機能します。前のリリースのコンポーネントの場合、このオプションは無視されます。
-xdi	このオプションは、System Manager で配置時プロパティをエクスポートするために使用します。
-xdn	このオプションは、 <i>repos_copy</i> ファイルの一部である配置時プロパティを無視するために使用します。
-xmsp	このオプションは、メンバーシップおよびセキュリティ情報をインポートおよびエクスポートします。これにより、役割およびセキュリティ・ポリシーを再作成する必要なしに、アップグレードできます。

表 11. 各種エンティティ・タイプのキーワード

エンティティ・タイプ	キーワード
ビジネス・オブジェクト	BusObj
コラボレーション・オブジェクト	Collaboration
コラボレーション・テンプレート	CollabTemplate
データベース接続プール	ConnectionPool
コネクタ	Connector
マップ	Map
関係	Relationship

リリース 4.3 の新規引き数

次のリストには、リリース 4.3 の `repos_copy` で提供されるすべての新規オプションが含まれています。

- `-xdi`
- `-xdn`
- `-xmsp`

リリース 4.2 の新規引き数

次のリストには、リリース 4.2 の `repos_copy` で提供されるすべての新規オプションが含まれています。

- `-dfoEntityType:Entity[+EntityType:Entity2]`
- `-doEntityType:Entity[+EntityType:Entity2]`
- `-mode`
- `-r*`
- `-summary`
- `-wi`
- `-xCompilePackage`

リリース 4.2 で除去された引き数

次の表に、リリース 4.2 で除去された `repos_copy` 引き数をリストします。

表 12. `repos_copy` から除去された引き数

除去された引き数	除去された理由
<code>[-xCompileUpdated]</code> <code>[-xCompileUpdatedCollabs]</code> <code>[-xCompileUpdatedMaps]</code>	すべてのコンパイル更新オプションは除去されました。これは、リリース 4.x より前のマップやテンプレートが、サーバーでサポートされないためです。
<code>[-xUncompress]</code>	新規のパッケージ形式では、すべての定義は専用の圧縮アルゴリズムを使用せずに Java Archive (JAR) 形式で保管されます。したがって、 <code>-xUncompress</code> 引き数は必要なくなりました。

表 12. repos_copy から除去された引き数 (続き)

除去された引き数	除去された理由
[-eProject][-w]	-e オプションでプロジェクト・タイプはサポートされなくなりました。プロジェクトは、サーバーによって保持されるのではなく、System Manager によってローカル・ファイルに保持されるようになりました。

repos_copy の使用シナリオ

このセクションでは、repos_copy を使用する多数の一般的な状況について説明します。この章は次のセクションから構成されます。

- 『repos_copy コマンドの印刷』
- 『パッケージの検証』
- 『パッケージの検証』
- 151 ページの『リポジトリへのパッケージの配置』
- 153 ページの『リポジトリとのメンバーシップのインポートおよびエクスポート』
- 154 ページの『リポジトリからのコンポーネントの削除』
- 155 ページの『パッケージへのコンポーネントのエクスポート』
- 156 ページの『リポジトリ内のコンポーネントのリストの印刷』

repos_copy コマンドの印刷

引き数を指定せずに repos_copy を実行して、コマンドとその引き数を印刷することができます。以下の例は、引き数を指定せずに実行された repos_copy と、その出力結果を示しています。

```
C:¥>repos_copy
No Command line arguments to ReposCopy were specified
Usage: repos_copy {-o[outputFile] | -i[inputFile]}
    [-sserverName] [-username] [-ppassword]
    [-ai] [-ar] [-arp] [-d] [-k] [-v]
    [-entityType:entityName1[+entityType:entityName2] -deep]
    [-fentityFileName]
    [-rrelationshipName1[:relationshipName2] ]
    [-xCompileAll] [-xCompileAllCollabs] [-xCompileAllMaps]
    [-xCompileCollab:collabTemplateName[+collabTemplateName]]
    [-xCompileMap:nativeMapName[+nativeMapName]]
    [-xcompilepackage]
    [-mode]
    [-doentityType:entityName1[+entityType:entityName2] -deep]
    [-dfoentityType:entityName1[+entityType:entityName2] -deep]
    [-summary]
    [-vp]
    [-vr]
```

パッケージの検証

パッケージをサーバーに配置する前に、コンポーネントのパッケージを検証できます。パッケージを実動モードのサーバーに配置する場合、すべての依存関係が解決されないと配置が失敗するので、この検証はとても役立ちます。System Manager では、ユーザー・プロジェクトや統合コンポーネント・ライブラリーを検証して依存

関係が問題ないか確認することはできないので、System Manager を使用して配置するときにはパッケージが有効であるか確認する唯一の方法は、配置を試行し、それが失敗したときのエラー情報を使用して依存関係を解決することになります。パッケージ内にコンポーネントが多数存在する場合、この処理は非常に時間がかかることがあります。

統合コンポーネント・ライブラリーを検証できなくても、それをパッケージ・ファイルにエクスポートして、`repos_copy` を使用しそのパッケージ・ファイルを検証することができます。

`repos_copy` を使用してパッケージ・ファイルを検証するには、`-i` オプションを使用して検証対象のパッケージ・ファイル名を指定し、`-vp` 引き数を使用してそのファイルを配置せずに検証します。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iWebSphereICS420DEVServer.jar -vp
```

`repos_copy` はパッケージの内容を検証し、依存関係が解決しているかどうかを示すメッセージを表示します。

リポジトリへのパッケージの配置

`-i` オプションを使用すると、コンポーネントのパッケージをリポジトリに配置できます。パッケージ・ファイル名を指定しない場合、ファイル名を入力するようプロンプトが出されます。

次の例は、リポジトリに配置される `WebSphereICS420DEVServer.jar` というファイルを示しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iWebSphereICS420DEVServer.jar
```

`repos_copy` ファイルを System Manager から作成した場合、配置時プロパティーが自動的に組み込まれます。このファイルがサーバーにコピーされると、これらのプロパティーは、サーバーに送信される前にコンポーネントに適用されます。

`repos_copy` ファイルがサーバーの内容をコピーして作成された場合、配置時プロパティーは組み込まれません。この場合は、`-xdi` オプションを使用して、配置時プロパティーを System Manager から配置記述子ファイルとしてエクスポートできます。配置時プロパティーがすでに `repos_copy` ファイルに存在する場合に `-xdi` オプションを使用すると、配置記述子ファイルに指定されたプロパティーで `repos_copy` ファイルに指定されたプロパティーがオーバーライドされます。

`repos_copy` コマンドの構文は、以下のとおりです。

```
reposcopy -sservername -uusername -ppassword -ireposcopyfile -xdideploymentdescriptorfile
```

`repos_copy` ファイルの配置構成は、以下のように `-xdn` オプションを使用すると、無視できます。

```
reposcopy -sservername -uusername -ppassword -ireposcopyfile -xdn
```

注: `-xdn` オプションが `-xdi` オプションと共に使用されると、`-xdi` オプションは無視されます。

配置時の重複コンポーネントの処理

一般に、リポジトリ内には、パッケージ・ファイルのコンポーネントと同じ名前のコンポーネントが存在する場合があります。この場合、リポジトリ内のコンポーネントをパッケージ・ファイルのコンポーネントに置換するかどうか決定する必要があります。-ai オプションは、重複するコンポーネントをリポジトリにロードしないよう指定します。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iCustomer.jar -ai
```

リポジトリ内のすべての重複コンポーネントを置換する場合、次の例のように -ar オプションを使用します。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iCustomerSyncInterface.jar -ar
```

-arp オプションを使用すると、リポジトリ内の重複コンポーネントを対話式に置換できます。これにより、個々の重複コンポーネントごとに置換するかどうか決定できます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iCustomerSyncInterface.jar -arp
```

注: -ar オプションと -arp オプションは、リリース 4.2.0 以降でのみ機能します。

スキーマのコンパイルおよび作成

実行時にマップとコラボレーションを実行するには、リポジトリで定義されているマップとコラボレーション・テンプレートをコンパイルする必要があります。関係が実行時に適切に機能するように、それらのスキーマを作成する必要があります。

実動モードで稼働しているサーバーにコンポーネントを配置する場合、すべてのテンプレートが自動的にコンパイルされ、すべての関係スキーマが作成されます。配置が成功するためには、マップおよびコラボレーション・テンプレートのコードが有効で、InterChange Server が関係定義の設定に指定されたデータベースと通信できる必要があります。

設計モードで稼働しているサーバーにコンポーネントを配置する場合、テンプレートは自動的にコンパイルされず、関係スキーマは自動的に作成されます。ただし、テンプレートのコンパイルに使用できるオプションがいくつかあり、関係スキーマを作成しないようにするオプションもあります。

次の例では -xCompilePackage オプションを使用し、-r オプションはいかなる形式でも使用していません。この結果、-i オプションで指定したパッケージが配置されるとき、マップおよびコラボレーション・テンプレートがコンパイルされ、関係スキーマが作成されます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iWebSphereICS420DEVServer.jar -xCompilePackage
```

配置を実行するときに、関係スキーマを作成したくない場合があります。例えば、ある環境から別の環境にパッケージを配置するとき、新規環境のデータベース・リソースを使用するために関係のプロパティを変更していない場合、関連するプロ

パティールを変更するまでスキーマを作成する必要はありません。次の例では、配置されるパッケージのすべての関係のスキーマを作成しないよう `-r*` オプションを使用しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-iWebSphereICS420DEVServer.jar -xCompilePackage -r*
```

注: `-r` オプションをアスタリスクなしで使用すると、スキーマを作成しない個々の関係の名前を指定できます。例えば `-rCustomer:Order` では、`Customer` および `Order` の関係スキーマは作成されませんが、配置されるパッケージのその他の関係スキーマは作成されます。

要確認: 配置後にマップとコラボレーション・テンプレートをコンパイルするオプションはありますが、`repos_copy` と `System Manager` のどちらを使用する場合も、配置時以外に関係スキーマを作成する方法はありません。したがって、データベース設定を変更する必要があるために配置時に関係スキーマを作成しないよう選択した場合、後で関係を再配置し、`repos_copy` によって関係スキーマを作成する必要があります。

リポジトリとのメンバーシップのインポートおよびエクスポート

メンバーシップおよびセキュリティー情報を `repos_copy` からインポートおよびエクスポートするには、`-xmsp` オプションを使用します。次にその例を示します。

```
repos_copy -sServerName -uUserName -pPassword -oOutputfile -xmsp
```

リポジトリの検証

サーバー・インスタンスを実動モードで開始するためには、リポジトリが有効な状態である必要があります。これは、サーバーがフローを正常に処理するためには、最終的にリポジトリが有効である必要があるためです。サーバー・リポジトリを検証するには、次の例のように `-vr` オプションを使用します。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull -vr
```

サーバー・リポジトリが有効である場合、`repos_copy` は次の出力をコンソールに書き込みます。

```
Validation Succeeded.All Dependencies Resolved.
```

リポジトリが有効でない場合、`repos_copy` は解決する必要がある依存関係のリストを印刷します。

リポジトリ内のコンポーネントのコンパイル

マップまたはコラボレーション・テンプレートをリポジトリに配置し、配置時にそれらをコンパイルしなかった場合、`repos_copy` を使用して後でコンパイルできます。配置には時間がかかるうえ、コンパイルによって操作にさらに時間がかかる場合があるため、コンポーネントが多数あるときにこの方法は便利です。配置が正常に完了するまで待ってからコンパイル作業を行うと、エラーが発生したときに環境の移行に長時間を費やすリスクが少なくなります。

次の例は、`-xCompileAll` オプションを使用してリポジトリ内のすべてのマップおよびコラボレーション・テンプレートをコンパイルする方法を示しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-xCompileAll
```

いずれかのタイプのすべてのコンポーネントをコンパイルするオプションがあります。

-xCompileAllCollabs を使用するとすべてのコラボレーション・テンプレートが、
-xCompileAllMaps を使用するとすべてのマップがコンパイルされます。次の例では、
-xCompileAllMaps を使用しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-xCompileAllMaps
```

いずれかのタイプのすべてのコンポーネントをコンパイルできるのと同様に、個々の
コンポーネントをコンパイルすることもできます。-xCompileCollab または
-xCompileMap オプションの後に、コロンとコラボレーション・テンプレートまたは
マップの名前を指定して、単一コンポーネントをコンパイルします。次の例では、
CustomerSync というコラボレーション・テンプレートがコンパイルされます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-xCompileCollab:CustomerSync
```

注: コンパイル・オプションは、WebSphere InterChange Server バージョン 4.2.0 以降でのみ機能します。

リポジトリからのコンポーネントの削除

repos_copy には、リポジトリ内のコンポーネントを削除するためのオプションがいくつかあります。リポジトリ全体、個々のコンポーネント、および個々のコンポーネントとそれらを参照するコンポーネントを削除できます。

注: コンポーネントを削除する場合、それらは非アクティブでなければなりません。単一のコンポーネントを削除する場合、まずそれを非アクティブにしないと削除操作は失敗します。単一のコンポーネントとそれを参照するすべてのコンポーネントを削除する場合、その単一のコンポーネントのみを非アクティブにするのではなく、それを参照するすべてのコンポーネントも非アクティブにする必要があります。リポジトリ全体は、コンポーネントがアクティブ状態のときに削除することができます。コンポーネントの状態を管理するには、System Monitor または Web ベースの System Monitor を使用します。System Monitor および Web ベースの System Monitor については、「システム管理ガイド」を参照してください。

リポジトリ全体の削除

-d オプションを使用すると、リポジトリ内のすべてのコンポーネントが削除されます。以下の例は、この構文を示しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin  
-pnull -d
```

repos_copy から、リポジトリ全体を削除するかどうかを尋ねるプロンプトが出されます。

指示物のないコンポーネントの削除

コンポーネントに指示物 (そのコンポーネントを参照し、システム内で自身の機能を実行するためにそのコンポーネントを必要とする他のコンポーネント) がない場合、個々のコンポーネントを削除できます。

-do オプションの後にエンティティ・タイプ、コロン、コンポーネントの名前を指定します。エンティティ・タイプは、149 ページの表 11 にリストされています。次の例では、**Customer** という関係が削除されます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin  
-pnull -doRelationship:Customer
```

指示物を持つコンポーネントの削除

コンポーネントに指示物 (そのコンポーネントを参照し、システム内で自身の機能を実行するためにそのコンポーネントを必要とする他のコンポーネント) がある場合、サーバーが設計モードで稼働していて、かつ特定のオプションを使用しないとコンポーネントを削除できません。

指示物がある場合の削除の強制: コンポーネントに指示物がある場合、`repos_copy` では `-do` オプションを使用してそのコンポーネントを削除できません。`-dfo` オプションを使用して、指示物があるコンポーネントを強制的に削除する必要があります。指示物があるコンポーネントを強制的に削除するとリポジトリが矛盾した状態になり、実動モードで稼働しているサーバーではこの状態が許可されないため、このオプションが機能するのは設計モードのサーバーのみになります。次の例では、`-dfo` オプションを使用して、システム内の他のコンポーネント (マップや関係など) から参照されている `Order` ビジネス・オブジェクトが削除されます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-dfoBusObj:Order
```

指示物の同時削除: 指示物があるコンポーネントを削除するもう 1 つの方法は、`-deep` オプションを使用して、指示物を一緒に削除する方法です。これにより、コンポーネントおよびそれを参照するすべてのコンポーネントが削除されます。次の例は、`-do` オプションを使用して `Customer` ビジネス・オブジェクトを削除するときの `-deep` オプション使用を示しています。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-doBusObj:Customer -deep
```

このオプションは `-dfo` オプションとは異なり、実動モードで稼働しているサーバーでサポートされます。これは、コンポーネントとともに指示物を削除するとリポジトリの有効状態が保たれるためです。ただし、この結果として多数のコンポーネントが削除される場合があることに留意してください。このオプションを使用する際には、このアクションの影響を考慮してください。

パッケージへのコンポーネントのエクスポート

`-o` オプションを使用すると、リポジトリからパッケージにコンポーネントをエクスポートできます。パッケージ・ファイルの名前を指定する必要があります。次の例のように `-o` オプションを単独で使用すると、リポジトリ全体がファイルにエクスポートされます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-oWebSphereICS420DEVServer.jar
```

-e オプションを使用して、エクスポートする個々のコンポーネントを指定できます。-e オプションを使用するには、149 ページの表 11 にリストされている適切な Entity Type キーワードと、その後にコンポーネントの名前を指定する必要があります。複数のコンポーネントを指定するには、プラス (+) 記号でそれらを連結します。次の例では、Customer ビジネス・オブジェクトおよび CustomerSync コラボレーション・テンプレートが、CustomerSyncInterface.jar というパッケージにエクスポートされます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-eBusObj:Customer+CollabTemplate:CustomerSync -oCustomerSyncInterface.jar
```

-deep オプションを使用すると、コンポーネントの依存関係も同時にエクスポートすることができます。前の例では、Customer ビジネス・オブジェクトはエクスポートされましたが、子ビジネス・オブジェクトはエクスポートされません。次の例では、-deep オプションを使用して、CustomerSync_ClarifyToSAP コラボレーション・オブジェクトおよびすべての依存関係がエクスポートされます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-eCollaboration:CustomerSync_ClarifyToSAP -oCustomerSyncInterface.jar -deep
```

特定のコンポーネントをエクスポートするときに、エンティティー・タイプ・キーワードとコンポーネント名を入力しないようにする場合、それらをテキスト・ファイルに保管して -f オプションを使用することができます。この方法は、同じコンポーネントを頻繁にエクスポートする場合にとっても便利です。次の例では、-f オプションを使用して、Components.txt というテキスト・ファイルにリストされたコンポーネントがロードされます。

```
C:\WebSphereICS420DEV>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull  
-fComponents.txt -oCustomerSyncInterface.jar -deep
```

ファイル Components.txt の内容は、次のようになります。エンティティー・タイプ・キーワードと名前の組み合わせごとに改行します。

```
BusObj:Customer  
Relationship:Customer  
CollabTemplate:CustomerSync
```

注: repos_copy と System Manager では、「依存関係」として識別するものに関しては整合性が適切ではありません。repos_copy を使用してコンポーネントの削除を試行するときに、そのコンポーネントに依存するコンポーネントがある場合、repos_copy はそれらの参照コンポーネントを依存関係としてリストします。ただし、System Manager でコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「依存関係を表示」を選択すると、ツールは選択されたコンポーネントが依存するコンポーネントをリストします。

リポジトリ内のコンポーネントのリストの印刷

repos_copy を実行してリポジトリ内のコンポーネントのリストを印刷する場合、-summary 引き数を使用できます。出力は XML 形式で作成されます。これは、コマンド行で表示するには特に便利ではありませんが、-summary 引き数と -o 引き数を結合して出力をファイルにリダイレクトすると、ファイルをブラウザまたは XML エディターで表示できます。この場合のコマンドの使用法は、次のようになります。

```
C:\>repos_copy -sWebSphereICS420DEVServer -uadmin -pnull -summary -oRepository.xml
```

repos_copy ファイルのロケール

repos_copy ユーティリティーはリポジトリからメタデータを読み取り、Unicode (UTF-8 形式) でファイルにデータを書き込みます。また、そのようなファイルを読み取り、それらをリポジトリに Unicode (基礎となるリポジトリ・データベースの指示により、UTF-8 または UCS-2) でロードします。

IBM WebSphere ICS 4.1.1 より前のバージョン・レベルで作成された repos_copy ファイルがリポジトリに正しくロードされるのは、コンポーネント・スケジュールの日時が完全に US 形式である場合のみです。(通常、これは問題になりません。repos_copy は、すべてのスケジュール日付を完全に US 形式のみで保管します。非互換性が発生するのは、repos_copy ファイルを手動で編集したことがある場合です。)

第 7 章 コネクタの構成

この章では、WebSphere Business Integration Adapters の構成手順について説明します。この章には以下のセクションがあります。

- 『Connector Configurator の使用』
- 180 ページの『コネクタの標準プロパティ』

WebSphere Business Integration Adapters の開発については、「コネクタ開発ガイド」を参照してください。

Connector Configurator の使用

この付録では、Connector Configurator を使用してアダプターの構成プロパティ値を設定する方法について説明します。

Connector Configurator を使用して次の作業を行います。

- コネクタ構成用のコネクタ固有のプロパティ・テンプレートの作成
- 構成ファイルの作成
- 構成ファイル内のプロパティの設定

ここでは、次のトピックについて説明します。

- 159 ページの『Connector Configurator の概要』
- 161 ページの『Connector Configurator の始動』
- 162 ページの『コネクタ固有のプロパティ・テンプレートの作成』
- 165 ページの『新規構成ファイルの作成』
- 169 ページの『構成ファイル・プロパティの設定』
- 179 ページの『グローバル化環境における Connector Configurator の使用』

Connector Configurator の概要

Connector Configurator では、次の統合ブローカーで使用するアダプターのコネクタ・コンポーネントを構成できます。

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker (WebSphere Message Brokers (WMQI) と総称)
- WebSphere Application Server (WAS)

使用するアダプターが DB2 Information Integrator をサポートする場合は、WMQI オプションと DB2 II 標準のプロパティを使用します (標準のプロパティに関する付録の「注」カラムを参照)。

Connector Configurator を使用して次の作業を行います。

- コネクタを構成するため、コネクタ固有のプロパティ・テンプレートを作成します。
- **コネクタ構成ファイル**を作成します。すなわち、インストールするコネクタごとに、1 つの構成ファイルを作成する必要があります。
- 構成ファイルでプロパティを設定します。
コネクタ・テンプレートで、プロパティに対して設定されているデフォルト値を変更することが必要になる場合があります。また、必要に応じて、サポートされるビジネス・オブジェクト定義と ICS の場合、コラボレーションで使用されるマップを指定するだけでなく、メッセージング、ロギング、トレース、およびデータ・ハンドラー・パラメーターも指定する必要があります。

Connector Configurator の実行モードと使用する構成ファイルのタイプは、実行する統合ブローカーによって異なります。例えば、使用している統合ブローカーが WMQI の場合、Connector Configurator を System Manager から実行するのではなく、直接実行します (161 ページの『スタンドアロン・モードでの Configurator の実行』を参照)。

コネクタ構成プロパティには、標準の構成プロパティ (すべてのコネクタが持つプロパティ) とコネクタ固有プロパティ (特定のアプリケーションまたはテクノロジーの場合にコネクタが必要とするプロパティ) の両方が含まれます。

標準プロパティはすべてのコネクタにより使用されるので、標準プロパティを新規に定義する必要はありません。ファイルを作成すると、Connector Configurator により標準プロパティがこの構成ファイルに挿入されます。ただし、Connector Configurator で各標準プロパティの値を設定する必要があります。

標準プロパティの範囲は、ブローカーと構成によって異なる可能性があります。特定のプロパティに特定の値が設定されている場合にのみ使用できるプロパティがあります。Connector Configurator の「標準のプロパティ」ウィンドウには、特定の構成で設定可能なプロパティが表示されます。

ただし**コネクタ固有プロパティ**の場合は、最初にプロパティを定義し、その値を設定する必要があります。このため、特定のアダプターのコネクタ固有プロパティのテンプレートを作成します。システム内で既にテンプレートが作成されている場合には、作成されているテンプレートを使用します。システム内でまだテンプレートが作成されていない場合には、162 ページの『新規テンプレートの作成』のステップに従い、テンプレートを新規に作成します。

UNIX でのコネクタの実行

Connector Configurator は、Windows 環境内でのみ実行されます。UNIX 環境でコネクタを実行する場合には、Windows で Connector Configurator を使用して構成ファイルを変更し、このファイルを UNIX 環境へコピーします。

Connector Configurator の一部のプロパティはディレクトリー・パスを使用します。ディレクトリー・パスは、デフォルトでは Windows 規則のディレクトリー・パスになっています。UNIX 環境で構成ファイルを使用する場合は、これらのパスが UNIX 規則に適合するようにディレクトリー・パスを変更します。ツールバーの

ドロップダウン・リストでターゲットのオペレーティング・システムを選択して、拡張検証に正しいオペレーティング・システム規則が使用されるようにします。

Connector Configurator の始動

以下の 2 種類のモードで Connector Configurator を開始および実行できます。

- スタンドアロン・モードで個別に実行
- System Manager から

スタンドアロン・モードでの Configurator の実行

ブローカーに関係なく、System Manager を実行しないで Connector Configurator を実行し、コネクタ構成ファイルを使用できます。

これを行うには、以下のステップを実行します。

- 「スタート」>「プログラム」から、「IBM WebSphere Business Integration Adapters」>「IBM WebSphere Business Integration Toolset」>「Connector Configurator」の順にクリックします。
- 「ファイル」>「新規」>「コネクタ構成」を選択します。
- 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

Connector Configurator を個別に実行して構成ファイルを生成してから、System Manager に接続してこの構成ファイルを System Manager プロジェクトに保存することもできます (168 ページの『構成ファイルの完成』を参照)。

System Manager からの Configurator の実行

System Manager から Connector Configurator を実行できます。

Connector Configurator を実行するには、以下のステップを実行します。

1. System Manager を開きます。
2. 「System Manager」ウィンドウで、「統合コンポーネント・ライブラリー」アイコンを展開し、「コネクタ」を強調表示します。
3. System Manager メニュー・バーから、「ツール」>「Connector Configurator」をクリックします。「Connector Configurator」ウィンドウが開き、「新規コネクタ」ダイアログ・ボックスが表示されます。
4. 「システム接続: Integration Broker」の隣のプルダウン・メニューをクリックします。使用しているブローカーに応じて、ICS、WebSphere Message Brokers、または WAS を選択します。

既存の構成ファイルを編集するには、以下のステップを実行します。

- 「System Manager」ウィンドウの「コネクタ」フォルダーでいずれかの構成ファイルを選択し、右クリックします。Connector Configurator が開き、この構成ファイルの統合ブローカー・タイプおよびファイル名が上部に表示されます。
- Connector Configurator から「ファイル」>「開く」を選択します。プロジェクト、またはコネクタ構成ファイルが格納されているディレクトリからコネクタ構成ファイルの名前を選択します。

- 「標準のプロパティ」タブをクリックし、この構成ファイルに含まれているプロパティを確認します。

コネクタ固有のプロパティ・テンプレートの作成

コネクタの構成ファイルを作成するには、コネクタ固有プロパティのテンプレートとシステム提供の標準プロパティが必要です。

使用するコネクタのコネクタ固有プロパティのテンプレートを新規に作成するか、または既存のコネクタ定義をテンプレートとして使用できます。

- テンプレートの新規作成については、162 ページの『新規テンプレートの作成』を参照してください。
- 既存のファイルを使用する場合には、既存のテンプレートを変更し、新しい名前でのこのテンプレートを保管します。既存のテンプレートは、`¥WebSphereAdapters¥bin¥Data¥App` ディレクトリにあります。

新規テンプレートの作成

このセクションでは、テンプレートでプロパティを作成し、プロパティの一般特性および値を定義し、プロパティ間の依存関係を指定する方法について説明します。次にそのテンプレートを保管し、新規コネクタ構成ファイルを作成するためのベースとして使用します。

Connector Configurator にテンプレートを作成するには、以下の手順を実行します。

1. 「ファイル」>「新規」>「コネクタ固有プロパティ・テンプレート」をクリックします。
2. 「コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。
 - 「新規テンプレート名を入力してください」の下の「名前」フィールドに、新規テンプレートの名前を入力します。テンプレートから新規構成ファイルを作成するためのダイアログ・ボックスを開くと、この名前が再度表示されます。
 - テンプレートに含まれているコネクタ固有のプロパティ定義を調べるには、「テンプレート名」表示でそのテンプレートの名前を選択します。そのテンプレートに含まれているプロパティ定義のリストが「テンプレートのプレビュー」に表示されます。
3. テンプレートを作成するときには、ご使用のコネクタに必要なプロパティ定義に類似したプロパティ定義が含まれている既存のテンプレートを使用できます。ご使用のコネクタで使用するコネクタ固有のプロパティが表示されるテンプレートが見つからない場合は、自分で作成する必要があります。
 - 既存のテンプレートを変更する計画の場合は、「変更する既存のテンプレートを選択してください: 検索テンプレート」の下の「テンプレート名」テーブル内のリストからテンプレート名を選択します。
 - このテーブルには、現在使用可能なすべてのテンプレートの名前が表示されます。テンプレートを検索することもできます。

一般特性の指定: 「次へ」をクリックしてテンプレートを選択すると、「プロパティ: コネクタ固有プロパティ・テンプレート」ダイアログ・ボックスが表示

されます。このダイアログ・ボックスには、定義済みプロパティの「一般」特性のタブと「値」の制限のタブがあります。「一般」表示には以下のフィールドがあります。

- **一般:**
 - プロパティ・タイプ
 - プロパティ・サブタイプ
 - 更新されたメソッド
 - 説明
- **フラグ**
 - 標準フラグ
- **カスタム・フラグ**
 - フラグ

プロパティ・サブタイプは、**プロパティ・タイプ**がストリングの場合に検索できます。これは、構成ファイルを保管するときの構文検査を指定するオプションの値です。デフォルトは空白で、プロパティはサブタイプ化されないことを意味します。

プロパティの一般特性の選択を終えたら、「**値**」タブをクリックします。

値の指定: 「**値**」タブを使用すると、プロパティの最大長、最大複数値、デフォルト値、または値の範囲を設定できます。編集可能な値も許可されます。これを行うには、以下のステップを実行します。

1. 「**値**」タブをクリックします。「一般」の表示パネルに代わり、「**値**」の表示パネルが表示されます。
2. 「**プロパティを編集**」表示でプロパティの名前を選択します。
3. 「**最大長**」および「**最大複数値**」のフィールドに、値を入力します。

新規のプロパティ値を作成するには、以下の手順を実行します。

1. 「**値**」列見出しの左の正方形を右マウス・ボタンでクリックします。
2. ポップアップ・メニューから、「**追加**」を選択して「**プロパティ値**」ダイアログ・ボックスを表示します。このダイアログ・ボックスではプロパティのタイプに応じて、値だけを入力できる場合と、値と範囲の両方を入力できる場合があります。
3. 新規のプロパティ値を入力して、「**OK**」をクリックします。右方の「**値**」パネルに値が表示されます。

「**値**」パネルには、3つの列のテーブルが表示されます。

「**値**」の列には、「**プロパティ値**」ダイアログ・ボックスで入力した値と、以前に作成した値が表示されます。

「**デフォルト値**」の列では、値のいずれかをデフォルトとして指定することができます。

「**値の範囲**」の列には、「**プロパティ値**」ダイアログ・ボックスで入力した範囲が表示されます。

値を作成し、グリッドに表示されたら、表の表示で編集できます。

表の既存の既存の値を変更するには、行番号をクリックして行全体を選択します。次に「値」フィールドを右マウス・ボタンをクリックし、「編集値 (Edit Value)」をクリックします。

依存関係の設定: 「一般」タブと「値」タブで変更を行ったら、「次へ」をクリックします。「依存関係: コネクター固有プロパティ・テンプレート」ダイアログ・ボックスが表示されます。

依存プロパティは、別のプロパティの値が特定の条件に合致する場合にのみ、テンプレートに組み込まれて、構成ファイルで使用されるプロパティです。例えば、テンプレートに `PollQuantity` が表示されるのは、トランスポート機構が `JMS` であり、`DuplicateEventElimination` が `True` に設定されている場合のみです。プロパティを依存プロパティとして指定し、依存する条件を設定するには、以下のステップを実行します。

1. 「使用可能なプロパティ」表示で、依存プロパティとして指定するプロパティを選択します。
2. 「プロパティを選択」フィールドで、ドロップダウン・メニューを使用して、条件値を持たせるプロパティを選択します。
3. 「条件演算子」フィールドで以下のいずれかを選択します。

`==` (等しい)

`!=` (等しくない)

`>` (より大)

`<` (より小)

`>=` (以上)

`<=` (以下)

4. 「条件値」フィールドで、依存プロパティをテンプレートに組み込むために必要な値を入力します。
5. 「使用可能なプロパティ」表示で依存プロパティを強調表示させて矢印をクリックし、「依存プロパティ」表示に移動させます。
6. 「完了」をクリックします。Connector Configurator により、XML 文書として入力した情報が、Connector Configurator がインストールされている `%bin` ディレクトリーの `%data%app` の下に保管されます。

パス名の設定: パス名の設定には、以下のような一般規則があります。

- Windows および UNIX でのファイル名の最大長は 255 文字です。
- Windows では、絶対パス名は [ドライブ:][ディレクトリー]¥ファイル名の形式でなければなりません。例えば、
`C:\WebSphereAdapters\%bin%\Data\Std\StdConnProps.xml` です。
- UNIX では、先頭の文字は / でなければなりません。
- キュー名は、先頭または中間にスペースがあってはなりません。

新規構成ファイルの作成

構成ファイルを新規に作成する場合は、その名前を指定して、統合ブローカーを選択する必要があります。

ファイルの拡張検証に関するオペレーティング・システムも選択します。ツールバーには、「ターゲット・システム」と呼ばれるドロップ・リストがあり、プロパティの拡張検証に関するターゲットのオペレーティング・システムを選択できます。選択可能なオプションは、「Windows」、「UNIX」、「その他」(Windows または UNIX でない場合)、および「なし (拡張検証なし)」(拡張検証をオフに切り替え) です。始動時のデフォルトは、「Windows」です。

Connector Configurator を始動するには、以下の手順を実行します。

- 「System Manager」ウィンドウで、「ツール」メニューから「**Connector Configurator**」を選択します。Connector Configurator が開きます。
- スタンドアロン・モードで、Connector Configurator を起動します。

構成ファイルの拡張検証に関するオペレーティング・システムを設定するには、以下の手順を実行します。

- メニュー・バーの「ターゲット・システム:」ドロップ・リストをプルダウンします。
- 実行するオペレーティング・システムを選択します。

その後、「ファイル」>「新規」>「コネクタ構成」を選択します。「新規コネクタ」ウィンドウで、新規コネクタの名前を入力します。

統合ブローカーを選択する必要もあります。選択したブローカーによって、構成ファイルに記述されるプロパティが決まります。ブローカーを選択するには、以下のステップを実行します。

- 「**Integration Broker**」フィールドで、ICS 接続、WebSphere Message Broker 接続、または WAS 接続のいずれかを選択します。
- この章で後述する説明に従って「新規コネクタ」ウィンドウの残りのフィールドに入力します。

コネクタ固有テンプレートからの構成ファイルの作成

コネクタ固有のテンプレートを作成すると、テンプレートを使用して構成ファイルを作成できます。

1. メニュー・バーの「ターゲット・システム:」ドロップ・リストを使用して、構成ファイルの拡張検証に関するオペレーティング・システムを設定します (上記の『新規構成ファイルの作成』を参照)。
2. 「ファイル」>「新規」>「コネクタ構成」をクリックします。
3. 以下のフィールドを含む「新規コネクタ」ダイアログ・ボックス表示されず。

- **名前**

コネクタの名前を入力します。名前では大文字と小文字が区別されます。入力する名前は、システムにインストールされているコネクタのファイル名に対応した一意の名前でなければなりません。

要確認: Connector Configurator では、入力された名前のスペルはチェックされません。名前が正しいかどうかを確認してください。

- システム接続

ICS、WebSphere Message Broker、または WAS のいずれかをクリックします。

- 「コネクタ固有プロパティ・テンプレート」を選択します。

コネクタ用に設計したテンプレートの名前を入力します。「テンプレート名」表示に、使用可能なテンプレートが表示されます。「テンプレート名」表示で名前を選択すると、「プロパティ・テンプレートのプレビュー」表示に、そのテンプレートで定義されているコネクタ固有のプロパティが表示されます。

使用するテンプレートを選択し、「OK」をクリックします。

4. 構成しているコネクタの構成画面が表示されます。タイトル・バーに統合ブローカーとコネクタの名前が表示されます。ここですべてのフィールドに値を入力して定義を完了するか、ファイルを保管して後からフィールドに値を入力するかを選択できます。
5. ファイルを保管するには、「ファイル」>「保管」>「ファイルに」をクリックするか、「ファイル」>「保管」>「プロジェクトに」をクリックします。プロジェクトに保管するには、System Manager が実行中でなければなりません。ファイルとして保管する場合は、「ファイル・コネクタを保管」ダイアログ・ボックスが表示されます。*.cfg をファイル・タイプとして選択し、「ファイル名」フィールド内に名前が正しいスペル (大文字と小文字の区別を含む) で表示されていることを確認してから、ファイルを保管するディレクトリーにナビゲートし、「保管」をクリックします。Connector Configurator の下部にあるメッセージ・パネル内の状況表示で、構成ファイルが正常に作成されたことが示されます。

要確認: ここで設定するディレクトリー・パスおよび名前は、コネクタの始動ファイルで指定するコネクタ構成ファイルのパスおよび名前に一致している必要があります。

6. この章で後述する手順に従って、「Connector Configurator」ウィンドウの各タブにあるフィールドに値を入力し、コネクタ定義を完了します。

既存ファイルの使用

以下の 1 つ以上の形式で既存ファイルを使用可能にする場合があります。

- コネクタ定義ファイル。

コネクタ定義ファイルは、特定のコネクタのプロパティと適用されるデフォルト値をリストするテキスト・ファイルです。一部のコネクタでは、提供パッケージの `¥repository` ディレクトリーにこのようなファイルが格納されています (通常、このファイルの拡張子は .txt で、例えば、XML コネクタの場合は CN_XML.txt です)。

- ICS リポジトリ・ファイル。
コネクタの以前の ICS の実装で使用した定義は、そのコネクタの構成で使用したリポジトリ・ファイルで使用可能です。このようなファイルは、通常、拡張子は .in または .out です。
- コネクタの以前の構成ファイル。
このようなファイルは、通常、拡張子は *.cfg です。

これらのファイル・ソースにはコネクタのほとんどまたはすべてのコネクタ固有プロパティが含まれている場合がありますが、コネクタの構成ファイルは、ファイルを開いてプロパティを設定するまで完成しません (詳細については後述します)。

既存ファイルを使用してコネクタを構成するには、Connector Configurator でそのファイルを開き、構成を修正し、そのファイルを再度保管する必要があります。

.txt、.cfg、または *.in ファイルをディレクトリから開くには、以下の手順を実行します。

1. Connector Configurator 内で、「ファイル」>「開く」>「ファイルから」をクリックします。
2. 「ファイル・コネクタを開く」ダイアログ・ボックス内で、以下のいずれかのファイル・タイプを選択して、使用可能なファイルを調べます。
 - 構成 (*.cfg)
 - ICS リポジトリ (*.in、*.out)

ICS 環境でコネクタを構成するためにリポジトリ・ファイルが使用されている場合は、このオプションを選択します。リポジトリ・ファイルに複数のコネクタ定義が含まれている場合は、ファイルを開くとすべての定義が表示されます。

- すべてのファイル (*.*)

コネクタのアダプター・パッケージに *.txt ファイルが含まれている場合、または別の拡張子の定義ファイルが使用可能な場合は、このオプションを選択します。

3. ディレクトリ表示内で、適切なコネクタ定義ファイルへ移動し、ファイルを選択し、「開く」をクリックします。

System Manager プロジェクトからコネクタ構成を開くには、以下の手順を実行します。

1. System Manager を始動します。System Manager から構成を開いたり System Manager に構成を保管したりするには、System Manager を始動しておく必要があります。
2. Connector Configurator を始動します。
3. 「ファイル」>「開く」>「プロジェクトから」をクリックします。

構成ファイルの完成

構成ファイルを開くか、プロジェクトからコネクタを開くと、「Connector Configurator」ウィンドウに構成画面が表示されます。この画面には、現在の属性と値が表示されます。

構成画面のタイトルには、ファイル内で指定された統合ブローカーとコネクタの名前が表示されます。正しいブローカーが設定されていることを確認してください。正しいブローカーが設定されていない場合、コネクタを構成する前にブローカー値を変更してください。これを行うには、以下のステップを実行します。

1. 「標準のプロパティ」タブで、BrokerType プロパティの値フィールドを選択します。ドロップダウン・メニューで、値 ICS、WMQI、または WAS を選択します。
2. 選択したブローカーに関連付けられているコネクタのプロパティが「標準のプロパティ」タブに表示されます。テーブルに、**プロパティ名**、**値**、**タイプ**、**サブタイプ** (タイプがstringの場合)、**説明**、および**更新メソッド**が表示されます。
3. ここでファイルを保管するか、または 172 ページの『サポートされているビジネス・オブジェクト定義の指定』の説明に従い残りの構成フィールドに値を入力することができます。
4. 構成が完了したら、「ファイル」>「保管」>「プロジェクトに」を選択するか、または「ファイル」>「保管」>「ファイルに」を選択します。

ファイルに保管する場合は、*.cfg を拡張子として選択し、ファイルの正しい格納場所を選択して、「保管」をクリックします。

複数のコネクタ構成を開いている場合、構成をすべてファイルに保管するには「すべてファイルに保管」を選択し、コネクタ構成をすべて System Manager プロジェクトに保管するには「すべてプロジェクトに保管」をクリックします。

構成ファイルを作成する前に、プロパティの拡張検証に関するターゲットのオペレーティング・システムを選択できる「ターゲット・システム」ドロップ・リストを使用しました。

Connector Configurator では、ファイルを保管する前に、必須の標準プロパティすべてに値が設定されているかどうかを確認されます。必須の標準プロパティに値が設定されていない場合、Connector Configurator は、検証が失敗したというメッセージを表示します。構成ファイルを保管するには、そのプロパティの値を指定する必要があります。

「ターゲット・システム」ドロップ・リストから値「Windows」、「UNIX」、または「その他」を選択して拡張検証機能の使用を選択した場合、プロパティのサブタイプとタイプが検証され、検証に失敗すると警告メッセージが表示されます。

構成ファイル・プロパティの設定

新規のコネクター構成ファイルを作成して名前を付けるとき、または既存のコネクター構成ファイルを開くときには、Connector Configurator によって構成画面が表示されます。構成画面には、必要な構成値のカテゴリに対応する複数のタブがあります。

Connector Configurator では、すべてのブローカーで実行されているコネクターで、以下のカテゴリのプロパティに値が設定されている必要があります。

- 標準のプロパティ
- コネクター固有のプロパティ
- サポートされるビジネス・オブジェクト
- トレース/ログ・ファイルの値
- データ・ハンドラー (保証付きイベント・デリバリーで JMS メッセージングを使用するコネクターにのみ適用可能)

注: JMS メッセージングを使用するコネクターの場合、データをビジネス・オブジェクトに変換するデータ・ハンドラーの構成用として、追加のカテゴリが表示される場合があります。

ICS で実行されているコネクターの場合、以下のプロパティの値も設定されている必要があります。

- 関連付けられたマップ
- リソース
- メッセージング (適用可能な場合)
- セキュリティ

要確認: Connector Configurator は、英語でも英語以外の文字セットのプロパティ値でも受け入れます。ただし、標準のプロパティ、コネクター固有プロパティ、およびサポートされているビジネス・オブジェクトの名前では、英語文字セットのみを使用する必要があります。

標準のプロパティとコネクター固有プロパティの相違点は以下のとおりです。

- コネクターの標準プロパティは、コネクターのアプリケーション固有のコンポーネントおよびブローカー・コンポーネントによって共有されます。すべてのコネクターが同じ標準プロパティのセットを使用します。これらのプロパティの説明は、各アダプター・ガイドの付録 A にあります。変更できるのはこれらの値の一部のみです。
- アプリケーション固有のプロパティは、コネクターのアプリケーション固有コンポーネント (アプリケーションと直接対話するコンポーネント) のみに適用されます。各コネクターには、そのアプリケーションだけで使用されるアプリケーション固有のプロパティがあります。これらのプロパティには、デフォルト値を持つものと持たないものがあります。デフォルト値の一部は変更できます。各アダプター・ガイドのインストールおよび構成の章に、アプリケーション固有のプロパティおよび推奨値が記述されています。

「標準プロパティ」と「コネクター固有プロパティ」のフィールドは、どのフィールドが構成可能であることを示すために色分けされています。

- 背景がグレーのフィールドは、標準のプロパティを示します。プロパティの値は変更できますが、名前の変更や除去を行うことはできません。
- 背景が白のフィールドは、アプリケーション固有のプロパティを示します。これらのプロパティは、アプリケーションまたはコネクタの特定のニーズによって異なります。値を変更し、またこれらのプロパティを削除できます。
- 「値」フィールドは構成できます。
- 「更新メソッド」フィールドが、プロパティごとに表示されます。これは、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。この設定は構成できません。

標準コネクタ・プロパティの設定

標準のプロパティの値を変更するには、以下の手順を実行します。

1. 値を設定するフィールド内でクリックします。
2. 値を入力するか、ドロップダウン・メニューが表示された場合にはメニューから値を選択します。

注: プロパティのタイプがstringの場合、「サブタイプ」列にサブタイプ値を持つことができます。このサブタイプは、プロパティの拡張検証に使用されます。

3. 標準のプロパティの値をすべて入力後、以下のいずれかを実行することができます。
 - 変更内容を破棄し、元の値を保持したままで Connector Configurator を終了するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「いいえ」をクリックします。
 - Connector Configurator 内の他のカテゴリの値を入力するには、そのカテゴリのタブを選択します。「標準のプロパティ」(またはその他のカテゴリ) で入力した値は、次のカテゴリに移動しても保持されます。ウィンドウを閉じると、すべてのカテゴリで入力した値を一括して保管するかまたは破棄するかを確認するプロンプトが出されます。
 - 修正した値を保管するには、「ファイル」>「終了」をクリックし (またはウィンドウを閉じ)、変更内容を保管するかどうかを確認するプロンプトが出されたら「はい」をクリックします。「ファイル」メニューまたはツールバーから「保管」>「ファイルに」をクリックする方法もあります。

特定の標準プロパティの詳細情報を表示するには、「標準のプロパティ」タブ付きシートでそのプロパティの「説明」列の項目を左マウス・ボタンでクリックします。全般ヘルプがインストールされている場合は、右方に矢印ボタンが表示されます。ボタンをクリックすると、「ヘルプ」ウィンドウが開いて標準のプロパティの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティの全般ヘルプが見つかりませんでした。

全般ヘルプ・ファイルは、インストールされている場合、`<ProductDir>\bin\Data\Std\Help\<RegionalSetting>` にあります。

コネクタ固有構成プロパティの設定

コネクタ固有の構成プロパティの場合、プロパティ名の追加または変更、値の構成、プロパティの削除、およびプロパティの暗号化が可能です。プロパティのデフォルトの長さは 255 文字です。

1. グリッドの左上端の部分で右マウス・ボタンをクリックします。ポップアップ・メニュー・バーが表示されます。プロパティを追加するときは「追加」をクリックします。子プロパティを追加するには、親の行番号で右マウス・ボタンをクリックし、「子を追加」をクリックします。
2. プロパティまたは子プロパティの値を入力します。

注: プロパティのタイプがストリングの場合、「サブタイプ」ドロップ・リストからサブタイプを選択できます。このサブタイプは、プロパティの拡張検証に使用されます。

3. プロパティを暗号化するには、「暗号化」ボックスを選択します。
4. 特定のプロパティの詳細情報を表示するには、そのプロパティの「説明」列の項目を左マウス・ボタンでクリックします。全般ヘルプがインストールされている場合は、ホット・ボタンが表示されます。ホット・ボタンをクリックすると、「ヘルプ」ウィンドウが開いて標準のプロパティの詳細が表示されます。

注: ホット・ボタンが表示されない場合、そのプロパティの全般ヘルプが見つかりませんでした。

5. 170 ページの『標準コネクタ・プロパティの設定』の説明に従い、変更内容を保管するかまたは破棄するかを選択します。

全般ヘルプ・ファイルがインストールされていて、AdapterHelpName プロパティがブランクの場合、Connector Configurator は

`<ProductDir>%bin%\Data\App\Help\<RegionalSetting>` に存在するアダプター固有の全般ヘルプ・ファイルをポイントします。あるいは、Connector Configurator は `<ProductDir>%bin%\Data\App\Help\<AdapterHelpName>\<RegionalSetting>` に存在するアダプター固有の全般ヘルプ・ファイルをポイントします。標準のプロパティに関する付録に説明されている AdapterHelpName プロパティを参照してください。

各プロパティごとに表示される「更新メソッド」は、変更された値をアクティブにするためにコンポーネントまたはエージェントの再始動が必要かどうかを示します。

要確認: 事前設定のアプリケーション固有のコネクタ・プロパティ名を変更すると、コネクタに障害が発生することがあります。コネクタをアプリケーションに接続したり正常に実行したりするために、特定のプロパティ名が必要な場合があります。

コネクタ・プロパティの暗号化: 「コネクタ固有のプロパティ」ウィンドウの「暗号化」チェック・ボックスにチェックマークを付けると、アプリケーション固有のプロパティを暗号化することができます。値の暗号化を解除するには、「暗号化」チェック・ボックスをクリックしてチェックマークを外し、「検証」ダイアログ・ボックスに正しい値を入力し、「OK」をクリックします。入力された値が正しい場合は、暗号化解除された値が表示されます。

各プロパティとそのデフォルト値のリストおよび説明は、各コネクターのアダプター・ユーザーズ・ガイドにあります。

プロパティに複数の値がある場合には、プロパティの最初の値に「暗号化」チェック・ボックスが表示されます。「暗号化」を選択すると、そのプロパティのすべての値が暗号化されます。プロパティの複数の値を暗号化解除するには、そのプロパティの最初の値の「暗号化」チェック・ボックスをクリックしてチェックマークを外してから、「検証」ダイアログ・ボックスで新規の値を入力します。入力した値が一致した場合は、複数值がすべて暗号化解除されます。

更新メソッド: 付録『標準のプロパティ』の 181 ページの『構成プロパティ値の概要』にある更新メソッドの説明を参照してください。

サポートされているビジネス・オブジェクト定義の指定

コネクターで使用するビジネス・オブジェクトを指定するには、Connector Configurator の「サポートされているビジネス・オブジェクト」タブを使用します。汎用ビジネス・オブジェクトと、アプリケーション固有のビジネス・オブジェクトの両方を指定する必要があり、またそれらのビジネス・オブジェクト間のマップの関連を指定することが必要です。

注: 一部のコネクターでは、使用されるアプリケーションでイベント通知または (メタオブジェクトを使用して) 追加の構成を行うために、サポートされているビジネス・オブジェクトとして特定のビジネス・オブジェクトを指定することが必要です。詳しくは、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

ご使用のブローカーが ICS の場合: ビジネス・オブジェクト定義がコネクターでサポートされることを指定する場合や、既存のビジネス・オブジェクト定義のサポート設定を変更する場合は、「サポートされているビジネス・オブジェクト」タブをクリックし、以下のフィールドを使用してください。

ビジネス・オブジェクト名: ビジネス・オブジェクト定義がコネクターによってサポートされることを指定するには、System Manager を実行し、以下の手順を実行します。

1. 「ビジネス・オブジェクト名」リストで空のフィールドをクリックします。ドロップ・リストに、System Manager プロジェクトに存在するすべてのビジネス・オブジェクト定義が表示されます。
2. ビジネス・オブジェクトを 1 つクリックして追加します。
3. ビジネス・オブジェクトの「エージェント・サポート」(以下で説明) を設定します。
4. 「Connector Configurator」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。追加のビジネス・オブジェクト定義に対して指定したサポートが含まれる変更済みのコネクター定義が System Manager の ICL (統合コンポーネント・ライブラリー) プロジェクトに保管されます。

サポート対象のリストからビジネス・オブジェクトを削除するには、次の手順に従います。

1. ビジネス・オブジェクト・フィールドを選択するため、そのビジネス・オブジェクトの左側の番号をクリックします。

2. 「Connector Configurator」ウィンドウの「編集」メニューから、「行を削除」をクリックします。リスト表示からそのビジネス・オブジェクトが除去されます。
3. 「ファイル」メニューから、「プロジェクトの保管」をクリックします。

サポートされるリストからビジネス・オブジェクトを削除すると、コネクタ定義が変更され、削除されたビジネス・オブジェクトはコネクタのこのインプリメンテーションで使用不可になります。コネクタのコードに影響したり、そのビジネス・オブジェクト定義そのものが System Manager から削除されることはありません。

エージェント・サポート: ビジネス・オブジェクトがエージェント・サポートを備えている場合、システムは、コネクタ・エージェントを介してアプリケーションにデータを配布する際にそのビジネス・オブジェクトの使用を試みます。

通常、コネクタのアプリケーション固有のビジネス・オブジェクトは、そのコネクタのエージェントによってサポートされますが、汎用ビジネス・オブジェクトはサポートされません。

ビジネス・オブジェクトがコネクタ・エージェントによってサポートされるよう指定するには、「エージェント・サポート」ボックスにチェックマークを付けます。「Connector Configurator」ウィンドウでは「エージェント・サポート」の選択の妥当性は検査されません。

最大トランザクション・レベル: コネクタの最大トランザクション・レベルとは、コネクタによってサポートされる最高のトランザクション・レベルのことです。

ほとんどのコネクタの場合、選択可能な項目は「最大限の努力」のみです。

トランザクション・レベルの変更を有効にするには、サーバーを再始動する必要があります。

ご使用のブローカーが WebSphere Message Broker の場合: スタンドアロン・モードで作業している (System Manager に接続していない) 場合、手でビジネス・オブジェクト名を入力する必要があります。

System Manager を実行している場合、「サポートされているビジネス・オブジェクト」タブの「ビジネス・オブジェクト名」列の下にある空のボックスを選択できます。コンボ・ボックスが表示され、コネクタが属する統合コンポーネント・ライブラリー・プロジェクトから選択可能なビジネス・オブジェクトのリストが示されます。リストから必要なビジネス・オブジェクトを選択します。

「メッセージ・セット ID」は、WebSphere Business Integration Message Broker 5.0 のオプションのフィールドです。この ID が提供される場合、一意である必要はありません。ただし、WebSphere MQ Integrator および Integrator Broker 2.1 の場合は、一意の ID を提供する必要があります。

ご使用のブローカーが WAS の場合: 使用するブローカー・タイプとして WebSphere Application Server を選択した場合、Connector Configurator にメッセー

ジ・セット ID は必要ありません。「サポートされているビジネス・オブジェクト」タブには、サポートされるビジネス・オブジェクトの「ビジネス・オブジェクト名」列のみが表示されます。

スタンドアロン・モードで作業している (System Manager に接続していない) 場合、手動でビジネス・オブジェクト名を入力する必要があります。

System Manager を実行している場合、「サポートされているビジネス・オブジェクト」タブの「ビジネス・オブジェクト名」列の下にある空のボックスを選択できます。コンボ・ボックスが表示され、コネクターが属する統合コンポーネント・ライブラリー・プロジェクトから選択可能なビジネス・オブジェクトのリストが示されます。このリストから必要なビジネス・オブジェクトを選択します。

関連付けられたマップ (ICS)

各コネクターは、現在 WebSphere InterChange Server でアクティブなビジネス・オブジェクト定義、およびそれらの関連付けられたマップのリストをサポートします。このリストは、「関連付けられたマップ」タブを選択すると表示されます。

ビジネス・オブジェクトのリストには、エージェントによってサポートされるアプリケーション固有のビジネス・オブジェクトと、コントローラーによってサブスクライブするコラボレーションへ送信される、対応する汎用オブジェクトが含まれます。マップの関連付けによって、アプリケーション固有のビジネス・オブジェクトから汎用ビジネス・オブジェクトへの変換、または汎用オブジェクトからアプリケーション固有のビジネス・オブジェクトへの変換の際に、どのマップが使用されるかが決定されます。

特定のソース・ビジネス・オブジェクトおよび宛先ビジネス・オブジェクトに対して固有に定義されているマップを使用する場合、そのマップは表示したときすでに適切なビジネス・オブジェクトに関連付けられているので、これらを変更する必要はありません (変更することはできません)。

1 つのサポートされているビジネス・オブジェクトによって複数のマップが利用できる場合、ビジネス・オブジェクトとそれによって使用されるマップを明示的にバインドする必要があります。

「関連付けられたマップ」タブには以下のフィールドが表示されます。

• ビジネス・オブジェクト名

これらは、「サポートされているビジネス・オブジェクト」タブで指定した、このコネクターでサポートされるビジネス・オブジェクトです。「サポートされているビジネス・オブジェクト」タブでビジネス・オブジェクトを追加指定した場合、その内容は、「Connector Configurator」ウィンドウの「ファイル」メニューから「プロジェクトに保管」を選択して、変更を保管した後に、このリストに反映されます。

• 関連付けられたマップ

この表示では、コネクターによってサポートされているビジネス・オブジェクトとともに使用できるようにシステムにインストールされているすべてのマップが表示されます。各マップのソース・ビジネス・オブジェクトは、「ビジネス・オブジェクト名」表示でマップ名の左側に表示されます。

• 明示的バインド

場合によっては、関連付けられたマップを明示的にバインドする必要があります。

特定のサポート対象ビジネス・オブジェクトに対して複数のマップが存在する場合にのみ、明示的なバインディングが必要となります。ICS は、ブート時、各コネクタでサポートされるそれぞれのビジネス・オブジェクトにマップを自動的にバインドしようとします。複数のマップによって同じビジネス・オブジェクトの入力が受け入れられる場合、サーバーは他のマップのスーパーセットであるマップを 1 つ見つけてバインドしようとします。

他のマップのスーパーセットであるマップがないと、サーバーは、ビジネス・オブジェクトを単一のマップにバインドすることができないため、バインディングを明示的に設定することが必要になります。

明示的にマップをバインドするには、次の手順に従います。

1. 「明示的 (Explicit)」列で、バインドするマップのチェック・ボックスにチェックマークを付けます。
2. ビジネス・オブジェクトに関連付けるマップを選択します。
3. 「Connector Configurator」ウィンドウの「ファイル」メニューで、「プロジェクトに保管」をクリックします。
4. プロジェクトを ICS に配置します。
5. 変更を有効にするため、サーバーをリブートします。

リソース (ICS)

「リソース」タブでは、コネクタ・エージェントが、コネクタ・エージェント並列処理を使用して同時に複数のプロセスを処理するかどうか、またどの程度処理するかを決定する値を設定できます。

すべてのコネクタがこの機能をサポートしているわけではありません。複数のプロセスを使用するよりも複数のスレッドを使用する方が通常は効率的であるため、Java でマルチスレッドとして設計されたコネクタ・エージェントを実行している場合、この機能を使用することはお勧めできません。

メッセージング (ICS)

「メッセージング」タブを使用すると、メッセージング・プロパティを構成できます。メッセージング・プロパティは、DeliveryTransport 標準プロパティの値として MQ を設定し、ブローカー・タイプとして ICS を設定した場合にのみ、使用可能です。これらのプロパティは、コネクタによるキューの使用方法に影響を及ぼします。

メッセージング・キューの検証: メッセージング・キューを検証するには、事前に以下を実行する必要があります。

- WebSphere MQ Series がインストールされていることを確認する。
- ホスト・マシン上のチャネルおよびポートを指定してメッセージ・キューを作成する。
- ホスト・マシンへの接続をセットアップする。

キューを検証するには、「メッセージング」タブの「メッセージング・タイプ」と「ホスト名」フィールドの右方の「検証」ボタンを使用します。

セキュリティ (ICS)

Connector Configurator の「セキュリティ」タブを使用すると、さまざまなプライバシー・レベルのメッセージを設定できます。この機能は、DeliveryTransport プロパティーが「JMS」に設定されている場合にのみ使用できます。

デフォルトでは、プライバシーはオフになっています。使用可能にするには、「プライバシー」ボックスにチェック・マークを付けます。

鍵ストア・ターゲット・システムの絶対パス名は、以下のとおりです。

- Windows の場合:
`<ProductDir>%connectors%security%<connectorname>.jks`
- UNIX の場合:
`opt/IBM/WebSphereAdapters/connectors/security/<connectorname>.jks`

このパスおよびファイルは、コネクタを始動する計画のシステム (つまり、ターゲット・システム) に存在する必要があります。

ターゲット・システムが現在実行しているシステムの場合のみ、右方の「参照」ボタンを使用できます。「プライバシー」が使用可能でなく、メニュー・バーの「ターゲット・システム」が Windows に設定されていない場合は、グレー表示されています。

3 つのメッセージ・カテゴリー (全メッセージ、全管理メッセージ、全ビジネス・オブジェクト・メッセージ) の「メッセージのプライバシー・レベル」は次のように設定できます。

- “” はデフォルトです。メッセージ・カテゴリーのプライバシー・レベルが設定されていない場合に使用されます。
- なし
デフォルトとは異なります。メッセージ・カテゴリーに、なしのプライバシー・レベルを意図的に設定する場合に使用します。
- 完全性
- プライバシー
- 完全性およびプライバシー

鍵の保守機能を使用すると、サーバーおよびアダプターの公開鍵を生成、インポート、およびエクスポートできます。

- 「鍵の生成」を選択すると、鍵を生成する keytool のデフォルトが設定された「鍵の生成」ダイアログ・ボックスが表示されます。
- 鍵ストア値のデフォルトは、「セキュリティ」タブの「鍵ストア・ターゲット・システムの絶対パス名」に入力した値に設定されます。
- 「OK」を選択すると、入力が検証され、鍵証明書が生成されて、出力が Connector Configurator のログ・ウィンドウに送信されます。

証明書をアダプター鍵ストアにインポートするには、事前にサーバー鍵ストアからエクスポートする必要があります。「**アダプター公開鍵のエクスポート**」を選択すると、「アダプター公開鍵のエクスポート」ダイアログ・ボックスが表示されます。

- エクスポート証明書は、デフォルトでは鍵ストアと同じ値に設定されています。ただし、ファイル拡張子は `<filename>.cer` です。

「**サーバー公開鍵のインポート**」を選択すると、「サーバー公開鍵のインポート」ダイアログ・ボックスが表示されます。

- インポート証明書は、デフォルトでは `<ProductDir>%bin%ics.cer` に設定されています (ファイルがシステムに存在する場合)。
- インポート証明書関連はサーバー名でなければなりません。サーバーが登録されている場合は、ドロップ・リストから選択できます。

アダプター・アクセス制御機能は、`DeliveryTransport` の値が `IDL` の場合にのみ使用できます。デフォルトでは、アダプターはゲスト ID でログインされます。「**ゲスト ID の使用**」ボックスにチェック・マークが付けられていない場合、「**アダプター ID**」と「**アダプター・パスワード**」フィールドが使用可能になります。

トレース/ログ・ファイル値の設定

コネクタ構成ファイルまたはコネクタ定義ファイルを開くと、`Connector Configurator` は、そのファイルのログおよびトレースの値をデフォルト値として使用します。`Connector Configurator` 内でこれらの値を変更できます。

ログとトレースの値を変更するには、以下の手順を実行します。

1. 「**トレース/ログ・ファイル**」タブをクリックします。
2. ログとトレースのどちらでも、以下のいずれかまたは両方へのメッセージの書き込みを選択できます。
 - コンソールに (STDOUT):
ログ・メッセージまたはトレース・メッセージを `STDOUT` ディスプレイに書き込みます。

注: `STDOUT` オプションは、Windows プラットフォームで実行しているコネクタの「**トレース/ログ・ファイル**」タブでのみ使用できます。

- ファイルに:
ログ・メッセージまたはトレース・メッセージを指定されたファイルに書き込みます。ファイルを指定するには、ディレクトリー・ボタン (省略符号) をクリックし、指定する格納場所に移動し、ファイル名を指定し、「**保管**」をクリックします。ログ・メッセージまたはトレース・メッセージは、指定した場所の指定したファイルに書き込まれます。

注: ログ・ファイルとトレース・ファイルはどちらも単純なテキスト・ファイルです。任意のファイル拡張子を使用してこれらのファイル名を設定できます。ただし、トレース・ファイルの場合、拡張子として `.trace` ではなく `.trc` を使用することをお勧めします。これは、システム内に存在する可能性がある他のファイルとの混同を避けるためです。ログ・ファイルの場合、通常使用されるファイル拡張子は `.log` または `.txt` です。

データ・ハンドラー

構成のデータ・ハンドラー・セクションは、DeliveryTransport に JMS の値を指定し、ContainerManagedEvents に JMS の値を指定した場合にのみ使用可能です。すべてのアダプターでデータ・ハンドラーを使用できるわけではありません。

これらのプロパティに使用する値については、『コネクターの標準プロパティ』の ContainerManagedEvents の下の説明を参照してください。詳しくは、「コネクター開発ガイド (C++ 用)」または「コネクター開発ガイド (Java 用)」を参照してください。

構成ファイルの保管

コネクターの構成が完了したら、コネクター構成ファイルを保管します。Connector Configurator では、構成中に選択したブローカー・モードで構成ファイルが保管されます。Connector Configurator のタイトル・バーには現在のブローカー・モード (ICS、WMQI、または WAS) が常に表示されます。

ファイルは XML 文書として保管されます。XML 文書は次の 3 通りの方法で保管できます。

- System Manager から、統合コンポーネント・ライブラリーに *.con 拡張子付きファイルとして保管します。
- 指定したディレクトリーに保管します。
- スタンドアロン・モードで、ディレクトリー・フォルダーに *.cfg 拡張子付きファイルとして保管します。デフォルトでは、ファイルは %WebSphereAdapters%bin%Data%App に保管されます。
- WebSphere Application Server プロジェクトが設定されている場合は、ここにも保管できます。

System Manager でのプロジェクトの使用法、および配置の詳細については、以下のインプリメンテーション・ガイドを参照してください。

- ICS: 「*WebSphere InterChange Server* インプリメンテーション・ガイド」
- WebSphere Message Brokers: 「*WebSphere Message Brokers* 使用アダプター・インプリメンテーション・ガイド」
- WAS: 「*アダプター実装ガイド (WebSphere Application Server)*」

構成ファイルの変更

既存の構成ファイルの統合ブローカー設定を変更できます。これにより、他のブローカーで使用する構成ファイルを新規に作成するときに、このファイルをテンプレートとして使用できます。

注: 統合ブローカーを切り替える場合には、ブローカー・モード・プロパティと同様に他の構成プロパティも変更する必要があります。

既存の構成ファイルのブローカー選択を変更するには (オプション)、以下の操作を実行します。

- 既存の構成ファイルを Connector Configurator で開きます。
- 「標準のプロパティ」タブを選択します。

- 「標準のプロパティ」タブの「**BrokerType**」フィールドで、ご使用のブローカーに合った値を選択します。
 現行値を変更すると、プロパティ・ウィンドウの利用可能なタブおよびフィールド選択がただちに更新され、選択した新規ブローカーに適したタブとフィールドのみが表示されます。

構成の完了

コネクタの構成ファイルを作成し、編集した後で、コネクタの始動時にコネクタが構成ファイルの位置を特定できるかどうかを確認してください。

これを行うには、コネクタが使用する始動ファイルを開き、コネクタ構成ファイルで使用されている格納場所とファイル名が、ファイルに対して指定した名前およびファイルを格納したディレクトリまたはパスと正確に一致しているかどうかを検証します。

グローバル化環境における Connector Configurator の使用

Connector Configurator はグローバル化され、構成ファイルと統合ブローカー間の文字変換を処理できます。Connector Configurator はネイティブ・エンコード方式を使用します。構成ファイルに書き込む場合は、UTF-8 エンコード方式を使用します。

Connector Configurator は、次について英語以外の文字に対応します。

- すべての値フィールド
- ログ・ファイルおよびトレース・ファイル・パス（「**トレース/ログ・ファイル**」タブで指定）

CharacterEncoding および Locale 標準構成プロパティのドロップ・リストに表示されるのは、サポートされる値の一部のみです。他の値をドロップ・リストに追加するには、製品ディレクトリの %Data%Std%stdConnProps.xml ファイルを手動で変更する必要があります。

例えば、Locale プロパティの値のリストにロケール en_GB を追加するには、stdConnProps.xml ファイルを開き、以下に太文字で示した行を追加してください。

```
<Property name="Locale"
isRequired="true"
updateMethod="component restart">
  <ValidType>String</ValidType>
  <ValidValues>
    <Value>ja_JP</Value>
    <Value>ko_KR</Value>
    <Value>zh_CN</Value>
    <Value>zh_TW</Value>
    <Value>fr_FR</Value>
    <Value>de_DE</Value>
    <Value>it_IT</Value>
    <Value>es_ES</Value>
    <Value>pt_BR</Value>
    <Value>en_US</Value>
    <Value>en_GB</Value>
  </ValidValues>
  <DefaultValue>en_US</DefaultValue>
</Property>
```

コネクタの標準プロパティ

ここでは、WebSphere Business Integration アダプターのコネクタ・コンポーネントの標準構成プロパティについて説明します。このセクションの内容は、以下の統合ブローカーで動作するコネクタを対象とします。

- WebSphere InterChange Server (ICS)
- WebSphere MQ Integrator、WebSphere MQ Integrator Broker、および WebSphere Business Integration Message Broker。これらは WebSphere Message Broker と総称します (また、Connector Configurator では WMQI で示します)。
- Information Integrator (II)
- WebSphere Application Server (WAS)

使用するアダプターが DB2 Information Integrator をサポートする場合は、WMQI オプションと DB2 II 標準のプロパティを使用します (183 ページの表 13の「注」カラムを参照)。

アダプター用に設定するプロパティは、使用する統合ブローカーにより異なります。統合ブローカーは Connector Configurator を使用して選択します。ブローカーを選択すると、Connector Configurator によりアダプターに構成する必要がある標準のプロパティがリストされます。

このコネクタ固有のプロパティの詳細については、本書の関連するセクションを参照してください。

新規プロパティ

以下の標準プロパティが、今回のリリースで追加されました。

- AdapterHelpName
- BiDi.Application
- BiDi.Broker
- BiDi.Metadata
- BiDi.Transformation
- CommonEventInfrastructure
- CommonEventInfrastructureContextURL
- ControllerEventSequencing
- jms.ListenerConcurrency
- jms.TransportOptimized
- ResultsSetEnabled
- ResultsSetSize
- TivoliTransactionMonitorPerformance

標準のコネクタのプロパティの概要

コネクタには、以下の 2 つのタイプの構成プロパティがあります。

- 標準の構成プロパティ。これはフレームワークで使用されます。

- アプリケーション、またはコネクタ固有の構成プロパティ。これはエージェントで使用されます。

これらのプロパティにより、アダプター・フレームワークとエージェントの実行時動作を決定します。

このセクションでは、Connector Configurator の始動方法と、すべてのプロパティに共通の特性について説明します。コネクタ固有の構成プロパティについては、該当するアダプターのユーザーズ・ガイドを参照してください。

Connector Configurator の始動

コネクタ・プロパティは、Connector Configurator から構成します。Connector Configurator には、System Manager からアクセスします。Connector Configurator の使用法の詳細については、本書の Connector Configurator のセクションを参照してください。

Connector Configurator と System Manager は、Windows システム上でのみ実行されます。コネクタを UNIX システム上で稼働している場合でも、これらのツールがインストールされた Windows マシンが必要です。

UNIX 上で動作するコネクタのコネクタ・プロパティを設定する場合は、Windows マシン上で System Manager を起動し、UNIX の統合ブローカーに接続してから、コネクタ用の Connector Configurator を開く必要があります。

構成プロパティ値の概要

コネクタは、以下の順序に従ってプロパティの値を決定します。

1. デフォルト
2. リポジトリ (WebSphere InterChange Server (ICS) が統合ブローカーである場合のみ有効)
3. ローカル構成ファイル
4. コマンド行

プロパティ・フィールドのデフォルトの長さは 255 文字です。string プロパティ・タイプの長さには制限はありません。整数タイプの長さは、アダプターが稼働しているサーバーにより決定されます。

コネクタは、始動時に構成値を取得します。実行時セッション中に 1 つ以上のコネクタ・プロパティの値を変更する場合は、プロパティの更新メソッドによって、変更を有効にする方法が決定されます。

プロパティの更新特性、つまりコネクタのプロパティの変更をいつどのようにして有効にするかは、プロパティの性質により異なります。

標準のコネクタ・プロパティには、以下の 4 種類の更新メソッドがあります。

• 動的

変更が System Manager に保管されると、即時に新しい値が有効になります。ただし、コネクタがスタンドアロン・モードで (System Manager から独立) 稼働

している場合、例えば、WebSphere Message Broker の 1 つで稼働している場合には、プロパティーは、構成ファイルのみを使用して変更できます。この場合、動的更新は実行できません。

- **エージェント再始動 (ICS のみ)**

新しい値は、コネクタ・エージェントを停止して再始動した後でのみ有効になります。

- **コンポーネント再始動**

新しい値は、System Manager でコネクタを停止して再始動した後でのみ、有効になります。エージェントまたはサーバー・プロセスは、停止して再始動する必要はありません。

- **システム再始動**

新しい値は、コネクタ・エージェントとサーバーを停止して再始動した後でのみ有効になります。

特定のプロパティーの更新方法を決定するには、「Connector Configurator」ウィンドウの「更新メソッド」列を参照するか、183 ページの表 13の「更新メソッド」列を参照してください。

標準のプロパティーは、以下の 3 つの場所に常駐することができます。一部のプロパティーは複数の場所に常駐できます。

- **ReposController**

プロパティーはコネクタ・コントローラーに常駐し、そこでのみ有効です。エージェント・サイドの値を変更しても、コントローラーには影響しません。

- **ReposAgent**

プロパティーはエージェントに常駐し、そこでのみ有効です。ローカル構成で、プロパティーに応じて、この値をオーバーライドできます。

- **LocalConfig**

プロパティーはコネクタの構成ファイルに常駐し、構成ファイルを使用してのみ動作できます。コントローラーはプロパティーの値を変更できず、システムが再配置されてコントローラーが明示的に更新されない限り、構成ファイルに行われた変更は認識しません。

標準のプロパティーのクイック・リファレンス

183 ページの表 13 に、標準のコネクタ構成プロパティーのクイック・リファレンスを示します。コネクタによっては必要のないプロパティーがあり、統合ブローカーによってプロパティーの設定が異なる場合があります。

各プロパティーの説明については、表に続くセクションを参照してください。

注: 183 ページの表 13 の「注」列の、「RepositoryDirectory は <REMOTE> に設定されます」という句は、ブローカーが InterChange Server であることを示します。ブローカーが WMQI または WAS の場合、リポジトリ・ディレクトリーは <ProductDir>%repository に設定されます。

表 13. 標準構成プロパティの要約

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
AdapterHelpName	有効な <RegionalSetting> ディレクトリーを含む <ProductDir>%bin¥Data ¥App¥Help¥ の有効なサブディレクトリーの 1 つ	テンプレート名 (有効な場合) またはブランク・ワールド	コンポーネント再始動	サポートされる地域の設定。chs_chn、cht_twn、deu_deu、esn_esp、fra_fra、ita_ita、jpn_jpn、kor_kor、ptb_bra、および enu_usa (デフォルト) を含む。
AdminInQueue	有効な JMS キュー名	<CONNECTORNAME>/ADMININQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
AdminOutQueue	有効な JMS キュー名	<CONNECTORNAME>/ADMINOUTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
AgentConnections	1 から 4	1	コンポーネント再始動	このプロパティは、DeliveryTransport の値が MQ または IDL で、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
AgentTraceLevel	0 から 5	0	動的プロカーが ICS の場合。それ以外は、コンポーネント再始動	
ApplicationName	アプリケーション名	コネクタ・アプリケーション名に指定された値	コンポーネント再始動	
BiDi.Application	以下の双方向属性の任意の有効な組み合わせ。 先頭の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。
BiDi.Broker	以下の双方向属性の任意の有効な組み合わせ。 先頭の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。BrokerType の値が ICS の場合、プロパティは読み取り専用です。
BiDi.Metadata	以下の双方向属性の任意の有効な組み合わせ。 先頭の文字: I、V 2 番目の文字: L、R 3 番目の文字: Y、N 4 番目の文字: S、N 5 番目の文字: H、C、N	ILYNN (5 文字)	コンポーネント再始動	このプロパティは、BiDi.Transformation の値が true の場合のみ有効です。

表 13. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
BiDi.Transformation	true または false	false	コンポーネント再始動	このプロパティは、BrokerType の値が WAS でない場合のみ有効です。
BrokerType	ICS、WMQI、WAS	ICS	コンポーネント再始動	
CharacterEncoding	任意のサポート・コード。以下に、このサブセットを示します。 ascii7、ascii8、SJIS、Cp949、GBK、Big5、Cp297、Cp273、Cp280、Cp284、Cp037、Cp437 .	ascii7	コンポーネント再始動	このプロパティは、C++コネクタの場合のみ有効です。
CommonEventInfrastructure	true または false	false	コンポーネント再始動	
CommonEventInfrastructureURL	URL スtring例えば、corbaloc:iiop: host:2809.	デフォルト値なし	コンポーネント再始動	このプロパティは、CommonEvent インフラストラクチャーの値が true の場合のみ有効です。
ConcurrentEventTriggeredFlows	1 から 32,767	1	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ContainerManagedEvents	ブランクまたは JMS	ブランク	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
ControllerEventSequencing	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ControllerStoreAndForwardMode	true または false	true	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ControllerTraceLevel	0 から 5	0	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
DeliveryQueue	任意の有効な JMS キュー名	<CONNECTORNAME>/DELIVERYQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
DeliveryTransport	MQ、IDL、または JMS	IDL、RepositoryDirectory の値が <REMOTE> の場合。それ以外は JMS	コンポーネント再始動	RepositoryDirectory の値が <REMOTE> でない場合、このプロパティの有効な唯一の値は JMS です。
DuplicateEventElimination	true または false	false	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。

表 13. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
EnableOidForFlowMonitoring	true または false	false	コンポーネント再始動	このプロパティは、BrokerType の値が ICS の場合のみ有効です。
FaultQueue	任意の有効なキュー名	<CONNECTORNAME>/FAULTQUEUE	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.FactoryClassName	CxCommon.Messaging.jms.IBMMQSeriesFactory、CxCommon.Messaging.jms.SonicMQFactory、または任意の Java クラス名	CxCommon.Messaging.jms.IBMMQSeriesFactory	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.ListenerConcurrency	1 から 32767	1	コンポーネント再始動	このプロパティは、jms.TransportOptimized の値が true の場合のみ有効です。
jms.MessageBrokerName	jms.FactoryClassName の値が IBM の場合は、crossworlds.queue.manager を使用します。	crossworlds.queue.manager	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.NumConcurrentRequests	正整数	10	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.Password	任意の有効なパスワード		コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
jms.TransportOptimized	true または false	false	コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS で、BrokerType の値が ICS の場合のみ有効です。
jms.UserName	任意の有効な名前		コンポーネント再始動	このプロパティは、DeliveryTransport の値が JMS の場合のみ有効です。
JvmMaxHeapSize	ヒープ・サイズ (メガバイト単位)	128m	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
JvmMaxNativeStackSize	スタックのサイズ (キロバイト単位)	128k	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
JvmMinHeapSize	ヒープ・サイズ (メガバイト単位)	1m	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
ListenerConcurrency	1 から 100	1	コンポーネント再始動	このプロパティは、DeliveryTransport の値が MQ の場合のみ有効です。

表 13. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
Locale	以下は、サポートされるロケールのサブセットです。 en_US、ja_JP、ko_KR、zh_CN、zh_TW、fr_FR、de_DE、it_IT、es_ES、pt_BR	en_US	コンポーネント再始動	
LogAtInterchangeEnd	true または false	false	コンポーネント再始動	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
MaxEventCapacity	1 から 2147483647	2147483647	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
MessageFileName	有効なファイル名	InterchangeSystem.txt	コンポーネント再始動	
MonitorQueue	任意の有効なキュー名	<CONNECTORNAME>/MONITORQUEUE	コンポーネント再始動	このプロパティは、DuplicateEventElimination の値が true で、ContainerManagedEvents に値がない場合のみ有効です。
OADAutoRestartAgent	true または false	false	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
OADMaxNumRetry	正整数	1000	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
OADRetryTimeInterval	正整数分	10	動的	このプロパティは、RepositoryDirectory の値が <REMOTE> に設定されて、BrokerType の値が ICS の場合のみ有効です。
PollEndTime	HH = 0 から 23 MM = 0 から 59	HH:MM	コンポーネント再始動	
PollFrequency	正整数 (ミリ秒)	10000	動的ブローカーが ICS の場合。それ以外はコンポーネント再始動	
PollQuantity	1 から 500	1	エージェント再始動	このプロパティは、ContainerManagedEvents の値が JMS の場合のみ有効です。
PollStartTime	HH = 0 から 23 MM = 0 から 59	HH:MM	コンポーネント再始動	

表 13. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
RepositoryDirectory	ブローカーが ICS の場合は <REMOTE>。それ以外は、任意の有効なローカル・ディレクトリー	ICS の場合、値は <REMOTE> に設定 WMQI および WAS の場合、値は <ProductDir %repository に設定	エージェント 再始動	
RequestQueue	有効な JMS キュー名	<CONNECTORNAME> /REQUESTQUEUE	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
ResponseQueue	有効な JMS キュー名	<CONNECTORNAME> /RESPONSEQUEUE	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
RestartRetryCount	0 から 99	3	動的 ICS の場 合。それ以外 は、コンポー ネント再始動	
RestartRetryInterval	分の値 1 から 2147483647	1	動的 ICS の場 合。それ以外 は、コンポー ネント再始動	
ResultsSetEnabled	true または false	false	コンポーネン ト再始動	DB2II をサポートするコネ クターのみが使用します。 このプロパティは、 DeliveryTransport の値が JMS で、 BrokerType の値 が WMQI の場合のみ有効で す。
ResultsSetSize	正整数	0 (結果セット・サイズが 無限であることを意味し ます)	コンポーネン ト再始動	DB2II をサポートする コネクターのみが使用しま す。このプロパティは、 ResultsSetEnabled の値が true の場合のみ有効です。
RHF2MessageDomain	mrm または xml	mrm	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS で、 WireFormat の値 が CwXML の場合のみ有効で す。
SourceQueue	任意の有効な WebSphere MQ queue 名	<CONNECTORNAME> /SOURCEQUEUE	エージェント 再始動	このプロパティは、 ContainerManagedEvents の 値が JMS の場合のみ有効で す。
SynchronousRequest Queue	任意の有効なキュー名	<CONNECTORNAME> /SYNCHRONOUSREQUEST QUEUE	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
SynchronousRequest Timeout	0 から任意の数 (ミリ秒)	0	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
SynchronousResponse Queue	任意の有効なキュー名	<CONNECTORNAME> /SYNCHRONOUSRESPONSE QUEUE	コンポーネン ト再始動	このプロパティは、 DeliveryTransport の値が JMS の場合のみ有効です。
TivoliMonitorTransaction Performance	true または false	false	コンポーネン ト再始動	

表 13. 標準構成プロパティの要約 (続き)

プロパティ名	指定可能な値	デフォルト値	更新メソッド	注
WireFormat	CwXML または CwBO	CwXML	エージェント再始動	このプロパティの値は、RepositoryDirectory の値が <REMOTE> に設定されていない場合は、CwXML でなければなりません。 RepositoryDirectory の値が <REMOTE> に設定されている場合は、CwBO でなければなりません。
WsifSynchronousRequestTimeout	0 から任意の数 (ミリ秒)	0	コンポーネント再始動	このプロパティは、BrokerType の値が WAS の場合のみ有効です。
XMLNamespaceFormat	short または long	short	エージェント再始動	このプロパティは、BrokerType の値が WMQI または WAS の場合のみ有効です。

標準のプロパティ

ここでは、標準のコネクター構成プロパティについて説明します。

AdapterHelpName

AdapterHelpName プロパティは、コネクター固有の全般ヘルプ・ファイルが存在するディレクトリーの名前です。ディレクトリーは <ProductDir>%bin%Data%App%Help に存在し、少なくとも言語ディレクトリー enu_usa を含んでいる必要があります。ロケールに応じて、他のディレクトリーを含むことができます。

デフォルト値はテンプレート名 (有効な場合) か、またはブランクです。

AdminInQueue

AdminInQueue プロパティで、コネクターに管理メッセージを送信するために統合ブローカーによって使用されるキューを指定します。

デフォルト値は <CONNECTORNAME>/ADMININQUEUE です。

AdminOutQueue

AdminOutQueue プロパティで、統合ブローカーに管理メッセージを送信するためにコネクターによって使用されるキューを指定します。

デフォルト値は <CONNECTORNAME>/ADMINOUTQUEUE です。

AgentConnections

AgentConnections プロパティで、ORB の初期設定時に開かれる ORB (オブジェクト・リクエスト・ブローカー) 接続の数を制御します。

これは、RepositoryDirectory の値が <REMOTE> に設定されて、DeliveryTransport プロパティの値が MQ または IDL の場合のみ有効です。

このプロパティのデフォルト値は 1 です。

AgentTraceLevel

AgentTraceLevel プロパティーで、アプリケーション固有のコンポーネントのトレース・メッセージのレベルを設定します。コネクターは、設定されたトレース・レベル以下の該当するトレース・メッセージをすべてデリバリーします。

デフォルト値は 0 です。

ApplicationName

ApplicationName プロパティーで、コネクター・アプリケーションの名前を一意に識別します。この名前は、システム管理者によって、統合環境をモニターするために使用されます。コネクターを実行する前に、このプロパティーに値を指定する必要があります。

デフォルトは、コネクターの名前です。

BiDi.Application

BiDi.Application プロパティーで、このアダプターでサポートされる任意のビジネス・オブジェクトの形式で外部アプリケーションからアダプターに送信されるデータの双方向のフォーマットを指定します。プロパティーで、アプリケーション・データの双方向の属性を定義します。これらの属性は、以下のとおりです。

- テキストのタイプ: 暗黙またはビジュアル (I または V)
- テキスト方向: 左から右または右から左 (L または R)
- 対称スワッピング: オン/オフ (Y または N)
- 整形 (アラビア語): オン/オフ (S または N)
- 数字整形 (アラビア語): ヒンディ語、コンテキスト、または公称 (H、C、または N)

このプロパティーは、BiDi.Transformation プロパティー値が true に設定されている場合のみ有効です。

デフォルト値は ILYNN です (暗黙、左から右、オン、オフ、公称)。

BiDi.Broker

BiDi.Broker プロパティーで、任意のサポートされるビジネス・オブジェクトの形式でアダプターから統合ブローカーに送信されるデータの双方向のフォーマットを指定します。データの双方向の属性を定義します。これは、上記の BiDi.Application にリストされています。

このプロパティーは、BiDi.Transformation プロパティー値が true に設定されている場合のみ有効です。BrokerType プロパティーが ICS の場合、プロパティー値は読み取り専用です。

デフォルト値は ILYNN です (暗黙、左から右、オン、オフ、公称)。

BiDi.Metadata

BiDi.Metadata プロパティーで、メタデータの双方向フォーマットまたは属性を定義します。メタデータは、外部アプリケーションへのリンクを確立し保守するためにコネクターによって使用されます。属性の設定値は、双方向機能を使用する各アダ

プターに固有です。使用するアダプターが双方向の処理をサポートする場合、詳細についてはアダプター固有のプロパティのセクションを参照してください。

このプロパティは、`BiDi.Transformation` プロパティ値が `true` に設定されている場合のみ有効です。

デフォルト値は `ILYNN` です (暗黙、左から右、オン、オフ、公称)。

BiDi.Transformation

`BiDi.Transformation` プロパティで、実行時に双方向の変換を実行するかどうかを指定します。

プロパティ値が `true` に設定されている場合、`BiDi.Application`、`BiDi.Broker`、および `BiDi.Metadata` プロパティは使用可能です。プロパティ値が `false` に設定されている場合、これらは隠蔽されます。

デフォルト値は `false` です。

BrokerType

`BrokerType` プロパティで、使用する統合ブローカーのタイプを識別します。可能な値は、`ICS`、`WMQI` (`WMQI`、`WMQIB`、または `WBIMB` の場合)、または `WAS` です。

CharacterEncoding

`CharacterEncoding` プロパティで、文字 (英字、数字、または句読点) を数値にマップするときに使用する文字コード・セットを指定します。

注: Java ベースのコネクタはこのプロパティを使用しません。C++ コネクタは、このプロパティに値 `ascii7` を使用します。

デフォルトでは、サポートされる文字エンコードの一部のみが表示されます。リストに他のサポートされる値を追加するには、製品ディレクトリー (`<ProductDir>`) の `¥Data¥Std¥stdConnProps.xml` ファイルを手動で変更する必要があります。詳細については、本書の `Connector Configurator` に関する付録を参照してください。

CommonEventInfrastructure

`Common Event Infrastructure (CEI)` とは、生成されたイベントを処理する単純なイベント管理機能です。`CommonEventInfrastructure` プロパティで、実行時に `CEI` を呼び出すかどうかを指定します。

デフォルト値は `false` です。

CommonEventInfrastructureContextURL

`CommonEventInfrastructureContextURL` を使用して、`Common Event Infrastructure (CEI)` サーバー・アプリケーションを実行する `WAS` サーバーにアクセスできます。このプロパティで、使用する `URL` を指定します。

このプロパティは、`CommonEventInfrastructure` が `true` に設定されている場合のみ有効です。

デフォルト値は空白・フィールドです。

ConcurrentEventTriggeredFlows

`ConcurrentEventTriggeredFlows` プロパティで、コネクタによってイベント・デリバリーのために並行して処理できるビジネス・オブジェクトの数を決定します。この属性の値を、並行してマップおよびデリバリーするビジネス・オブジェクトの数に設定します。例えば、このプロパティの値を 5 に設定すると、5 個のビジネス・オブジェクトが並行処理されます。

このプロパティを 1 よりも大きい値に設定すると、ソース・アプリケーションのコネクタが、複数のイベント・ビジネス・オブジェクトを同時にマップして、複数のコラボレーション・インスタンスにそれらのビジネス・オブジェクトを同時にデリバリーすることができます。これにより、統合ブローカーへのビジネス・オブジェクトのデリバリーにかかる時間、特にビジネス・オブジェクトが複雑なマップを使用している場合のデリバリー時間が短縮されます。ビジネス・オブジェクトがコラボレーションまでの到達に要する時間を短縮すると、システム全体のパフォーマンスを向上させることができます。

フロー全体 (ソース・アプリケーションから宛先アプリケーションまで) の並行処理を実装するには、以下のプロパティを構成する必要があります。

- 「並行イベントの最大数」プロパティをマルチスレッドを使用するのに十分な大きさに設定して、マルチスレッドを使用するようにコラボレーションを構成する必要があります。
- 宛先アプリケーションのアプリケーション固有コンポーネントを、複数の要求を並行して実行できるように構成する必要があります。つまり、このコンポーネントがマルチスレッド化されているか、またはコネクタ・エージェント並列処理を使用でき、複数プロセスに対応するよう構成されている必要があります。並列処理の度合い構成プロパティを 1 よりも大きな値に設定する必要があります。

`ConcurrentEventTriggeredFlows` プロパティは、単一スレッドで順次に処理されるコネクタのポーリングには影響しません。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は 1 です。

ContainerManagedEvents

`ContainerManagedEvents` プロパティにより、JMS イベント・ストアを使用する JMS 対応コネクタが、保証付きイベント・デリバリーを提供できるようになります。保証付きイベント・デリバリーでは、イベントはソース・キューから除去され、単一 JMS トランザクションとして宛先キューに配置されます。

このプロパティを JMS に設定した場合には、保証付きイベント・デリバリーを使用できるように次のプロパティも設定する必要があります。

- `PollQuantity` = 1 から 500
- `SourceQueue` = `/SOURCEQUEUE`

また、`MimeType` および `DHClass` (データ・ハンドラー・クラス) プロパティを使用してデータ・ハンドラーを構成する必要もあります。 `DataHandlerConfigMOName`

(メタ・オブジェクト名、オプション) を追加することもできます。これらのプロパティの値を設定するには、Connector Configurator の「データ・ハンドラー」タブを使用します。

これらのプロパティはアダプター固有ですが、以下に一部のサンプル値を示します。

- `MimeType = text/xml`
- `DHClass = com.crossworlds.DataHandlers.text.xml`
- `DataHandlerConfigMOName = MO_DataHandler_Default`

「データ・ハンドラー」タブにあるこれらの値のフィールドは、ContainerManagedEvents プロパティの値が JMS に設定されている場合にのみ表示されます。

注: ContainerManagedEvents が JMS に設定されていると、コネクターは `pollForEvents()` メソッドを呼び出さないため、そのメソッドの機能は使用できません。

ContainerManagedEvents property プロパティは、DeliveryTransport プロパティが JMS に設定されている場合のみ有効です。

デフォルト値はありません。

ControllerEventSequencing

ControllerEventSequencing プロパティにより、コネクター・コントローラー内のイベント順序付けを使用可能にします。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合のみ有効です (BrokerType は ICS)。

デフォルト値は true です。

ControllerStoreAndForwardMode

ControllerStoreAndForwardMode プロパティにより、宛先のアプリケーション固有のコンポーネントが使用不可であることをコネクター・コントローラーが検出した後の、コネクター・コントローラーの動作を設定します。

このプロパティを true に設定した場合、イベントが ICS に到達したときに宛先側のアプリケーション固有のコンポーネントが使用不可であれば、コネクター・コントローラーはそのアプリケーション固有のコンポーネントへの要求をブロックします。アプリケーション固有のコンポーネントが作動可能になると、コネクター・コントローラーはアプリケーション固有のコンポーネントにその要求を転送します。

ただし、コネクター・コントローラーが宛先側のアプリケーション固有のコンポーネントにサービス呼び出し要求を転送した後でこのコンポーネントが使用不可になった場合、コネクター・コントローラーはその要求を失敗させます。

このプロパティが `false` に設定されている場合、宛先のアプリケーション固有のコンポーネントが使用不可であることをコネクター・コントローラーが検出すると、コネクター・コントローラーはすべてのサービス呼び出し要求を失敗させます。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です (`BrokerType` プロパティの値は `ICS`)。

デフォルト値は `true` です。

ControllerTraceLevel

`ControllerTraceLevel` プロパティにより、コネクター・コントローラーのトレース・メッセージのレベルを設定します。

このプロパティは、`RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合のみ有効です。

デフォルト値は `0` です。

DeliveryQueue

`DeliveryQueue` プロパティで、コネクターがビジネス・オブジェクトを統合ブローカーに送信するために使用するキューを定義します。

このプロパティは、`DeliveryTransport` プロパティの値が `JMS` に設定されている場合のみ有効です。

デフォルト値は `<CONNECTORNAME>/DELIVERYQUEUE` です。

DeliveryTransport

`DeliveryTransport` プロパティで、イベントのデリバリーのためのトランスポート機構を指定します。指定可能な値は、WebSphere MQ の `MQ`、CORBA IIOP の `IDL`、Java Messaging Service の `JMS` です。

- `RepositoryDirectory` プロパティの値が `<REMOTE>` に設定されている場合、`DeliveryTransport` プロパティの値は、`MQ`、`IDL`、または `JMS` が可能で、デフォルトは `IDL` です。
- `RepositoryDirectory` プロパティの値がローカル・ディレクトリーの場合は、指定可能な値は `JMS` のみです。

`RepositoryDirectory` プロパティの値が `MQ` または `IDL` の場合、コネクターは、サービス呼び出し要求と管理メッセージを `CORBA IIOP` で送信します。

デフォルト値は `JMS` です。

WebSphere MQ と IDL: イベントのデリバリー・トランスポートには、`IDL` ではなく `WebSphere MQ` を使用してください (1 種類の製品だけを使用する必要がある場合を除きます)。WebSphere MQ が `IDL` よりも優れている点は以下のとおりです。

- 非同期 (ASYNC) 通信:
WebSphere MQ を使用すると、アプリケーション固有のコンポーネントは、サーバーが利用不能である場合でも、イベントをポーリングして永続的に格納することができます。
- サーバー・サイド・パフォーマンス:
WebSphere MQ を使用すると、サーバー・サイドのパフォーマンスが向上します。最適化モードでは、WebSphere MQ はイベントへのポインターのみをリポジトリ・データベースに格納するので、実際のイベントは WebSphere MQ キュー内に残ります。これにより、サイズが大きい可能性のあるイベントがリポジトリ・データベースに書き込まれなくなります。
- エージェント・サイド・パフォーマンス:
WebSphere MQ を使用すると、アプリケーション固有のコンポーネント側のパフォーマンスが向上します。WebSphere MQ を使用すると、コネクタのポーリング・スレッドはイベントを取り出し、そのイベントをコネクタのキューに追加してから、次のイベントを取り出します。これは IDL よりも高速です。IDL の場合、コネクタのポーリング・スレッドはイベントを取り出し、ネットワーク経由でサーバー・プロセスにアクセスし、リポジトリ・データベース内にイベントを永続的に格納してから、次のイベントを取り出す必要があります。

JMS: JMS トランスポート機構により、Java Messaging Service (JMS) を使用した、コネクタとクライアント・コネクタ・フレームワークとの間の通信が可能になります。

JMS をデリバリー・トランスポートとして選択した場合、追加の JMS プロパティ (jms.MessageBrokerName、jms.FactoryClassName、jms.Password、および jms.UserName など) が Connector Configurator 内に表示されます。jms.MessageBrokerName および jms.FactoryClassName プロパティは、このトランスポートに必須です。

次の環境でコネクタのために JMS トランスポート機構を使用すると、メモリー制限が課される場合があります。

- AIX 5.0
- WebSphere MQ 5.3.0.1
- ICS が統合ブローカー

この環境では、WebSphere MQ クライアント内でメモリーが使用されるため、(サーバー側の) コネクタ・コントローラーと (クライアント側の) コネクタの両方を始動するのは困難な場合があります。インストールされたシステムでのプロセス・ヒープ・サイズが 768MB より小さい場合には、次の変数およびプロパティを設定します。

- CWSHaredEnv.sh スクリプトに LDR_CNTRL 環境変数を設定します。

このスクリプトは、製品ディレクトリー (<ProductDir>) 下の %bin ディレクトリーに存在します。テキスト・エディターを使用して、CWSHaredEnv.sh スクリプトの最初の行として、次の行を追加します。

```
export LDR_CNTRL=MAXDATA=0x30000000
```

この行により、ヒープ・メモリーの使用量が最大 768 MB (3 セグメント× 256 MB) に制限されます。プロセス・メモリーがこの限度を超えると、ページ・スワッピングが発生することがあり、システムのパフォーマンスが悪化する場合があります。

- `IPCCBaseAddress` プロパティの値を 11 または 12 に設定します。このプロパティの詳細については、「システム・インストール・ガイド (UNIX 版)」を参照してください。

DuplicateEventElimination

このプロパティの値が `true` に設定されていると、JMS 対応コネクターが確実に、重複イベントがデリバリー・キューにデリバリーされないようにできます。この機能を使用するには、コネクターの作成時に、アプリケーション固有コード内のビジネス・オブジェクト `ObjectEventId` の属性として一意のイベント ID をコネクターに設定する必要があります。

注: このプロパティの値が `true` に設定されている場合、`MonitorQueue` プロパティで、保証付きイベント・デリバリーの提供を使用可能にする必要があります。

デフォルト値は `false` です。

EnableOidForFlowMonitoring

このプロパティの値が `true` に設定されていると、アダプター・ランタイムは着信 `ObjectEventID` をフロー・モニターの外部キーとしてマーク付けします。

このプロパティは、`BrokerType` プロパティが `ICS` に設定されている場合のみ有効です。

デフォルト値は `false` です。

FaultQueue

コネクターが、メッセージの処理中にエラーを検出すると、メッセージ (と状況表示および問題の記述) を `FaultQueue` プロパティで指定されたキューに移動します。

デフォルト値は `<CONNECTORNAME>/FAULTQUEUE` です。

jms.FactoryClassName

`jms.FactoryClassName` プロパティで、JMS プロバイダーのためにインスタンスを生成するクラス名を指定します。`DeliveryTransport` プロパティの値が `JMS` に設定されている場合は、このプロパティを設定する必要があります。

デフォルトは `CxCommon.Messaging.jms.IBMMQSeriesFactory` です。

jms.ListenerConcurrency

`jms.ListenerConcurrency` プロパティで、JMS コントローラーの並行リスナーの数を指定します。これで、コントローラー内でメッセージを並行して取り出して処理するスレッドの数を指定します。

このプロパティは、`jms.OptimizedTransport` プロパティの値が `true` の場合のみ有効です。

デフォルト値は 1 です。

jms.MessageBrokerName

`jms.MessageBrokerName` で、JMS プロバイダーに使用するブローカー名を指定します。デリバリー・トランスポート・メカニズム (`DeliveryTransport` プロパティ) として JMS を指定する場合は、このコネクタ・プロパティを設定する必要があります。

リモートのメッセージ・ブローカーに接続する場合、このプロパティには以下の値が必要です。

`QueueMgrName:Channel:HostName:PortNumber`

ここで、

`QueueMgrName` はキュー・マネージャーの名前。

`Channel` はクライアントが使用するチャネル。

`HostName` は、キュー・マネージャーが存在するマシンの名前。

`PortNumber` は、キュー・マネージャーが `listen` するために使用するポート番号。

例:

```
jms.MessageBrokerName = WBIMB.Queue.Manager:CHANNEL1:RemoteMachine:1456
```

デフォルト値は `crossworlds.queue.manager` です。ローカルのメッセージ・ブローカーに接続するときは、デフォルトを使用してください。

jms.NumConcurrentRequests

`jms.NumConcurrentRequests` プロパティで、1 つのコネクタに対して同時に送信できる並行サービス呼び出し要求の最大数を指定します。最大数に到達すると、新規のサービス呼び出しはブロックされ、処理するには別の要求が完了するのを待機する必要があります。

デフォルト値は 10 です。

jms.Password

`jms.Password` プロパティで、JMS プロバイダーのパスワードを指定します。このプロパティの値はオプションです。

デフォルト値はありません。

jms.TransportOptimized

`jms.TransportOptimized` プロパティで、WIP (処理中の作業) を最適化するかどうかを判別します。WIP を最適化するには、WebSphere MQ プロバイダーが必要です。最適化された WIP が作動するには、メッセージング・プロバイダーが以下のことを可能でなければなりません。

1. キューからでなくてもメッセージを読み取りできる。
2. メッセージ全体を受信側のメモリー・スペースに転送しなくても、特定の ID でメッセージを削除できる。
3. 特定の ID (リカバリーのために必要) を使用してメッセージを読み取りできる。

4. 読み込まれなかったイベントが発生する時点を追跡できる。

JMS API は、上記の 2 および 4 の条件に該当しないので、最適化 WIP に使用できませんが、MQ Java API は 4 つのすべての条件に適合するので、最適化 WIP に必須です。

このプロパティは、DeliveryTransport の値が JMS で、BrokerType の値が ICS の場合のみ有効です。

デフォルト値は false です。

jms.UserName

jms.UserName プロパティで、JMS プロバイダーのユーザー名を指定します。このプロパティの値はオプションです。

デフォルト値はありません。

JvmMaxHeapSize

JvmMaxHeapSize プロパティで、エージェントの最大ヒープ・サイズを指定します (メガバイト単位)。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 128m です。

JvmMaxNativeStackSize

JvmMaxNativeStackSize プロパティで、エージェントの最大ネイティブ・スタック・サイズを指定します (キロバイト単位)。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 128k です。

JvmMinHeapSize

JvmMinHeapSize プロパティで、エージェントの最小ヒープ・サイズを指定します (メガバイト単位)。

このプロパティは、RepositoryDirectory プロパティの値が <REMOTE> に設定されている場合のみ有効です。

デフォルト値は 1m です。

ListenerConcurrency

ListenerConcurrency プロパティで、統合ブローカーとして ICS を使用する場合は WebSphere MQ Listener でのマルチスレッド化をサポートします。このプロパティにより、データベースに複数イベントをバッチで書き込みできるので、システム・パフォーマンスが向上します。

このプロパティは、MQ トランスポートを使用するコネクタの場合のみ有効です。 `DeliveryTransport` プロパティの値は、MQ でなければなりません。

デフォルト値は 1 です。

Locale

`Locale` プロパティで、言語コード、国または地域、および、必要な場合には、関連した文字コード・セットを指定します。このプロパティの値で、データの照合やソートの順序、日付と時刻の形式、通貨記号などの国/地域別情報を決定します。

ロケール名は次のような形式になります。

```
ll_TT.codeset
```

ここで、

`ll` は 2 文字の言語コード (小文字)。

`TT` は 2 文字の国/地域コード (大文字)。

`codeset` は関連文字コード・セットの名前 (オプション)。

デフォルトでは、サポートされるロケールの一部のみがリストされます。リストに他のサポートされる値を追加するには、`<ProductDir>%bin` ディレクトリーの `%Data%Std%stdConnProps.xml` ファイルを変更します。詳細については、本書の `Connector Configurator` に関する付録を参照してください。

コネクタが国際化に対応していない場合、このプロパティの有効な値は `en_US` のみです。特定のコネクタが国際化に対応しているかどうかを判別するには、そのアダプターのユーザー・ガイドを参照してください。

デフォルト値は `en_US` です。

LogAtInterchangeEnd

`LogAtInterchangeEnd` プロパティで、統合ブローカーのログ宛先にエラーを記録するかどうかを指定します。

ログ宛先にログを記録すると、電子メール通知もオンになります。これにより、エラーまたは致命的エラーが発生すると、`InterchangeSystem.cfg` ファイルに `MESSAGE_RECIPIENT` の値として指定された受信側の電子メール・メッセージが生成されます。例えば、コネクタがアプリケーションへの接続を失うと、`LogAtInterChangeEnd` の値が `true` の場合、指定されたメッセージ宛先に電子メール・メッセージが送信されます。

このプロパティは、`RespositoryDirectory` の値が `<REMOTE>` に設定されている場合のみ有効です (`BrokerType` プロパティの値は `ICS`)。

デフォルト値は `false` です。

MaxEventCapacity

`MaxEventCapacity` プロパティで、コントローラー・バッファ内のイベントの最大数を指定します。このプロパティは、フロー制御機能で使用されます。

このプロパティは、`RespositoryDirectory` の値が `<REMOTE>` に設定されている場合のみ有効です (`BrokerType` プロパティの値は `ICS`)。

値は 1 から 2147483647 の正の整数にすることができます。

デフォルト値は 2147483647 です。

MessageFileName

MessageFileName プロパティーで、コネクタ・メッセージ・ファイルの名前を指定します。メッセージ・ファイルの標準の場所は、製品ディレクトリーの `¥connectors¥messages` です。メッセージ・ファイルが標準位置に格納されていない場合は、メッセージ・ファイル名を絶対パスで指定します。

コネクタ・メッセージ・ファイルが存在しない場合は、コネクタは `InterchangeSystem.txt` をメッセージ・ファイルとして使用します。このファイルは、製品ディレクトリーに格納されます。

注: コネクタについて、コネクタ独自のメッセージ・ファイルがあるかどうかを判別するには、該当するアダプターのユーザー・ガイドを参照してください。

デフォルト値は `InterchangeSystem` です。

MonitorQueue

MonitorQueue プロパティーで、コネクタが重複イベントをモニターするために使用する論理キューを指定します。

このプロパティーは、`DeliveryTransport` プロパティーの値が `JMS` で、`DuplicateEventElimination` の値が `true` の場合のみ有効です。

デフォルト値は `<CONNECTORNAME>/MONITORQUEUE` です。

OADAutoRestartAgent

OADAutoRestartAgent プロパティーで、コネクタが自動再始動およびリモート再始動機能を使用するかどうかを指定します。この機能は、WebSphere MQ により起動される Object Activation Daemon (OAD) を使用して、異常シャットダウン後にコネクタを再始動したり、System Monitor からリモート・コネクタを始動したりします。

自動再始動機能およびリモート再始動機能を使用可能にするには、このプロパティーを `true` に設定する必要があります。WebSphere MQ により起動される OAD 機能の構成方法については、「システム・インストール・ガイド (Windows 版)」または「システム・インストール・ガイド (UNIX 版)」を参照してください。

このプロパティーは、`RespositoryDirectory` の値が `<REMOTE>` に設定されている場合のみ有効です (`BrokerType` プロパティーの値は `ICS`)。

デフォルト値は `false` です。

OADMaxNumRetry

OADMaxNumRetry プロパティーで、WebSphere MQ により起動される Object Activation Daemon (OAD) が、異常シャットダウン後にコネクタの再始動を自動的に試行する回数の最大数を指定します。このプロパティーを有効にするには、OADAutoRestartAgent プロパティーを `true` に設定する必要があります。

このプロパティは、RespositoryDirectory の値が <REMOTE> に設定されている場合のみ有効です (BrokerType プロパティの値は ICS)。

デフォルト値は 1000 です。

OADRetryTimeInterval

OADRetryTimeInterval プロパティで、WebSphere MQ により起動される Object Activation Daemon (OAD) の再試行時間間隔の分数を指定します。コネクタ・エージェントがこの再試行時間間隔内に再始動しない場合は、コネクタ・コントローラーはコネクタ・エージェントを再び再始動するように OAD に要求します。OAD は OADMaxNumRetry プロパティに指定した回数だけこの再試行処理を繰り返します。このプロパティを有効にするには、OADAutoRestartAgent プロパティを true に設定する必要があります。

このプロパティは、RespositoryDirectory の値が <REMOTE> に設定されている場合のみ有効です (BrokerType プロパティの値は ICS)。

デフォルト値は 10 です。

PollEndTime

PollEndTime プロパティで、イベント・キューのポーリングを停止する時刻を指定します。形式は HH:MM です。ここで、HH は 0 から 23 時、MM は 0 から 59 分です。

このプロパティには必ず有効な値を指定してください。デフォルト値は HH:MM (値なし) で、変更する必要があります。

アダプター・ランタイムが以下を検出した場合、

- PollStartTime が設定され、PollEndTime が設定されていないか、または
- PollEndTime が設定され、PollStartTime が設定されていない

PollFrequency プロパティに設定された値を使用してポーリングします。

PollFrequency

PollFrequency プロパティで、1 つのポーリング・アクションの終了と次のポーリング・アクションの開始の間の時間 (ミリ秒) を指定します。これは、ポーリング・アクション間の間隔ではありません。むしろ、以下のようなロジックです。

- ポーリングして PollQuantity プロパティの値で指定された数のオブジェクトを取得します。
- これらのオブジェクトを処理します。一部のコネクタでは、この処理の一部が別のスレッドで行われることがあります。これは次のポーリング・アクションに対して非同期に実行されます。
- PollFrequency プロパティで指定された間隔で遅延します。
- サイクルを繰り返します。

このプロパティには、以下の値が有効です。

- ポーリング・アクション間のミリ秒数 (正整数)。
- ワード no。コネクタは、ポーリングを実行しません。このワードは小文字で入力します。

- **ワード key。**コネクタは、コネクタのコマンド・プロンプト・ウィンドウで文字 `p` が入力されたときにのみポーリングを実行します。このワードは小文字で入力します。

デフォルトは 10000 です。

要確認: このプロパティの使用に関して制約事項のあるコネクタもあります。制約事項が存在する場合、それらはアダプタのインストールおよび構成の章に説明されています。

PollQuantity

PollQuantity プロパティで、コネクタがポーリングするアプリケーションからの項目数を指定します。アダプタにコネクタ固有のポーリング数設定プロパティがある場合、標準プロパティの値は、このコネクタ固有のプロパティの設定値によりオーバーライドされます。

このプロパティは、**DeliveryTransport** プロパティが **JMS** に設定されて、**ContainerManagedEvents** プロパティに値がある場合のみ有効です。

電子メール・メッセージもイベントとみなされます。コネクタは、電子メールをポーリングしている場合、以下のように動作します。

- ポーリングされると、コネクタはメッセージの本文を検出し、添付として読み取ります。この **MIME** タイプにはデータ・ハンドラが指定されていないので、コネクタはメッセージを無視します。
- コネクタは最初の **BO** 添付を処理します。この **MIME** タイプにはデータ・ハンドラが使用可能なので、コネクタはビジネス・オブジェクトを **Visual Test Connector (VTC)** に送信します。
- 2 回目にポーリングされると、コネクタは 2 番目の **BO** 添付を処理します。この **MIME** タイプにはデータ・ハンドラが使用可能なので、コネクタはビジネス・オブジェクトを **Visual Test Connector (VTC)** に送信します。
- 受け入れると、3 番目の **BO** 添付が伝送されます。

PollStartTime

PollStartTime プロパティで、イベント・キューのポーリングを開始する時刻を指定します。形式は **HH:MM** です。ここで、**HH** は 0 から 23 時、**MM** は 0 から 59 分です。

このプロパティには必ず有効な値を指定してください。デフォルト値は **HH:MM** (値なし) で、変更する必要があります。

アダプタ・ランタイムが以下を検出した場合、

- **PollStartTime** が設定され、**PollEndTime** が設定されていないか、または
- **PollEndTime** が設定され、**PollStartTime** が設定されていない

PollFrequency プロパティに設定された値を使用してポーリングします。

RepositoryDirectory

RepositoryDirectory プロパティは、コネクタが XML スキーマ文書を読み取るリポジトリの場所です。この XML スキーマ文書には、ビジネス・オブジェクト定義のメタデータが含まれています。

統合ブローカーが ICS の場合は、この値を <REMOTE> に設定する必要があります。これは、コネクタが InterChange Server リポジトリからこの情報を取得するためです。

統合ブローカーが WebSphere Message Broker または WAS の場合、この値にはデフォルトで <ProductDir>Repository が設定されています。ただし、任意の有効なディレクトリ名に設定できます。

RequestQueue

RequestQueue プロパティで、統合ブローカーがビジネス・オブジェクトをコネクタに送信するために使用するキューを指定します。

このプロパティは、DeliveryTransport プロパティの値が JMS に設定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/REQUESTQUEUE です。

ResponseQueue

ResponseQueue プロパティで、応答メッセージをコネクタ・フレームワークから統合ブローカーにデリバリーする JMS 応答キューを指定します。統合ブローカーが ICS の場合、サーバーは要求を送信し、JMS 応答キューの応答メッセージを待ちます。

このプロパティは、DeliveryTransport プロパティの値が JMS に設定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/RESPONSEQUEUE です。

RestartRetryCount

RestartRetryCount プロパティで、コネクタが再始動を試行する回数を指定します。このプロパティが並列に接続されたコネクタに使用される場合は、マスター・コネクタのアプリケーション固有のコンポーネントがクライアント・コネクタのアプリケーション固有のコンポーネントの再始動を試行する回数を指定します。

デフォルト値は 3 です。

RestartRetryInterval

RestartRetryInterval プロパティで、コネクタが再始動を試行する間隔 (単位: 分) を指定します。このプロパティが並列にリンクされたコネクタに使用される場合は、マスター・コネクタのアプリケーション固有のコンポーネントがクライアント・コネクタのアプリケーション固有のコンポーネントの再始動を試行する間隔を指定します。

このプロパティの可能な値は、1 から 2147483647 です。

デフォルト値は 1 です。

ResultSetEnabled

ResultSetEnabled プロパティーで、Information Integrator がアクティブの場合に結果セットのサポートを使用可能にするか使用不可にします。このプロパティーは、アダプターが DB2 Information Integrator をサポートする場合のみ使用できます。

このプロパティーは、DeliveryTransport プロパティーの値が JMS で、BrokerType の値が WMQI の場合のみ有効です

デフォルト値は false です。

ResultSetSize

ResultSetSize プロパティーで、Information Integrator に戻すことのできるビジネス・オブジェクトの最大数を定義します。このプロパティーは、アダプターが DB2 Information Integrator をサポートする場合のみ使用できます。

このプロパティーは、ResultSetEnabled プロパティーの値が true に設定されている場合のみ有効です。

デフォルト値は 0 です。これは、結果セットのサイズが無限であることを意味します。

RHF2MessageDomain

RHF2MessageDomain プロパティーにより、JMS ヘッダーのドメイン名フィールドの値を構成できます。JMS トランスポートを介してデータを WebSphere Message Broker に送信するときに、アダプター・フレームワークにより JMS ヘッダー情報と、ドメイン名、および固定値 mrm が書き込まれます。構成可能なドメイン名を使用すると、WebSphere Message Broker がメッセージ・データを処理する仕組みを追跡できます。

サンプル・ヘッダーを次に示します。

```
<mcd><Msd>mrm</Msd><Set>3</Set><Type>
Retek_POPhyDesc</Type><Fmt>CwXML</Fmt></mcd>
```

このプロパティーは、BrokerType の値が WMQI または WAS の場合のみ有効です。また、DeliveryTransport プロパティーの値が JMS で、WireFormat プロパティーの値が CwXML の場合も有効です。

指定可能な値は mrm および xml です。デフォルト値は mrm です。

SourceQueue

SourceQueue プロパティーで、JMS イベント・ストアを使用する JMS 対応コネクタ用の保証付きイベント・デリバリーのサポートで、コネクタ・フレームワークの JMS ソース・キューを指定します。詳細については、191 ページの『ContainerManagedEvents』を参照してください。

このプロパティーは、DeliveryTransport の値が JMS で、ContainerManagedEvents の値が指定されている場合のみ有効です。

デフォルト値は <CONNECTORNAME>/SOURCEQUEUE です。

SynchronousRequestQueue

`SynchronousRequestQueue` プロパティーで、同期応答を要求する要求メッセージを、コネクタ・フレームワークからブローカーにデリバリーします。このキューは、コネクタが同期実行を使用する場合にのみ必要です。同期実行の場合、コネクタ・フレームワークは、同期要求キューにメッセージを送信し、同期要求キューのブローカーから戻される応答を待機します。コネクタに送信される応答メッセージは、元のメッセージの ID と一致する相関 ID を持ちます。

このプロパティーは、`DeliveryTransport` の値が `JMS` の場合のみ有効です。

デフォルト値は `<CONNECTORNAME>/SYNCHRONOUSREQUESTQUEUE` です。

SynchronousRequestTimeout

`SynchronousRequestTimeout` プロパティーで、コネクタが同期要求に対する応答を待機する時間 (単位: ミリ秒) を指定します。コネクタは、指定された時間内に応答を受信しなかった場合、元の同期要求メッセージ (およびエラー・メッセージ) を障害キューに移動します。

このプロパティーは、`DeliveryTransport` の値が `JMS` の場合のみ有効です。

デフォルト値は `0` です。

SynchronousResponseQueue

`SynchronousResponseQueue` プロパティーで、同期要求に対する応答として送信される応答メッセージを、ブローカーからコネクタ・フレームワークにデリバリーします。このキューは、コネクタが同期実行を使用する場合にのみ必要です。

このプロパティーは、`DeliveryTransport` の値が `JMS` の場合のみ有効です。

デフォルトは `<CONNECTORNAME>/SYNCHRONOUSRESPONSEQUEUE` です。

TivoliMonitorTransactionPerformance

`TivoliMonitorTransactionPerformance` プロパティーで、`IBM Tivoli Monitoring for Transaction Performance (ITMTP)` を実行時に起動するかどうかを指定します。

デフォルト値は `false` です。

WireFormat

`WireFormat` プロパティーで、トランスポート上のメッセージ・フォーマットを指定します。

- `RepositoryDirectory` プロパティーの値がローカル・ディレクトリーの場合、値は `CwXML` です。
- `RepositoryDirectory` プロパティーの値がリモート・ディレクトリーの場合、値は `CwB0` です。

WsifSynchronousRequestTimeout

`WsifSynchronousRequestTimeout` プロパティーで、コネクターが同期要求に対する応答を待機する時間 (単位: ミリ秒) を指定します。コネクターは、指定された時間内に応答を受信しなかった場合、元の同期要求メッセージ (およびエラー・メッセージ) を障害キューに移動します。

このプロパティーは、`BrokerType` の値が `WAS` の場合のみ有効です。

デフォルト値は `0` です。

XMLNamespaceFormat

`XMLNamespaceFormat` プロパティーで、ビジネス・オブジェクト定義の XML フォーマットの短縮または長いネーム・スペースを指定します。

このプロパティーは、`BrokerType` の値が `WMQI` または `WAS` の場合にのみ有効です。

デフォルト値は `short` です。

第 8 章 データベース接続プールの構成

実行時にプロセスがデータベースへの新規接続を確立するには、時間がかかります。コラボレーション・プロセスとマップ・プロセスによって使用できるように事前にデータベース接続プールを確立することによって、新規接続の確立にかかる時間を短縮できます。データベース接続プールによってパフォーマンスが向上し、コラボレーション内から直接データベースにアクセスできるようになります。

1 つのデータベース接続プールは、複数のコラボレーションおよびマップで使用することができます。また、各コラボレーションまたはマップは、複数のデータベース接続プールを使用できます。

この章を構成するセクションは次のとおりです。

- 『データベース接続プールを使用する状況』
- 210 ページの『データベース接続プールを使用しない状況』
- 211 ページの『データベース接続プールとデータベース接続の作成』
- 214 ページの『データベース接続プールの検証』
- 215 ページの『データベース接続プールの変更』
- 216 ページの『コラボレーションおよびマップでのデータベース接続プールの使用』
- 217 ページの『トランザクション・ブラケットの構成』

データベース接続プールを使用する状況

このセクションでは、データベース接続プールが役立つような状況をいくつか示します。

ルーティングの実行

ビジネス・オブジェクト内の 1 つ以上のフィールドの値に応じて、ビジネス・オブジェクトを別の宛先アプリケーションへ経路指定することが、ビジネス・プロセスのロジックで指示される場合があります。

例えば、CustomerType などの属性の値に応じて、異なるアプリケーションでサイトが顧客エンティティを保管および処理する場合があります。この場合、コラボレーション・テンプレートは、その属性の値を検索および評価し、その値に基づいてどの宛先アプリケーションへビジネス・オブジェクトを送信するかを決定する必要があります。Java ではこれを制御フロー構造で実行できますが、値と宛先アプリケーションのペア化はコラボレーション・テンプレート内にハードコーディングされます。プロシージャの変更のために変更が必要な場合や新しい値やアプリケーションがインターフェースに導入されたために追加が必要な場合は、コラボレーション・テンプレートを変更および再コンパイルして、再配置する必要があります。それよりも柔軟な実装では、値と宛先アプリケーションのペア化がデータベース表に保管されます。そうした方法を実装するには、以下の手順を実行します。

1. ルーティング値を保管する列と、適切な宛先アプリケーションをルーティング値と関連付ける情報を保管する列を持つデータベース表を作成します。表 14 に、そのような表の例を示します。

表 14. ルーティング表の例

ルーティング値	宛先アプリケーション値
Customer	AppA
Federal	AppB
Reseller	AppC
Academic	AppD

2. 211 ページの『データベース接続プールとデータベース接続の作成』の説明に従って、System Manager 内にデータベース接続プールとデータベース接続を作成します。
3. コラボレーション・テンプレートで、次の操作を実行するロジックを設計します。
 - a. ルーティングに使用するため属性の値を検索し、その値を変数に保管します。
 - b. データベースへの接続を取得します。
 - c. 宛先アプリケーション値を保管した列内の値を検索する SQL 照会を実行します。ここで、ルーティング値を保管する列内の値は、コラボレーション・テンプレート内の変数に保管された値と等しくなります。
 - d. 決定ノードを使用し、データベース表から戻された値に応じて、コラボレーション・ロジックを分岐させます。各分岐は、アプリケーション・コネクタへのビジネス・オブジェクトの送信、すなわち適切な宛先アプリケーションへのビジネス・オブジェクトの送信を担当する、別々のサービス呼び出しノードにつながります。

データベース表でのルックアップの実行

表内で同等の値をルックアップすることによって、値を変換しなければならない場合もあります。多くの場合、こうした操作はルックアップ関係の実装によって行いますが、ルックアップ関係の使用が常に意味をなすわけではありません。ルックアップ関係は、インターフェースに含まれた各アプリケーションが独自の方法でデータの断片を表現する必要があるような状況のために特に設計されています。そのため、各アプリケーションごとに参加者が作成され、統合ブローカーが各アプリケーションを接続するのとほぼ同じ方法で、ルックアップ関係自体が各参加者を接続します。値を他の値のいずれかに変換しなければならないが、インターフェースに含まれる各アプリケーションのためにそのデータの個別の表現を保持する必要はないという場合もあります。そのような場合は、関連する値を保管するための表をリポジトリ内に作成し、データベース接続と SQL SELECT ステートメントを使用して目的の値を検索する必要があります。

さらに、ルックアップ関係用の API を使用すると、アプリケーション間で関連データを容易に抽出することができますが、より複雑な照会の助けにはなりません。ルックアップ関係 API は、データの断片を受け取り、そのデータが関係内で他のアプリケーション・データの断片と共有しているキー値を戻すか、またはキー値を受け取り、それに関連したデータの断片を戻すために設計されています。ただし、ルッ

クアック関係 API は、複数の列値を戻すことや、ストアド・プロシージャーを実行することはできません。これは、CwDBCConnection クラスの API が実行できません。

この要件は、データベース照会ではなく、「if/else」ステートメントや「switch/case」ステートメントなどの制御構造を使用することによって Java コードで満たすこともできます。それぞれの方法について次に示す利点と欠点を考慮し、状況に応じて適切な方法を選択してください。

- データベース接続を使用している場合は、データベースに新規行を追加すればよいだけなので、新規レコード (新規宛先アプリケーションとペアになった新規ルーティング値など) を追加した方が簡単です。コンポーネントを変更する必要はありません。新規ペアはすぐに有効になります。ただし、Java コードでは、コンポーネントのコードを変更し、それを再コンパイルして、場合によっては停止し再始動する必要があるため (コンポーネントがコラボレーションである場合)、新しい関連の追加はそれほど便利ではありません。また、インターフェースのコンポーネントを事前に一時停止し、コンポーネントのコンパイル中に進行中のトランザクションがないことを確認する必要があります。そうしないと、失敗する場合があります。
- Java コードでのロックアップは、データベースへの接続を確立して照会を実行するよりも速く実行できますが、これは、Java 制御構造が繰り返すべき関連がいくつあるかによって異なります。
- データベース接続を使用するインターフェースを別の環境へ移行することは、Java コードだけに依存するインターフェースを移行する場合に比べて、あまり便利ではありません。これは、接続を確立するマップまたはコラボレーションの移行だけでなく、新しい環境に含まれる各表の再作成も忘れずに行う必要があるためです。

情報の永続化

ビジネス・インテグレーション・システムのオペレーションに関する情報を、データベースに保管して永続的なものにし、問題解決や履歴分析のために参照できるようにしたいと考える顧客もいます。

この要件を満たすには、以下の手順を実行します。

1. 目的のデータを保管するのに必要なだけの列を持つデータベース表を作成します。
2. 211 ページの『データベース接続プールとデータベース接続の作成』の説明に従って、System Manager 内にデータベース接続プールとデータベース接続を作成します。
3. コラボレーション・テンプレートで、データベースへの接続を取得し、目的のデータを適切な列へ挿入する SQL 照会を実行するロジックを設計します。

通常、こうした要件には、コラボレーションが処理中のビジネス・オブジェクトに含まれる情報 (処理中のエンティティの基本キーなど) またはシステム自体に関する情報 (ビジネス・オブジェクト要求の正常な処理など) の永続化が含まれます。リリース 4.2.0 では、ビジネス・オブジェクト・プローブ機能を使用して、基本キー値などのビジネス・オブジェクト・データを永続化することができます。詳細については、「コラボレーション開発ガイド」を参照してください。リリース 4.2.0 で

は、永続モニター機能を使用して、システムに関する情報の一部を永続化することができます。詳細については、「システム管理ガイド」を参照してください。

データベース接続プールを使用しない状況

アプリケーションのデータベースへ直接接続する場合は、データベース接続プールを使用すべきではありません。アプリケーション・データベースと対話するには、IBM WebSphere Business Integration アダプターのみを使用してください。アダプターは、アプリケーションのプログラミング・インターフェースが提供するビジネス・ロジックを使用するので、データベース接続は、アプリケーションをサポートしていないデータベースに対してのみ確立されます。例えば、SQL UPDATE ステートメントを実行するためにアプリケーション・データベースへ直接接続すると、API が更新操作に応じて実行するはずの関連ロジックを妨げてしまいます。これにより、アプリケーションとビジネス・プロセスの整合性が失われます。

アプリケーションから情報を検索する必要があるが、アプリケーションへのビジネス・オブジェクト要求の送信と応答の受信のパフォーマンスに影響を与えるのでアダプターは使用したくないという場合は、いくつかの代案があります。

- 操作で処理する必要のあるフィールドに対する属性だけを持つカスタム・ビジネス・オブジェクトを設計できます。例えば、特定のビジネス・プロセスに含まれるアプリケーション・エンティティー用に設計された大規模なアプリケーション固有のビジネス・オブジェクトがあるとして、そのエンティティーに関する情報のうち小規模なサブセットのみを別のビジネス・プロセスの一部として取り出す必要がある場合は、その少数のフィールドのためにビジネス・オブジェクト全体を取り出すことによるパフォーマンスへの影響が気にかかるでしょう。この影響を最小にするために、アプリケーション固有のビジネス・オブジェクトのコピーを作成し、その構造体を、必要なフィールドだけに縮小することができます。そうすれば、処理する必要があるのは少量の情報だけになります。

これは、各要求が、その要求時に最新の情報を検索するので、頻繁に変化するアプリケーション・データベース内の情報検索に最適な方法です。

- IBM WebSphere InterChange Server データベースのホストであるサーバーで、アプリケーション情報を複製することができます。データベース・ベンダーが提供するユーティリティーを使用し、表の構造とデータをファイルヘスクリプト化して、アプリケーション自体が使用している以外のデータベースで表およびそのデータを複製することができます。

この方法は、参照表など、フラットで静的な少量の情報を検索する場合に最も有用です。しかし、複数の表に渡る大規模なエンティティーでは、照会の作成と維持が難しくなるため、この方法は使用しないでください。一方、ビジネス・オブジェクトを使用したエンティティーの表示は、開発と維持が容易です。また、新規レコードが頻繁に追加される動的な表では、複製された表を手動で頻繁に更新しなければならず、それを怠るとインターフェースの情報が古くなってしまいうため、この方法は適していません。

データベース接続プールとデータベース接続の作成

データベース接続プールは、予約された多数のデータベース接続によって構成されています。予約されたデータベース接続を利用できるのは、そのプールを使用するように設計されたコラボレーション・プロセスとマップ・プロセスのみです。

データベース接続プールを作成するには、データベース接続を確立するのに必要な値を事前に定義します。IBM WebSphere InterChange Server Business Integration システムでは、このデータベース接続情報が保管され、割り当てたコラボレーション・プロセスとマップ・プロセスに対してすばやく接続が確立できるように、実行時にこの情報が使用されます。

定義するデータベース接続値の構成は、1 つ以上のプールによって使用されることがあります。各プールに対して接続の数を指定します。これらの接続は割り当てられて使用された後、プールに戻されます。

注: プールを作成する前に、System Manager 内のデータベース接続プールを使用する InterChange Server インスタンスに接続することをお勧めします。こうすると、接続を検証することができます。

データベース・プールを作成するには、以下の手順を実行します。

1. System Manager の統合コンポーネント・ライブラリーで「データベース接続プール」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「新規データベース接続の作成」を選択します。

「データベース接続」ダイアログが表示されます。

注: このダイアログの名前から判断すると、この時点でデータベース接続を作成しているように思われますが、実際には、データベース接続プールを作成しています。後続のステップで、このプール内でデータベース接続を定義します。

2. 「データベース・ドライバー」ドロップダウン・メニューから、データベース・ベンダーに応じて適切な値を選択します (DB2 (Type 2)、MS SQL Server(Type 4)、または Data Direct Oracle(Type 4))。
3. 「データベース・ドライバー」ドロップダウン・メニューから MS SQL Server(Type 4) または Data Direct Oracle(Type 4) を選択した場合は、データベース・サーバーが常駐するコンピューターの名前を「ホスト名」フィールドに入力します。
4. 「データベース」フィールドに、データベースの名前を入力します。
5. 「データベース・ドライバー」ドロップダウン・メニューから MS SQL Server(Type 4) または Data Direct Oracle(Type 4) を選択した場合は、クライアントがデータベース・サーバーと通信するポート番号を「ポート番号」フィールドに入力します。
6. プールを使用する InterChange Server インスタンスに接続している場合、「接続サーバー」ドロップダウン・メニューからインスタンスを選択します。
7. 接続が切断されたか確立できない場合、「最大接続試行回数」フィールドに、WebSphere InterChange Server がデータベース接続の回復を試行する回数を入力します。

8. 「**DBConnection 名**」フィールドにプールの名前を入力します。マップまたはコラボレーション・テンプレートで接続を確立するための Java コードを記述する際に、このデータベース接続プール名を指定します。
9. 「**ログイン**」フィールドに、InterChange Server が指定のデータベースへのログインに使用するユーザー名を入力します。
10. 「**パスワード**」フィールドに、ステップ 6d (123 ページ) で指定したユーザー名のパスワードを入力します。
11. プール内で作成する個々のデータベース接続オブジェクトすべてに対してこのプールが確立する接続の最大数を「**最大接続数**」フィールドに入力するか、「**無制限**」チェック・ボックスをオンにして、データベース・サーバー構成とライセンスが許可する限りの接続を確立させます。

警告: このフィールドを処理する際は注意してください。これは単一行のテキスト・フィールドではありませんが、単一行の値を指定する場合に、誤って **Enter** キーを押す可能性があります。その場合、入力した値は表示されなくなるため、おそらく値を再入力することになります。そうすると、データベース接続プールの作成を完了しようとしたときに、このフィールドに有効な値を入力する必要があるというエラーが表示されます。このフィールドでは、**Enter** キーを押さないでください。

12. 「**接続再試行間隔**」フィールドに、InterChange Server がデータベース接続の回復を試行する間に待機する時間 (秒) を指定します。

213 ページの図 50 に、「新規データベース接続プール (New Database Connection Pool)」ウィザードを示します。

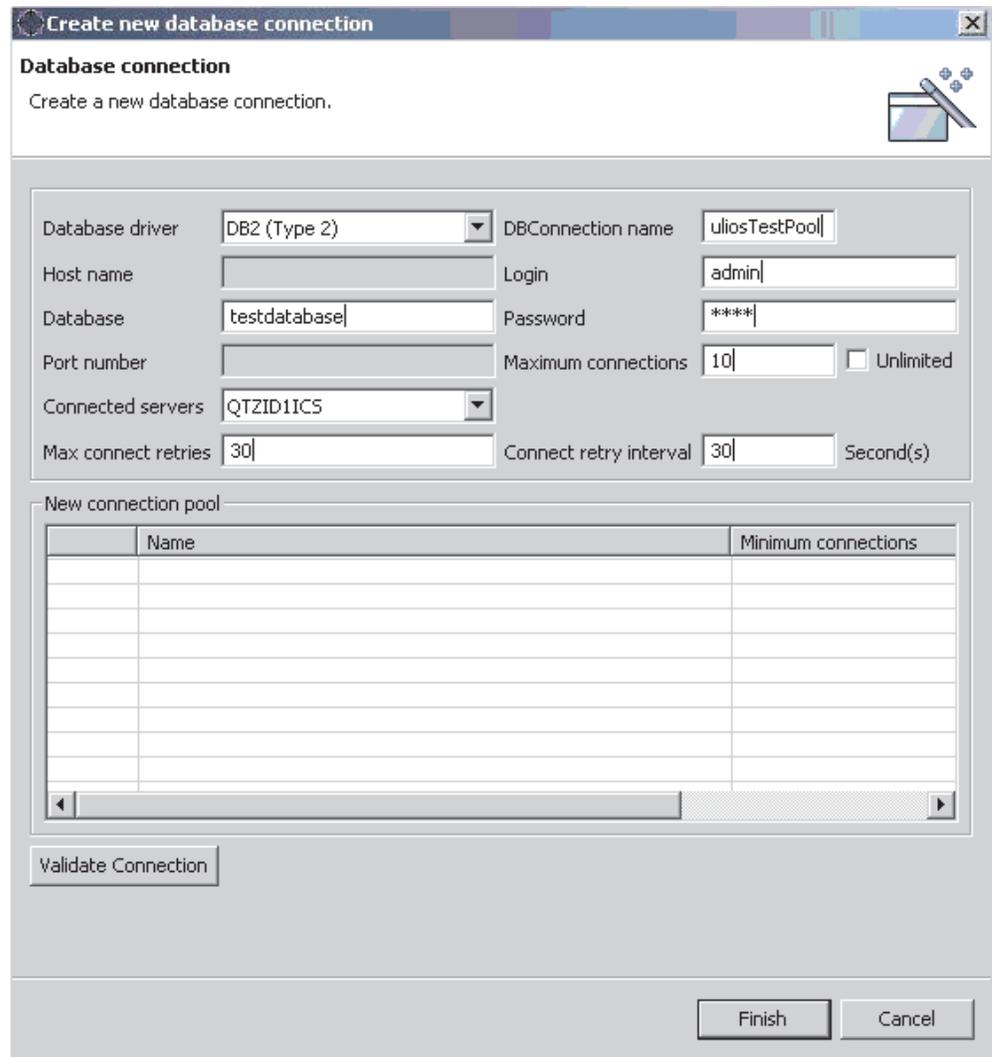


図 50. 「新規データベース接続の作成」ウィザード

13. プール内で新規データベース接続オブジェクトを作成するには、ウィザードの下部にある「新規接続プール」表内の行を右マウス・ボタンでクリックして、コンテキスト・メニューから「新規接続プール」を選択します。

注: このインターフェースから判断すると、この時点でもデータベース接続プールを作成しているように思われますが、実際には、前の手順で作成したプール内でデータベース接続オブジェクトを作成しています。

「接続プール」ダイアログが表示されます。

14. 「名前」フィールドにデータベース接続オブジェクトの名前を入力します。
15. 「最小接続数」フィールドに、データベース接続オブジェクトが確立する接続の最小数を入力します。作成する全プールの最小数の合計が、データベース接続プール・オブジェクト自体に対して指定した最大数を超えることはできません。System Manager は、プールの接続の最大数が超過するような接続オブジェクトの追加や既存の接続オブジェクトの変更は許可しません。214 ページの図 51 に、「接続プール」ダイアログを示します。



図 51. 「接続プール」ダイアログ

16. 「完了」をクリックします。

System Manager がデータベース接続プール・オブジェクトを保管し、統合コンポーネント・ライブラリーの「データベース接続プール」フォルダーの下にアイコンが表示されます。

17. ここでウィザードを使用して別のデータベース接続プールを作成するか、「キャンセル」をクリックしてウィザードを閉じます。

データベース接続プールの検証

データベース接続プールの作成後、それを検証して、マップおよびコラボレーションが実行時に使用できることを確認する必要があります。

データベース接続プールを検証するには、以下の手順を実行します。

1. InterChange Server インスタンスを開始します。
2. 37 ページの『InterChange Server への接続』の説明に従って、System Manager を InterChange Server インスタンスに接続します。
3. 以下のいずれかを実行します。
 - 統合コンポーネント・ライブラリーでデータベース接続プール・オブジェクト、またはユーザー・プロジェクトでそのショートカットを右マウス・ボタンでクリックし、コンテキスト・メニューから「接続を検証」を選択し、サブメニューから InterChange Server インスタンス名を選択します。
 - InterChange Server コンポーネント管理ビューで、サーバー・リポジトリ内のデータベース接続プール・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「接続を検証」を選択します。
4. データベース接続プールが妥当である場合は、System Manager によって、検証が成功したことを示すプロンプトが出されます。

System Manager によって、検証が成功しなかったというエラーが表示された場合は、データベースの可用性、指定したユーザー・アカウントの特権、および構成情報の正確性を検査して問題を解決します。

データベース接続プールの変更

データベース接続プールとデータベース接続オブジェクトの両方で、いくつかのプロパティを変更できます。

データベース接続プールを変更するには、次のいずれかを実行します。

- 統合コンポーネント・ライブラリーでデータベース接続プール・コンポーネントを右マウス・ボタンでクリックするか、ユーザー・プロジェクトでショートカットを右マウス・ボタンでクリックして、コンテキスト・メニューから「プロパティ」を選択します。
- 統合コンポーネント・ライブラリーでデータベース接続プール・コンポーネントを選択するか、ユーザー・プロジェクトでショートカットを選択して、次のいずれかを実行します。
 - メニュー・バーから「コンポーネント」>「プロパティ」を選択します。
 - **Alt+Enter** を押します。

次のセクションでは、データベース接続プールとデータベース接続オブジェクトで変更できるプロパティを示します。

- 『ログイン情報の変更』
- 『接続数の変更』

ログイン情報の変更

古いアカウントを使用してセキュリティが侵犯されるのを防ぐため、多くの企業が、定期的にパスワードを変更する習慣を持っています。サイトの場合は、データベース接続プールに関するログイン情報の変更がこれにあたります。

データベース・リソースへのアクセスに使用するログイン名を変更するには、「データベース接続」画面の「ログイン」フィールドに新しい値を入力する必要があります。

指定したログインのパスワードを変更するには、「データベース接続」画面で「変更」をクリックし、「パスワードの変更」プロンプトで以下の手順を実行します。

1. 「旧パスワード」フィールドに現在のパスワードを入力します。
2. 「新パスワード」フィールドに新規パスワードを入力します。
3. 「確認パスワード」フィールドに再度新規パスワードを入力します。

接続数の変更

データベース接続プールの接続の最大数とデータベース接続オブジェクトの接続の最小数を変更できます。データベース接続プールで定義された全データベース接続オブジェクトの最小接続数の合計が、そのプール自体に指定された接続の最大数を超えることはできません。

データベース接続プールの最大接続数の変更

データベース接続プールの最大接続数を変更するには、以下の手順を実行します。

1. 『データベース接続プールの変更』の説明に従って、編集するデータベース接続プールのプロパティにアクセスします。

2. プール内の接続オブジェクトの合計数から算出される接続の最大数を変更するには、次のいずれかを実行します。
 - 「最大接続数」フィールドに、プールが確立すべき接続の最大数を入力します。
 - 「無制限」チェック・ボックスをオンにして、データベース・サーバー構成とライセンスが許可する限りの接続を確立させます。
3. 「OK」をクリックします。

データベース接続オブジェクトの最小接続数の変更

接続オブジェクトの最小接続数を変更するには、以下の手順を実行します。

1. 215 ページの『データベース接続プールの変更』の説明に従って、編集するデータベース接続プールのプロパティにアクセスします。
2. 「新規接続プール」ペインで、最小接続数を変更する接続オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「編集」を選択します。
3. 「最小接続数」フィールドに新しい値を入力します。
4. 「OK」をクリックします。

データベース接続回復力設定の変更

データベース接続プールのデータベース接続回復力の設定を変更するには、以下の手順を実行します。

1. 215 ページの『データベース接続プールの変更』の説明に従って、編集するデータベース接続プールのプロパティにアクセスします。
2. 接続が切断されたか確立できない場合、「最大接続試行回数」フィールドに、WebSphere InterChange Server がデータベース接続の回復を試行する回数を入力します。「接続再試行間隔」フィールドに、InterChange Server がデータベース接続の回復を 2 回試行する間に待機する時間 (秒) を指定します。
3. 「OK」をクリックします。

データベース接続オブジェクトの削除

データベース接続オブジェクトを削除するには、以下の手順を実行します。

1. 215 ページの『データベース接続プールの変更』の説明に従って、データベース接続プールのプロパティにアクセスします。
2. 「新規接続プール」ペインで、削除する接続オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「削除」を選択します。
3. 「OK」をクリックします。

コラボレーションおよびマップでのデータベース接続プールの使用

マップおよびコラボレーション・テンプレートでデータベース接続プールを使用するには、BaseCollaboration クラスの `getDBConnection()` メソッドを使用します。

```
CwDBConnection getDBConnection(String ConnectionPoolName);
```

ストリング `ConnectionPoolName` は、System Manager で作成したプール名またはこれから作成するプール名に完全に一致する必要があります。名前が完全に一致しない場合、あるいはコラボレーション・テンプレートまたはマップで接続プール名を

設定しても System Manager でデータベース接続プールを作成しない場合、そのコラボレーション・オブジェクトまたはマップは失敗します。

コラボレーションでは、テンプレートからコラボレーション・オブジェクトを作成する前でも作成した後でも、テンプレートでプール名を設定できます。新規接続プールの名前に一致する新規コラボレーション・オブジェクトを再構成する必要はありません。

データベース接続プールを使用するためのコラボレーション・テンプレート開発の詳細については、「[コラボレーション開発ガイド](#)」を参照してください。

データベース接続プールを使用するためのマップ開発の詳細については、「[マップ開発ガイド](#)」を参照してください。

トランザクション・ブラケットの構成

暗黙的なデータベース・トランザクション・モード (接続が確立されるとただちにトランザクションが開始され、プロセスが終了するとトランザクションが終了) または明示的なデータベース・トランザクション・モード (トランザクションの開始と終了は、ユーザーがプログラムで設定) のいずれかでデータベース接続プールを使用するために、コラボレーションまたはマップを構成できます。

コラボレーション・オブジェクト用のトランザクション・ブラケットを構成するには、237 ページの『[暗黙的なデータベース・トランザクション](#)』を参照してください。

マップ用のトランザクション・ブラケットを構成するには、「[マップ開発ガイド](#)」を参照してください。

第 9 章 コラボレーション・オブジェクトの構成

この章では、コラボレーション・オブジェクトの構成について説明します。この章は次のセクションから構成されます。

- 『コラボレーション・オブジェクトとそのグループ』
- 220 ページの『コラボレーション・オブジェクトの作成』
- 225 ページの『コラボレーション・オブジェクトの変更』

コラボレーション・テンプレートの設計と作成、および Process Designer の使用については、「コラボレーション開発ガイド」を参照してください。

コラボレーション・オブジェクトとそのグループ

コラボレーション・オブジェクトはコラボレーション・テンプレートのインスタンスです。コラボレーションを構成するには、以下の手順に従います。

1. コラボレーション・テンプレートから新規コラボレーション・オブジェクトを作成します。詳細については、220 ページの『コラボレーション・オブジェクトの作成』を参照してください。
2. 内部的に (1 つのコネクターまたは別のコラボレーションへ) または外部的に (e-business Web アプリケーション・サーブレットなどのコラボレーションを呼び出す 1 つの外部プロセスへ) コラボレーション・オブジェクトのポートをバインドします。詳細については、228 ページの『コラボレーション・オブジェクト・ポートのバインディング』を参照してください。
3. コラボレーション・オブジェクトの一般プロパティを構成します。詳細については、231 ページの『コラボレーション・オブジェクトの一般プロパティの構成』を参照してください。
4. コラボレーション・オブジェクトのコラボレーション固有のプロパティを構成します。詳細については、239 ページの『コラボレーション固有のプロパティの構成』を参照してください。

内部バインドを作成するとき、コラボレーション・ポートによって予期されるビジネス・オブジェクトをサポートするコネクターまたはコラボレーションにのみバインドできます。これによってバインドされたコンポーネント間での通信が可能となり、コラボレーションは、コラボレーション・ビジネス・オブジェクトを送受信できるだけでなく、要求への応答も受信できるようになります。ポートをすべてバインドするまで、コラボレーションは実行されません。

1 つのコラボレーション・テンプレートから複数のコラボレーション・オブジェクトを作成できます。例えば、同じテンプレートから作成された 2 つのコラボレーションによって、サイトにある 2 セット of アプリケーション上で同じロジックを実装できます。

コラボレーション・オブジェクト・グループ

コラボレーション・オブジェクト・グループは、それぞれが相互にバインドされた 2 つ以上のコラボレーション・オブジェクトのセットです。

別のコラボレーション・オブジェクトからの着信ビジネス・オブジェクトの受信や別のコラボレーション・オブジェクトへの発信ビジネス・オブジェクトの送信ができるようコラボレーション・オブジェクトのポートを構成するときに、これらのコラボレーション・オブジェクトの組み合わせがコラボレーション・オブジェクト・グループとなります。任意の数のコラボレーション・オブジェクトを 1 つのグループにバインドできます。

あるコラボレーション・オブジェクト・グループのメンバーであるコラボレーション・オブジェクトの振る舞いは、グループのメンバーでないコラボレーション・オブジェクトの振る舞いとは異なります。1 つのグループのメンバーであるコラボレーション・オブジェクトに `start`、`pause`、または `stop` などのコマンドを適用すると、コマンドはグループのすべてのメンバーに対して有効となります。

グループの 1 つのメンバーをトランザクションとして実行するには、コラボレーション・オブジェクト・グループのすべてのメンバーが同じトランザクション・レベルをサポートしている必要があります。トランザクション・レベルが「なし」のコラボレーション・オブジェクトをトランザクション・レベルが「最大限の努力」のコラボレーション・オブジェクトにバインドする場合には、両方のコラボレーション・オブジェクトともレベル「なし」で実行します。

コラボレーション・オブジェクトの作成

このセクションでは、System Manager ウィザードを使用して新規コラボレーション・オブジェクトを作成する方法について説明します。ここでは、ウィザードの一般的な流れを説明します。また、個々のコラボレーション・オブジェクト構成については、この章内の他のセクションを参照先として示します。ウィザードを使用してコラボレーション・オブジェクトを作成するには、以下の手順を実行します。

1. System Manager で、統合コンポーネント・ライブラリーの「**コラボレーション・オブジェクト**」フォルダーを右マウス・ボタンでクリックし、コンテキスト・メニューから「**新規コラボレーション・オブジェクトの作成**」を選択します。

「新規コラボレーションの作成」ウィザードが表示され、インストールされているテンプレートが「**テンプレート名**」列に示され、(開発者が入力した場合は)「**説明**」列に説明が示されます。このダイアログは、コラボレーション・オブジェクトを最初に作成して構成する手順を段階的に説明するウィザードの最初の画面です。構成値は後から変更できます。221 ページの図 52 に、「コラボレーション・テンプレートの選択」画面を示します。

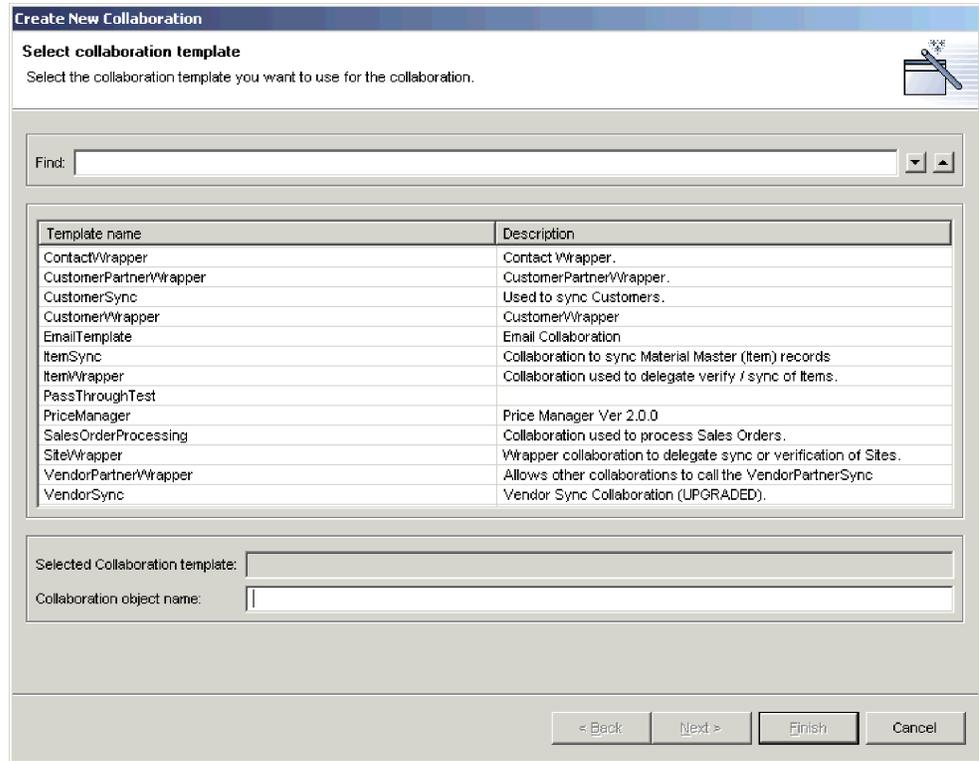


図 52. コラボレーション・テンプレートの選択

2. テンプレートのリストから、オブジェクトの基になるコラボレーション・テンプレートを選択します。選択した名前が、リストの下にある「**選択したコラボレーション・テンプレート**」フィールドに表示されます。

「検索」フィールドにテキストを入力して、その文字から始まる名前をもつコラボレーション・テンプレートをダイアログに表示させることもできます。該当する名前がいくつかある場合は、上矢印と下矢印を使用して、その文字パターンを共有するテンプレート名を見ていきます。

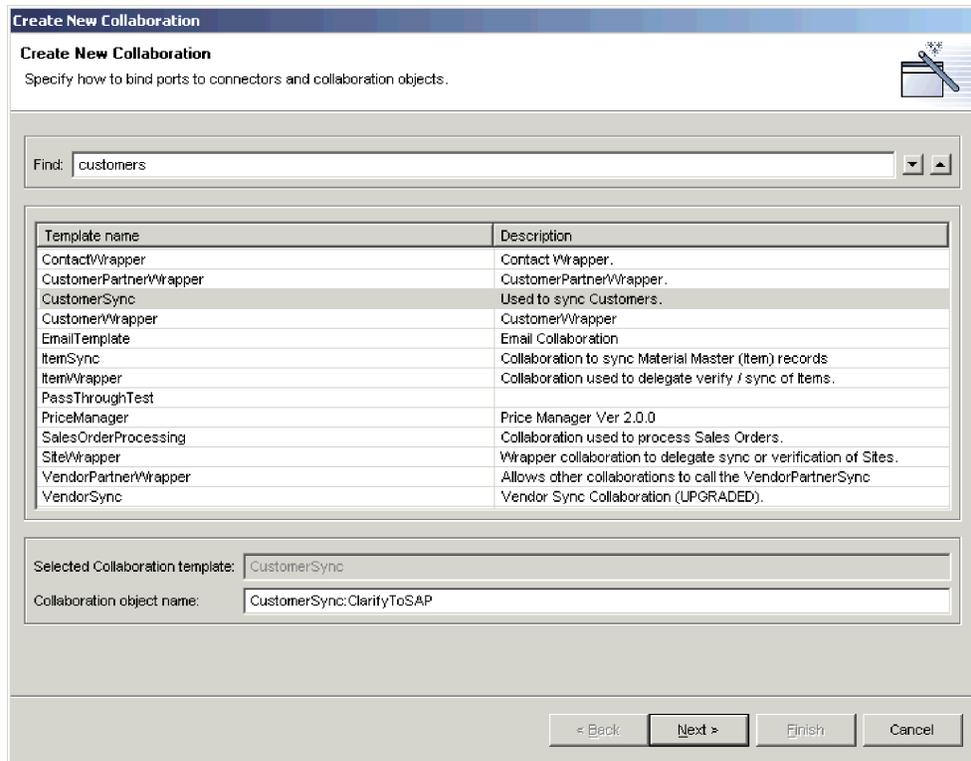


図 53. 「検索」フィールドの使用

3. 「コラボレーション・オブジェクト名」フィールドにコラボレーション・オブジェクトの名前を入力します。

コラボレーション・オブジェクト名に関する要件および規則については、「コンポーネント命名ガイド」を参照してください。

4. 「次へ」をクリックして、ウィザードの「ポートをバインド」画面に進みます。

「ポートをバインド」画面には、コラボレーション・テンプレートで定義されているポート、各ポートによってサポートされているビジネス・オブジェクト定義、ポートがバインドされている可能性のあるコンポーネントのタイプ、および各ポートをバインドできる指定されたタイプの特定コンポーネントが表示されます。223 ページの図 54 に、「ポートをバインド」画面を示します。

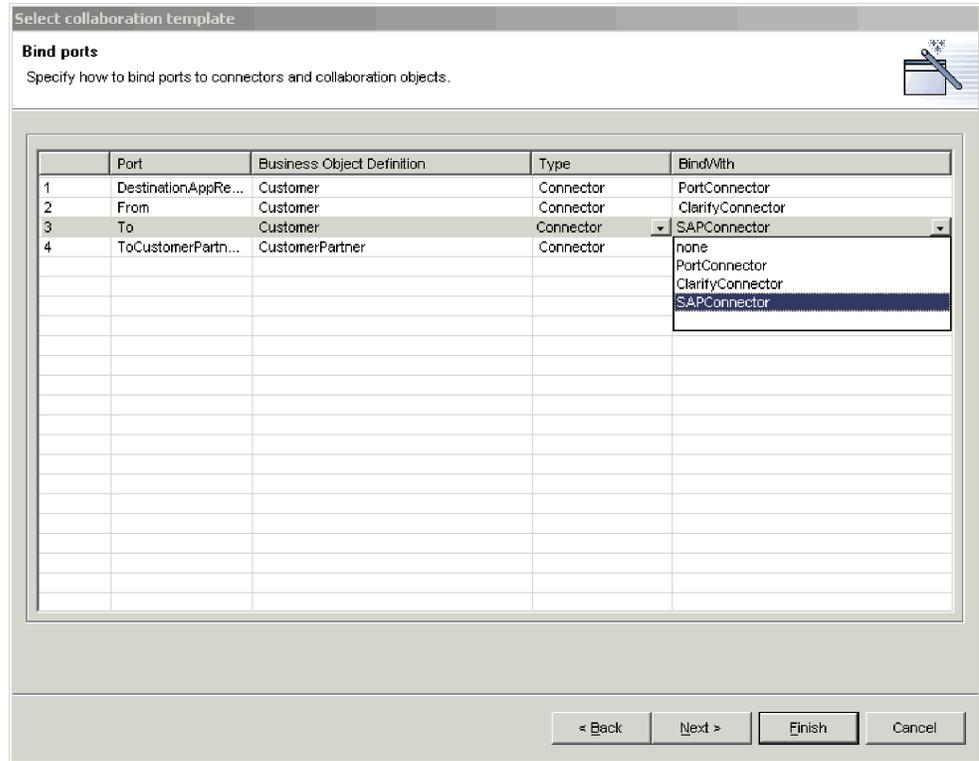


図 54. コラボレーション・ポートのバインディング

「タイプ」列で選択するオプションに応じて、「バインド先」列に、「ビジネス・オブジェクト定義」フィールド内のビジネス・オブジェクト・タイプをサポートするコネクタまたはコラボレーションが表示されます。表示されているポートごとに、「タイプ」列から、「コラボレーション」、「コネクタ」、または「Web サービス」を選択してから、「バインド先」列で、目的のコンポーネントを選択します。

外部のプログラマチック・エンティティ（サブレットなど）からビジネス・オブジェクト要求を受信できるようにポートを構成するには、230 ページの『外部ポート・バインディングの構成』の指示に従う必要があります。

ウィザードのこの画面では、一部またはすべてのポートをバインドしても、ポートをまったくバインドしなくてもかまいません。この時点ですべてのポートをバインドしない場合は、ウィザードの終了後、いつでもポートのバインディングを変更できます。したがって、ポートにバインドする必要がある特定のコンポーネントが必要なビジネス・オブジェクト定義をサポートしていない場合は、コラボレーション・オブジェクト・ウィザードを終了し、必要なコンポーネントにビジネス・オブジェクト定義のサポートを追加してから、コラボレーション・オブジェクトを再構成することができます。

ポートのバインディングの詳細およびポートのバインディングの変更方法については、228 ページの『コラボレーション・オブジェクト・ポートのバインディング』を参照してください。

5. 「次へ」をクリックして、ウィザードの「コラボレーションの一般プロパティ」画面に進みます。この画面では、WebSphere InterChange Server Business

Integration で定義されたすべてのコラボレーション・テンプレートに属するプロパティを構成できます。図 55 に、「コラボレーションの一般プロパティ」画面を示します。

The screenshot shows a 'Properties' dialog box titled 'Collaboration general properties'. It contains the following fields and options:

- Create from template:** A text field containing 'CustomerSynchronization'.
- Effective transaction level:** A dropdown menu set to 'None'.
- Minimum transaction level:** A text field containing 'None'.
- System trace level:** A dropdown menu set to '0 - No tracing'.
- Collaboration trace level:** A dropdown menu set to '0'.
- E-mail notification address:** An empty text field.
- Pause when a critical error occurs:** An unchecked checkbox.
- Maximum number of concurrent events:** A dropdown menu set to '1'.
- Persist service call in transit state:** An unchecked checkbox.
- Recovery mode:** A dropdown menu set to 'Always'.
- Implicit database transaction:** A checked checkbox.
- Maximum event capacity:** A text field containing '2147483647'.
- Blocking type:** A checked checkbox.
- Event isolation:** A checked checkbox.

At the bottom right, there are 'OK' and 'Cancel' buttons.

図 55. コラボレーション・オブジェクトの一般プロパティの構成

各プロパティを目的の値に設定します。

コラボレーションの一般プロパティの詳細および既存のコラボレーション・オブジェクトの一般プロパティの変更方法については、231 ページの『コラボレーション・オブジェクトの一般プロパティの構成』を参照してください。

6. 「次へ」をクリックして、ウィザードの「プロパティ」画面に進みます。この画面では、特定のコラボレーション・テンプレートに固有のプロパティを構成できます。

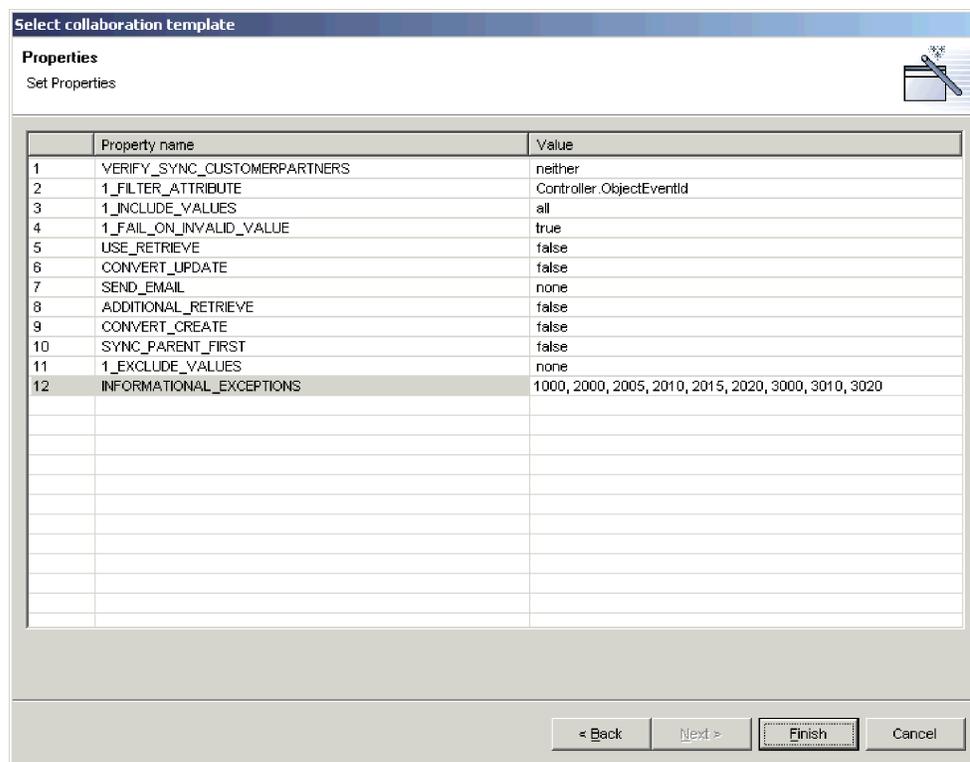


図 56. コラボレーション・テンプレート固有のプロパティの構成

各プロパティの値を入力するか、デフォルト値を受け入れます。各プロパティについては、特定のコラボレーション・テンプレートの資料を参照してください。

「完了」をクリックして、コラボレーション・オブジェクト作成のウィザードを終了します。新規コラボレーション・オブジェクトが「グラフィック表示」タブに表示されます。コラボレーション・オブジェクトのグラフィック表示およびツリー表示については、226 ページの『コラボレーション・オブジェクトの表示』を参照してください。

コラボレーション・オブジェクトの変更

このセクションでは、コラボレーション・オブジェクトの変更方法について説明します。コラボレーション・オブジェクトのポート・バインディングまたはプロパティを変更する理由としては、次のことが考えられます。

- 220 ページの『コラボレーション・オブジェクトの作成』の説明に従い、ウィザードを使用してコラボレーション・オブジェクトを作成しましたが、外部のプログラマチック・エンティティにポートをバインドする必要に気が付きました。その操作は、ウィザードを使用して行うことはできません。
- 220 ページの『コラボレーション・オブジェクトの作成』の説明に従い、ウィザードを使用してコラボレーション・オブジェクトを作成したが、必要なコンポーネントが、ポートに関連したビジネス・オブジェクト定義をサポートしていないため。この場合は、コンポーネント定義を変更する必要があります。

- 別の統合コンポーネント・ライブラリーまたは InterChange Server からコラボレーション・オブジェクト定義をインポートしたため。この場合は、環境固有の変更を行う必要があります。
- 実稼働環境で正常に稼働しているが、エラーが発生した場合に E メール通知を送信するための E メール別名など、管理上の変更を必要とするため。この場合は、コラボレーション・オブジェクトを変更する必要があります。

コラボレーション・オブジェクトを変更するには、System Manager の「統合コンポーネント・ライブラリー」フォルダーまたは「ユーザー・プロジェクト」フォルダーで、「コラボレーション・オブジェクト」フォルダー内のコラボレーション・オブジェクトをダブルクリックします。

コラボレーション・オブジェクトの表示

コラボレーション・オブジェクトには、グラフィック表示とツリー表示の 2 つの表示の方法があります。次のセクションでは、これらの表示およびそれぞれの利点について説明します。

コラボレーション・オブジェクトのグラフィック表示

コラボレーション・オブジェクトのグラフィック表示は、新規コラボレーション・オブジェクトの作成後にデフォルトで示される表示です。227 ページの図 57 に示されているように、グラフィック表示では、コラボレーション・オブジェクトを示すアイコンが中央に表示され、コラボレーション内の各ポートを示すアイコンがその外側に放射状に示されます。この表示は、ビジネス・プロセスの「フロー」を示すのに便利です。ビュー・ペインで各ポートをドラッグ・アンド・ドロップして、コラボレーション通信の方向を示す場所にそれらを置くことができます。例えば、トリガー・ビジネス・オブジェクトを受信するポートをペインの左側に置き、宛先アプリケーションへビジネス・オブジェクトを送信するポートをペインの右側に置くことができます。

注: システムが自動的に、トリガー・ビジネス・オブジェクトを受信するポートを一方に、宛先アプリケーションへビジネス・オブジェクトを送信するポートを他方に表示することはできません。ポートの名前と機能を決定するのはコラボレーション・テンプレートの開発者であり、開発者が考えている特定のポートが果たす役割をシステムが認識する方法はありません。コラボレーション・テンプレートの資料を読み、テンプレートに基づいて、ツールの使用法に最も適した方法で、コラボレーション・オブジェクトのグラフィック表示内の各ポートの位置を変更してください。

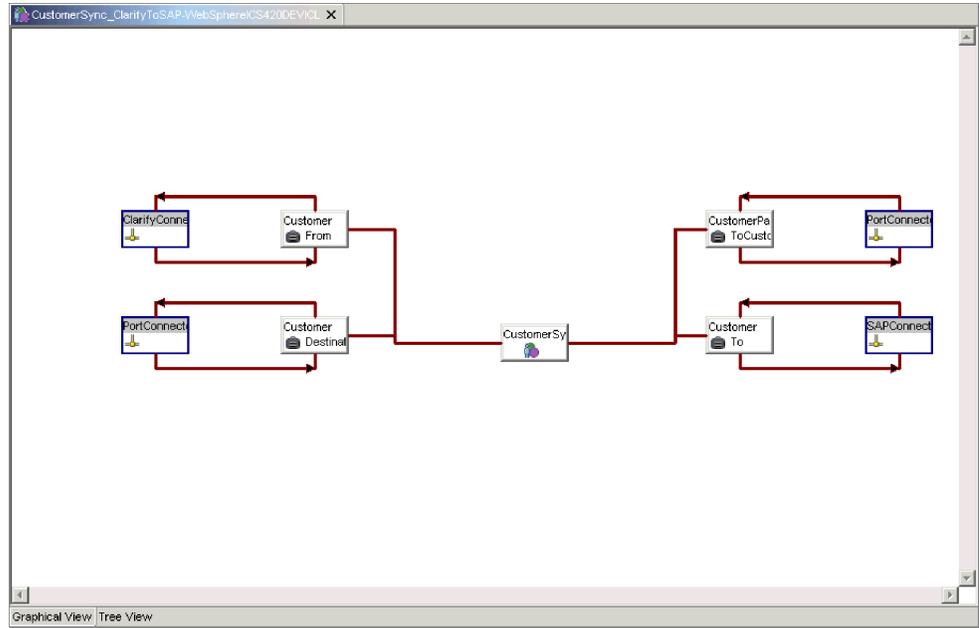


図 57. コラボレーション・オブジェクトのグラフィック表示

コラボレーション・オブジェクトのツリー表示

「ツリー表示」タブをクリックすると、グラフィック表示からコラボレーション・オブジェクトのツリー表示へ切り替えることができます。228 ページの図 58 に示されているように、ツリー表示では、コラボレーション・オブジェクトを示すアイコンが階層ツリーの一番上に表示され、各ポートを示すアイコンがその下に分岐して示されます。この表示ではビジネス・プロセスの「フロー」はあまりわかりませんが、各ポートおよびそれらにバインドされたコンポーネントが整然と表示されるので、再構成する必要がある特定のポートを容易に見つけることができます。多数のポートを持つテンプレートを基にしたコラボレーション・オブジェクトの場合、グラフィック表示では非常に複雑な表示になるため、この表示が特に役立ちます。コラボレーション・オブジェクトのツリー表示でアイコンの位置を変更することはできません。

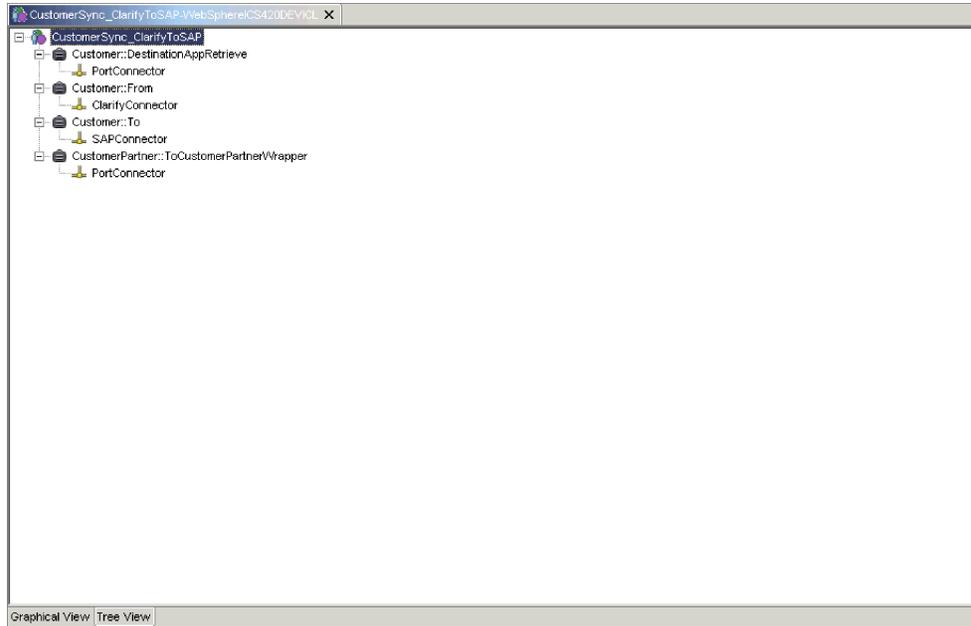


図 58. コラボレーション・オブジェクトのツリー表示

コラボレーション・オブジェクト・ポートのバインディング

コラボレーション・ポートは、コラボレーションがビジネス・オブジェクトを送受信するインターフェースです。ポートによってサポートされるビジネス・オブジェクトのタイプを交換するには、コラボレーション・オブジェクト・ポートをさまざまなコンポーネントにバインドして構成します。

コラボレーション・オブジェクト・ポートのバインディングは、220 ページの『コラボレーション・オブジェクトの作成』の説明に従いウィザードを使用して最初にコラボレーション・オブジェクトを作成する際に、実行できます。また、コラボレーション・オブジェクトの作成後にポートのバインディングを編集することもできます。グラフィック表示でポート・バインディングを編集するには、ポートのアイコンをダブルクリックするか、ポートのアイコンを右マウス・ボタンでクリックしてコンテキスト・メニューから「**ポートをバインド**」を選択します。ツリー表示でポート・バインディングを編集するには、ポートのアイコンを右マウス・ボタンでクリックしてコンテキスト・メニューから「**ポートをバインド**」を選択します。

コネクタや他のコラボレーション・オブジェクトのポートなど、内部コンポーネントへコラボレーション・ポートをバインドするには、「タイプ」ペイン内の「**内部**」というラベルのラジオ・ボタンをオンにしたまま、229 ページの『内部ポート・バインディングの構成』を参照してください。

ウェブ・サーブレットなどの外部プログラマチック・エンティティーにコラボレーション・ポートをバインドするには、「タイプ」ペイン内の「**外部**」というラベルのラジオ・ボタンをオンにして、230 ページの『外部ポート・バインディングの構成』を参照してください。

内部ポート・バインディングの構成

コネクタまたは別のコラボレーション・オブジェクトのポートへコラボレーション・ポートをバインドするには、以下の手順を実行します。

1. ポート構成ダイアログの「タイプ」ペインにある「内部」というラベルのラジオ・ボタンをオンにします。
2. 「バインド先」ペインで適切なコンポーネント・タイプ（「コネクタ」または「コラボレーション」）を選択します。
3. リストから特定のコンポーネントを選択します。
4. 「OK」をクリックします。

図 59 に、内部バインディング・タイプのポート構成ダイアログを示します。

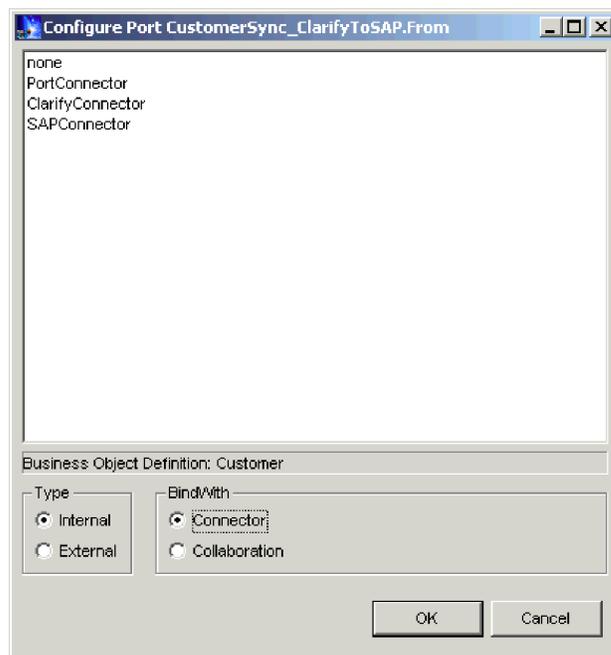


図 59. 内部コラボレーション・オブジェクト・ポートの構成

「ビジネス・オブジェクト定義」フィールドに表示されているビジネス・オブジェクト定義タイプをサポートするコンポーネントのみがダイアログにリストされます。予期したコンポーネントが表示されない場合は、それが指定のタイプではないか（その場合は、「バインド先」ラジオ・ボタンを変更する必要があります）、あるいはコンポーネントがビジネス・オブジェクト定義をサポートしていません。「バインド先」ラジオ・ボタンが適切なコンポーネント・タイプに設定されているにも関わらず、予期したコンポーネントが表示されない場合は、そのコンポーネントの定義を変更してビジネス・オブジェクト定義に対するサポートを追加し、その後、ポート構成ダイアログを起動します。

コラボレーション・オブジェクトを始動するには、すべてのポートをバインドする必要があるため、ポートのバインドが完了するまでコラボレーション・オブジェクトを実行することはできません。

コラボレーション・テンプレートの中には、ビジネス・ロジックのオプション・コースをサポートするポートが定義されているものもあります。例えば、多くのコラボレーション・テンプレートは、操作が成功したことを確認するために、宛先アプリケーションで作成されたばかりのエンティティを検索できるように設計されています。多くの場合、この種の振る舞いはオプションであり、そのコラボレーションのコラボレーション固有プロパティで構成できます。コラボレーションを始動するには、前述の説明に従って、コラボレーション・オブジェクトのポートをすべてコンポーネントにバインドする必要があります。したがって、オプションの振る舞いを利用するつもりがない場合でも、それをサポートするためのポートを構成する必要があります。その場合は、未使用のコンポーネント (PortConnector など) にビジネス・オブジェクト定義のみに対するサポートを追加し、オプションの機能を使用しないようにコラボレーション・プロパティを構成します。

外部ポート・バインディングの構成

コネクタまたは別のコラボレーション・オブジェクトのポートへコラボレーション・ポートをバインドするには、以下の手順を実行します。

1. ポート構成ダイアログの「タイプ」ペインにある「外部」というラベルのラジオ・ボタンをオンにします。
2. ビジネス・オブジェクト要求をポートで受信する場合は「構成」ペインで「入力」ラジオ・ボタンを選択し、ビジネス・オブジェクト応答をポート外へ送信する場合は「出力」ラジオ・ボタンを選択します。
3. 次のいずれかを実行して、目的のコンポーネントをポートと関連付けます。
 - ビジネス・オブジェクト要求をポートで受信するのかビジネス・オブジェクト応答をポート外へ送信するのかに応じて、統合コンポーネント・ライブラリー・フォルダーから「入力マップ (Incoming Maps)」または「出力マップ (Outgoing Maps)」列のあるペインへビジネス・オブジェクト定義をドラッグ・アンド・ドロップします。

「ビジネス・オブジェクト・タイプ」ダイアログが表示されたら、ドラッグ・アンド・ドロップしたビジネス・オブジェクトが、それを変換するマップ内でソース・オブジェクトであるか宛先オブジェクトであるかに応じて、「ソース・ビジネス・オブジェクト」または「宛先ビジネス・オブジェクト」ラジオ・ボタンを選択します。
 - ビジネス・オブジェクト要求をポートで受信するのかビジネス・オブジェクト応答をポート外へ送信するのかに応じて、ポートによってサポートされるビジネス・オブジェクトのタイプを変換するマップ定義を、統合コンポーネント・ライブラリー・フォルダーから「入力マップ (Incoming Maps)」または「出力マップ (Outgoing Maps)」列のあるペインへドラッグ・アンド・ドロップします。
4. 「OK」をクリックします。

231 ページの図 60 に、外部バインディング・タイプのポート構成ダイアログを示します。

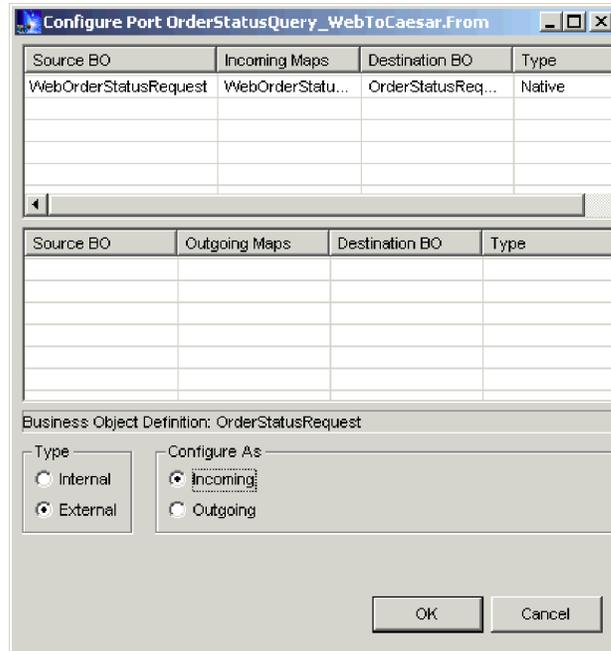


図 60. 外部コラボレーション・オブジェクト・ポートの構成

外部プログラマチック・エンティティーとの統合を実装する方法の詳細については、「アクセス開発ガイド」を参照してください。

コラボレーション・オブジェクトの一般プロパティーの構成

コラボレーションの一般プロパティーは、オブジェクトの基になるテンプレートを開発者がどう設計したかにかかわらず、すべてのコラボレーション・オブジェクトに属するプロパティーです。これは、システム全体でのコラボレーション・オブジェクトの振る舞いに影響します。

コラボレーション・オブジェクトの一般プロパティーを変更するには、以下の手順を実行します。

1. 次のいずれかを実行して、コラボレーション・オブジェクトの「プロパティー」ダイアログにアクセスします。
 - 統合コンポーネント・ライブラリーでコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティー」を選択します。デフォルトでは、ダイアログに「コラボレーションの一般プロパティー」タブが表示されます。
 - グラフィック表示でコラボレーション・オブジェクト・アイコンをダブルクリックします。デフォルトでは、ダイアログに「コラボレーションの一般プロパティー」タブが表示されます。
 - グラフィック表示またはツリー表示でコラボレーション・オブジェクト・アイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティー」を選択します。デフォルトでは、ダイアログに「コラボレーションの一般プロパティー」タブが表示されます。

2. プロパティを目的の値に設定します。各プロパティおよびそれらで使用できる値については、表 15 およびそれ以降のセクションを参照してください。
3. 「OK」をクリックします。

表 15. コラボレーション・オブジェクトの一般プロパティ

プロパティ名	指定可能な値
作成のテンプレート	コラボレーション・オブジェクトの基になるテンプレートの読み取り専用名
有効トランザクション・レベル	なし、最小限の努力、最大限の努力、または緊急
最小トランザクション・レベル	コラボレーション・テンプレートの設計に応じて、なし、最小限の努力、最大限の努力、または緊急
システム・トレース・レベル	0 - トレースなし 1 - コラボレーション操作 2 - およびコラボレーション・イベント 3 - および状態トランザクション 4 - および着信/発信メッセージ 5 - およびメッセージの詳細な内容
コラボレーション・トレース・レベル	0 から 5
E メール通知アドレス	任意の有効な E メール別名
クリティカル・エラーの場合は一時停止	オンまたはオフ
並行イベントの最大数	0 から 9999
転送中状態でサービスの呼び出しを持続	オンまたはオフ
リカバリー・モード	「常時」または「据え置き」
暗黙的なデータベース・トランザクション	オンまたはオフ
最大イベント容量	1 から 2147483647 までの整数
ブロッキング・タイプ	オンまたはオフ
イベント分離	オンまたはオフ

作成のテンプレート

この読み取り専用テキスト・フィールドには、コラボレーション・オブジェクトの基になるコラボレーション・テンプレートの名前が表示されます。

有効トランザクション・レベル

有効トランザクション・レベルは、すべてのコラボレーション・オブジェクトの最大トランザクション・レベルの最高値と、オブジェクトにバインドされたすべてのコネクタの最大トランザクション・レベルの最低値との間の範囲です。

特定のトランザクション・レベルがサポートされていないコネクタにバインドする必要がある場合は、コラボレーションの有効トランザクション・レベルを下げるすることができます。

有効トランザクション・レベルを変更するには、「有効トランザクション・レベル」ドロップダウン・リストから希望の値を選択し、「OK」をクリックします。

注: トランザクション・レベルに互換性のないオブジェクトはバインドできません。これらのオブジェクトをコラボレーションにバインドする必要がある場合は、必要に応じてサポートされるトランザクション・レベルを調整してください。

最小トランザクション・レベル

最小トランザクション・レベルは、コラボレーション・テンプレートおよびそれを基にするすべてのコラボレーション・オブジェクトのトランザクション・レベルの最低値を指定します。

例えば、コラボレーション・テンプレートの開発者がテンプレートに対して「最大限の努力」の最小トランザクション・レベルを指定した場合は、そのテンプレートを基にしたすべてのオブジェクトが「最大限の努力」または「緊急」で稼働する必要があります。232 ページの『有効トランザクション・レベル』に関するセクションで説明されているように、コラボレーション・オブジェクトにバインドされたすべてのコンポーネントが有効トランザクション・レベルをサポートしている必要があります。最小トランザクション・レベルを使用すると、コラボレーション・テンプレートの開発者は、テンプレートを基にしたコラボレーション・オブジェクトを実行するインターフェース全体の最低トランザクション・レベルを指定できます。

コラボレーション・オブジェクトを構成する際、「最小トランザクション・レベル」フィールドは読み取り専用です。これは、コラボレーション・テンプレートでしか変更できません。

コラボレーション・テンプレートの変更の詳細については、「コラボレーション開発ガイド」を参照してください。

システム・トレース・レベル

コラボレーションの実行に関する情報がサーバー出力で報告されるようにコラボレーション・オブジェクトを構成することができます。そのためには、「システム・トレース・レベル」ドロップダウン・メニューで希望の値を選択します。表 16 に、さまざまなレベルおよびそこで報告される情報のタイプを示します。

表 16. システム・トレース・レベル

システム・トレース・レベル	報告される情報
0 - トレースなし	このレベルでは情報はトレースされません。
1 - コラボレーション操作	コネクタからのビジネス・オブジェクトの受信およびシナリオの開始をトレースします。
2 - およびコラボレーション・イベント	レベル 1 のメッセージと、順方向実行とロールバックの両方を含めた各シナリオの開始および完了を印刷します。
3 - および状態トランザクション	レベル 1 および 2 のメッセージと、各シナリオの決定ブロックまたはアクション・ノードの実行を印刷します。
4 - および着信/発信メッセージ	レベル 1 から 3 までのメッセージと、各シナリオによる各ビジネス・オブジェクトの送信と受信を印刷します。

表 16. システム・トレース・レベル (続き)

システム・トレース・レベル	報告される情報
5 - およびメッセージの詳細な内容	レベル 1 から 4 までのメッセージと、処理中のビジネス・オブジェクトの構造を各属性の値とともに印刷します。

コラボレーション・トレース・レベル

コラボレーションの開発者は、テンプレート固有のトレースでコラボレーション・テンプレートをコード化します。システム・トレース (233 ページの『システム・トレース・レベル』を参照) はコラボレーション・ランタイム全般に関するトレース情報を提供しますが、コラボレーション・トレースは、特定のコラボレーションに関する情報を示します。例えば、次のようになります。

- コラボレーションが処理しているビジネス・オブジェクト内のデータに基づいて下される業務決定を報告するように、開発者がコラボレーションを設計します。
- 開発者は、システム・タイプのトレースを提供できますが、コラボレーション実行におけるある特定の時点でのみ、それを行います。例えば、システム・トレース・レベル 5 を指定すると、サーバー出力にビジネス・オブジェクト・データがすべてダンプされます。これは、問題のデバッグに非常に役立ちますが、単一のコラボレーション実行中に何度も発生すると、目を通すのがたいへんです。コラボレーションの開発者は、あるレベルのコラボレーション・トレースで、コラボレーション・フローで特に問題のある一時点でのみ全ビジネス・オブジェクトのダンプを設定することができます。そうすれば、ユーザーがサーバー出力で目にするビジネス・オブジェクトのダンプは 1 つだけになります。

コラボレーションのトレース・レベルを設定するには、「**コラボレーション・トレース・レベル**」ドロップダウン・メニューで 0 から 5 までの希望の値を選択します。

特定のコラボレーションの異なるトレース・レベルでどのような情報が報告されるかを確認するには、そのコラボレーション・テンプレートの資料を参照してください。

コラボレーション・テンプレートを変更し、コラボレーション・トレースを実装する方法の詳細については、「**コラボレーション開発ガイド**」を参照してください。

E メール通知アドレス

特定のコラボレーション・オブジェクトに関連するエラーが発生した場合に E メール通知が送信されるように、コラボレーション・オブジェクトを構成することができます。インターフェース固有の管理が望ましい場合は、この機能を使用すると、管理が容易になります。例えば、あるサイトに、WebSphere InterChange Server 自体に関連するエラーに責任を負う管理者、顧客アカウント同期インターフェースに特に責任を負う管理者、オーダー処理インターフェースに特に責任を負う管理者がそれぞれいるような場合です。

「**E メール通知アドレス**」フィールドに、エラー通知の送信先となる SMTP 準拠の E メール・アドレスを入力してください。コンマ区切りで複数のアドレスを入力できます。システムで E メール通知を使用できるようにする方法の詳細については、

133 ページの『System Manager を使用した E メール通知プロパティの構成』および「システム管理ガイド」を参照してください。

「E メール通知アドレス」フィールドでの値の設定以外にも、E メール通知を送信するタイミングについてさまざまな設定が、多数のコラボレーション・テンプレートに用意されています。どのくらいの設計が可能かはコラボレーション・テンプレートの設計で決まるので、各コラボレーション・テンプレートの E メール通知機能については、それぞれの資料を参照してください。

クリティカル・エラーの場合は一時停止

コラボレーションがコネクタにビジネス・オブジェクト要求を送信したときに、コネクタによるビジネス・オブジェクト要求の処理を妨げるエラーが発生する場合があります。例えば、次のようなエラーがあります。

- コネクタがアプリケーションにログインできない。
- コネクタとアプリケーションとの通信がタイムアウトになる。
- コネクタ・エージェントが不明状態に変更された。

これらのタイプのエラーが発生すると、コラボレーションがコネクタへ要求を送信しても、その問題のためにフローが失敗します。問題を解決するまで、送信されたすべての要求について同じことが起こり、その結果、トランザクションのボリュームが大きくエラーの継続時間が長い場合には、インターフェースで多数のフローが失敗することになります。

クリティカル・エラーのために要求が失敗した後はコネクタへの要求の送信を停止するように、コラボレーション・オブジェクトを構成することができます。そのためには、「クリティカル・エラーの場合は一時停止」チェック・ボックスをオンにします。

クリティカル・エラー、クリティカル・エラーに対するコラボレーション・オブジェクトの対応、およびこの機構の機能の詳細については、「システム管理ガイド」を参照してください。

並行イベントの最大数

複数のイベント・トリガー・フローを並行して処理するように、コラボレーション・オブジェクトを構成することができます。それにより、インターフェースのスループットが向上します。これを行うには、「並行イベントの最大数」ドロップダウン・メニューで 0 から 9999 までの数を指定して、コラボレーション・オブジェクトで並行処理するイベント数を設定します。

このコラボレーション機能の利点を完全に活用するには、同様の振る舞いをするインターフェースに関わる他のコンポーネントも構成する必要があります。詳細については、361 ページの『イベントにより起動されるフローの並行処理の実装』を参照してください。

転送中状態でサービスの呼び出しを持続

コラボレーションが、バインドされている宛先コネクタにビジネス・オブジェクト要求を送信してから、それぞれのアプリケーションで要求が正常に処理されたかどうかの応答を受け取るまでの間にエラーが発生する場合があります。コネクタが要求を処理できなかった場合は、エラーを解決してから要求を再び処理する必要

があります。しかし、コネクタが要求を正常に処理し、InterChange Server へ処理成功の通知を送信するプロセス中にエラーが発生した場合、InterChange Server は通知を受け取りません。要求の状態に関する最終レコードには、まだ要求を処理する必要があると示されます。この不正確な状態レコードによって、要求は二度処理され、データが重複することになります。

非トランザクション・コラボレーションの構成時にこの問題への方策を講じるには、「転送中状態でサービスの呼び出しを継続」チェック・ボックスをオンにします。これにより、InterChange Server は、エラーが発生した場合に、宛先アプリケーションへ転送中のビジネス・オブジェクト要求を保持します。システムがリカバリーしたときに要求は送信されないため、宛先アプリケーションで要求が二度処理されるリスクが少なくなります。次に、Flow Manager を使用して要求および宛先アプリケーションを調べ、エラーが発生する前に要求が正常に処理されたかどうかを判別することができます。正常に処理された要求は破棄し、正常に処理されなかった要求は再実行依頼する必要があります。

この種の転送関連の障害を処理するために、プログラマチックな方策を講じることもできます。詳細については、「コラボレーション開発ガイド」で、サービス呼び出しトランスポート例外の処理に関する説明を参照してください。

リカバリー・モード

前のリリースでは、致命的なエラーが発生すると、システムを再始動したときに、状態の発生時に処理中だったすべてのフローがリカバリーされました。すべてのフローが永続的な保管から読み取られ、処理が再実行依頼されました。そのようなフローが多数ある場合は、メモリーイベントを取り出すことによってシステム・メモリーがほぼ完全に消費され、メモリー不足に関連する別のシステム・エラーが発生することもありました。また、InterChange Server がリカバリー処理を完了するまで、各ツールを使用してシステムを効率的に管理することができませんでした。

リリース 4.0.0 では、「リカバリー・モード」ドロップダウン・メニューを値「据え置き」に設定することにより、コラボレーション・オブジェクトのリカバリーを据え置くことができるようになりました。このようにして、リカバリーを据え置くようコラボレーション・オブジェクトを構成すると、致命的なシステム・エラーの発生時に進行中だったフローは失敗したフローとして扱われ、システムの再始動後すぐにはリカバリーされなくなります。管理者は、システムの再始動後に Flow Manager を使用してフローを再実行依頼することができます。また、任意の管理アクションを、リカバリー措置の前に実行できます。

要確認: インターフェースの中には、イベントを受信した順序で処理することが重要なものもあります。イベントの順序がインターフェースの整合性にとって重要でない場合は、失敗したフローおよび新規フローが処理される順序にかかわらず、据え置かれたリカバリーを実装することができます。しかし、イベントの順序が重要な場合に、据え置かれたリカバリーを実装する必要があるときは、イベントの順序が確実に維持されるように、管理手順を文書化し、それに従う必要があります。例えば、管理者は、システムの再始動時に、失敗したフローが解決される前にコラボレーションが新規フローを受信および処理することがないようにする必要があります。そのためには、当該のコラボレーションへ新規イベントを送信するソース・コネクタ・エージェントの始動とポーリングを管理します。

リカバリーの据え置きの一貫性は、リリース 4.2.0 で導入された、最適化されたりリカバリー方法によって減少しました。サーバーは、進行中のトランザクションのビジネス・オブジェクト・データ全体をメモリーへ読み取るのではなく、永続的な保管内でビジネス・オブジェクトを見つけるために必要な情報だけを読み取ります。リカバリーの据え置きも使用できますが、この新しい最適化されたりリカバリー方法があれば、その必要はないでしょう。

暗黙的なデータベース・トランザクション

構成中のコラボレーション・オブジェクトの基になるコラボレーション・テンプレートがトランザクション・データベース・ロジックを実装する場合は、暗黙と明示のいずれかのトランザクション・ブラケットに対してコラボレーション・オブジェクトを構成する必要があります。開発者が記述したコードによってトランザクション・セマンティクスが明示的に処理されるようにコラボレーションが開発された場合は、「暗黙的なデータベース・トランザクション」チェック・ボックスをオフにしておく必要があります。ただし、コラボレーションがデータベース・トランザクションの明示的な管理で開発されていない場合は、「暗黙的なデータベース・トランザクション」チェック・ボックスをオンにする必要があります。

暗黙および明示的なトランザクション・ブラケットの詳細については、「コラボレーション開発ガイド」を参照してください。

最大イベント容量

InterChange Server が受信したビジネス・オブジェクトは、処理のためメモリー内のキューに入れられます。リリース 4.2.0 以前は、インターフェースでトランザクションのボリュームが大きく、処理率が低い場合に、多数のビジネス・オブジェクトがキューに入れられ、それによって InterChange Server で致命的なメモリー不足が発生することがありました。

コラボレーション・オブジェクトの最大イベント容量を構成することによって、メモリー不足のリスクを減らすことができます。これを行うには、「最大イベント容量」フィールドで、コラボレーション・オブジェクトについてキューに入れるイベントの最大数を指定します。システムは、このプロパティーで構成された数より多いイベントをメモリー内のキューに入れることはしません。コラボレーション・オブジェクトの 238 ページの『ブロック・タイプ』プロパティーをどう構成するかによって、システムは、コラボレーションの新規ビジネス・オブジェクトの受信に対して異なる対応をします。

このプロパティーの有効な値の範囲は、1 から 2147483647 です。

「ブロック・タイプ」プロパティーの詳細については、238 ページの『ブロック・タイプ』を参照してください。

コネクタ定義のフロー制御プロパティーの詳細および Connector Configurator を使用してコネクタ定義を変更する方法については、159 ページの『第 7 章 コネクタの構成』を参照してください。

フロー制御に関するシステム・プロパティーの詳細については、「システム管理ガイド」を参照してください。

ブロッキング・タイプ

『ブロッキング・タイプ』プロパティは、237 ページの『最大イベント容量』プロパティとともに使用して、コラボレーションへのビジネス・オブジェクトのフローを調整します。

コラボレーション・オブジェクトの構成時に「**ブロッキング・タイプ**」チェック・ボックスをオンにした場合、コラボレーションのメモリー内のキューに入れられたイベント数が「**最大イベント容量**」プロパティで指定された数と等しくなると、コラボレーションへのビジネス・オブジェクトの送信を担当するコネクタ・コントローラーが、送信を停止します。コラボレーションのキュー内のイベント数が、「**最大イベント容量**」プロパティで指定された数と等しくなくなると、コネクタ・コントローラーは、コラボレーションへのイベントの送信を再開します。

コラボレーション・オブジェクトの構成時に「**ブロッキング・タイプ**」チェック・ボックスをオフにした場合、コラボレーションのメモリー内のキューに入れられたイベント数が「**最大イベント容量**」プロパティで指定された数と等しくなると、コネクタ・コントローラーによってコラボレーションへ送信された新規イベントは、データベース内に永続的に保管されます。その後、現在コラボレーションのキュー内にあるイベントが処理され、イベントがデータベースからメモリーへ読み取られます。

「**最大イベント容量**」プロパティの詳細については、237 ページの『最大イベント容量』を参照してください。

フロー制御に関するシステム・プロパティの詳細については、「システム管理ガイド」を参照してください。

イベント分離

イベント分離により、2 つのコラボレーションが同じデータを並行して処理しないことが保証されます。場合によっては、複数のコラボレーションが同じタイプのビジネス・オブジェクトを処理することがあります。あるイベントが到着すると、特定のコラボレーションが起動されます。このコラボレーションが実行を開始すると、その実行中は、このコラボレーションが、InterChange Server 内でそのビジネス・オブジェクトのインスタンスに対する唯一のアクセス権を持ちます。同じデータに関連する別のイベントが到着した場合、InterChange Server は、実行中のコラボレーションが最初のイベントの処理を完了するまで、新たに到着したイベントをキューに入れます。

注: イベント分離の詳細については、「コラボレーション開発ガイド」を参照してください。

デフォルトでは、「**イベント分離**」チェック・ボックスにはチェック・マークが付いています。設定は実行時に変更できますが、更新されたプロパティは、処理されていない新規イベント (例えば、まだ WebSphere MQ または WIPQ にあるイベント) に対してのみ有効です。すでに処理中のイベントに対しては、新しい値は適用されません。

イベント分離には複数のコラボレーション・オブジェクトが関与しているので、1 つのイベントに対してイベント分離をオフにした場合、その他のイベントに関してイベント分離が正しく機能しなくなります。イベント分離を有効にするには、同じ

ソース、宛先コネクタ、およびビジネス・オブジェクト参照を持つすべてのコラボレーション・オブジェクトに対して、イベント分離を使用可能にする必要があります。

コラボレーション固有のプロパティの構成

コラボレーション開発者は、実行時に実行されるビジネス・プロセス・ロジックに作用する独自のプロパティをコラボレーションが個別に持てるように、コラボレーション・テンプレートを設計できます。これにより、非常に柔軟なアーキテクチャが実現され、実装固有の要件に合わせたコラボレーション・テンプレートのカスタマイズも可能になります。

コラボレーション・オブジェクトのテンプレート固有のプロパティを変更するには、以下の手順を実行します。

1. 次のいずれかを実行して、コラボレーション・オブジェクトの「プロパティ」ダイアログにアクセスします。
 - 統合コンポーネント・ライブラリーでコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティ」を選択して、「プロパティ」タブをクリックします。
 - グラフィック表示でコラボレーション・オブジェクト・アイコンをダブルクリックし、「プロパティ」タブをクリックします。
 - グラフィック表示またはツリー表示でコラボレーション・オブジェクト・アイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「プロパティ」を選択して、「プロパティ」タブをクリックします。
2. 「プロパティ名」フィールドにリストされたプロパティの「値」フィールドに、希望の値を入力します。
3. 「OK」をクリックします。

コラボレーション・テンプレートに固有の各プロパティおよびそれらに有効な値については、コラボレーション・テンプレートの資料を参照してください。

テンプレートへのプロパティの追加またはコラボレーション・テンプレート内の既存のプロパティの使用法の変更については、「コラボレーション開発ガイド」を参照してください。

第 10 章 Relationship Manager の使用

Relationship Manager を使用すると、参加者とそのデータを含み、関係の実行時データに対する操作の表示、実行を行うことができます。関係に関する背景情報については、「マップ開発ガイド」を参照してください。

関係定義は、Relationship Designer を使用して作成します。実行時に、関係のインスタンスに対して、さまざまなアプリケーションからの情報を関連付けるデータが読み込まれます。この関係インスタンス・データは、関係を使用するマップが実行されるときに作成されます。このデータは、関係定義に指定されている関係表に格納されます。Relationship Manager には、データベース・ベンダーに関係なく、関係表と連携動作するグラフィカル・インターフェースが用意されています。

関係インスタンスごとに、Relationship Manager は、関係インスタンスの参加者定義と、キー属性と非キー属性の集合である参加者インスタンスの階層リストを表示します。関係ツリーには、関係インスタンス内の各参加者に関する詳細情報、エンティティのタイプ、その値、および最終変更日付などが表示されます。関係インスタンス ID は、関係インスタンスが関係表に保管されるときに自動的に生成されます。Relationship Manager は、関係ツリーの最上位レベルにこのインスタンス ID を表示します。

242 ページの図 61 に、Relationship Manager における同一関係に対する関係ツリーの例を示します。

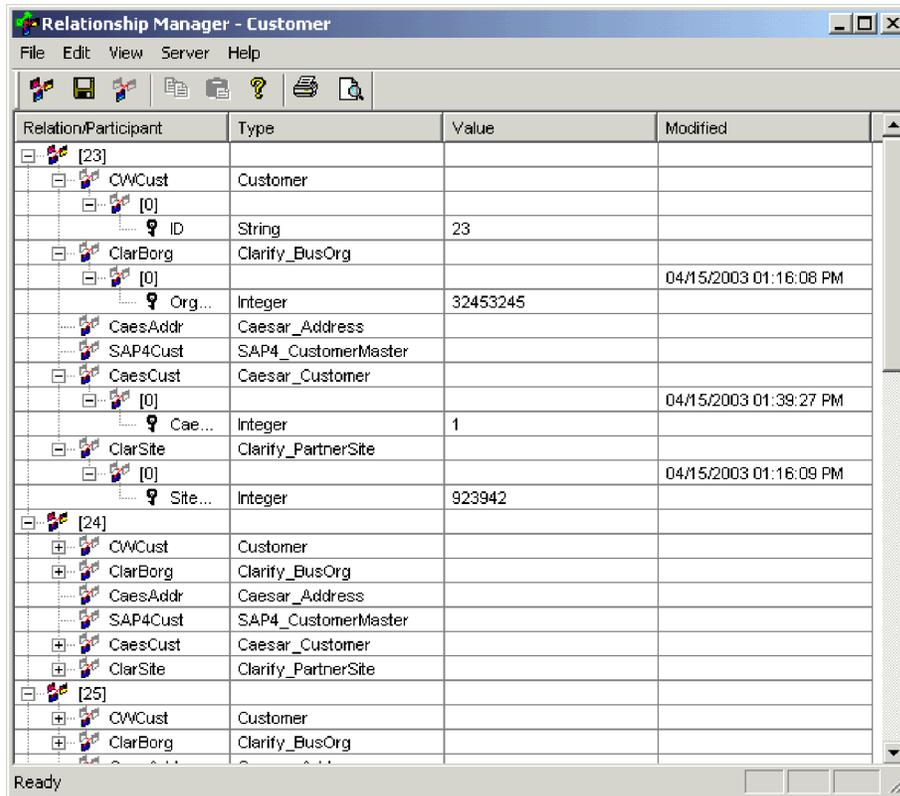


図 61. Relationship Manager

Relationship Manager を使用すると、すべてのレベル (関係インスタンス・レベル、参加者インスタンス・レベル、および属性レベル) でエンティティを操作することができます。例えば、以下のことを行うために、Relationship Manager を使用できます。

- 関係インスタンスを作成、削除すること
- 参加者の追加・削除など、関係インスタンスの内容を変更すること
- 参加者のデータを追加、保管すること、ファイルから参加者のデータを読み込んだり、ファイルに保管したりすること、参加者のデータを別の関係からコピーし、関係インスタンスに貼り付けること、それによって、(参加者のタイプが同一であることを条件に) 新規参加者を作成すること
- 参加者をアクティブ、非アクティブにすること
- インスタンス ID、ビジネス・オブジェクト属性値、またはデータに基づき、参加者を検索すること
- ある時間間隔内で参加者のアクティビティをフィルターに掛けること
- データに関する問題が発生した場合にその状況を修復すること。例えば、ソース・アプリケーションから破壊されたデータまたは矛盾するデータが、汎用的な宛先アプリケーション関係表に送信された場合、Relationship Manager を使用して、データが信頼できると認識している時点まで、データをロールバック (すなわちクリーンアップ) することができます。

Relationship Manager の始動

Relationship Manager を起動するには、以下のいずれかを実行します。

- 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「管理」>「Relationship Manager」を選択します。
- Relationship Designer で、関係定義を選択し、メニュー・バーから「ツール」>「Relationship Manager」を選択します。

Relationship Manager が始動します。この時点では、Relationship Manager は、サーバーから切断されているため、『InterChange Server への接続』の説明に従って、InterChange Server サーバーに接続してから、次の操作に進む必要があります。

サーバーへの接続および接続の切断

関係インスタンスとデータを操作するには、InterChange Server に Relationship Manager を接続する必要があります。以下のセクションの説明に従って、Relationship Manager をサーバーに接続し、また、接続の切断を行います。

- 『InterChange Server への接続』
- 244 ページの『InterChange Server からの切断』

InterChange Server への接続

Relationship Manager を InterChange Server に接続するには、以下の手順を実行します。

1. Relationship Manager のメニュー・バーから、「サーバー」>「サーバーに接続」を選択します。
2. 以下のいずれかの操作を実行し、接続先の InterChange Server インスタンスを名前を「サーバー名」フィールドに入力します。
 - InterChange Server インスタンスの名前を「サーバー名」フィールドに入力します。

要確認: InterChange Server インスタンスの名前では大文字小文字が区別されるため、名前は正確に指定してください。

 - ドロップダウン・メニューからキャッシュに入っているサーバー名を選択します。
 - 以下の手順を実行し、ネットワークの InterChange Server インスタンスを参照します。
 - a. 参照ボタンをクリックします。
 - b. 「サーバー」ダイアログで、リストから目的の InterChange Server インスタンスを選択します。
 - c. 「OK」をクリックします。
3. 「ユーザー名」フィールドに InterChange Server インスタンスと対話するユーザーの名前を入力します。
4. ステップ 3 で指定したユーザー名に対するパスワードを「パスワード」フィールドに入力します。

5. System Manager において、InterChange Server インスタンスに接続する必要のあるたびに、ユーザー名とパスワードを指定しないで済むようにするには、「ユーザー名とパスワードを保管」チェック・ボックスを有効にします。
6. この時点で関係を開く場合は、「関係」フィールドに関係定義の名前を入力します。

この時点で関係を開かなくても、サーバーに接続した後に開くことができます。詳細については、『関係を開く』を参照してください。

7. 「接続」をクリックします。

Relationship Manager で、InterChange Server に接続し、ステップ 6 の説明に従って、開く対象の関係を指定した場合は、Relationship Manager が、245 ページの『関係インスタンスの検索』に説明されている、「関係を検索」ウィンドウを表示します。

図 62 に、「InterChange Server に接続」ダイアログを示します。



図 62. InterChange Server に接続

InterChange Server からの切断

InterChange Server と Relationship Manager の接続を切断するには、Relationship Manager のメニュー・バーから、「サーバー」>「切断」を選択します。

関係の処理

Relationship Manager の起動、InterChange Server への接続を終了すると、Relationship Manager を使用して、以下のセクションの説明に従って、関係データを処理できるようになります。

関係を開く

Relationship Manager がサーバーに接続された後、以下のいずれかの操作を実行して、Relationship Manager の関係定義を開きます。

1. Relationship Manager のメニュー・バーから、「ファイル」>「開く」を選択します。
2. 「関係を開く」ウィンドウで、開く対象の関係の名前を選択します。

図 63 に、「関係を開く」ウィンドウを示します。

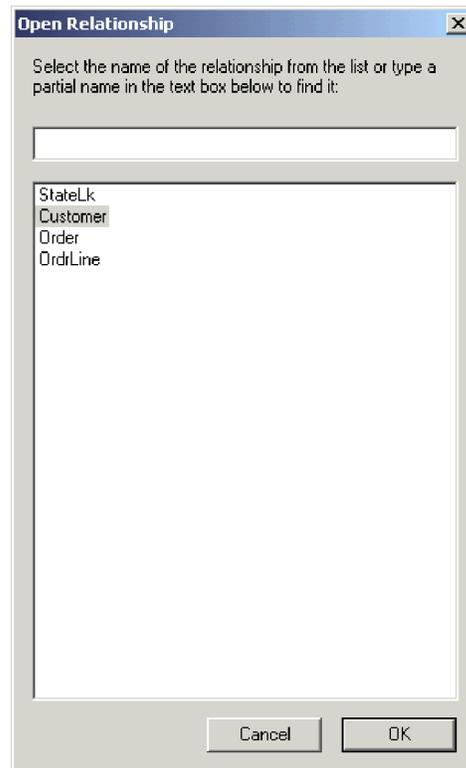


図 63. 関係を開く

3. 「OK」をクリックします。

関係を開くと、Relationship Manager が、『関係インスタンスの検索』に説明する、「関係を検索」ウィンドウを表示します。

関係インスタンスの検索

Relationship Manager のメニュー・バーから、「ファイル」>「関係を検索」を選択して、関係インスタンスを検索するか、ある関係に対して存在するインスタンスの数を取得します。この操作で、「関係を検索」ウィンドウが表示されます。

246 ページの図 64 に、「関係を検索」ウィンドウを示します。

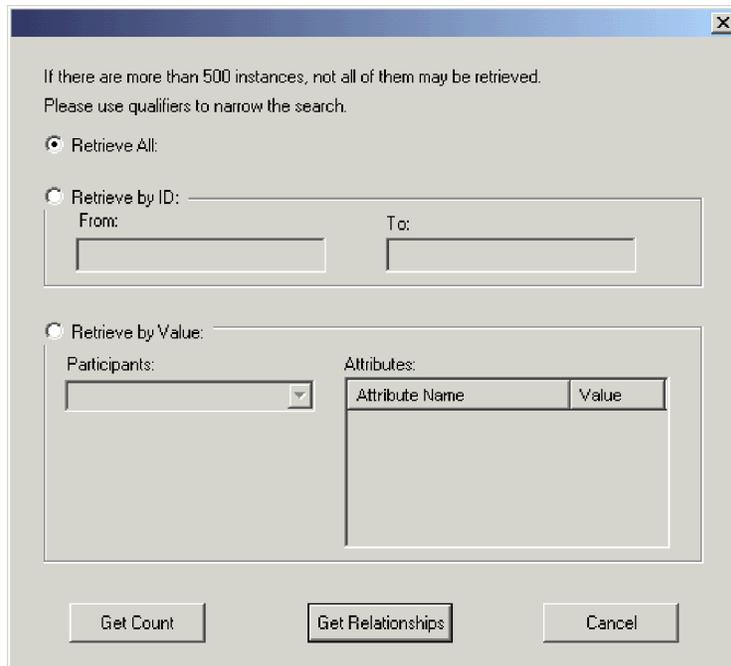


図 64. 関係インスタンスの検索

「関係を検索」ウィンドウは、関係を開くか、InterChange Server への接続後に開く対象の関係を指定した場合にも表示されます。

「関係を検索」ウィンドウでは、以下の操作を実行できます。

- 『すべてのインスタンスの検索』の説明に従って、関係の最初の 500 個のインスタンスを検索する。
- 『関係 ID による検索』の説明に従って、関係インスタンス ID に基づいて、ある範囲の関係インスタンスを検索する。
- 247 ページの『参加者データによる検索』の説明に従って、ユーザーによって指定された特定の値の参加者を含む関係インスタンスを検索する。
- 248 ページの『関係インスタンスの個数の取得』の説明に従って、関係のインスタンスの個数を取得する。

関係定義内の参加者の個数および各関係インスタンス内の参加者のインスタンスの個数に応じて、上記の検索クエリーは、時間が相当かかる場合があります。

すべてのインスタンスの検索

以下の操作を実行して、関係の最初の 500 個のインスタンスを検索します。

1. 「関係を検索」ウィンドウで、「すべてを検索」をクリックします。
2. 「関係を取得」をクリックします。

Relationship Manager が、この関係の最初の 500 個のインスタンスを表示します。

関係 ID による検索

以下の操作を実行して、ある範囲のインスタンスを最高 500 個まで検索します。

1. 「関係を検索」ウィンドウで、「ID で検索」をクリックします。

2. 「開始値:」フィールドに検索する範囲内の最初のインスタンスの ID を入力します。
3. 「終了値:」フィールドに検索する範囲内の最後のインスタンスの ID を入力します。
4. 「関係を取得」をクリックします。

Relationship Manager により、指定した ID の範囲内のインスタンスが、最高 500 個まで表示されます。

参加者データによる検索

以下の操作を実行して、選択した参加者のキー属性または非キー属性の値に基づき、関係インスタンスを検索します。

1. 「関係を検索」ウィンドウで、「値で検索」をクリックします。
2. その値に基づいて検索する参加者を、「参加者」ドロップダウン・メニューから選択します。

同一関係の場合、ドロップダウン・メニューには、参加者の名前、続いて参加者が関係付けられているビジネス・オブジェクト定義が表示されます。

ルックアップ関係の場合、ドロップダウン・メニューには、参加者の名前、続いてワード「データ」が表示されます。

3. 「属性」ペインの「値」列に、表 17 に表示されている値の型のいずれかを入力します。

表 17. 参加者データによる関係インスタンスの検索用にサポートされている値

値	説明
参加者データ	<p>選択されている参加者のデータ</p> <p>例えば、関係が同一関係である場合は、参加者インスタンスがその中に存在する関係インスタンスを発見するため、既知の参加者インスタンスの ID を指定することが考えられます。</p> <p>関係が、ルックアップ関係の場合は、参加者インスタンスの非キー・データ値を指定することが考えられます。</p>
%	<p>任意の文字ストリングです。このオプションでは、大文字小文字が区別され、数字も文字セット内に含まれます。</p> <p>例えば、アメリカの短縮名を格納する参加者に対して、%A を指定すると、CA、GA、IA、LA、MA、PA、VA、WA という値が戻ってきます。</p>
_	<p>任意の単一文字です。</p> <p>例として、_00 の場合は、100、200、a00、b00 などが取得されます。</p>

4. 「関係を取得」をクリックします。

指定した値に一致する最初の 500 個の関係インスタンスが、Relationship Manager によって表示されます。

関係インスタンスの個数の取得

検索条件を満たす関係インスタンスの個数を取得するには、246 ページの『すべてのインスタンスの検索』、246 ページの『関係 ID による検索』、または 247 ページの『参加者データによる検索』の説明に従って、検索条件用のオプションを選択してから、「関係を取得」ではなく、「**カウントを取得**」をクリックします。

関係インスタンスの作成

以下の操作を実行して、関係の新規インスタンスを作成します。

1. 以下の操作のいずれかを実行して、新規の関係インスタンスを作成します。
 - メニュー・バーから「ファイル」>「新規」を選択します。
 - キーボード・ショートカット **Ctrl+N** を使用します。
 - ツールバーの「**新規関係**」をクリックします。

Relationship Manager が、新規の関係インスタンスを表示します。

階層関係ツリーの先頭で強調表示され、関係アイコンの入力行上には、関係インスタンス ID のプレースホルダー があり、3 つの疑問符 (???) を表示します。関係インスタンスまたはその任意の参加者を保管すると、InterChange Server は自動的に、新規の関係インスタンス ID を生成し、Relationship Manager が、3 つの疑問符をその関係インスタンス ID で置き換えます。

2. ??? プレースホルダー・アイコンの横にあるプラス (+) 記号をクリックして、新規の関係インスタンスを展開します。

関係ツリーには、関係インスタンスの下に参加者定義、参加者インスタンス、および参加者キーと非キー属性が降順に表示されます。

3. 以下の操作を実行して、関係インスタンス内に新規の参加者インスタンスを作成します。
 - a. 関係ツリーで、そのインスタンスを作成する参加者定義を選択します。
 - b. 参加者のインスタンスを追加するには、以下のいずれかを実行します。
 - リスト中の参加者定義を右マウス・ボタンでクリックし、コンテキスト・メニューから、「**参加者を追加**」を選択します。
 - 標準ツールバーの「**参加者を追加**」をクリックします。
 - c. その横にあるプラス (+) 記号をクリックして、新規の参加者インスタンスを展開します。
 - d. 新規の参加者インスタンスを選択します。
 - e. 参加者インスタンスの「**値**」列をクリックしたら、セル内に目的の値を入力します。

注: 属性の「**値**」フィールドに、3 つの疑問符 (???) が表示されている場合は、その参加者は、InterChange Server に管理されています。ユーザーが関係インスタンスを保管するときに、InterChange Server が自動的に生成するため、そのような参加者に対しては値を入力できません。値は、関係インスタンス ID と同じ値にします。

この時点で、表 18 におけるいずれのタスクでも実行できます。

表 18. 参加者データに対するタスク

タスク	操作
参加者インスタンスを保管	<p>新規の参加者インスタンスを保管するには、その参加者インスタンスを右マウス・ボタンで右クリックし、コンテキスト・メニューから「参加者を保管」を選択します。Relationship Manager が、その参加者のデータを該当する関係表に保管します。参加者インスタンスの「変更」列には、この場合は、作成日になりますが、参加者保管日付が表示されます。</p> <p>注: 参加者データの保管が完了すると、変更できなくなります。参加者データを変更するには、参加者を削除してから、再度作成する必要があります。</p>
さらに参加者インスタンスを追加	<p>前のリストのステップ 3 (248 ページ) を繰り返します。</p> <p>注: 同一関係を処理している場合は、参加者定義に対して、1 つの参加者インスタンスしか作成できません。</p>
参加者を削除	<p>必要な場合、保管した参加者インスタンスは、参加者インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから、「参加者を削除」を選択して削除することができます。Relationship Manager が、関係表から参加者インスタンスを除去します。データベースから参加者インスタンスを除去しない場合は、「参加者を非アクティブ化」オプションを使用します (250 ページの『参加者の非アクティブ化とアクティブ化』を参照)。非アクティブ化された参加者は、参加者の値とそのインスタンス ID を持ち続けます。</p>
関係インスタンスを保管	<p>以下のいずれかのタスクを実行して、関係インスタンスを保管します。</p> <ul style="list-style-type: none"> メニュー・バーから、「ファイル」>「保管」を選択します (メニュー・バーは関係インスタンスを選択するとアクティブになります)。 関係インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから、「関係を保管」を選択します。 <p>InterChange Server が関係インスタンス ID を生成し、Relationship Manager が ??? プレースホルダーをその新規の ID で置き換えます。Relationship Manager は、すべての保管された参加者インスタンスの変更日付を今日の日付に更新します。</p> <p>注: 関係インスタンスを保管する前に、少なくとも 1 つの参加者インスタンスとすべてのキー属性データを作成しておく必要があります。</p>
すべての関係インスタンスを保管	<p>メニュー・バーから、「ファイル」>「すべて保管」を選択します。関係インスタンス ID を持たない関係インスタンスがあれば、InterChange Server がその関係インスタンスに対して、関係インスタンス ID を生成します。Relationship Manager が、新規の ID で “???” プレースホルダーを置き換えます。</p> <p>Relationship Manager は、すべての保管された参加者インスタンスの変更日付を今日の日付に更新します。</p>

関係インスタンスの削除

関係表から関係インスタンスを削除するには、削除する関係インスタンスを選択してから、以下の操作のいずれかを実行します。

- メニュー・バーから、「**ファイル**」>「**関係を削除**」を選択します。
- 関係インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから、「**削除**」を選択します。

関係インスタントとそのデータが、現在の関係に対する複数の関係表から削除されます。

参加者の非アクティブ化とアクティブ化

参加者インスタンスは、非活動化すなわち非アクティブ化することができます。参加者インスタンスを非アクティブ化すると、関係インスタンスから参加者インスタンスが除去され、Relationship Manager ウィンドウに表示されなくなりますが、そのレコードは関係表に残るため、後日、再アクティブ化することが可能です。

参加者の非アクティブ化

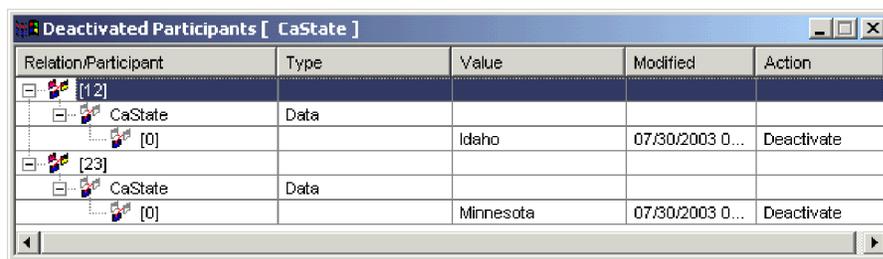
参加者インスタンスを非アクティブ化するには、非アクティブ化する参加者インスタンスを右マウス・ボタンで右クリックし、コンテキスト・メニューから「参加者を非アクティブ化」を選択します。参加者は、Relationship Manager 表示から除去されますが、関係表からは除去されません。

参加者のアクティブ化

参加者インスタンスをアクティブ化するには、以下の手順を実行します。

1. メニュー・バーから、「表示」>「参加者を非アクティブ化」を選択します。

図 65 に示すような、「非アクティブ化された参加者」ウィンドウが表示されます。



Relation/Participant	Type	Value	Modified	Action
[1 2]				
CaState	Data			
[0]		Idaho	07/30/2003 0...	Deactivate
[23]				
CaState	Data			
[0]		Minnesota	07/30/2003 0...	Deactivate

図 65. 参加者を非アクティブ化

2. リストから、アクティブ化する非アクティブ状態の参加者を含む関係インスタンスを選択します。
3. リストで、非アクティブ状態の参加者インスタンスが表示されるまで、関係インスタンスを展開します。
4. 再アクティブ化する参加者インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「アクティブ化」を選択します。
5. メニュー・バーから「編集」>「最新表示」を選択します。

アクティブ化された参加者インスタンスが、Relationship Manager ウィンドウで、その関係インスタンス内に表示されます。

注: 同一関係の参加者インスタンスが、非アクティブ化され、別の参加者がその場所に追加される (すなわち、同じインスタンス ID を割り当てられる) と元の参加者は、「非アクティブ化参加者」リストから除去されますが、データベースには残ります。

参加者のコピー

新規参加者インスタンスは、既存の参加者インスタンスをコピーして作成できます。参加者インスタンスをコピーするには、以下の手順を実行します。

1. 関係インスタンスで、参加者定義を右マウス・ボタンでクリックして、コンテキスト・メニューから、「参加者を追加」を選択します。
2. コピー元の参加者インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから、「参加者をコピー」を選択します。
3. 新たに作成する参加者インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから、「参加者を貼り付け」を選択します。

ビジネス・オブジェクト・ファイルのロードとアンロード

同じタイプのビジネス・オブジェクト・ファイルを参加者にロードすることができます。参加者にビジネス・オブジェクト・データ・ファイルをロードするには、以下の手順を実行します。

1. ビジネス・オブジェクト・ファイルをロードする先の参加者インスタンスを右マウス・ボタンでクリックし、「ビジネス・オブジェクトを持つ参加者をロード」を選択します。

「参加者ウィンドウ」ウィンドウには、図 66 に示すように、クリックした参加者インスタンスに関連するビジネス・オブジェクトが表示されます。

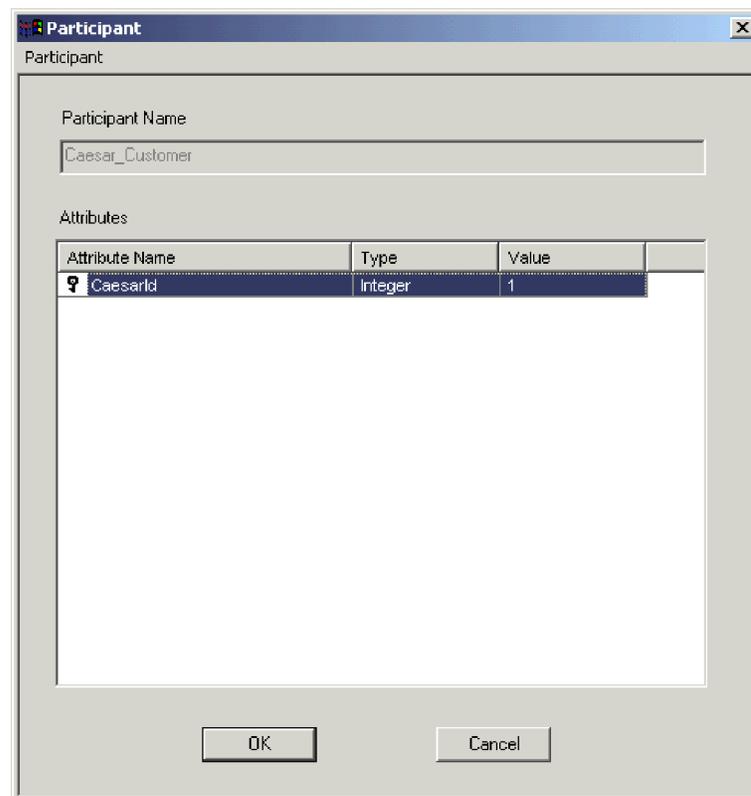


図 66. ビジネス・オブジェクトを持つ参加者をロード

2. メニュー・バーから、「参加者」>「ロード」を選択します。

3. ロードするビジネス・オブジェクト・ファイルにナビゲートし、開きます。
4. 「OK」をクリックします。

注: ビジネス・オブジェクト・ファイル内に複数のインスタンスが存在する場合は、最初の関係インスタンスのみがロードされます。

関係データの処理

Relationship Manager の重要な機能は、関係表に格納されている関係の実行時データにアクセスし、操作できる能力です。ここでは、Relationship Manager を使用して、実行時データにアクセスし、操作する方法について説明します。

参加者の検索

参加者インスタンスは、さまざまな条件に基づいて検索できます。検索条件をより具体的にすることにより、1 つの重複しない参加者インスタンスまたは参加者インスタンスのグループを検索で見つけることができます。以下のように、参加者インスタンスを検索できます。

ビジネス・オブジェクトによるインスタンスの検索

このオプションでは、そのデータ型がビジネス・オブジェクト内の属性であるインスタンスを検索します。

ビジネス・オブジェクトによるインスタンス検索を行うには、以下の手順を実行します。

1. Relationship Manager で参加者インスタンスを選択します。
2. メニュー・バーから、「編集」>「インスタンスをビジネス・オブジェクトで検索」を選択します。

Relationship Manager が、251 ページの図 66 に示すような「参加者」ウィンドウを表示します。

3. 「値」セルに、それにより検索を行う参加者値を入力します。
4. 「OK」をクリックします。

Relationship Manager が、ダイアログ・ボックスに一致するものがあれば、そのインスタンスを表示します。

5. Relationship Manager が表示したダイアログで、いずれかのインスタンスをダブルクリックし、そのインスタンスにナビゲートし、強調表示します。

データによるインスタンスの検索

このオプションでは、その型が Data であるインスタンスを検索します。

データによるインスタンス検索を行うには、以下の手順を実行します。

1. Relationship Manager で参加者インスタンスを選択します。
2. メニュー・バーから、「編集」>「インスタンスをデータで検索」を選択します。

Relationship Manager が、図 67 に示すような「インスタンスをデータで検索」ウィンドウを表示します。

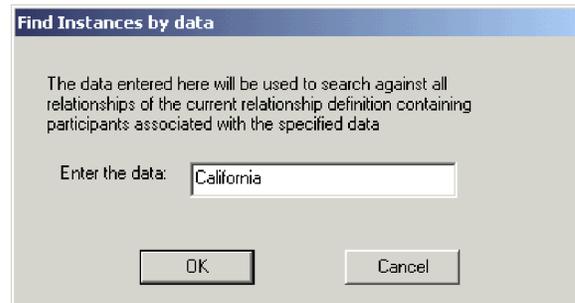


図 67. インスタンスをデータで検索

3. 「データを入力してください:」セルに、それにより検索を行う参加者値を入力します。
4. 「OK」をクリックします。

Relationship Manager が、ダイアログ・ボックスに一致するものがあれば、そのインスタンスを表示します。

5. Relationship Manager が表示したダイアログで、いずれかのインスタンスをダブルクリックし、そのインスタンスにナビゲートし、強調表示します。

表示する参加者のフィルター処理

ある日付の範囲で、作成または変更した参加者のみを表示するように、参加者をフィルターに掛けることができます。表示する参加者をフィルターに掛けるには、以下の手順を実行します。

1. Relationship Manager で参加者を選択します。
2. メニュー・バーから、「表示」>「フィルター」を選択します。

Relationship Manager が、図 64 に示すような「フィルター」ダイアログを表示します。

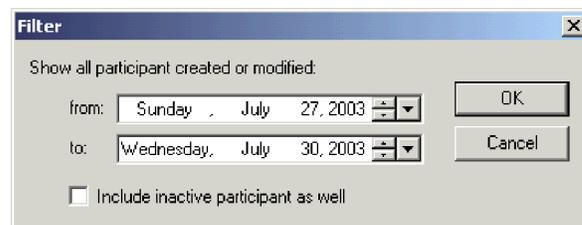


図 68. 参加者のフィルター処理結果

3. 「フィルター」ダイアログで、参加者の作成日付または変更日付の中で最も古いものを、「自:」フィールドに入力し、最も新しい日付を「至:」フィールドに入力します。

以下の方法を使用して、日付値を入力します。

- テキスト・フィールドに英字を入力して、曜日を巡回します。例えば、土曜日や日曜日に当たる日付を巡っていくには、**S** を入力します。
 - 小さな上矢印、下矢印をクリックして、日付を 1 だけ増加または減少します。
 - 大きな下矢印をクリックして、日付選択に使用できるカレンダーを表示します。
4. 非アクティブな参加者も結果の表示に含める場合は、「**アクティブでない参加者も含める**」 チェック・ボックスをクリックします。
 5. 「**OK**」 をクリックします。

Relationship Manager が、「結果をフィルター操作」ダイアログに、フィルター処理実行の間における活動履歴を表示します。フィルター処理結果の表示には、「**アクティブでない参加者も含める**」 オプション・ボックスにチェックマークが付いている場合は、非アクティブな参加者も含まれます。図 69 に、「結果をフィルター操作」ダイアログを示します。

Relation/Participant	Type	Value	Modified	Action
[-] [1]				
[-] CIState	Data			
[+] [0]		AY	07/30/2003 ...	Create
[-] [23]				
[-] CIState	Data			

図 69. フィルター処理された参加者データの表示

参加者のクリーンアップ

ソース・アプリケーションまたは汎用オブジェクト内の矛盾するデータまたは壊れたデータのため、参加者をクリーンアップする場合は、以下の操作を実行します。

1. Relationship Manager で参加者を選択します。
2. メニュー・バーから、「表示」>「参加者をクリーンアップ」を選択します。

Relationship Manager が、255 ページの図 70 に示すような「参加者をクリーンアップ」ダイアログを表示します。

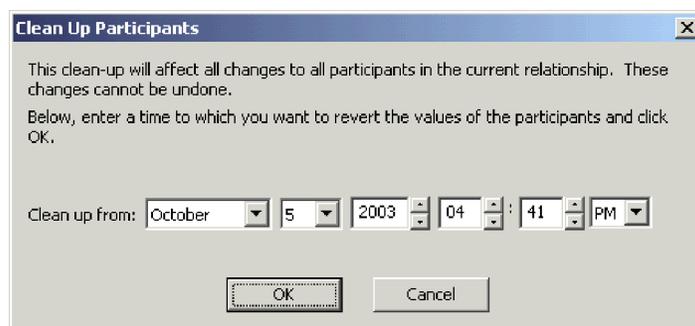


図 70. 参加者をクリーンアップ

3. 「参加者をクリーンアップ」ダイアログ・ボックスで、その時点から特定の値を元に戻す日付を「**クリーンアップ元:**」フィールドに入力します。

以下の方法を使用して、日付値を入力します。

- テキスト・フィールドに英字を入力して、曜日を巡回します。例えば、土曜日や日曜日に当たる日付を巡っていくには、**S** を入力します。
 - 小さな上矢印、下矢印をクリックして、日付を 1 だけ増加または減少します。
 - 大きな下矢印をクリックして、日付選択に使用できるカレンダーを表示します。
4. 「**OK**」をクリックします。

指定した時点から、追加、非アクティブ化、アクティブ化された参加者はすべて、データベースから削除されます。削除された参加者、またはその値が変更された参加者は、クリーンアップできません。

関係データの印刷

Relationship Manager を使用すると、関係の実行時データに関する情報を印刷できます。Relationship Manager は、そのメインウィンドウにデータを表示する形式とほぼ同様なツリー表示形式で実行時データを作成します。この出力ツリーの印刷は、以下の印刷機能を使用して、管理することができます。

- **印刷:** システム・プリンターにフォーマット済み実行時データを送信します。
- **印刷プレビュー:** Relationship Manager のウィンドウにフォーマット済み実行時データを送信します。
- **ページ設定:** 実行時データのレイアウトを変更して、ページに収まるようにします。

プリンターへの関係データの送信

Relationship Manager の印刷機能は、メインウィンドウの関係ツリーの現在の内容をプリンターに送信します。以下の操作を実行して、関係の実行時データを印刷します。

1. 印刷する情報が表示されるまで、Relationship Manager の関係ツリーを拡張します。
2. 以下の方法のいずれかを使用して、関係の実行時データを印刷します。

- メニュー・バーから、「ファイル」>「印刷」を選択します。
- キーボード・ショートカット **Ctrl+P** を使用します。
- ツールバーの「印刷」をクリックします。

関係データの印刷プレビューの使用

以下の操作を実行して、印刷された場合の状態を確認するため、関係データの出力をプレビュー表示します。

1. 印刷する情報が表示されるまで、Relationship Manager の関係ツリーを拡張します。
2. 以下の方法のいずれかを使用して、印刷プレビューを開始します。
 - メニュー・バーから、「ファイル」>「印刷プレビュー」を選択します。
 - キーボード・ショートカット **Ctrl+Alt+P** を使用します。
 - ツールバーの「印刷プレビュー」をクリックします。

ページの設定

Relationship Manager は、そのメインウィンドウにおける表示形式に従って、出力ツリーを印刷しようとします。出力ツリーが、幅が狭くてページに十分収まる場合は、特別なフォーマット設定作業を行う必要はありません。ただし、出力ツリーがページに収まらない場合は、フォーマット設定オプションを設定して、出力ツリーのレイアウトを変更してページに収まるようにします。

印刷フォーマットは、以下のいずれかの方法で開始できる「印刷ページ設定」機能を使用して設定できます。

- メニュー・バーから、「ファイル」>「ページ設定」を選択し、プリンター設定、用紙サイズ、または用紙の向きを設定できる「印刷ページ設定」ダイアログを表示します。
- キーボード・ショートカット **Shift+Alt+P** を使用します。

上記のどのコマンドを使用しても、「印刷ページ設定」ダイアログが表示されません。

「印刷ページ設定」ダイアログから、以下のフォーマット情報を指定することができます。

- 列フォーマット・オプションが、「印刷ページ設定」ダイアログの「列設定」領域に表示されます。表 19 に、この領域に表示されるオプションを示します。

表 19. 列フォーマット・オプション

列フォーマット・オプション	説明
全ページに列見出しを出力	このオプションを使用可能にすると、列見出し (Relation、Type、Value など) が、最初のページだけでなく、各ページにも印刷されます。

表 19. 列フォーマット・オプション (続き)

列フォーマット・オプション	説明
全データに合わせて列のサイズを変更	このオプションを使用可能にすると、Relationship Manager が、列自体よりも幅の広いストリングが収まるように、各列の間隔を広げようとしています。結果のツリーが、印刷ページに収まらない場合は、Relationship Manager は、列内部の余分な空気を消去しようとしています。このフィールドとそのサブフィールドがグレー表示されている場合、フィールドの値は、Relationship Manager が、出力ツリーをフォーマットするとき使用するデフォルト値を示します。Relationship Manager が出力ツリーを生成するたびに、ユーザーがフォーマット設定を確認する場合は、それらのフィールドをグレー表示状態のままにします。ユーザーが、ここのフィールドをアクティブすると、Relationship Manager は、生成する各出力ツリーに対して、ユーザーが設定したオプションを使用します。
必要に応じてフォントを縮小	このオプションを使用可能にすると、Relationship Manager は、データがページよりもまだ大きい場合に、フォント・サイズを小さくしようとしています。フォント・サイズの縮小は、3 段階で行われます。1 段階ごとに、フォント・サイズは、20% 小さくなります。
新しいフォント・サイズについて確認	このオプションを使用可能にすると、Relationship Manager は、出力ツリーがページに収まるように、フォント・サイズを十分縮小した後、どのように操作を継続するかを、プロンプトでユーザーに選択するように要求します。以下の選択項目が用意されています。 <ul style="list-style-type: none"> • Print with this font • Restore and print with the original font size • Cancel printing フォント・サイズの縮小過程で、フォント・サイズは、元のサイズの半分になる場合があります。このオプションを使用不可にすると、印刷は、Relationship Manager がページに出力ツリーが収まるフォント・サイズを検出した時点で開始されます。
印刷/プレビュー時に変更した場合は列幅を調整	このオプションを使用可能にすると、Relationship Manager は、ページに収まるように出力ツリーを調節するために行った列サイズに対する変更内容を保管します。以降、出力ツリーの印刷時に、保管した設定が使用されます。

「データに合わせて列のサイズを変更 (Resize columns to fit data)」フィールドとその関連のサブフィールドは、Relationship Manager が、出力ツリーをフォーマット設定するときに行う操作を示します。それらのフィールドは、以下の 3 つの状態のいずれかで表示されます。

- チェックマークが選択されていない (フィールドは空): Relationship Manager が、出力ツリーをフォーマットするときに関連するタスクを実行しないことを示します。

- チェックマークが選択され、グレー表示になっている: Relationship Manager が、出力ツリーをフォーマットするときに関連するタスクを条件実行すること、すなわち、Relationship Manager が、関連の操作を行う前に確認ダイアログを表示することを示します。
- チェックマークが選択されている: Relationship Manager が、出力ツリーをフォーマット設定するときに関連のタスクを無条件に実行すること、すなわち、Relationship Manager が、関連の操作を行う前に確認ダイアログを表示しないことを示します。

表 19 に示すように、これらのフィールドは、最初は、灰色状態で表示されます。これらのフィールドは、フォーマット設定処理の間に表示される確認ダイアログから設定できます。例えば、ある列の値が、その列の現在の幅では収まらない場合、Relationship Manager は、「はい」と「いいえ」オプション、および「これをデフォルトの選択内容にする」フィールドを持つ確認ダイアログを表示します。「これをデフォルトの選択内容にする」チェック・ボックスを選択すると、Relationship Manager が、以下のように、確認ダイアログからユーザーが選択したオプションに基づいて、「データに合わせて列のサイズを変更 (Resize columns to fit data)」フィールドを設定します。

- はい: Relationship Manager は、関連のチェック・ボックスの選択を解除します。したがって、関連の操作を実行しません。
- いいえ: Relationship Manager は、関連のチェック・ボックスを選択します。したがって、関連の操作を無条件に実行します。
- 余白フォーマット・オプションが、「印刷ページ設定」ダイアログの「余白設定」領域に表示されます。表 20 に、この領域に表示されるオプションを示します。

表 20. 余白フォーマット・オプション

余白フォーマット・オプション	説明
最大ページ領域を使用	このオプションを使用可能にすると、Relationship Manager は、(実際のプリンター性能に従って) ページの最大印字領域を使用します。
以下の余白(インチ)を使用	このオプションを使用可能にすると、左右上下の余白を指定の値に設定できます。

- ページ・フォーマット・オプションが、「印刷ページ設定」ダイアログの「ページ設定」領域に表示されます。表 21 に、この領域に表示されるオプションを示します。

表 21. ページ・フォーマット・オプション

ページ・フォーマット・オプション	説明
全ページにページ見出しを印刷	このオプションを使用可能にすると、Relationship Manager は、最初のページだけでなく、各ページの最上部にも、関係名と時刻を印刷します。
ページ・フッターを印刷	このオプションを使用可能にすると、Relationship Manager は、各ページの最下部にページ番号を印刷します。

表 21. ページ・フォーマット・オプション (続き)

ページ・フォーマット・オプション	説明
印刷プレビューと文書を同期	このオプションを使用可能にすると、Relationship Manager は、印刷プレビューに最後に表示した関係データをそのメインウィンドウに表示します。例えば、印刷のプレビュー版の 3「ページ」目まで表示した後、印刷プレビューを終了すると、Relationship Manager は、関連データをスクロールし、そのメインウィンドウに印刷のプレビュー版の 3 ページ目を表示します。このオプションを使用不可にすると、Relationship Manager は、そのメインウィンドウに印刷プレビューの実行前に表示したのと同じデータを表示します。

- グリッド・フォーマット・オプションが、「印刷ページ設定」ダイアログの「グリッド設定」領域に表示されます。表 22 に、この領域に表示されるオプションを示します。

表 22. グリッド・フォーマット・オプション

グリッド・フォーマット・オプション	説明
グリッドを印刷 (リセットするにはチェックマークを外す)	このオプションを使用可能にすると、Relationship Manager は、行と列を区分するため、出力ツリーにグリッドを表示します。このフィールドを 2 回クリックすると、出力ツリーに、縦方向と横方向のグリッド線が表示されます。
水平グリッドのみ	このオプションを使用可能にすると、Relationship Manager は、出力ツリーに (行を区分するため) 水平方向のグリッドのみを表示します。出力ツリーが、水平方向のグリッドを表示しない場合、各印刷ページは、2 から 3 行のデータを余分に収めることができます。
垂直グリッドのみ	このオプションを使用可能にすると、Relationship Manager は、出力ツリーに (列を区分するため) 垂直方向のグリッドのみを表示します。

第 11 章 Test Connector の使用

Test Connector は、コネクターのアクティビティをシミュレートすることによって、実際にコネクターを実行する手間を省いて統合コンポーネントをテストできるようにします。この章を構成するセクションは次のとおりです。

- 『推奨されるテスト手順』
- 262 ページの『Test Connector の始動』
- 263 ページの『Test Connector のシャットダウン』
- 263 ページの『コネクター・プロファイルの作成および編集』
- 264 ページの『コネクターのエミュレート』
- 265 ページの『ビジネス・オブジェクトの処理』

推奨されるテスト手順

WebSphere Business Integration システムでコンポーネントをテストするための推奨されるテスト手順を以下に示します。

1. ご使用の統合ブローカーが InterChange Server の場合、System View ビューの使用を検討してください。送信したフローが成功したか失敗したかを判断するのに大変便利です。

詳しくは、「システム管理ガイド」を参照してください。

2. Test Connector をセットアップし、ソース・コネクターをエミュレートします。
 - a. Test Connector を起動します。詳細については、262 ページの『Test Connector の始動』を参照してください。
 - b. インターフェースにソース・コネクターのプロファイルを作成します。詳細については、263 ページの『新規プロファイルの作成』を参照してください。
 - c. Test Connector をエージェントに接続し、ソース・コネクターのエミュレーションを開始します。詳細については、264 ページの『コネクターのエミュレート』を参照してください。
3. Test Connector のインスタンスをセットアップし、インターフェースに関する各宛先コネクターをエミュレートします。
 - a. Test Connector を起動します。詳細については、262 ページの『Test Connector の始動』を参照してください。
 - b. 宛先コネクターのプロファイルを作成します。詳細については、263 ページの『新規プロファイルの作成』を参照してください。
 - c. Test Connector をエージェントに接続し、宛先コネクターのエミュレーションを開始します。詳細については、264 ページの『コネクターのエミュレート』を参照してください。
 - d. インターフェースに関するすべての宛先コネクターについて、上記の 3a から 3c を繰り返します。

4. 各「Test Connector」ウィンドウでエミュレートされるコネクタを容易に識別できるように Test Connector のインスタンスを画面に配置します。
5. ソース・コネクタから要求ビジネス・オブジェクトを送信します。ソース Test Connector で以下の手順を実行します。
 - a. テストが必要なインターフェースによって管理されるビジネス・オブジェクトを作成します。詳細については、265 ページの『要求ビジネス・オブジェクトの作成』を参照してください。
 - b. ファイルにビジネス・オブジェクトを保管し、以後のテストで使用できるようにします。詳細については、269 ページの『ビジネス・オブジェクトの保管』を参照してください。
 - c. ビジネス・オブジェクトを送信します。詳細については、266 ページの『要求ビジネス・オブジェクトの送信』を参照してください。
6. 宛先コネクタから要求ビジネス・オブジェクトに対する応答をシミュレートします。宛先 Test Connector のウィンドウで以下の手順を実行します。
 - a. 要求ビジネス・オブジェクトを受け入れます。詳細については、270 ページの『要求ビジネス・オブジェクトの受け入れ』を参照してください。
 - b. 応答としてビジネス・オブジェクトを送信します。詳細については、271 ページの『応答ビジネス・オブジェクトの送信』を参照してください。
7. 各インターフェースのテストに必要な回数だけステップ 5 からステップ 6 を繰り返します。

Test Connector の始動

Test Connector を始動するには、統合ブローカーに応じて以下のいずれかの操作を実行します。

- 統合ブローカーが InterChange Server である場合は、「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「開発」>「Test Connector」を選択します。
- ご使用の統合ブローカーが WebSphere Application Server または WebSphere Message Broker (WebSphere MQ Integrator、WebSphere MQ IntegratorBroker、または WebSphere Business Integration Message Broker) の場合、「スタート」>「プログラム」>「IBM WebSphere Business Integration Adapters」>「ツール」>「Test Connector」を選択します。

「Test Connector」ウィンドウには以下のペインがあります。

- 「ビジネス・オブジェクト・エディター (Business Object Editor)」ペインでは、送信するビジネス・オブジェクト・インスタンスを作成できます。
- 「ビジネス・オブジェクト要求リスト」ペインには、コネクタが受信したビジネス・オブジェクト要求が表示されます。
- 「出力」ペインには、ビジネス・オブジェクトが送信されたときなどに、Test Connector の操作に関するメッセージが表示されます。

Test Connector のシャットダウン

Test Connector をシャットダウンし、コネクタ・エージェントのエミュレーションを停止するには、メニュー・バーから「ファイル」>「終了」を選択します。「シャットダウン」プロンプトが表示された場合は、「はい」をクリックします。

コネクタ・プロファイルの作成および編集

Test Connector は、プロファイルを使用してコネクタのエミュレートに必要な情報を格納します。プロファイルは、エミュレート対象のコネクタごとに作成する必要があります。既存のプロファイルを編集および削除することができます。

ファイルへのコネクタ定義の保管

Test Connector を使用してコネクタをエミュレートするには、ファイルにコネクタ定義を保管する必要があります。ファイルにコネクタ定義を保管するには、以下の手順を実行します。

1. Connector Configurator でコネクタ定義を開きます。
2. メニュー・バーから「ファイル」>「別名保管」>「ファイルに」を選択します。
3. ファイルを保管するディレクトリーにナビゲートし、「ファイル名」フィールドに名前を入力し、「タイプとして保管 (Save as type)」ドロップダウン・メニューに値「構成ファイル (*.cfg)」が表示されていることを確認し、「保管」をクリックします。

Connector Configurator が、コネクタ定義を指定の名前のファイルに保管します。

新規プロファイルの作成

プロファイルは、Test Connector でエミュレートするすべてのコネクタについて作成する必要があります。プロファイルは、コネクタの名前、使用する構成ファイル、コネクタが通信する統合ブローカーのタイプなどの情報を指定します。新規コネクタ・プロファイルを作成するには、以下の手順を実行します。

1. メニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択し、「コネクタ・プロファイル」ウィンドウを表示させます。
2. 「コネクタ・プロファイル」ウィンドウで、メニュー・バーから「ファイル」>「新規プロファイル」を選択します。
3. 「新規プロファイル」ウィンドウで「参照」をクリックし、『ファイルへのコネクタ定義の保管』で準備しているコネクタの構成ファイルにナビゲートします。
4. 「コネクタ名」フィールドにコネクタの名前を入力します。コネクタ定義の名前は、必ず統合ブローカー・リポジトリに存在する正確な名前を入力してください。例えば、JText 用アダプターの場合は JTextConnector と入力する必要があります。ワード JText と Connector の間にはスペースを入れず、大文字と小文字も正確に区別してください。
5. 「ブローカー・タイプ」ドロップダウン・メニューで適切な統合ブローカー (ICS、WMQI または WAS) を選択します。

注: ご使用のブローカーが WebSphere Message Broker の場合、WMQI を選択してください。

6. ステップ 5 (263 ページ) でブローカー・タイプとして ICS を選択した場合は、以下の手順も実行します。
 - a. 「サーバー」フィールドに InterChange Server インスタンスの名前を入力します。

名前は正確に入力してください。この名前では大文字と小文字が区別されません。名前が誤っている場合、Test Connector は InterChange Server と通信できません。

- b. 「ユーザー ID」と「パスワード」フィールドは、セキュリティーを使用可能にした ICS ブローカー・タイプを使用している場合にのみ、使用され、検証されます。それ以外の場合、これらのフィールドはオプションです。
7. 「OK」をクリックし、「新規プロファイル」ウィンドウを閉じます。

「コネクタ・プロファイル」ウィンドウの「コネクタ」列にはコネクタの名前が表示され、「サーバー」列には InterChange Server インスタンスの名前が表示され (統合ブローカーが ICS の場合)、「構成ファイル」列にはコネクタ構成ファイルのパスおよび名前が表示されます。

8. 「OK」をクリックし、「コネクタ・プロファイル」ウィンドウを閉じます。

プロファイルの編集

既存のコネクタ・プロファイルを変更するには、以下の手順を実行します。

1. Test Connector のメニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択するか、キーボード・ショートカット **Ctrl+N** を使用して「コネクタ・プロファイル」ウィンドウを表示させます。
2. 「コネクタ・プロファイル」ウィンドウで、編集するプロファイルを選択し、メニュー・バーから「編集」>「プロファイルを編集」を選択します。
3. 「新規プロファイル」ウィンドウのフィールドに新しい値を入力し、「参照」ボタンを使用し、必要に応じて構成ファイルを変更して編集します。
4. 「OK」をクリックし、「新規プロファイル」ウィンドウを閉じます。

プロファイルの削除

コネクタ・プロファイルを削除するには、以下の手順を実行します。

1. Test Connector のメニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択するか、キーボード・ショートカット **Ctrl+N** を使用して「コネクタ・プロファイル」ウィンドウを表示させます。
2. 「コネクタ・プロファイル」ウィンドウで、削除するプロファイルを選択し、メニュー・バーから「編集」>「プロファイルを削除」を選択します。

コネクタのエミュレート

コネクタのプロファイルを作成すると、そのプロファイルを使用して Test Connector をエージェントに接続できます。Test Connector をエージェントに接続すると、Test Connector は、選択されたプロファイルで定義されているコネクタのエミュレーションを開始します。

Test Connector をエージェントに接続するには、以下の手順を実行します。

1. Test Connector のメニュー・バーから「ファイル」>「プロファイルを作成/選択」を選択します。
2. 「コネクタ・プロファイル」ウィンドウで、プロファイルを開くコネクタの名前を選択します。
3. 「OK」をクリックします。
4. メニュー・バーから「ファイル」>「接続」を選択します。

Test Connector がコネクタのエミュレーションを開始するときには、「出力」ペインにメッセージが表示されます。接続が完了すると、「出力」ペインに作動可能であることを示すメッセージが表示され、**ビジネス・オブジェクト・タイプ**のリストが「**ビジネス・オブジェクト・エディター (Business Object Editor)**」ペインに取り込まれます。

ビジネス・オブジェクトの処理

ビジネス・プロセス・インターフェースが正しく開発されているかどうかを調べるには、ビジネス・オブジェクトを正常に交換および処理できることを確認する必要があります。このセクションでは、以下の作業の方法について説明します。

- ビジネス・オブジェクト・テスト・データの作成、変更、削除、および保管
- ビジネス・オブジェクトの属性値の比較による処理中の変更内容の簡単な表示
- ビジネス・オブジェクトの送受信

要求ビジネス・オブジェクトの処理

要求ビジネス・オブジェクトは、インターフェースを起動するイベントのソースであるコネクタを Test Connector がエミュレートしているときに Test Connector から送信するビジネス・オブジェクトです。要求ビジネス・オブジェクトの処理は、ビジネス・オブジェクト・インスタンスの作成、データの取り込み、要求の送信から構成されます。

要求ビジネス・オブジェクトの作成

Test Connector に新規ビジネス・オブジェクトを作成するには、以下の手順を実行します。

1. 「**ビジネス・オブジェクト・エディター (Business Object Editor)**」ペインで、「**ビジネス・オブジェクト・タイプ**」ドロップダウン・メニューから、作成するビジネス・オブジェクトの名前を選択します。
2. 「**ビジネス・オブジェクト・インスタンス**」フィールドの横にある「**作成**」をクリックします。
3. 「**新規インスタンス**」ダイアログが表示されたら、「**名前を入力してください**」フィールドにインスタンスの名前を入力します。
4. 「**動詞**」ドロップダウン・メニューから適切な動詞を選択します。
5. 「**ビジネス・オブジェクト・ロケール**」ドロップダウン・メニューから適切なロケールを選択します。

6. トップレベル・オブジェクト内の単純属性および子ビジネス・オブジェクトの値を指定します。詳細については、268 ページの『ビジネス・オブジェクト属性の値の設定』を参照してください。
7. 「OK」をクリックします。

要求ビジネス・オブジェクトの送信

ビジネス・オブジェクトを作成またはロードし、属性の値を指定すると、いくつかの方法でそのビジネス・オブジェクトを要求として統合ブローカーに送信できます。

要求ビジネス・オブジェクトの非同期送信: ソース・コネクタが非同期モードで要求ビジネス・オブジェクトを送信する場合は、応答ビジネス・オブジェクトの取得を予期していません。要求ビジネス・オブジェクトがディスパッチされると、トランザクションにおけるソース・コネクタの役割は終了します。通常、応答ビジネス・オブジェクトは統合ブローカーによって処理されます。Test Connector のデフォルトのモードは非同期モードです。

ビジネス・オブジェクトを非同期的に送信するには、以下の手順を実行します。

1. メニュー・バーから「要求」>「モード」>「非同期」を選択します。

注: Test Connector はデフォルトでは「非同期」モードで動作するため、このステップは、以前にコネクタから同期要求を送信している場合にのみ実行する必要があります。また、各要求を送信する前にモードを設定する必要はありません。

2. メニュー・バーから「要求」>「送信」を選択します。

コネクタ定義で指定したブローカーが InterChange Server の場合は、ビジネス・オブジェクト要求がサーバーに送信され、処理されます。

コネクタ定義で指定したブローカーが、サポートされるメッセージ・ブローカーか WebSphere Application Server の場合には、ビジネス・オブジェクトは RequestQueue 標準プロパティで指定しているキューに置かれます。

要求ビジネス・オブジェクトの同期送信: ソース・コネクタが要求ビジネス・オブジェクトを同期的に送信する場合は、宛先アプリケーションが要求を処理した後、統合ブローカーから応答ビジネス・オブジェクトを取得することを予期しています。同期モードでは、Test Connector は、ソース・コネクタの Synchronous Request Queue プロパティによって指定されたキューに応答ビジネス・オブジェクトを入れます。Test Connector のデフォルトのモードは非同期モードです。

1. メニュー・バーから「要求」>「モード」>「同期」を選択し、Test Connector を同期モードに設定します。
2. メニュー・バーから「要求」>「送信」を選択します。
3. コネクタ定義で指定したブローカーが InterChange Server の場合は、「コラボレーションの選択」ダイアログが表示されます。ビジネス・オブジェクトの送信先コラボレーションを「コラボレーション」ドロップダウン・メニューから選択し、「OK」をクリックします。

コネクタ定義で指定したブローカーが InterChange Server の場合は、処理を選択したコラボレーション・オブジェクトの構成済みポートにビジネス・オブジェクト要求が送信されます。

コネクタ定義で指定したブローカーが サポートされるメッセージ・ブローカーか、WebSphere Application Server の場合には、ビジネス・オブジェクトは SynchronousRequestQueue 標準プロパティで指定しているキューに置かれます。

バッチ・モードでの要求ビジネス・オブジェクトの送信: バッチ・モードでは、送信する特定のビジネス・オブジェクトのインスタンスの数を Test Connector で指定できます。また、インスタンスごとに固有値に設定するトップレベル・オブジェクトの 1 つの属性 (基本キー属性など) も指定できます。Test Connector は、指定した単一属性の値を増分しながら、指定した回数だけビジネス・オブジェクトをコピーし、各ビジネス・オブジェクトを送信します。このオプションにより、多数のビジネス・オブジェクトを素早く簡単に作成できます。

選択した属性が一致関係の一部として動的相互参照に参加するキー・フィールドである場合は、初期値およびそれに従う値はすべて固有でなければなりません。さもなければ、相互参照ロジックが失敗し、要求ビジネス・オブジェクトも失敗します。

値が固有であることを確認するには、Relationship Manager を使用するか、以下のように関係参加者の表に対して SQL ステートメントを実行します。

- 参加者の現在の最高値を判別し、「初期値」フィールドをそれより高い値に設定します。バッチの最初のビジネス・オブジェクト・インスタンスおよびその後のインスタンスが固有になります。
- 参加者の既存の表エントリーを削除します。これにより、バッチ・ビジネス・オブジェクトの値と同じ属性値を持つエントリーが存在しないことが保証されます。

バッチ・モードでビジネス・オブジェクトを送信するには、以下の手順を実行します。

1. 「**ビジネス・オブジェクト・タイプ**」ドロップダウン・メニューから、送信するビジネス・オブジェクトの名前を選択します。
2. メニュー・バーから「**要求**」>「**バッチを送信**」を選択します。
3. 「**バッチ・モード**」ウィンドウで、「**動詞**」ドロップダウン・メニューから、必要な動詞を選択する。
4. 「**ビジネス・オブジェクト・ロケール**」ドロップダウン・メニューから適切なロケールを選択します。
5. バッチの各ビジネス・オブジェクト要求で増分するトップレベル・ビジネス・オブジェクトの属性を「**属性**」リストから選択します。

通常、選択する属性は、ビジネス・オブジェクトを一意的に識別する属性 (基本キーなど) にしてください。

6. 「**初期値**」フィールドに、増分する属性の開始値を入力します。
7. 「**ビジネス・オブジェクト数**」フィールドに、生成および送信するビジネス・オブジェクトの数を入力します。
8. 「**OK**」をクリックします。

Test Connector が、指定した数のビジネス・オブジェクトを生成します。これらのビジネス・オブジェクトでは指定の属性の値がインスタンスごとに増分され、それ以外はすべて同一です。

コネクタ定義で指定したブローカーが InterChange Server の場合は、ビジネス・オブジェクト要求がサーバーに送信され、処理されます。

コネクタ定義で指定したブローカーが、サポートされるメッセージ・ブローカー WebSphere Application Server の場合には、ビジネス・オブジェクトは RequestQueue 標準プロパティで指定しているキューに置かれます。

ビジネス・オブジェクト属性の値の設定

以下のセクションでは、ビジネス・オブジェクト・インスタンスの単純属性および複合属性の値を設定するためのさまざまな方法について説明します。

- 『単純属性の値の設定』
- 『子ビジネス・オブジェクトの追加』
- 『子ビジネス・オブジェクトの除去』
- 269 ページの『子ビジネス・オブジェクトの動詞の設定』

単純属性の値の設定

単純属性の値を指定するには、「値」列で該当するセルをクリックし、値を入力します。

子ビジネス・オブジェクトの追加

子ビジネス・オブジェクトのインスタンスを追加するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「インスタンスを追加」を選択します。

子ビジネス・オブジェクトを表す属性の横に正符号 (+) が追加され、少なくとも 1 つの子ビジネス・オブジェクト・インスタンスが存在することが示されます。子オブジェクト属性を展開すると、インスタンスごとに番号付きのエントリが表示されます。個々のインスタンスの横にも正符号 (+) があるため、個々のインスタンスを展開したり属性の値を設定したりすることができます。

さらに子ビジネス・オブジェクト・インスタンスを追加するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「インスタンスを追加」を選択します。

注: 子ビジネス・オブジェクトを参照する属性の「カード」プロパティが値 1 (単一カーディナリティ) に設定されている場合は、子オブジェクトに追加できるインスタンスは 1 つのみです。

子ビジネス・オブジェクトの除去

子ビジネス・オブジェクトのインスタンスを除去するには、インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「インスタンスを削除」を選択します。

子ビジネス・オブジェクトのすべてのインスタンスを削除するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「全インスタンスを削除」を選択します。

子ビジネス・オブジェクトの動詞の設定

子ビジネス・オブジェクトの動詞を設定すると、値がビジネス・プロセスに及ぼす影響をテストできます。これは、子オブジェクトの相互参照用を含むロジックのトラブルシューティングを行う場合に役立ちます。

子ビジネス・オブジェクト・インスタンスの動詞を設定するには、その子ビジネス・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「動詞を設定」を選択します。「動詞の選択」プロンプトが表示されたら、適切な動詞を選択し、「OK」をクリックします。

「応答ビジネス・オブジェクト」ツールバーの使用

宛先コネクタによって受信されたビジネス・オブジェクトの属性は、応答として送信する前に編集できます。この場合には「応答ビジネス・オブジェクト」ダイアログのツールバーを使用します。このツールバーにあるツールバー・ボタンを使用すると、ビジネス・オブジェクトの値を設定できます。詳細については、271 ページの『応答ビジネス・オブジェクトの編集』を参照してください。

ビジネス・オブジェクトの保管

Test Connector でビジネス・オブジェクトを保存すると、将来のテストに使用したり、テクニカル・サポートと共用して問題のトラブルシューティングに使用したり、応答データとして使用したりすることができます。作成したビジネス・オブジェクトや宛先コネクタの「Test Connector」ウィンドウに要求として表示されるビジネス・オブジェクトを含め、任意のビジネス・オブジェクトを保管できます。デフォルトでは、ビジネス・オブジェクトはビジネス・オブジェクト拡張子 (.bo) が付いたファイルに保管されます。

(特にテスト・データ・ファイルの場合は) 各インターフェースまたは各コネクタに専用のサブディレクトリーを持つディレクトリーまたはディレクトリー構造を作成することをお勧めします。この編成により、必要なファイルを探しやすくなり、テストが効率的になります。さらに、ビジネス・オブジェクトのテスト・データ・ファイルにはそのビジネス・オブジェクト定義と同じ名前を付けることをお勧めします。

要求ビジネス・オブジェクトの保管

要求として作成したビジネス・オブジェクト・インスタンスを保管するには、以下の手順を実行します。

1. 保管するビジネス・オブジェクトを選択してください。
2. メニュー・バーから「編集」>「ビジネス・オブジェクトを保管」を選択します。
3. 適切なディレクトリーにナビゲートし、「ファイル名」フィールドにファイルの名前を指定します。
4. 「保管」をクリックします。

応答ビジネス・オブジェクトの保管

Test Connector の宛先インスタンスによって受信され、応答として送信されるビジネス・オブジェクト・インスタンスを保管するには、以下の手順を実行します。

1. 「ビジネス・オブジェクト要求リスト」 ペインでビジネス・オブジェクト・インスタンスを選択します。
2. メニュー・バーから「要求」>「応答を編集」を選択します。
3. 「ビジネス・オブジェクトを保管」をクリックします。
4. 適切なディレクトリーにナビゲートし、「ファイル名」フィールドにファイルの名前を指定します。
5. 「保管」をクリックします。

ビジネス・オブジェクトのロード

ファイルに保管したビジネス・オブジェクトをロードするには、以下の手順を実行します。

1. Test Connector のメニュー・バーから「編集」>「ビジネス・オブジェクトをロード」を選択します。
2. ビジネス・オブジェクトのテスト・データ・ファイルにナビゲートして開きます。
3. 「新規インスタンス」ダイアログが表示されたら、「名前を入力してください」フィールドにインスタンスの名前を入力します。
4. 「OK」をクリックします。

ビジネス・オブジェクトの削除

Test Connector からビジネス・オブジェクトを削除するには、メニュー・バーから「編集」>「ビジネス・オブジェクトを削除」を選択します。

注: このアクションでは Test Connector からビジネス・オブジェクトが削除されるのみです。ビジネス・オブジェクト定義に対するコネクターのサポートは削除されません。

要求ビジネス・オブジェクトの受け入れ

要求としてビジネス・オブジェクトを送信すると、トランザクションが失敗しなかった場合は、インターフェースの宛先コネクターをエミュレートしている Test Connector インスタンスの「ビジネス・オブジェクト要求リスト」ペインにビジネス・オブジェクトが表示されます。

要求ビジネス・オブジェクトを受け入れると、必要に応じて編集できます。詳細については、271 ページの『応答ビジネス・オブジェクトの編集』を参照してください。

応答ビジネス・オブジェクトの処理

応答ビジネス・オブジェクトは、インターフェースでビジネス・オブジェクト要求の宛先となるコネクターをエミュレートしているときに Test Connector から送信するビジネス・オブジェクトです。応答ビジネス・オブジェクトの処理は、ビジネス・オブジェクト・インスタンスの値の編集およびブローカーへの応答の送信から構成されます。

応答ビジネス・オブジェクトの編集

Test Connector の宛先インスタンスでビジネス・オブジェクト要求を受信したときには、一般に属性の値を編集します。例えば、関係に参加する基本キー属性の固有値を指定したり、他の属性の値を変更してビジネス・オブジェクトの実際の値に応じて応答を変えるマップまたはコラボレーション・ロジックをテストしたりすることができます。

要求ビジネス・オブジェクトを受信すると、Test Connector は「応答ビジネス・オブジェクト」ウィンドウを表示します。

ビジネス・オブジェクト属性の値を設定するには、以下の手順を実行します。

1. 「ビジネス・オブジェクト・エディター (Business Object Editor)」ペインで、ビジネス・オブジェクト・インスタンスを選択します。
2. メニュー・バーから「要求」>「応答を編集」を選択します。
3. 以下の操作を実行し、ビジネス・オブジェクトの属性を編集します。
 - 268 ページの『ビジネス・オブジェクト属性の値の設定』で説明しているいずれかの方法を使用してビジネス・オブジェクト属性の値を変更します。
 - 「ビジネス・オブジェクトをデフォルトにリセット」をクリックし、ビジネス・オブジェクト属性の値をビジネス・オブジェクト定義で指定されたデフォルト値に設定します。
 - 「ビジネス・オブジェクト値をクリア」をクリックし、ビジネス・オブジェクトのすべての属性の値をクリアします。
 - 「ビジネス・オブジェクトをロード」をクリックし、ビジネス・オブジェクトの属性にファイルのテスト・データを取り込みます。

保管したデータをビジネス・オブジェクト要求にロードする機能は、応答として送信する前に応答ビジネス・オブジェクトにデータを取り込む必要がある場合に非常に有用です。応答データを必要とする属性ごとに値を手動で入力する代わりに、値を 1 回入力し、ビジネス・オブジェクトを保管しておく (269 ページの『ビジネス・オブジェクトの保管』を参照)、以後のテストでは保管したデータをロードできます。

応答ビジネス・オブジェクトの送信

要求ビジネス・オブジェクトを受け入れたら、必要に応じてビジネス・オブジェクトを編集し、応答として送信します。

表 23 に、Test Connector の応答オプション、および C++ コネクタと Java コネクタの両方における対応するコネクタ戻りコードを示します。C++ コネクタまたは Java コネクタの戻りコードについては、「コネクタ開発ガイド (Java 用)」または「コネクタ開発ガイド (C++用)」を参照してください。

表 23. Test Connector の応答タイプおよびコネクタ戻りコード

Test Connector の 応答タイプ	C++ コネクタの戻りコード	Java コネクタの戻りコード
成功	BON_SUCCESS	SUCCESS
失敗	BON_FAIL	FAIL
複数のヒット	BON_MULTIPLE_HITS	MULTIPLE_HITS

表 23. Test Connector の応答タイプおよびコネクタ戻りコード (続き)

Test Connector の 応答タイプ	C++ コネクタの戻りコード	Java コネクタの戻りコード
内容による検索に失敗	BON_FAIL_RETRIEVE_BY_CONTENT	RETRIEVEBYCONTENT_FAILED
検出なし	BON_BO_DOES_NOT_EXIST	BO_DOES_NOT_EXIST
重複する値	BON_VALDUPES	VALDUPES

要求ビジネス・オブジェクトに応答するには、以下の手順を実行します。

1. 「ビジネス・オブジェクト要求リスト」ペインでビジネス・オブジェクトを選択します。
2. メニュー・バーから「要求」>「応答」を選択します。
3. 「応答」サブメニューから項目を選択します。

ビジネス・オブジェクト・インスタンスの比較

Test Connector は、同じタイプの 2 つのビジネス・オブジェクトを比較し、値が異なる属性を表示できます。この機能を使用すると、トランザクション実行の異なる時点におけるビジネス・オブジェクトに対する変更を表示させることができます (例えば、統合ブローカーに送信されたビジネス・オブジェクトを、統合ブローカーが更新する前と後で比較できます)。2 つのビジネス・オブジェクトを比較するには、以下の手順を実行します。

1. 265 ページの『要求ビジネス・オブジェクトの作成』または 270 ページの『ビジネス・オブジェクトのロード』の手順に従い、要求ビジネス・オブジェクト・インスタンスを作成します。
2. 「ビジネス・オブジェクト要求リスト」ペインで要求ビジネス・オブジェクト・インスタンスと比較する応答ビジネス・オブジェクト・インスタンスを選択します。
3. メニュー・バーから「編集」>「ビジネス・オブジェクトを比較」を選択します。

Test Connector によって「ビジネス・オブジェクトの比較」ウィンドウが開かれ、2 つのビジネス・オブジェクトで値が異なる属性が表に表示されます。

4. 「OK」をクリックしてウィンドウを閉じます。

第 12 章 統合テスト環境の使用

統合テスト環境とはビジネス・プロセス・インターフェース全体を適宜テストすることができる、ワークベンチ・パースペクティブです。1 ページの『インターフェース』で説明するように、インターフェースはコラボレーション・オブジェクトを中心とし、したがって統合テスト環境でのテストもコラボレーション・オブジェクトを中心に実行されます。

統合テスト環境で、テストを実行するには、テスト・サーバーとして InterChange Server を登録する、テスト・ユニットを作成する、インターフェース内のコンポーネントをサーバーに配置する、サーバーを開始する、インターフェースでコネクタをエミュレートする、およびコネクタ間でビジネス・オブジェクトを交換するなど、さまざまなタスクを実行する必要があります。以下に、統合テスト環境の使用についての特性を示します。

- 一部のタスクは、1 回しか実行する必要はありません。例えば、インターフェースのテスト・ユニットは、1 回作成する必要があるだけです。
- 複数回実行する必要のあるタスクもあります。例えば、特定の属性内の値を変更する場合のインターフェースの対応方法をテストすることがあります。この場合、インターフェースのビジネス・オブジェクト要求を複数回送信する必要があります。
- 一部のタスクは複数の方法で実行できます。例えば、テスト・ユニットを準備する前にコンポーネントをサーバーに配置、タスク・マネージャー・ビューを使用してテスト・ユニットのすべてのコンポーネントを配置、またはテスト・ユニット・ビューを使用して単一コンポーネントの配置などを行うことができます。

この章のほとんどのセクションでは、テストを実行するための大きなタスクの一部として実行する必要のあるタスクの説明か、または統合テスト環境インターフェースの特定の要素について説明します。

333 ページの『統合テスト環境を使用したテストの実行』セクションでは、単一インターフェースの単一テストを実行するために通常従う、ワークフローについて説明します。これは、特定のサブタスクまたはインターフェース・エレメントを説明したこの章のセクションの参照リストであり、また統合テスト環境を使用するためのタスク指向構造が説明され、一方他のセクションはよりインターフェース指向の説明になります。いくつかの方法の内の 1 つでタスクを実行する状況の場合、ほとんどの状況で最も効果的で効率的な方法を参照します。この章をできる限り有効に使用するには、333 ページの『統合テスト環境を使用したテストの実行』で説明するワークフローに従うことをお勧めしますが、推奨した方法がユーザーの環境や状況に適切でない場合、他のセクションの特定のサブタスクで置き換えます。

テスト・サーバーとしての InterChange Server の登録

統合テスト環境を使用してインターフェースをテストするには、InterChange Server インスタンスがテスト・モードで始動できるように登録する必要があります。これにより、統合テスト環境で以下のようなことを実行できます。

- InterChange Server を始動するバッチ・ファイルを検索し、統合テスト環境インターフェース内から始動できるようにします。
- InterChange Server のログおよびトレース情報を、コンソールでなく統合テスト環境内のビューにリダイレクトします。

InterChange Server をテスト・モードで始動するように登録するには、35 ページの『InterChange Server インスタンスの登録』を参照して、サーバー・インスタンスをローカル・テスト・サーバーとして登録するために必要な各ステップを実行します。統合テスト環境は、同じマシンまたはリモート・マシンにインストールされているサーバー・インスタンスで動作します。

統合テスト環境の始動

統合テスト環境を始動するには、以下の手順を実行します。

1. 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「管理」>「System Manager」を選択します。
2. メニュー・バーから「ウィンドウ」>「パースペクティブを開く」>「その他」を選択します。
3. パースペクティブのリストから「統合テスト環境」を選択して、「OK」をクリックします。

ワークベンチが始動し、表示されます。32 ページの図 2 に、統合テスト環境パースペクティブを示し、『統合テスト環境インターフェース』で、インターフェースおよびそのエレメントについて説明します。

統合テスト環境インターフェース

統合テスト環境パースペクティブには、オープンするデフォルト構成で、いくつかのビューおよび 1 つのエディターがあります。275 ページの図 71 にデフォルトの統合テスト環境パースペクティブを示します。

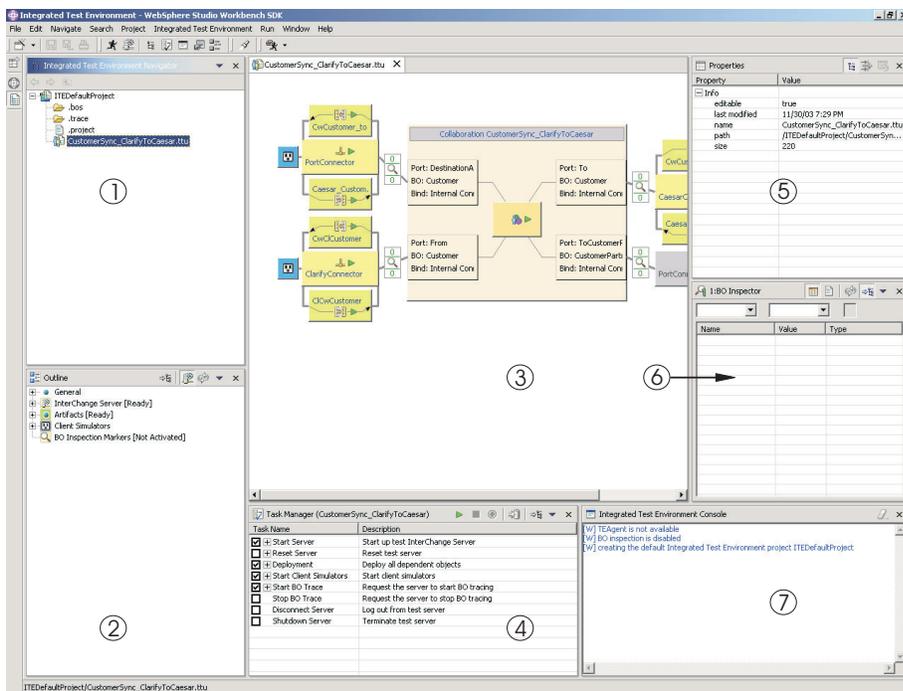


図 71. 統合テスト環境パースペクティブ

表 24 に、統合テスト環境パースペクティブのインターフェース・エレメントを示します。番号は 図 71 と対応しています。

表 24. 統合テスト環境パースペクティブのインターフェース・エレメント

インターフェース・エレメント番号	インターフェース・エレメント名
1	『統合テスト環境ナビゲーター・ビュー』
2	『アウトライン・ビュー』
3	276 ページの『テスト・ユニット・エディター』
4	276 ページの『タスク・マネージャー・ビュー』
5	276 ページの『プロパティ・ビュー』
6	276 ページの『BO インспекター・ビュー』
7	276 ページの『統合テスト環境コンソール・ビュー』

統合テスト環境ナビゲーター・ビュー

このビューでは、作成するプロジェクトとテスト・ユニットをリストします。テスト・プロジェクトとテスト・ユニットの使用の詳細については、280 ページの『テスト・プロジェクトおよびテスト・ユニットの使用』を参照してください。

アウトライン・ビュー

このビューでは、インターフェースを構成するコンポーネントとエレメントをリストし、276 ページの『テスト・ユニット・エディター』で表示されます。このビューで項目を選択すると、そのプロパティが 276 ページの『プロパティ・ビュー』

ー』に表示されます。また、このビューを使用すると、インターフェースの依存関係が解決されているかどうか、およびテストに先立ちその他の必要条件が満たされているかどうかの判断も可能です。

これはデフォルトのワークベンチ・ビューです。

テスト・ユニット・エディター

このエディターでは、インターフェースのグラフィカル表現を示します。これにより、以下のようなことを実行できます。

- コンポーネントの配置。詳細については、305 ページの『テスト・ユニット・ビューを使用したリポジトリの管理』を参照してください。
- コンポーネントの状態の管理。詳細については、305 ページの『テスト・ユニット・ビューを使用したコンポーネント状態の管理』を参照してください。
- ビジネス・オブジェクト・データの表示。詳細については、325 ページの『BO インспекター・ビューの使用』を参照してください。

タスク・マネージャー・ビュー

タスク・マネージャー・ビューにより、InterChange Server の始動および Client Simulator ビューの開始など、テストの一部として実行する必要のあるタスクを実行できます。このビューの詳細については、293 ページの『タスク・マネージャー・ビューの使用』を参照してください。

プロパティ・ビュー

ITE ナビゲーター・ビュー、アウトライン・ビュー、およびテスト・ユニット・ビューで項目を選択して、このビューで項目のプロパティを表示できます。プロパティ情報はユーザーにはあまり必要でないため、このビューは Client Simulator ビューや BO インспекター・ビューとオーバーレイできます。

Client Simulator ビューの詳細については、309 ページの『Client Simulator ビューの使用』を参照してください。

BO インспекター・ビューの詳細については、325 ページの『BO インспекター・ビューの使用』を参照してください。

これはデフォルトのワークベンチ・ビューです。

BO インспекター・ビュー

このビューを使用すると、ビジネス・オブジェクト・データを表示できます。詳細については、325 ページの『BO インспекター・ビューの使用』を参照してください。

統合テスト環境コンソール・ビュー

統合テスト環境コンソールには、テスト・タスクについての情報が表示されます。例えば、InterChange Server インスタンスを始動し、Client Simulator ビューを開始すると、項目が追加されます。この情報を使用して、テスト・プロセス自体をトラブルシューティングします。

統合テスト環境から InterChange Server を始動する場合、「InterChange Server コンソール」という 2 番目のタブがこのビューに追加されます。「InterChange Server コンソール」は、InterChange Server のログおよびトレース情報を表示します。ここに表示される情報を使用して、テストするインターフェースをトラブルシューティングします。

これはデフォルトのワークベンチ・ビューです。

サーバー構成の選択

ローカル・テスト・サーバーとしてすでに複数の InterChange Server インスタンスが登録されている場合は、統合テスト環境で作業するサーバー構成を選択する必要があります。サーバー構成を選択するには、以下の手順を実行します。

注: これらのステップを実行する前に、InterChange Server を System Manager パースペクティブに登録する必要があります。詳細については、35 ページの『InterChange Server インスタンスの登録』を参照してください。

1. 「InterChange Server 構成」ウィンドウを表示するには、以下のいずれかを実行します。
 - 統合テスト環境のメニュー・バーから「**統合テスト環境**」> 「**テスト・サーバー構成**」を選択します。
 - アウトライン・ビューで「InterChange Server」ノードを展開し、「登録」ノードを右マウス・ボタンでクリックして、コンテキスト・メニューから「**テスト・サーバーの構成を表示**」を選択します。
2. 「InterChange Server 構成」ウィンドウで、ウィンドウの左側のペインから使用するテスト・サーバーを選択します。

サーバーを選択すると、その構成情報がウィンドウの右側のペインに表示されます。ウィンドウの一番上の読み取り専用テキスト・フィールドに、サーバーが実行中かどうか、また統合テスト環境がサーバーに接続されているかどうかを示されます。

278 ページの図 72 に、「InterChange Server 構成」ウィンドウを示します。

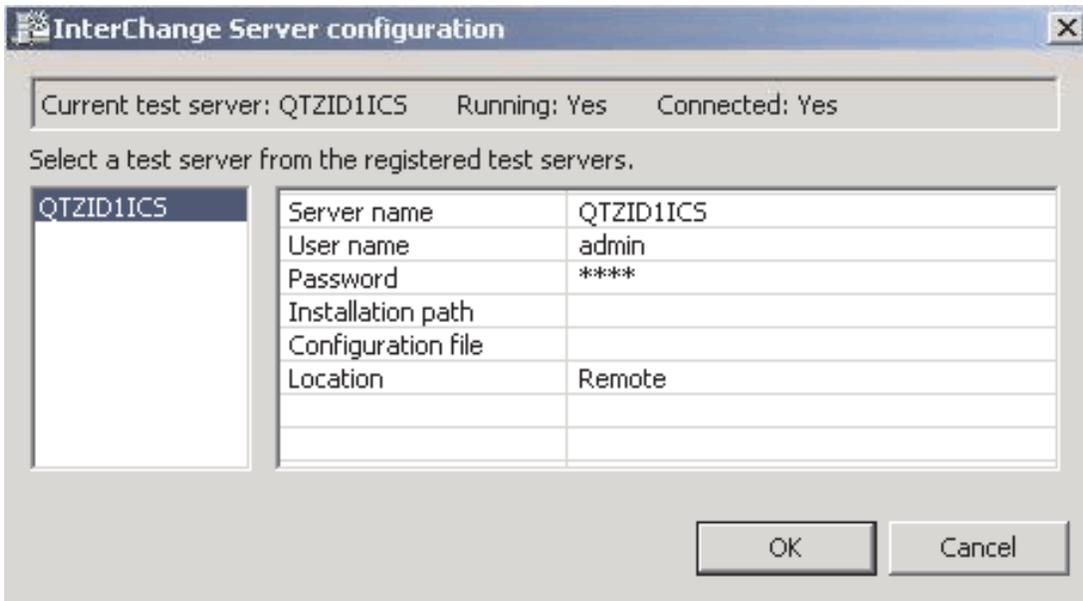


図 72. 「ICS テスト・サーバー構成」ウィンドウ

3. 「OK」をクリックします。

設計モードで始動するための InterChange Server の構成

テストするコンポーネントを統合テスト環境を使用して配置する場合は、InterChange Server インスタンスが設計モードで開始するように `start_server.bat` バッチ・ファイルを変更する必要があります。これは、統合テスト環境では依存関係が直ちに解決されないような順序でコンポーネントが配置されることがあるからです。このような状態で正常に配置するには、サーバーは設計モードで実行する必要があります。

39 ページの『InterChange Server のモード』で説明するように、通常は、コマンド行かショートカットの「ターゲット」フィールドで `-design` オプションを渡して、InterChange Server を設計モードで始動します。統合テスト環境を使用する場合、サーバーを統合テスト環境パースペクティブで始動し、ショートカットを使用しません。したがって InterChange Server ショートカットを `-design` オプションを使用するように変更しても、サーバーは設計モードで始動しません。

統合テスト環境を使用する場合にサーバーを設計モードで始動するには、`-design` オプションを `start_server.bat` バッチ・ファイルに追加するか、統合テスト環境がオプションを自身で追加するための一時バッチ・ファイルを使用するようにします。ご使用の設定に応じて、以下のセクションのいずれかの説明に従ってください。

start_server.bat ファイルの手動による編集

以下の手順を実行して、`start_server.bat` バッチ・ファイルをユーザー自身で編集し、`-design` オプションを追加します。

1. テキスト・エディターを使用して製品インストールの `bin` ディレクトリー内の `start_server.bat` という名前のバッチ・ファイルをオープンします。

2. サーバーを始動するために Java プログラムが実行されるバッチ・ファイル内の行に **-design** オプションを追加します。**-design** オプションを、**-s%SERVERNAME%** オプションの後に追加する必要があります。以下は、バッチ・ファイルの表示例を示します。

```
%CWJAVA% -Djava.ext.dirs=%JRE_EXT_DIRS%;"%MQ_LIB%";"%DB2_LIB%"
-Duser.home="%CROSSWORLDS%" -mx%CW_MEM_HEAP%m -DTEAgent=1200
-DCW_MEMORY_MAX=%CW_MEM_HEAP% %ORB_PROPERTY% -classpath %JCLASSES%
ServerWrapper -s%SERVERNAME% -design %2 %3
```

3. ファイルの保管して閉じます。

統合テスト環境によるカスタム・バッチ・ファイルの作成を有効にする

統合テスト環境を設定して、InterChange Server が自動的に設計モードで始動するようになります。この方法では、統合テスト環境は `start_server.bat` ファイルを読み取り、`-design` オプションが正しく指定されているかどうかを判別します。正しく指定されていない場合、統合テスト環境は `start_server.bat` ファイルのコピーとして `start_server_ite_tmp.bat` というファイルを作成し、`-design` オプションを追加します。統合テスト環境からサーバーを始動する場合に、この一時ファイルが使用されます。

統合テスト環境の設定を行って一時バッチ・ファイルを作成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「**ウィンドウ**」>「**設定**」を選択します。
2. 「**統合テスト環境**」を展開します。
3. 「**テスト・サーバー**」を選択します。
4. 「**サーバー起動スクリプト (start_server.bat) の検証**」チェック・ボックスを有効にすることにより、統合テスト環境が必要な構成情報を追加するためのカスタム・バッチ・ファイルを使用するようにします。

統合テスト環境が作成できるカスタム・バッチ・ファイルを使用しない場合は、「**サーバー起動スクリプト (start_server.bat) の検証**」チェック・ボックスを使用不可のままにします。

5. 「**OK**」をクリックします。

要確認: この方法を機能させるには、サーバーを統合テスト環境から始動する必要があります。プログラム・グループ・ショートカットやコマンド行を使用してサーバーを始動すると、この方法は機能しません。これは、これらの始動方法では一時始動スクリプトが使用されないためです。

テスト・モードでのサーバーの開始

ローカル・サーバーは ITE タスク・マネージャーから起動できます。ローカル・サーバーの始動は、登録パネルから取得したインストール・ディレクトリーから、サーバー始動スクリプトを実行するローカル・プロセスを起動して行います。ローカル・サーバー始動スクリプトを変更して、サーバーがフラグ `-design` および `-test` でテスト・サーバーとして自動的に始動するようになります。リモート・サーバーは、このように始動することはできません。したがって、登録サーバーがリモート

ト・サーバーの場合、サーバー起動オプションは使用不可になります。この場合、サーバーはコマンド行から手動で起動する必要があります。

注: サーバー (ローカルまたはリモート) が外部からコマンド行で起動された場合、ITE はサーバーにその実行モード (テスト・モードか設計モード、または両方のモード) を照会し、モードに応じて ITE 自体を調整してサーバーに対応します。

サーバーのシャットダウンは IDL 呼び出しとして実装されるため、ローカルおよびリモート・サーバーのいずれも ITE からシャットダウンできます。このことから、サーバーが接続およびログインされてから、シャットダウン操作が要求されることとなります。

コンソール出力の作業

サーバーが ITE から起動された場合、サーバーのコンソール出力は、ローカル・サーバーの ITE 内で、サーバー・プロセスから InterChange Server コンソール・ウィンドウにリダイレクトされます。リモート・サーバーおよびローカル・サーバーが ITE 外からコマンド行で開始された場合、このようにコンソール出力を取得することはできません。この問題を解決するため、サーバー・コンソール出力は CORBA 経由でトランスポートされます。ITE はコールバック・ハンドルで明示的にコンソールのリダイレクトを要求し、別の IDL 呼び出してサービスを終了する場合があります。サーバーはすべてのメッセージについて ITE に通知します。

リモート・コンソールのリダイレクトは、コンソール・ログによってパフォーマンスが低下する場合、オンまたはオフにすることができます。コンソール・リダイレクトは、デフォルトではオンになっています。

注: 有線でのコンソール出力は、手動で始動されたサーバーのみを対象にしています (ITE から始動されたサーバーでない)。ITE から起動されたサーバ (ローカル・サーバー) の場合、コンソールはコンソール・ビューアーにリダイレクトされます。

テスト・プロジェクトおよびテスト・ユニットの使用

Eclipse ベース・プラットフォームのすべてのリソースはプロジェクトに含まれます。詳細は、28 ページの『プロジェクト』を参照してください。統合コンポーネントを組み込むには統合コンポーネント・ライブラリー・プロジェクトを作成し、コンポーネントにショートカットを組み込むにはユーザー・プロジェクトを作成し、またテストの定義を組み込むには統合テスト環境に**テスト・プロジェクト**を作成する必要があります。

テスト・ユニットとは、テストを定義するワークベンチに作成されるリソースです。統合テスト環境はインターフェース全体をテストできるように設計され、インターフェースは通常、1 ページの『インターフェース』で説明するように、コラボレーション・オブジェクトを中心とし、したがってテスト・ユニットもコラボレーション・オブジェクトが中心となります。単一のテスト・プロジェクトで実行するすべてのテスト・ユニットを定義するか、または複数のテスト・プロジェクトを作成して、設定に従って編成するのに適切なテスト・ユニットを追加することができます。

テスト・プロジェクト設定の構成

テスト・プロジェクトまたはテスト・ユニットを作成する前に、それらに関連する設定を行います。テスト・プロジェクトおよびテスト・ユニットに関連する設定を行うには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「統合テスト環境」を選択します。

図 73 に統合テスト環境の設定を示します。

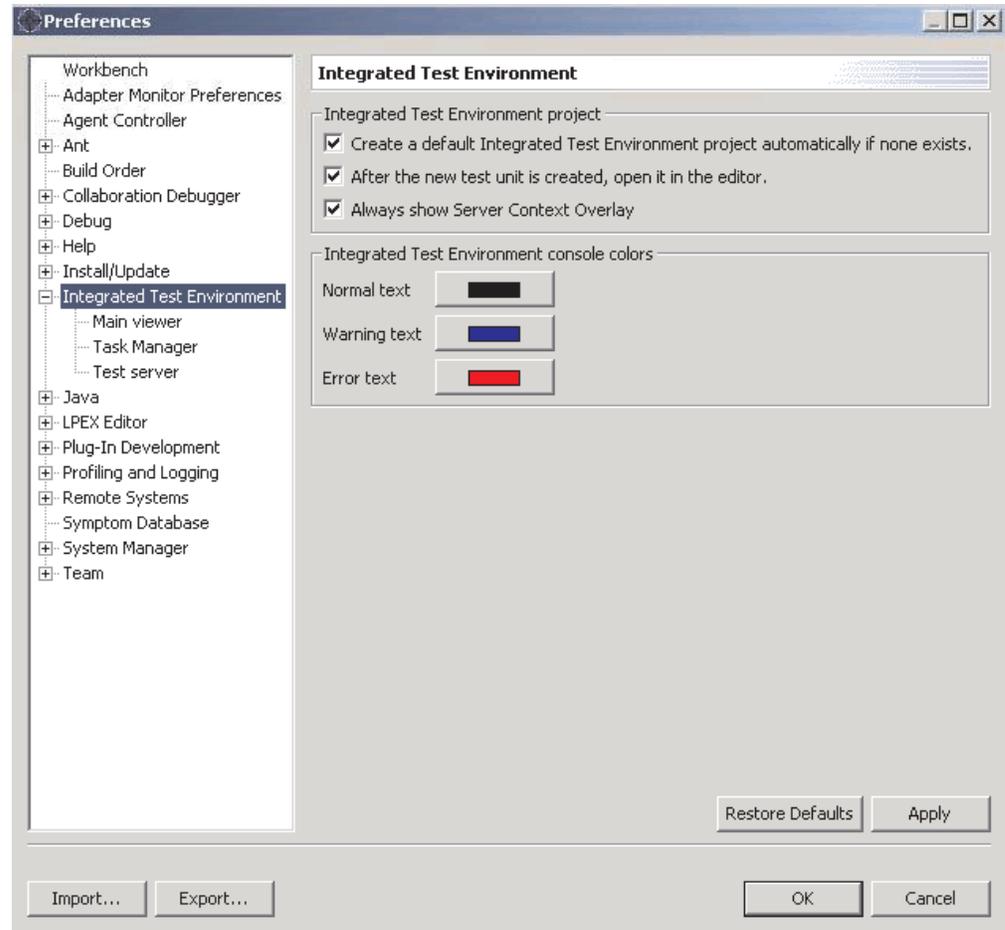


図 73. 統合テスト環境の設定

3. テスト・ユニットの作成時、既存プロジェクトが存在しない場合に統合テスト環境が DefaultITEProject という名前のデフォルトのプロジェクトを作成するように設定する場合は、「何も存在しない場合は自動的にデフォルト ITE プロジェクトを作成」チェック・ボックスを有効にします。
4. テスト・ユニット定義を作成後にテスト・ユニット・エディターで自動的に開く場合は、「作成時にエディターで新規テスト・ユニットを開く」チェック・ボックスを有効にします。

このオプションを有効にしない場合のテスト・ユニットのオープンについて詳しくは、285 ページの『テスト・ユニットのオープン』を参照してください。

5. サーバー・コンテキスト・オーバーレイを常時有効にしておく場合は、「サーバー・コンテキスト・オーバーレイを常に表示」チェック・ボックスを有効にします。

サーバー・コンテキスト・オーバーレイについて詳しくは、303 ページの『サーバー・コンテキスト・オーバーレイの使用』を参照してください。

6. 「OK」をクリックします。

テスト・プロジェクトの作成

テスト・プロジェクトを作成し、作成する個々のテスト・ユニットを保管するには、以下の手順を実行します。

1. メニュー・バーから「ファイル」>「新規」>「統合テスト環境プロジェクト」を選択します。
2. 「新規統合テスト環境プロジェクト」画面の「プロジェクト名」フィールドにテスト・プロジェクトの名前を入力します。

プロジェクト名には英数字および下線のみが使用でき、英語で指定します。

3. デフォルトのロケーション (ワークスペース) にライブラリー用フォルダーを作成し、ライブラリーに指定した名前と同じ名前を付けるには、「デフォルト・ロケーションの使用」チェック・ボックスを有効にしておきます。

ライブラリー・フォルダーの名前およびロケーションを指定する場合は、以下の手順を実行します。

- a. 「デフォルト・ロケーションの使用」チェック・ボックスをクリアします。
- b. ライブラリーに使用するディレクトリーの絶対パスおよび名前を「ロケーション」フィールドに入力するか、「ブラウズ」をクリックして既存のディレクトリーを選択します。

注: 「デフォルト・ロケーションの使用」チェック・ボックスを使用して System Manager で作成する以外に、ワークスペースのパスにライブラリー用フォルダーを作成する方法はありません。

283 ページの図 74 に、「新規統合テスト環境プロジェクト」ウィザードを示します。

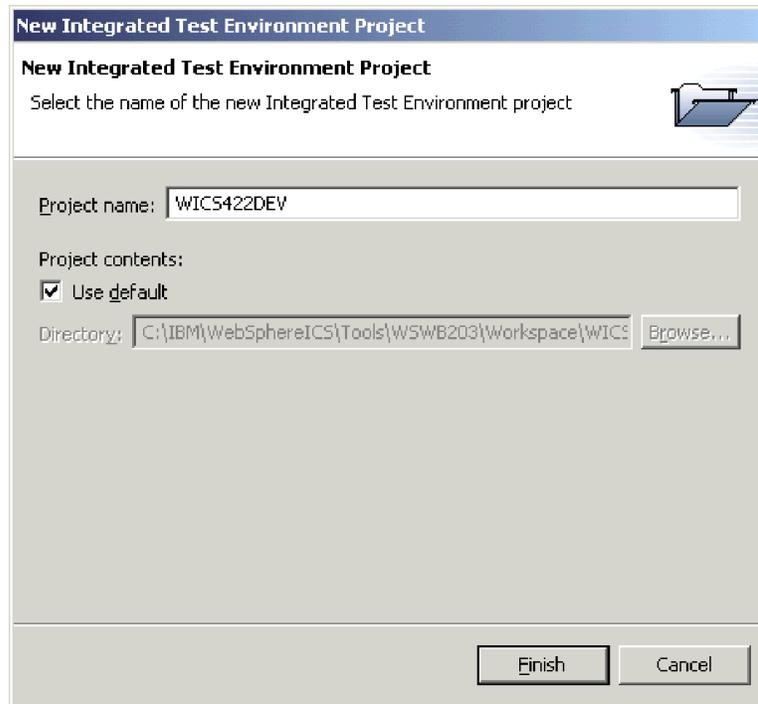


図 74. 新規 WBI ITE プロジェクトの作成

4. 「完了」をクリックします。

プロジェクトが作成され、フォルダーが「ITE ナビゲーター」ビューに追加されます。

テスト・ユニットの作成

テスト・ユニットとは、実行するテストの構成情報を含むワークベンチ・リソースです。テスト・ユニットは、統合テスト環境内かまたは System Manager 内で作成できます。

統合テスト環境設定で「作成時にエディターで新規テスト・ユニットを開く」チェック・ボックスを有効にした場合、テスト・ユニットを作成後にその単位が開きます。この設定を有効にしていない場合、285 ページの『テスト・ユニットのオープン』の説明に従ってテスト・ユニットを開く必要があります。統合テスト環境の設定について詳しくは、281 ページの『テスト・プロジェクト設定の構成』を参照してください。

統合テスト環境でのテスト・ユニットの作成

統合テスト環境内でテスト・ユニットを作成するには、以下の手順を実行します。

1. メニュー・バーから「ファイル」>「新規」>「ITE テスト・ユニット」を選択します。
2. 「コラボレーションの選択」画面で、システムに定義されているすべての統合コンポーネント・ライブラリー内のすべてのコラボレーション・オブジェクトのリストから、テストするコラボレーション・オブジェクトを選択します。

図 75 に、「コラボレーションの選択」画面を示します。

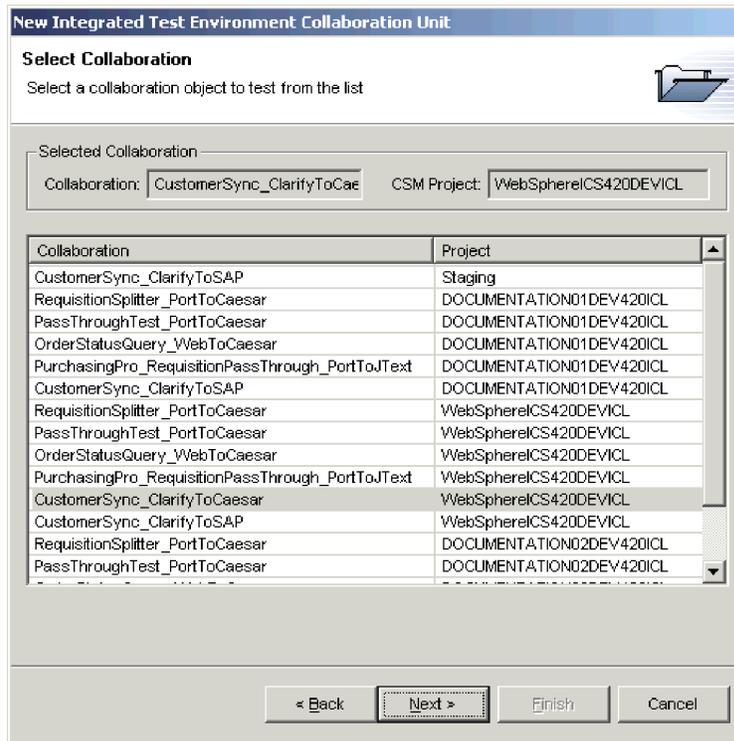


図 75. テストするコラボレーションの選択

3. 「次へ」をクリックします。
4. 「ITE テスト・ユニットの作成」画面で、以下の手順を実行します。
 - a. 「テスト・ユニット」フィールドに、テスト・ユニットの名前を入力します。
 - b. 「ITE プロジェクト」ドロップダウン・メニューから、テスト・ユニットを作成するテスト・プロジェクトを選択します。

285 ページの図 76 に、「ITE テスト・ユニットの作成」画面を示します。

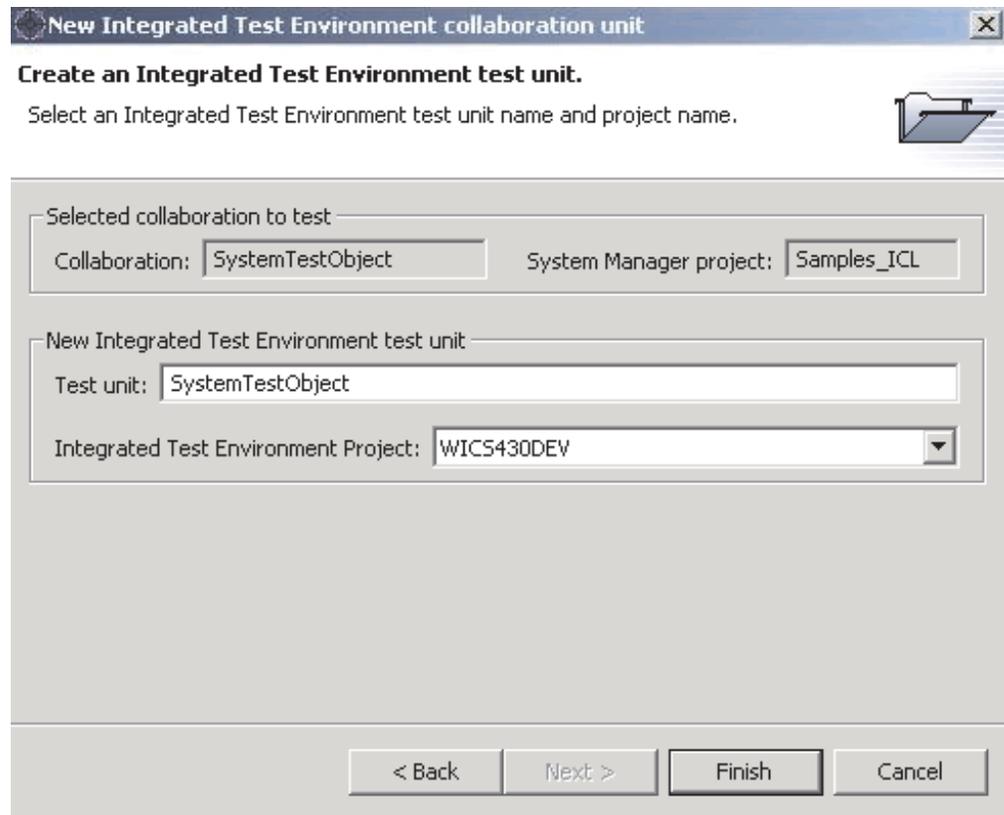


図 76. テスト・ユニット名とテスト・プロジェクトの指定

5. 「完了」をクリックします。

System Manager 内でのテスト・ユニットの作成

System Manager 内でテスト・ユニットを作成するには、以下の手順を実行します。

1. テスト・ユニットを含むテスト・プロジェクトを作成します。詳細は 282 ページの『テスト・プロジェクトの作成』を参照してください。
2. インターフェースを表すコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「統合テスト環境でのデバッグ」を選択します。
3. 「ITE テスト・ユニットの作成」画面で、「テスト・ユニット」フィールドにテスト・ユニットの名前を入力し、「ITE プロジェクト」ドロップダウン・メニューからテスト・ユニットの作成先となるテスト・プロジェクトを選択します。

図 76 に、「ITE テスト・ユニットの作成」画面を示します。

4. 「完了」をクリックします。

テスト・ユニットのオープン

テスト・ユニットを作成したら、それをオープンしてそのレイアウトを表示し、構成できます。テスト・ユニットをオープンするには、以下の手順を実行します。

1. 「ITE ナビゲーター」ビューでテスト・ユニットを含むテスト・プロジェクトを展開します。
2. テスト・ユニットをオープンするには、以下のいずれかを実行します。

- 使用するテスト・ユニットを右マウス・ボタン・クリックし、コンテキスト・メニューから「開く」を選択します。
- 使用するテスト・ユニットをダブルクリックします。

テスト・ユニットが開き、テスト・ユニット・エディターにコラボレーション・オブジェクトが表示され、パースペクティブのビューにインターフェース固有の情報が取り込まれます。275 ページの図 71 に、テスト・ユニットをオープンした後の統合テスト環境パースペクティブを表示します。

アウトライン・ビューの使用

アウトライン・ビューを使用すると、以下のセクションで説明するさまざまな操作を実行できます。

- 287 ページの『アウトライン・ビューを使用した依存の処理』
- 289 ページの『アウトライン・ビューを使用したりポジトリの管理』
- 289 ページの『アウトライン・ビューを使用したコンポーネント状態の管理』
- 290 ページの『アウトライン・ビューを使用したタスクの実行』
- 290 ページの『アウトライン・ビューを使用したテスト作動可能の検証』
- 291 ページの『アウトライン・ビューを使用した Client Simulator ビューの表示およびクローズ』
- 292 ページの『アウトライン・ビューを使用した BO インスペクター・ビューの表示およびクローズ』
- 292 ページの『アウトライン・ビューを使用した BO マーカーの最新表示』

図 77 に、アウトライン・ビューを示します。

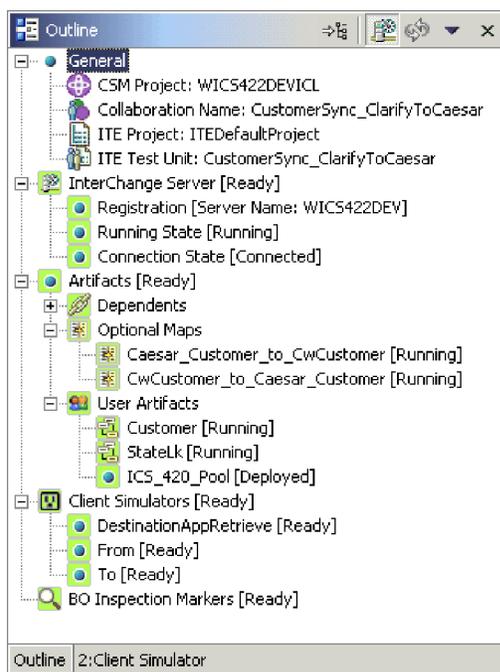


図 77. 「ユーザーの成果物」が表示されたアウトライン・ビュー

アウトライン・ビューを使用した依存の処理

94 ページの『依存関係および参照』で説明するように、IBM WebSphere InterChange Server コンポーネントは、適切に機能するために他のコンポーネントに依存しています。インターフェースをテストするには、すべての依存関係を解決する必要があります。

統合テスト環境のアウトライン・ビューには、インターフェースをテストするために解決する必要のあるコンポーネントの依存関係がリストされます。コンポーネント間の複数の依存関係をシステムで判別して、それらをアウトライン・ビューの「成果物」ノードの下にある「**従属オブジェクト**」ノードにリストすることができます。

96 ページの『システムで検出できない依存関係』で、一部のコンポーネントが自動的に検出できない状態について説明します。インターフェースをテストするときに統合テスト環境のビューを使用してコンポーネントを配置する場合は、システムで検出できない依存関係も解決する必要があります。このためには、ユーザー自身が、検出できない依存関係を指定します。これを行うと、アウトライン・ビューの「**成果物**」ノードの下にある「**ユーザーの成果物**」ノードの下にコンポーネントがリストされます。

ユーザー従属オブジェクトの追加

ユーザー従属としてコンポーネントをテスト・ユニットに追加するには、以下のいずれかを実行します。

1. アウトライン・ビューで、「**成果物**」ノードを展開します。
2. 「**ユーザーの成果物**」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ユーザーの成果物を追加**」を選択します。
3. 「**ユーザーの従属オブジェクト**」ウィンドウで、追加するコンポーネントを選択します。**Shift** を押しながらの連続行の選択や **Ctrl** を押しながらの非連続行の選択などの、標準の複数選択手法を使用できます。288 ページの図 78 に、「ユーザーの従属オブジェクト」ダイアログを示します。

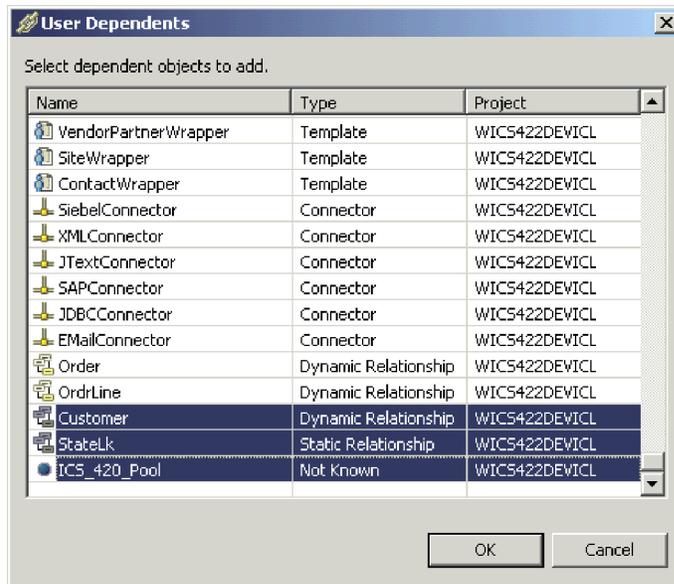


図 78. ユーザー従属オブジェクトの追加

4. 「OK」をクリックします。

注: 従属として未登録のコンポーネントのみがリストに表示されます。

ユーザーの従属オブジェクトの削除

アウトライン・ビューで「成果物」ノードを展開し、以下のいずれかを実行して、ユーザー成果物のリストからコンポーネントを削除します。

- 「ユーザーの成果物」ノードを右マウス・ボタンでクリックし、「すべてのユーザーの成果物を削除」を選択して、リストからユーザー成果物をすべて削除します。
- 「ユーザーの成果物」ノードを展開して削除するユーザー成果物を選択し、選択した成果物のいずれかを右マウス・ボタンでクリックして、コンテキスト・メニューから「削除」を選択します。

例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のよう、Windows の標準的な選択方法を使用できます。

- **Shift** を押したまま、連続する項目を選択します。
- **Ctrl** を押したまま、連続しない項目を選択します。

ユーザーの従属オブジェクトの保管

テスト・ユニットに追加するユーザーの従属オブジェクトを保管して、毎回追加する必要がないようにできます。ユーザーの従属オブジェクトを保管するには、以下のいずれかを実行します。

1. アウトライン・ビューで「成果物」ノードを展開します。
2. 「ユーザーの成果物」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「ユーザーの従属オブジェクト保管」を選択します。

アウトライン・ビューを使用したリポジトリの管理

インターフェースをテストするには、インターフェースに必要なすべてのコンポーネントをサーバー・リポジトリに配置する必要があります。アウトライン・ビューを使用すると、InterChange Server リポジトリと連動して、必要なコンポーネントが含まれていることを確認できます。

コンポーネントを配置する場合は、コンポーネントを配置する「**配置**」オプション、またはコンポーネントがすでにサーバー・リポジトリに存在する場合でもコンポーネントを配置する「**上書きして配置**」オプションのいずれかを使用します。選択したコンポーネントがすでにサーバー・リポジトリに存在する場合に「**配置**」オプションを使用すると、配置に失敗します。

以下のいずれかを実行し、コンテキスト・メニューから「**配置**」または「**上書きして配置**」のいずれかを選択して、目的のコンポーネントを配置します。

- 「成果物」ノードを右マウス・ボタンでクリックし、すべてのコンポーネントを配置します。
- 「従属オブジェクト」ノード、「オプションのマップ」ノード、または「ユーザーの成果物」ノードを右マウス・ボタンでクリックし、これらのノードの下にリストされているすべてのコンポーネントを配置します。
- 個々のコンポーネントを右マウス・ボタンでクリックします。

アウトライン・ビューを使用してサーバー・リポジトリからコンポーネントを削除するには、ビュー内のコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サーバーから除去**」を選択します。

注: 複数のコンポーネントを一度に削除するには、以下のような Windows の標準的な選択方法を使用できます。

- **Shift** を押したまま、連続する項目を選択します。
- **Ctrl** を押したまま、連続しない項目を選択します。

統合テスト環境コンソールおよび InterChange Server コンソールに項目が追加され、配置が成功したかどうかを示されます。

アウトライン・ビューを使用したコンポーネント状態の管理

統合テスト環境内のコンポーネントの状態を変更できます。これによりテスト中に発見した問題を解決できます。つまり、System Manager、repos_copy、または System Monitor を使用しないで、コンポーネントを非活動化し、それを変更し、再配置して、再活動化できます。

コンポーネントの状態を変更するには、アウトライン・ビュー内のコンポーネントを右マウス・ボタンでクリックして、目的の状態操作（「**開始**」、「**一時停止**」、または「**停止**」）を選択します。すべてのコンポーネント・タイプにすべての状態操作が用意されているとは限りません。

コンポーネント状態の詳細については、「システム管理ガイド」を参照してください。

アウトライン・ビューを使用したタスクの実行

アウトライン・ビューを使用して、テスト実行の一部である幾つかのタスクを実行できます。タスクおよびタスク・グループについては、294 ページの表 25 を参照してください。

「サーバーを始動」タスク・グループ内のタスクをすべて実行するには、アウトライン・ビューの「InterChange Server」ノードまたは「実行状態」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サーバーを始動**」を選択します。

「サーバーをシャットダウン」タスクを実行するには、アウトライン・ビューの「InterChange Server」ノードまたは「実行状態」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サーバーを停止**」を選択します。

「サーバーへ接続」タスクを実行するには、アウトライン・ビューの「接続状態」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サーバーへ接続**」を選択します。

「サーバーの切断」タスクを実行するには、アウトライン・ビューの「接続状態」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**サーバーの切断**」を選択します。

「クライアント・シミュレーターを開始」タスク・グループ内のタスクをすべて実行するには、アウトライン・ビューの「クライアント・シミュレーター」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**クライアント・シミュレーターを開始**」を選択します。

「BO トレースの開始」タスク・グループ内のタスクをすべて実行するには、アウトライン・ビューの「BO 検査マーカ―」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**BO トレースの開始**」を選択します。

「BO トレースの停止」タスクを実行するには、アウトライン・ビューの「BO 検査マーカ―」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**BO トレースの停止**」を選択します。

アウトライン・ビューを使用したテスト作動可能の検証

アウトライン・ビューには、テストを実行するために満たす必要のある幾つかの条件の状態がビジュアル標識として示されます。これにより、ログ・テキストで標識を検索しなくても、統合テスト環境でテストを実行する準備ができたかどうかを迅速に確認できます。満たす必要のある各条件の背景は、すでに条件が満たされている場合は緑色に、まだ条件が満たされていない場合は赤色になります。各条件には、状況を示す句も表示されます。以下に、満たす必要のある条件を持つ各種のノードと、要件の説明および条件が満たされた場合に示される句を示します。

- 「InterChange Server」ノード。インターフェースをテストするには、「作動可能」状態である必要があります。このノードは、サーバーの状態、およびそのサーバーに接続する統合テスト環境の状態を表し、以下のノードを含みます。
 - 「登録」ノード。通信対象の統合テスト環境が構成された InterChange Server インスタンスがリストされます。

使用する InterChange Server インスタンスの指定について詳しくは、277 ページの『サーバー構成の選択』を参照してください。

- 「実行状態」ノード。InterChange Server インスタンスが実行中であるかどうかを示します。テストを実行するには、「作動可能」状態である必要があります。
- 「接続状態」ノード。統合テスト環境が InterChange Server インスタンスに接続されているかどうかを示します。テストを実行するには、「接続しました」状態である必要があります。
- 「成果物」ノード。インターフェースをテストするには、「作動可能」状態である必要があります。このノードが「作動可能」状態の場合は、このノードの下にある各ノードのすべてのコンポーネントがサーバーに配置され、実行中である必要があります。

コネクタ、コラボレーション・オブジェクトなど、状態を持つコンポーネントは、「実行中」状態である必要があります。ビジネス・オブジェクト定義、コラボレーション・テンプレートなど、状態を持たないがインターフェースに必要なコンポーネントは、「配置が完了」状態である必要があります。

- 「クライアント・シミュレーター」ノード。インターフェースをテストするには、「作動可能」状態である必要があります。このノードが「作動可能」状態の場合は、このノードの下に表示されるすべてのポートが関連した Client Simulator ビューを持ち、サーバーに接続されている必要があります。
- 「BO 検査マーカー」ノード。インターフェースのテスト中に処理するビジネス・オブジェクト・トレースに対して「作動可能」状態である必要があります。このノードが「作動可能」状態の場合は、「BO トレースの開始」タスクが正常に開始されている必要があります。

アウトライン・ビューを使用した Client Simulator ビューの表示およびクローズ

Client Simulator ビューを使用すると、インターフェースをテストするときに、コネクタとアクセス・クライアントをエミュレートできます。Client Simulator ビューには、テスト・データを持つビジネス・オブジェクト・インスタンスを作成したり、そのインスタンスをビジネス・オブジェクト要求として送信したり、ビジネス・オブジェクト応答としてその要求に応答するためのインターフェースが用意されています。

コネクタのプロファイルを特定の Client Simulator ビューにロードするには、「クライアント・シミュレーター」ノードの下にある目的のポートを右マウス・ボタンでクリックし、「クライアント・シミュレーター *x* を使用」を選択します。ここで、*x* は、そのコネクタに関して使用する Client Simulator ビュー位置の番号です。

310 ページの表 29 に、各種 Client Simulator ビューが表示されるデフォルトのインターフェース位置を示します。

統合テスト環境内のビューは、テストの各段階で、ほかのビューを使用することにより隠すことができます。例えば、BO インспекター・ビューを使用して Client Simulator ビューを隠すことができます。隠した Client Simulator ビューを表示する

には、「クライアント・シミュレーター」ノードの下にある目的のポートを右マウス・ボタンでクリックし、「クライアント・シミュレーターを表示」を選択して、Client Simulator ビューにフォーカスを与えます。

Client Simulator ビューを閉じるには、「クライアント・シミュレーター」ノードの下にある目的のポートを右マウス・ボタンでクリックし、「クライアント・シミュレーターを閉じる」を選択します。

注: このオプションは Client Simulator ビューを閉じるだけでなく、Client Simulator ビューでのコネクタのエミュレーションを停止させます。

Client Simulator ビューの使用法について詳しくは、309 ページの『Client Simulator ビューの使用』を参照してください。

アウトライン・ビューを使用した BO インспекター・ビューの表示およびクローズ

ビジネス・オブジェクト・トレースを使用する場合、統合テスト環境により、インターフェースの実行中の特定時点でビジネス・オブジェクト・データが記録されます。

すべてのビジネス・オブジェクト・マーカーの BO インспекター・ビューをインターフェースに表示するには、アウトライン・ビューの「BO 検査マーカー」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべての BO ビューアーを表示」を選択します。BO マーカー・アイコンは、304 ページの図 81 に示してあります。

インターフェースに表示されたすべての BO インспекター・ビューを閉じるには、アウトライン・ビューの「BO 検査マーカー」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「すべての BO ビューアーを閉じる」を選択します。

327 ページの表 31 に、各種 BO インспекター・ビューが表示されるデフォルトのインターフェース位置を示します。

ビジネス・オブジェクト・トレースの詳細については、294 ページの表 25 を参照してください。

BO インспекター・ビューの使用法の詳細については、325 ページの『BO インспекター・ビューの使用』を参照してください。

アウトライン・ビューを使用した BO マーカーの最新表示

ビジネス・オブジェクト・トレースを使用しているが、トレース情報の自動更新が使用不可の場合は、BO マーカーを明示的に最新表示して更新する必要があります。

BO マーカーを更新するには、アウトライン・ビューで「BO 検査マーカー」ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「BO マーカーを最新表示」を選択します。

タスク・マネージャー・ビューの使用

タスク・マネージャー・ビューにより、インターフェースをテストするときに実行する必要のある多くのタスクを自動化できます。

テストを実行するときには、一般的に特定の段階で複数のタスクを実行する必要があります。例えば、テストの実行を準備する場合、InterChange Server を始動し、それに統合テスト環境パースペクティブを接続する必要があります。テストの後の段階でインターフェース内のコネクタとアクセス・クライアントに関する Client Simulator ビューを起動し、それらを InterChange Server インスタンスに接続する必要があります。テストの実行中に特定の段階に関連することのあるすべてのタスクを容易に実行するために、タスク・マネージャーにより関連タスクがグループで表示されます。例えば、Client Simulator ビューを起動して InterChange Server インスタンスに接続するタスクは、「クライアント・シミュレーターを開始」という名前のタスク・グループに集められます。タスク・マネージャーにより、単一タスクまたはタスク・グループ全体として実行することができます。

図 79 に、複数のタスクとタスク・グループが選択されたタスク・マネージャー・ビューを示します。

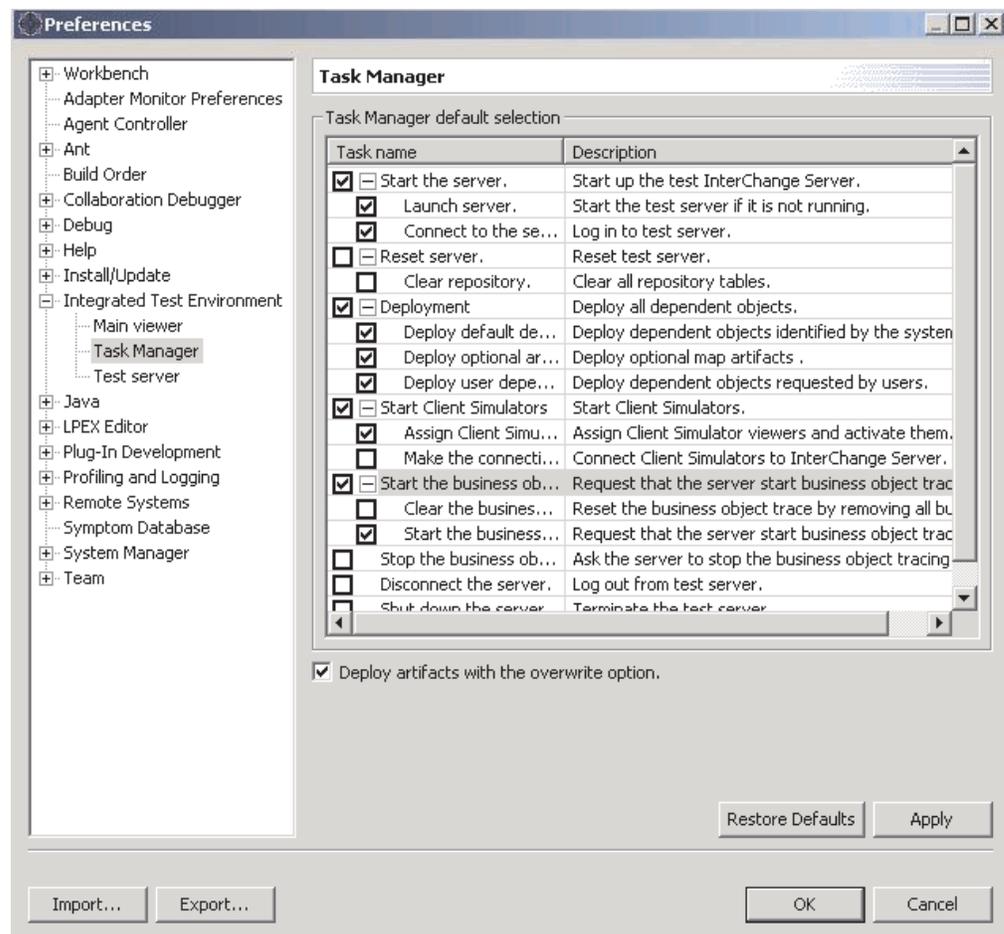


図 79. タスク・マネージャー・ビュー

表 25 に、タスク・マネージャー・ビューのタスクおよびタスク・グループを示します。

表 25. タスク・マネージャー・ビュー内のタスクおよびタスク・グループ

タスク名	タスクの説明
サーバーを始動	InterChange Server の始動に関連するすべてのタスクのグループです。
サーバーを起動	InterChange Server インスタンスを始動します。このタスクは、 サーバーを始動タスク・グループ に含まれます。
サーバーに接続	統合テスト環境を InterChange Server インスタンスに接続することにより、コンポーネントの管理と配置、およびデータの表示を可能にします。このタスクは、 サーバーを始動タスク・グループ に含まれます。 注: 統合テスト環境がサーバーに接続するには、IBM Java オブジェクト・リクエスト・ブローカー (ORB) が実行されている必要があります。
サーバーのリセット	リポジトリをクリア ・タスクのグループです。
リポジトリをクリア	InterChange Server インスタンスのリポジトリを削除します。統合テスト環境を使用して多数のコンポーネントを配置する際、コンポーネントの一部がすでにリポジトリに存在する場合は、このタスクを実行します。このタスクは、 サーバーのリセット・タスク・グループ に含まれます。
配置	タスク・マネージャー・ビューを使用してコンポーネントを InterChange Server リポジトリに配置するのに関連するすべてのタスクのグループです。 インターフェースをテストするために、インターフェースのすべての従属コンポーネントを配置する必要があります。 統合テスト環境を使用してコンポーネントを配置するには、InterChange Server インスタンスを設計モードで始動する必要があります。詳細については、278 ページの『設計モードで始動するための InterChange Server の構成』を参照してください。 依存関係ビューの詳細については、286 ページの『アウトライン・ビューの使用』を参照してください。
デフォルトの従属オブジェクトを配置	インターフェースが機能するために必要であるとシステムで判別されたすべてのコンポーネントと、依存関係ビューの デフォルトの従属オブジェクト・ノード にリストされているすべてのコンポーネントを配置します。 このタスクは、 配置タスク・グループ に含まれます。
オプションの成果物を配置	
ユーザーの従属オブジェクトを配置	アウトライン・ビューの「 ユーザーの成果物 」ノードに追加するすべてのコンポーネントを配置します。 このタスクは、 配置タスク・グループ に含まれます。

表 25. タスク・マネージャー・ビュー内のタスクおよびタスク・グループ (続き)

タスク名	タスクの説明
クライアント・シミュレーターを開始	コネクター・エージェントおよびアクセス・クライアントのエミュレートに関連するすべてのタスクのグループです。
クライアント・シミュレーター・ビューアーの割り当て (Assign Client Simulator Viewers)	<p>インターフェースに関係するすべてのコネクターおよびアクセス・クライアントの Client Simulator ビュー・ウィンドウを開始します。</p> <p>このタスクは、クライアント・シミュレーターを開始タスク・グループに含まれます。</p>
サーバー接続	<p>Client Simulator ビューを InterChange Server インスタンスに接続します。</p> <p>ビジネス・オブジェクトを交換するには、コネクター・エージェントとアクセス・クライアントを InterChange Server に接続する必要があります。</p> <p>デフォルトでは、コネクターはリポジトリ内にある自身の定義を使用して始動します。311 ページの『リポジトリ定義を使用したコネクターのエミュレート』で説明するように、Client Simulator ビューを使用してエージェントを構成ファイルと関連付けることができます。</p> <p>エミュレート対象のコネクターの DeliveryTransport プロパティーが、値 IDL に設定されている場合、Client Simulators はこの方法でしか自動的にサーバーへ接続できません。</p> <p>このタスクは、クライアント・シミュレーターを開始タスク・グループに含まれます。</p> <p>注: 統合テスト環境がサーバーに接続するには、IBM ORB が実行されている必要があります。</p>
BO トレースの開始	<p>ビジネス・オブジェクト・トレースに関連するすべてのタスクのグループです。</p> <p>ビジネス・オブジェクト・トレースでは、ビジネス・オブジェクトがシステムのコンポーネントで処理されると、ビジネス・オブジェクト・データを記録します。詳細については、325 ページの『BO インспекター・ビューの使用』を参照してください。</p>

表 25. タスク・マネージャー・ビュー内のタスクおよびタスク・グループ (続き)

タスク名	タスクの説明
BO トレース・フォルダーをクリア	<p>ビジネス・オブジェクト・トレース・データが保管されるフォルダーをクリアします。次の理由で、ビジネス・オブジェクト・トレース・フォルダーをクリアする必要があります。</p> <ul style="list-style-type: none"> • ビジネス・オブジェクト・トレース・ファイルは、統合テスト環境パースペクティブで表示するためにロードするには、ディスク・ストレージを必要としリソースを使用します。ストレージとリソースを保持するためにフォルダーをクリアする必要がある場合があります。 • 古いテスト・データを統合テスト環境で表示しないようにできます。以前のテストのビジネス・オブジェクト・テスト・データをクリアして、その後のテストのデータのみを表示するようにできます。 <p>このタスクは、BO トレースの開始タスク・グループに含まれます。</p>
BO トレースの開始	<p>ビジネス・オブジェクト・トレースを開始します。テストの実行中にビジネス・オブジェクト・データを保管するために、統合テスト環境のビジネス・オブジェクト・トレースを開始する必要があります。</p> <p>このタスクは、BO トレースの開始タスク・グループに含まれます。</p>
BO トレースの停止	<p>ビジネス・オブジェクト・トレースを停止します。</p> <p>ビジネス・オブジェクト・トレースは大量のリソースを使用するタスクであるため、テストの目的によっては、かえって問題となる場合があります。このような場合は、ビジネス・オブジェクト・トレースを停止できます。</p>
サーバーの切断	<p>統合テスト環境を InterChange Server インスタンスから切断します。ただし、InterChange Server インスタンスはシャットダウンしません。</p>
サーバーをシャットダウン	<p>InterChange Server インスタンスをシャットダウンします。</p>

タスクの選択

タスクやタスク・グループを選択するには、いくつかの方法があります。表 26 に、各種方法を示します。最も効率的にするために各種の方法を組み合わせることをお勧めします。

表 26. タスク・マネージャー・ビューでタスクおよびタスク・グループを選択する方法

実現する内容	方法
グループ内の単一タスクにチェックマークを付ける	タスクの横のチェック・ボックスをクリックします。
タスク・グループにチェックマークを付ける	タスク・グループの横のチェック・ボックスをクリックします。

表 26. タスク・マネージャー・ビューでタスクおよびタスク・グループを選択する方法 (続き)

実現する内容	方法
ビュー内のすべてのタスクにチェックマークを付ける	タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「 すべてにチェックマークを付ける 」を選択します。
ビュー内のすべてのタスクのチェックマークを解除する	タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「 すべてにチェックマークを外す 」を選択します。
設定タスクにチェックマークを付ける	タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「 設定の選択内容 」を選択します。 詳細については、300 ページの『設定タスク選択の構成』を参照してください。
デフォルトのタスクにチェックマークを付ける	タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「 デフォルトの選択内容 」を選択します。 タスクおよびタスク・グループのデフォルトの選択は、以下のとおりです。 <ul style="list-style-type: none"> • サーバーを始動 <ul style="list-style-type: none"> - サーバーを起動 - サーバー初期化待ち - サーバーに接続 • 配置 <ul style="list-style-type: none"> - デフォルトの従属オブジェクトを配置 - オプションの成果物を配置 - ユーザーの従属オブジェクトを配置 • クライアント・シミュレーターを開始 <ul style="list-style-type: none"> - クライアント・シミュレーター・ビューアーの割り当て • BO トレースの開始 <ul style="list-style-type: none"> - BO トレースの開始

タスク・マネージャー・ビューの実行

タスク・マネージャー・ビューで選択したタスクを実行するには、いくつかの方法があります。次のセクションで、各種方法を説明します。

- 『選択したタスク・グループの実行』
- 298 ページの『タスク・グループの実行』
- 298 ページの『単一タスクの実行』

選択したタスク・グループの実行

複数のタスク・グループ内のすべてのタスクを一度に実行できます。これにより、さまざまな段階で手動で管理しないで、テストのためのインターフェースを容易に準備できます。

例えば、便利な方法は、**サーバーを始動**、**クライアント・シミュレーターを開始**、および **BO トレースの開始**の各タスク・グループ内のすべてのタスクを実行する方法です。

選択したタスク・グループ内のすべてのタスクを実行するには、以下の手順を実行します。

1. 実行するタスクのチェック・ボックスを使用可能にします。
2. 実行するタスクが所属するタスク・グループのチェック・ボックスを使用可能にします。
3. 選択したタスク・グループ内の選択したタスクを実行するには、以下のいずれかを実行します。
 - メニュー・バーから「**統合テスト環境**」>「**実行**」を選択します。
 - タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**実行**」を選択します。
 - タスク・マネージャー・ビューのタイトル・バーの「**タスクの実行**」ボタンをクリックします。

矛盾する結果となるタスクまたはタスク・グループのチェック・ボックスを使用可能にしないでください。例えば、**サーバーを起動**と**サーバーをシャットダウン**・タスクの両方を使用可能にしないでください。

タスク・グループの実行

単一のタスク・グループを実行するには、タスク・グループを右マウス・ボタン・クリックして「**実行**」を選択します。右マウス・ボタン・クリックしたタスク・グループ内のタスクのみが実行されます。タスクを実行するにはタスク・グループのチェック・ボックスは使用可能である必要はありません。しかしグループ内の実行するタスクのチェック・ボックスは使用可能になっている必要があります。

クライアント・シミュレーターを開始タスク・グループでは、この方法を頻繁に使用することがあります。テストを最初に準備するためにタスク・グループを頻繁に実行します。詳細は 297 ページの『**選択したタスク・グループの実行**』を参照してください。しかし、一連のテストの過程でオープンした Client Simulator ビューをクローズし、その後起動して再度接続する必要がある場合があります。このとき、すでに実行したすべてのタスクを実行する必要はありません。**クライアント・シミュレーターを開始**タスク・グループを実行するだけで構いません。

単一タスクの実行

単一タスクを実行するには、タスクを右マウス・ボタン・クリックして「**実行**」を選択します。右マウス・ボタン・クリックしたタスクのみが実行されます。タスクを実行するのに、タスクのチェック・ボックスは使用可能である必要はありません。

この方法はサーバーのシャットダウンに使用できます。直後にサーバーを再始動するが、必要なチェック・ボックスをすべて再度使用可能にしたい場合があります。この場合は、**サーバーをシャットダウン**・タスクを右マウス・ボタン・クリックして、シャットダウンのみ実行します。

タスクの停止

現在実行中のタスクのキューの停止をタスク・マネージャーに指示できます。これにより現在実行中のどのタスクでも終了できますが、その後のタスクは実行されません。例えば、**サーバーを起動**タスクが実行中にタスクの停止を選択すると、タスク・マネージャーによりサーバーの始動が完了されます。

現在実行中のタスクを停止するには、以下のいずれかを実行します。

- タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**タスクの停止**」を選択します。
- タスク・マネージャー・ビューのタイトル・バーの「**タスクの停止**」ボタンをクリックします。

タスクの強制終了

現在実行中のタスクのキューの強制終了をタスク・マネージャーに指示できます。これにより現在実行中のタスクが終了し、その後のタスクも実行されません。例えば、**サーバーを起動**タスクが実行中にタスクの強制終了を選択すると、タスク・マネージャーがサーバーの始動を完了できなくなります。

現在実行中のタスクを強制終了するには、タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**タスクの強制終了**」を選択します。

タスクの一時停止

現在実行中のタスクを一時停止して、タスク・マネージャーが現在実行中のタスクは完了するが、残りのタスクは一時停止を解除するまで実行しないようにします。これにより一連のタスクを開始し、後のタスクに到達する前に実行する必要があるが実行するのを忘れたアクションを、残りのタスクを停止して再始動しないで、実行できます。例えば、サーバーを始動して従属オブジェクトを配置するが、コンポーネントの一部がすでにリポジトリに存在することを、タスクが開始してから気がついた場合、配置段階が開始する前にタスクを一時停止します。その後、`repos_copy` を使用してリポジトリを削除し、配置が開始されたときに失敗しないようにできます。

現在実行中のタスクを一時停止するには、以下のいずれかを実行します。

- タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**タスクの一時停止**」を選択します。
- タスク・マネージャー・ビューのタイトル・バーの「**タスクの一時停止**」ボタンをクリックします。

タスクの実行を再開するには、以下のいずれかを実行します。

- タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**タスクの一時停止**」を選択します。
- タスク・マネージャー・ビューのタイトル・バーの「**タスクの一時停止**」ボタンをクリックします。

設定タスク選択の構成

選択内容を設定として保管し、最もよく使用する選択方式を適宜適用できるようにできます。

タスク選択の設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「統合テスト環境」ノードを展開します。
3. 「タスク・マネージャー」を選択します。
4. タスク・グループを展開し、設定を適用するときに選択する個々のタスクおよびタスク・グループのチェック・ボックスを使用可能にします。
5. 「OK」をクリックします。

図 80 に、タスク・マネージャーの設定インターフェースを示します。

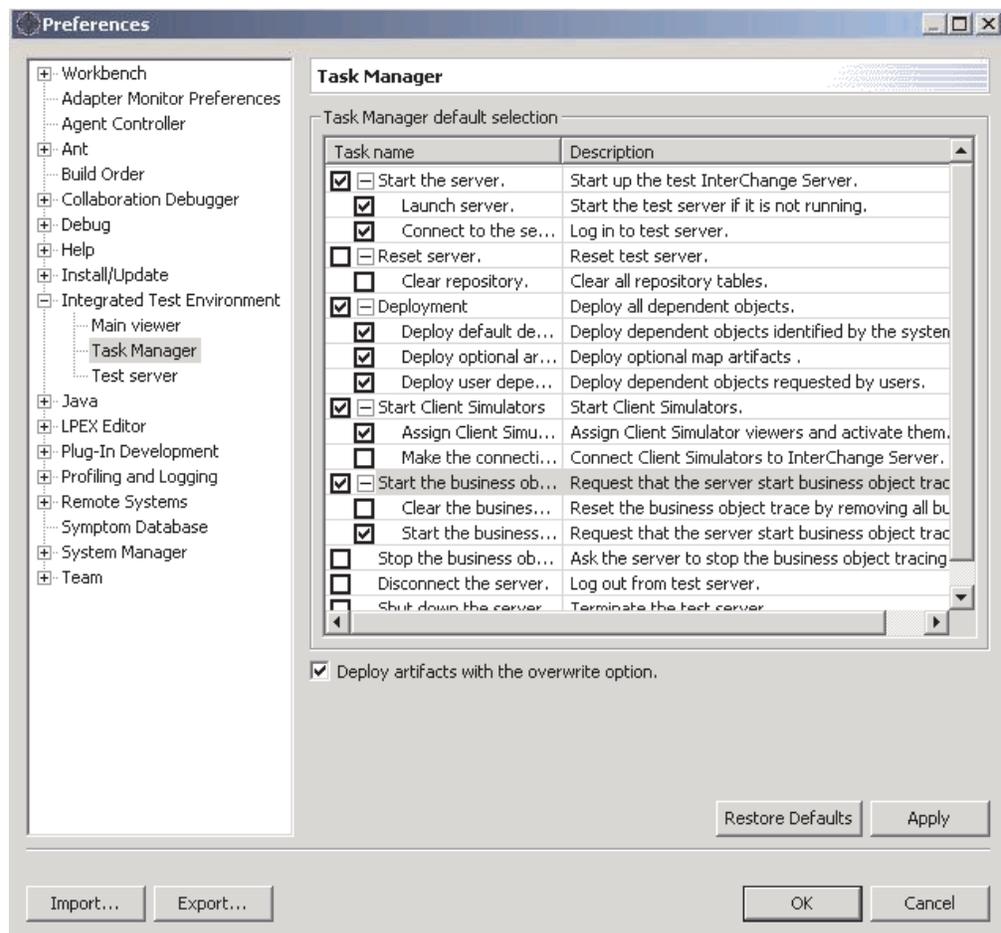


図 80. 統合テスト環境のタスク設定

デフォルトのタスク選択の構成

特定のテスト・ユニットをオープンするたびにデフォルトで使用可能にするタスクおよびタスク・グループを構成するには、以下の手順を実行します。

1. デフォルトの選択を保管するテスト・ユニットをオープンします。
2. タスク・グループを展開し、テスト・ユニットをオープンするときに選択する個々のタスクおよびタスク・グループのチェック・ボックスを使用可能にします。
3. 選択内容を保管するには、以下のいずれかを実行します。
 - タスク・マネージャー・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「**選択内容の保管**」を選択します。
 - タスク・マネージャー・ビューのタイトル・バーの「**現在の選択内容の保管**」ボタンをクリックします。

次回にテスト・ユニットをオープンすると、選択したタスクがタスク・マネージャー・ビューで選択されています。

統合テスト環境コンソール・ビューおよび InterChange Server コンソール・ビューの使用

274 ページの『統合テスト環境インターフェース』で説明するように、統合テスト環境コンソール・タブには統合テスト環境の実行についての情報が表示され、InterChange Server コンソールには InterChange Server の実行についての情報が表示されます。

統合テスト環境コンソール・ビューを使用して、統合テスト環境タスクがいつ完了したか、また正常であったかどうかを判断できます。例えば、統合テスト環境が InterChange Server に正常に接続されたときか、または統合テスト環境が Map Designer を起動してマップをデバッグするときにメッセージが書き込まれます。

統合テスト環境から InterChange Server を始動した場合、InterChange Server コンソール・ビューには MS-DOS コンソールと同じようなサーバーのロギングおよびトレース出力が表示されます。このビューを使用すると、テストするインターフェースをトラブルシューティングできます。このビューには、統合テスト環境パースペクティブ内に含まれ便利に検索できる利点があります。

次のセクションでは、統合テスト環境コンソール・ビューおよび InterChange Server コンソール・ビューを使用するときに、頻繁に実行する一部のタスクについて説明します。

テキストの検索

統合テスト環境コンソール・ビューや InterChange Server コンソール・ビューで特定の句を検索することがあります。例えば、InterChange Server が正常に始動したことを確認することがあります。これを行うには、InterChange Server コンソール・ビューから「作業可能 (is ready)」というテキストを検索します。

統合テスト環境コンソール・ビューまたは InterChange Server コンソール・ビューでテキスト・ストリングを検索するには、以下の手順を実行します。

1. 統合テスト環境コンソール・ビューまたは InterChange Server コンソール・ビューで右マウス・ボタン・クリックし、コンテキスト・メニューから「**検索/置換**」を選択します。
2. 検索するテキスト・ストリングを「**検索**」フィールドに入力します。
3. 「**進む (Forward)**」ラジオ・ボタンを使用可能にしてカーソルの後のテキストを検索するか、または「**戻る (Backward)**」ラジオ・ボタンを使用可能にしてカーソルの前のテキストを検索します。
4. 「**大文字と小文字を区別**」チェック・ボックスなどのその他のオプションを使用可能にします。
5. 「**次を検索**」をクリックします。

テキストのコピー

コンソール・ビューからテキストをコピーして、トラブルシューティングの支援を受けるために、他の開発者やテクニカル・サポートに示すことがあります。

コンソール・ビューからテキストをコピーするには、必要なテキストを選択し右マウス・ボタン・クリックして、コンテキスト・メニューから「**コピー**」を選択します。

ビュー・ウィンドウの最大化

すべてのワークベンチのエディターやビューアーのように、タイトル・バーをダブルクリックすると、統合テスト環境コンソール・ビューや InterChange Server コンソール・ビューを最大化できます。ほとんどのビューアーの場合、最大化は便利ではありませんが、InterChange Server コンソール・ビューで情報を調べてインターフェースをトラブルシューティングする場合は、非常に便利です。

ビュー・ウィンドウのクリア

統合テスト環境コンソール・ビューまたは InterChange Server コンソール・ビューをクリアして、ビューに表示されている情報を消去する必要があることがあります。これは、インターフェースの問題をデバッグするのに関連する情報のみを表示する場合に便利です。コンソール・ビューをクリアし、問題を再現し、トラブルシューティングできて、問題に関連しない情報から検索する必要がなくなります。

統合テスト環境コンソールの設定の構成

統合テスト環境コンソールの設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「**ウィンドウ**」>「**設定**」を選択します。
2. 「**統合テスト環境**」を選択します。

281 ページの図 73 に統合テスト環境の設定を示します。

3. 「**標準テキスト**」ラベルの右にあるボタンをクリックします。
4. 「**色**」ダイアログから、統合テスト環境コンソールで警告またはエラー以外のメッセージに使用する色を選択します。
5. 「**OK**」をクリックします。
6. 「**警告テキスト**」および「**エラー・テキスト**」の各ラベルについて、ステップ 3 からステップ 5 を繰り返し、これらのメッセージ・タイプの色を設定します。

7. 「OK」をクリックします。

テスト・ユニット・ビューの使用

276 ページの『テスト・ユニット・エディター』に説明するように、テスト・ユニット・ビューではインターフェースがグラフィカルに表示され、インターフェースのコンポーネントを操作するためのメカニズムが示されます。

サーバー・コンテキスト・オーバーレイの使用

サーバー・コンテキスト・オーバーレイでは、テスト・ユニット・ビュー内のインターフェースのコンポーネントについての情報が表示され、コンテキスト・メニュー・オプションが使用可能になってコンポーネントを配置しその状態を操作できます。

次のセクションで説明するように、サーバー・コンテキスト・オーバーレイを使用可能または最新表示するには、統合テスト環境をサーバーに接続する必要があります。詳しくは、294 ページの表 25 のサーバーに接続タスクを参照してください。

サーバー・コンテキスト・オーバーレイが使用可能な場合、テスト・ユニット・ビューに表示される各コンポーネントのアイコンにコンポーネントの状態を示すアイコンが追加されます。インターフェースをテストするには、インターフェースに関係するすべてのコンポーネントが実行されている必要があるため、この機能は便利です。

304 ページの図 81 に、サーバー・コンテキスト・オーバーレイが使用可能になっているテスト・ユニット・ビューを示します。ほとんどのコンポーネントには緑の矢印アイコンがあり、開始済み状態であることを示しています。**ClarifyConnector** コンポーネントにはオレンジ色の円があり、一時停止状態を示しています。コラボレーション・オブジェクト **CustomerSync_ClarifyToCaesar** には赤色の正方形のアイコンがあり、停止状態であることを示しています。304 ページの表 27 に、各種のコンポーネント状態アイコンとその意味を説明します。

コンポーネント状態の詳細については、「システム管理ガイド」を参照してください。

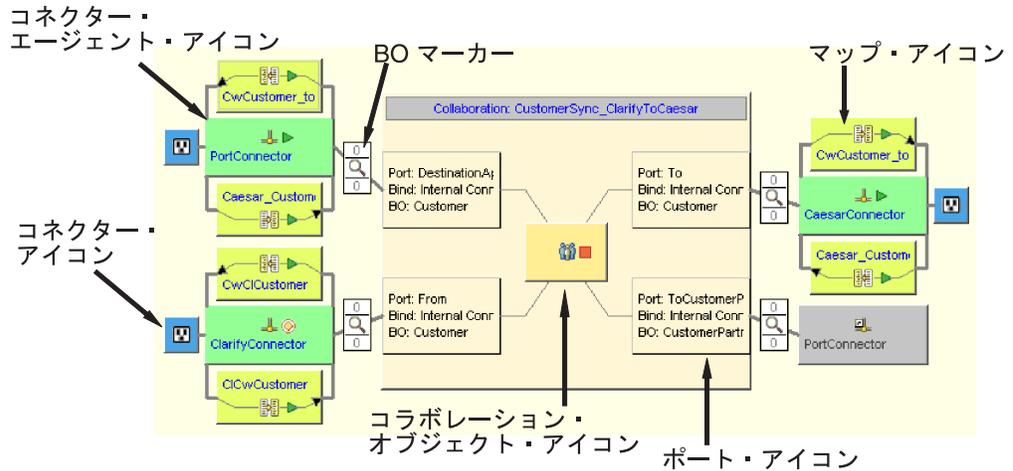


図 81. サーバー・コンテキスト・オーバーレイが使用可能なテスト・ユニット・ビュー

表 27. コンポーネント状態アイコンと意味

アイコン	意味
緑色の矢印	コンポーネントは開始済みです。
オレンジ色の円	コンポーネントは一時停止しています。
赤色の正方形	コンポーネントは停止しています。
疑問符	コンポーネントの状態は不明です。
赤色の円に白の「X」印	コンポーネントはリポジトリに存在しません。
青色のチェックマーク	コンポーネントは正常に配置されました。
赤色の「X」印	統合テスト環境はサーバーから切断されているので、サーバー・コンテキスト・オーバーレイでコンポーネントの現在の状態を表示できません。

サーバー・コンテキスト・オーバーレイを使用可能および使用不可にする

サーバー・コンテキスト・オーバーレイを使用可能にするには、テスト・ユニット・ビューの背景を右マウス・ボタン・クリックして、コンテキスト・メニューから「サーバー・コンテキスト・オーバーレイ」を選択します。

サーバー・コンテキスト・オーバーレイが使用可能な場合、テスト・ユニット・ビューにコンポーネント状態アイコンが表示され、コンポーネントの右マウス・ボタン・クリック・メニュー項目が使用可能になります。

サーバー・コンテキスト・オーバーレイを使用不可にするには、テスト・ユニット・ビューの背景を右マウス・ボタン・クリックして、コンテキスト・メニューから「サーバー・コンテキスト・オーバーレイ」を選択します。

サーバー・コンテキスト・オーバーレイを常時表示するには、統合テスト環境の設定で「サーバー・コンテキスト・オーバーレイを常に表示」オプションを有効にします。281 ページの図 73 に統合テスト環境の設定を示します。

サーバー・コンテキスト・オーバーレイの最新表示

サーバー・コンテキスト・オーバーレイ内の状態情報は動的ではないので、サーバー・コンテキスト・オーバーレイが最初に使用可能になってから、統合テスト環境の外部アクションによってコンポーネントの状態が変更されると、サーバー・コンテキスト・オーバーレイに表示される状態情報は不正確になります。例えば、System Manager の InterChange Server コンポーネント管理ビューを使用してコンポーネントの状態を変更しても、サーバー・コンテキスト・オーバーレイは依然として変更前のコンポーネントの状態を反映しています。現在の状態情報でサーバー・コンテキスト・オーバーレイを更新するには、最新表示する必要があります。

サーバー・コンテキスト・オーバーレイを最新表示するには、テスト・ユニット・ビューの背景を右マウス・ボタン・クリックして、コンテキスト・メニューから「サーバー・コンテキスト・オーバーレイの最新表示」を選択します。

テスト・ユニット・ビューを使用したリポジトリの管理

インターフェースをテストするには、インターフェースに必要なすべてのコンポーネントをサーバー・リポジトリに配置する必要があります。アウトライン・ビューを使用すると、InterChange Server リポジトリと連動して、必要なコンポーネントが含まれていることを確認できます。

すでにサーバー・リポジトリに存在するコンポーネントを配置するには、テスト・ユニット・ビューでコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「上書きして配置」を選択します。

サーバー・リポジトリに存在しないコンポーネントを配置するには、テスト・ユニット・ビューでコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「配置」を選択します。選択したコンポーネントがすでにサーバー・リポジトリに存在する場合に「配置」オプションを使用すると、配置に失敗します。

テスト・ユニット・ビューを使用してサーバー・リポジトリからコンポーネントを削除するには、ビュー内のコンポーネントを右マウス・ボタンでクリックし、コンテキスト・メニューから「サーバーから除去」を選択します。

統合テスト環境コンソールおよび InterChange Server コンソールに項目が追加され、配置が成功したかどうかを示されます。

テスト・ユニット・ビューを使用したコンポーネント状態の管理

テスト・ユニット・ビューを使用して、コンポーネントの状態を変更できます。これによりテスト中に発見した問題を解決できます。つまり、System Manager、repos_copy、または System Monitor を使用しないで、コンポーネントを非活動化し、それを変更し、再配置して、再活動化できます。

コンポーネントの状態を変更するには、サーバー・コンテキスト・オーバーレイ内のコンポーネントのアイコンを右マウス・ボタンでクリックし、目的の状態操作（「開始」、「一時停止」、または「停止」）を選択します。すべてのコンポーネント・タイプにすべての状態操作が用意されているとは限りません。

コンポーネント状態の詳細については、「システム管理ガイド」を参照してください。

テスト・ユニット・ビューを使用した Client Simulator ビューの使用、表示、およびクローズ

Client Simulator ビューを使用すると、インターフェースをテストするときに、コネクタとアクセス・クライアントをエミュレートできます。Client Simulator ビューには、テスト・データを持つビジネス・オブジェクト・インスタンスを作成したり、そのインスタンスをビジネス・オブジェクト要求として送信したり、ビジネス・オブジェクト応答としてその要求に応答するためのインターフェースが用意されています。

コネクタのプロファイルを特定の Client Simulator ビューにロードするには、テスト・ユニット・ビューのコネクタのアイコンを右マウス・ボタンでクリックし、「クライアント・シミュレーター *x* を使用」を選択します。ここで、*x* は、そのコネクタに関して使用する Client Simulator ビュー位置の番号です。

310 ページの表 29 に、各種 Client Simulator ビューが表示されるデフォルトのインターフェース位置を示します。

統合テスト環境内のビューは、テストの各段階で、ほかのビューを使用することにより隠すことができます。例えば、BO インспекター・ビューを使用して Client Simulator ビューを隠すことができます。隠した Client Simulator ビューを表示するには、コネクタ・アイコンを右マウス・ボタンでクリックし、「クライアント・シミュレーターを表示」を選択して、Client Simulator ビューにフォーカスを与えます。

Client Simulator ビューを閉じるには、コネクタ・アイコンを右マウス・ボタンでクリックし、「クライアント・シミュレーターを閉じる」を選択します。

注: このオプションは Client Simulator ビューを閉じるだけでなく、Client Simulator ビューでのコネクタのエミュレーションを停止させます。

Client Simulator ビューの使用法について詳しくは、309 ページの『Client Simulator ビューの使用』を参照してください。

BO インспекター・ビューの表示

ビジネス・オブジェクト・トレースを使用する場合、統合テスト環境により、インターフェースの実行中の特定時点でビジネス・オブジェクト・データが記録されます。

BO マーカーのビジネス・オブジェクト・データを特定の BO インспекター・ビューに表示するには、BO マーカーを右マウス・ボタンでクリックし、「BO ビューアー *x* に表示」を選択します。ここで、*x* は、そのマーカーに関して使用する BO インспекター・ビュー位置の番号です。BO マーカー・アイコンは、304 ページの図 81 に示してあります。

327 ページの表 31 に、各種 BO インспекター・ビューが表示されるデフォルトのインターフェース位置を示します。

ビジネス・オブジェクト・トレースの詳細については、294 ページの表 25 を参照してください。

BO インスペクター・ビューの使用法の詳細については、325 ページの『BO インスペクター・ビューの使用』を参照してください。

BO マーカーの最新表示

ビジネス・オブジェクト・トレースを使用しているが、トレース情報の自動更新が使用不可の場合は、BO マーカーを明示的に最新表示して更新する必要があります。

BO マーカーを更新するには、サーバー・コンテキスト・オーバーレイを右マウス・ボタン・クリックして、コンテキスト・メニューから「BO マーカーの最新表示」を選択します。

テスト・ユニット・ビューを使用したマップのデバッグ

ビジネス・オブジェクト・トレースを使用すると、統合テスト環境内でテストの実行中に作成されたデータを使用して、汎用オブジェクトをアプリケーション固有のオブジェクトに変換するマップに関してデバッグ・モードで Map Designer を起動できます。

テスト・ユニット・ビューを使用してマップをデバッグするには、以下の手順を実行します。

1. サーバー・コンテキスト・オーバーレイを使用可能にします。詳しくは、304 ページの『サーバー・コンテキスト・オーバーレイを使用可能および使用不可にする』を参照してください。
2. 以下のセクションのいずれかの説明に従って、BO トレースの開始タスクを実行します。
 - 290 ページの『アウトライン・ビューを使用したタスクの実行』
 - 293 ページの『タスク・マネージャー・ビューの使用』
3. インターフェースを使用してビジネス・オブジェクトを送信する場合は、デバッグするマップのアイコンを右マウス・ボタン・クリックしてコンテキスト・メニューから「マップのデバッグ」を選択します。
4. 「BO インスタンスの選択」ウィンドウで、使用するビジネス・オブジェクト・インスタンスを選択し、「OK」をクリックします。

Map Designer がデバッグ・モードで起動し、指定されたマップがオープンされ、指定されたテスト・データが設定されます。Map Designer のデバッグ機能の使用法の詳細については、「マップ開発ガイド」を参照してください。

注: Map Designer は、統合テスト環境とは別個に実行されます。Map Designer が起動されると、統合テスト環境は Map Designer との通信を続行しません。

テスト・ユニット・ビュー・スケールの変更

テスト・ユニット・ビューのスケールの変更には、複数のオプションを使用できません。

テスト・ユニット・ビュー・スケールを変更するには、サーバー・コンテキスト・オーバーレイを右マウス・ボタン・クリックして、コンテキスト・メニューから必要なスケールを選択します。表 28 に、スケール・オプションのリストと説明を示します。

表 28. テスト・ユニット・ビュー・スケールと説明

テスト・ユニット・ビュー・スケールのタイプ	説明
ウィンドウに合わせて表示	インターフェースの図を変形してテスト・ユニット・ビューの幅に合わせます。
ウィンドウ全体表示	インターフェースの図を変形してテスト・ユニット・ビュー全体に合わせて表示します。通常、これによりインターフェース・アイコンは縦方向に伸びます。
標準目盛りの表示	これはデフォルトのスケールです。
二重目盛りの表示	インターフェースの図を 200 % 拡大して表示します。

テスト・ユニット・ビューの設定の構成

テスト・ユニット・ビューの設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「統合テスト環境」を展開します。
3. 「メイン・ビューアー」を選択します。

281 ページの図 73 に統合テスト環境の設定を示します。

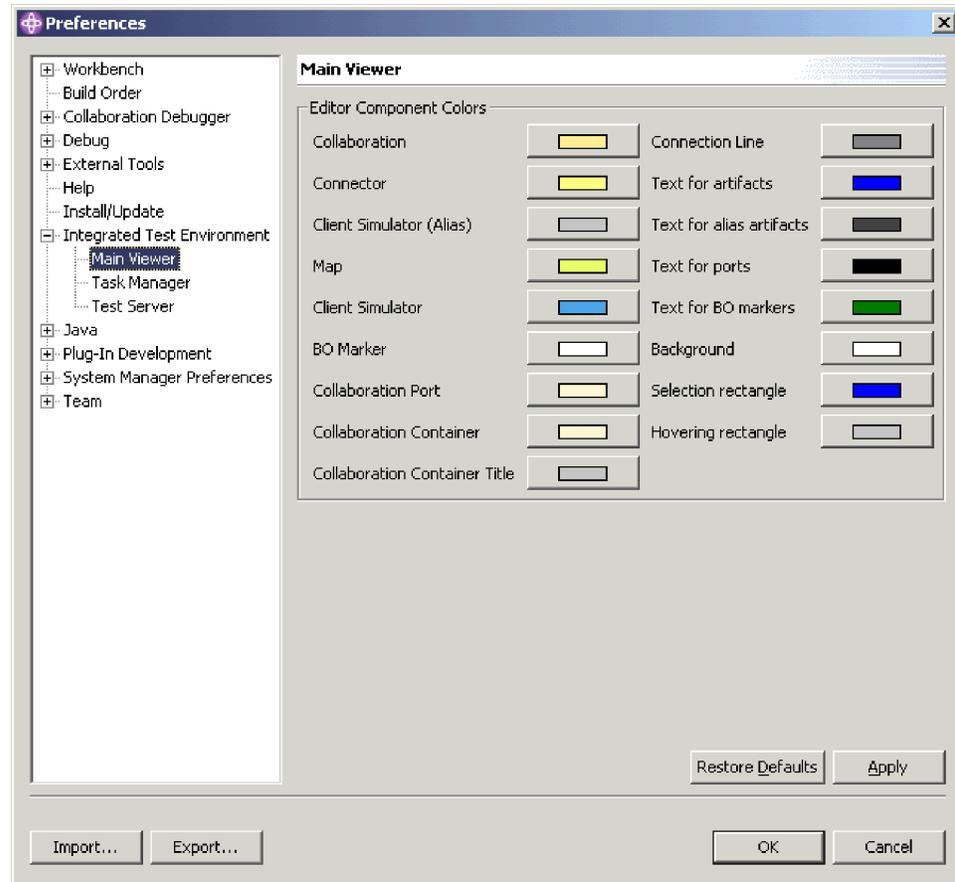


図 82. テスト・ユニット・ビューの設定

4. 「コンポーネント・カラーを編集 (Edit Component Colors)」ペインのコンポーネント・ラベルの右にあるボタンをクリックします。
5. 「色」ダイアログから、テスト・ユニット・ビューで、選択したコンポーネント・タイプの背景に使用する色を選択します。
6. 「OK」をクリックします。
7. ほかのコンポーネント・タイプについてもステップ 4 からステップ 6 を繰り返します。
8. 「OK」をクリックします。

Client Simulator ビューの使用

Client Simulator ビューにより、IBM WebSphere InterChange Server インターフェースのクライアント (コネクタ・エージェントおよびアクセス・クライアント) をエミュレートできます。このビューに、サーバーへのクライアントの接続、ビジネス・オブジェクト要求の作成および送信、ビジネス・オブジェクト応答による応答を行うインターフェースが用意されます。

Client Simulator ビューの表示およびクローズ

次のセクションで、Client Simulator ビューを表示およびクローズできる多くの方法を説明します。

表 29 の「デフォルトのインターフェース位置」列には、インターフェース内のデフォルト位置が「Client Simulator ビュー番号」列の Client Simulator ビューでオーバーレイされるビューをリストします。

表 29. デフォルトの Client Simulator ビューの位置

Client Simulator ビュー番号	デフォルトのインターフェース位置
Test Connector 1	ITE ナビゲーター
Test Connector 2	アウトライン
Test Connector 3	プロパティ
Test Connector 4	依存関係
Test Connector 5	ITE ナビゲーター
Test Connector 6	アウトライン
Test Connector 7	プロパティ
Test Connector 8	依存関係
Test Connector 9	ITE ナビゲーター

Client Simulator ビューの表示

テストするインターフェース内のコネクタおよびアクセス・クライアントについて Client Simulator ビューを表示するには、次のいずれかの手法を使用します。

- 291 ページの『アウトライン・ビューを使用した Client Simulator ビューの表示およびクローズ』ならびに 294 ページの表 25
- 293 ページの『タスク・マネージャー・ビューの使用』ならびに 294 ページの表 25
- 306 ページの『テスト・ユニット・ビューを使用した Client Simulator ビューの使用、表示、およびクローズ』ならびに 294 ページの表 25
- メニュー・バーから「統合テスト環境」>「すべてのクライアント・シミュレーター・ビューを表示」を選択し、9 個の Client Simulator ビューをすべて表示します。
- メニュー・バーから「ウィンドウ」>「ビューの表示」>「x:クライアント・シミュレーター」を選択して特定の Client Simulator を起動します。ここで、x は、そのコネクタに関して使用する Client Simulator ビュー位置の番号です。

表 29 に、各種 Client Simulator ビューが表示されるデフォルトのインターフェース位置を示します。

Client Simulator ビューのクローズ

Client Simulator ビューをクローズするには、次のいずれかの手法を使用します。

- 291 ページの『アウトライン・ビューを使用した Client Simulator ビューの表示およびクローズ』ならびに 294 ページの表 25
- 306 ページの『テスト・ユニット・ビューを使用した Client Simulator ビューの使用、表示、およびクローズ』。
- メニュー・バーから「統合テスト環境」>「すべてのクライアント・シミュレーター・ビューを閉じる」を選択し、オープンしている Client Simulator ビューをすべてクローズします。

- Client Simulator ビューのタイトル・バーの右側の「閉じる」ボタンをクリックするか、Client Simulator ビューのタイトル・バーを右マウス・ボタンでクリックして、コンテキスト・メニューから「閉じる」を選択します。

サーバーへの Client Simulator ビューの接続

インターフェースのテストの一部としてコネクタをエミュレートするには、Client Simulator ビューをサーバーに接続する必要があります。

Client Simulator ビューを使用する場合、リポジトリ内の定義か構成ファイルを使用してコネクタをエミュレートできます。また、Client Simulator ビューを使用してアクセス・クライアントをエミュレートすることもできます。

タスク・マネージャー・ビューを使用してエージェントをサーバーに接続することもできます。詳細については、294 ページの表 25 を参照してください。

リポジトリ定義を使用したコネクタのエミュレート

コネクタのリポジトリ定義を使用してサーバーに接続するには、以下のいずれかを実行します。

- ツールバーの「接続」ボタンをクリックします。
- Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「接続」を選択します。

注: リポジトリ定義を使用してサーバーに接続するには、エミュレートするコネクタの `DeliveryTransport` プロパティが値 `IDL` に設定されている必要があります。

構成ファイルを使用したコネクタのエミュレート

構成ファイルを使用してサーバーに接続するには、以下の手順を実行します。

1. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「*.cfg と接続します」を選択します。
2. コネクタ構成ファイルに移動してオープンします。

注: エミュレートするコネクタの `DeliveryTransport` プロパティが、値 `IDL` に設定されていない場合に、メニューから「サーバー」>「接続」を選択すると、統合テスト環境では使用する構成ファイルの指定を求めるダイアログが表示されます。

アクセス・クライアントのエミュレート

アクセス・クライアントをエミュレートするには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「接続」を選択します。

サーバーへのクライアントの接続の確認

Client Simulator がサーバーへ正常に接続されたかどうかを判断するには、以下の標識を使用します。

- アウトライン・ビューでは、クライアントが通信に使用するポートのノードが緑色になり、「作動可能」を読み取ります。

- クライアントがコネクターの場合は、メッセージ「[AppConnector: コネクターは回復されました。]」が Client Simulator ビューの状況ペインに書き込まれます。
- クライアントがアクセス・クライアントの場合は、メッセージ「作動可能」が Client Simulator ビューの状況ペインに書き込まれます。
- 「**BO タイプ**」および「**BO インスタンス**」の各ドロップダウン・メニューが、Client Simulator ビューの入力ペインで使用可能になります。

状況ペインの詳細については、314 ページの『状況ペインの使用』を参照してください。

入力ペインの詳細については、『入力ペインの使用』を参照してください。

サーバーからの切断

コネクターをエミュレートする Client Simulator をサーバーから切断するには、Client Simulator ビューの右端の下矢印をクリックし、「サーバー」>「切断」を選択します。

Client Simulator ビューは、クローズされてもサーバーから切断されます。アクセス・クライアントをエミュレートする Client Simulator ビューをサーバーから切断する唯一の方法は、Client Simulator ビューのクローズです。Client Simulator ビューのクローズについて詳しくは、310 ページの『Client Simulator ビューのクローズ』を参照してください。

Client Simulator ビュー・ペインとペインの配置

Client Simulator ビューには幾つかのペインがあり、コネクターまたはアクセス・クライアントの振る舞いのシミュレートに使用します。

入力ペインの使用

Client Simulator ビューの入力ペインを使用して、ビジネス・オブジェクト要求を処理し送信します。ビジネス・オブジェクト・インスタンスを作成し、その属性にデータを追加して、それを要求として送信できます。インターフェース内のソース・クライアントをエミュレートする Client Simulator ビューで作業する場合、通常は入力ペインを使用します。

313 ページの図 83 に、ソース・コネクターをエミュレートする Client Simulator ビューの入力ペインを示します。

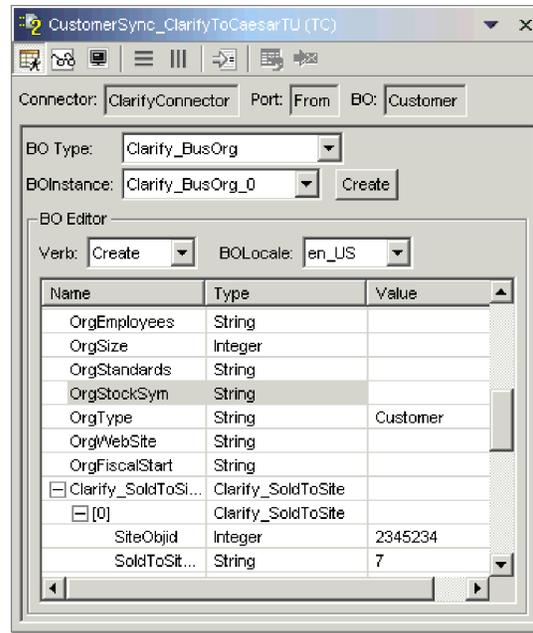


図 83. Client Simulator ビューの入力ペイン

結果ペインの使用

Client Simulator ビューの結果ペインを使用して、受信したビジネス・オブジェクト要求を処理し、ビジネス・オブジェクト応答を送信します。インターフェース内の宛先クライアントをエミュレートする Client Simulator ビューで作業する場合、通常は結果ペインを使用します。

314 ページの図 84 に、宛先コネクタをエミュレートする Client Simulator ビューの結果ペインを示します。

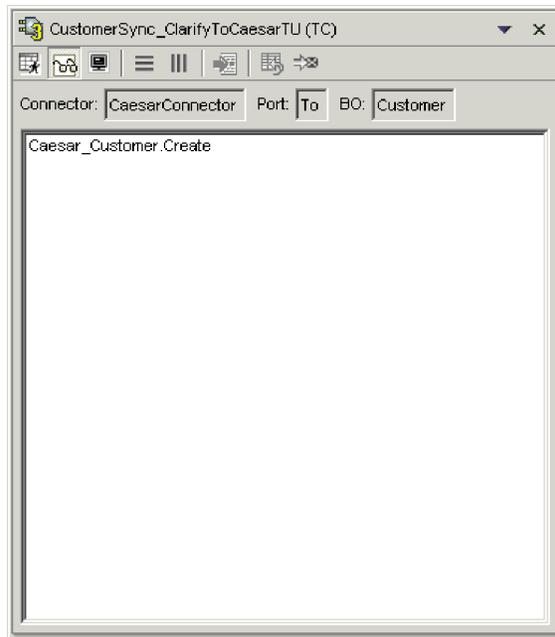


図 84. Client Simulator ビューの結果ペイン

状況ペインの使用

状況ペインには、Client Simulator ビューの実行についてのメッセージが表示されます。例えば、コネクタがサーバーに正常に接続されると、メッセージ「[AppConnector: コネクタは回復されました。]」が状況ペインに書き込まれます。

状況ペインは主に、Client Simulator がサーバーに正常に接続されたことの確認と、トラブルシューティングの実行に使用されます。例えば、Client Simulator ビューからビジネス・オブジェクト要求を送信したが、ビジネス・オブジェクト・タイプのサブスクリプションがない場合は、メッセージが状況ペインに書き込まれます。これにより、インターフェース内の宛先クライアントをエミュレートする Client Simulator ビューの結果ペインに ビジネス・オブジェクト要求が表示されない理由を判別できます。

315 ページの図 85 に、Client Simulator の状況ペインを示します。

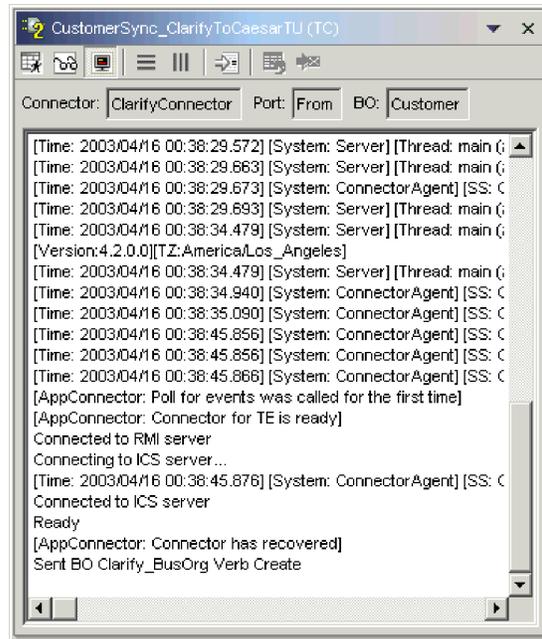


図 85. Client Simulator ビューの状況ペイン

すべてのペインを縦方向に配置

入力ペイン、結果ペイン、および状況ペインを積み重ね、同時に表示されるように、Client Simulator ビューを配置できます。この表示は、Client Simulator ビューで、イベント通知と要求処理の両方を行う必要のあるクライアントをエミュレートする場合に便利です。

Client Simulator ビュー・ペインを縦方向に配置するには、以下のいずれかを実行します。

- Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「ウィンドウ」>「縦方向に配置」を選択します。
- Client Simulator ビューのタイトル・バーの「すべてを縦方向に表示」ボタンをクリックします。

316 ページの図 86 に、縦方向に配置された Client Simulator ビュー・ペインを示します。

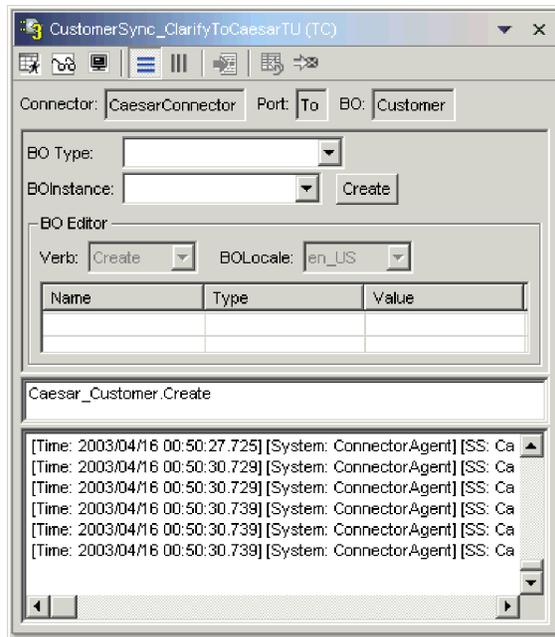


図 86. 縦方向に配置された Client Simulator ビュー・ペイン

すべてのペインを横方向に配置

入力、結果、および状況ペインを左から右に、同時に表示されるように、Client Simulator ビューを配置できます。この表示は、Client Simulator ビューで、イベント通知と要求処理の両方を行う必要のあるクライアントをエミュレートする場合に便利です。

Client Simulator ビュー・ペインを横方向に配置するには、以下のいずれかを実行します。

- Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「ウィンドウ」>「横方向に配置」を選択します。
- Client Simulator ビューのタイトル・バーの「すべてを横方向に表示」ボタンをクリックします。

図 86 に、横方向に配置された Client Simulator ビュー・ペインを示します。

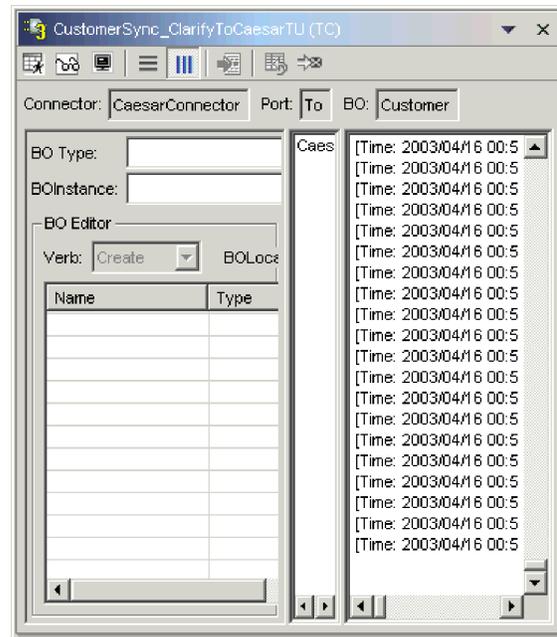


図 87. 横方向に配置された Client Simulator ビュー・ペイン

クライアント・シミュレーター・ビューでのビジネス・オブジェクト要求の処理

要求ビジネス・オブジェクトは、インターフェースを起動するイベントのソースであるコネクタをエミュレートしているときに、Client Simulator ビューから送信するビジネス・オブジェクトです。要求ビジネス・オブジェクトの処理は、ビジネス・オブジェクト・インスタンスの作成、データの取り込み、要求の送信から構成されます。

要求ビジネス・オブジェクトの作成

新規のビジネス・オブジェクト・インスタンスを作成するには、以下の手順を実行します。

1. 入力ペインで、「**BO タイプ**」ドロップダウン・メニューから作成するビジネス・オブジェクトの名前を選択します。
2. 以下のいずれかを実行します。
 - 「**BO インスタンス**」フィールドの横にある「**作成**」をクリックします。
 - Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「編集」>「**ビジネス・オブジェクトを作成**」を選択します。
3. 「新規インスタンス」ダイアログが表示されたら、「名前を入力してください」フィールドにインスタンスの名前を入力します。
4. 「**動詞**」ドロップダウン・メニューから適切な動詞を選択します。
5. 「**ビジネス・オブジェクト・ロケール**」ドロップダウン・メニューから目的のロケールを選択します。

6. トップレベル・オブジェクト内の単純属性および子ビジネス・オブジェクトの値を指定します。詳細については、320 ページの『ビジネス・オブジェクト属性の値の設定』を参照してください。
7. 「OK」をクリックします。

要求ビジネス・オブジェクトの非同期送信

ソース・クライアントが非同期モードで要求ビジネス・オブジェクトを送信する場合は、応答ビジネス・オブジェクトの返送を予期していません。要求ビジネス・オブジェクトがディスパッチされると、トランザクションにおけるソース・クライアントの役割は終了します。通常、応答ビジネス・オブジェクトは統合ブローカーによって処理されます。Client Simulator ビューのデフォルトのモードは非同期モードです。

ビジネス・オブジェクトを非同期的に送信するには、以下の手順を実行します。

1. 要求ビジネス・オブジェクト・インスタンスを作成します。詳しくは、317 ページの『要求ビジネス・オブジェクトの作成』を参照してください。
2. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「モード」>「非同期」を選択します。

注: Client Simulator ビューはデフォルトでは「非同期」モードで動作するため、このステップの実行が必要となるのは、以前にビューから同期要求を送信している場合のみです。各要求を送信する前にモードを設定する必要はありません。

3. 要求を送信するには、以下のいずれかを実行します。
 - Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「送信」を選択します。
 - Client Simulator ビューのツールバーの「ビジネス・オブジェクトを送信」ボタンをクリックします。

要求ビジネス・オブジェクトの同期送信

ソース・クライアントが要求ビジネス・オブジェクトを同期的に送信する場合は、宛先アプリケーションが要求を処理した後、統合ブローカーから応答ビジネス・オブジェクトが返送されることを予期しています。

1. 要求ビジネス・オブジェクト・インスタンスを作成します。詳しくは、317 ページの『要求ビジネス・オブジェクトの作成』を参照してください。
2. Client Simulator ビューを同期モードに設定するには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「モード」>「同期」を選択します。
3. 要求を送信するには、以下のいずれかを実行します。
 - Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「送信」を選択します。
 - Client Simulator ビューのツールバーの「ビジネス・オブジェクトを送信」ボタンをクリックします。
4. 「コラボレーションの選択」ダイアログが表示される場合、「コラボレーション」ドロップダウン・メニューからビジネス・オブジェクトの送信先のコラボレーションを選択して、「OK」をクリックします。

処理を選択したコラボレーション・オブジェクトの構成済みポートにビジネス・オブジェクト要求が送信されます。

バッチ・モードでのビジネス・オブジェクトの送信

バッチ・モードでは、送信する特定のビジネス・オブジェクトのインスタンスの数を Client Simulator ビューで指定できます。また、インスタンスごとにトップレベル・オブジェクト内の任意の属性 (基本キー属性など) を固有値に設定するよう指定することもできます。Client Simulator ビューは、インスタンスごとに指定した単一属性の値を増分しながら、指定した回数だけビジネス・オブジェクトをコピーし、各インスタンスを送信します。このオプションにより、多数のビジネス・オブジェクトを素早く簡単に作成できます。

選択した属性が一致関係の一部として動的相互参照に参加するキー・フィールドである場合は、初期値およびそれに従う値はすべて固有でなければなりません。さもなければ、相互参照ロジックが失敗し、要求ビジネス・オブジェクトも失敗します。

値が固有であることを確認するには、Relationship Manager を使用するか、以下のように関係参加者の表に対して SQL ステートメントを実行します。

- 参加者の現在の最高値を判別し、「初期値」フィールドをそれより高い値に設定します。バッチの最初のビジネス・オブジェクト・インスタンスおよびその後のインスタンスが固有になります。
- 参加者の既存の表エントリーを削除します。これにより、バッチ・ビジネス・オブジェクトの値と同じ属性値を持つエントリーが存在しないことが保証されます。

バッチ・モードでビジネス・オブジェクトを送信するには、以下の手順を実行します。

1. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「サーバー」>「バッチを送信」を選択します。
2. 「バッチ・モード」ウィンドウで、「動詞」ドロップダウン・メニューから、必要な動詞を選択する。
3. 「ビジネス・オブジェクト・ロケール」ドロップダウン・メニューから目的のロケールを選択します。
4. バッチの各ビジネス・オブジェクト要求で増分するトップレベル・ビジネス・オブジェクトの属性を「属性」リストから選択します。

通常、選択する属性は、ビジネス・オブジェクトを一意的に識別する属性 (基本キーなど) にしてください。

5. 「初期値」フィールドに、増分する属性の開始値を入力します。
6. 「ビジネス・オブジェクト数」フィールドに、生成および送信するビジネス・オブジェクトの数を入力します。
7. 「OK」をクリックします。

Client Simulator ビューが、指定した数のビジネス・オブジェクトを生成します。これらのビジネス・オブジェクトは、指定の属性の値がインスタンスごとに増分され、それ以外はすべて同一です。

図 88 では、以下のようなバッチ・モード構成を示しています。

- 50 個のビジネス・オブジェクトを送信します。
- 属性 OrgObjid の値を増分します。
- 属性の開始値は 10000 です。

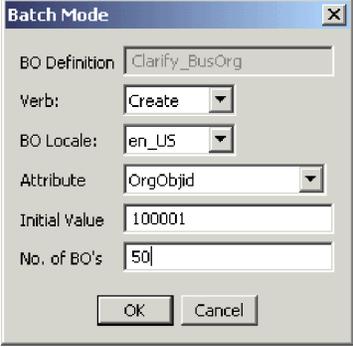


図 88. バッチ・モードでのビジネス・オブジェクトの送信

ビジネス・オブジェクト属性の値の設定

ビジネス・オブジェクト要求の作成および受信したビジネス・オブジェクト要求の編集の際に、ビジネス・オブジェクト属性の値を設定できるので、応答として戻すことができます。

要求として送信するビジネス・オブジェクトの属性の値を設定するには、Client Simulator ビューの入力ペインを使用する必要があります。詳細については、312 ページの『入力ペインの使用』を参照してください。

応答として送信するために編集するビジネス・オブジェクトの属性の値を設定するには、「応答ビジネス・オブジェクト」ウィンドウを使用します。詳細については、322 ページの『応答ビジネス・オブジェクトの編集』を参照してください。

単純属性の値の設定

単純属性の値を指定するには、入力ペインまたは「応答ビジネス・オブジェクト」ウィンドウの「値」列のセルをクリックして、値を入力します。

子ビジネス・オブジェクトの追加

子ビジネス・オブジェクトのインスタンスを追加するには、入力ペインまたは「応答ビジネス・オブジェクト」ウィンドウで子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「インスタンスを追加」を選択します。

子ビジネス・オブジェクトを表す属性の横に正符号 (+) が追加され、少なくとも 1 つの子ビジネス・オブジェクト・インスタンスが存在することが示されます。子オブジェクト属性を展開すると、インスタンスごとに番号付きのエントリが表示されます。個々のインスタンスの横にも正符号 (+) があるため、個々のインスタンスを展開したり属性の値を設定したりすることができます。

さらに子ビジネス・オブジェクト・インスタンスを追加するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「**インスタンスを追加**」を選択します。

注: 子ビジネス・オブジェクトを参照する属性の「**カード**」プロパティが値 1 (単一カーディナリティー) に設定されている場合は、子オブジェクトに追加できるインスタンスは 1 つのみです。

子ビジネス・オブジェクトの除去

子ビジネス・オブジェクトのインスタンスを除去するには、インスタンスを右マウス・ボタンでクリックし、コンテキスト・メニューから「**インスタンスを削除**」を選択します。

子ビジネス・オブジェクトのすべてのインスタンスを削除するには、子ビジネス・オブジェクトを表す属性を右マウス・ボタンでクリックし、コンテキスト・メニューから「**全インスタンスを削除**」を選択します。

子ビジネス・オブジェクトの動詞の設定

子ビジネス・オブジェクトの動詞を設定すると、値がビジネス・プロセスに及ぼす影響をテストできます。これは、子オブジェクトの相互参照用を含むロジックのトラブルシューティングを行う場合に役立ちます。

子ビジネス・オブジェクト・インスタンスの動詞を設定するには、その子ビジネス・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「**動詞を設定**」を選択します。「動詞の選択」プロンプトが表示されたら、適切な動詞を選択し、「**OK**」をクリックします。

ビジネス・オブジェクト属性のデフォルト値へのリセット

ビジネス・オブジェクト要求の属性をデフォルト値にリセットするには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「**編集**」>「**ビジネス・オブジェクトをリセット**」を選択します。

ビジネス・オブジェクト応答の属性をデフォルト値にリセットするには、「**応答ビジネス・オブジェクト**」ウィンドウのツールバーの「**ビジネス・オブジェクトをデフォルトにリセット**」ボタンをクリックします。

ビジネス・オブジェクト属性値のクリア

ビジネス・オブジェクト要求の属性の値をクリアするには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「**編集**」>「**ビジネス・オブジェクトをクリア**」を選択します。

ビジネス・オブジェクト応答の属性の値をクリアするには、「**応答ビジネス・オブジェクト**」ウィンドウのツールバーの「**ビジネス・オブジェクト値をクリア**」ボタンをクリックします。

応答ビジネス・オブジェクトの処理

応答ビジネス・オブジェクトは、インターフェースでビジネス・オブジェクト要求の受信側となるコネクターをエミュレートしているときに、Client Simulator ビュー

から送信するビジネス・オブジェクトです。要求ビジネス・オブジェクトの処理は、ビジネス・オブジェクト・インスタンス内の値の編集およびサーバーへの応答の返送から構成されます。

317 ページの『クライアント・シミュレーター・ビューでのビジネス・オブジェクト要求の処理』に従ってビジネス・オブジェクト要求を作成および送信し、インターフェースで正常に処理されると、ビジネス・オブジェクトがインターフェース内の任意の宛先コネクタの結果ペインに表示されます。314 ページの図 84 に、ビジネス・オブジェクト要求が受信された結果ペインを示します。

『応答ビジネス・オブジェクトの編集』で、受信されたビジネス・オブジェクト要求の編集方法を説明し、『応答ビジネス・オブジェクトの送信』で、インスタンスを応答として送信する方法を説明します。

応答ビジネス・オブジェクトの編集

インターフェースの宛先クライアントをエミュレートする Client Simulator ビューでビジネス・オブジェクト要求を受信する場合、属性の値を編集するのが一般的です。例えば、関係に参加する基本キー属性の固有値を指定したり、他の属性の値を変更してビジネス・オブジェクトの実際の値に応じて応答を変えるマップまたはコラボレーション・ロジックをテストしたりすることができます。ビジネス・オブジェクト属性の値を設定するには、以下の手順を実行します。

1. 「応答ビジネス・オブジェクト」ウィンドウでビジネス・オブジェクト要求を編集するには、次のいずれかを実行します。
 - 結果ペインでビジネス・オブジェクト・インスタンスをダブルクリックします。
 - 結果ペインでビジネス・オブジェクト・インスタンスを選択し、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックして、「応答」>「応答を編集」を選択します。
2. ビジネス・オブジェクトの属性を編集するには、以下のいずれかを実行します。
 - 320 ページの『ビジネス・オブジェクト属性の値の設定』で説明しているいずれかの方法を使用してビジネス・オブジェクト属性の値を変更します。
 - ファイルからビジネス・オブジェクト・データをインポートします。詳しくは 324 ページの『応答ビジネス・オブジェクトのインポート』を参照してください。

保管したデータをビジネス・オブジェクト要求にインポートする機能は、応答として送信する前に応答ビジネス・オブジェクトにデータを設定する必要がある場合に非常に有用です。応答データを必要とする属性ごとに値を手動で入力する代わりに、値を 1 回入力し、ビジネス・オブジェクトをエクスポートして (324 ページの『応答ビジネス・オブジェクトのエクスポート』を参照)、その後のテストで保管したデータをインポートします。

応答ビジネス・オブジェクトの送信

要求ビジネス・オブジェクトを編集後 (編集が必要な場合)、応答としてサーバーに返送します。

323 ページの表 30 に、応答オプションをリストし、C++ および Java コネクタ両方に関する対応するコネクタ戻りコードを示します。C++ コネクタまたは

Java コネクタの戻りコードについては、「コネクタ開発ガイド (C++ 用)」または「コネクタ開発ガイド (Java 用)」を参照してください。

表 30. Client Simulator ビューの応答タイプおよびコネクタ戻りコード

Client Simulator ビューの応答タイプ	C++ コネクタの戻りコード	Java コネクタの戻りコード
成功	BON_SUCCESS	SUCCESS
失敗	BON_FAIL	FAIL
複数のヒット	BON_MULTIPLE_HITS	MULTIPLE_HITS
内容による検索に失敗	BON_FAIL_RETRIEVE_BY_CONTENT	RETRIEVEBYCONTENT_FAILED
検出なし	BON_BO_DOES_NOT_EXIST	BO_DOES_NOT_EXIST
重複する値	BON_VALDUPES	VALDUPES

要求ビジネス・オブジェクトに応答するには、以下の手順を実行します。

1. 要求ペインでビジネス・オブジェクト・インスタンスを選択します。
2. 以下のいずれかを実行します。
 - Client Simulator ビューのツールバーの「正常に応答」ボタンまたは「応答に失敗」ボタンをクリックします。
 - Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「応答」サブメニューを選択して、目的の応答タイプを選択します。

ビジネス・オブジェクトの保管

入力ペインで作成したビジネス・オブジェクト・インスタンスを保管することにより、特定のコネクタ定義をエミュレートするために Client Simulator ビューをオープンしたときにいつでもそのインスタンスを使用できます。「**BO インスタンス**」フィールドからインスタンスを名前を選択でき、新規要求を作成する処理を行う必要はありません。

「**BO インスタンス**」フィールドに現在リストされているビジネス・オブジェクト・インスタンスを保管するには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「編集」>「全ビジネス・オブジェクトを保管」を選択します。

ビジネス・オブジェクトは、テスト・ユニットが定義されるテスト・プロジェクトのディレクトリーの .bos ディレクトリー内のビジネス・オブジェクト・タイプに関する名前のディレクトリーに、拡張子なしで、ビジネス・オブジェクト・インスタンスの名前のファイルに保管されます。このファイルは Test Connector や Map Designer で使用されたのと同じフォーマットで保管され、これらのツールでのテストにも使用することができます。

ビジネス・オブジェクトの削除

Client Simulator ビューからビジネス・オブジェクト・インスタンスを削除するには、Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「編集」>「ビジネス・オブジェクトを削除」を選択します。

ビジネス・オブジェクトのエクスポート

ビジネス・オブジェクト・インスタンスをファイルにエクスポートして、データを適宜アーカイブし、テクニカル・サポートと共用することができます。テスト・データ・ファイルは .bo 拡張子を付けて保管され、Test Connector や Map Designer、また統合テスト環境によるテストで使用できます。

要求ビジネス・オブジェクトのエクスポート

Client Simulator ビューの入力ペインからビジネス・オブジェクト・インスタンスをエクスポートするには、以下の手順を実行します。

1. 「**BO インスタンス**」フィールドでエクスポートするビジネス・オブジェクト・インスタンスを選択します。
2. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「編集」>「**ビジネス・オブジェクトをエクスポート**」を選択します。
3. ファイルを保管するディレクトリーに移動して、「**ファイル名**」フィールドに名前を指定します。
4. 「**保管**」をクリックします。

応答ビジネス・オブジェクトのエクスポート

結果ペインでビジネス・オブジェクト応答を編集するときに、「応答ビジネス・オブジェクト」ウィンドウからビジネス・オブジェクト・インスタンスをエクスポートするには、以下の手順を実行します。

1. 「応答ビジネス・オブジェクト」ウィンドウのツールバーの「**ビジネス・オブジェクトを保管**」をクリックします。
2. ファイルを保管するディレクトリーに移動して、「**ファイル名**」フィールドに名前を指定します。
3. 「**保管**」をクリックします。

ビジネス・オブジェクトのインポート

Map Designer または Test Connector から保管された、または統合テスト環境からエクスポートされたビジネス・オブジェクト・テスト・データ・ファイルをインポートできます。

要求ビジネス・オブジェクトのインポート

Client Simulator ビューの入力ペインからビジネス・オブジェクト・インスタンスをインポートするには、以下の手順を実行します。

1. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「編集」>「**ビジネス・オブジェクトをインポート**」をクリックします。
2. テスト・データ・ファイルに移動してオープンします。

応答ビジネス・オブジェクトのインポート

結果ペインでビジネス・オブジェクト応答を編集するときに、ビジネス・オブジェクト・インスタンスを「応答ビジネス・オブジェクト」ウィンドウにインポートするには、以下の手順を実行します。

1. 「応答ビジネス・オブジェクト」ウィンドウのツールバーの「**ビジネス・オブジェクトをロード**」をクリックします。

2. テスト・データ・ファイルに移動してオープンします。

ビジネス・オブジェクト・インスタンスの比較

Client Simulator ビューでは、同じタイプの 2 つのビジネス・オブジェクトを比較し、値が異なる属性を表示できます。この機能を使用すると、トランザクション実行の異なる時点におけるビジネス・オブジェクトに対する変更を表示させることができます。例えば、サーバーに送信されたビジネス・オブジェクトと、サーバーで処理されてソース・コネクタに戻された後の同じビジネス・オブジェクトを比較できます。2 つのビジネス・オブジェクト・インスタンスを比較するには、以下の手順を実行します。

1. 入力ペインで比較する要求ビジネス・オブジェクト・インスタンスを選択します。
2. 応答ペインで比較する応答ビジネス・オブジェクトを選択します。
3. Client Simulator ビューのタイトル・バーの右端の下矢印をクリックし、「応答」>「ビジネス・オブジェクトを比較」を選択します。

ビジネス・オブジェクト・データ出力のトレース

ビジネス・オブジェクト・データ出力をトランスポートするため CORBA が使用されます。このソリューションは問題を含むため、ビジネス・オブジェクト・データが大きい場合、サーバーは実際のビジネス・オブジェクト・データをローカルにキャッシュに入れ、クライアントにはメタデータのみを通知します。この呼び出しは小さいことから、ビジネス・オブジェクトのインスペクション・ポイントをすばやく更新できます。サーバーにキャッシュされたデータは、サーバーが再始動されるか ITE から要求されたときにクリアされます。

ビジネス・オブジェクト・データのトランスポートは、リモート・サーバーのみが対象です。ローカル・サーバーの場合、ローカル・ファイル・システムの共用はパフォーマンスを最適化するために使用してください。ビジネス・オブジェクトのトレースがオンの場合、ITE はローカル・サーバーにファイル・システムの場所を通知します。サーバーがリモート側で稼働している場合、ファイル・システムの場所は空になります。

BO インспекター・ビューの使用

ビジネス・オブジェクト・トレースを使用する場合、統合テスト環境により、システムがビジネス・オブジェクトを処理する際にビジネス・オブジェクトについての情報が収集されます。統合テスト環境は、ビジネス・オブジェクトがマップで処理され、またコラボレーションで処理された後で、ビジネス・オブジェクトのイメージを取り込みます。

例えば、コネクタがビジネス・オブジェクト要求をコラボレーション・オブジェクトに送信して、そこから宛先コネクタに送信され、さらに宛先コネクタで処理して応答を戻すインターフェースをテストする場合、統合テスト環境は次のビジネス・オブジェクト・データを取り込みます。

- ビジネス・オブジェクト要求を InterChange Server に送信するときに、ソース・コネクタで呼び出されるマップで作成される汎用ビジネス・オブジェクト。

- 宛先コネクタにより呼び出されるマップに対して、入力として指定される汎用ビジネス・オブジェクト。
- ビジネス・オブジェクト応答を InterChange Server に戻すときに、宛先コネクタで呼び出されるマップで作成される汎用ビジネス・オブジェクト。
- 失敗したフローに関連する例外メッセージ。

図 89 に、3 つの入力ビジネス・オブジェクト、2 つの結果ビジネス・オブジェクト、および 1 つの例外メッセージを取り込んだ BO インспекター・ビューを示します。

Object Name	Value	Type
in.50021.bo		
in.50023.bo		
result.50021.bo	Verb:Create	Customer
result.50023.bo	en_US	
exception.50019.txt	22	
CustomerId	11111	
CustomerNumber	AN_CUST8	
CustomerAddress[0]	Verb:CxBlank	CustomerAddress
Locale	en_US	
ObjectId	666666	
AddressLine1	1725 Learning L...	
AddressLine2	Cxignore	
AddressLine3	Cxignore	
AddressLine4	Cxignore	
City	Trenton	
Region	Cxignore	
State	30	
Country	USA	
District	Cxignore	
PhoneInfo[0]	Verb:CxBlank	PhoneInfo
Locale	en_US	
ObjectId	Cxignore	
PhoneType	Fax	
PhoneCountry	Cxignore	
PhoneNumberA...	Cxignore	
PhoneNumber	1 (923) 234-9080	
PhoneExtension	Cxignore	
ObjectEventId	Cxignore	

図 89. ビジネス・オブジェクト・インспекター・ビュー

次のセクションでは、ビジネス・オブジェクト・トレースと BO インспекター・ビューの使用方法を説明します。

ビジネス・オブジェクト・トレースの動作

以下は、ビジネス・オブジェクトがインターフェースで処理されるときの、テスト・ユニット・エディターおよび BO インспекター・ビューでの BO マーカーに関連する動作を説明します。

- ソース・コネクタからビジネス・オブジェクト要求を送信する場合、テスト・ユニット・エディターのコネクタ・アイコンに付加された BO マーカーは緑色になり、BO マーカーの拡大鏡の図の上に数値 1 が表示されます。これは、1 つのビジネス・オブジェクトを処理したことを示します。

この時点でこの BO マーカーの BO インспекター・ビューを表示し、インバウンド・マップで作成された汎用ビジネス・オブジェクトを参照できます。これは、BO インспекター・ビューの左上のドロップダウン・メニューの「in」接頭部で識別されます。

- ビジネス・オブジェクトがインターフェース内の以降のコンポーネント (コラボレーション・オブジェクトなど) で処理され、エラーがない場合、インターフェース内の宛先コネクタに関連するアウトバウンド・マップで処理されます。

これが発生すると、サーバー・コンテキスト・オーバーレイの宛先コネクタ・アイコンに付加された BO マーカーは緑色になり、BO マーカーの拡大鏡の図の上に数値 1 が表示され、これもビジネス・オブジェクトを処理したことを示します。

この時点でこの BO マーカーの BO インспекター・ビューを表示し、アウトバウンド・マップで処理された汎用ビジネス・オブジェクトを参照できます。これは、BO インспекター・ビューの左上のドロップダウン・メニューの「out」接頭部で識別されます。

- この時点で宛先コネクタ内の要求ビジネス・オブジェクトに応答すると、インターフェースは処理を完了します。BO マーカーの拡大鏡の図の下に数値 1 が表示され、宛先コネクタのインバウンド・マップで作成された汎用オブジェクトが処理されたことを示します。

この時点で、最初の BO インспекター・ビューで汎用ビジネス・ビジネス・オブジェクトを表示できます。これは、BO インспекター・ビューの左上のドロップダウン・メニューの「result」接頭部で識別されます。同じ BO インспекター・ビューには「in」ビジネス・オブジェクト・インスタンスのデータも含まれるので、ドロップダウン・メニューから「result」インスタンスを選択する必要があります。

- インターフェースの実行中にエラーが発生すると、BO マーカーの拡大鏡の周囲に赤色の円が描画されます。BO マーカーに関連する BO インспекター・ビューでエラーを選択できます (「exception」接頭部で識別)。エラー・テキストを表示するには、BO インспекター・ビューをテキスト・ビューに変更する必要があります。

BO インспекター・ビューの表示およびクローズ

次のセクションで、BO インспекター・ビューを表示およびクローズできる多くの方法を説明します。

表 31 の「デフォルトのインターフェース位置」列には、インターフェース内のデフォルト位置が「BO ビューアーの番号」列の BO インспекター・ビューでオーバーレイされるビューをリストします。

表 31. デフォルトの BO インспекター・ビューの位置

BO ビューアーの番号	デフォルトのインターフェース位置
BO ビューアー 1	アウトライン
BO ビューアー 2	プロパティ
BO ビューアー 3	依存関係

BO インспекター・ビューの表示

BO インспекター・ビューを表示するには、次のいずれかの手法を使用します。

- 292 ページの『アウトライン・ビューを使用した BO インспекター・ビューの表示およびクローズ』の説明に従って、アウトライン・ビューで BO インспекター・ビューを表示します。
- 306 ページの『BO インспекター・ビューの表示』の説明に従って、テスト・ユニット・ビューで BO インспекター・ビューを表示します。
- メニュー・バーから「統合テスト環境」>「すべての BO ビューを表示」を選択して、3 つの BO インспекター・ビューをすべて表示します。
- メニュー・バーから「パースペクティブ」>「ビューの表示」>「x: BO インспекター」を選択して特定の BO インспекター・ビューを起動します。ここで、x は、表示する BO インспекター・ビュー・インスタンスの番号です。

BO インспекター・ビューのクローズ

BO インспекター・ビューをすべてクローズするには、次のいずれかの手法を使用します。

- 292 ページの『アウトライン・ビューを使用した BO インспекター・ビューの表示およびクローズ』の説明に従って、アウトライン・ビューで BO インспекター・ビューをクローズします。
- メニュー・バーから「統合テスト環境」>「すべての BO ビューを閉じる」を選択して、オープンしている BO インспекター・ビューをすべてクローズします。
- BO インспекター・ビューのタイトル・バーの右側の「閉じる」ボタンをクリックするか、または BO インспекター・ビューのタイトル・バーを右マウス・ボタン・クリックして、コンテキスト・メニューから「閉じる」を選択します。

ビジネス・オブジェクト・データの最新表示

インターフェースのコンポーネントがビジネス・オブジェクトを処理すると、テスト・ユニット・エディター内の BO マーカーのカウンターが更新されます。これは、その BO マーカーに関連した BO インспекター・ビューでビジネス・オブジェクト・データを更新して、作成された最新のビジネス・オブジェクトを表示できることを示します。

BO インспекター・ビューでビジネス・オブジェクト・データを更新するには、以下のいずれかを実行します。

- BO インспекター・ビューのツールバーで「BO データの最新表示」ボタンをクリックします。
- BO インспекター・ビューのタイトル・バーの右端の下矢印をクリックし、「最新表示」を選択します。

ビジネス・オブジェクト・データを最新表示すると、BO インспекター・ビューの左上角のドロップダウン・メニューから最新のビジネス・オブジェクト・インスタンスを選択できます。

ビジネス・オブジェクト・マーカの最新表示

サーバー・コンテキスト・オーバーレイ内の BO マーカの詳細については、307 ページの『BO マーカの最新表示』を参照してください。

BO インспекター・ビュー・スタイル

BO インспекター・ビューで利用できるビュー・スタイルは、いくつかあります。

テーブル・ビュー・スタイルの使用

テーブル・ビュー・スタイルを使用すると、ビジネス・オブジェクト・データが、Client Simulator ビューの入力ペインまたは「応答ビジネス・オブジェクト」ウィンドウでの表示と同じように、テーブル形式で表示されます。

326 ページの図 89 に、テーブル・ビュー・スタイルの BO インспекター・ビューを示します。

テーブル・ビュー・スタイルを使用するには、以下のいずれかを実行します。

- BO インспекター・ビューのツールバーで「**テーブル・ビューを表示**」ボタンをクリックします。
- BO インспекター・ビューのタイトル・バーの右端の下矢印をクリックし、「**テーブル・ビューを表示**」を選択します。

テキスト・ビュー・スタイルの使用

テキスト・ビュー・スタイルを使用すると、ビジネス・オブジェクト・データが、テキスト・マークアップ形式で表示されます。Client Simulator ビューまたは「応答ビジネス・オブジェクト」ウィンドウからエクスポートした場合は、この形式で保管されます。

330 ページの図 90 に、テキスト・ビュー・スタイルでの BO インспекター・ビューを示します。

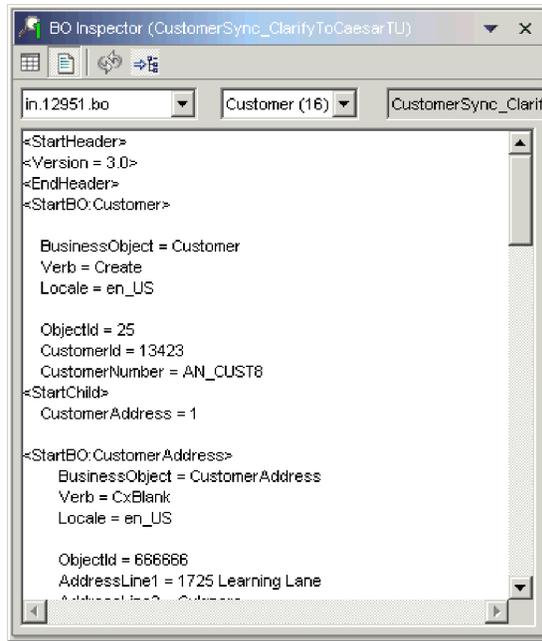


図 90. テキスト・ビュー・スタイルでビジネス・オブジェクト・データが表示された BO インспекター・ビュー

BO インспекター・ビューで例外メッセージを表示するには、テキスト・ビュー・スタイルに変更する必要があります。BO インспекター・ビューは、テーブル・ビュー・スタイルで例外メッセージを表示できません。

図 91 に、テキスト・ビュー・スタイルでの BO インспекター・ビュー内の例外メッセージを示します。

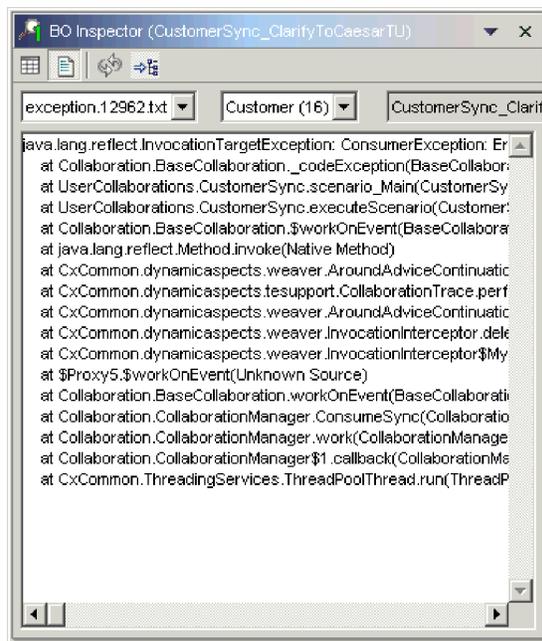


図 91. テキスト・ビュー・スタイルで例外情報が表示された BO インспекター・ビュー

テキスト・ビュー・スタイルを使用するには、以下のいずれかを実行します。

- BO インспекター・ビューのツールバーで「テキスト・ビューを表示」ボタンをクリックします。
- BO インспекター・ビューのタイトル・バーの右端の下矢印をクリックし、「テキスト・ビューを表示」を選択します。

ビューを縦方向に配置

テーブル・ビューの下にテキスト・ビューを同時に表示するように、BO インспекター・ビューを配置できます。

BO インспекター・ビュー・ペインを縦方向に配置するには、BO インспекター・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「縦方向に配置」を選択します。

図 92 に、縦方向に配置された BO インспекター・ビュー・ペインを示します。

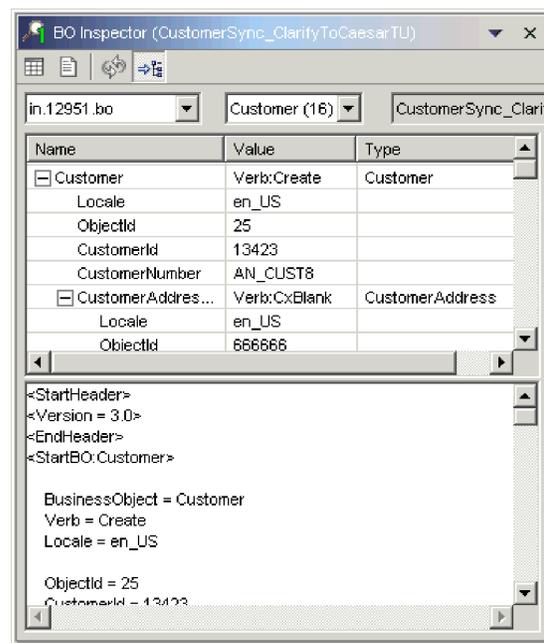


図 92. 縦方向に配置された BO インспекター・ビュー・ペイン

ビューを横方向に配置

テーブル・ビューの右側にテキスト・ビューを同時に表示するように、BO インспекター・ビューを配置できます。

BO インспекター・ビュー・ペインを横方向に配置するには、BO インспекター・ビューのタイトル・バーの右端の下矢印をクリックし、メニューから「横方向に配置」を選択します。

332 ページの図 93 に、横方向に配置された BO インспекター・ビュー・ペインを示します。

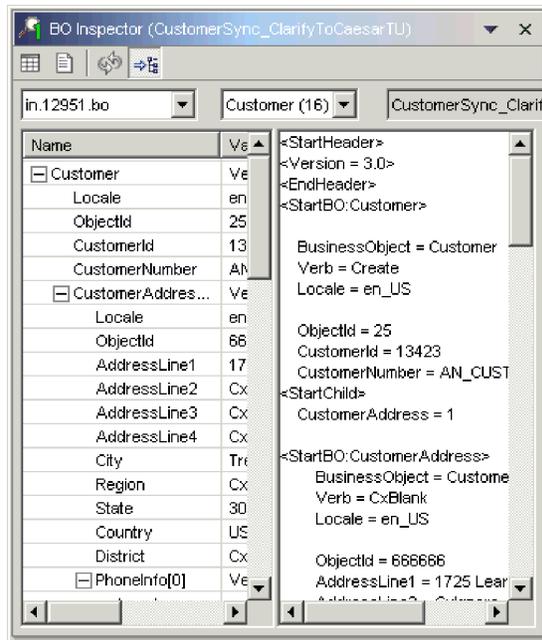


図 93. 横方向に配置された BO インспекター・ビュー・ペイン

BO インспекター・ビューからのテスト・データの保管

BO インспекター・ビュー内のテスト・データを保管する組み込みメカニズムはありませんが、次の次善策を使用することができます。

1. BO インспекター・ビューをテキスト・ビュー・スタイルに切り替えます。詳しくは、329 ページの『テキスト・ビュー・スタイルの使用』を参照してください。
2. テキスト・ビュー・ペインですべてのテキストを選択します。
3. キーボード・ショートカット **Ctrl+C** を使用してテキストをコピーします。
4. コピーされたテキストをテキスト・ファイルに貼り付け、それを **.bo** 拡張子を付けて保管します。

保管したテキスト・ファイルを、Map Designer、Test Connector、および統合テスト環境で使用できます。

ビジネス・オブジェクト・トレース設定の構成

ビジネス・オブジェクト・トレースは、大量のリソースを使用するタスクです。デフォルトでは、2 時間後にオフになるように設定されています。ただし、このタイムアウト値は変更できます。さらに、大量のトランザクションの場合にリソースを多く使用するタスクとなる、ビジネス・オブジェクトの処理で BO マーカーのカウンターを自動的に更新しないように、統合テスト環境を構成することができます。ビジネス・オブジェクト・トレースに関連する設定を行うには、以下の手順を実行します。

1. メニュー・バーから「**ウィンドウ**」>「**設定**」を選択します。

2. 「統合テスト環境」を展開して、「テスト・サーバー」を選択します。
3. 「BO トレース」ペインで以下の手順を実行して、ビジネス・オブジェクト・トレースの設定を行います。
 - a. 開始後ビジネス・オブジェクト・トレースをオフにする時間 (分) を、「BO トレースのタイムアウト」フィールドに入力します。
 - b. ビジネス・オブジェクトの処理で BO マーカーのカウンターを自動的に更新するように統合テスト環境を構成するには、「自動更新を使用可能にする」チェック・ボックスを使用可能のままにしておきます。
4. 「OK」をクリックします。

統合テスト環境を使用したテストの実行

この章のほとんどでは、統合テスト環境パースペクティブのテストまたは特定のインターフェース・エレメントの一部として実行するサブタスクを説明しますが、このセクションでは、テストを実行するために、通常従うワークフローを説明します。ここではほとんどのサブタスクとインターフェース・エレメントが関連します。サブタスクを完了するのに複数の方法がある場合は、最も効果的で効率的な手法を推奨します。統合テスト環境を使用してインターフェースをテストするには、以下の手順を実行します。

1. 統合テスト環境を使用して InterChange Server インスタンスにコンポーネントを配置できますが、次の理由により、前もってすべての配置アクティビティを実行することをお勧めします。
 - テスト・プロセスの一部として、マップやコラボレーション・テンプレートをコンパイルする必要がないようにします。
 - テスト段階の前にコンポーネントも開始できるようにします。コンポーネントは配置されていないと開始できません。コネクタを配置する場合、コネクタを開始するにはサーバーを再始動する必要があります。ほとんどすべてのインターフェースはコネクタに関係があるので、インターフェースのコンポーネントをテスト・プロセスの一部として配置するのは、通常、効率的ではありません。
 - 複数インターフェースをテストする必要がある場合、テスト前に一度配置して、テスト時に各インターフェースが適切に配置されることを監視しなくても済むようにできます。

System Manager を使用したコンポーネントの配置の詳細については、70 ページの『サーバーへのコンポーネントの展開』を参照してください。

repos_copy を使用したコンポーネントの配置の詳細については、143 ページの『第 6 章 repos_copy の使用』を参照してください。

統合テスト環境を使用したコンポーネントの配置 (これはお勧めしません) の詳細については、以下のいずれかを参照してください。

- 289 ページの『アウトライン・ビューを使用したリポジトリの管理』
- 293 ページの『タスク・マネージャー・ビューの使用』
- 305 ページの『テスト・ユニット・ビューを使用したリポジトリの管理』

2. インターフェースのテストに必要なすべてのコンポーネントがアクティブ状態であることを確認します。

コンポーネントを開始するには、次のインターフェースのいずれかを使用します。

- System Monitor (「システム管理ガイド」で説明)
 - InterChange Server コンポーネント管理ビュー (85 ページの『リポジトリ内のコンポーネントの状態の管理』で説明)
 - アウトライン・ビュー (289 ページの『アウトライン・ビューを使用したコンポーネント状態の管理』で説明)
 - 305 ページの『テスト・ユニット・ビューを使用したコンポーネント状態の管理』
3. テストに使用する InterChange Server を、「ローカル・テスト・サーバー」として登録します。詳しくは、35 ページの『InterChange Server インスタンスの登録』を参照してください。
 4. 統合テスト環境パースペクティブを始動します。詳しくは、274 ページの『統合テスト環境の始動』を参照してください。
 5. テストに使用するサーバーを選択します。詳しくは、277 ページの『サーバー構成の選択』を参照してください。

使用するサーバー・インスタンスがダイアログにリストされない場合、サーバー・インスタンス・ビューから削除して再登録してください。

6. テスト・ユニットを含むテスト・プロジェクトを作成します。詳細については、282 ページの『テスト・プロジェクトの作成』を参照してください。
7. テストするインターフェースのテスト・ユニットを作成します。詳細については、283 ページの『統合テスト環境でのテスト・ユニットの作成』を参照してください。
8. 統合テスト環境を使用してテストするインターフェース内にコンポーネントを配置する場合は、以下の手順を実行します。
 - 287 ページの『ユーザー従属オブジェクトの追加』で説明したように、インターフェースにユーザーの従属オブジェクトを追加します。
 - 278 ページの『設計モードで始動するための InterChange Server の構成』の説明に従います。
9. IBM Java オブジェクト・リクエスト・ブローカーが始動していることを確認します。詳しくは、「システム・インストール・ガイド (Windows 版)」または「システム・インストール・ガイド (UNIX 版)」を参照してください。
10. タスク・マネージャー・ビューを使用して、サーバーの始動、統合テスト環境エージェントのサーバーへのバインド、および統合テスト環境のサーバーへの接続を行います。詳しくは、293 ページの『タスク・マネージャー・ビューの使用』を参照してください。
11. サーバー・コンテキスト・オーバーレイを使用可能にします。詳しくは、304 ページの『サーバー・コンテキスト・オーバーレイを使用可能および使用不可にする』を参照してください。
12. インターフェース内のクライアントの Client Simulator ビューを表示します。

ユーザーに適した方式に Client Simulator ビューを編成することをお勧めします。例えば、パースペクティブの位置 1 にソース・コネクタを配置し (統合テスト環境ナビゲーター・ビューと共用)、パースペクティブの位置 4 に宛先コネクタのビューを配置 (プロパティ・ビューと共用) するのが最も容易である場合があります。

インターフェースの Client Simulator ビューを表示および編成するには、次のいずれかを実行します。

- 「コネクタ・ビューの開始」タスクを実行し (詳しくは、294 ページの表 25 を参照)、ビューを必要な位置までドラッグ・アンド・ドロップします。
 - 306 ページの『テスト・ユニット・ビューを使用した Client Simulator ビューの使用、表示、およびクローズ』の説明に従って、各コネクタ定義に使用する Client Simulator ビューを選択します。
13. 311 ページの『サーバーへの Client Simulator ビューの接続』の説明に従って、Client Simulator ビューをサーバーに接続します。

クライアントがサーバーに正常に接続したことを確認します。詳しくは、311 ページの『サーバーへのクライアントの接続の確認』を参照してください。

クライアントがサーバーに正常に接続したことを確認後、ソース・コネクタの Client Simulator ビューでは入力ペインを使用するように構成し (詳しくは 312 ページの『入力ペインの使用』を参照)、宛先コネクタの Client Simulator ビューでは結果ペインを使用するように構成します (詳しくは 313 ページの『結果ペインの使用』を参照)。

14. ビジネス・オブジェクト・トレースを使用する場合は、この時点で開始して、次ステップでビジネス・オブジェクトの送信を開始したらデータを取り込むようにします。ビジネス・オブジェクト・トレース・タスクの開始の詳細については、294 ページの表 25 を参照してください。
15. アウトライン・ビューを使用して、インターフェースのテストの準備ができていることを確認します。詳しくは 290 ページの『アウトライン・ビューを使用したテスト作動可能の検証』を参照してください。
16. ソース・コネクタからビジネス・オブジェクト要求を作成および送信するには、以下を実行します。
- a. ビジネス・オブジェクト・インスタンスを作成して、要求として送信します。詳しくは、317 ページの『要求ビジネス・オブジェクトの作成』を参照してください。
 - b. ビジネス・オブジェクト・インスタンスの属性の値を設定します。詳しくは 320 ページの『ビジネス・オブジェクト属性の値の設定』を参照してください。
 - c. ファイルにビジネス・オブジェクト・インスタンスを保管し、以後のテストで使用できるようにします。詳しくは、323 ページの『ビジネス・オブジェクトの保管』を参照してください。
 - d. ビジネス・オブジェクト・インスタンスを要求として送信します。詳しくは、318 ページの『要求ビジネス・オブジェクトの非同期送信』かまたは 318 ページの『要求ビジネス・オブジェクトの同期送信』の適切な方を参照してください。

17. InterChange Server コンソール・ビューを使用して、ビジネス・オブジェクトの処理を監視します。詳しくは、301 ページの『統合テスト環境コンソール・ビューおよび InterChange Server コンソール・ビューの使用』を参照してください。
18. 各種コンポーネントが処理を完了する際にビジネス・オブジェクトを調べるには、以下を実行します。
 - インターフェース内の BO マーカーの BO インスペクター・ビューを表示します。詳しくは、328 ページの『BO インスペクター・ビューの表示』を参照してください。
 - BO インスペクター・ビューでビジネス・オブジェクト・インスタンスを調べます。詳しくは、329 ページの『テーブル・ビュー・スタイルの使用』を参照してください。
19. 宛先 Client Simulator ビューの結果ペイン内の応答ビジネス・オブジェクトを編集します。詳しくは、322 ページの『応答ビジネス・オブジェクトの編集』を参照してください。
20. 応答としてビジネス・オブジェクト応答を送信します。詳しくは、322 ページの『応答ビジネス・オブジェクトの送信』を参照してください。
21. ステップ 16 (335 ページ) から 20 まで繰り返して、インターフェースを再度テストするか、またはステップ 6 (334 ページ) から 20 まで繰り返して別のインターフェースをテストします。

第 13 章 Collaboration Debugger の使用

Collaboration Debugger は、ワークベンチ・パースペクティブであり、これを使用すると、実行中のコラボレーション・プロセスを一時停止するポイントを指定したり、このポイントで処理されるフローのデータをインスペクションできます。

コラボレーションをデバッグするには、コラボレーション・オブジェクトへの Collaboration Debugger の接続、ブレークポイントの構成、コラボレーションによるフローの進行の管理、およびフローのデータのインスペクションなど、多数のタスクを実行する必要があります。以下に、Collaboration Debugger の使用についての特性を示します。

- 一部のタスクは、1 回しか実行する必要はありません。例えば、コラボレーション・オブジェクトへの Collaboration Debugger の接続は、通常、1 回のみです。
- 通常は、幾つかのタスクを複数回実行します。例えば、フローの進行は、通常、フローの処理の過程で、複数回にわたって管理します。
- ただし、各フローの処理時に実行しないタスクもあります。

コラボレーションのデバッグは、性質上、非直線的であるため、本章のセクションでは、パースペクティブのインターフェースや実行する特定のタスクについて、インターフェースの使用頻度またはタスクの実行頻度の高い順に説明します。359 ページの『コラボレーションのデバッグ』では、コラボレーションのデバッグのワークフローを示し、インターフェースやタスクを直線的に説明します。

この章を構成するセクションは次のとおりです。

- 『Collaboration Debugger の始動』
- 339 ページの『Collaboration Debugger インターフェース』
- 345 ページの『Collaboration Debugger の接続および切り離し』
- 347 ページの『コラボレーション・テンプレート・エディターの使用』
- 350 ページの『イベントの処理』
- 351 ページの『ブレークポイントの処理』
- 355 ページの『デバッグ操作の実行』
- 358 ページの『変数の処理』
- 359 ページの『コラボレーションのデバッグ』

Collaboration Debugger の始動

Collaboration Debugger を始動するには、以下の手順を実行します。

1. 「スタート」>「プログラム」>「IBM WebSphere InterChange Server」>「IBM WebSphere Business Integration Toolset」>「管理」>「System Manager」を選択します。
2. メニュー・バーから「ウィンドウ」>「パースペクティブを開く」>「その他」を選択します。

3. パースペクティブのリストから「コラボレーション・デバッガー」を選択し、「OK」をクリックします。

ワークベンチが始動し、表示されます。340 ページの図 95 に、Collaboration Debugger のパースペクティブを示します。このインターフェースおよびエレメントについては、339 ページの『Collaboration Debugger インターフェース』で説明します。

Collaboration Debugger ビューの設定の構成

Collaboration Debugger のインターフェースが、設計に従って現在使用しているパースペクティブまたは Collaboration Debugger のいずれかで開かれるように、Collaboration Debugger を構成できます。現在使用しているパースペクティブでインターフェースを開くことの利点は、コンポーネントの配置やイベントの送信などのタスクの実行中、Collaboration Debugger ビューを使用するパースペクティブ間をナビゲートする必要がないという点です。Collaboration Debugger でインターフェースを開くことの利点は、ワークベンチ内で同じ位置を共用する必要のあるインターフェースの数を削減できるという点です。

Collaboration Debugger ビューの設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「コラボレーション・デバッガー」を選択します。

339 ページの図 94 に、Collaboration Debugger のビュー設定を示します。

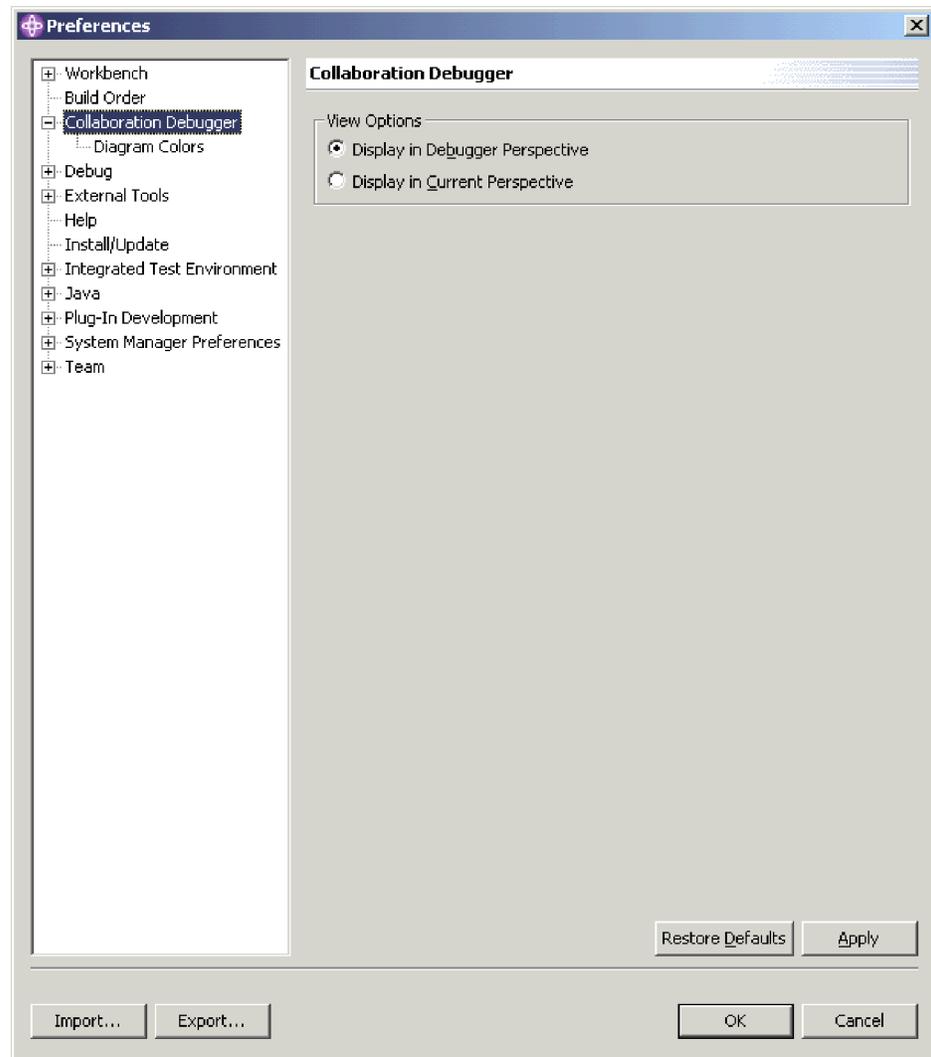


図 94. Collaboration Debugger ビューの設定

3. Collaboration Debugger 自体で Collaboration Debugger ビューを開くことができるようにするには、「デバッガー・パースペクティブでの表示 (Display in Debugger Perspective)」ラジオ・ボタンを有効にします。

現在アクティブとなっているパースペクティブで Collaboration Debugger ビューを開くことができるようにするには、「現在のパースペクティブでの表示 (Display in Current Perspective)」ラジオ・ボタンを有効にします。

4. 「OK」をクリックします。

Collaboration Debugger インターフェース

Collaboration Debugger パースペクティブをデフォルト構成で開いた場合、このパースペクティブには幾つかのビューと 1 つのエディターが表示されます。340 ページの図 95 に、デフォルトの Collaboration Debugger パースペクティブを示します。

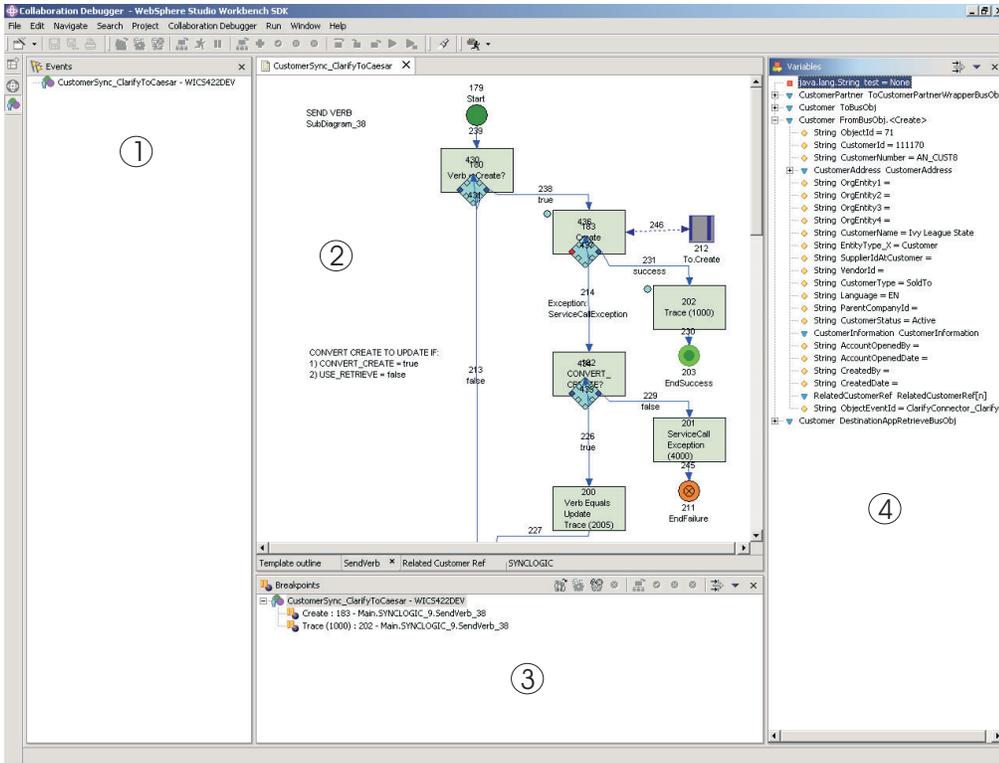


図 95. Collaboration Debugger パースペクティブ

340 ページの表 32 に、Collaboration Debugger パースペクティブのインターフェース・エレメントを示します。番号は、図 95 と対応しています。

表 32. Collaboration Debugger パースペクティブのインターフェース・エレメント

インターフェース・エレメント番号	インターフェース・エレメント名
1	『イベント・ビュー』
2	341 ページの『コラボレーション・テンプレート・エディター』
3	342 ページの『ブレークポイント・ビュー』
4	343 ページの『変数ビュー』

イベント・ビュー

「イベント」ビューには、デバッグするコラボレーション・オブジェクトに対する処理用に作成されたフローがリストされます。このビューには、フローの処理に影響したり、現在実行中のノードを突き止めるための操作が示されます。

341 ページの図 96 に、「イベント」ビューを示します。

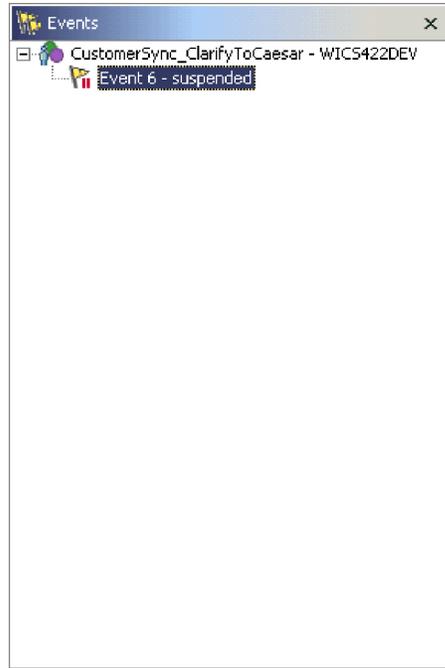


図 96. イベント・ビュー

イベントの処理および「イベント」ビューの使用については、350 ページの『イベントの処理』を参照してください。

コラボレーション・テンプレート・エディター

「コラボレーション・テンプレート」エディターは、読み取り専用のエディターであり、デバッグするコラボレーション・オブジェクトのベースとなっているコラボレーション・テンプレートのダイアグラムが表示されます。このエディターを使用すると、ビジネス・プロセスの設計を表示したり、コラボレーションのデバッグに関連したタスクの多くを実行できます。

342 ページの図 97 に、「コラボレーション・テンプレート」エディターを示します。

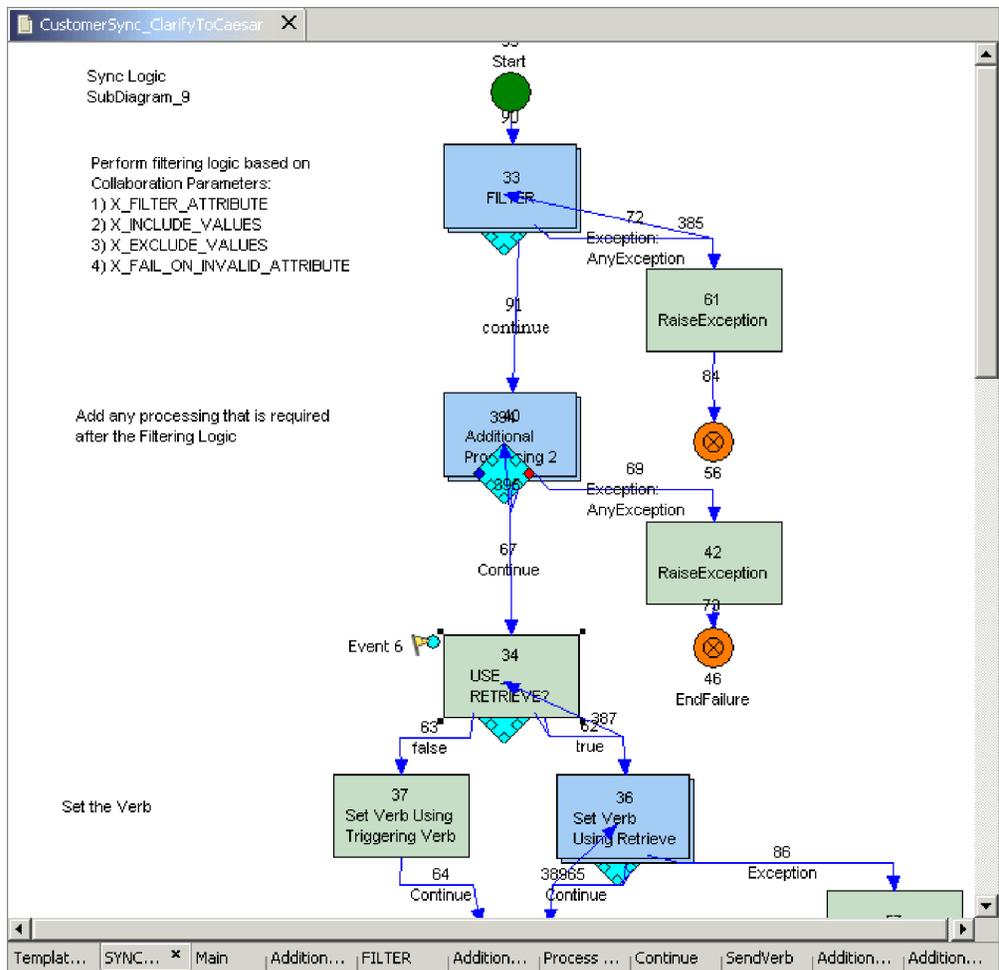


図 97. コラボレーション・テンプレート・エディター

以下のセクションには、「コラボレーション・テンプレート」エディター、およびこのエディターを使用して実行できるタスクについての情報が含まれます。

- 347 ページの『コラボレーション・テンプレート・エディターの使用』
- 351 ページの『ブレイクポイントの処理』
- 355 ページの『デバッグ操作の実行』

ブレイクポイント・ビュー

「ブレイクポイント」ビューを使用すると、コラボレーション・オブジェクト内のブレイクポイントを管理したり、ブレイクポイントとの相互作用時にフローの処理を管理できます。

343 ページの図 98 に、「ブレイクポイント」ビューを示します。

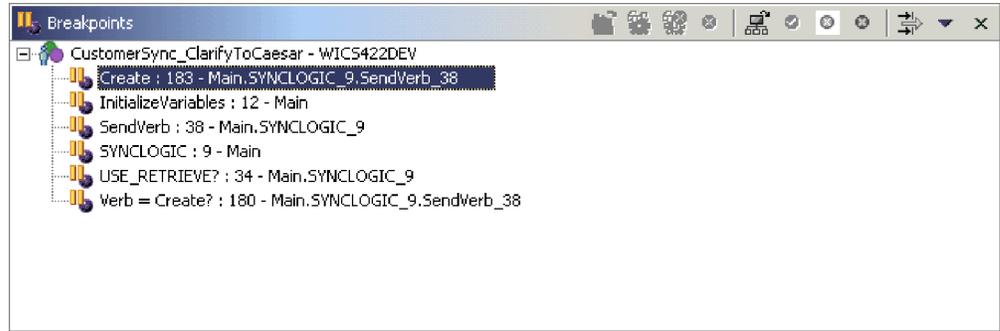


図 98. ブレークポイント・ビュー

一般に「ブレークポイント」ビューを使用して実行するタスクについて詳しくは、以下のセクションを参照してください。

- 351 ページの『ブレークポイントの処理』
- 355 ページの『デバッグ操作の実行』

変数ビュー

「変数」ビューには、デバッグ操作に応答して一時停止したときに処理されるフローについての情報が表示されます。このビューには、コラボレーション・テンプレートの「宣言」セクションで宣言された Java プリミティブ型または `BusObj` の変数がすべてリストされます。デバッグ操作を実行し、コラボレーションを介してフローを管理すると、「変数」ビューの変数が、その時点で実行されているノードでの値に更新されます。

344 ページの図 99 に、「変数」ビューを示します。

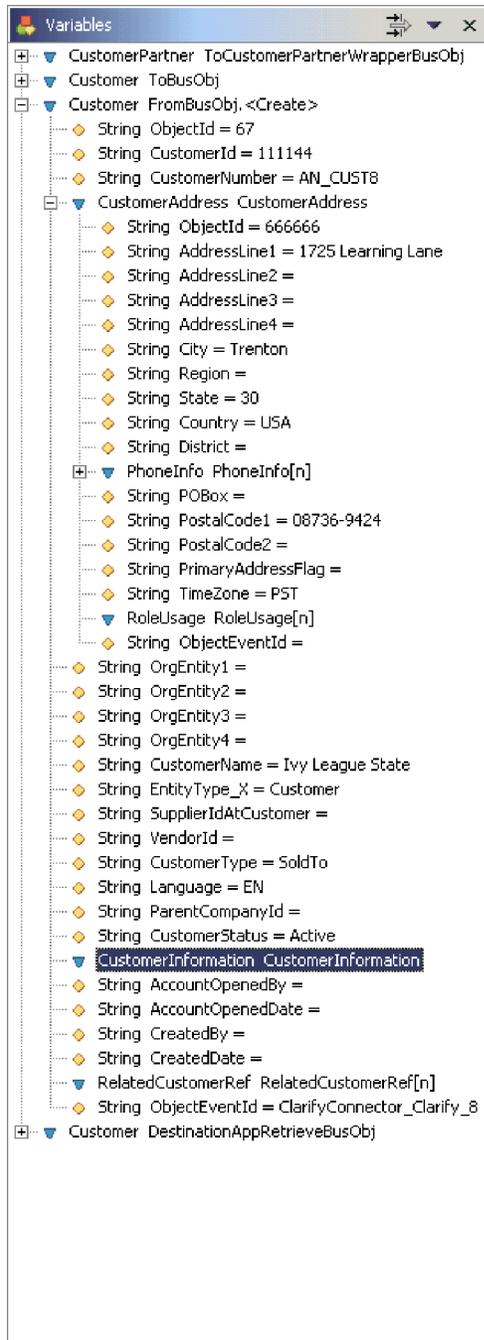


図 99. 変数ビュー

「変数」ビューおよびビューの内容に影響するタスクについて詳しくは、以下のセクションを参照してください。

- 350 ページの『イベントの処理』
- 355 ページの『デバッグ操作の実行』
- 358 ページの『変数の処理』

Collaboration Debugger の接続および切り離し

デバッグ・セッションの開始、ブレイクポイントの設定、およびビジネス・プロセスによるフローの管理を行うには、Collaboration Debugger をコラボレーション・オブジェクトに接続する必要があります。

Collaboration Debugger をコラボレーション・オブジェクトに接続するには、コラボレーション・オブジェクトがアクティブである必要があります。コンポーネント状態、およびコンポーネント状態の変更方法について詳しくは、「システム管理ガイド」を参照してください。

コラボレーション・オブジェクトのデバッグが完了したら、Collaboration Debugger を切り離して、フローの標準処理を再開します。コラボレーション・オブジェクトを停止すると、Collaboration Debugger は自動的にコラボレーション・オブジェクトから切り離されます。

Collaboration Debugger の接続

以下のセクションで説明するように、Collaboration Debugger は、System Manager または統合テスト環境のどちらを使用しても、コラボレーション・オブジェクトに接続できます。

- 『System Manager による Collaboration Debugger の接続』
- 『統合テスト環境による Collaboration Debugger の接続』

System Manager による Collaboration Debugger の接続

System Manager を使用して Collaboration Debugger をコラボレーション・オブジェクトに接続するには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、コラボレーション・オブジェクトが存在するサーバー・インスタンスを展開し、「コラボレーション・オブジェクト」フォルダーを展開します。
2. Collaboration Debugger の接続先となるコラボレーション・オブジェクトを選択します。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。
 - **Shift** を押したまま、連続する項目を選択します。
 - **Ctrl** を押したまま、連続しない項目を選択します。
3. 選択したコラボレーション・オブジェクトのいずれかを右マウス・ボタンでクリックし、コンテキスト・メニューから「**デバッガーを接続**」を選択します。

統合テスト環境による Collaboration Debugger の接続

統合テスト環境を使用して Collaboration Debugger をコラボレーション・オブジェクトに接続するには、テスト・ユニット・ビューでコラボレーション・オブジェクトのセンター・アイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「**デバッガーを接続**」を選択します。

テスト・ユニット・ビューについて詳しくは、303 ページの『テスト・ユニット・ビューの使用』を参照してください。

Collaboration Debugger による Collaboration Debugger の接続

Collaboration Debugger 自体を使用して Collaboration Debugger をコラボレーション・オブジェクトに接続するには、次のいずれかを実行します。

- 「コラボレーション・テンプレート」エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**デバッガーを接続**」を選択します。
- 「ブレークポイント」ビューまたは「イベント」ビューでコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「**デバッガーを接続**」を選択します。
- 「ブレークポイント」ビューまたは「イベント」ビューでコラボレーション・オブジェクトを選択し、メニュー・バーから「**コラボレーション・デバッガー**」>「**デバッガーを接続**」を選択します。
- 「ブレークポイント」ビューでコラボレーション・オブジェクトを選択し、「ブレークポイント」ビューのツールバーの「**デバッガーを接続**」をクリックします。
- 「ブレークポイント」ビューでコラボレーション・オブジェクトを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「**デバッガーを接続**」を選択します。

Collaboration Debugger の切り離し

以下のセクションで説明するように、Collaboration Debugger は、System Manager または統合テスト環境のどちらを使用しても、コラボレーション・オブジェクトから切り離すことができます。

- 『System Manager による Collaboration Debugger の切り離し』
- 『統合テスト環境による Collaboration Debugger の切り離し』

System Manager による Collaboration Debugger の切り離し

System Manager を使用して Collaboration Debugger をコラボレーション・オブジェクトから切り離すには、以下の手順を実行します。

1. InterChange Server コンポーネント管理ビューで、コラボレーション・オブジェクトが存在するサーバー・インスタンスを展開し、「コラボレーション・オブジェクト」フォルダーを展開します。
2. Collaboration Debugger を切り離すコラボレーション・オブジェクトを選択します。例えば、展開する複数のユーザー・プロジェクトを一度に選択するには、以下のように、Windows の標準的な選択方法を使用できます。
 - **Shift** を押したまま、連続する項目を選択します。
 - **Ctrl** を押したまま、連続しない項目を選択します。
3. 選択したコラボレーション・オブジェクトのいずれかを右マウス・ボタンでクリックし、コンテキスト・メニューから「**デバッガーを切り離す**」を選択します。

統合テスト環境による Collaboration Debugger の切り離し

統合テスト環境を使用して Collaboration Debugger をコラボレーション・オブジェクトから切り離すには、テスト・ユニット・ビューでコラボレーション・オブジェ

クトのセンター・アイコンを右マウス・ボタンでクリックし、コンテキスト・メニューから「デバッガーを切り離す」を選択します。

テスト・ユニット・ビューについては、303ページの『テスト・ユニット・ビューの使用』を参照してください。

Collaboration Debugger による Collaboration Debugger の切り離し

Collaboration Debugger 自体を使用して Collaboration Debugger をコラボレーション・オブジェクトから切り離すには、次のいずれかを実行します。

- 「コラボレーション・テンプレート」エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「デバッガーを切り離す」を選択します。
- 「ブレークポイント」ビューまたは「イベント」ビューでコラボレーション・オブジェクトを右マウス・ボタンでクリックし、コンテキスト・メニューから「デバッガーを切り離す」を選択します。
- 「ブレークポイント」ビューまたは「イベント」ビューでコラボレーション・オブジェクトを選択し、メニュー・バーから「コラボレーション・デバッガー」>「デバッガーを切り離す」を選択します。
- 「ブレークポイント」ビューでコラボレーション・オブジェクトを選択し、「ブレークポイント」ビューのツールバーの「デバッガーを切り離す」をクリックします。
- 「ブレークポイント」ビューでコラボレーション・オブジェクトを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「デバッガーを切り離す」を選択します。

Collaboration Debugger から切り離されたコラボレーション・オブジェクトの削除

Collaboration Debugger から切り離したコラボレーション・オブジェクトをすべて削除し、ビューに表示されないようにするには、以下の手順を実行します。

- 「イベント」ビューまたは「ブレークポイント」ビュー内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「切り離したすべてを除去」を選択します。
- 「ブレークポイント」ビューのツールバーの「切り離したすべてを除去」をクリックします。
- 「ブレークポイント」ビューのタイトル・バーの右上端のドロップダウン矢印をクリックし、「切り離したすべてを除去」を選択します。

コラボレーション・テンプレート・エディターの使用

「コラボレーション・テンプレート」エディターは、読み取り専用のエディターであり、デバッグするコラボレーション・オブジェクトのベースとなっているコラボレーション・テンプレートのダイアグラムが表示されます。このエディターを使用すると、ビジネス・プロセスの設計を表示したり、コラボレーションのデバッグに関連したタスクの多くを実行できます。

「コラボレーション・テンプレート」エディターを使用すると、ダイアグラムを使用した作業 (詳しくは『ダイアグラムのオープンおよびクローズ』を参照) やノードのラベルや ID を使用した作業 (詳しくは『ノードの ID およびラベルの表示および非表示』を参照) が可能になります。

「コラボレーション・テンプレート」エディター固有のタスクについて説明した上記のセクションのほかに、以下のセクションでは、複数のインターフェースを使用して実行可能なタスクについて説明します。

- 347 ページの『コラボレーション・テンプレート・エディターの使用』
- 351 ページの『ブレイクポイントの処理』
- 355 ページの『デバッグ操作の実行』

ダイアグラムのオープンおよびクローズ

コラボレーション・テンプレートで、サブダイアグラムおよびイテレーター・ノードに属するノードのブレイクポイントを管理するには、ノードのダイアグラムをオープンする必要があります。

サブダイアグラムまたはイテレーター・ノードのダイアグラムをオープンするには、次のいずれかを実行します。

- ノードを選択して、**Enter** キーを押します。
- ノードをダブルクリックします。
- ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ダイアグラムを開く**」を選択します。

サブダイアグラムまたはイテレーター・ノードのダイアグラムをクローズするには、ダイアグラムのタブの右上端にある **X** をクリックします。

ノードの ID およびラベルの表示および非表示

コラボレーション・テンプレート・ダイアグラムでノードおよびリンクに割り当てられた固有の ID およびラベルを使用すると、ノードやリンクの識別、およびテンプレートで定義されたビジネス・プロセスの認識が簡単になります。ID やラベルを Collaboration Debugger で表示することにより、トラブルシューティング作業を容易化できます。

ダイアグラムに固有 ID を表示するには、エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**uniqueID を表示**」を選択します。

ダイアグラムで固有 ID を非表示にするには、エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**uniqueID を隠す**」を選択します。

ダイアグラムにラベルを表示するには、エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**ラベルを表示**」を選択します。

ダイアグラムに固有 ID を表示するには、エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**ラベルを表示**」を選択します。

コラボレーション・テンプレート・エディターの設定の構成

コラボレーション・テンプレート・エディターの設定を構成するには、以下の手順を実行します。

1. ワークベンチのメニュー・バーから「ウィンドウ」>「設定」を選択します。
2. 「コラボレーション・デバッガー」を展開します。
3. 「ダイアグラムの色」を選択します。

図 100 に、コラボレーション・テンプレート・ダイアグラムの色の設定を示します。

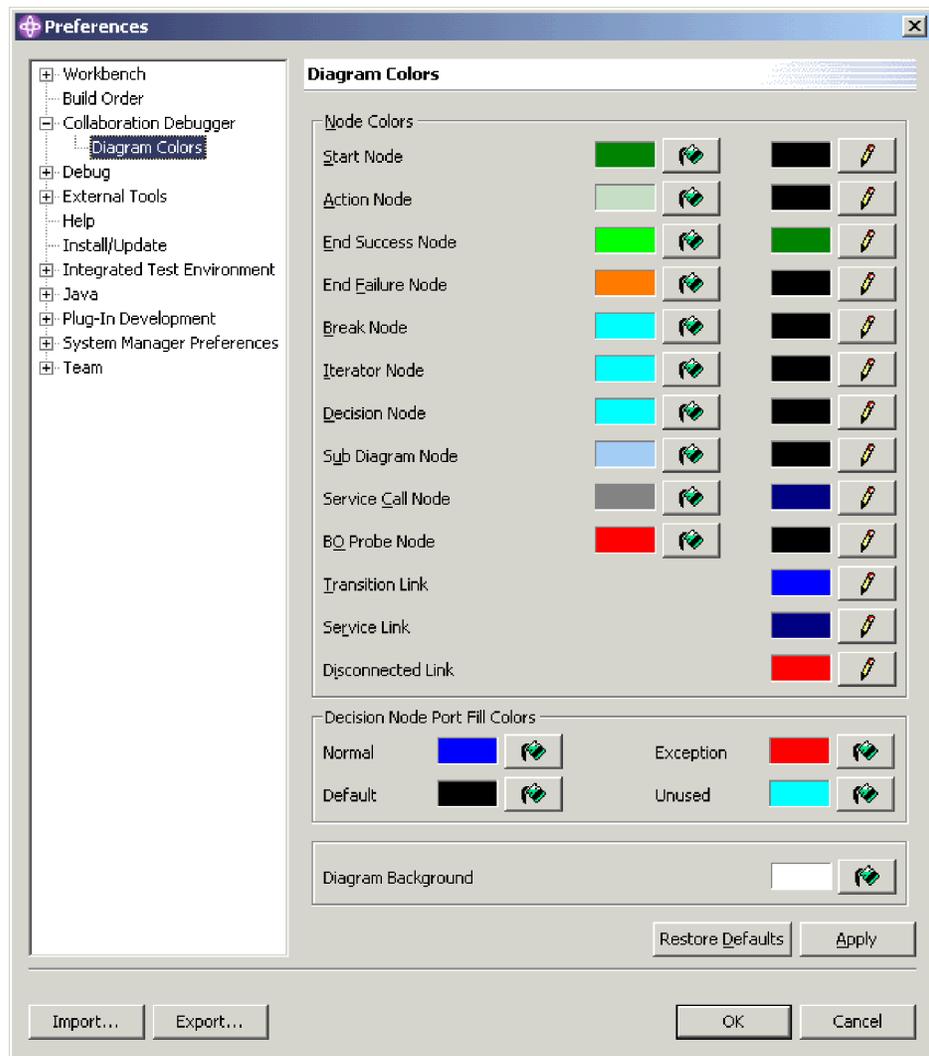


図 100. コラボレーション・テンプレート・ダイアグラムの色の設定

4. 各種のインターフェース・エレメントに関連付けられたペイント・バケツ・アイコンをクリックして、それぞれのエレメントに任意の塗りつぶす色を設定します。

各種のインターフェース・エレメントに関連付けられたペンシル・アイコンをクリックして、それぞれのエレメントに任意の線の色を設定します。

「色」ダイアログから、テスト・ユニット・ビューでコンポーネント・タイプの背景に使用する色を選択し、「OK」をクリックします。

5. 「OK」をクリックします。

イベントの処理

イベントは、コラボレーション・オブジェクトの実行を起動するフローです。コラボレーションでは、エンタープライズ・ソフトウェア・リソース間でイベント・データを処理したり、データを調整しながら、ビジネス・プロセス・ロジックを実行してイベント・データを管理します。ビジネス・ロジックにおける重大なポイントでデータをインスペクションできるように、イベントの処理を一時停止するには、Collaboration Debugger を使用するのが一般的です。

イベントの表示

デバッグするコラボレーション・オブジェクトに対する処理用の新しいフローを配信すると、新しいフローには番号が割り当てられ、「イベント」ビューのコラボレーション・オブジェクトの下にリストされます。

図 101 に、コラボレーション・オブジェクトによって受信され、「イベント」ビューに表示されたイベントを示します。



図 101. イベント・ビューに表示されたイベント

イベントを表示したら、「コラボレーション・テンプレート」エディターで、処理が一時停止されたノードに Collaboration Debugger のフォーカスを与えます。フォーカスを与えると、選択されていることを示す黒色の正方形がノードの各隅に表示されます。ノードの左上隅には、イベント・インスタンスでラベル付けされたフラグ・アイコンが表示されます。図 102 に、コラボレーション・テンプレート・エディターに表示される、イベントの処理が一時停止した状態のノードを示します。

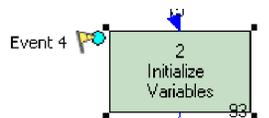


図 102. コラボレーション・テンプレート・エディターに表示されるイベント

コラボレーション内に設定されたアクティブのブレークポイントがある場合、フローは、デバッグ操作の実行を待機する一方で処理を中断し、処理が失敗したノードに同様のイベントを表すフラグ・アイコンをドローします。

イベントを表示することにより、コラボレーション・ダイアグラムにイベントを配置し、「変数」ビューにフローの現在のデータを取り込みます。

「変数」ビューは、デバッグ操作を実行すると動的に更新されるため、通常は以下の状態でのみ、イベントの表示を必要とします。

- コラボレーション・オブジェクトが処理を必要とするフローを最初に受信したとき
- (ブレイクポイントの表示などのために) Collaboration Debugger の任意の場所をナビゲートしたため、イベントを再度配置する必要がある場合

イベントを表示するには、以下のいずれかを実行します。

- 「イベント」ビューでイベントを右マウス・ボタンでクリックし、コンテキスト・メニューから「表示」を選択します。
- 「イベント」ビューでイベントを選択し、メニュー・バーから「コラボレーション・デバッガー」>「表示」を選択します。

イベントの中断

357 ページの『イベントの実行』の説明に従ってイベントを実行すると、ブレイクポイントで一時停止されることなく、標準的な実行のフローが進行します。フローのデバッグを再開する場合は、イベントを中断します。

いったん Collaboration Debugger によるイベントの実行を指定すると、処理が迅速に行われるため通常は中断できません。ただし、サービス呼び出しを使用するとイベントを中断できます。サービス呼び出しを使用してビジネス・オブジェクトを送信すると、ビジネス・プロセスは応答が受信されるまでブロックされます。これにより、要求へ応答する前にフローを中断する十分な時間を確保できるため、コラボレーションが応答を受信したときにデバッグを再開できます。

イベントを中断するには、以下のいずれかを実行します。

- 「イベント」ビューでイベントを右マウス・ボタンでクリックし、コンテキスト・メニューから「中断」を選択します。
- 「イベント」ビューでイベントを選択し、メニュー・バーから「コラボレーション・デバッガー」>「中断」を選択します。

イベントの実行

イベントの実行について詳しくは、357 ページの『イベントの実行』を参照してください。

ブレイクポイントの処理

ブレイクポイントは、Collaboration Debugger がコラボレーション・オブジェクトの実行を一時停止するために使用する命令です。ビジネス・プロセスで、フローのデータをインスペクションするノードにブレイクポイントを設定すると、変換を監視したり、問題をトラブルシューティングできます。

ブレイクポイントを設定できるノードのタイプは以下のとおりです。

- アクション・ノード
- サブダイアグラム・ノード
- イテレーター・ノード
- 終了成功ノード
- 終了障害ノード

- イテレーター・ノード
- ブレーク・ノード

ブレークポイントは決定ノードには設定できません。

コラボレーション・テンプレートに表示される各種のノードについて詳しくは、「[コラボレーション開発ガイド](#)」を参照してください。

以下のセクションでは、デバッグするコラボレーションでのブレークポイントの配置方法について説明します。

- 『ブレークポイントの設定』
- 『ブレークポイントの表示』
- 353 ページの『ブレークポイントの削除』
- 353 ページの『ブレークポイントの使用不可化』
- 354 ページの『ブレークポイントの使用可能化』

コラボレーション・オブジェクトにブレークポイントを配置したら、355 ページの『[デバッグ操作の実行](#)』の説明に従って、フローの処理を管理します。

ブレークポイントの設定

サポートされたノードにブレークポイントを設定するには、以下のいずれかを実行します。

- ノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ブレークポイントを設定**」を選択します。
- ノードを選択し、Collaboration Debugger のツールバーの「**ブレークポイントを設定**」をクリックします。
- ノードを選択し、メニュー・バーから「**コラボレーション・デバッガー**」>「**ブレークポイントを設定**」を選択します。
- まだブレークポイントが設定されていないノードをダブルクリックします。

ブレークポイントの表示

ブレークポイントが設定されたノードを選択してフォーカスを与えるには、以下のいずれかを実行します。

- 「ブレークポイント」ビューでブレークポイントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**表示**」を選択します。
- 「ブレークポイント」ビューでブレークポイントをダブルクリックします。
- 「ブレークポイント」ビューでブレークポイントを選択し、**Enter** キーを押します。
- 「ブレークポイント」ビューでブレークポイントを選択し、「ブレークポイント」ビューのツールバーの「**表示**」をクリックします。
- 「ブレークポイント」ビューでブレークポイントを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「**表示**」を選択します。
- 「ブレークポイント」ビューでブレークポイントを選択し、Collaboration Debugger のツールバーの「**表示**」をクリックします。

- 「ブレークポイント」ビューでブレークポイントを選択し、メニュー・バーから「コラボレーション・デバッガー」>「ブレークポイントを表示する」を選択します。

ブレークポイントの削除

ブレークポイントを削除するには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ブレークポイントを削除**」を選択します。
- 「ブレークポイント」ビューでブレークポイントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**削除**」を選択します。
- 「ブレークポイント」ビューでブレークポイントを選択し、「ブレークポイント」ビューのツールバーの「**削除**」をクリックします。
- 「ブレークポイント」ビューでブレークポイントを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「**削除**」を選択します。
- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを選択し、Collaboration Debugger のツールバーの「**ブレークポイントを削除**」をクリックします。
- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを選択し、メニュー・バーから「**コラボレーション・デバッガー**」>「**ブレークポイントを削除**」を選択します。
- ブレークポイントが設定されたノードをダブルクリックします。

ブレークポイントの使用不可化

ブレークポイントを使用不可にするには、以下のいずれかを実行します。これを行うと、今後指定したノードで実行が一時停止されなくなります。ただし、必要な場合は再度アクティブにできます。

- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**ブレークポイントを使用不可にする**」を選択します。
- 「ブレークポイント」ビューでブレークポイントを右マウス・ボタンでクリックし、コンテキスト・メニューから「**使用不可**」を選択します。
- 「ブレークポイント」ビューでブレークポイントを選択し、「ブレークポイント」ビューのツールバーの「**使用不可**」をクリックします。
- 「ブレークポイント」ビューでブレークポイントを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「**使用不可**」を選択します。
- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを選択し、Collaboration Debugger のツールバーの「**ブレークポイントを使用不可にする**」をクリックします。
- 「コラボレーション・テンプレート」エディターでブレークポイントが設定されたノードを選択し、メニュー・バーから「**コラボレーション・デバッガー**」>「**ブレークポイントを使用不可にする**」を選択します。

ブレイクポイントの使用可能化

使用不可となっているブレイクポイントを使用可能にするには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディターでブレイクポイントが設定されているノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「ブレイクポイントを使用可能にする」を選択します。
- 「ブレイクポイント」ビューでブレイクポイントを右マウス・ボタンでクリックし、コンテキスト・メニューから「使用可能」を選択します。
- 「ブレイクポイント」ビューでブレイクポイントを選択し、「ブレイクポイント」ビューのツールバーの「使用可能」をクリックします。
- 「ブレイクポイント」ビューでブレイクポイントを選択し、タイトル・バーの右上端のドロップダウン矢印をクリックして、「使用可能」をクリックします。
- 「コラボレーション・テンプレート」エディターでブレイクポイントが設定されたノードを選択し、Collaboration Debugger のツールバーの「ブレイクポイントを使用可能にする」をクリックします。
- 「コラボレーション・テンプレート」エディターでブレイクポイントが設定されたノードを選択し、メニュー・バーから「コラボレーション・デバッガー」>「ブレイクポイントを使用可能にする」を選択します。

ブレイクポイントのフィルタリング

「ブレイクポイント」ビューをフィルタリングすることにより、特定のタイプのノードに設定されたブレイクポイントのみを表示できます。「ブレイクポイント」ビューをフィルタリングするには、以下の手順を実行します。

1. 「ブレイクポイント・フィルター」ダイアログを表示するには、以下のいずれかを実行します。
 - 「ブレイクポイント」ビューのタイトル・バーの「フィルター」を選択します。
 - 「ブレイクポイント」ビューのタイトル・バーの右上端の下矢印をクリックし、ドロップダウン・メニューから「フィルター」を選択します。

Collaboration Debugger によって、359 ページの図 104 に示すような「ブレイクポイント・フィルター」ダイアログが表示されます。

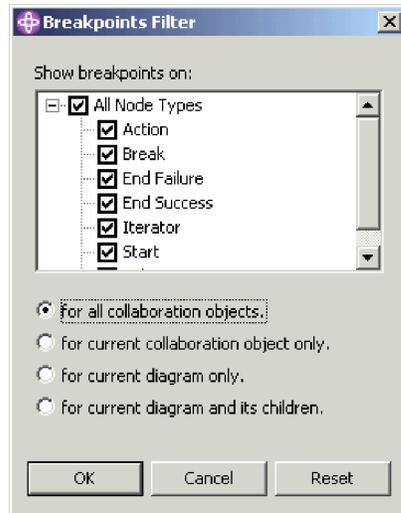


図 103. Breakpoints Filter

2. 以下の方法を使用して、フィルターの設定を構成します。
 - 「すべてのノード・タイプ」チェック・ボックスを有効にし、サポートされるすべてのノード・タイプを表示します。
 - 特定のノード・タイプに対応するチェック・ボックスのみを有効にし、特定のタイプのみを表示します。
 - 「リセット」をクリックし、デフォルトのフィルター設定を適用します。
3. 以下のラジオ・ボタンのいずれかを有効にし、フィルターの適用範囲を指定します。
 - すべてのコラボレーション・オブジェクト
 - 現在のコラボレーション・オブジェクトのみ
 - 現在のダイアグラムのみ
 - 現在のダイアグラムとその子
4. 「OK」をクリックします。

デバッグ操作の実行

デバッグするコラボレーション・オブジェクトにブレークポイントを配置したら、デバッグ操作を実行し、ブレークポイントを介してイベントのフローを管理します。使用可能な操作は以下のとおりです。

- 356 ページの『ステップオーバー』
- 356 ページの『ステップイントゥ』
- 356 ページの『ステップアウト』
- 357 ページの『実行』
- 357 ページの『特定のノードに対する実行』
- 357 ページの『イベントの実行』

ステップオーバー

「ステップオーバー」操作を実行すると、InterChange Server が一時停止されたフローを再開し、ブレークポイントが設定されたノードが実行されます。次のノードに移るとフローは一時停止されます。

実行が失敗したノードをステップオーバーするには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「ステップオーバー」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、ツールバーの「ステップオーバー」をクリックします。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、メニュー・バーから「コラボレーション・デバッガー」>「ステップオーバー」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、キーボード・ショートカットの **F6** キーを使用します。

ステップイントゥ

サブダイアグラムまたはイテレーター・ノードで「ステップイントゥ」操作を実行すると、InterChange Server が一時停止されたフローを再開します。実行は、フローがサブダイアグラムまたはイテレーター内の開始ノードに到達するまで進み、到達すると一時停止します。

実行が失敗したサブダイアグラムまたはイテレーターをステップイントゥするには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「ステップイントゥ」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、ツールバーの「ステップイントゥ」をクリックします。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、メニュー・バーから「コラボレーション・デバッガー」>「ステップイントゥ」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、キーボード・ショートカットの **F5** キーを使用します。

ステップアウト

サブダイアグラムまたはイテレーター内のノードで「ステップアウト」操作を実行すると、InterChange Server が一時停止されたフローを再開します。実行は、サブダイアグラムまたはイテレーターが完了するか、フローがサブダイアグラムまたはイテレーター内のアクティブなブレークポイントに到達するまで進みます。

実行が失敗したサブダイアグラムまたはイテレーターからステップアウトするには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディターの任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**ステップアウト**」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、ツールバーの「**ステップアウト**」をクリックします。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、メニュー・バーから「**コラボレーション・デバッガー**」>「**ステップアウト**」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、キーボード・ショートカットの **F7** キーを使用します。

実行

「実行」操作を実行すると、InterChange Server が一時停止されたフローを再開して、ビジネス・プロセスが完了するか、アクティブなブレイクポイントが設定されたノードに到達するまで、処理の継続が可能になります。

一時停止されたフローを実行するには、以下のいずれかを実行します。

- 「コラボレーション・テンプレート」エディター内の任意の場所を右マウス・ボタンでクリックし、コンテキスト・メニューから「**実行**」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、ツールバーの「**実行**」をクリックします。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、メニュー・バーから「**コラボレーション・デバッガー**」>「**実行**」を選択します。
- 「コラボレーション・テンプレート」エディターにフォーカスを与え、キーボード・ショートカットの **F8** キーを使用します。

特定のノードに対する実行

フローで「このノードに接続」操作を実行すると、InterChange Server が一時停止されたフローを再開して、操作を実行したノードまたはアクティブなブレイクポイントが設定されたノードに到達するまで、処理の継続が可能になります。

特定のノードに対して一時停止されたフローを実行するには、以下のいずれかを実行します。

- 実行の対象となるノードを右マウス・ボタンでクリックし、コンテキスト・メニューから「**このノードに接続**」を選択します。
- 実行の対象となるノードを選択し、ツールバーの「**選択先まで実行**」をクリックします。
- 実行の対象となるノードを選択し、メニュー・バーから「**コラボレーション・デバッガー**」>「**選択先まで実行**」を選択します。

イベントの実行

「イベントを実行する」操作を実行すると、InterChange Server が一時停止されたフローを再開して、処理の継続が可能になるため、ブレイクポイントがアクティブであっても、ビジネス・プロセスのブレイクポイントをバイパスできます。

イベントを実行するには、以下のいずれかを実行します。

- 「イベント」ビューでイベントを右マウス・ボタンでクリックし、コンテキスト・メニューから「実行」を選択します。
- 「イベント」ビューでイベントを選択し、コンテキスト・メニューから「コラボレーション・デバッガー」>「イベントを実行する」を選択します。
- 「イベント」ビューでイベントを選択し、ツールバーの「イベントを実行する」をクリックします。

変数の処理

「変数」ビューには、コラボレーション・テンプレート定義の「宣言」タブで宣言されたすべての変数の値が表示されます。これには、Process Designer によって自動的に宣言され、「ポートおよびトリガー・イベント」タブで作成した各ポートに関連付けられたビジネス・オブジェクトのほかに、ユーザー自身が「宣言」タブに追加したすべての変数が含まれます。

イベントを表示 (詳しくは、350 ページの『イベントの表示』を参照) するか、フローを管理 (詳しくは、355 ページの『デバッグ操作の実行』を参照) すると、「変数」ビュー内に表示される変数情報が更新されます。BusObj 型の変数を展開すると、その属性を表示できます。

現時点では、triggeringBusObj 変数を表示することはできません。この変数は、コラボレーション・オブジェクトの実行を起動した元のフローを参照する、システムで宣言された変数であり、通常は非常に重要な変数とされています。

triggeringBusObj 変数の値のレコードを保守する場合は、コラボレーション・テンプレートの「宣言」セクションで新しい BusObj 変数を宣言し、それをシナリオの最初のノードで triggeringBusObj 変数の値に初期設定します。

さらに、作成した複数のコラボレーション・テンプレートで、processingBusObj という名前の変数を操作します。ただし、triggeringBusObj 変数はそのままにしておきます。processingBusObj 変数は、通常、テンプレート・レベルではなく、シナリオ・レベルで宣言されます。ただし、「変数」ビューには表示されません。「変数」ビューで processingBusObj 変数を使用可能にするには、変数の宣言をシナリオ定義からテンプレート定義に移動します。

「変数」ビューをフィルタリングすると、特定のタイプの変数のみを表示できます。「変数」ビューをフィルタリングするには、以下の手順を実行します。

1. 「変数フィルター」ダイアログを表示するには、以下のいずれかを実行します。
 - 「変数」ビューのタイトル・バーの「フィルター」をクリックします。
 - 「変数」ビューのタイトル・バーの右上端の下矢印をクリックし、ドロップダウン・メニューから「フィルター」を選択します。

Collaboration Debugger によって、359 ページの図 104 に示すような「変数フィルター」ダイアログが表示されます。

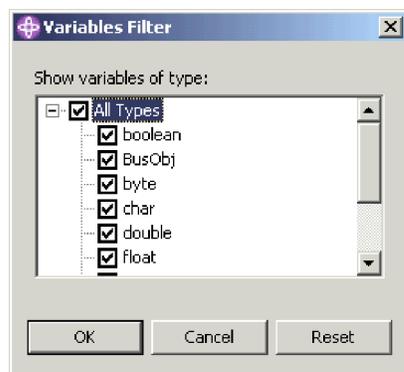


図 104. Variables Filter

2. 以下の方法を使用して、フィルターの設定を構成します。
 - 「すべてのタイプ」チェック・ボックスを有効にし、サポートされるすべての変数タイプを表示します。
 - 特定の変数タイプのチェック・ボックスのみを有効にし、特定のタイプのみを表示します。
 - 「リセット」をクリックし、デフォルトのフィルター設定を適用します。
3. 「OK」をクリックします。

コラボレーションのデバッグ

以下のワークフローをガイドラインとして使用し、コラボレーション・オブジェクトを効率的にデバッグします。

1. コラボレーション・テンプレートを作成します。作成したテンプレートに基づいてコラボレーション・オブジェクトを作成します。それら両方を InterChange Server インスタンスに配置して、コラボレーション・オブジェクトが開始されたことを確認します。
2. Collaboration Debugger パースペクティブを始動します。詳しくは、337 ページの『Collaboration Debugger の始動』を参照してください。
3. Collaboration Debugger をコラボレーション・オブジェクトに接続します。詳しくは、345 ページの『Collaboration Debugger の接続』を参照してください。
4. 目的のロケーションにブレークポイントを設定します。詳しくは、351 ページの『ブレークポイントの処理』を参照してください。

通常、フローが変換されるノード、フローがコラボレーションから送信されるノード、およびロジックの問題発生部分が特定されているノードにブレークポイントを設定するのが最も便利です。

5. Test Connector または統合テスト環境を使用して、コラボレーション・オブジェクトの起動などのイベントを作成および送信します。Test Connector の詳細については、261 ページの『第 11 章 Test Connector の使用』を参照してください。

統合テスト環境について詳しくは、273 ページの『第 12 章 統合テスト環境の使用』を参照してください。

6. 実行依頼されたイベントを表示します。詳しくは、350 ページの『イベントの表示』を参照してください。
7. 構成した複数のブレイクポイントをまたぐフローを管理します。詳しくは、355 ページの『デバッグ操作の実行』を参照してください。
8. 「変数」ビューを使用して、コラボレーションによるイベントの処理時に、イベント変更のデータを監視します。詳しくは、358 ページの『変数の処理』を参照してください。
9. ブレイクポイントを変更し、異なるデータを持つフローを処理するには、ステップ 4 (359 ページ) からステップ 8 を繰り返します。

第 14 章 パフォーマンスの調整

この章では、さまざまな技法や構成を実装してシステム・パフォーマンスを向上させる方法について説明します。この章は次のセクションから構成されます。

- 『イベントにより起動されるフローの並行処理の実装』
- 363 ページの『コネクタ・エージェントによる要求の並行処理の実装』
- 366 ページの『コネクタ・エージェントの配布』
- 367 ページの『静的関係のキャッシング』
- 367 ページの『データベース接続プールの使用』
- 367 ページの『メモリー・チェッカー・スレッドの使用』

イベントにより起動されるフローの並行処理の実装

コラボレーション・オブジェクトおよびコネクタ・コントローラーは、イベントにより起動される複数のフローを同時に処理するよう構成できます。これによって、イベントにより起動されるインターフェースのパフォーマンスが大幅に向上します。

イベントにより起動されるフローのコラボレーションでの並行処理

コラボレーションは、イベントにより起動される複数のフローを並行して処理するように構成できます。並行イベント処理を適切に行うと、システムのスループットおよびイベント処理への応答時間が改善されます (下のヒントを参照)。デフォルトでは、コラボレーションはイベントにより起動されるフローを一度に 1 つずつ処理します。

コラボレーションがイベントにより起動されるフローを並行処理しているとき、コラボレーションはこれらのフローの依存関係を識別し、それらがコネクタ・コントローラーから送信された順に処理します。イベントにより起動されるフローの並行処理は、データの競合がないフローで実行されますが、データの競合があるフローは、受信された順に処理されます。

コラボレーションがイベントにより起動される複数のフローを処理するよう構成するには、235 ページの『並行イベントの最大数』を参照してください。

ヒント: イベントにより起動されるフローをコラボレーションで並行処理するには、追加のシステム・リソースが必要です。最高のパフォーマンスを実現するためには、並行イベントの処理に使用するシステム・リソースが確実にアイドル状態ではないようにします。例えば、コラボレーション・キュー内のイベント数が 10 に達することがない場合は、「**並行イベントの最大数**」の値を 10 に設定しないでください。

コラボレーションのインバウンド・ポートが、Access Interface を介する外部呼び出しを受信するためだけにバインドされていて、他のコネクタにバインドされていない場合、「**並行イベントの最大数**」の値を 0 に設定す

るとパフォーマンスを向上させることができます。しかし、コラボレーションがコネクタとの双方向交換に使用されている場合は、この値を 0 に設定しないでください。

イベントにより起動されるフローのコラボレーション・オブジェクト・グループでの並行処理

コラボレーション・オブジェクト・グループ内の各コラボレーションは、イベントにより起動される多数の並行フローを処理するために個々に構成できます。グループ内のすべてのコラボレーションには、イベントにより起動される並行フローの数を同じ値に設定して、並行率の低いコラボレーションがボトルネックにならないようにすることをお勧めします。

イベントにより起動されるフローのコネクタ・コントローラーでの並行処理

コネクタ・コントローラーは、イベントにより起動される複数のフローを並行して処理するように構成できます。

コネクタがイベントにより起動されるフローを並行処理するよう構成すると、イベント・フローの処理パフォーマンスが向上します。これは、複数のビジネス・オブジェクトを同時にマッピングに変換できるためです。

コネクタ・コントローラーがイベントにより起動されるフローを並行処理するよう構成するには、「**ConcurrentEventTriggeredFlows**」プロパティを、並行処理するフローの最大数に設定します。このプロパティの詳細については、191 ページの『**ConcurrentEventTriggeredFlows**』を参照してください。Connector Configurator を使用してコネクタ・プロパティを設定する方法の詳細については、159 ページの『第 7 章 コネクタの構成』の概要を参照してください。

ヒント: 「**ConcurrentEventTriggeredFlows**」プロパティが 1 より大きい値に設定されている場合、コネクタ・コントローラーはアプリケーションからイベントを受け取ったのと同じ順序を保持します。イベントにより起動されるフローをコネクタで並行処理するには、追加のシステム・リソースが必要です。最高のパフォーマンスを実現するためには、並行イベントの処理に使用するシステム・リソースが確実にアイドル状態ではないようにします。例えば、コネクタの InterChange Server で受け取るイベント数が 10 に達することがない場合は、「**ConcurrentEventTriggeredFlows**」プロパティの値を 10 に設定しないでください。「サーバー統計」ウィンドウを使用して、コネクタの MQ キュー項目数をモニターすることにより、コネクタ・キュー内のイベント数を判別します。この統計をモニターすると、「**ConcurrentEventTriggeredFlows**」プロパティの値を設定するのに役立ちます。

コネクタが宛先としてのみ使用されている場合、

「**ConcurrentEventTriggeredFlows**」の値を 0 に設定すると、パフォーマンスが向上します。しかし、コネクタがコラボレーションとの双方向交換に使用されている場合は、この値を 0 に設定しないでください。

コネクタ・エージェントによる要求の並行処理の実装

Java でマルチスレッドを使用するように設計されたコネクタ・エージェントは、追加の構成を行わなくてもマルチスレッドの並行処理を実行できます。さらに、マルチスレッド化用に設計されていないコネクタでも、コネクタ・エージェント並列処理 (CAP) と呼ばれる機能を使用するように構成できるものがあります。この機能により、コネクタを複数のコネクタ・プロセスとして実行して、複数の要求を処理することができます。コネクタ・エージェント並列処理は、単一のコネクタ・エージェントの主プロセスから並行する複数プロセスをインスタンス化し、逐次的でなく並行してフローを処理できるようにします。

IBM WebSphere InterChange Server ビジネス・インテグレーション・システムでのコネクタ・エージェント並列処理の使用は、Connector Configurator の「リソース」タブで構成できます。コネクタ・エージェントの並行処理の使用は、一部の環境にはパフォーマンスの向上をもたらしますが、その他の環境にはそうではない場合があるので、コネクタ・エージェントのデフォルトの振る舞いは、シリアル処理を使用するよう設定されています。デフォルト設定を変更する前に、特定のコネクタ用の IBM Business Integration Adapters の資料を調べて、アプリケーションのアーキテクチャーまたは要件によって並行処理の効果的な使用が妨げられていないか分析し、ユーザー固有の環境で使用可能なリソースを最も効果的に使用する方法を検討します。実稼働環境に実装する前に、この設定に対して行った変更内容をテストする必要があります。

コネクタ・エージェント並列処理を使用する理由

一般的に、次の環境ではコネクタ・エージェント並列処理の使用を検討する必要があります。

- コネクタ・エージェントが C++ で作成されている。C++ CDK を使用するコネクタ・エージェントは本質的にシングルスレッドですが、コネクタ・エージェント並列処理を使用して処理を並行して実行するよう構成することができます。
- コネクタ・エージェントは Java で作成されたが、アプリケーション API の呼び出しには同期ロックを保持する必要があるため、Java のマルチスレッド機能を使用できない。このようなコネクタは、コネクタ・エージェント並列処理を使用して処理を並行して実行するよう構成することができます。

コネクタ・エージェント並列処理を使用しない理由

コネクタが上記のカテゴリに分類される場合でも、おそらくコネクタ・エージェント並列処理を使用しないと考えられる環境があります。

- コネクタが Java ベースであり、マルチスレッド操作用に設計されている場合。この場合、マルチスレッドを並行して実行すると (コネクタはコネクタ・エージェント並列処理を使用しなくてもこの操作を実行できます) リソースの消費が少なく済むので、コネクタ・エージェント並列処理を実行しないと考えられます。
- コネクタの機能に関係なく、アプリケーションのアーキテクチャーまたは要件によって並行処理が禁止される場合。例えば TPI コネクタの場合、Cyclone サーバーでインスタンス化できるのは 1 つだけです。

- コネクタがグローバル・リソースを排他的に使用する必要がある場合。この場合、複数のプロセスが並行して同一のグローバル・リソースを排他的に使用しようとするか振る舞いを調べる必要があります。このような状況では、コネクタ・エージェント並列処理を効果的に使用できる場合とそうでない場合があります。

要求を並行処理するためのコネクタ・エージェントの構成

コネクタ・エージェント並列処理の使用の設定を変更するには、Connector Configurator を開始し、「リソース」タブを選択します。図 105 に「リソース」タブを示します。

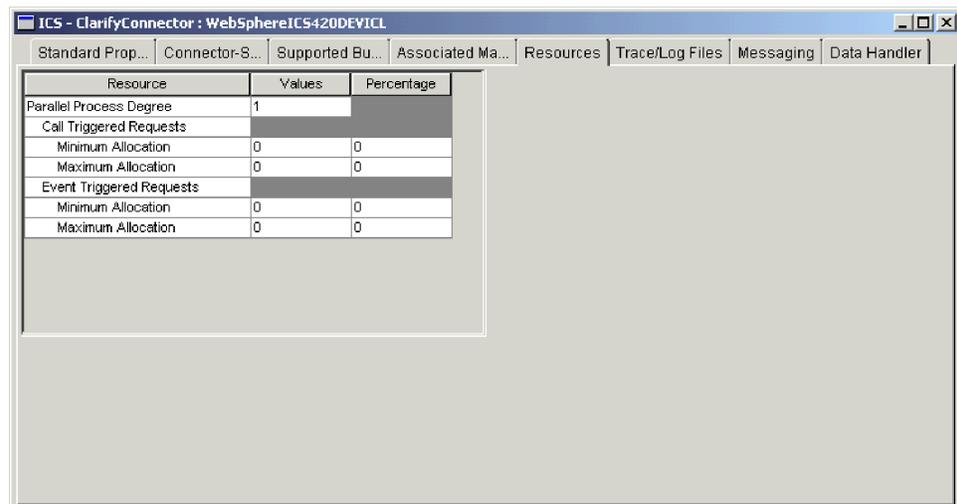


図 105. 「リソース」タブでの値の構成

シリアル処理または並行処理のいずれかの「リソース」タブの値を設定します。

1 からそれより大きい値に変更する場合、または大きい値から 1 に変更する場合、変更を適用するにはコネクタ・エージェントをリポートする必要があります。値がすでに 1 より大きい場合は、1 より大きい別の値に動的に変更できます。

シリアル処理の場合

デフォルトでは、コネクタ定義の「並行処理の度合い」プロパティは値 1 に設定され、シリアル処理を使用するようになっています。「並行処理の度合い」プロパティが 1 に設定されると、ウィンドウ内の他のすべての値は 0 に設定されます。

並行処理の場合

並行処理を使用するようコネクタを構成するには、以下の手順を実行します。

1. 「並列処理の度合い」プロパティを値 2 より大きい値に設定します。コネクタは指定された数のスレーブ・プロセスを並行して実行します。

注: 値はスレーブ・プロセスの数を表します。各コネクタ・エージェントには、ポーリング・プロセス、マスター・プロセス、構成可能な数のスレーブ・プロセスがあります。例えば、値が 2 である場合、コネクタは合計

4 つのプロセスを使用します (コネクターのポーリング・プロセス、コネクターのマスター・プロセス、2 つのスレーブ・プロセス)。

2. 「呼び出しによりトリガーされた要求 (Call Triggered Requests)」プロパティから字下げされた「最小割り振り (Minimum Allocation)」プロパティの「値」列に、呼び出しにより起動されるフロー用に生成するスレーブ・プロセスの最小数を入力します。
3. 「呼び出しによりトリガーされた要求 (Call Triggered Requests)」プロパティから字下げされた「最大割り振り (Maximum Allocation)」プロパティの「値」列に、呼び出しにより起動されるフロー用に生成するスレーブ・プロセスの最大数を入力します。
4. 「イベントによりトリガーされた要求 (Event Triggered Requests)」プロパティから字下げされた「最小割り振り (Minimum Allocation)」プロパティの「値」列に、イベントにより起動されるフロー用に生成するスレーブ・プロセスの最小数を入力します。
5. 「イベントによりトリガーされた要求 (Event Triggered Requests)」プロパティから字下げされた「最大割り振り (Maximum Allocation)」プロパティの「値」列に、イベントにより起動されるフロー用に生成するスレーブ・プロセスの最大数を入力します。

呼び出しにより起動されるフローは同期型で、アクセス・クライアントによって開始されます。アクセス・クライアントは、要求に対して応答が戻されることを期待します。コネクターが呼び出しにより起動されるフローとイベントにより起動されるフローを処理する場合、呼び出しにより起動されるフローにより多くのスレーブ・プロセスを専用化して、開始プログラムができるだけ早く応答を受け取るようにすることもできます。

デフォルト値をそのまま使用することも、各要求タイプに割り当てるプロセスの最小数と最大数を入力することもできます。両方の要求タイプの最大数の合計が、「並行処理の度合い」プロパティの値を超えないようにしてください。また、最小数が最大数を超えることはできません。

コネクター・エージェント並列処理を効果的に使用するには、コネクター・エージェントの「並行処理の度合い」値と、コネクター・エージェントに要求を送信するコラボレーション・オブジェクトの「並行イベントの最大数」値を調整します。1 つのコラボレーションのみがこのコネクター・エージェントに要求を送信する場合、そのコラボレーションの「並行イベントの最大数」値を、コネクター・エージェントの「並行処理の度合い」値以上に設定します。複数のコラボレーションがこのコネクター・エージェントに要求を送信する場合、「並行処理の度合い」値を、そのコラボレーションの「並行イベントの最大数」値の合計と同じ値に設定します。

以下に示すデフォルト以外の Java 仮想マシン (JVM) ヒープ/スタック・サイズのパラメータ値を持つコネクターを使用している場合、コネクターの構成ファイルを編集する必要があります。コネクター構成ファイルに設定した JVM 値は、コネクター・エージェント・マスター・プロセスからスレーブ・プロセスへと伝搬されません。これらの値を編集するには、XML エディターを使用して

InterchangeSystem.cfg ファイルを開くか、ローカル・ファイルを使用している場

合はローカル・コネクタ・エージェントの構成ファイルを開いて、JVM
<connector_name> セクションの下にある値を編集します。

JVM パラメーターおよびそれに対応する Java VM オプションは、次のとおりです。

```
MIN_HEAP_SIZE    (-Xms)  
MAX_HEAP_SIZE    (-Xmx)  
MAX_NATIVE_STACK_SIZE(-Xss)
```

デフォルトでは、次の値が使用されます。

```
MIN_HEAP_SIZE=1m  
MAX_HEAP_SIZE=128m  
MAX_NATIVE_STACK_SIZE=128k
```

コネクタに異なる値を設定する必要がある場合があります。例えば、構成ファイルの JVM セクションに次の値を設定することができます。

```
[JVM SAPConnector]  
MIN_HEAP_SIZE=256m  
MAX_HEAP_SIZE=512m  
MAX_NATIVE_STACK_SIZE=1m
```

コネクタ・エージェントの配布

コネクタ・エージェントを配布することによって、ビジネス・インテグレーション・システムのパフォーマンスを向上させることができます。コネクタ・エージェントの配布によってパフォーマンスが向上する理由は、次のとおりです。

- 各 InterChange Server およびコネクタ・エージェント・プロセスは、自身の Java 仮想マシン (JVM) 内で実行されます。コンピューター上で非常に多くの JVM インスタンスが実行されると、パフォーマンスが低下し始めます。しきい値はコンピューターの仕様によって異なりますが、コネクタ・エージェントを配布することによって、1 つの JVM で InterChange Server をホスティングしているマシンの負荷が減少します。
- 一般的に、個々のコネクタのパフォーマンスは、ネットワーク・トポロジーに関して、コネクタが通信するアプリケーションに近いほど向上します。コネクタ・エージェントは、それ自体がアプリケーション・サーバーをホストしているコンピューター上にインストールするのが理想ですが、環境によってはこのことは不可能な場合があります。ただし、アプリケーション・ホスト・コンピューターと InterChange Server ホスト・コンピューターが異なるサブネット上にある場合でも、アプリケーション・ホスト・コンピューターが存在するのと同じサブネット内の別のコンピューターにコネクタ・エージェントがインストールされていれば、通常、コネクタ・エージェントはよりよいパフォーマンスを実現します。

コネクタ・エージェントを分配する方法については、「*WebSphere Business Integration Adapters* インストール・ガイド」を参照してください。

静的関係のキャッシング

静的関係をキャッシュに入れて、それに含まれる参加者データをメモリーにロードしたり、データベース内に保管されているデータではなく、メモリー内のデータに対してデータの照会を発行することができます。これによって、フローごとにデータベースから情報を検索するよりも、関係のルックアップがかなり速くなります。

キャッシュする静的関係の構成の詳細については、「マップ開発ガイド」を参照してください。

データベース接続プールの使用

フローの処理中に、データベース表から情報を検索したり情報を挿入したりすることはよく見られます。インターフェースでフローごとにデータベースへの接続を確立する必要がある場合、パフォーマンスに影響を及ぼします。これは特に、接続ごとにログイン・プロセスを実行する必要があることが原因です。

データベース接続プールは、データベース・リソースへの接続のプールを確立および保持する統合コンポーネントです。接続が最初に確立されると、他のコンポーネントはその接続を使用してデータベース・リソースに対して照会を発行できます。あるコンポーネントが接続を使用して操作を終了すると、接続は解放されてプールに戻されるので、別のコンポーネントがそれを使用できます。これにより、接続が必要になるたびにログインする必要がなくなります。

データベース接続プールの作成方法の詳細については、207 ページの『第 8 章 データベース接続プールの構成』を参照してください。マップでのデータベース接続プールの使用方法の詳細については、「マップ開発ガイド」を参照してください。コラボレーション・テンプレートでのデータベース接続プールの使用方法の詳細については、「コラボレーション開発ガイド」を参照してください。

メモリー・チェッカー・スレッドの使用

InterChange Server には、メモリー・チェッカー・スレッドが備えられています。これを使用すると、システムによるイベントの処理方法を規定したり、それに伴ってシステム・メモリーの使用量を規定することができます。これにより、メモリー不足が原因でシステムが破損するリスクが軽減します。

メモリー・チェッカー・スレッドは、InterChange Server で使用されるメモリー量を定期的に測定し、それが設定された許容範囲内にあるかどうか評価します。メモリー使用量が許容範囲内でない場合、メモリー・チェッカー・スレッドはシステム・コンポーネントを管理して、メモリー使用量を削減します。

メモリー使用量が指定された下限しきい値を超える場合、メモリー・チェッカー・スレッドは、システム内のコネクタ・コントローラーのイベント・リスナーを構成可能な最大時間の間スリープさせます。時間はメモリー使用量が下限しきい値を超えた度合いによって異なりますが、「しきい値でのコネクタの休止時間 (分)」プロパティに指定された時間を越えません。このようにしてイベント・リスナー・スレッドをスリープさせることにより、メモリー・チェッカー・スレッドはイベントの InterChange Server への送達速度をスローダウンし、上限メモリーしきい値を超えるリスクを軽減します。

メモリー使用量が「メモリーの上限しきい値 (パーセント)」プロパティに指定された上限しきい値を超える場合、メモリー・チェッカー・スレッドはシステム内のコネクタを休止します。コネクタが休止すると、現在キューに入れられているビジネス・オブジェクトの処理は継続されますが、新規イベントはポーリングされません。したがって、コネクタはキューに入れられたビジネス・オブジェクト数を削減するので、新規ビジネス・オブジェクトの処理にさらなるメモリーを使用することなく、使用中のメモリー量を削減します。コネクタは、「しきい値でのコネクタの休止時間」プロパティに指定された最小時間の間、休止します。「しきい値でのコネクタの休止時間」プロパティに指定された時間が経過すると、メモリー・チェッカー・スレッドはメモリー使用量を再度調べます。メモリー使用量が上限しきい値を超えていない場合、メモリー・チェッカー・スレッドはコネクタを再始動します。ただし、メモリー使用量がまだ上限しきい値を超える場合、メモリー・チェッカー・スレッドはコネクタを休止状態のままにします。

メモリー・チェッカー・スレッドは、調査と状態管理操作の合間に、「メモリー・チェッカーのスリープ時間」プロパティに指定された時間の間スリープします。

メモリー・チェッカー・スレッドを使用するには、以下のセクションの説明に従う必要があります。

1. 『CW_MEMORY_MAX パラメーターの設定』
2. 369 ページの『メモリー・チェッカー・スレッド・プロパティの構成』
3. 370 ページの『追加の考慮事項』

CW_MEMORY_MAX パラメーターの設定

InterChange Server 始動スクリプトの Java パラメーター CW_MEMORY_MAX で、InterChange Server が使用可能な最大ヒープ・メモリーを設定する必要があります。メモリー・チェッカー・スレッドは、この最大ヒープ・メモリー量のパーセンテージを指定する構成可能なプロパティ値に基づいてアクションを行います。

CW_MEMORY_MAX パラメーターは、InterChange Server の最大ヒープ・サイズを指定する -mx パラメーターと同じ値に設定します。デフォルトでは、CW_MEMORY_MAX パラメーターは -mx パラメーターと同じ値に設定されますが、-mx パラメーターを変更する場合には CW_MEMORY_MAX パラメーターも同様に変更する必要があります。

以下に、Windows コンピューター上の InterChange Server 用の、変更されていない start_server.bat 始動スクリプトの例を示します。

```
REM This is the -mx param value for the Interchange Server's memory heap
set CW_MEM_HEAP=512
```

```
REM Start the InterChange Server
%CWJAVA% -Djava.ext.dirs=%JRE_EXT_DIRS%;"%MQ_LIB%";"%DB2_LIB%"
-Duser.home="%CROSSWORLDS%" -mx%CW_MEM_HEAP%m -DTEAgent=1200
-DCW_MEMORY_MAX=%CW_MEM_HEAP% %ORB_PROPERTY% -classpath %JCLASSES%
ServerWrapper -s%SERVERNAME% %2 %3
```

CW_MEM_HEAP 変数は値 512 に設定され、この値を使用して -mx パラメーターが設定されるので、512 メガバイトのメモリーが Java ヒープ用に予約されます。

CW_MEMORY_MAX パラメーターも、CW_MEM_HEAP 変数を使用して同じ値に設定されます。CW_MEMORY_MAX パラメーターの先頭には -D が付加されます。

メモリー・チェッカー・スレッド・プロパティの構成

InterChange Server 構成ファイルを編集して、メモリー・チェッカー・スレッドのプロパティを設定する必要があります。これらのプロパティは、InterChange Server 構成インターフェースの「各種」タブに公開されます。InterChange Server の構成については、134 ページの『System Manager を使用してその他のプロパティを構成』を参照してください。「各種」タブの「サーバーのメモリー」ペインで以下の手順を実行して、メモリー・チェッカー・スレッドを構成します。

1. 「メモリー・チェッカーのスリープ時間」フィールドに、メモリー・チェッカー・スレッドが各操作間にスリープする時間を設定します。メモリー・チェッカー・スレッドを使用不可にするには、「メモリー・チェッカーのスリープ時間」フィールドに値 0 (デフォルト値) を設定します。
2. 「メモリーの上限しきい値 (パーセント)」フィールドに、InterChange Server が使用可能な合計メモリー (CW_MEMORY_MAX パラメーターに指定されている) のパーセントを設定します。この値を超えた場合、メモリー・チェッカー・スレッドはコネクタを休止して、メモリー使用量が上限しきい値より下がるまでコネクタが新規イベントをポーリングしないようにします。デフォルト値は 90 です。これは、上限しきい値が CW_MEMORY_MAX パラメーターに指定されている合計メモリーの 90 % であることを意味します。例えば、CW_MEMORY_MAX パラメーターが 512 メガバイトに設定されている場合、上限しきい値はおよそ 460 メガバイトになります。
3. 「メモリーの下限しきい値 (パーセント)」フィールドに、InterChange Server が使用可能な合計メモリー (CW_MEMORY_MAX パラメーターに指定されている) のパーセントを設定します。この値を超えた場合、メモリー・チェッカー・スレッドはコネクタがイベントを InterChange Server に送達する速度を低下させます。デフォルト値は 80 です。これは、下限しきい値が CW_MEMORY_MAX パラメーターに指定されている合計メモリーの 80 % であることを意味します。例えば、CW_MEMORY_MAX パラメーターが 512 メガバイトに設定されている場合、下限しきい値はおよそ 410 メガバイトになります。
4. 「しきい値でのコネクタの休止時間」フィールドに、下限メモリーしきい値を超えたときにコネクタによる InterChange Server へのイベント送達速度が低下される最大時間と、上限メモリーしきい値を超えたときにコネクタが休止を続ける最小時間の両方を指定する時間数 (分単位) を設定します。デフォルト値は 5 です。

370 ページの図 106 に、メモリー・チェッカー・スレッドの構成に使用する「サーバーのメモリー」プロパティの「各種」タブを示します。

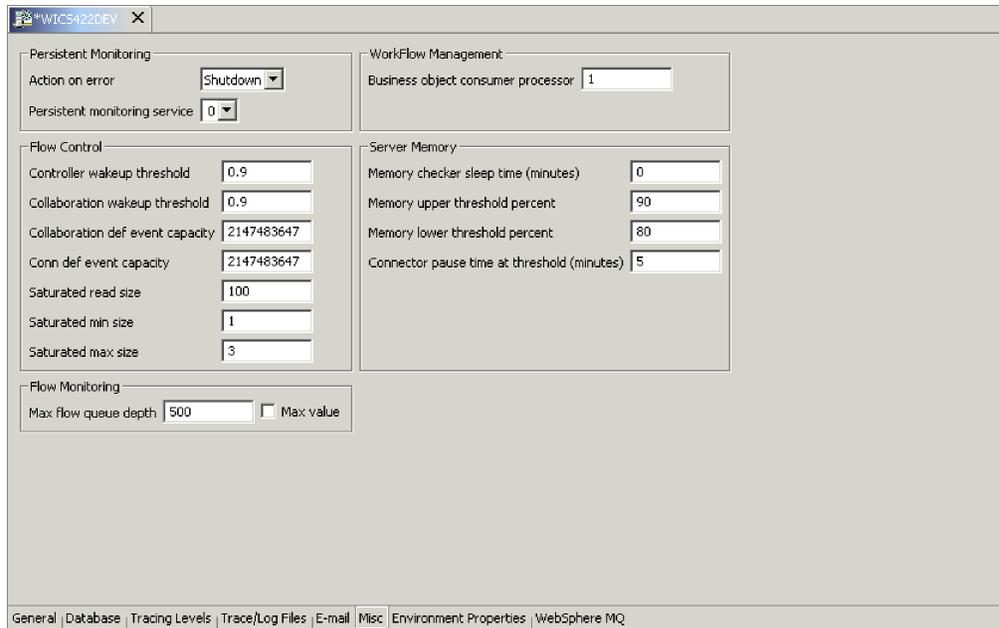


図 106. メモリー・チェッカー・スレッド・プロパティの構成

追加の考慮事項

メモリー・チェッカー・スレッドを使用するには、以下のことを考慮してください。

- 少数のトランザクションを処理する InterChange Server インスタンスを実行している場合は、メモリー・チェッカー・スレッドを使用しないでください。例えば、`-mx` パラメーターを 512 メガバイトより少なく設定している場合、Java 仮想マシンのガーベッジ・コレクション・スレッドはメモリーを効果的に開放することができません。
- 以下の手法を使用して、メモリー・チェッカー・スレッド・プロパティの最善の構成を決定してください。
 - まず、「メモリーの下限しきい値 (パーセント)」および「メモリーの上限しきい値 (パーセント)」プロパティを近い値に設定し、システムの振る舞いをテストしてから、値の差を増やしてもう一度システムをテストします。このプロセスを繰り返して、最適な差を判別します。
 - 「しきい値でのコネクターの休止時間」プロパティを 1 分から 5 分の間に設定します。
 - InterChange Server が大量のワークロードを処理する必要がある場合、「メモリー・チェッカーのスリープ時間」プロパティを小さい数値に設定して、測定および規定アクションをより頻繁に行うようにします。

特記事項

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Burlingame Laboratory Director
IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

以下は、IBM Corporation の商標です。

IBM
IBM ロゴ
AIX
CICS
CrossWorlds
DB2
DB2 Universal Database
IMS
Informix
iSeries
Lotus
Lotus Domino
Lotus Notes
MQIntegrator
MQSeries
MVS
OS/400
Passport Advantage
SupportPac
WebSphere
z/OS

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

MMX および Pentium は、Intel Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。
System Manager およびその他のパースペクティブには、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれます。



IBM WebSphere InterChange Server (5724-178)、バージョン 4.3.0 および IBM WebSphere Business Integration Toolset (5724-177)、バージョン 4.3.0

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アクセス・クライアント 311
アプリケーション固有のビジネス・オブジェクト 109, 112, 113
依存 287
イベント
 実行 351, 357
 中断 351
 表示 350
イベントにより起動されるフロー処理
 コネクタ・コントローラーでの 362
 コラボレーションでの 361
 コラボレーション・オブジェクト・グループでの 362
イベントの実行 351, 357
イベントの処理 350
イベントの中断 351
イベントの表示 350
イベント分離 238
イベント・ビュー 340
インストール済みかつ使用可能であること
 プラグインの確認 23
インストールの推奨事項 16
インターフェース 1
 Collaboration Debugger 339
インターフェース, テスト 117
インターフェース, 配置 117
インターフェースのライフ・サイクル 3
 ローカルでのインターフェースの開発 3
 ローカルでのインターフェースのテスト 4
エディター
 コラボレーション・テンプレート 341
エミュレート
 コネクタ 311

[カ行]

開発
 コラボレーション・テンプレート 116
開発, ローカルでのインターフェースの 3
開発および構成 10
開発タスク 111
開発統合環境
 準備 22
開発統合環境の準備 22
開発統合システムへの配置 4
開発フロー 109
外部ポート・バインディング 230
カスタム・コネクタ, 開発 112
環境
 カスタマイズ 19
 準備 15
環境のカスタマイズ 19
環境の準備 10, 15
環境変数, 構成 136
関係 110
関係 ID による検索
 Relationship Manager 内で 246
関係, 構成 115
関係インスタンスの検索
 Relationship Manager 内で 245
関係インスタンスの個数の取得
 Relationship Manager 内で 248
関係インスタンスの削除
 Relationship Manager 内で 249
関係データの印刷
 Relationship Manager 内で 255
関係データの印刷プレビューの使用
 Relationship Manager 内で 256
関係の構成 115
関係を開く
 Relationship Manager 内で 244
管理, コンポーネント状態 305
管理, リポジトリ 305
キーボード・ショートカット
 Ctrl+Shift+P 256
キー・ポーリング
 使用 21
キー・ポーリングの使用 21
検証 11
検証, データベース接続プール 214
検証, リポジトリ 153
構成
 外部ポート・バインディング 230
 環境変数 136
 コネクタ 21, 117, 159
 コネクタ・エージェント 364
 コラボレーション・オブジェクト 117, 219, 231
 コラボレーション・グループ 220
 コラボレーション・テンプレート・エディター 349

構成 (続き)

 コラボレーション・テンプレート・エディターの設定 349
 サーバー・プロパティ 119
 セキュリティー 137
 セキュリティー・プロパティ 124
 ツール設定 20
 データベース接続プール 115
 データベース・プロパティ 122, 127
 統合テスト環境 302
 トランザクション・ブラケット 217
 トレース・レベル 130
 内部ポート・バインディング 229
 ビジネス・オブジェクト・トレース設定 332
 プライバシー 139
 メモリー・スレッド・プロパティ 369
 要求を処理するためのコネクタ・エージェント 364
 ロギング, 開発環境での 18
 ロギング・プロパティ 131
 ClearCase 接続用の System Manager 23
 Collaboration Debugger 338
 Collaboration Debugger ビューの設定 338
 E メール・プロパティ 133
 WebSphere Interchange Server 119
構成, データベース接続プール 207
個々の開発環境
 準備 15
個々の開発環境の準備 15
コネクタ
 エミュレート 311
 構成 21, 112, 117, 159
 データ・ハンドラー 109
コネクタ・エージェント
 構成 364
 配布 366
 並行処理 363, 364
 並列処理 363
コネクタ・エージェントによる要求の並行処理の実装 363
コネクタ・エージェント並列処理を使用しない理由 363
コネクタ・エージェント並列処理を使用する理由 363
コネクタ・コントローラーでのイベントにより起動されるフローの処理 362

- 子ビジネス・オブジェクト 320
- コラボレーション 216
 - デバッグ 359
- コラボレーションのデバッグ 359
- コラボレーション・エディター
 - ブレイクポイント・ビュー 342
 - 変数ビュー 343
- コラボレーション・オブジェクト 111
 - 構成 117
 - ビュー 226
 - ポート 228
- コラボレーション・オブジェクト、構成 219, 231
- コラボレーション・オブジェクト、作成 220
- コラボレーション・オブジェクト、変更 225
- コラボレーション・オブジェクト・グループ 220
- コラボレーション・オブジェクト・グループ、イベントにより起動されるフロー処理 362
- コラボレーション・グループ
 - 構成 220
- コラボレーション・テンプレート 110, 219
 - エディター 341, 347
 - エディターの設定 349
 - 開発 116
- コラボレーション・テンプレート・エディターの使用 347
- コンパイル、スキーマ 152
- コンポーネント、印刷 156
- コンポーネント、パッケージへのエクスポート 155
- コンポーネント、リポジトリからの削除 154
- コンポーネント、リポジトリ内のコンパイル 153
- コンポーネント状態、管理 289, 305

[サ行]

- サーバー、切断 312
- サーバー・プロパティ、構成 119
- 削除、データベース接続オブジェクト 216
- 作成
 - 関係インスタンス
 - Relationship Manager 内で 248
 - コラボレーション・オブジェクト 220
 - 統合コンポーネント・ライブラリー 20
 - ユーザー・プロジェクト 21
 - 要求ビジネス・オブジェクト 317
- 作成、データベース接続プール 211

- 作成、テスト・プロジェクト 282
- 参加者
 - Relationship Manager 内で 251, 254
- 参加者データによる検索
 - Relationship Manager 内で 247
- 参加者のアクティブ化
 - Relationship Manager 内で 250
- 参加者の検索
 - Relationship Manager 内で 252
- 参加者の非アクティブ化
 - Relationship Manager 内で 250
- 実行 357
- 実行、統合テストの 5
- 実装の段階
 - 開発および構成 10
 - 環境の準備 10
 - 検証 11
 - 設計 10
 - 配置 12
 - 要件の発見および評価 8
- 実動の開始 5
- ショートカット
 - プロパティの変更 19
 - 編成 19
- ショートカットのプロパティの変更 19
- ショートカットの編成 19
- 使用、データベース接続プールの 367
- 処理、コラボレーションでの並行したイベントにより起動されるフローの 361
- 推奨事項
 - インストール 16
- スキーマ、コンパイル 152
- ステップアウト 356
- ステップイントゥ 356
- ステップオーバー 356
- すべてのインスタンスの検索
 - Relationship Manager 内で 246
- 静的関係のキャッシング 367
- セキュリティー、構成 137
- セキュリティー・プロパティ、構成 124
- 設計 10
- 設計モード 278
- 設計モードでの InterChange Server の処理 17
- 接続
 - ClearCase に 24
 - InterChange Server への
 - Relationship Manager 内で 243
 - Relationship Manager 内のサーバーへの 243
- 切断、Relationship Manager 内のサーバーから 243
- ソース・コード制御
 - 使用 23
- ソース・コード制御の使用 23

[タ行]

- ダイアグラム
 - オープンおよびクローズ 348
- ダイアグラムのオープンおよびクローズ 348
- タスク、一時停止 299
- タスク、停止 299
- タスクの選択 296
- 単純属性 320
- 調整、パフォーマンス 361
- ツール設定
 - 構成 20
- 追加、ワークスペースへのプロジェクト 24
- 追加の考慮事項、メモリー・チェッカー・スレッド 370
- データによるインスタンスの検索
 - Relationship Manager 内で 252
- データベース接続、作成 211
- データベース接続オブジェクト、削除 216
- データベース接続の作成 211
- データベース接続プール 115
- データベース接続プール、検証 214
- データベース接続プール、構成 207
- データベース接続プール、コラボレーションでの 216
- データベース接続プール、作成 211
- データベース接続プール、変更 215
- データベース接続プール、マップでの 216
- データベース接続プールの使用 367
- データベース表 208
- データベース・パスワード 124, 130
- データベース・プロパティ、構成 122, 127
- テスト、アプリケーション固有のビジネス・オブジェクト 113
- テスト、インターフェース 117
- テスト、接続 113
- テストが作動可能 290
- テスト・データ、保管 332
- テスト・プロジェクト 280
- テスト・プロジェクト、作成 282
- テスト・ユニット、オープン 285
- デバッグ、マップ 307
- デバッグ操作
 - イベントの実行 357
 - 実行 355
 - ステップアウト 356
 - ステップイントゥ 356
 - ステップオーバー 356
 - 特定のノードに対する実行 357
- デバッグ操作の実行 355
- 同期モード 318

- 統合コンポーネント 1
 - インターフェース 1
 - ソリューション
 - ソリューション 2
 - 統合コンポーネント・ライブラリー 2
 - ユーザー・プロジェクト 2
- 統合コンポーネント・ライブラリー 2
 - 作成 20
- 統合テスト環境 273
 - Collaboration Debugger の切り離し 346
 - Collaboration Debugger の接続 345
- 統合テスト環境、概要 275
- 統合テスト環境、コンソール・ビュー 276
- 統合テスト環境、サーバー構成 277
- 統合テスト環境コンソールの設定、構成 302
- 統合テスト環境による Collaboration Debugger の切り離し 346
- 統合テストの実行 5
- 登録、InterChange Server 273
- 特定のノード
 - に対する実行 357
- 特定のノードに対する実行 357
- トランザクション・ブラケット、構成 217
- トランスポート・プロトコルとしての IDL の使用 21
- トレース・レベル、構成 130

[ナ行]

- 内部ポート・バインディング 229
- ノードの ID およびラベル
 - 表示および非表示 348
- ノードの ID およびラベルの表示および非表示 348

[ハ行]

- パースベクティブのインターフェース
 - Collaboration Debugger 340
- 配置 12
- 配置、インターフェース 117
- 配置、開発統合システムへの 4
- 配置、パッケージ 151
- 配布、コネクタ・エージェントの 366
- バインディング、コラボレーション・オブジェクト・ポート 228
- パッケージ、エクスポート 155
- パッケージ、検証 150
- パッケージ、配置 151
- バッチ・ファイル 17, 279
- バッチ・ファイルの変更 17

- バッチ・モード 319
- パフォーマンスの調整 361
 - イベントにより起動されるフローのコネクタ・コントローラーでの並行処理 362
 - イベントにより起動されるフローのコラボレーションでの並行処理 361
 - イベントにより起動されるフローのコラボレーション・オブジェクト・グループでの並行処理 362
- パフォーマンス・テストの実行 5
- 汎用ビジネス・オブジェクト 110
- 汎用ビジネス・オブジェクト、開発 115
- 汎用ビジネス・オブジェクトの開発 115
- ビジネス・インテグレーション・システム
 - の定義 3
- ビジネス・インテグレーション・システムのライフ・サイクル
 - 開発統合環境への配置 4
 - 実動の開始 5
 - 統合テストの実行 5
- パフォーマンス・テストの実行 5
- ビジネス・オブジェクト 109
 - インポート 324
 - エクスポート 324
 - 構成 332
 - 属性 320
 - 要求 317
 - ITE 内で削除 323
 - ITE に保存 323
- ビジネス・オブジェクトによるインスタンスの検索
 - Relationship Manager 内で 252
- ビジネス・オブジェクトのサポート 117
- ビジネス・オブジェクト・ファイルのロードとアンロード
 - Relationship Manager 内で 251
- ビジネス・プロセス・インターフェース、開発 109
- ビジネス・プロセス・インターフェースの開発 109
- 必要なソフトウェア
 - インストール 15
- 必要なソフトウェアのインストール 15
- 非同期モード 318
- ビュー
 - 変数 343
- 表示する参加者のフィルター処理
 - Relationship Manager 内で 253
- プライバシー、構成 139
- プラグインがインストール済みかつ使用可能であることの確認 23
- プリンターへの関係データの送信
 - Relationship Manager 内で 255
- ブレークポイント
 - 削除 353

- ブレークポイント (続き)
 - 使用可能化 354
 - 使用不可化 353
 - 処理 351
 - 設定 352
 - ビュー 342
 - 表示 352
 - フィルタリング 354
- ブレークポイントの削除 353
- ブレークポイントの使用可能化 354
- ブレークポイントの使用不可化 353
- ブレークポイントの処理 351
- ブレークポイントの設定 352
- ブレークポイントの表示 352
- ブレークポイントのフィルタリング 354
- プロジェクト、ユーザー 2
- プロジェクトの作成、ユーザーの 21
- ブロッキング・タイプ 238
- ページの設定
 - Relationship Manager 内で 256
- 並行処理の実装
 - コネクタ・エージェント並列処理を使用しない理由 363
 - コネクタ・エージェント並列処理を使用する理由 363
 - 要求を並行処理するためのコネクタ・エージェントの構成 364
- 変更、コラボレーション・オブジェクト 225
- 変更、接続数 215
- 変更、データベース接続プール 215
- 変更、ログイン情報 215
- 変数の処理 358
- 変数ビュー 343

[マ行]

- マップ 110, 216
- マップ、開発 115
- マップ、デバッグ 307
- マップの開発 115
- メモリー・チェッカー・スレッドの使用 367
 - 追加の考慮事項 370
 - メモリー・チェッカー・スレッド・プロパティの構成 369
 - CW_MEMORY_MAX パラメーターの設定 368
- メモリー・チェッカー・スレッド・プロパティの構成 369

[ヤ行]

- ユーザー従属 287
- ユーザー・プロジェクト 2

ユーザー・プロジェクト (続き)
作成 21
有効トランザクション・レベル 232
ユニット 280
要求を並行処理するためのコネクタ・エ
ージェントの構成 364
要件の発見および評価 8

[ラ行]

ライブラリー、統合コンポーネント 2
リカバリー・モード 236
リポジトリ (InterChange Server)
ロード 143
リポジトリ、管理 289, 305
リポジトリ、検証 153
リポジトリ、コンポーネントのコンパ
イル 153
リポジトリ、コンポーネントの削除
154
リポジトリ、コンポーネント・リストの
印刷 156
リポジトリ、削除 154
ルーティング 207
ローカルでのインターフェースの開発 3
ローカルでのインターフェースのテスト
4
ローカルでのインターフェースの配置 4
ロギング、開発環境での
構成 18
ロギング・プロパティ、構成 131

[ワ行]

ワークスペース
設定 20
プロジェクトの追加 24
ワークスペースの設定 20

C

ClearCase
接続 24
System Manager の構成 23
へのプロジェクトの移動 24
ClearCase へのプロジェクトの移動 24
Collaboration Debugger
イベント・ビュー 340
インターフェース 339
切り離されたコラボレーション・オブ
ジェクトの削除 347
切り離し 346, 347
始動 337
使用 337
接続 345, 346

Collaboration Debugger (続き)
統合テスト環境での切り離し 346
パースペクティブのインターフェース
340
ビューの設定 338
ブレークポイント・ビュー 342
System Manager による 346
Collaboration Debugger から切り離された
コラボレーション・オブジェクトの削除
347
Collaboration Debugger による
Collaboration Debugger の切り離し 347
Collaboration Debugger の切り離し 346
Collaboration Debugger の始動 337
Collaboration Debugger の使用 337
Collaboration Debugger の接続
統合テスト環境での 345
Collaboration Debugger での 345, 346
System Manager による 345
CSM (CrossWorlds System Manager) 30
CW_MEMORY_MAX パラメーターの設定
368
CW_MEMORY_MAX パラメーターの設定
368

E

E メール通知 234
E メール・プロパティ、構成 133

I

IDL、トランスポート・プロトコルとして
の
使用 21
InterChange Server インスタンスの定義
3
InterChange Server からの切断
Relationship Manager 内で 244
InterChange Server、設計モードでの
処理 17
InterChange Server、設計モードでの開始
278
InterChange Server、登録 273

R

Rational ClearCase
使用 23
Rational ClearCase の使用 23
Relationship Manager
関係 ID による検索 246
関係インスタンスの検索 245
関係インスタンスの個数の取得 248
関係インスタンスの削除 249

Relationship Manager (続き)
関係インスタンスの作成 248
関係データの印刷 255
関係データの印刷プレビューの使用
256
関係を開く 244
サーバーへの接続および接続の切断
243
参加者データによる検索 247
参加者のアクティブ化 250
参加者のクリーンアップ 254
参加者の検索 252
参加者のコピー 251
参加者の非アクティブ化 250
参加者の非アクティブ化とアクティブ
化 250
すべてのインスタンスの検索 246
データによるインスタンスの検索 252
ビジネス・オブジェクトによるイン
スタンスの検索 252
ビジネス・オブジェクト・ファイルの
ロードとアンロード 251
表示する参加者のフィルター処理 253
プリンターへの関係データの送信 255
ページの設定 256
InterChange Server からの切断 244
InterChange Server への接続 243
Relationship Manager の始動 243
Relationship Manager の使用 241
repos_copy
コマンド 143
repos_copy オプション 143
repos_copy の構文 143
repos_copy、印刷 150

S

start_server.bat ファイル 278
System Manager
ClearCase 接続
構成 23
Collaboration Debugger の切り離し
346
Collaboration Debugger の接続 345
System Manager による Collaboration
Debugger の切り離し 346

V

VCM
使用 25
VCM の使用 25

W

WebSphere InterChange Server 開発モデル

 インターフェースのライフ・サイクル

 3

 ローカルでのインターフェースのテ

 スト 4

 ローカルでのインターフェースの配

 置 4

 ローカルでのインターフェースの開発

 3



Printed in Japan