

# Lotus Domino 7 アプリケーションの 開発

DB2 統合によるストレージ機能および  
リレーショナル機能の追加

Web サービスによる Lotus  
Domino アプリケーションの拡張

ユーザビリティの改善と  
新しい設計ツールおよび診断



Philip Monson  
Lisa Chase  
Dick Larsson  
Manny Santana  
Frank Grapengiesser

SG88-4022-00  
( 英文原典 : REDP-4102-00)





International Technical Support Organization

**Lotus Domino 7 アプリケーションの開発**

2006 年 3 月

**注意：**本書および本書で紹介する製品をご使用になる前に、ix 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 Lotus Notes/Domino のバージョン 7、および IBM Lotus Domino Designer のバージョン 7 に適用されます。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。  
(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： REDP-4102-00  
Lotus Domino 7 Application Development

発行： 日本アイ・ビー・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 2007.1**

© Copyright International Business Machines Corporation 2006. All rights reserved.

© Copyright IBM Japan 2007





# 目次

特記事項	ix
商標	x
前書き	xi
この Redpaper を作成したチーム	xi
<b>第 1 章 Lotus Domino 7 アプリケーションの開発の概要</b>	<b>1</b>
1.1 ITSO Electronics のシナリオ	2
1.2 本書の構成	5
1.3 Lotus Domino Designer でのアプリケーション開発	5
<b>第 2 章 ITSO Electronics アプリケーション拡張のための DB2 統合の使用</b>	<b>7</b>
2.1 Lotus Domino と DB2 の統合について	8
2.2 DB2 Access ビュー	10
2.2.1 DB2 Access ビューの作成	11
2.2.2 DB2 Access ビューへの TIMEDATES の エクスポート	15
2.2.3 DB2 Access ビューへの日付範囲の挿入	15
2.2.4 DB2 Access ビューのステータスの確認	15
2.2.5 DB2 Access ビュー作成時の他の考慮事項	16
2.2.6 DB2 Access ビューの ITSO Electronics アプリケーションへの導入	16
2.3 Query ビュー	18
2.3.1 Query ビューの作成	18
2.3.2 join を使用した Query ビューの作成	20
2.3.3 union を使用した Query ビューの作成	21
2.3.4 Query ビューを @ 式で動的に作成する	22
2.3.5 Query ビューから文書を開く	24
2.3.6 Web ブラウザーで Query ビューを開く	25
2.3.7 Query ビューおよび LotusScript	25
2.3.8 ITSO Electronics アプリケーションの組み込み Query ビュー	25
2.3.9 Query ビューに関する詳細な考慮事項	26
2.4 DB2 でフェデレートされたデータについて	27
2.4.1 DB2 でフェデレーションを有効にするには	27
2.4.2 DB2 フェデレーションが有効な Notes データベースの実装	37
2.5 DB2 および Lotus Domino のセキュリティー上の考慮事項	39
2.6 DB2 および Lotus Domino のトラブルシューティング	40
2.6.1 DB2 Access ビューに必要な ACL	41
2.6.2 Query ビューのオープン時のエラー	41
2.6.3 問題の根本的な原因究明に役立てるために収集するデータ	42
<b>第 3 章 Web サービスによる ITSO Electronics アプリケーションの拡張</b>	<b>43</b>
3.1 サービス指向アーキテクチャーと Web サービス	44
3.2 Web サービス要素と WSDL	45
3.2.1 WSDL 文書	45
3.2.2 Simple Object Access Protocol	46
3.2.3 WSDL のインポート	46
3.3 Web サービスを実装するビジネス上の理由	47
3.4 ITSO Electronics アプリケーションへの Web サービスの追加	47
3.4.1 ITSO Electronics に Web サービスを追加するビジネス上の要件	48
3.4.2 ユーザーの要件	48

3.5 Lotus Domino Designer 7 での単純な Web サービスの作成	48
3.5.1 新しい Web サービスの作成	48
3.5.2 ブラウザーでの Web サービスの表示	50
3.5.3 生成された WSDL 文書	50
3.5.4 単純な Web サービスのテスト	51
3.6 Lotus Domino Designer 7 での複雑な Web サービスの作成	57
3.7 WSDL 文書のエクスポート	62
3.8 WSDL 文書のインポート	62
3.9 Web サービスでの例外およびエラー処理	65
3.10 Web サービスのセキュリティ	65
3.11 Java を使用した Web サービスのコンシューム	66
3.12 LS2J を使用した Web サービスのコンシューム	71
3.12.1 LS2J について	72
3.12.2 Web サービスの作成	72
3.12.3 Java を使用して Web サービスをコンシュームするためのスクリプト・ライブラ リの作成	73
3.12.4 LotusScript エージェントの作成	75
3.12.5 認証と Web サービス	76
3.13 どのスタイルの WSDL を使用するか	76
<b>第 4 章 ITSO Electronics アプリケーションへの新しい設計要素の実装</b>	77
4.1 右クリック・メニューのビュー・アクション	78
4.2 自動保存のフォームのプロパティ	79
4.2.1 Lotus Notes クライアントでの自動保存の使用可能化	79
4.2.2 フォームでの自動保存の使用可能化	80
4.2.3 自動保存機能を使用して保存された文書のリカバリー	82
4.2.4 手動での自動保存の使用	83
4.2.5 自動保存データベースへのアクセス	83
4.3 共有列	83
4.3.1 共有列の作成	84
4.3.2 共有列の使用	85
4.4 ビューおよびフォルダの拡張された実行時カスタマイズ	87
4.5 式言語に対する追加機能	92
4.6 LotusScript、COM、および OLE クラスに対する追加機能	93
4.7 Java および CORBA クラスに対する追加機能	93
<b>第 5 章 ITSO Electronics アプリケーションの診断およびトラブルシューティング</b>	95
5.1 ITSO Electronics アプリケーションにおけるコード・プロファイル	96
5.2 ITSO Electronics アプリケーションにおける LotusScript デバッグ	97
5.3 ITSO Electronics アプリケーションにおける Java デバッグ	99
5.3.1 Lotus Notes クライアントでの Java デバッグの有効化	99
5.3.2 エージェント、Web サービス、またはスクリプト・ライブラリーで Java デバッグ を使用可能にする	101
5.3.3 デバッガーの JVM との接続	102
5.3.4 セキュリティー構成	108
<b>付録 A 追加の資料</b>	109
Web 資料の場所	109
Web 資料の使用	109
<b>関連資料</b>	111
IBM Redbooks	111
オンライン・リソース	111
IBM Redbooks の入手方法	112



IBM からのヘルプ .....	112
------------------	-----



# 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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。

〒106-8711 東京都港区六本木 3-2-12 IBM World Trade Asia Corporation Intellectual Property Law & 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。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。

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証できません。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。

## 著作権使用許諾：

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

# 商標

以下は、米国内における IBM Corporation の商標です。

AIX 5L™

AIX®

DB2 Universal Database™

DB2®

developerWorks®

Lotus Domino Designer®

Lotus Domino®

IBM®

Lotus Enterprise Integrator®


Lotus Notes®

Lotus Workflow™

Lotus®

Lotus Notes®

Rational®

Redbooks (ロゴ) ™

Redbooks™

WebSphere®

Workplace Client Technology™

Workplace™

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。

# 前書き

IBM® Lotus® Domino® Designer® ソフトウェアは、バージョン 7 で、Lotus Domino ソフトウェア・ベースのアプリケーションを開発するための最初のコラボレーション・アプリケーション開発ツールになるという期待に応えます。Lotus Domino Designer ソフトウェアは、IBM Workplace™ ファミリーの不可欠な部分として企業を支援し、企業が従業員の生産性を向上させ、Lotus Domino アプリケーションを迅速に構築および配置できるようにします。これにより、組織は変化する率優先的事業活動に素早く対応できます。新しい機能は、Web 標準との緊密な統合、IBM テクノロジーとのインターオペラビリティおよび使いやすさに焦点を当てています。

この IBM Redpaper の目的は、IBM Lotus Notes® および IBM Lotus Domino 7 の最新機能の使用方法をアプリケーション開発者に紹介することです。この Redpaper では、架空企業のサンプル・アプリケーションを開発者がどのように拡張するか示すことで、IBM DB2® Universal Database™ の統合、Web サービスなどの強力なアプリケーション開発機能について説明します。

## この Redpaper を作成したチーム

この Redpaper は、Cambridge Center の International Technical Support Organization で働く、世界各国の専門技術者のチームにより制作されました。

**Philip Monson** は、マサチューセッツ州ケンブリッジにある ITSO Lotus Center のプロジェクト・リーダーです。Philip は、内部で使用する初期バージョンの Lotus Notes が発表されたときに会社に加わり、それ以来 Lotus と IBM に 15 年間関わっています。彼は、IT、営業、および開発部門で、管理、技術、およびコンサルティングの職務を担当してきました。

**Lisa Chase** は、IBM Software グループの Lotus Software 部門のエンジニアです。彼女は過去 4 年にわたって IBM Messaging and Collaborative Software グループの一員であり、Lotus Notes Client チームのメンバーです。現在は Lotus Domino Access for Microsoft® Outlook® クライアントのチーム・エンジニアリングに参加しています。Lotus Notes クライアントのカレンダーおよびスケジュール機能や Microsoft Outlook および Exchange に関する知識を保持しており、現在の Lotus Domino Migration Tool の中心的な責任を担っています。

**Dick Larsson** は、Ekakan (<http://www.ekakan.com>) の開発および調査に関する統括責任者です。Ekakan は、コラボレーション・ソリューションおよび知識管理を企業が改善および開発できるよう支援する会社です。彼の Lotus Domino および J2EE™ に関する広範な経験および新しいテクノロジーを直感的に把握する力は、ヨーロッパの開発者コミュニティ内で高く評価されてきました。Larsson 氏は、Ekakan に加わる前、スウェーデンにある Lotus Notes に関する有数のコンサルタント会社で、開発に関する各種の上級職についていました。Larsson 氏の連絡先は <mailto:dick.larsson@ekakan.com> です。

**Manny Santana** は、オーストラリアにある IBM グローバル・サービスの IT スペシャリストで、シドニーを拠点にしています。彼には、IBM Global Web Architecture 環境および IBM Global Notes Architecture 環境で、各種内部 IBM アプリケーションを開発、導入、およびサポートした 10 年を超える経験があります。彼は、シドニー大学で理学士号を取得しており、また Lotus Notes/Domino 6/6.5 および 7 の IBM Certified Advanced Application Developer です。

**Frank Grapengiesser** は、ダブリンにある IBM Lotus のお客様サポートでソフトウェア・エンジニアとして 5 年を超える勤務をしています。彼は現在 Lotus Workflow™ およびデータベース設計の分野でお客様をサポートしています。これより前、彼はドイツのマンハイムにある Lotus Notes アプリケーションを開発する Bechtle AG に 2 年半勤務していました。彼は、ハノーバー大学で機械工学の修士号を取得しています。

本プロジェクトにおいて、以下の方々に貢献していただきました。厚く感謝いたします。

Mark Jourdain、プロダクト・マネージャー - アプリケーション開発、IBM Software グループ、Lotus

Andre Guirard、Lotus Notes クライアント・チーム、IBM Software グループ、Lotus

John Curtis、シニア・テクニカル・スタッフ・メンバー - NSF/DB2 の透過性、IBM Software グループ、Lotus

John Grosjean、Lotus Domino バックエンド クラス - Web サービス、IBM Software グループ、Lotus

Steve Nikopoulos、Lotus Domino プログラマー、IBM Software グループ、Lotus

Jane L. Wilson、ナレッジ・システム設計者、IBM Software グループ、Lotus


Mary Jrolf、顧問ソフトウェア・エンジニア、IBM Software グループ、Lotus

Russell Butek、Web サービス・コンサルタント、IBM Software グループ、アプリケーション  
および統合

Gary Rheume、シニア・ソフトウェア・エンジニア、IBM Software グループ、Lotus

Jennifer Dunne、IT スペシャリスト、IBM Software グループ、Lotus

Tom McGary、シニア・ソフトウェア・エンジニア、IBM Software グループ、Lotus



# Lotus Domino 7 アプリケーションの開発の概要

Lotus Domino 7 アプリケーションの開発では、機能性が大きく進歩し、強力で、簡単に配布でき、拡張が容易な IBM Lotus Domino アプリケーションをユーザーが作成できるようになりました。IBM Lotus Domino Designer は、このリリースで、IBM DB2 Universal Database (UDB) と Lotus Domino との統合の出現で強化されたデータ・ストレージ機能およびリレーショナル機能、Web サービスのサポートにより強化されたデータ・アクセスおよびデータ操作、追加のアプリケーション設計要素、および拡張されたアプリケーション・デバッグ・ツールおよび診断ツールをアプリケーション開発者に提供します。Lotus Notes/Domino 7 は、固有のビジネス要件を満たす Lotus Domino アプリケーションを作成するためのツールを開発者に提供し、その一方で Lotus Notes、Lotus Domino Web Access、またはインターネット・ブラウザ・インターフェースのいずれかを介してアクセスしても高機能で使いやすいインターフェースを Lotus Domino アプリケーションの利用者に提供します。さらに、Lotus Notes/Domino 7 で導入された Web サービス・サポートにより、Microsoft .NET、Java™、Java 2 Platform、Enterprise Edition (J2EE)、およびその他のアプリケーションがカスタム・コーディングなしで Lotus Domino とやりとりができます。

この Redpaper では、架空の会社である ITSO Electronics のために拡張したサンプル・アプリケーションを介して、Lotus Domino 7 アプリケーションの開発環境の新しい機能と要素について説明します (Redpaper チームを ITSO Electronics 社のコンサルタントとお考えください)。アプリケーション開発者の方を対象に、Lotus Domino と統合された DB2 のリレーショナル機能およびストレージ機能を使用する Lotus Domino アプリケーションを設計する方法について説明し、その後、アプリケーションに Web サービスを統合することで、会社が外部のサプライヤーおよびパートナーを扱う能力を向上させる方法について説明します。これらでは、Lotus Domino 7 で新しく採用された設計要素を使用します。最後に、LotusScript と Java の新しいデバッグ機能、およびプロファイル機能を使用して、問題またはパフォーマンスの問題をデバッグおよび診断する方法について説明します。

## 1.1 ITSO Electronics のシナリオ

前述したように、本書では次の章から Lotus Domino 7 の新しい主要なアプリケーション開発機能について、より詳細に説明していきます。この説明を簡単にするために、本書全体でサンプル・アプリケーションを使用して、ITSO Electronics のビジネスの文脈において新しい各機能をどのように実装するかを例示します。このサンプル・アプリケーションは、IBM Redbooks の Web サイトからのダウンロードにより利用可能です。このサンプル・アプリケーションのダウンロード方法については、109 ページの付録 A、「追加の資料」を参照してください。

**注：**このアプリケーションは、Lotus Domino 7 の新しいアプリケーション開発機能の使用例を示すことを目的としています。実動設定での使用を目的としたものではありません。

### ITSO Electronics アプリケーションの概要

架空の会社 ITSO Electronics では、Lotus Domino を使用して営業部門をサポートしています。この会社では、長年にわたり、営業追跡アプリケーションを構築して、営業部隊が顧客との対話を追跡できるようにしています。データベースには Lotus Notes クライアントまたは Web ブラウザーからアクセスできます。

この ITSO Electronics 営業追跡アプリケーションには、2 つのメイン・データベースがあります。Sales データベースと Customers データベースです。図 1-1 と 4 ページの図 1-2 は、Lotus Notes からアクセスしたときのユーザー・インターフェースです。3 番目のデータベースである Products データベースは、製品キーワード用に使用されており、営業部隊は直接アクセスできません。最後に、4 番目のデータベースが Web サービス要素用として既存のアプリケーション環境に追加されています。

**注：**学習の便宜を図るために、本書で説明する新機能を含んでいるサンプル・シナリオ・データベースは、109 ページの付録 A、「追加の資料」からすべて入手できます。



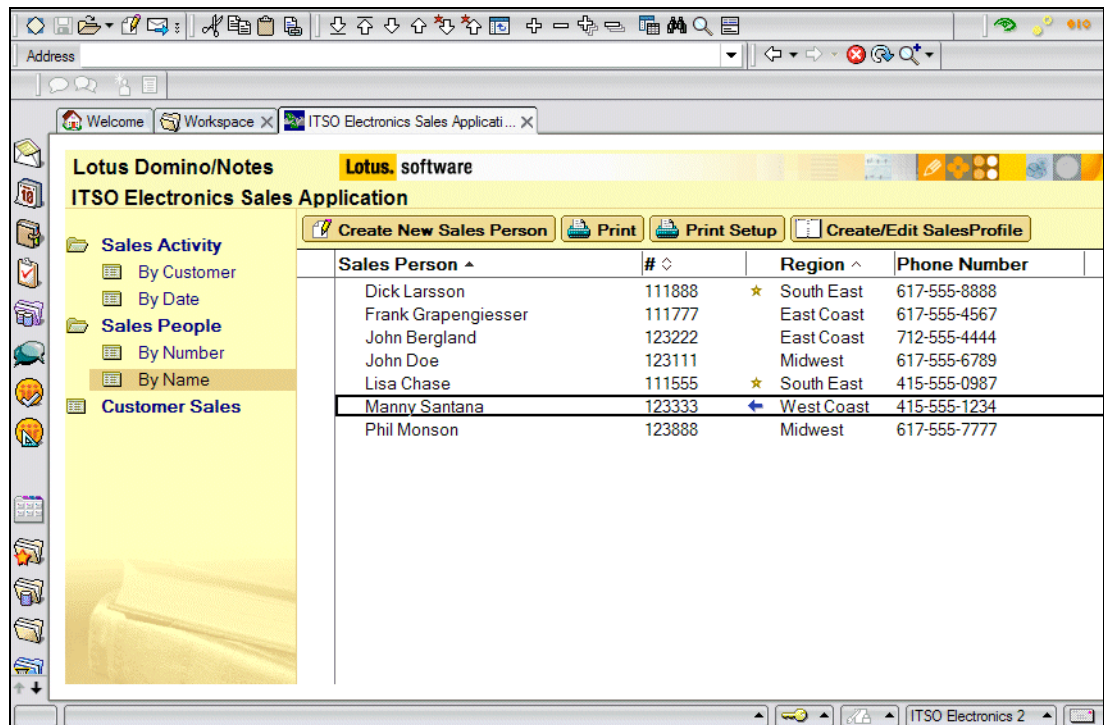


図 1-1 営業追跡アプリケーションの Sales データベースの Lotus Notes インターフェース

注：Web サービスは、いずれの Lotus Domino データベースにも追加または取り込みが可能なので、Web サービスの実装で個別の Lotus Domino データベースを使用する必要はありません。本書ではデモンストレーションの目的で、Web サービス情報を単純化および分離するために個別のデータベースを使用しています。

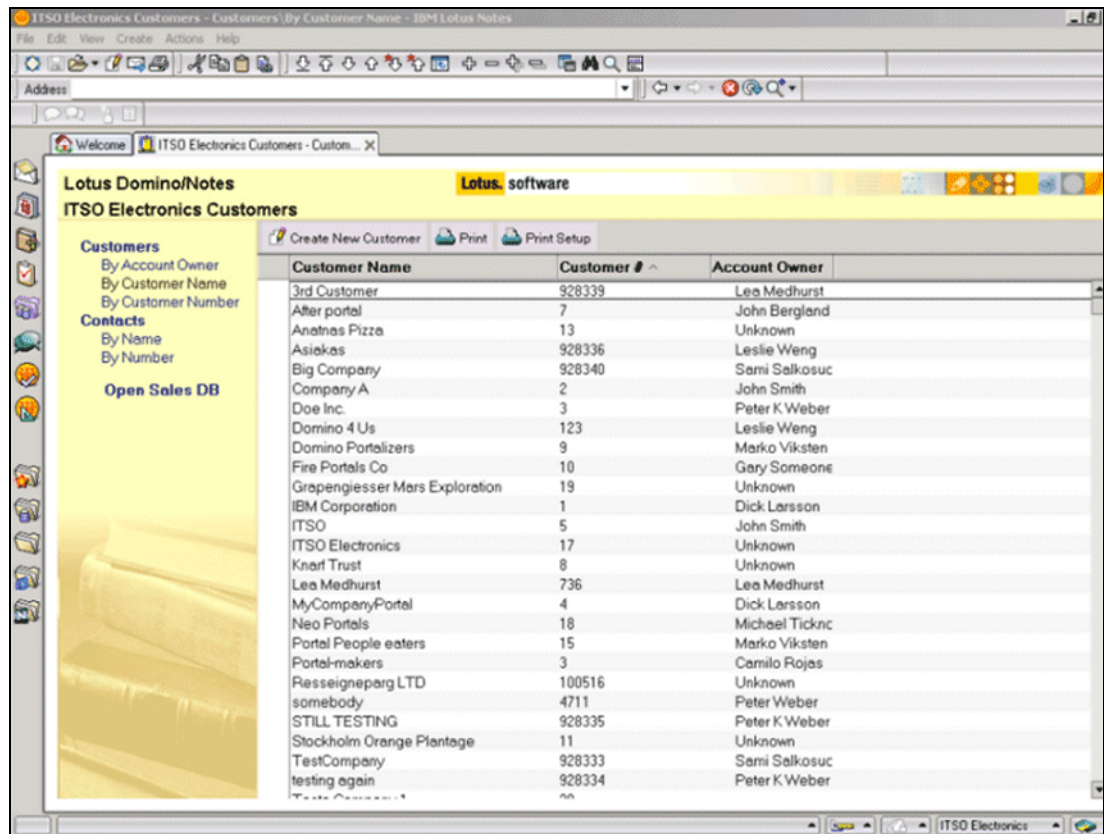


図 1-2 営業追跡アプリケーションの Customers データベースの Lotus Notes インターフェース

図 1-3 は、ITSO Electronics アプリケーションのアーキテクチャーを図示したものです。Sales、Customers、Products、および Web Services の各データベースは DB2 UDB サーバー上にありますが、Lotus Domino サーバーからアクセスできます。

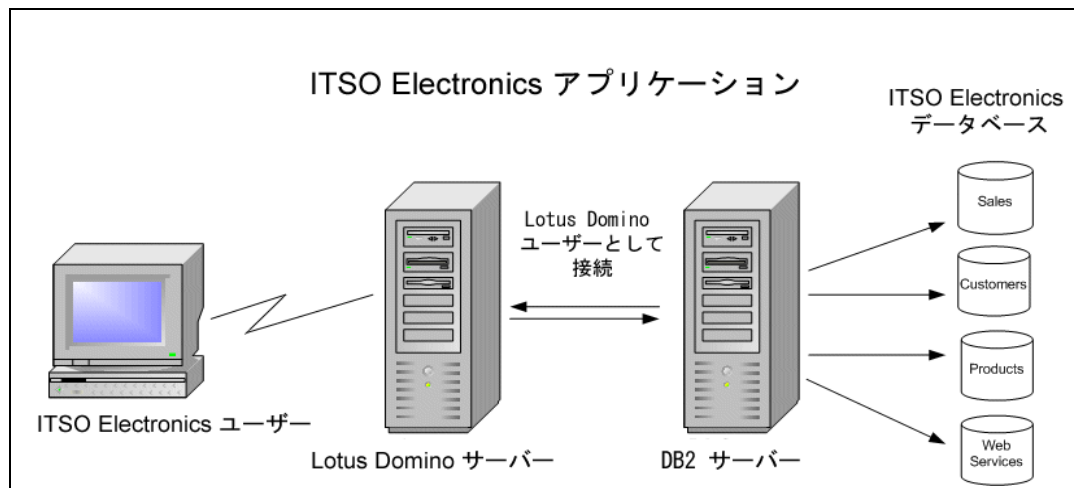


図 1-3 ITSO Electronics アプリケーションのアーキテクチャー

## 1.2 本書の構成

この章に続く 4 つの章では、Lotus Domino 7 での ITSO Electronics アプリケーションの開発について説明します。

7 ページの第 2 章、「ITSO Electronics アプリケーション拡張のための DB2 統合の使用」では、DB2 と Lotus Domino の統合の拡張について説明し、DB2 のデータ・ストレージおよびリレーショナル機能で ITSO Electronics の Lotus Domino アプリケーションを、最終的には皆さんの Lotus Domino アプリケーションを極めて強力にする方法について説明します。Lotus Enterprise Integrator® を使用する従来のリレーショナル・データベース統合機能と対照させながら、Lotus Domino アプリケーション開発における DB2 アーキテクチャーの利点を検証します。DB2 と Lotus Domino の統合を準備する方法、DB2 アクセス・ビューとクエリー・ビュー、DB2 でのフェデレーション、DB2 データ・ストレージを Lotus Domino と統合する場合のセキュリティに関する考慮事項について説明します。また、DB2 環境をインストールおよび構成する場合のトラブルシューティングに関するヒントも含まれています。

Lotus Domino アプリケーション開発ツールキットに追加されたもののなかで、最も魅力的なのはおそらく Web サービスです。43 ページの第 3 章、「Web サービスによる ITSO Electronics アプリケーションの拡張」で、Web サービスのテクノロジーについて説明し、さらに Web サービスが ITSO Electronics の Lotus Domino アプリケーションの機能と範囲をどのように拡張するかを説明します。COM 統合により、5.0.x リリース以降の Lotus Domino ではオープン・アクセスが可能になりましたが、Web サービスは、このデータ・アクセス・モデルの制約を超えて、Lotus Domino の開発者がデータ・サービス統合を行えるようにします。Web サービス記述言語 (WSDL)、Simple Object Access Protocol (SOAP)、Web サービス要素を Lotus Domino アプリケーションに追加するプロセス、例外およびエラー処理、セキュリティ、および Web サービスのコンシュームについて説明し、さらにサービス指向アーキテクチャー (SOA)、Web サービスを SOA スキームに適合させる方法について説明します。

Lotus Domino Designer 7 には新しい強力な設計要素がいくつか含まれていますが、これについては 77 ページの第 4 章、「ITSO Electronics アプリケーションへの新しい設計要素の実装」で、ITSO Electronics アプリケーションの拡張の際に説明し実例を示します。右クリック・アクション、新しい共有列機能について、さらには LotusScript、Java クラスなどの式言語に追加されたものについても詳細に説明します。

95 ページの第 5 章、「ITSO Electronics アプリケーションの診断およびトラブルシューティング」では、Lotus Domino 7 アプリケーション開発環境の新しい診断ツールおよびデバッグ・ツールについて説明します。ITSO Electronics アプリケーションを例にして、Lotus Notes アプリケーションを最適化するコード・プロファイルの機能、LotusScript のデバッグ機能に追加されたもの、およびリモート Java デバッグなどの新しい Java デバッグ機能について説明します。

## 1.3 Lotus Domino Designer でのアプリケーション開発

IBM Lotus Domino Designer は、Lotus Domino サーバーを使用するビジネス・アプリケーションを作成するための強力なアプリケーション開発環境です。本書ではその目的のために、対象読者を、Lotus Domino リリース 6 のアプリケーション開発機能および Lotus Domino リリース 6 の Lotus Domino Designer の設計要素について既に理解している経験豊富な Lotus Domino アプリケーションの開発者と想定しています。ITSO Electronics サンプル・アプリケーションの開発デモンストレーションは、データ・ストレージおよびリレーショナル要素として DB2 を追加するところから開始するので、DB2 コンポーネントなしで Lotus Domino アプリケーションを構築する方法に精通していることが前提となります。Lotus Domino Designer のアプリケーション開発環境について理解していない場合は、IBM Redbook の「Domino Designer 6: 開発者用ハンドブック」(SG88-8519) を参照して、Lotus Domino Designer の設計要素および機能について理解してください。この Redbook は以下から入手できます。

[http://www-1.ibm.com/support/docview.wss?rs=929&q=ibm&uid=std34506ead6866440f049256dc7004917c3&loc=ja\\_JP&cs=utf-8&cc=jp&lang=ja](http://www-1.ibm.com/support/docview.wss?rs=929&q=ibm&uid=std34506ead6866440f049256dc7004917c3&loc=ja_JP&cs=utf-8&cc=jp&lang=ja)



## ITSO Electronics アプリケーション ン拡張のための DB2 統合の使用

この章では、IBM Lotus Notes と Lotus Domino 7 を IBM DB2 Universal Database (UDB) と統合可能にする新機能について説明します。新しい設計機能については、必要に応じて ITSO Electronics アプリケーションを例に挙げ、説明します。

以降のページで説明するトピックは、次のとおりです。

- ▶ Lotus Domino と DB2 の統合について
- ▶ DB2 Access ビュー
- ▶ Query ビュー
- ▶ DB2 でフェデレートされたデータについて
- ▶ DB2 および Lotus Domino のセキュリティ上の考慮事項
- ▶ DB2 および Lotus Domino のトラブルシューティング

**注** :DB2 対応の IBM Lotus Notes および Lotus Domino 7 は、通常の用途ではなく評価目的の使用のみに限定して提供されるなど、機能制限付きでリリースされています。機能制限プログラムの詳細とこのトピックに関するその他の重要な情報については、次の URL を参照してください。

<http://www.ibm.com/lotus/nsfdb2>

## 2.1 Lotus Domino と DB2 の統合について

Lotus Notes および Lotus Domino は、Lotus Enterprise Integrator、Lotus Domino Enterprise Connection Services、Data Connection Resources、および Lotus Connectors LotusScript Extensions などの製品や機能を通じ、リレーショナル・データベース、トランザクション、およびエンタープライズ・リソース・プランニング (ERP) システムとの統合における確かな実績を上げています。Lotus は、これらの機能や製品を Lotus Notes および Lotus Domino 7 で継続的に提供するとともに、IBM DB2 リレーショナル・データベースとの統合を図るための、プログラム非依存型かつ分散処理型の新たなアプリケーション開発オプションを提供することにより、業務別のビジネス・ユーザーや上級開発者などが容易に利用できるようにします。IBM が展開する Lotus および DB2 製品は、ともに連携動作することにより統合機能が改善されています。さらに、Lotus Domino 7 Server には、新規または既存の Lotus Domino データベースを DB2 対応データベースとして選択的に指定できる機能があります。DB2 対応の Lotus Domino データベースには Lotus Domino 7 Server を介して透過的にアクセスできますが、データベースは DB2 システムに保管されます。これによって、DB2 対応の Lotus Domino データベースでリレーショナル構造を使用でき、DB2 対応 Lotus Domino データベースのデータをリレーショナル構造で使用できます。その一方で、複製、メール・ルーティング、管理アクセスなどの Lotus Notes API の機能は引き続きサポートされます。IBM DB2 はリレーショナル・データベース・システムであるため、個々の DB2 のデータベース、テーブル、ビュー、フェデレーテッド・オブジェクトを、共通データ・フィールドおよびリレーションシップを使用して新規のデータ・セットに容易に組み合わせることが可能です。この機能により、開発者は新たな方法でデータを処理および表示したり、Lotus Domino データ、DB2 データ、およびリレーショナル機能を使用する、さらに効果的に統合された Lotus Domino アプリケーションと他のアプリケーションをより容易に作成したりすることができ、この強力な新機能については、この章の後半でさらに詳しく説明します。

Lotus Domino と DB2 の新しい統合機能を使用する前に、まず DB2 サーバーをインストールして、ユーザーの Lotus Domino 7 サーバーが DB2 サーバーに接続されるように構成する必要があります。Lotus Domino 7 では、DB2 の新規統合機能は、Microsoft Windows® または IBM AIX® 5L™ のオペレーティング・システムで使用できます。

### DB2 のローカル・インストール

Lotus Domino と DB2 の統合のセットアップ時に、ローカルとリモートのどちらでインストールするかを選択できます。インストールのタイプをローカルに指定すると、Lotus Domino サーバーと DB2 サーバーは、同一マシンにインストールされます。ローカルのインストールは、テスト環境または小規模の実動システムに適しています。図 2-1 に、システムのローカル・インストール例を示します。

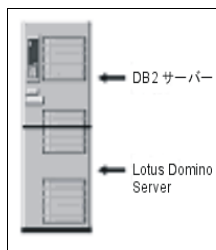


図 2-1 ローカル・インストール: Lotus Domino サーバーと DB2 サーバーを同一マシンにインストール

### DB2 のリモート・インストール

サーバーの負荷が大きい、より規模の大きいインストールまたはシステムの場合は、恐らくリモート・インストールが適しています。この場合、Lotus Domino サーバーと DB2 サーバーをそれぞれ別のマシンにインストールします。9 ページの図 2-2 に、リモート・インストールの例を示します。

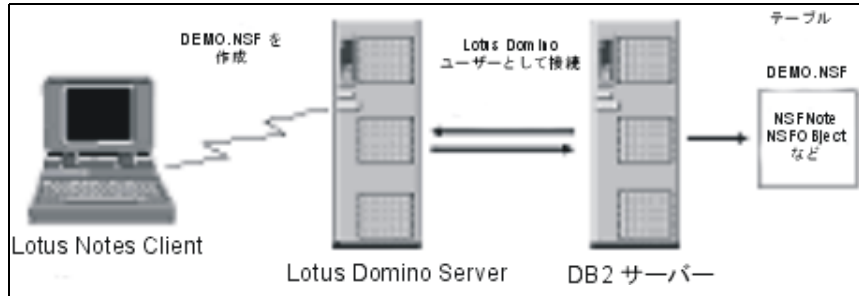


図2-2 リモート・インストール: Lotus Domino サーバーとDB2 サーバーを異なるマシンにインストール

DB2 統合には2つのレベルがあります。最初のシナリオでは、Lotus Domino サーバーと DB2 サーバーを同一マシンにインストールしていますが、次のシナリオでは、それぞれを別個のマシンにインストールします。

Lotus Domino サーバーと DB2 サーバーを同じマシンにインストールしている場合は、次の手順に従ってください。

1. Lotus Domino 7 を Lotus Domino サーバーにインストールします。
2. 次の各要素を DB2 サーバーにインストールします。
  - a. 次のファイルを含む DB2 UDB Enterprise Server Edition バージョン 8.2.2
    - i. DB2 UDB Enterprise Server バージョン 8.2
    - ii. DB2 Universal Database バージョン 8 フィックス・パック 9a
  - b. 利用制限の有効化キー
3. DB2 Access サーバーをインストールします。

Lotus Domino サーバーと DB2 サーバーをそれぞれ異なるマシンにインストールしている場合は、次の手順に従ってください。

1. Lotus Domino 7 を Lotus Domino サーバーにインストールします。
2. 次の各要素を DB2 サーバーにインストールします。
  - a. 次のファイルを含む DB2 UDB Enterprise Server Edition バージョン 8.2.2
    - i. DB2 UDB Enterprise Server バージョン 8.2
    - ii. DB2 Universal Database バージョン 8 フィックス・パック 9a
  - b. 利用制限の有効化キー
3. DB2 ランタイム・クライアントを Lotus Domino サーバーにインストールします。
4. DB2 Access サーバーを DB2 サーバーと同じマシンにインストールします。

**注** :DB2 Access サーバーをインストールするマシンに 7.0 より古いバージョンの Lotus Notes および Lotus Domino がインストールされている場合はすべて削除してください。これらの製品がインストールされていると、システムが稼働しません。先行バージョンの Lotus Notes/Domino サーバーを保持する場合は、インストール先のプログラム・フォルダーの名前を変更してください。nnotes.dll が PATH にないことを確認してください。

本書では、Lotus Domino 7 の新機能に焦点を当てます。したがって、セットアップ・プロセスについては概要のみ説明します。Lotus Domino と DB2 の統合の実際のセットアップおよび保守手順については、次を参照してください。

- ▶ 「Lotus Domino Administrator 7 Help」のトピック『Domino and DB2』



[http://www.lotus.com/1dd/doc/domino\\_notes/7.0/help7\\_admin.nsf/](http://www.lotus.com/1dd/doc/domino_notes/7.0/help7_admin.nsf/)

- ▶ Readme70.pdf (CD 版)
- ▶ 下記 IBM Training and certification サイトで提供される DB2 教育用リソース  
<http://www.ibm.com/software/sw-lotus/services/education.nsf/wdocs/educationhomepage>
  - Developing Applications with IBM Lotus Domino Enabled for DB2, E-learning Edition: Self Paced (N7D400SP)  
<http://www.ibm.com/software/sw-lotus/services/education.nsf/81F04813DC9CE3C7852566DA004C5CB2/A6DF60493B0D2B13852570A7006C7E9B>
  - Introducing IBM DB2 Concepts (N7D020)  
<https://education.lotus.com/rw/lewwschd.nsf/5FDC13DD71D1143185256575001E5133/E55E8003DCEC00FA8525701A006959C4>

**ヒント:** セットアップ中に、DB2 ユーザー名 の数を設定する必要があります。DB2 のユーザー名は、次の DB2 命名規則に準拠し、名前は常に大文字のみで指定する必要があります。

- ▶ Linux® および UNIX® ベースのシステムでは、ユーザー ID は最大 8 文字で指定できます。
- ▶ Microsoft Windows では、ユーザー名は最大 30 文字で指定できます。現行の Microsoft Windows NT®, Windows 2000、Windows XP、および Windows Server 2003 では、実質上の上限は 20 文字です。
- ▶ Windows NT、Windows 2000、Windows XP、および Windows Server 2003 に接続し、クライアント認証を使用しない Windows 以外の 32 ビット・クライアントでは、ユーザー名とパスワードが明示的に指定されている場合に 8 文字を超えるユーザー名がサポートされます。
- ▶ 使用できない名前および ID は次のとおりです。
  - USERS、ADMINS、GUESTS、PUBLIC、LOCAL、または SQL で予約されているすべての用語
  - IBM、SQL、または SYS で始まる言葉
  - アクセント付き文字が含まれる言葉

## 2.2 DB2 Access ビュー

Lotus Notes および Lotus Domino は、DB2 などのリレーショナル・データベースとはまったく異なるストレージ概念を持ちます。リレーショナル・データベースでは、適切に定義されたテーブル形式で情報が編成されていますが、Lotus Domino データベースでは、文書もしくは「notes」は階層のないファイル形式のフィールド・リストとして構成され、格納されます。このように情報が非構造化方式で格納されると、DB2 パースペクティブから Lotus Domino データへのアクセスは容易ではありません。この問題を解決するために、Lotus Domino 7 には DB2 Access ビューという共有リソースがあります。このリソースを使用すると、Lotus Domino のデータを DB2 のビューとして定義でき、構造化照会言語 (SQL) でクエリー可能なテーブルにデータを整理することができます。DB2 Access ビューはバックグラウンド設計要素であるため、エンド・ユーザーには表示されません。

DB2 Access ビューは、DB2 サーバーに格納されるデータベースでのみ使用可能です。DB2 Access ビューを作成する前に、ご使用の Lotus Domino サーバーを DB2 対応に設定し、DB2 システムでユーザー・データベースをホストするように指定する必要があります。ご使用のデータベースの複製をローカルで作成したり、DB2 対応でない Lotus Domino 7.0 Server 上のデータベースを複製したりしても、DB2 Access ビューはデータベースに残りますが、Lotus Domino Designer では使用または表示ができなくなります。

Lotus Notes データベースを DB2 対応にすると、Lotus Domino は DB2 データベース・スキーマを作成して、Lotus Notes データベースのデータとして、全ユーザーからアクセス可能な



データ・ビューをこのスキーマに保持します。スキーマ名は、NSF ファイル名をベースに命名されます。作成するすべての DB2 Access ビューもこのスキーマに格納されます。

DB2 Access ビューを作成すると、情報がコピーされ、DB2 テーブルに冗長的に保管されます。DB2 Access ビューでデータを作成および初期化すると、Lotus Domino はフリー・フォームの Lotus Domino データと DB2 Access ビューの構造化列のデータ間での更新を自動的に同期化します。このため、DB2 Access ビューをリフレッシュする必要はありません。

## 2.2.1 DB2 Access ビューの作成

DB2 Access ビューを新規作成するには、以下の手順に従ってください。Lotus Domino 7 Designer については、次の URL も参照してください。

[http://www.lotus.com/1dd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/1dd/doc/domino_notes/7.0/help7_designer.nsf)

1. DB2 対応データベースを Lotus Domino Designer 7.0 で開く
2. 設計ペインで、「共有リソース」→「DB2 Access ビュー」を選択します。
3. 設計アクション・ボタンの「新規 DB2 Access ビュー」をクリックします。図 2-3 に示すウィンドウが開きます。

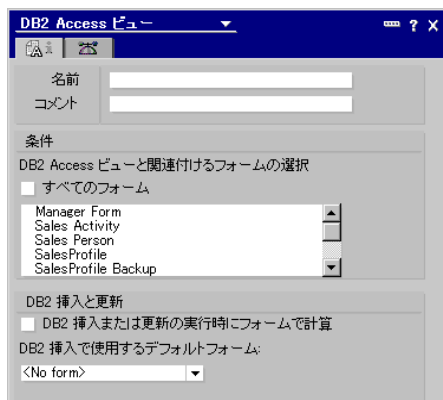


図 2-3 DB2 Access ビューの基本プロパティ・タブ

4. 「DB2 Access ビュー」ウィンドウで次のように実行します。
  - DB2 Access ビューに名前を割り当てます。
  - この DB2 Access ビューに関連付けられているフォームを 1 つ以上選択します。

文書の「フォーム」を指定するフィールドの内容に基づいて、含める文書を選択できます。すべてのフォームを選択するか、リストから使用可能なフォームをいくつか選択することもできます。文書にフォーム・フィールドがない場合、「すべてのフォーム」を選択すると、DB2 Access ビューにのみ含まれることになります。
  - 「DB2 挿入と更新」設定は、次のオプションを指定して INSERT または UPDATE SQL ステートメントを DB2 で実行する場合にのみ有効です。
    - 「DB2 挿入または更新の実行時にフォームで計算」

計算フィールドを含むフォームを選択し、SQL で文書を作成または更新する場合に計算式を使用するときにこのオプションを有効にします。
    - 「DB2 挿入で使用するデフォルトフォーム」

DB2 ユーザーは、データの挿入、更新、削除 (適正な権限を持つ場合) を DB2 の Access ビューで実行できます。Lotus Domino 7.0 では、SQL の INSERT ステートメントで作成する場合に選択可能な文書 (FORM ベース) のタイプは 1 つのみです。あらゆるタイプの文書 (FORM) は、SQL から更新または削除できます。

5. 図 2-4 に示すように、ウィンドウの「詳細」タブに切り替えます。

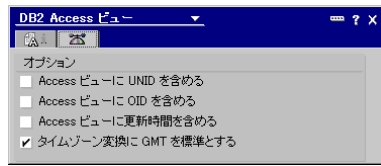


図 2-4 DB2 Access ビューの「詳細」タブ

このタブで選択可能なオプションは次のとおりです。

– 「Access ビューに UNID を含める」

このチェック・ボックスを選択すると、DB2 Access ビューから Lotus Notes 文書の文書固有 ID にアクセス可能になります。

– 「Access ビューに OID を含める」

OID は、Origination Identifier の省略形です。このオプションを選択して、Query ビューの SQL 式でこれを選択すると、ビューのエントリーを選択してテーブルとしてコピーするときに文書リンクを使用できます。

– 「Access ビューに更新時間を含める」

このオプションを選択すると、Lotus Notes で変更した時刻が DB2 Access ビューの各文書に記載されます。このタイム・スタンプは GMT に変更されることに注意してください。

– 「タイムゾーン変換に GMT を標準とする」

このオプションを選択すると、DB2 ビューのすべての日付と時刻とが GMT に標準化されます。このオプションは、さまざまなタイム・ゾーンのユーザーがアクセスする分散 DB2 アプリケーションで特に役立ちます。

6. 「DB2 Access ビュー」プロパティ・ウィンドウを閉じます。

7. 「フィールドの選択」アクションを選択すると、図 2-5 に示すウィンドウが開きます。

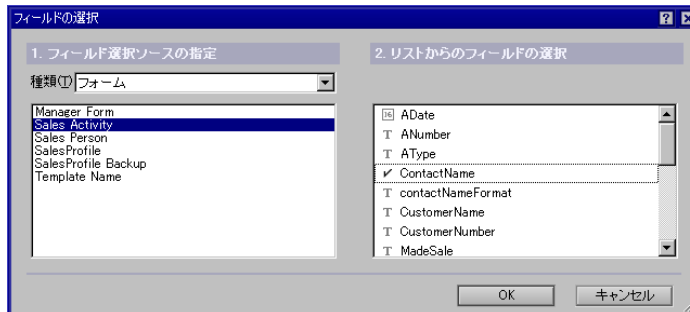


図 2-5 DB2 Access ビューの「フィールドの選択」ウィンドウ

8. 「フィールドの選択」ウィンドウで次の手順を実行します。

a. 各フォームに含めるフィールドを選択します。

Lotus Domino 7 では、DB2 Access ビューには最大で 84 列のみ含めることができます。これ以上のデータを格納する必要がある場合、複数の DB2 Access ビューを定義する必要があります。

b. 「OK」をクリックして選択内容を確定します。

選択した DB2 Access ビューのエントリーのリストがワーク・ペインに表示されます。

9. 別のフィールドを追加するために、「フィールドの挿入」アクションを選択します。

選択したフォームにない別のフィールドを追加する場合があります。「フォーム」フィールドなどがその例です。アクション「フィールドの挿入」を実行すると、このようなフィールドを手動で追加できます。このアクションをクリックするたびに、無名の DB2 エントリーがワーク・ペインに追加されます。

10. 図 2-6 のように、各 Access ビューのエントリーをダブルクリックしてプロパティを編集します。

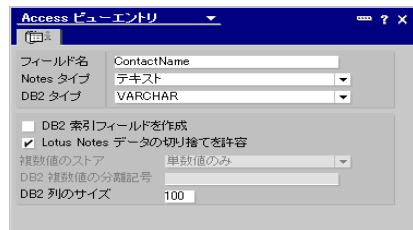


図 2-6 Access ビュー・エントリーのプロパティ

最初の 3 つは、「フィールドの挿入」アクションで追加されるフィールドに取り込むためのプロパティです。フォームで選択されるフィールドには、適切なプロパティが自動的に取り込まれます。このウィンドウには次の各フィールドがあります。

– 「フィールド名」

このフィールドの名前が DB2 Access ビューの DB2 列のフィールド名と付き合わされるため、この名前がフィールドの実際の名前として文書に表示される必要があります (そうでないと、一致するフィールドが検出されず、DB2 ビューの列が空白になります)。

**注:** DB2 Access ビューのフィールドを指定する方法として「フィールドの選択」を選択した場合は、プロパティでフィールド名を編集できます。ただし、文書のフィールド名に一致しなくなった名前にフィールド名を変更すると、DB2 で対応する列が空白になります。

– 「Notes タイプ」

Lotus Notes で使用する、このフィールドのデータ型を示します。

**注:** 式、リッチ・テキスト、およびリッチ・テキスト・ライトは、DB2 Access ビューで使用する Lotus Notes のデータ型としてサポートされていません。

– 「DB2 タイプ」

DB2 で使用する、このフィールドのデータ型を示します。Lotus Notes に、Lotus Notes 用に選択したデータ型に対応するデフォルト値が表示されます。

**注:** DB2 Access ビューに整数を定義している場合、または、SQL INSERT ステートメントまたは UPDATE ステートメントを使用して列を実数 (小数点が含まれる) で更新しようとした場合、DB2 は更新を実行する前にこの値を整数に変換するため、小数点以下の値は切り捨てられます。

DB2 Access ビューのすべてのエントリーに関連する属性は次のとおりです。

- 「DB2 索引フィールドを作成」：この列を DB2 索引フィールドとして作成し、これによってデータベース・レコードが高速検索用にキー指定されます。次のコマンドを実行すると、このオプションで単一の DB2 索引がテーブルの列に作成されます。

> CREATE INDEX ON TABLE x (coly)

このコマンド実行には一定のスペースが使用され、更新にはさらにリソースが必要になりますが、このようなクエリーを非常に高速で実行できるようになります。これは主に、次のように TABLE x で多数の行が取得されるためです。

> SELECT cola, colb, colc FROM x WHERE coly=<some value>

一部の Lotus Domino 開発者にとっては DB2 の索引作成は高度なトピックですが、Lotus Domino Designer では現時点で作成できない、多様な select ステートメントを最適化する索引が一部存在します。例えば、このコマンドは、次のクエリーを最適化します。

> CREATE INDEX ON TABLE x (cola, colb, colc)

この例では、索引はデータのソートに使用されます。このような処理では、索引がないと、テーブルの規模が大きくなった場合に最もリソースを消費する操作の 1 つとなります。

> SELECT cola, colb, colc FROM x WHERE cola = <some value> ORDER BY cola, colb, colc

- 「DB2 列のサイズ」：このプロパティーで DB2 Access ビューの列の長さを定義します。このフィールドの値の選択では、競合するターゲット間で妥協点の見極めが必要となる場合があります。一定のフィールドにできるだけ多くの値を表示する必要がありますが、最大列長が DB2 データベースにおける DB2 Access ビューの消費スペースに直接影響を与えることに注意してください。Lotus Notes のフィールドのマッピングでは、DAV テーブルに保存可能なデータの合計長に制限があります。データ・フィールドの合計長と DB2 への対応を含め、16,384 バイトを超えるデータは保存できません。16k を超えるデータを保存する必要がある場合は、テキスト・フィールドの一部を LONG VARCHAR にマッピングできます。フィールドを LONG VARCHAR にマッピングする場合、小さい「ロケーター」が「インライン」（上限の 16,384 バイトまでカウント）に保管され、実際のフィールド・データ（各フィールドで最大 32,000 バイト）はキャラクター・ラージ・オブジェクト (CLOB) として保管されます。CLOB を使用すると、DAV に十分すぎるほどのストレージ容量が提供されますが (84\*32,000)、データが「インライン」に適合しない場合は LONG VARCHAR のみを使用する必要があります。CLOB ストレージは、インライン・ストレージより処理が低速で、消費ディスク・スペースが多いためです。DB2 Access ビューを検証または作成する場合、DB2 Access ビューの定義が最大長を超過していないかどうかは Lotus Domino サーバーで確認されるため、長さを手動で計算する必要はありません。ただし、必要に応じて「インライン」データ長を概算する場合は、適宜表 2-1 のガイドラインを参照してください。

表 2-1 DB2 Access ビューのデータ型と長さ

型	DAV の長さ	コメント
VARCHAR	宣言長 + 5 バイト	例：VARCHAR(40) の場合、45 バイトのストレージが必要
LONG VARCHAR	25	インライン・ロケーター
DOUBLE	9	固定サイズ
TIMESTAMP	11	固定サイズ
DATE	5	固定サイズ
TIME	4	固定サイズ

型	DAV の長さ	コメント
INTEGER	9	固定サイズ

- 「Lotus Notes データの切り捨てを許容」：DB2 Access ビューのエントリに定義した最大列長よりフィールドのサイズが大きくなる場合があります。切り捨てを許可した場合、最大長に達すると、超過した部分が切り捨てられます。切り捨てを許可しない場合、上限に達すると、DB2 Access ビューへの取り込み中にコンソール・ログに次のメッセージが記録されます。「アクセステーブルのフィールドデータが定義した長さを超えています」

Lotus Domino 7 では、デフォルトで「Lotus Notes データの切り捨てを許容」が表示されます。DB2 Access ビューを使用する限り、すべての DB2 Access ビュー・エントリのテキスト・データに対し、このプロパティを有効にすることを強くお勧めします。ただし、例外があります。DB2 Access ビューの更新を DB2 で SQL 送信する場合は、この設定でデータを指定の長さに切り捨てる許可が与えられます。この場合、データの損失につながる可能性があります。

**重要：**すべての Access ビュー・エントリに対し、「Lotus Notes データの切り捨てを許容」オプションを選択することをお勧めします。

## 2.2.2 DB2 Access ビューへの TIMEDATES のエクスポート

Lotus Notes TIMEDATE は、日付、時刻、タイム・ゾーン、および夏時間調整インディケーターを保管できますが、DB2 のタイム・スタンプで保管できるのは日付と時刻のみです。DB2 では、すべての日付と時刻は、DB2 サーバーのタイム・ゾーンに対してローカルであると想定しています。このため、TIMEDATE を DB2 Access ビューにエクスポートするか、または Query ビューに DB2 で TIMESTAMPS を読み取った場合に（これらが DB2 Access ビューまたは他の DB2 テーブルのいずれからにかかわらず）、表示時刻が不正確になる可能性があります。

DB2 Access ビューに日付と時刻を保存するオプションは 2 つあります。ローカル時刻または GMT (グリニッジ標準時) への変更です。時刻の処理方法については、「Access ビューエントリ」プロパティ・ページの「詳細タブ」で、「タイムゾーン変換に GMT を標準とする」フィールドを必要に応じて設定してください。

**注：**既存の DB2 Access ビューに対してこの設定を変更した場合は、DB2 Access ビューを再構築する必要があります。

## 2.2.3 DB2 Access ビューへの日付範囲の挿入

SQL を使用して DB2 Access ビュー内の文書の日付範囲を挿入または更新する場合は、yyyy-mm-dd-hh.mm.ss.subsec または 2005-12-12-08.06.30.123456 などの完全な DB2 タイム・スタンプで日付範囲を指定する必要があります。この操作が失敗すると、挿入された値が NULL になります。

## 2.2.4 DB2 Access ビューのステータスの確認

Lotus Domino Designer で DB2 Access ビューを選択すると、データベース内のすべての DB2 Access ビューのリストが表示されます。リストの各 DB2 Access ビューの前には、図 2-7 に示すように、各 DB2 Access ビューのステータスを示すアイコンが表示されます。

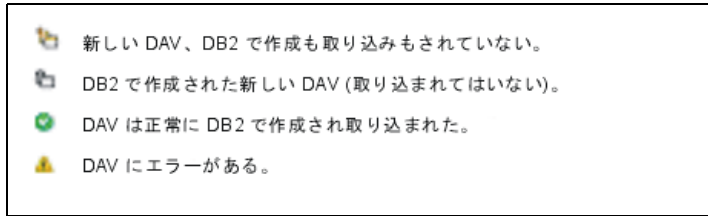


図 2-7 DB2 Access ビューのステータスを表すアイコン

## 2.2.5 DB2 Access ビュー作成時の他の考慮事項

データベースには、DB2 Access ビューをいくつでも作成できます。例えば、フォーム別に 1 ビューずつ作成できます。ただし、大きい DB2 Access ビューを 1、2 個作成して、アクセスするすべてのフォームやすべての文書のデータをここに格納する方法が容易です。DB2 Access ビューの列数を制限するには、複数のフォームで対応するフィールドで同一名を使用するようにしてください。

Lotus Domino のフィールドとデータ型には、非常に高い柔軟性があります。同一フォームを基本とする Lotus Notes と Lotus Domino の 2 つの文書でデータ型が異なる場合があります。この柔軟性という利点が、DB2 Access ビューを作成しようとする際に問題となる場合があります。DB2 には、データ型について非常に厳格な制約があります。多くの場合、データベースのデータをクリーンアップして、同じデータ型を持つ DB2 Access ビューで使用されるフィールドのすべての内容を確認する必要があります。

Lotus Notes 文書で特定のフィールドが見つからない、またはその値が空であることは問題ではありません。DB2 Access ビューでは、このフィールドは単純に NULL に設定され、すべてのクエリーで空の値が表示されます。リッチ・テキストおよび式のフィールドは DB2 Access ビューではサポートされていません。このいずれかのフィールド・タイプのフィールドを含めると、エラー・メッセージは表示されませんが、内容が NULL と表示されます。

DB2 Access ビューとそのエンティティ（テーブル、ビュー、トリガー、および索引）のサイズは、Lotus Notes データベースの合計ファイル・サイズには含まれません。したがって、DB2 対応の Lotus Notes データベースに大量の DB2 Access ビューを関連付けている場合、DB2 対応の Lotus Notes データベースの上限を超えたディスク・スペースが占有されることに注意が必要です。しかし、IBM 内では、DB2 Access ビューが大量のディスク・スペースを占有することに関する問題は報告されていません。社内の初期デプロイメントでは、大数の列を持つ DB2 Access ビューであっても、通常占有するのはデータベース・スペースの 2、3 % のみであることが判明しています。

## 2.2.6 DB2 Access ビューの ITSO Electronics アプリケーションへの導入

このシナリオの例では、ITSO Electronics で新機能の Query ビューを活用します。この機能については、18 ページの 2.3、「Query ビュー」で説明します。Query ビューの準備として、ここで DB2 Access ビューをいくつか作成します。この例では、アプリケーションの Lotus Notes データベースごとに DB2 Access ビューを 1 つずつ作成することにします。

最初に、ITSO Electronics Sales データベースに、DB2 Access ビューの SALES DAV を作成します。表 2-2 にフィールドの選択を示します。

表 2-2 Sales データベースに作成する DB2 Access ビューの SALES DAV

DB2 Access ビューに含まれるフィールド	フォーム
ANumber	「Sales Activity」
AType	「Sales Activity」

DB2 Access ビューに含まれるフィールド	フォーム
ContactName	「Sales Activity」
CustomerName	「Sales Activity」
CustomerNumber	「Sales Activity」
MadeSale	「Sales Activity」
Product1	「Sales Activity」
Product2	「Sales Activity」
Product3	「Sales Activity」
Product4	「Sales Activity」
SName	「Sales Activity」、「Sales Person」
SNumber	「Sales Activity」、「Sales Person」
SPhone	「Sales Person」
STitle	「Sales Person」
FORM	「Sales Activity」、「Sales Person」

次に、ITSO Electronics の Customer データベースに、CUSTOMERDAV という DB2 Access ビューを作成します。表 2-3 を参照してください。

表 2-3 Customer データベースに作成する DB2 Access ビューの CUSTOMERDAV

DB2 Access ビューに含まれるフィールド	フォーム
customerAddress	「Customer」
CustomerDescription	「Customer」
customerName	「Customer」
customerNumber	「Customer」
ownerName	「Customer」
FORM	「Customer」

最後に、ITSO Electronics の Products データベースに PRODUCTSDAV という DB2 Access ビューを作成します。

表 2-4 Products データベースに作成する DB2 Access ビューの PRODUCTSDAV

DB2 Access ビューに含まれるフィールド	フォーム
productDescription	「Product」
SupplierAddress	「Supplier」
SupplierDescription	「Product」、「Supplier」
SupplierName	「Supplier」
SupplierNumber	「Product」、「Supplier」
productName	「Product」

DB2 Access ビューに含まれるフィールド	フォーム
productNumber	「Product」
FORM	「Product」、「Supplier」

作成した各 DB2 Access ビューは、18 ページの 2.3、「Query ビュー」で Query ビューを作成するときに使用します。

## 2.3 Query ビュー

Query ビューは、新しいタイプの Lotus Notes データベース (NSF) ビューです。このビューには、構造化照会言語 (SQL) で記述したクエリーを使用してデータを取り込みます。一見したところ、Query ビューは、単に別のデータの収集方法または表示方法のように見えます。しかし注意して見ると、Query ビューは Lotus Domino におけるデータの新たな表示方法であることがわかります。初回は、複数の Lotus Notes データベースの情報を 1 つのビューに表示する場合があります。この場合、次の SQL 構文が使用できます。

### ▶ As Join

20 ページの 2.3.2、「join を使用した Query ビューの作成」に示されるように、複数の DB2 Access ビューの関連するエントリー外の文書を Query ビューの 1 行に結合します。

### ▶ As Union

複数の DB2 Access ビューのエントリーを 1 つのビューとして組み立てます。手順については、21 ページの 2.3.3、「union を使用した Query ビューの作成」で説明します。

### ▶ Group by

### ▶ Having

このクエリーは動的に構築しますが、SQL はデータベースのスペースを占有しません。同様に、SQL ステートメントは動的に構築できますが、Lotus Notes の式言語を使用します。ユーザーは、@Username や @Today などの @ 式を自由に使用できます。従来の Lotus Notes ビューのような索引の問題は発生しません。@Prompt などのユーザーのやりとりにより、ビューの内容を制御することも可能です。手順については、22 ページの 2.3.4、「Query ビューを @ 式で動的に作成する」を参照してください。

### 2.3.1 Query ビューの作成

Query ビューは、Lotus Domino の文書データを直接には表示できません。Lotus Domino データを表示するには、11 ページの 2.2.1、「DB2 Access ビューの作成」で説明するように、まず DB2 Access ビューを作成する必要があります。DB2 Access ビューの配置後、次の手順で Query ビューの作成を開始できます。

1. Lotus Domino Designer で、「ビュー」に移動します。「新規ビュー」アクション・ボタンをクリックします。図 2-8 に示すように、「ビューの作成」ウィンドウで、選択条件として「SQL クエリーの使用」を選択します。





図 2-8 「ビューの作成」 ウィンドウ

- 次に、ビューの SQL ステートメントを入力します。 22 ページの 2.3.4、「Query ビューを @ 式で動的に作成する」に示すように、式言語に基づいて動的に SQL ステートメントを作成できます。この機能を有効にするために、例 2-1 に示すように、静的 SQL ステートメントを引用符で囲む必要があります。

例 2-1 SQL ステートメントは引用符で囲む必要がある

---

“SELECT \* FROM SALES.SALESDAV”

---

データベースへの DB2 Access ビューのアドレス指定は、DB2 スキーマ名および DB2 Access ビュー名を使用して、DB2SCHEMANAME.DAVNAME のように行います。例 2-1 では、DB2 Access ビューに SALES.SALESDAV という名前が付けられています。サーバーにデータベースを配置するときは、通常「.NSF」の直前のファイル名の一部が DB2 スキーマ名です。上記の例では、SALES.NSF が SALES になっています。この DB2 スキーマ名が既に使用されている場合は、DB2 スキーマ名に SALES2 や SALES3 などの数字が付けられます。

**重要** :Query ビューの SQL ステートメントを作成するときは、この時点で DB2 スキーマ名が最終的な DB2 スキーマ名になるかどうかはわかりません。スキーマ名が適切かどうかを確認するためのベスト・プラクティスは、22 ページの 2.3.4、「Query ビューを @ 式で動的に作成する」で説明する動的 SQL ステートメントを使用する方法です。

- SQL ステートメントが終了すると、ビューの列を定義できます。通常のビューと同様に行ってください。
- 設計をプレビューするには、ツールバーの「Notes プリビュー」ボタンをクリックします。Lotus Domino 7 では、ビュー設計の「更新」ボタンをクリックしてもビューは表示されません。

例 2-2 のように、読みやすくするために SQL ステートメントを複数行に分割して記述すると、エラー・メッセージが表示されます。

例 2-2 SQL をこのように記述するとエラー・メッセージが表示される

---

“SELECT  
SALES.SALESDAV.CUSTOMERNAME  
FROM  
SALES.SALESDAV”

---

今回の ITSO Electronics アプリケーションの作成過程で、例 2-3 のように + で連結された行を含む SQL を記述すれば、読みやすさは変わらずに、Lotus Domino 7 で動作することが明らかになっています。各サブステートメントは引用符で囲む必要があります。

例 2-3 この方法で SQL ステートメントを記述すると読みやすくエラーにならない

```
"SELECT "+
"SALES.SALESDAV.CUSTOMERNAME "+
"FROM "+
"SALES.SALESDAV"
```

SQL では、データ・ソースの別名を使用して SQL ステートメントを簡単に記述することもできます。20 ページの例 2-3 に示したコードは、例 2-4 のように記述することが可能です。この例ではほとんど違いはありませんが、より複雑な例の場合に理解しやすくなります。

例 2-4 別名を利用したステートメントの例

```
"SELECT "+
"S.CUSTOMERNAME "+
"FROM "+
"SALES.SALESDAV S"
```

## 2.3.2 join を使用した Query ビューの作成

Lotus Domino 7 では、複数の DB2 Access ビューのデータを 1 つの Query ビューに結合できます。このとき、DB2 Access ビューが同じデータベース内にあるかどうかは関係ありません。各ビューのエントリは、DB2 Access ビューで使用可能な複数の文書からの結合データとして表示されます。

図 2-9 に、2 つの DB2 Access ビュー、ORDERDAV および CUSTOMERDAV の情報が SQL ResultSet に結合された様子を示します。

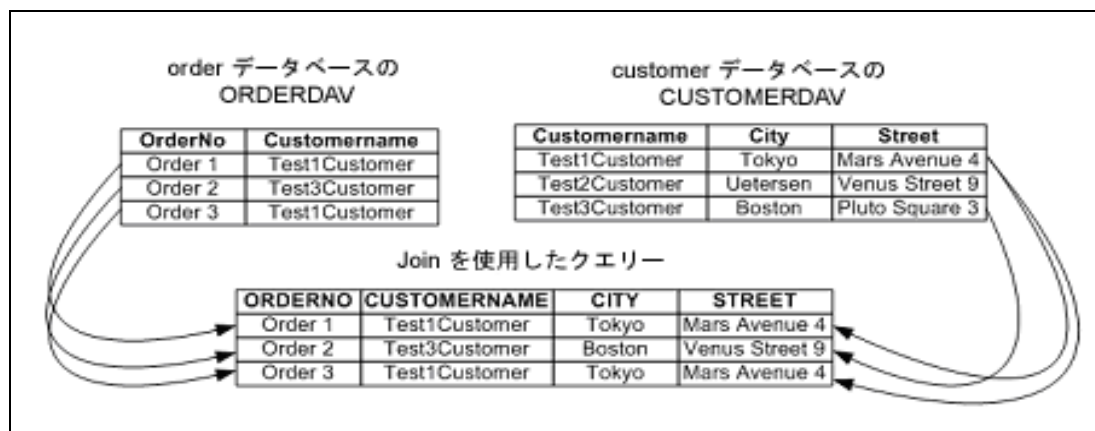


図 2-9 join を使用した Query ビュー

例 2-5 は、上記の目的で使用する当該 SQL ステートメントです。

例 2-5 Query ビューを作成するために join を使用した SQL ステートメント

```
"SELECT "+
"S.AType, S.Anumber,S.Product1 , C.CUSTOMERNAME, C.OWNERNAME, "+
" S.CUSTOMERNUMBER, S.MadeSale "+
"FROM "+
"SALES.SALESDAV S "+
"Left outer JOIN "+
```

```
"CUSTOMER.CUSTOMERDAV C "+
"ON "+
"S.CustomerNumber=C.CustomerNumber "+
"WHERE "+
"C.OWNERNAME<>' '"
```

図 2-10 に、20 ページの例 2-5 の SQL ステートメントで作成した Customer Sales ビューを示します。

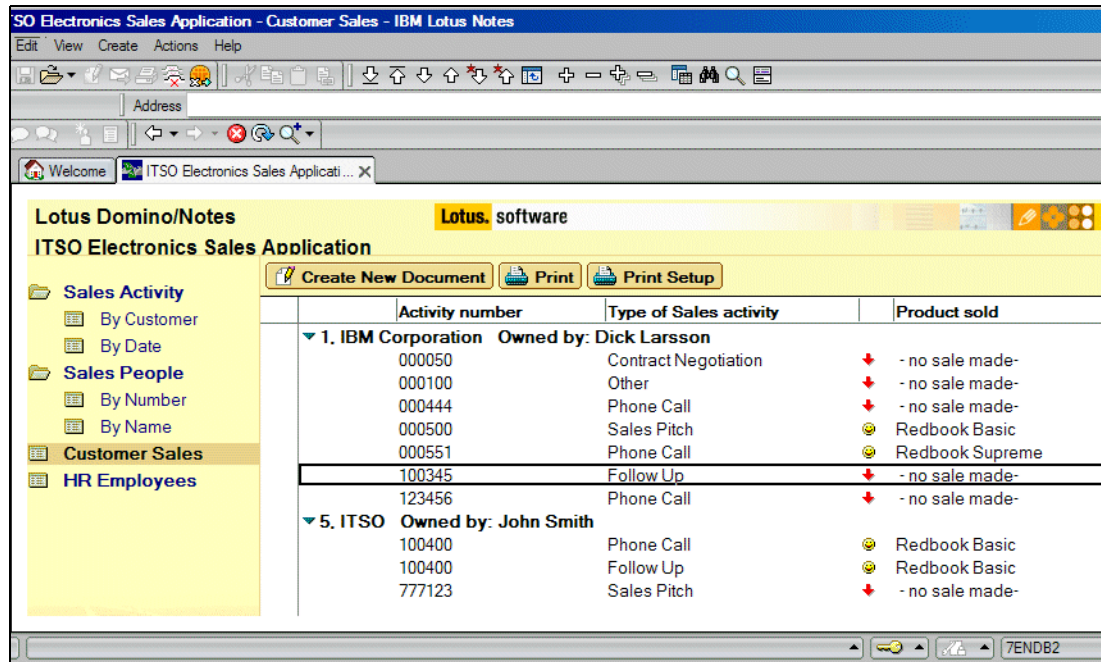


図 2-10 join を使用して作成した Query ビュー

### 2.3.3 union を使用した Query ビューの作成

ITSO Electronics は、外部のすべての企業の連絡先を 1 つのビューに統合して表示しようと考えています。サプライヤー情報は Products データベースに保管され、一方で、カスタマー・データは Customer データベースに保管されています。Lotus Notes および Lotus Domino リリース 6 では、単一のデータベースにデータを保存しない限り、この目的を達成できませんでした。Lotus Domino 7 では、SQL ステートメントの union を使用して Query ビューを作成できます。図 2-11 に、union で別個の DB2 Access ビューから Query ビューを作成する方法を示します。

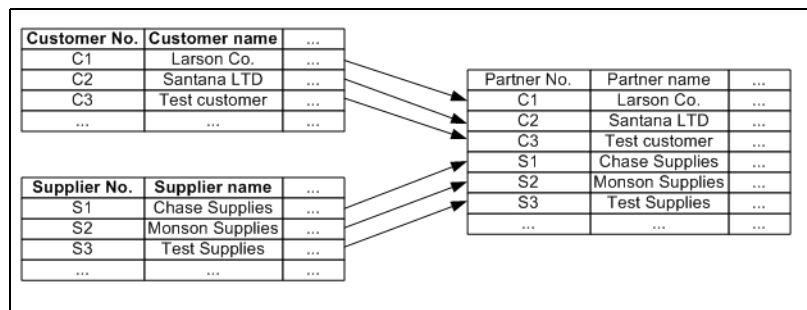


図 2-11 union を使用した Query ビュー

22 ページの例 2-6 に、SQL ステートメントを使用して ITSO Electronic の外部連絡先ビューのデータを収集する方法を示します。union を使用するためには、選択したそれぞれの式で、同一列名を持つ同一数の列を戻す必要があります。また、各結果セットで対応する列は、フィールドのタイプが一致していなければなりません。

例 2-6 union を使用した SQL

```
"SELECT "+
"C.CUSTOMERNUMBER as NO, C.CUSTOMERNAME as EXTERNALNAME, "+
"C.CUSTOMERADDRESS as ADDRESS, C.CUSTOMERDESCRIPTION as Description, "+
"C.Form as form "+
"FROM "+
"CUSTOMER.CUSTOMERDAV C "+
"WHERE "+
"C.FORM='Customer' "+
"UNION ALL "+
"SELECT "+
"S.SUPPLIERNUMBER as NO, S.SUPPLIERNAME as EXTERNALNAME, "+
"S.SUPPLIERADDRESS as ADDRESS, S.SUPPLIERDESCRIPTION as Description, "+
"S.Form as form "+
"FROM "+
"PRODUCTS.PRODUCTDAV S " +
"WHERE "+
"FORM ='Supplier'"
```

図 2-12 に、ITSO Electronics の複数のデータベースから例 2-6 の SQL コードでカスタマー・データとサプライヤー・データを取り出し、この両方を表示した Query ビューを示します。

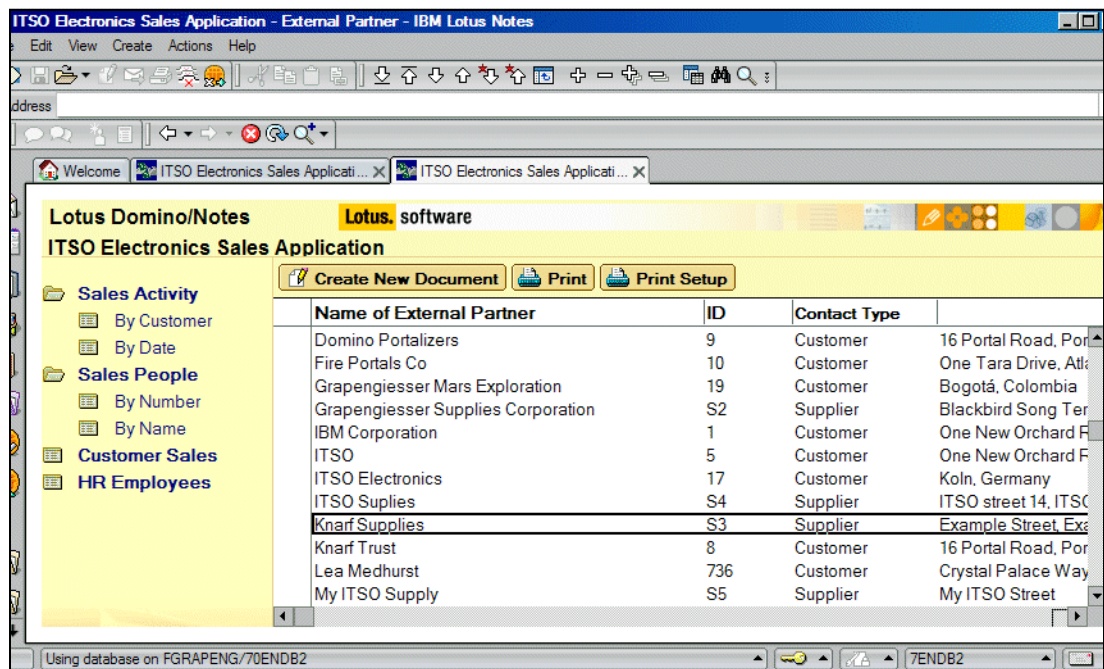


図 2-12 union を使用して作成した Query ビュー

### 2.3.4 Query ビューを @ 式で動的に作成する

これまでの例で使用した SQL ステートメントは、すべて静的ステートメントです。しかし多くの場合、Query ビューを開いた直後に SQL ステートメントを作成するのが望ましいです。Lotus Domino 7 では、@ 式を使用した Query ビューの計算を実行できます。

23 ページの例 2-7 に、DB2 Access ビューの SALES DAV からカスタマー「ITSO」のすべての営業活動を収集する単純な SQL ステートメントの例を示します。これを実行するには、フィルターの S.CUSTOMERNAME='ITSO' を使用します。

#### 例 2-7 静的 SQL ステートメント

```
"SELECT "+
"S.CUSTOMERNAME, S.ATYPE,S.ANumber , S.MADESALE "+
"FROM "+
"SALES.SALES DAV S "+
"WHERE S.FORM='Sales Activity' AND S.CUSTOMERNAME='ITSO'"
```

この静的ステートメントの代わりに、ビューを開くときに企業名を動的に選択できます。これを行うには、コードに **@Prompt** コマンドを追加して、エンド・ユーザーが表示する企業の名前を入力できるようにします。詳細については、例 2-8 を参照してください。

#### 例 2-8 式言語を使用した SQL ステートメントの動的アセンブル

```
_CUSTOMER:=@Prompt([OKCANCELEDIT];"Company selection"; "Please enter Company name you would like to search for.;"");
"SELECT "+
"S.CUSTOMERNAME, S.ATYPE,S.ANumber , S.MADESALE "+
"FROM "+
"SALES.SALES DAV S "+
"WHERE S.FORM='Sales Activity' AND S.CUSTOMERNAME='"+_CUSTOMER+"'"
```

ユーザーがこのマシンを起動するたびに、クエリーするカスタマー名の入力を求めるプロンプトが表示されます。

18 ページの 2.3.1、「Query ビューの作成」で説明したように、データベースが別のサーバーにコピーまたは複製された場合は、このデータベースの SQL スキーマ名が同じままであるかどうか確認できません。このため、SQL ステートメントでデータベース・スキーマに静的参照しないことをお勧めします。Lotus Notes および Lotus Domino 7 では、この制限の回避策として、**@DB2Schema** 関数を使用する方法があります。詳細については、92 ページの 4.5、「式言語に対する追加機能」を参照してください。現在のデータベースの SQL DB2Schema を取得するには、**@DB2Schema** コマンドとともに **@DBName** 関数を使用します。これらのコマンドが例 2-7 の SQL ステートメントに統合されている場合、例 2-9 のようなコードを使用します。

#### 例 2-9 現在のデータベースを動的に計算するスキーマ

```
_Schema:= @DB2Schema(@DBName);
"SELECT "+
"S.CUSTOMERNAME, S.ATYPE,S.ANumber , S.MADESALE "+
"FROM "+
_Schema +".SALES DAV S "+
"WHERE S.FORM='Sales Activity' AND S.CUSTOMERNAME='ITSO'"
```

これ以外の場合、別のデータベースの DB2 Access ビューにアクセスする SQL ステートメントを記述する必要があるかもしれません。このデータベースが現在のデータベースと同一フォルダーにあると想定した場合、例 2-7 に基づいた SQL ステートメントは、例 2-10 の参照と同様のものになります。

#### 例 2-10 Lotus Domino サーバーと同一フォルダーにあるデータベースへの動的参照

```
_pathOnly:=@LeftBack(@Subset(@DbName;-1);"¥¥");
_path:=@If(_pathOnly="";"";_pathOnly+"¥¥");
_Server:=@Subset(@DbName;1);
_Schema:= @DB2Schema(_Server:@Trim(_path+"Sales.nsf"));
"SELECT "+
```

```
"S.CUSTOMERNAME, S.ATYPE,S.ANumber , S.MADESALE "+  
"FROM "+  
_Schema +".SALESDAV S "+  
"WHERE S.FORM='Sales Activity' "
```

---

SQL ステートメントを作成時に、データベースにアクセスするパスが特定できない場合、次のいずれかのコマンドを使用すると、データベース・パスに関する情報を構成文書から得ることができます。

- ▶ @DBLookup
- ▶ @Environment
- ▶ @GetProfileField

### 2.3.5 Query ビューから文書を開く

従来の Lotus Notes のビューでは、ビュー・エントリは、それぞれが文書に直接リンクします。したがって、各ビューから文書をオープン、コピー、および削除できます。SQL ステートメントを基本とする Query ビューでは、必ずしもそうではありません。Query ビューから文書をオープン、コピー、および削除するには、事前に一定の条件を満たす必要があります。Query ビューの行は、クエリに DB2 Access ビューの #NOTEID 列が含まれていて、この #NOTEID が現在開かれている Lotus Notes データベースの有効な Note ID の場合にのみ、Lotus Notes データベースの文書にリンクします。

例えば、オリジナル・サンプルの Query ビューで、次のクエリを実行します。

```
> SELECT * FROM SALES.SALESDAV
```

この場合、DB2 Access ビューで #NOTEID 列が既に選択されているため、Query ビューの行とこれらの行に関連付けられている Lotus Notes 文書との間にリンクが作成されます。次のクエリでは、DB2 Access ビューで #NOTEID が選択されていないため、この Query ビューの行は、関連付けられている Lotus Notes 文書にリンクされず、ユーザーがこれを操作 (ダブルクリックなど) しようとする、エラーが生成されます。

```
> SELECT CUSTOMERNAME FROM SALES.SALESDAV
```

DB2 Access ビューから限定された数の列を選択している場合、#NOTEID 列を明示的に選択する必要があります。例えば、次のように実行します。

```
> SELECT CUSTOMERNAME, #NOTEID FROM SALES.SALESDAV
```

複数の DB2 Access ビューにわたり JOIN を実行している場合は、現在の Lotus Notes データベースで有効な Note ID である #NOTEID 列を明示的に選択してください。

ビューおよび Lotus Notes 文書の間の基本リンクは #NOTEID に基づいています。しかし、コピーまたは文書リンクの作成などを行う関数では、文書のオリジネーター ID (OID) が必要です。アプリケーションでこの機能が必要な場合 (または Query ビューで通常のビューにより近い操作が必要な場合)、DB2 Access ビューに #OID がエクスポートされるようにして (Designer で DB2 Access ビューの詳細プロパティを使用)、クエリに #OID を組み込んでください。

複数の DB2 Access ビューからデータを選択している場合、次のようにどの DB2 Access ビューでこれらの列を選択するかを明示的に指定する必要があります。

```
SELECT S.AType, S.ANumber,S.Product1 , C.CUSTOMERNAME, C.OWNERNAME, C.#NOTEID, C.#OID,  
S.CUSTOMERNUMBER, S.MadeSale FROM SALES.SALESDAV S Left outer JOIN CUSTOMER.CUSTOMERDAV C  
ON S.CustomerNumber=C.CustomerNumber WHERE C.OWNERNAME<>''
```

### 2.3.6 Web ブラウザーで Query ビューを開く

Web ブラウザーから Query ビューを開く機能は Lotus Domino 7.0.2 から利用可能になりました。Lotus Domino 7.0.2 より前のリリースでは利用できませんのでご注意ください。

### 2.3.7 Query ビューおよび LotusScript

Query ビューでは、ビュー内に文書を開くことができ、LotusScript を使用して開いた文書にアクセスすることもできます (24 ページの 2.3.5、「Query ビューから文書を開く」を参照してください)。この場合は、例 2-11 などの簡単なスクリプトで操作できます。Query ビューから文書を開けない場合、このスクリプトはマークの付いた行に設定されていない Object 変数をスローします。

例 2-11 ビューをループする LotusScript の簡単なスクリプト

---

```
Sub Click(Source As Button)
    Dim session As New notesession
    Dim db As NotesDatabase
    Dim view As NotesView
    Dim doc As notesdocument
    Dim Item As notesitem
    Set db=session.currentdatabase
    Set view =db.Getview("<A View>")
    Set doc=view.GetFirstDocument ' <=====
    Do While Not doc Is Nothing
        Set Item =doc.GetFirstItem("<I T E M   I N   D A T A B A S E>")
        Print(item.text)
        Set doc=view.GetNextDocument(doc)
    Loop
End Sub
```

---

LotusScript で文書を開くことを許可していない Query ビューのデータにアクセスする方法は他にもあります。例 2-12 に、ビュー・エントリー・クラスを介してこのようなビューにアクセスする方法を示します。

例 2-12 ビュー・エントリーにアクセスするための簡単なスクリプト

---

```
Sub Click(Source As Button)
    Dim session As New notesession
    Dim db As NotesDatabase
    Dim view As NotesView
    Dim doc As notesdocument
    Dim Item As notesitem
    Dim entry As NotesViewEntry
    Set db=session.currentdatabase
    Set view =db.Getview("<your viewname>")
    Set Entry = View.GetEntryByKey( "<viewEntryName>", True)
    MsgBox(entry.ColumnValues(1))
End Sub
```

---

### 2.3.8 ITSO Electronics アプリケーションの組み込み Query ビュー

他のビューと同様、Query ビューもフォーム、サブフォーム、ページ、または文書に組み込むことができます。ITSO Electronics アプリケーションでも、Query ビューの Customer Sales のバリエーションを Sales Person フォーム内に組み込んでいます。この Sales Person 文書にアクセスするユーザーは、26 ページの図 2-13 のように、文書を閉じることなく Customer Sales ビューの情報を簡単に表示できます。組み込みビューの作成方法の詳細については、次の URL で入手可能な「Lotus Domino 7 Designer Help」を参照してください。



The screenshot shows the Lotus Notes Designer interface. At the top, there's a toolbar with 'Edit' and 'Close' buttons. Below that is a 'Lotus software' header. The main area displays a 'Sales Person' form with the following fields:

Name	Dick Larsson
Employee Number	111888
Job Position	Sales Person
Phone Number	617-555-8888
Region	South East

Below the form is a query view titled 'Sales Activity Number' and 'Sales Activity Type'. It shows a list of activities for 'Dick Larsson' under 'IBM Corporation'.

Sales Activity Number	Sales Activity Type
▼ IBM Corporation	
▼ Owned by: Dick Larsson	
000500	Sales Pitch
100345	Follow Up
123456	Phone Call
000100	Other
000050	Contract Negotiation
000551	Phone Call
000444	Phone Call

図 2-13 ITSO Electronics アプリケーションの Sales Person フォームに組み込まれた Query ビュー

## 2.3.9 Query ビューに関する詳細な考慮事項

このセクションでは、Query ビューに関する詳細な考慮事項について説明します。

### 多値項目の分離

DB2 Access ビューはテキスト・リストをテキスト・ストリングとして保管しますが、Query ビューはデータを DB2 から直接読み込むため、このテキストがそのまま Query ビューからの戻り値になります。このため、Query ビューから分離する多値項目がある場合は、多値の列で @Explode 式を使用する必要があります。さらに、多数のデータ行を処理するようにビューをセットアップするには、「ビュー」プロパティ → 「スタイル」タブで、「行の高さ」と「行の高さを内容に合わせる」の各オプションを設定してデータが処理されるようにします。

### Query ビューの更新

Query ビューは Lotus Notes とフェデレーテッド・データを組み合わせて構築されるため、Query ビューの更新方法は、追加または変更されたデータの性質やその追加方法によって異なります。

### SQL クエリーの最大行数の設定

デフォルトでは、SQL クエリーから戻される行の最大数は 500 です。この値を設定するには、次の 2 つの方法があります。

- ▶ NOTES.INI パラメーターの DB2QueryViewRowLimit=value を使用する  
この値を 0 にすると、戻される行数が「上限なし」に設定されます。
- ▶ 「ビュー」プロパティ・ウィンドウの「オプション」タブで、「SQL クエリーからのリターンの最大行」を選択し、行数を指定します。27 ページの図 2-14 を参照してください。



両方とも設定された場合は、値の小さい設定が使用されます。

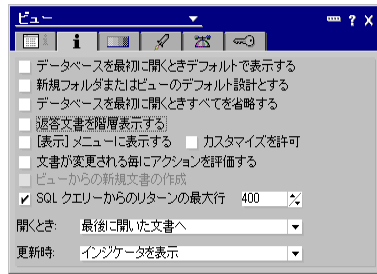


図2-14 「ビュー」プロパティ・ウィンドウの「オプション」タブ

## 2.4 DB2 でフェデレートされたデータについて

これまでは、DB2 Access ビューのクエリーにより Lotus Domino データを表示する Query ビューについて説明してきました。一方、DB2 および Lotus Domino の統合アーキテクチャの強力なコンポーネントである、DB2 フェデレーションを使用すると、DB2 データベースから認識できるすべてのデータが Query ビューに表示されるようにビューを構築できます。Lotus Domino データベースでフェデレーションをサポートするように構築することで、フェデレートしたデータを Lotus Domino データベース以外のデータベース (Oracle、Microsoft SQL、または他の DB2 データベース) に保管しても Lotus Domino で使用可能にすることができます。Query ビューは、Lotus Domino データのみ表示、フェデレーテッド・データのみ表示、または Lotus Domino データとフェデレーテッド・データを組み合わせた表示などの設定で作成できます。

### 2.4.1 DB2 でフェデレーションを有効にするには

DB2 対応の Lotus Domino データベースでフェデレーションをサポートするように構成すると、現在のデータベース (Domino ディレクトリでサーバー文書の DB2 タブに指定したデータベース) 以外のオブジェクトにアクセスしたり、フェデレーテッド・オブジェクトとしての参照対象を作成したりすることができます。これらの DB2 フェデレーテッド・オブジェクトを使用すると、多様な DB2 データベースの他の DB2 テーブルにアクセスできるだけでなく、他のデータベース管理システム (DBMS) にもアクセスできます。

ITSO Electronics アプリケーションでは、従業員情報が含まれている、人事 (HR) 部門所有の別の DB2 データベースの情報にアクセスするフェデレーテッド Query ビューを新規作成します。次の手順は、フェデレーションを有効にする DB2 サーバーの構成方法を示します。

#### 1. Query ビューが動作するように、環境変数

DB2\_ALLOW\_SETAUTH\_WITH\_REMOTECONNECT を設定します。次のパスおよびコマンドを使用します。

```
C:\Program Files\IBM\SQLLIB\BIN>db2set -i DB2 DB2_ALLOW_SETAUTH_WITH_REMOTECONNECT=1
```

ここで、-i には、ユーザーの DB2 インスタンスの名前 (通常、このシナリオ同様 DB2) を指定します。

#### 2. DB2 を再起動して、環境変数を有効にします。

DB2 のフェデレーションを有効にするには、次のパスおよびコマンドで、DB2 Database Manager の構成も更新する必要があります。

```
C:\Program Files\IBM\SQLLIB\BIN>db2 update dbm cfg using FEDERATED YES
```

- DB2 コントロール・センターで、ユーザーの Lotus Domino データベースに接続します。 Lotus Domino データベースでフェデレーションのサポートが有効になっていることを確認してください。ユーザーの Lotus Domino データベースに DB2 フェデレーションが設定されているか確認するには、DB2 インスタンスを右クリックして、「**パラメーターの構成**」を選択します。図 2-15 に示すように、「DBM 構成」ウィンドウが開きます。「環境」セクションをスクロールダウンしてキーワード FEDERATED を検索し、値が図 2-15 のように「YES」に設定されていることを確認します。値が「NO」の場合、手順 1 で **db2set** コマンドを実行していれば保留されている値は「YES」のため、FEDERATED キーワードの値が「YES」に設定されるように DB2 を再起動する必要があります。

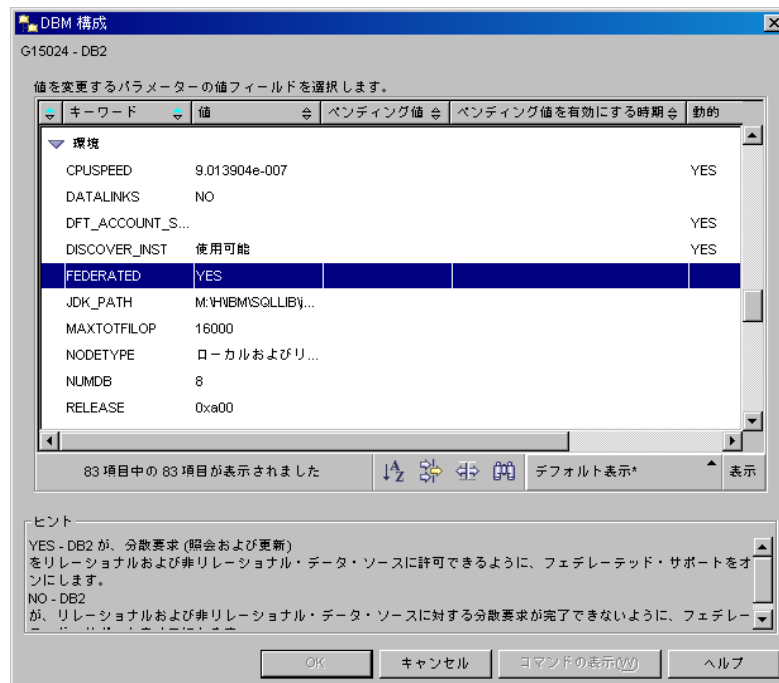


図 2-15 DB2 コントロール・センター: DBM 構成

4. このシナリオでは、人事 (HR) 部門の DB2 データベースである外部 DB2 データベース用、または他の任意のサード・パーティ・オブジェクト用に DB2 でフェデレーション・ラッパーを作成するには、DB2 コントロール・センターに戻って Lotus Domino データベースから、図 2-16 のように「フェデレーテッド・データベース・オブジェクト」フォルダーに移動し、「新規ラッパーの作成」アクションをクリックします。

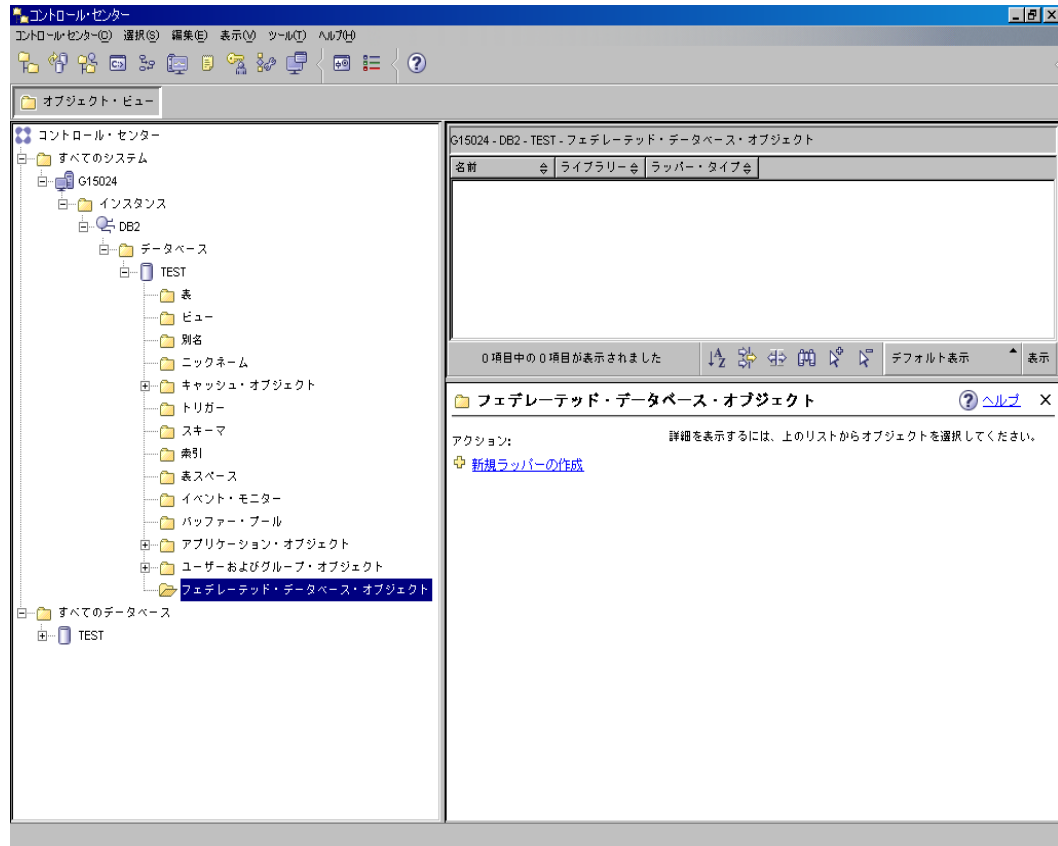


図2-16 DB2 コントロール・センターのフェデレーテッド・データベース・オブジェクト

5. 図 2-17 に示すように、「ラッパー作成」ウィンドウが開きます。「ラッパー」タブで、「データ・ソース」ドロップダウン・リストから、このシナリオで使用する「DB2 UDB」を選択します。設定した値 DB2WRAP でラッパー名を指定します。「OK」をクリックします。



図2-17 「ラッパー作成」ウィンドウ

注：ラッパーの詳細、およびDB2におけるフェデレーテッド・データのサポートの有効化については、次の URL の DB2 Information Center を参照してください。

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

6. ラッパーを作成すると、図 2-18 のように、DB2 コントロール・センターの「フェデレーテッド・データベース・オブジェクト」以下にラッパーが表示されます。「サーバー定義」フォルダーで、「新規サーバー定義の作成」アクションを選択します。31 ページの図 2-19 に示すように、「サーバー定義の作成」ウィンドウが開きます。

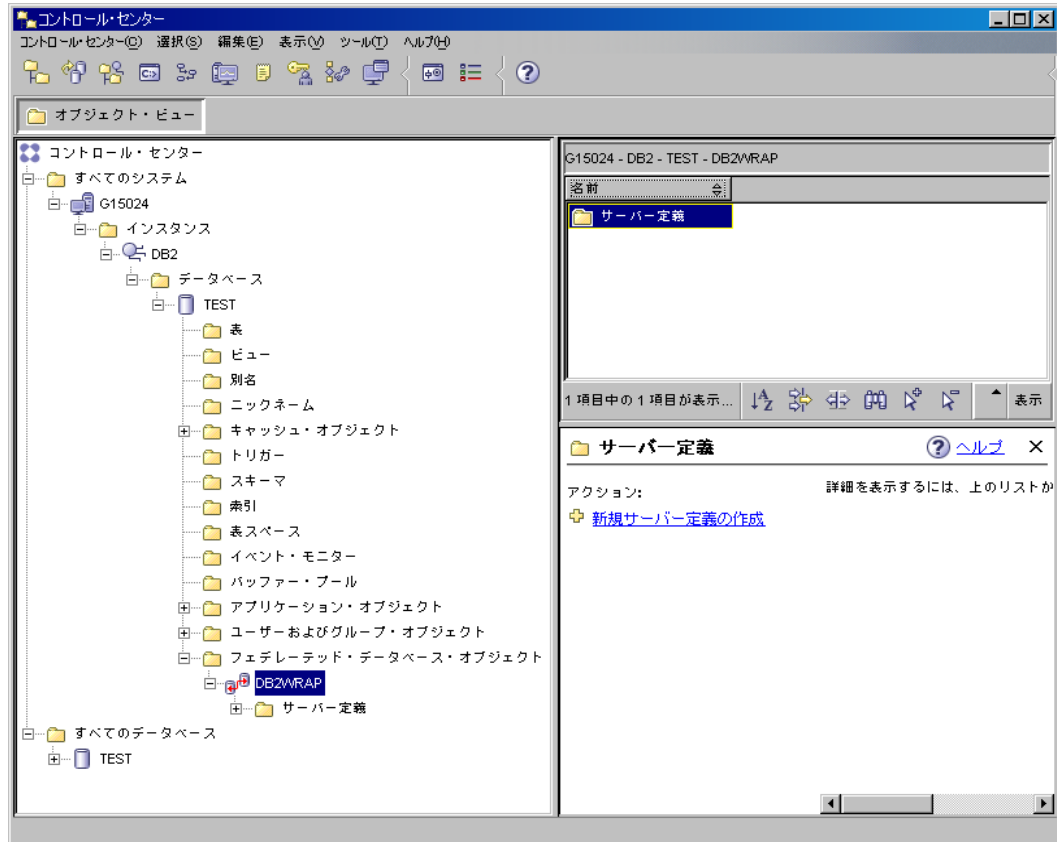


図 2-18 DB2 コントロール・センターのサーバー定義

7. 「サーバー定義の作成」ウィンドウで、「発見」をクリックします。オブジェクトのリストが表示されます。適切なオブジェクトを選択します。このシナリオでは、図 2-19 のように、**SAMPLE** データベースを選択します (HR DB2 データベースの名前)。

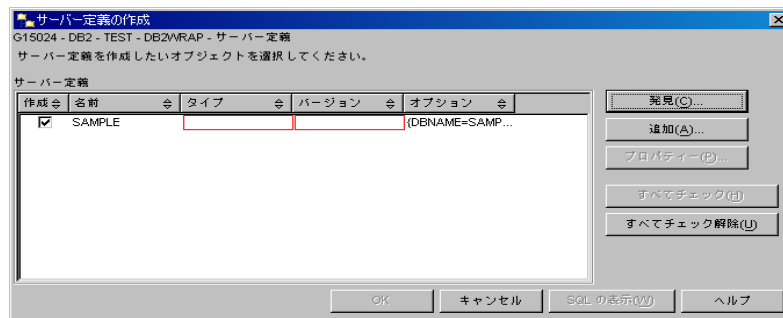


図 2-19 サーバー定義の作成: 「発見」 ボタン

8. 「プロパティ」をクリックすると、図 2-20 のように、「サーバー定義の作成」ウィンドウが開きます。「サーバー定義」タブのすべてのフィールドに関連する値を入力します。このシナリオでは、図 2-20 に表示されているすべてのフィールドです。「タイプ」フィールドでは、ドロップダウン・リストから「DB2/UDB」を選択し、「バージョン」フィールドでは、ドロップダウン・リストから「8.2」を選択します。「ユーザー ID」フィールドおよび「パスワード」のフィールドでは、外部の DB2 データベースにアクセスする DB2 ユーザー名 (または DB2 administrator) を使用します。このシナリオでは、DB2 HR のデータベース名です。

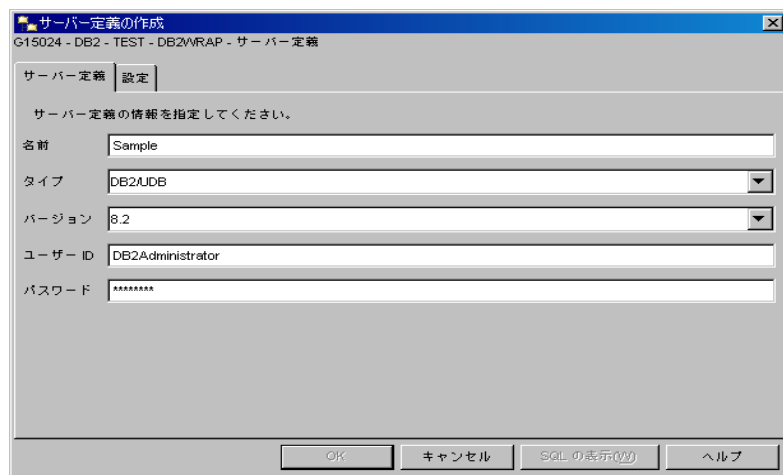


図 2-20 「サーバー定義の作成」ウィンドウ

9. 図 2-21 のように、「設定」タブに移動し、DBNAME オプションの「値」フィールドに DB2 データベースの名前を入力します。このシナリオでは、これは HR DB2 データベース名の SAMPLE です。ここで「OK」ボタンが使用可能になります。「OK」をクリックします。

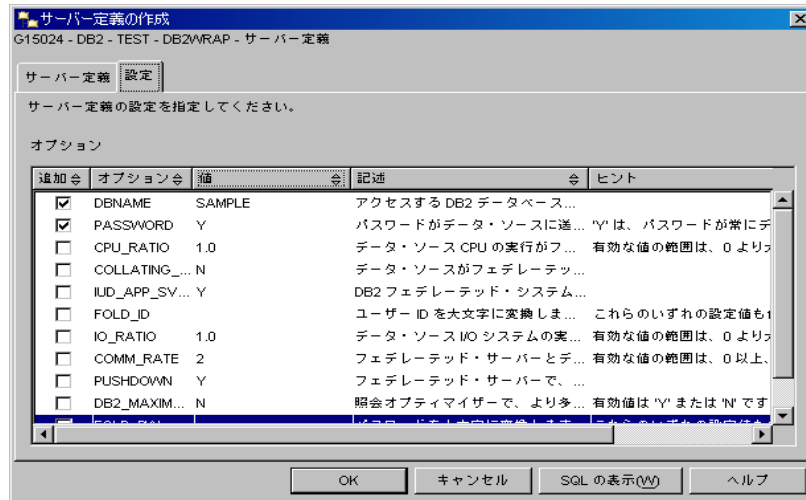


図 2-21 「サーバー定義の作成」：「設定」タブ

10. 次に、ユーザー・マッピングをこのフェデレーション・ラッパーに追加する必要があります。これによって、Lotus Domino サーバーのユーザーにリモート接続ユーザー名とパスワードがマッピングされます。シナリオでは、Lotus Domino サーバーのユーザーは DB2DOMINO で、これは DB2 サーバーの有効化で作成されたアカウントです。図 2-22 に示すように、DB2 コントロール・センターの SAMPLE データベースに新しく作成されたサーバー定義から「新規ユーザー・マッピングの作成」アクションを選択します。

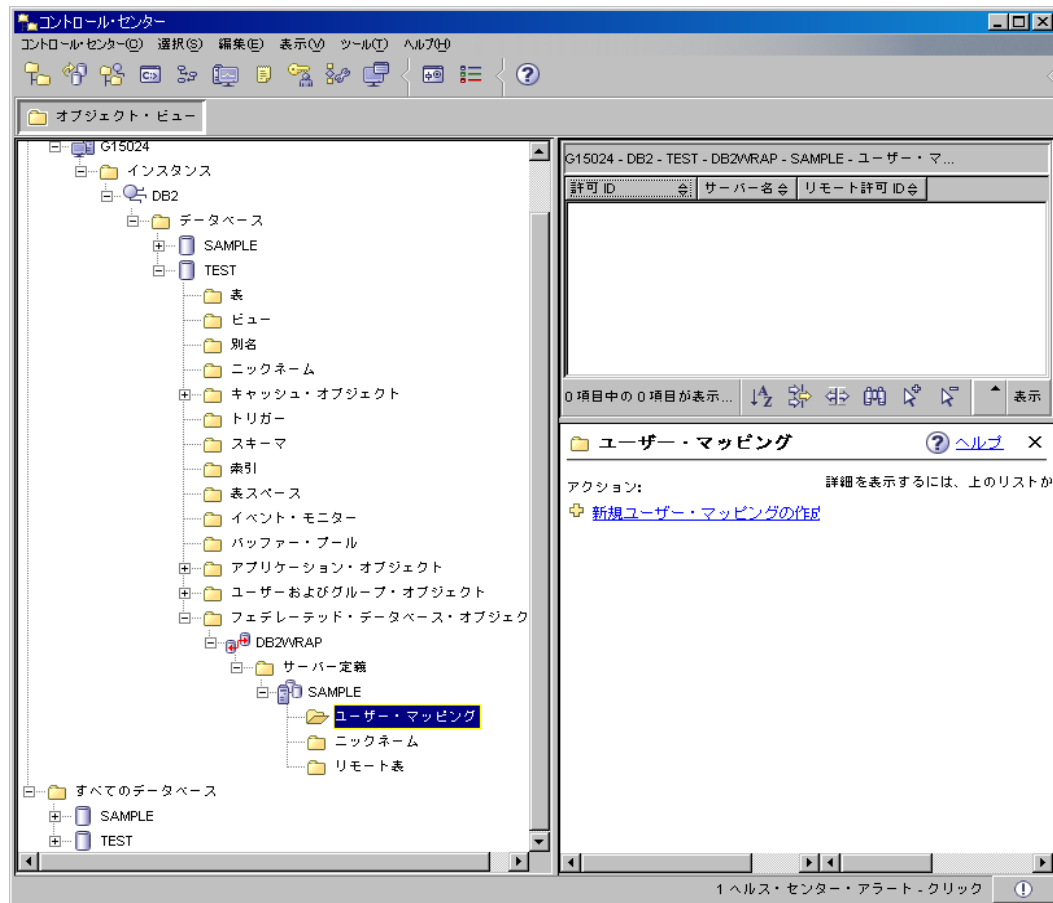


図 2-22 DB2 コントロール・センター: 「ユーザー・マッピング」

11. 図 2-23 に示すように、「ユーザー・マッピングの作成」ウィンドウが開きます。「ユーザー」タブで、リモート ID にマッピングするためのローカル・ユーザーの ID を 1 つ以上選択します。シナリオでは、ユーザー ID **DB2DOMINO** を選択します。

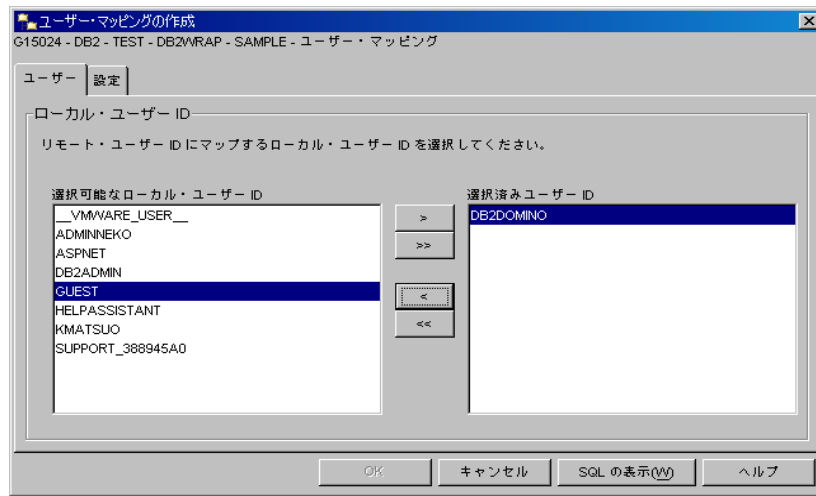


図 2-23 「ユーザー・マッピングの作成」ウィンドウ

12. 「設定」タブに移動し、「REMOTE\_AUTHID」オプションおよび「REMOTE\_PASSWORD」オプションに適切な値を入力します。シナリオでは、図 2-24 のように、これらのオプションに対して、DB2DOMINO と DB2DOMINO パスワードをそれぞれ指定します。(今回の DB2 サーバーでは、ユーザー ID も DB2DOMINO ですが、シナリオではこれをユーザー・マッピングに使用します)。ただし、DB2 に接続する Lotus Domino のユーザー ID (この例では DB2DOMINO) と、リモートの DB2 マシンで有効なユーザー ID (オプション REMOTE\_AUTHID と REMOTE\_PASSWORD) とのマッピングでは、ユーザー ID に固有の名前を付けることができます。ここで「OK」ボタンが使用可能になります。「OK」をクリックします。

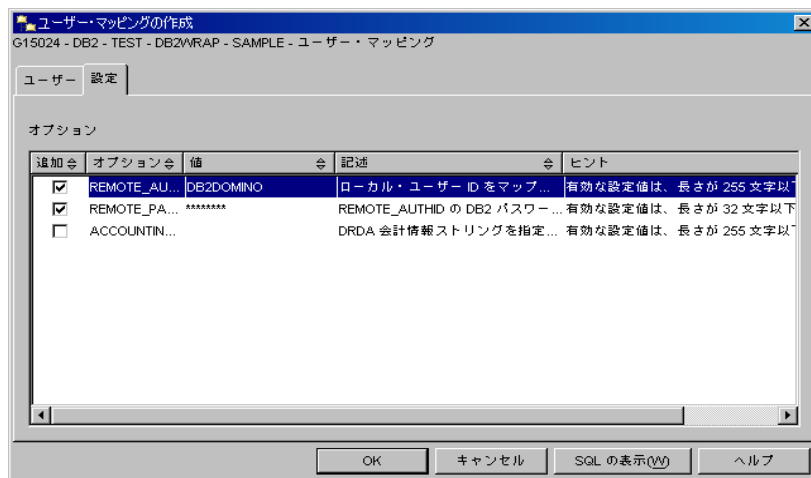


図 2-24 「ユーザー・マッピングの作成」：「設定」タブ

**注：**ユーザー・マッピングで問題が発生した場合は、DB2 コントロール・センターに戻って接続するデータベースを選択して右クリックし、「接続」を選択してユーザー ID とパスワードを入力してください。このシナリオのアプリケーションでは、DB2DOMINO のユーザー ID とパスワードを使用します。



13. DB2 コントロール・センターで、ローカル・オブジェクトを外部オブジェクトにマッピングするニックネームを作成します。ニックネームは、Query ビューの SQL ステートメントで使用します。図 2-25 に示すように、「ニックネーム」フォルダーを選択し、「新規ニックネームの作成」アクションをクリックします。

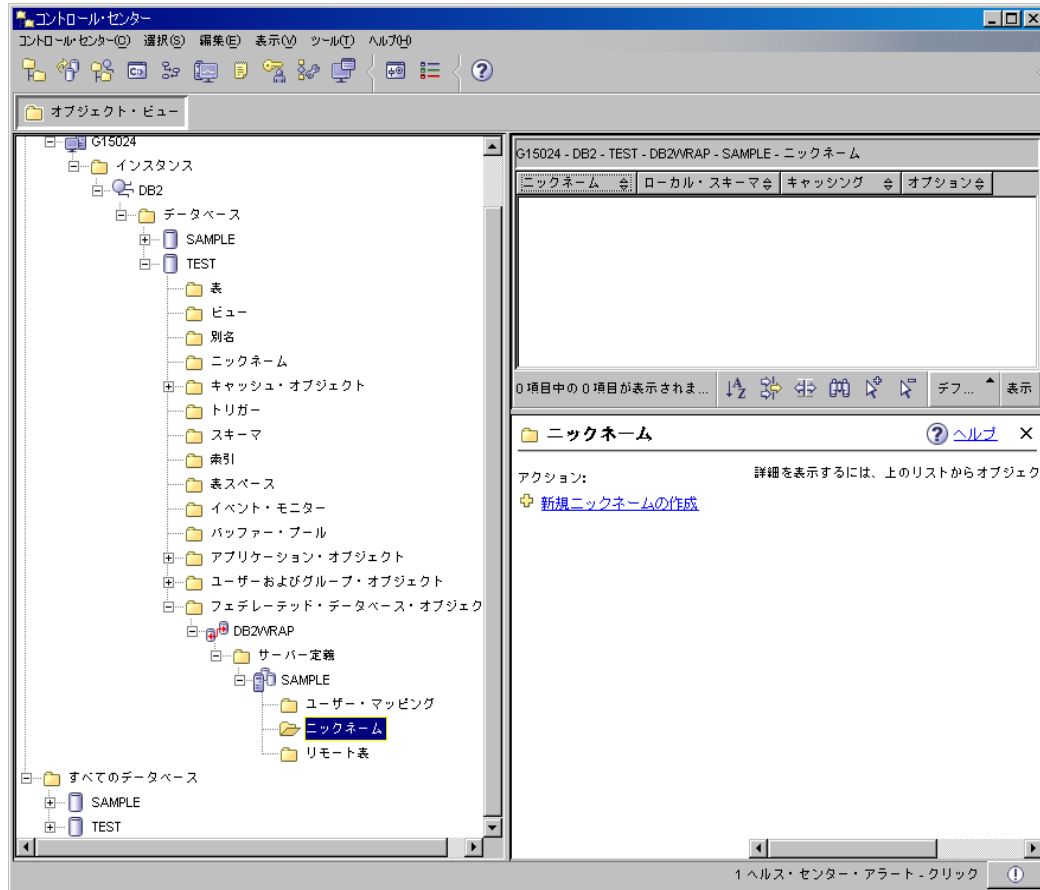


図 2-25 DB2 コントロール・センター：「新規ニックネームの作成」

14. 「ニックネームの作成」ウィンドウが開きます。「発見」をクリックすると、図 2-26 のように、「発見」ウィンドウが開きます。「OK」をクリックします。

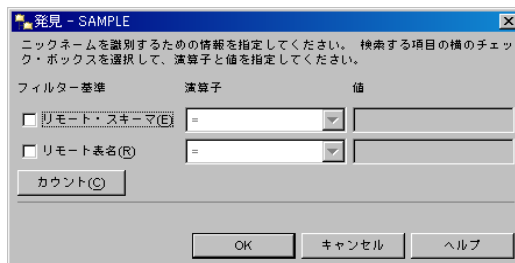


図 2-26 「ニックネームの作成」：「発見」ウィンドウ

15. 次に「ニックネームの作成」ウィンドウで、ニックネームを定義可能なオブジェクトを指定します。ITSO Electronics アプリケーションについては、図 2-27 のように「すべてチェック解除」をクリックしてから、ニックネーム EMPLOYEE に対し「作成」オプションを選択します。「OK」をクリックします。



図 2-27 「ニックネームの作成」ウィンドウ

16. ニックネーム EMPLOYEE が作成され、DB2 Command Center に表示されます。このニックネームをダブルクリックすると、図 2-28 に示すように、フェデレートされた EMPLOYEE テーブルが表示されます。

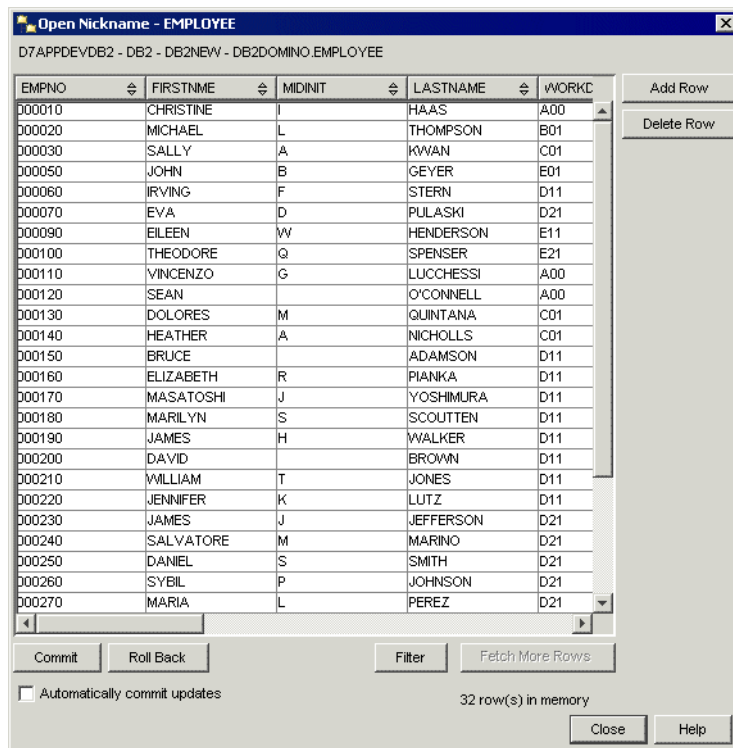


図 2-28 「Open Nickname」: 「EMPLOYEE」ウィンドウ

17. 最後の手順では、ニックネーム・オブジェクトに権限を付与します。これは、フェデレートド・テーブルへのアクセス権が付与されるマッピング済み DB2 ユーザーが必要とします。このシナリオでは、MANNYSANTANA というユーザー ID のユーザーにアクセス権を付与する必要があります。この手順はコマンド・プロンプトまたはコマンド・エディター 2 で行います。Lotus Domino サーバーのユーザーまたは別の管理者としてユーザーのデータベースにログオンします。以下のいずれかのアクションを実行します。
- コマンド・プロンプトから、次のコマンドを実行して権限を付与します。

GRANT SELECT ON DB2DOMINO.EMPLOYEE TO MANNYSANTANA

- 図 2-29 に示すように、コマンド・エディター 2 を使用して同じコマンドを実行します。

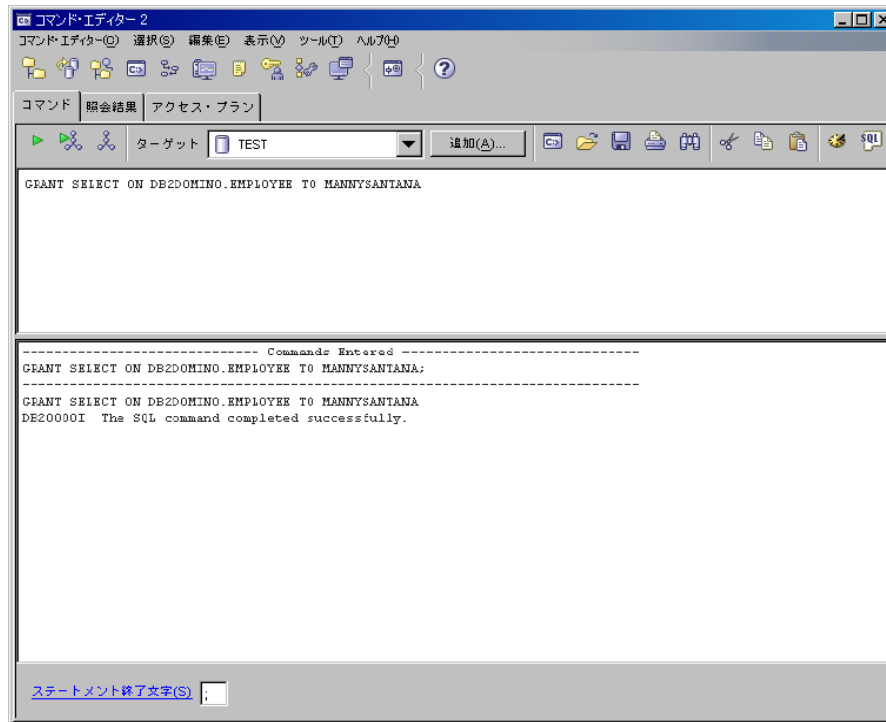


図 2-29 コマンド・エディター 2

これで、フェデレーテッド・データを持つ Query ビューを作成できます。これにより、ユーザーの Lotus Domino アプリケーションのデータ・アクセス機能が飛躍的に向上します。

## 2.4.2 DB2 フェデレーションが有効な Notes データベースの実装

DB2 サーバーに DB2 対応 Lotus Notes フェデレーションをセットアップしたら、ITSO Electronics アプリケーションにフェデレーテッド Query ビューを作成すると、フェデレーテッド・データにアクセスできます。これが DB2 フェデレーション機能を導入する最後の手順です。以下の手順に従って、ITSO Electronics アプリケーションにフェデレーテッド Query ビューを作成します。

1. Lotus Domino Designer で ITSO Electronics の Sales データベースを開き、「作成」→「ビュー」を選択します。38 ページの図 2-30 に示すように、「ビューの作成」ウィンドウが開きます。新規作成するフェデレーテッド Query ビューの名前として、「HR Employees」と入力し、「SQL クエリーの使用」ラジオ・ボタンを選択します。SQL クエリー・ストリングを入力し、「カスタマイズ」をクリックしてビューをカスタマイズし、新規列を追加します。

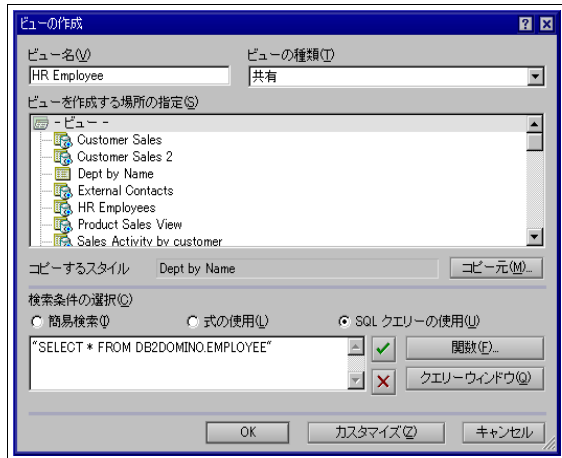


図 2-30 「ビューの作成」 ウィンドウ

2. 図 2-31 に示すように、フェデレートされた Employee テーブルから、「Employee No (number)」、「First Name」、「Last Name」の各列を新たに追加します。各列では、フェデレーテッド・テーブルの列名を列値の式として指定します。

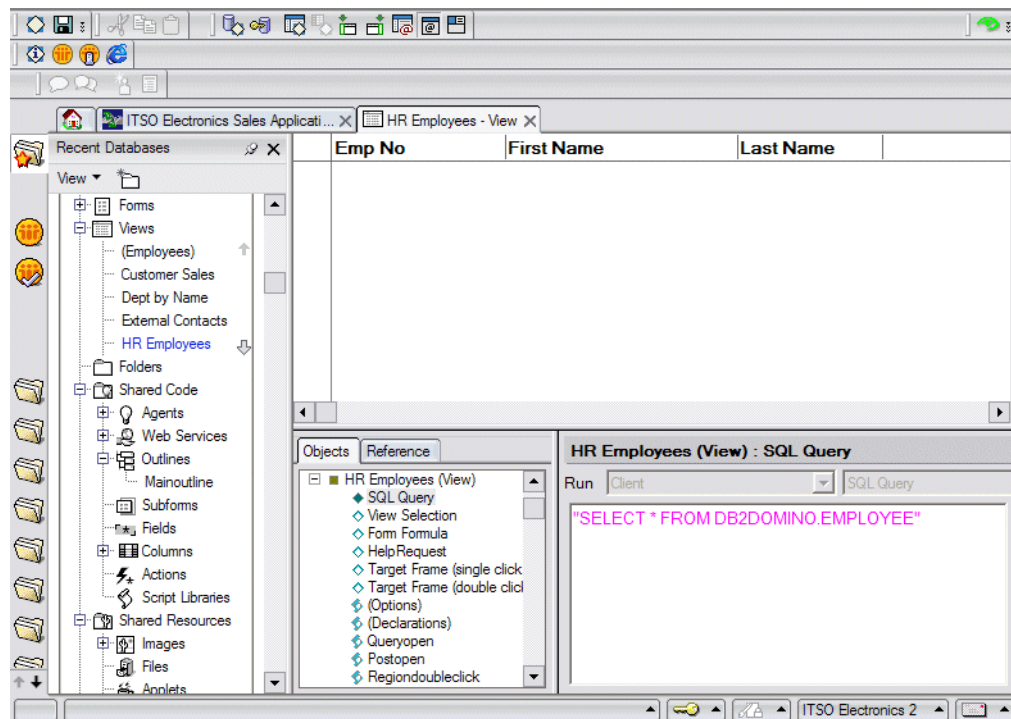


図 2-31 Lotus Domino Designer の「HR Employees」ビュー

3. ビューを保存し、アウトラインにビューを追加したら、Lotus Notes クライアントから HR Employee フェデレーテッド Query ビューを選択して、HR Employees テーブルから情報を表示します。39 ページの図 2-32 を参照してください。

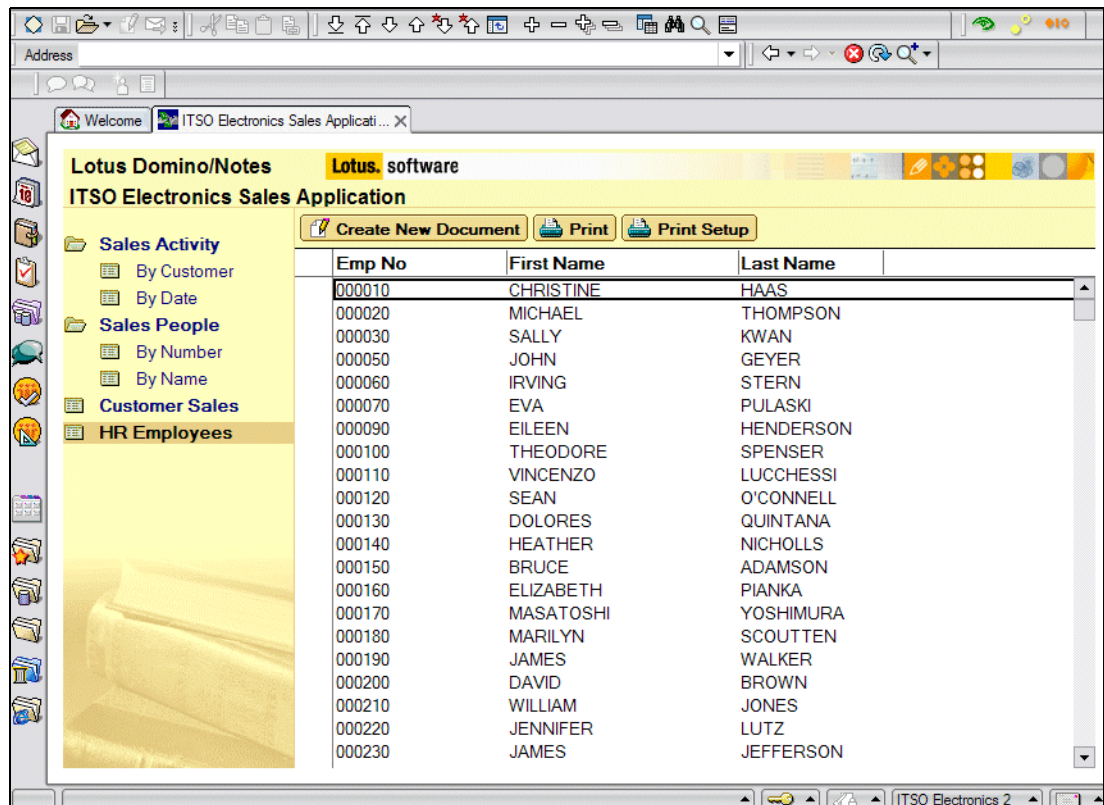


図 2-32 フェデレーテッド Query ビュー HR Employees

## 2.5 DB2 および Lotus Domino のセキュリティー上の考慮事項

DB2 対応の Lotus Domino を実行する場合は、セキュリティー上の考慮事項がいくつかあります。

- ▶ Lotus Domino データベースを DB2 サーバーで保管しても、アクセス・コントロール・リストは維持されるため、読者フィールドや作成者フィールドも維持されます。
- ▶ NSF を DB2 で保管すると、Lotus Domino はユーザーの NSF データを保持するためにテーブル・セットを作成および管理しますが、このテーブルには直接にはアクセスできません。DB2 Access ビューを定義した場合のみ、このデータにアクセスできます。
- ▶ 暗号化データは DB2 ベースのデータベースに保管できますが、DB2 Access ビューからはこのデータにアクセスできません。
- ▶ Lotus Notes データに DB2 Access ビューから SQL でアクセスした場合、Lotus Notes および Lotus Domino 7 では、ユーザーレベルのセキュリティー・チェックは、Lotus Notes ユーザー名と DB2 ユーザー名のマッピングにより常にユーザー側で実行されます。
- ▶ SQL を使用して DB2 Access ビューから外部の Lotus Notes/Lotus Domino にアクセスする場合、デフォルトの Lotus Notes ユーザー名と DB2 名のマッピングも使用されます。
- ▶ 外部の Lotus Notes からユーザー名をマッピングしない場合、以下の設定を DB2 Access サーバーの NOTES.INI ファイルに追加すると、DB2 GRANT メカニズムにより DB2 Access ビューへのアクセスを制御できます。

Allow\_Anonymous\_Access\_From\_DB2= 以下のいずれかの設定値

0 - 匿名アクセスを無効にする

1 - DB2 GRANT メカニズムを通じて匿名アクセスを有効にする

この NOTES.INI 設定を有効にすると、DB2 ユーザー名と Lotus Notes ユーザー名とのマッピングが行われない場合は常に、匿名で Lotus Notes にアクセスされます。

**注：**匿名アクセスを有効にして名前のマッピング要件を回避しても、Lotus Domino による DB2 Access ビューへのアクセス制御は存在します。匿名でアクセスするには、Lotus Domino サーバーは匿名アクセスを許可する必要があります。DB2 Access ビューに関連付けられている NSF のデフォルトのアクセス・レベルが、要求される SQL 操作を実行するための十分な権限を持っている必要があります。例えば、読み取り権限の SELECT や 作成者権限の INSERT などです。

- ▶ DB2 コントロール・センターで DB2 Access ビューを表示しようとしたときの DB2 アカウントに Lotus Domino の DB2 Access ビューへのアクセス権がない場合、次のエラー・メッセージが表示されます。

```
"com.ibm.db.DataException: A database manager error occurred.: [IBM][CLI Driver][DB2/NT]
SQL0443N Routine "ISREADER3" (specific name "") has returned an error SQLSTATE with
diagnostic text "14512 : No matching Notes user found for DB2 user and anon".
SQLSTATE=04004
```

セキュリティーおよび DB2 統合に関する詳細については、次の URL の「Lotus Domino 7 Designer Help」の DB2 Access ビューのセキュリティーに関するトピックを参照してください。

[http://www.lotus.com/ldd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/ldd/doc/domino_notes/7.0/help7_designer.nsf)

本書では、DB2 と Lotus Domino の統合におけるセキュリティーについてのみ説明します。より広範なセキュリティーおよび Lotus Domino の概要については、刊行予定の Redpaper「*Security Considerations in Notes and Domino 7*」(REDP-4104)を参照してください。

## 2.6 DB2 および Lotus Domino のトラブルシューティング

このセクションでは、DB2 統合に関する問題のトラブルシューティングについていくつか説明します。このセクションでの説明に加え、「Lotus Domino 7 Designer Help」の DB2 統合とトラブルシューティングに関するトピックも参照してください。

Lotus Domino DB2 の統合で問題が発生した場合、最初に DB2 Access サーバーの構成をテストしてください。DB2 Access テスト・ツールは、サーバー文書のすべての DB2 Access サーバーのフィールド・パラメーターおよび NOTES.INI ファイルのすべての DB2 Access サーバー設定をテストします。すべてのフィールドと設定が正しい場合は、DB2 Access サーバーと選択された Lotus Domino サーバーとの接続をテストし、DB2 の機能とプロパティーが存在するか検証して、DB2 Access サーバーの接続文書が有効かどうかを判別し、DB2 Access サーバー上で Domino ディレクトリのオープンを試行します。DB2 Access テスト・ツールで問題が発見された場合、Lotus Domino サーバー・コンソールか Lotus Domino Administrator クライアントに情報が戻されます。DB2 Access テスト・ツールを使用するには、以下の手順を実行します。

1. Lotus Domino Administrator から、「設定」タブに移動します。
2. 「**接続**」をクリックします。DB2 Access Server 接続文書を選択します。
3. 「ツール」パネルで、「**DB2 Server**」→「**DB2 アクセスのテスト**」をクリックします。
- 41 ページの図 2-33 に、DB2 Access テスト・ツールの出力例を示します。



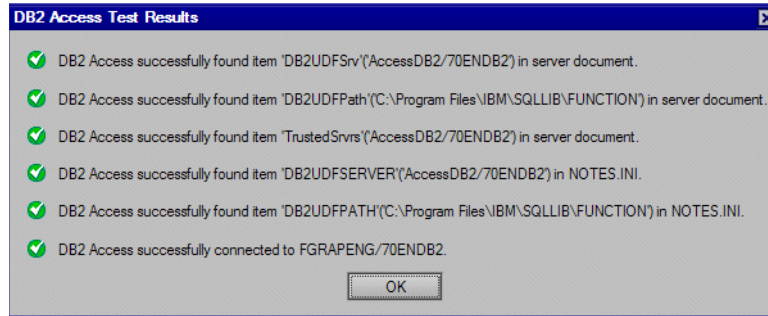


図 2-33 DB2 Access テスト・ツールの出力

## 2.6.1 DB2 Access ビューに必要な ACL

DB2 Access ビューを使用する場合、DB2 対応の Lotus Notes データベースへのアクセスを試みたユーザー側に問題が発生したときは、Lotus Domino サーバーの Domino ディレクトリにあるユーザーのユーザー文書の「ユーザー名」フィールドに指定されている各ユーザーの ACL エントリーが同一かどうか確認してください。

## 2.6.2 Query ビューのオープン時のエラー

通常、Query ビューを開くときにエラー・メッセージが表示された場合は、Lotus Domino サーバーのサーバー・コンソールを確認することをお勧めします。サーバー・コンソールに表示されるエラー・メッセージは、問題の根本的な原因を直接示している場合が多いためです。

以下のリストは、一般的なエラー・メッセージとその周辺情報を示します。

### ▶ DB2 権限の違反です。

このエラー・メッセージで別途通知されていた場合でも、この問題は多くの場合セキュリティ上の問題とは関係ありません。サーバー・コンソールを確認すると、次のように詳細情報が表示されています。

- DB2 は列情報の収集に失敗しました。DB2 権限の違反です。42S02 - [IBM][CLI Driver][DB2/NT] SQL0204N "SALES.SALESDAV" は未定義の名前です。SQLSTATE=42704 - \*LOCAL.DB2.051103024144

このエラー・メッセージで考えられる問題の原因は、次の 3 つです。

- この例の DB2 スキーマの Sales が存在しない。
- この例の DB2 Access ビューの SALESDAV が存在しない。
- DB2 Access ビューが取り込まれていない。

- DB2 は列情報の収集に失敗しました。DB2 権限の違反です。42S22 - [IBM][CLI Driver][DB2/NT] SQL0206N "C.CUSTOMERNAM" は使用された部分で有効なコンテキストではありません。SQLSTATE=42703 - \*LOCAL.DB2.051103024144

このエラーが表示された理由には、呼び出された列が DB2 Access ビュー (例では CUSTOMERNAM) に存在しないことが考えられます。

### ▶ DB2 の動的 SQL エラーです。

このエラー・メッセージは、SQL ステートメントの構文エラーを示しています。サーバー・コンソールで SQL ステートメントを確認してください。

DB2 は列情報の収集に失敗しました。DB2 の動的 SQL エラーです。42601 - [IBM][CLI Driver][DB2/NT] SQL0104N "SELECT \*" の後に予期しないトークン "FROMSALES2" が検

出されました。予期されるトークンは次のようになります: "<from>". SQLSTATE=42601 - G921557D.GD04.0088C6170623

この例では、コマンドの FROM と SALES2 の間のスペースが欠落していました。

- ▶ ビューを開く: 指定のクエリーは結果セットを生成できないため、実行できませんでした。

セキュリティ上の理由から、Query ビューの SQL ステートメントでは、結果セットを戻すステートメント数が制限されています。UPDATE コマンドまたは INSERT コマンドは、選択式として使用すると正しく動作しません。例えば、最初のコマンドが SELECT の場合、SQL ステートメントの冒頭で、SELECT ではなく SELCT などと誤って入力すると、このエラーが発生する場合があります。

- ▶ サーバーの現バージョンにこの関数は導入されていません。

DB2 対応でないシステムまたはサーバーのローカルの複製で Query ビューを開こうとしたことが想定されます。

- ▶ DB2 が文書の取り出しに失敗しました。DB2 対応関数の例外: 38503 - [IBM][CLI Driver][DB2/NT] SQL0430N ユーザー定義関数 "DOMINO.ISREADER3" (指定の名前 "SQL051024022301200") は異常終了しました。SQLSTATE=38503 - \*LOCAL.DB2.051028175704

先行バージョンの Lotus Notes および Lotus Domino がシステムに導入されていることがこの問題の一因と考えられます。先行バージョンを削除するか、フォルダーの名前を変更すると、このような問題を解決できます。


**ヒント:** 問題が発生しなくなるまで、コードまたはアプリケーション設計の各工程を簡略化してください。これにより、問題の根本的な原因を特定できる場合がよくあります。多くの場合、明白な問題の回避策が存在します。

### 2.6.3 問題の根本的な原因究明に役立てるために収集するデータ

トラブルシューティングの関連情報を収集すると、問題解決の多くの場合において役立ちます。ユーザー自身、同僚、IBM 担当者にとって、これらが問題の原因を特定するための有効な手掛かりとなります。

- ▶ 使用中のソフトウェア・コンポーネントに関する正確なバージョン情報
  - オペレーティング・システム
  - DB2 サーバー
  - ランタイム・クライアント
  - Lotus Notes および Lotus Domino
  - DB2 Access サーバー
- ▶ 使用するパッチ・レベル情報および地域化情報
- ▶ すべてのエラー・メッセージの画面キャプチャー
- ▶ 以下のコンポーネントのうち 1 つが破損した場合は、次のいずれかのコンポーネントでの障害発生時に診断情報を含む NSD ファイルが作成されます。
  - Lotus Notes クライアント
  - Lotus Domino サーバー
  - DB2 Access サーバー
- ▶ すべての使用可能なログ情報
- ▶ データベース設計
- ▶ 問題が発生する前に行っていた各作業手順の詳細な説明
- ▶ Lotus Domino サーバーおよび DB2 Access サーバーの NOTES.INI





# Web サービスによる ITSO Electronics アプリケーションの 拡張

IBM Lotus Domino Designer 7 は、Web サービスを追加することで、新しい標準をサポートするという伝統を守り、rapid application development モデルを拡張しています。Web サービス機能により、Lotus Domino アプリケーションを Lotus Domino サーバーでホストされる Web サービスに変換することができます。これにより、各種プラットフォーム間でのデータ・アクセスおよびデータ操作が改善されます。Web サービスは、内蔵タイプのモジュラー・アプリケーションで、カスタム・コーディングした接続に依存しない連携が可能です。これは、Web サービスがオープン・スタンダードに基づいて構築されているためです。Web サービスにより、さまざまなアプリケーションおよび IT システムが相互に対話できるので、サービス・コンポーネントが動的に結合および再結合できます。これにより、企業は変化するビジネス・ニーズおよび顧客ニーズに素早く対応できます。

この章では、ITSO Electronics アプリケーションに Web サービスを実装して拡張する方法を説明します。次の分野に焦点を当てます。

- ▶ サービス指向アーキテクチャー：Web サービスを SOA スキームに適合させる方法
- ▶ Web サービスと Web サービス記述言語 (WSDL)
- ▶ Web サービスを実装するビジネス上の理由
- ▶ Web サービス要素：ITSO Electronics アプリケーションへの単純な Web サービス要素および複雑な Web サービス要素の追加
- ▶ WSDL 文書のエクスポートおよびインポート
- ▶ Web サービスでの例外およびエラー処理
- ▶ Web サービスのセキュリティ
- ▶ Java および LS2J/LotusScript を使用した Web サービスのコンシューム

## 3.1 サービス指向アーキテクチャーと Web サービス

サービス指向アーキテクチャー (SOA) は、サービスと呼ばれるアプリケーションの異なる機能単位を、それらのサービス間に明確なインターフェースと契約を介在させることで関連付けるコンポーネント・モデルです。このインターフェースは、サービスの実装先のハードウェア・プラットフォーム、オペレーティング・システム、およびプログラミング言語に依存しない中立的な方法で定義されます。これにより、サービスは、さまざまなシステム上に構築されても、共通の方法で相互に対話できます。Web サービスは、中立的なプラットフォーム非依存のアプリケーション開発の思想を具現化したものであるため、SOA アプリケーション・アーキテクチャーの不可欠なコンポーネントです。

特定の実装に強く結び付けない中立的なインターフェースを定義するというこの特徴は、サービスの疎結合と呼ばれています。疎結合されたシステムの利点は、アプリケーション全体を構成する各サービスの内部の構造および実装を素早く変更して進歩させられる点です。一方、密結合では、アプリケーションの異なるコンポーネント間のインターフェースが機能および形式において強固に関連付けられます。このため、密結合は、アプリケーションの一部または全体で形式の変更が必要になった場合には脆弱さを露呈してしまいます。

疎結合システムが必要になった原因は、変化する方針、ビジネス上の強み、ビジネスの焦点、パートナー関係、業界標準、およびビジネスの本質に影響を与えるその他のビジネス関連の要因など、変化する環境に、ビジネスのニーズに基づいて素早く適用できるビジネス・アプリケーションが必要になったからです。オンデマンド・ビジネスの環境では、物事を行う方法または仕事の方法がオンデマンド方式で必要に応じて変化します。そのような環境に関連して、ビジネスを柔軟に行えることが必要になったのです。

サービス指向アーキテクチャーは新しいものではありませんが、過去数十年間に現れた従来の密結合モデル、オブジェクト指向モデルの代わりとなるモデルです。SOA ベースのシステムでは、オブジェクト指向設計で個々のサービスを構築することも事実ですが、その設計全体はサービス指向です。システム内でオブジェクトを使用できるので、SOA はオブジェクト・ベースと言えますが、全体としてはオブジェクト指向ではありません。この違いはインターフェース自体にあります。SOA システムのプロトタイプの例としてよく挙げられるのが、Common Object Request Broker Architecture (CORBA) です。CORBA は SOA と同様の概念を定義しており、しばらくの間存続していました。

ただし、現在の SOA は、Extensible Markup Language (XML) に基づいた最新技術に依存しており、この点が CORBA とは異なります。CORBA の古いインターフェース定義言語 (IDL) ではなく、Web サービス記述言語 (WSDL) という XML ベースの言語でインターフェースを記述することで、より動的で柔軟なインターフェース・システムにサービスは移行されました。

SOA を実装する方法は、Web サービスの他にもあります。CORBA のような方法と、IBM WebSphere® MQ などの Message Oriented Middleware システムを使用する方法です。ただし、アーキテクチャー・モデルにするには、サービス記述以外のことも実行する必要があります。アプリケーション全体でサービス間のワークフローをどのように実行するか定義する必要があります。さらに、ビジネスの運用と、ビジネスで使用するソフトウェアの運用の間の変換点を見つける必要があります。つまり、SOA で、ITSO Electronics などの企業の商用プロセスをテクニカル・プロセスに関連付け、この 2 つの間のワークフロー関係をマップする必要があります。例えば、サプライヤーへの支払いはビジネス・プロセスで、一方、電子パーツ・データベースを更新して新しい出荷品を含めるのはテクニカル・プロセスです。したがって、ワークフローも SOA の設計において重要な役割を果たします。

## 3.2 Web サービス要素と WSDL

Web サービスは、相互運用可能なマシン同士がネットワークを介して対話できるようにするために設計されたソフトウェア・システムです。Web サービスには、マシンが処理できるフォーマット (具体的には WSDL) で記述されるインターフェースがあります。他のシステムは、SOAP メッセージを使用して、インターフェースの記述で定められた方法で Web サービスと対話します。SOAP メッセージは、通常、XML シリアライゼーション、その他の Web 関連の標準と HTTP を使用して送信されます。Lotus Domino Designer 7 には、Web サービス設計要素が含まれており、また Web サービス記述言語 (WSDL) を組み込みでサポートしています。WSDL は、ネットワークで使用する XML ベースのサービスを記述するための標準仕様です。WSDL により、サービス・プロバイダーは、根底にあるランタイム実装に関係なく、そのプロバイダーのシステムに対する要求の基本フォーマットを簡単に記述できます。

Lotus Domino 7 では、Web サービスの実装を次のように制限しています。

- ▶ ネイティブでサポートされるのはプロバイダー・エンティティのみです。
- ▶ バインディングは HTTP POST プロトコルを使用した Simple Object Access Protocol (SOAP) である必要があります。

Web サービス要素には次の特性があります。

- ▶ Lotus Domino データベースでは、コンパイルされた Web サービスは、Web エージェントと同じく独立型プログラムです。Web サービスを使用するには、Web アクセスが可能なサーバー上にその Web サービスが存在しているか、その Web サービスの複製が存在している必要があります。サーバーに配置した後、Web サービスには、次の Lotus Domino URL コマンドでアクセスできます。
  - **?OpenWebService**: HTTP POST プロトコルを使用して、Web サービスを呼び出します。HTTP GET プロトコルが使用されると、**?OpenWebService** コマンドはサービスに関する情報を戻します。
  - **?WSDL**: Web サービスに WSDL 文書を照会します。
- ▶ Web サービスは、Lotus Notes または Lotus Domino Designer のプレビューで HTTP セッションを介してテストできます。
- ▶ Web サービスにはエージェントと同じセキュリティ機能があります。
- ▶ 既存の WSDL 文書をインポートして、スケルトン Web サービスを生成できます。このスケルトン・コードは Web サービスの定義に対応します。生成後、実装コードを追加します。
- ▶ または、最初から Web サービスを作成することもできます。その場合、Java または LotusScript コードを記述して、Web サービスの定義を作成します。

### 3.2.1 WSDL 文書

WSDL 文書は、Web サービスを定義し、次の要素を含みます。

#### ▶ Service

全体としての Web サービス。WSDL 文書では、ルートにある <definitions> 要素でサービスが定義されます。子要素には、<types>、<message>、<portType>、<binding>、および <service> があります。<service> 要素には、サービスの名前を定義する “name” 属性があります。

#### ▶ Types

サービスが使用するデータ型。WSDL 文書では、<types> 要素でデータ型が定義されます。この要素は、1 つ以上の <xsd:schema> 要素を含みます。データ型の定義については、以下を参照してください。

<http://www.w3.org/TR/xmlschema-2>

► Message

サービスとやり取りされるデータの抽象的定義。任意の数のメッセージを使用できます。WSDL 文書では、<message> 要素で各メッセージが定義されます。各 <message> 要素には、名前と型でデータを定義する 1 つ以上の <part> 要素が含まれます。

► PortType

サービスがサポートする操作の抽象セット。WSDL 文書では、<portType> 要素でポート・タイプが定義されます。各 <portType> 要素には、1 つ以上の <operation> 要素が含まれます。

► Operation

サービスがサポートするアクションの抽象的定義。WSDL 文書では、<operation> 要素で操作が定義されます。各 <operation> 要素には、関連するメッセージを指定する <input> 要素と <output> 要素が含まれます。オプションで <fault> 要素も指定できます。

► Binding

ポート・タイプの具体的なプロトコルとデータ・フォーマットの仕様。WSDL 文書では、<binding> 要素でバインディングが定義されます。<binding> 要素には、ポート・タイプの名前を指定する「type」属性があります。SOAP エンコードでは、<binding> 要素に、「style」属性（「rpc」または「document」）と「transport」属性（「http://schemas.xmlsoap.org/soap/http」）を持つ <soap:binding> 要素が含まれます。<binding> 要素には、各操作のデータ・フォーマットを定義する <operation> 要素が含まれます。

► Port

バインディングとネットワーク・アドレスの組み合わせで定義される単一エンドポイント。WSDL 文書では、<service> 要素の下位の <port> 要素でポートが定義されます。<port> 要素には、バインディングの名前を定義する「binding」属性と、そのバインディングに関連するネットワーク・エンドポイントの名前を定義する「address location」属性があります。

## 3.2.2 Simple Object Access Protocol

Lotus Domino 7 の Web サービス要素は、Simple Object Access Protocol (SOAP) 1.1 に基づいたホスティング環境を提供します。SOAP は、非集中の分散環境にあるピア間で、構造化および型定義された情報を交換するために使用する XML ベースの情報を定義しています。SOAP メッセージは、基本的には SOAP 送信側から SOAP 受信側に対して片方向伝送されますが、より複雑な対話パターンを実装するアプリケーションで SOAP メッセージを組み合わせることが通常は行われます。

以下は World Wide Web Consortium (W3C) のドラフト仕様です。

「SOAP は、非集中の分散環境で構造化情報を交換するための軽量プロトコルです。SOAP は XML ベースのプロトコルで、次の 3 つの部分で構成されます。1 つは、メッセージの中に何があってそれをどう処理するのかということを記述するためのフレームワークを定義したエンベロープで、もう 1 つは、アプリケーションが定義したデータ型のインスタンスを表現するためのエンコード規則の集合で、最後の 1 つは、リモート・プロシージャ・コールとそれへのレスポンスを表現するための規約です。」<sup>1</sup>

## 3.2.3 WSDL のインポート

Lotus Domino Designer 7 では、アプリケーション開発者は、既存の WSDL ファイルを Lotus Domino Designer アプリケーションにインポートし、LotusScript クラスまたは Java クラスを

<sup>1</sup> <http://www.w3.org/Submission/2000/05/> より

生成できます。ただし、Lotus Domino Designer 7 は、既存のすべての WSDL ファイルを解釈できるわけではありません。WSDL ファイルで使用されている XML 要素によっては解釈できない場合もあるので、アプリケーション開発者の方は注意してください。

### 3.3 Web サービスを実装するビジネス上の理由

Lotus Domino リリース 5 以降、Lotus Domino サーバーの外部にアプリケーションを作成して、そのアプリケーションで Lotus Domino データに直接アクセスできます。これは、バックエンド・クラスに対し COM バインディングおよび CORBA バインディングを使用することで行われます。ただし、これらの外部アプリケーションでは、データの妥当性検査とデータの整合性の問題が依然としてまたは正しく解決されません。Web サービスは、直接データ・アクセスとネットワーク操作を実行する機能の両方を利用できる適切な解決策を提供します。Web サービスは、アプリケーションの機能とデータをネットワークに公開する方法を提供します。Java、Microsoft .NET、および他のアプリケーションの開発者は、Web サービスをコンシュームするだけで Lotus Domino のデータにアクセスできます。

Web サービスを軸に設計された Lotus Domino アプリケーションの利点は、顧客に現在の製品の情報を提供する ITSO Electronics アプリケーションの例でもっとも適切に示すことができます。例えば、ITSO Electronics アプリケーションで、製品情報にユーザーがアクセスできるようにする場合、顧客データベースに外部アプリケーションがアクセスできるようにする必要があります。これは、そのアプリケーションが Java、J2EE、.NET のうちのいずれで作成されている場合も同じです。顧客のアプリケーションは、情報の要求を Web サービスに対して送信するので、相手のプラットフォームおよびオペレーティング・システムが何なのかは特定する必要がありません。Web サービス要素は、XML メッセージとしてデータ要求を受け取ります。したがって、ユーザーが顧客アプリケーションのフォームにクエリーを入力して製品情報を取得する場合、Web サービスは、XML フォーマットの SOAP メッセージとして要求を受け取り、同じ方法で顧客に応答を送信します。ITSO Electronics アプリケーションは、拡張されたサポートを使用してその機能を拡張することで、多数の製造メーカーから価格情報を取得できます。Web サービス要素は、Java を使用して複数の URL アドレスにデータ要求を送信でき、XML プロトコルを使用して価格データを問い合わせることができます。ITSO Electronics アプリケーションでは、要求はすべて SOAP メッセージとして送信され、応答も同じフォーマットで受信されます。これらの送受信は、データの送信元のデータベースの種類、オペレーティング・システム、またはプラットフォームについてユーザーが知ることなく行われます。ユーザーが分かるのは、無制限のデータ・ストアにアクセスしていること、それらのデータが極めて効率的に提示されることだけです。

### 3.4 ITSO Electronics アプリケーションへの Web サービスの追加

Web サービス設計要素と WSDL サポートを使用することで、Web サービス用の Java コードまたは LotusScript コードを短時間で作成または拡張できます。これにより、Java 2 Platform、Enterprise Edition (J2EE) アプリケーション、Microsoft .NET アプリケーションなどの外部システムに、数多くの Lotus Domino アプリケーションを公開できます。このセクションでは、Lotus Domino アプリケーション開発環境で Web サービスを実装して、ITSO Electronics アプリケーションを拡張する方法を説明します。

**注：** Web サービスはどの Lotus Notes/Domino データベースにも追加できますが、この Redpaper ではその目的のために、すべての Web サービスを、ITSO Electronics アプリケーションの個別のデータベース (webservices.nsf および wsconsumer.nsf) に追加します。

### 3.4.1 ITSO Electronics に Web サービスを追加するビジネス上の要件

ITSO Electronics アプリケーションに Web サービスを実装するビジネス上の主要な要件は、顧客が現在の製品情報を表示できるようにする必要があるということです。Web サービスはこのニーズを完璧に満たします。ITSO Electronics は、将来 Web サービスを使用して、顧客がプラットフォーム、オペレーティング・システム、またはインターフェースに関係なく送り状などのインボイス情報にアクセスできるようにする予定です。

### 3.4.2 ユーザーの要件

例として使用している ITSO Electronics アプリケーションの Web サービスの主要なユーザーは、ITSO Electronics の顧客です。このアプリケーションで、プログラマチックな仕組みと、直感的で使いやすいユーザー・インターフェースを介して、顧客が現在の製品情報にアクセスできるようにする必要があります。

## 3.5 Lotus Domino Designer 7 での単純な Web サービスの作成

IBM Lotus Domino Designer 7 を使用すると、どれだけ Web サービスの作成が簡単になるかをここでは見ていきます。

最初にエラーおよび例外処理、ロギング、およびパフォーマンスを除外し、核となる機能に集中します。

ITSO Electronics では、Java と LotusScript の両方をサポートする必要があります。

### 3.5.1 新しい Web サービスの作成

このセクションでは、ITSO Electronics アプリケーションを例にして、新しい Web サービスの作成方法を説明します。

1. Lotus Domino Designer でデータベースを開きます。これは、シナリオの ITSO Web Services データベースです。設計リストから、「共有コード」→「Web サービス」を選択し、「新規 Web サービス」をクリックします。「Web サービス」プロパティ・ウィンドウで、図 3-1 に示されているように、Web サービスの名前として ProductInfo と入力します。

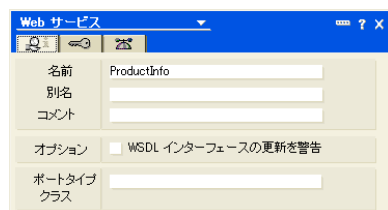


図 3-1 「Web サービス」プロパティ・ウィンドウ

2. LotusScript 宣言を作成するために、図 3-2 に示されているように、例 3-1 のコードを Web サービスの「Declarations」に貼り付けるか入力します。

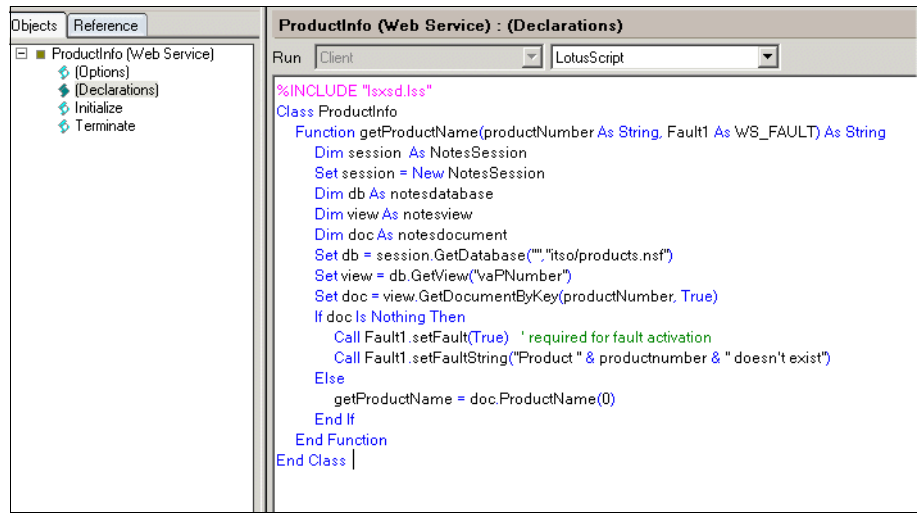


図 3-2 単純な Web サービスのソース・コード

例 3-1 単純な Web サービスのコード

```
%INCLUDE "lsxsd.lss"
Class ProductInfo
    Function getProductname(productNumber As String, Fault1 As WS_FAULT) As String
        Dim session As NotesSession
        Set session = New NotesSession
        Dim db As notesdatabase
        Dim view As notesview
        Dim doc As notesdocument
        Set db = session.GetDatabase("", "itso/products.nsf")
        Set view = db.GetView("vaPNumber")
        Set doc = view.GetDocumentByKey(productNumber, True)
        If doc Is Nothing Then
            Call Fault1.setFault(True) ' required for fault activation
            Call Fault1.setFaultString("Product " & productnumber & " doesn't exist")
        Else
            getProductname = doc.ProductName(0)
        End If
    End Function
End Class
```

3. 「Web サービス」プロパティ・ウィンドウを再び開き、保存を行う前に、図 3-3 に示されているように「ポートタイプ」クラスを ProductInfo に設定します。

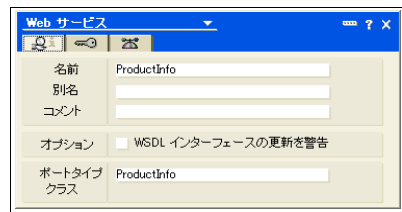


図 3-3 「Web サービス」プロパティ・ウィンドウ: 「ポートタイプ」クラス

4. 「詳細」タブに移動し、図 3-4 に示されているように、「プログラミングモデル」のデフォルト値が「RPC」になっていること、および「SOAP メッセージ形式」のデフォルト値が「RPC/ エンコード」になっていることを確認します。



図 3-4 「Web サービス」プロパティ・ウィンドウ：「詳細」タブ

5. プロパティ・ウィンドウを閉じ、Web サービスを保存します。

### 3.5.2 ブラウザーでの Web サービスの表示

新しく作成した ProductInfo Web サービスをテストするために、ブラウザーを開き、次の URL を入力します。

`http://servername/databasename.nsf/webservicename?openwebservice`

ITSO Electronics アプリケーションのシナリオでは、次の URL を使用します。

`http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductInfo?openwebservice`

図 3-5 に示されているように、Lotus Domino が Web サービスのポート名、ポートの操作の名前、さらには WSDL 文書へのリンクを表示して応答します。

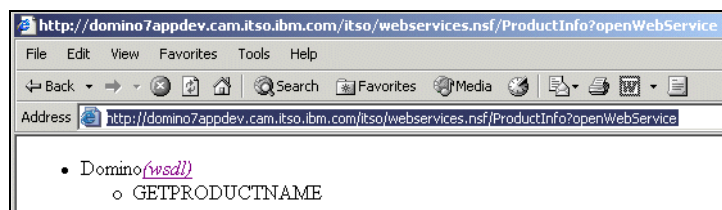


図 3-5 Web ブラウザーで開かれた Web サービス

### 3.5.3 生成された WSDL 文書

次の Lotus Domino URL コマンドで、HTTP GET の応答として Web サービスの WSDL 文書を取得できます。サフィックスとして ?WSDL を使用します。

`http://servername/databasename.nsf/webservicename?WSDL`

例えば、このシナリオでは以下を使用します。

`http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductInfo?WSDL`

51 ページの例 3-2 は、結果の WSDL 文書です。

WSDL の詳細については、45 ページの 3.2.1、「WSDL 文書」を参照してください。



例 3-2 サフィックス ?WSDL を使用して Web サービスにアクセスした場合にサーバーから返されるコード

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="urn:DefaultNamespace" xmlns:intf="urn:DefaultNamespace"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:DefaultNamespace">
  <wsdl:message name="GETPRODUCTNAMERequest">
    <wsdl:part name="PRODUCTNUMBER" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="GETPRODUCTNAMEResponse">
    <wsdl:part name="GETPRODUCTNAMEReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:portType name="ProductInfo">
    <wsdl:operation name="GETPRODUCTNAME" parameterOrder="PRODUCTNUMBER">
      <wsdl:input message="impl:GETPRODUCTNAMERequest" name="GETPRODUCTNAMERequest"/>
      <wsdl:output message="impl:GETPRODUCTNAMEResponse" name="GETPRODUCTNAMEResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="DominoSoapBinding" type="impl:ProductInfo">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="GETPRODUCTNAME">
      <wsdlsoap:operation soapAction=""/>
      <wsdl:input name="GETPRODUCTNAMERequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:DefaultNamespace"
use="encoded"/>
      </wsdl:input>
      <wsdl:output name="GETPRODUCTNAMEResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="urn:DefaultNamespace"
use="encoded"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ProductInfoService">
    <wsdl:port binding="impl:DominoSoapBinding" name="Domino">
      <wsdlsoap:address
location="http://domino7appdev.cam.itso.ibm.com:80/itso/webservices.nsf/ProductInfo?OpenWebService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

?WSDL サフィックスで生成される結果のコードは複雑ですが、WSDL 文書はマシンが読み取るもので、エンド・ユーザーが読むものではないので問題ありません。例えば、Lotus Domino Designer は WSDL 文書をインポートし、LotusScript または Java でスケルトン・クラスを作成できます。これについては、46 ページの 3.2.3、「WSDL のインポート」で説明します。

### 3.5.4 単純な Web サービスのテスト

Web サービスをテストおよびコンシュームできるアプリケーションは数多くあります。本書ではその目的のために、IBM Rational® Application Developer を使用して Web サービスをテストします。Rational Application Developer を使用して Web サービスのテストを開始するには、次の手順に従います。

1. Rational Application Developer を開きます。
2. 「実行」→「Web サービス・エクスプローラーの起動」を選択します。
3. 「WSDL ページ」アイコンをクリックします (右上の右から 2 番目のアイコン)。

4. 「Navigator」 ペインに移動し、図 3-6 に示されているように「WSDL Main」を選択します。

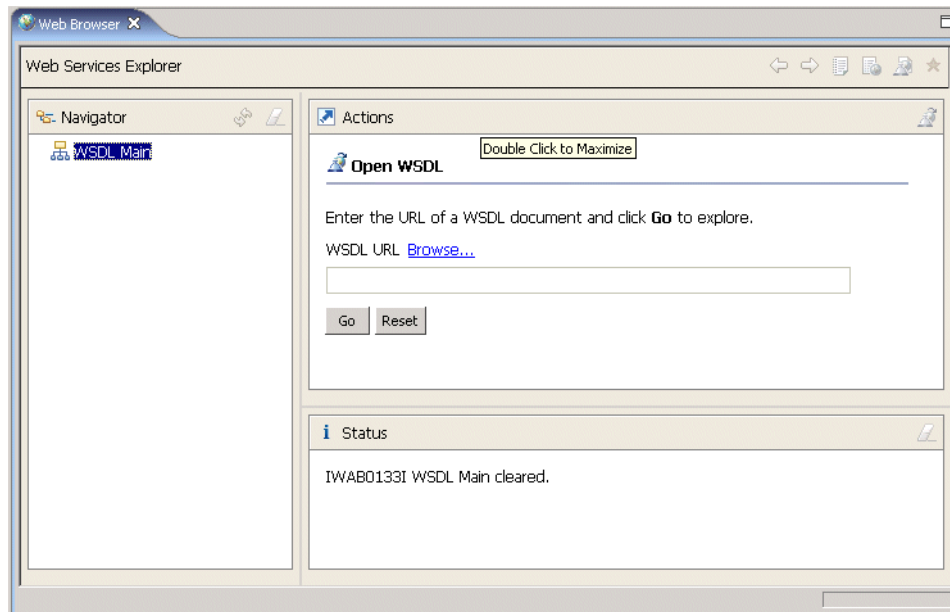


図 3-6 「Navigator」 ペインで「WSDL Main」を選択

5. 「Open WSDL」 ペインに移動し、「WSDL URL」を指定します。ITSO Electronics アプリケーションのシナリオでは、次の URL を使用します。  
`http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductInfo?wsdl`
6. 「Go」をクリックします。図 3-7 に示されているように「WSDL Binding Details」 ペインが開きます。

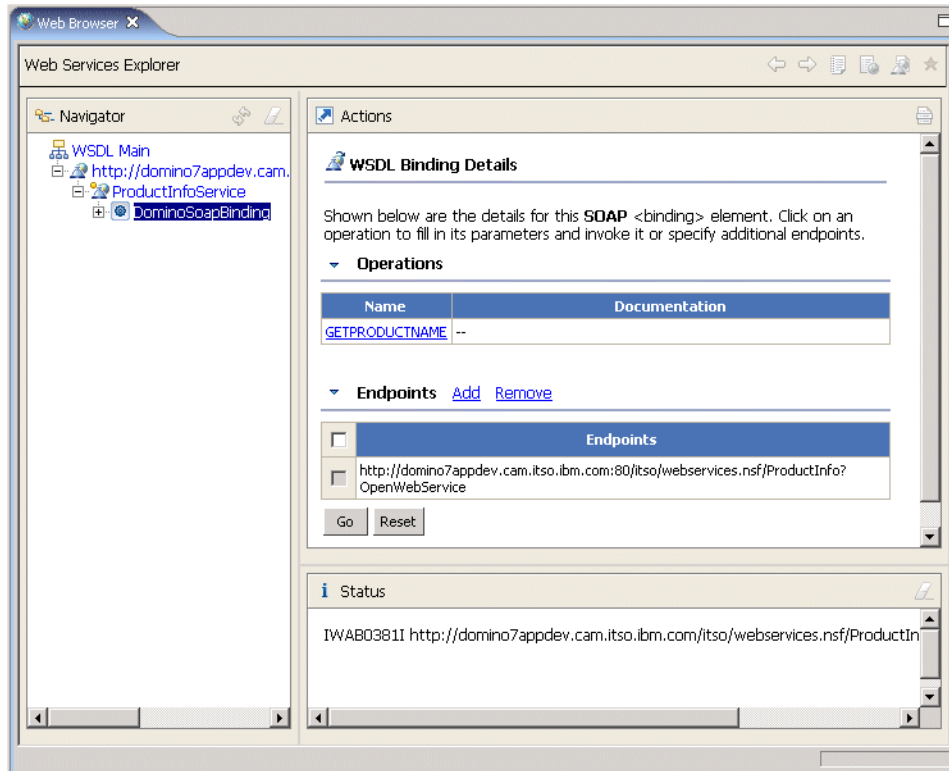


図 3-7 「WSDL Binding Details」 ペイン

7. 「WSDL Binding Details」 ペインには、SOAP 要素に関する操作が表示されます。ペインにリストされている操作をクリックし、パラメーターを入力して操作を呼び出すか、追加のエンドポイントを指定します。このシナリオでは、「GETPRODUCTNAME」操作を選択します。選択すると、図 3-8 に示されているように「Invoke a WSDL Operation」ペインが表示されます。

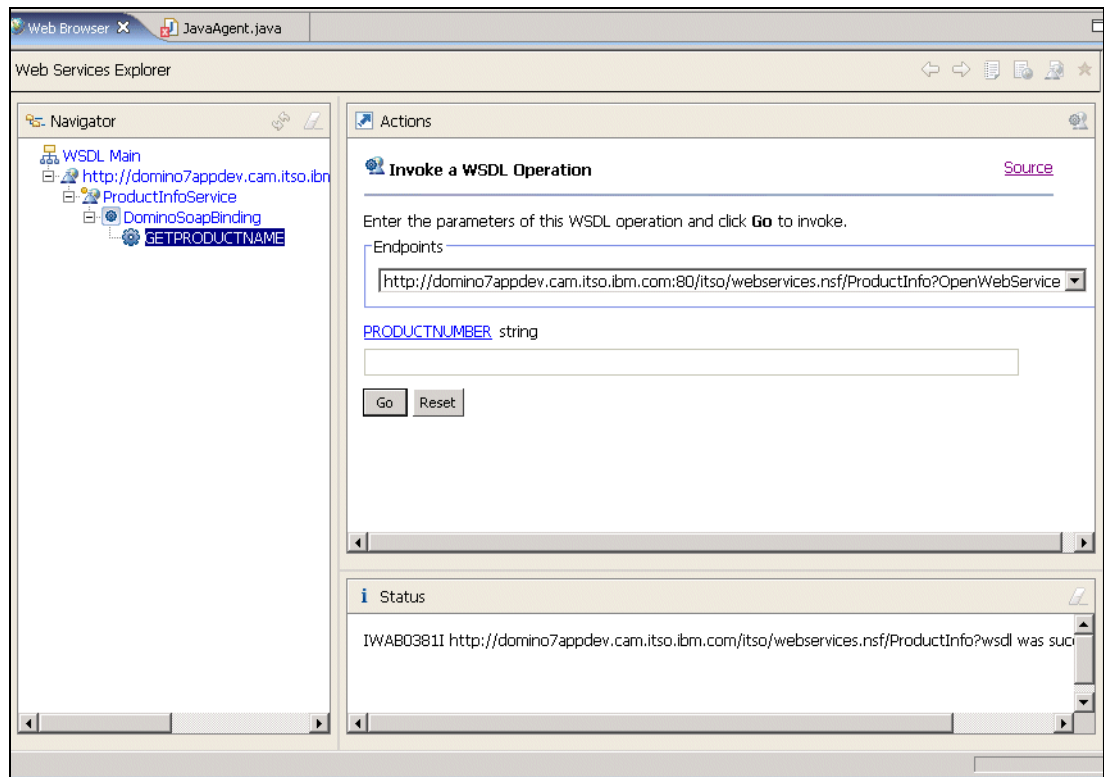


図 3-8 「Invoke a WSDL Operation」 ペイン

8. 「PRODUCTNUMBER」に値を入力し、「Go」をクリックします。「Status」ペインに結果が表示されます。このシナリオでは「PRODUCTNUMBER」に 45000 と入力します。図 3-9 に示されているように、結果の文字列として Redbook Basic を受け取ります。

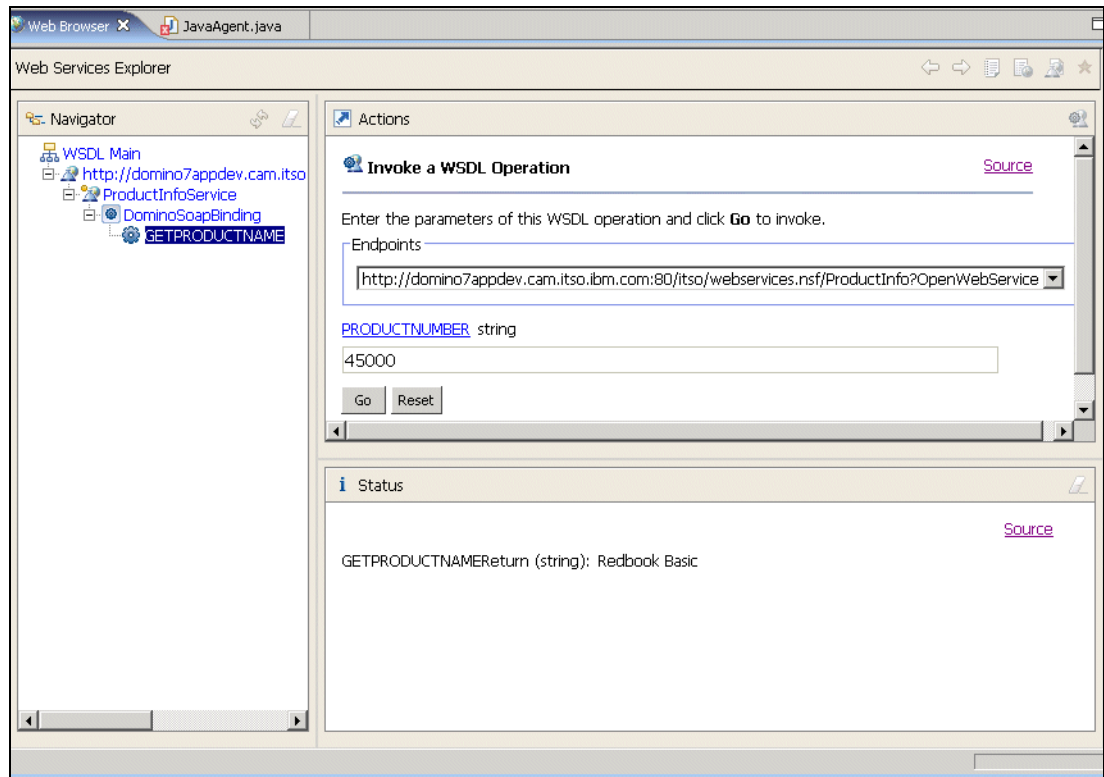


図 3-9 「Invoke a WSDL Operation」の「Status」ペイン

9. 背後で実際に実行された内容を表示するために、「Status」ペインで「Source」をクリックします。例 3-3 に、SOAP 要求エンベロープで Lotus Domino サーバーに送信されたコードを示します。Lotus Domino サーバーは、56 ページの例 3-4 に示されている SOAP 応答エンベロープを送信することで、このコードに応答します。このコードは、56 ページの図 3-10 に示されているように、「Status」ペインに表示されます。

#### 例 3-3 SOAP 要求エンベロープ

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <SOAP-ENV:Body>
- <ns0:GETPRODUCTNAME xmlns:ns0="urn:DefaultNamespace"
  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <PRODUCTNUMBER xsi:type="xsd:string">45000</PRODUCTNUMBER>
  </ns0:GETPRODUCTNAME>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

#### 例 3-4 サーバーが送り返す SOAP 応答エンベロープ

```

soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
- <soapenv:Body>
- <ns1:GETPRODUCTNAMEResponse xmlns:ns1="urn:DefaultNamespace"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <GETPRODUCTNAMEReturn xsi:type="xsd:string">Redbook Basic</GETPRODUCTNAMEReturn>
</ns1:GETPRODUCTNAMEResponse>
</soapenv:Body>
</soapenv:Envelope>

```

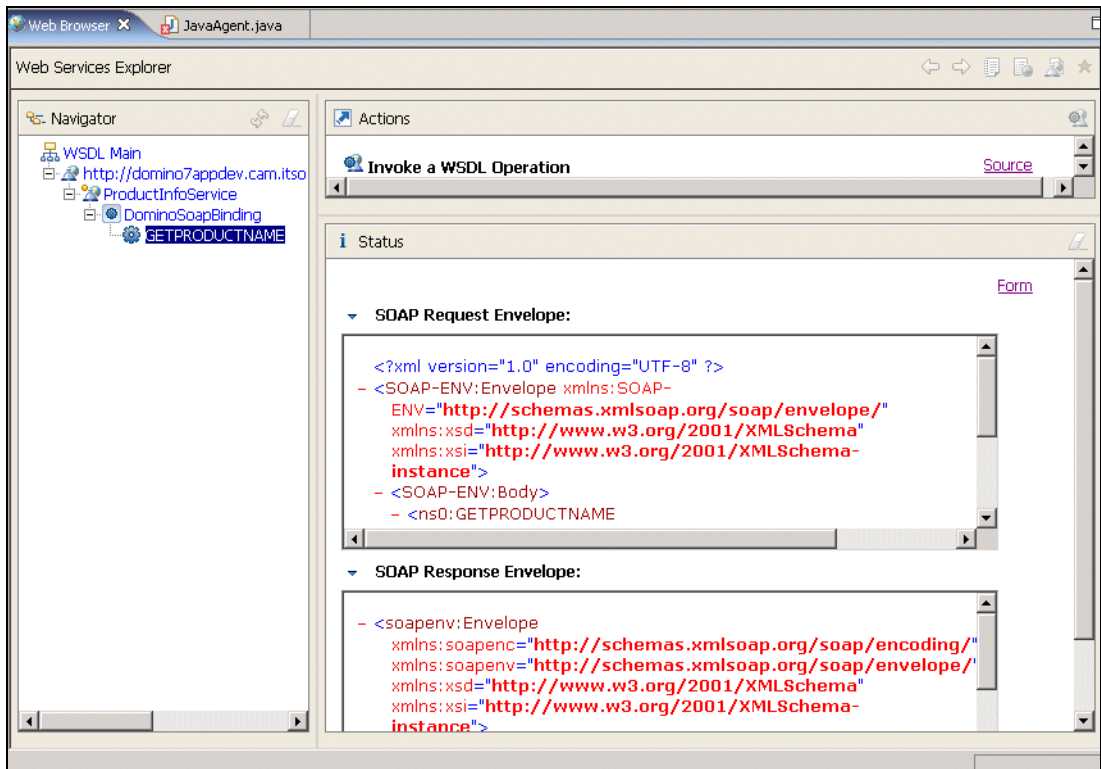


図 3-10 「Status」 ペインに表示されたソース・コード

- 最後に XML フォーマットで違いを確認するために、Lotus Domino Designer に戻り、詳細なプロパティ・ウィンドウで、56 ページの図 3-11 に示されているように SOAP メッセージのフォーマットを「ラップ」に変更します。もう一度 Rational Application Developer から Web サービス操作の「GETPRODUCTNAME」を呼び出します。57 ページの例 3-5 および 57 ページの例 3-6 に示されているような結果が表示されます。



図 3-11 「SOAP メッセージ形式」の「ラップ」

例 3-5 「SOAP メッセージ形式」の「ラップ」を使用した場合の SOAP 要求エンベロープ

```
<?xml version="1.0" encoding="UTF-8" ?>
- <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:q0="urn:DefaultNamespace" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <SOAP-ENV:Body>
- <q0:GETPRODUCTNAME>
  <PRODUCTNUMBER>45000</PRODUCTNUMBER>
</q0:GETPRODUCTNAME>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

例 3-6 「SOAP メッセージ形式」の「ラップ」を使用した場合の SOAP 応答エンベロープ

```
- <soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <ns1:GETPRODUCTNAMEResponse xmlns:ns1="urn:DefaultNamespace">
  <GETPRODUCTNAMEReturn>Redbook Basic</GETPRODUCTNAMEReturn>
</ns1:GETPRODUCTNAMEResponse>
</soapenv:Body>
</soapenv:Envelope>
```

ヒント：Web サービス・ブラウザーは、Web サービスから最新の WSDL 文書を取得する必要があります。WSDL 文書を変更した場合、コンシューマー (Web サービス・ブラウザーなど) には必ず最新バージョンの WSDL を使用させる必要があります。このようにしないと、操作が期待どおりに処理されない場合があります。

## 3.6 Lotus Domino Designer 7 での複雑な Web サービスの作成

前の例では、プリミティブ・データ型を共有する方法を説明しました。プリミティブ・データ型とは、配列、数値、文字列などで、前の例では文字列を使用しました。次に、複雑なデータ型の実装について説明します。ITSO Electronics アプリケーションのビジネス要件は、顧客が製品情報に直接アクセスできるようにすることです。Web サービスを使用することで、このデータを公開できます。顧客とサプライヤーは特定製品の価格を検索できますし、入手できるすべての製品のコレクションを取得できます。

顧客が製品情報にアクセスできるようにするためには、次の製品情報を共有する必要があります。

- ▶ 製品名
- ▶ 価格
- ▶ 数量
- ▶ 説明

例 3-7 に、Product オブジェクトのデータ表現を示します。このオブジェクトは、このセクションでこの後作成する Web サービスで複雑なデータ型として使用します。この例では、プログラミング言語として Java が使用されています。

例 3-7 Product.java のコード

```
public class Product
{
```

```

private String name;
private int number;
private String description;
private double price;
public String getName(){ return name; }
public int getNumber(){ return number; }
public double getPrice(){ return price; }
public String getDescription() { return description; }

public void setName(String newName)
{
    name = newName;
}
public void setNumber(int newNumber)
{
    number = newNumber;
}
public void setDescription(String newDescription)
{
    description = newDescription;
}
public void setPrice(double newPrice)
{
    price = newPrice;
}
}

```

---

48 ページの 3.5.1、「新しい Web サービスの作成」では、プリミティブ・データ型を使用して Web サービスを作成しました。この例では、同じ手順を使用して、パートナーに詳細な製品情報を提供する Product オブジェクト用の Web サービスを作成します。以下の手順に、この Web サービスの作成方法をまとめます。

1. Lotus Domino Designer で ITSO Web Services データベースを開き、「共有コード」→「Web サービス」を選択します。「新規 Web サービス」をクリックします。
2. Web サービスの名前として ProductService を使用し、ポートタイプ・クラスとして ProductService を指定します。
3. 「(Options)」ペインで、「Java」に切り替えます。
4. 「新規クラス」をクリックし、デフォルトのコードを 58 ページの例 3-8 の ProductService.java のコードに置き換えます。
5. 「新規クラス」をもう一度クリックし、57 ページの例 3-7 の Product.java のコードを貼り付けます。
6. Web サービスを保存します。

#### 例 3-8 ProductService.java

---

```

import lotus.domino.*;
import lotus.domino.types.*;
import java.util.*;

public class ProductService
{
    private Session session = WebServiceBase.getCurrentSession();
    public Object[] getAllProducts()
    {
        Product[] result = null;
        Database db = null;
        View view = null;
    }
}

```



```

        Product product = null;
        ArrayList tempList = new ArrayList();
        try
        {
            db = getProductDb();
            view = db.getView("vaPNumber");
            Document doc2, doc = view.getFirstDocument();
            while(doc != null)
            {
                product = getProductFromDocument(doc);
                tempList.add(product);
                doc2 = doc;
                doc = view.getNextDocument(doc);
                doc2.recycle();
            }
            view.recycle();
            db.recycle();
        } catch (NotesException e) { e.printStackTrace(); }
        return tempList.toArray();
    }
    private Product getProductFromDocument(Document doc)
    {
        Product product = null;
        try
        {
            String name = doc.getItemValueString("ProductName");
            String description = doc.getItemValueString("ProductDescription");
            double price = doc.getItemValueDouble("ProductPrice");
            int productNumber = doc.getItemValueInteger("productNumber");
            product = new Product();
            product.setNumber(productNumber);
            product.setName(name);
            product.setPrice(price);
            product.setDescription(description);
        }
        catch (Exception e)
        {}
        return product;
    }
    private Database getProductDb()
    {
        Database db = null;
        try{
            db = session.getDatabase("", "itso¥¥products.nsf");
        } catch (NotesException e) {}
        return db;
    }
    public Product getProduct(double productNumber)
    {
        Product product = null;
        try
        {
            Database db = getProductDb();
            View view = db.getView("vaPNumber");
            Document doc = view.getDocumentByKey(new Double(productNumber), true);
            if(doc != null)
            {
                product = getProductFromDocument(doc);
                doc.recycle();
                doc = null;
            }
        }
        catch (Exception e)
        {}
        return product;
    }

```

```

    }
    view.recycle();
    db.recycle();
  } catch (NotesException e){}
  return product;
}
}

```

この ProductService という名前の新しい Web サービスは、図 3-12 に示されているように、Rational Application Developer からコンシュームできます。

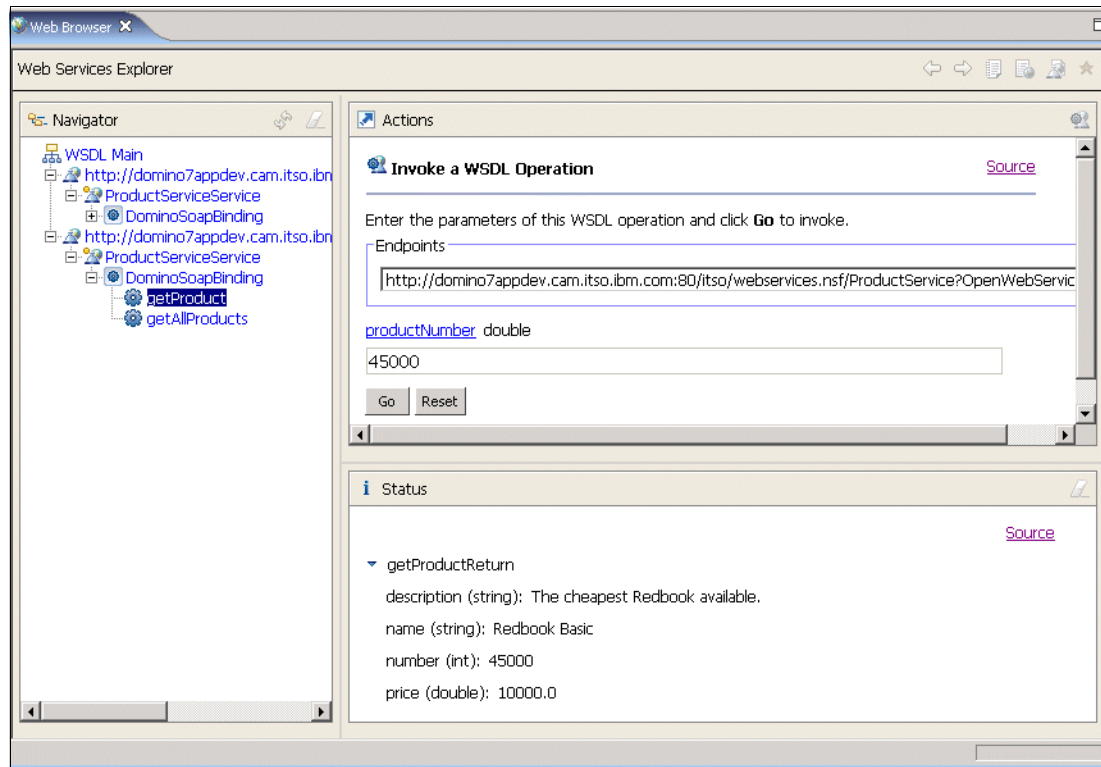


図 3-12 WSDL 操作の getProduct の呼び出し

例 3-9 に、Lotus Domino サーバーから戻される、特定の製品番号に対応する SOAP 応答エンベロープを示します。特定の製品番号を使用して WSDL 操作の getProduct を呼び出すと、この SOAP 応答エンベロープが戻されます。

#### 例 3-9 サーバーが送り返す SOAP 応答エンベロープ

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getProductResponse xmlns:ns1="urn:DefaultNamespace"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <getProductReturn href="#id0" />
    </ns1:getProductResponse>
  </multiRef xmlns:ns2="urn:DefaultNamespace"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id0" soapenc:root="0"
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns2:Product">
    <description xsi:type="xsd:string">This is the basic Redbook offering.</description>
    <name xsi:type="xsd:string">Basic Redbook</name>
  </multiRef>
</soapenv:Envelope>

```

```

<number xsi:type="xsd:int">20501</number>
<price xsi:type="xsd:double">102.0</price>
</multiRef>
</soapenv:Body>
</soapenv:Envelope>

```

作成した ProductService には、配列として製品オブジェクトのコレクションを送信する、getAllProducts という名前の別の操作も備わっています。例 3-10 に、RPC/ エンコードの SOAP メッセージ・フォーマットでこのコレクションを示します。

#### 例 3-10 SOAP 応答エンベロープ

```

- <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
- <soapenv:Body>
- <ns1:getAllProductsResponse xmlns:ns1="urn:DefaultNamespace"
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
- <getAllProductsReturn xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  soapenc:arrayType="xsd:anyType[9]" xsi:type="soapenc:Array">
  <item href="#id0" />
  <item href="#id1" />
  <item href="#id2" />
  <item href="#id3" />
  <item href="#id4" />
  <item href="#id5" />
  <item href="#id6" />
  <item href="#id7" />
  <item href="#id8" />
</getAllProductsReturn>
</ns1:getAllProductsResponse>
- <multiRef xmlns:ns2="urn:DefaultNamespace"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id1" soapenc:root="0"
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns2:Product">
  <description xsi:type="xsd:string">Redpaper mid-tier line available at newsagents and
  department stores.</description>
  <name xsi:type="xsd:string">Redpaper Deluxe</name>
  <number xsi:type="xsd:int">10002</number>
  <price xsi:type="xsd:double">2299.0</price>
</multiRef>
- <multiRef xmlns:ns3="urn:DefaultNamespace"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id8" soapenc:root="0"
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns3:Product">
  <description xsi:type="xsd:string">The best Redbook money can buy.</description>
  <name xsi:type="xsd:string">Redbook Plus</name>
  <number xsi:type="xsd:int">45500</number>
  <price xsi:type="xsd:double">50000.0</price>
</multiRef>

```

and so on.....

```

- <multiRef xmlns:ns10="urn:DefaultNamespace"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" id="id6" soapenc:root="0"
  soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xsi:type="ns10:Product">
  <description xsi:type="xsd:string">The ultimate Redbook experience.</description>
  <name xsi:type="xsd:string">Redbook Supreme</name>
  <number xsi:type="xsd:int">20564</number>
  <price xsi:type="xsd:double">1000.0</price>
</multiRef>
</soapenv:Body>

```

</soapenv:Envelope>

## 3.7 WSDL 文書のエクスポート

WSDL 文書をエクスポートして、Web サービスの呼び出し方法に関する情報を顧客に提供します。多くの場合、Web サービス・クライアントを作成するツールで WSDL 文書からコードを自動生成できます。

WSDL 文書をエクスポートするには、次の手順に従います。

1. Lotus Domino Designer で ITSO Web Services データベースを開き、「共有コード」→「Web サービス」を選択します。
2. **ProductService** を選択し「WSDL のエクスポート」をクリックします。
3. XML ファイルである WSDL ファイルが生成されます。このシナリオでは、62 ページの図 3-13 に示されているように、.wsdl 拡張子を使用してエクスポートした WSDL ファイルを識別します。

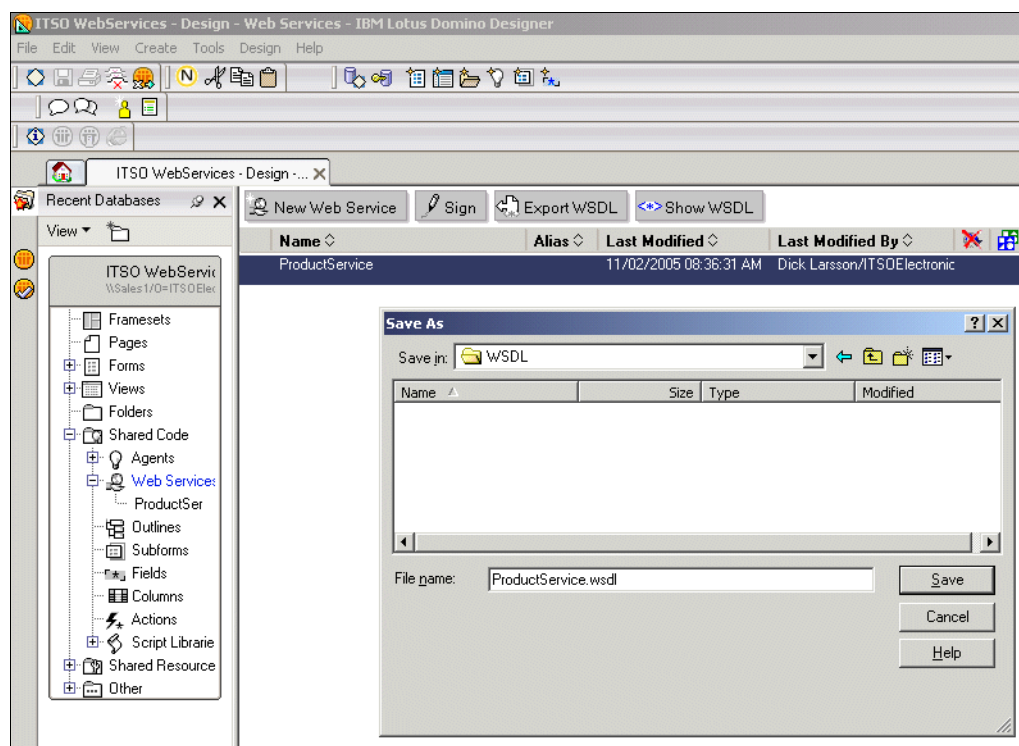


図 3-13 WSDL 文書のエクスポート

## 3.8 WSDL 文書のインポート

WSDL 文書を Web サービスにインポートすると、WSDL の内容に基づいて実装クラスが生成されます。クラスの public プロトタイプ・メソッド、関数、およびサブルーチンは、WSDL 文書の最初の <service> 要素の最初の <port> 要素で定義されている Web サービス操作に対応します。プロトタイプ・メソッドで参照されるその他のクラスまたは型も、public データ・メンバーと共に WSDL 文書から生成されます。プロトタイプ・インターフェースを変更すると、Web サービスを保存するときに対応する WSDL 文書も変更されます。「WSDL

インターフェースの更新を警告」プロパティを使用することで、このような変更をモニターおよび防止できます。

ITSO Electronics のビジネス・パートナー、場合によっては顧客は、お返しに、Web サービスを作成するために会社が使用できる WSDL 文書を提供できます。既存の WSDL 文書をインポートして、スケルトン Web サービスを生成できます。このスケルトン・コードは Web サービスの定義に対応します。生成後、実装コードを追加します。

WSDL 文書をインポートするには、次の手順に従います。

1. Lotus Domino Designer で ITSO Web Services データベースを開き、「共有コード」→「Web サービス」を選択します。
2. 「WSDL のインポート」をクリックし、インポートする WSDL ファイルを選択します。このシナリオでは、63 ページの図 3-14 に示されているように、前のセクションでエクスポートしたファイルをインポートします。

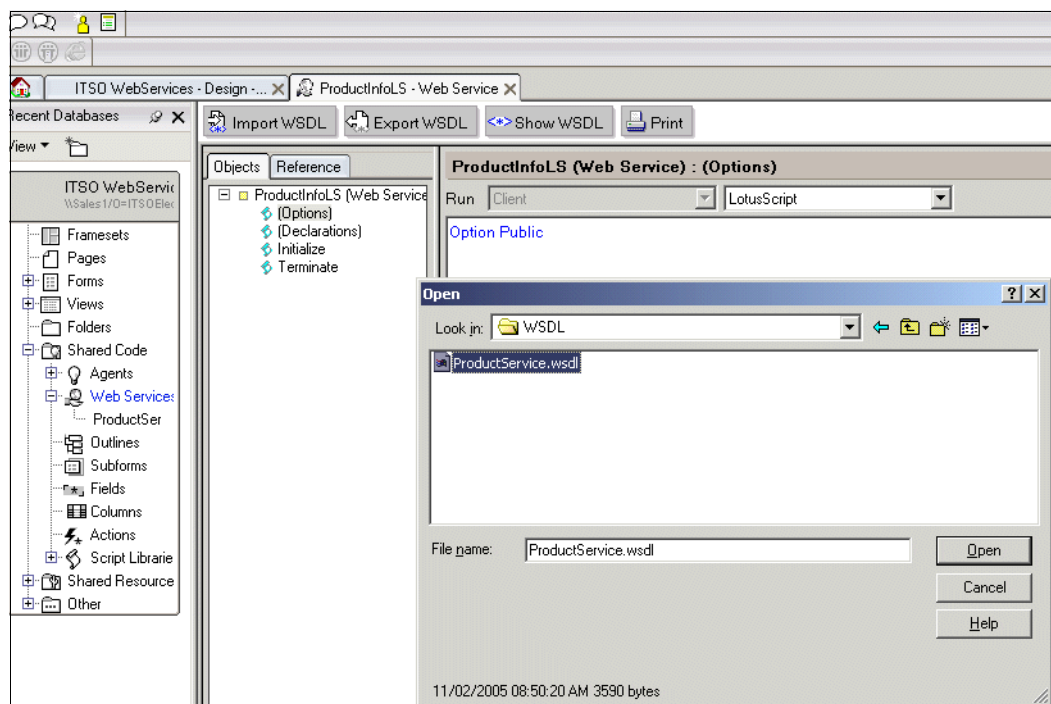


図 3-14 WSDL 文書のインポート

3. WSDL 文書をインポートすると、スケルトン・コードが生成されます。このシナリオでは、64 ページの例 3-11 に示されているように、LotusScript のスケルトン・コードが生成されます。「Web サービス」プロパティ・ウィンドウを開き、図 3-15 に示されているように「WSDL インターフェースの更新を警告」オプションを選択します。

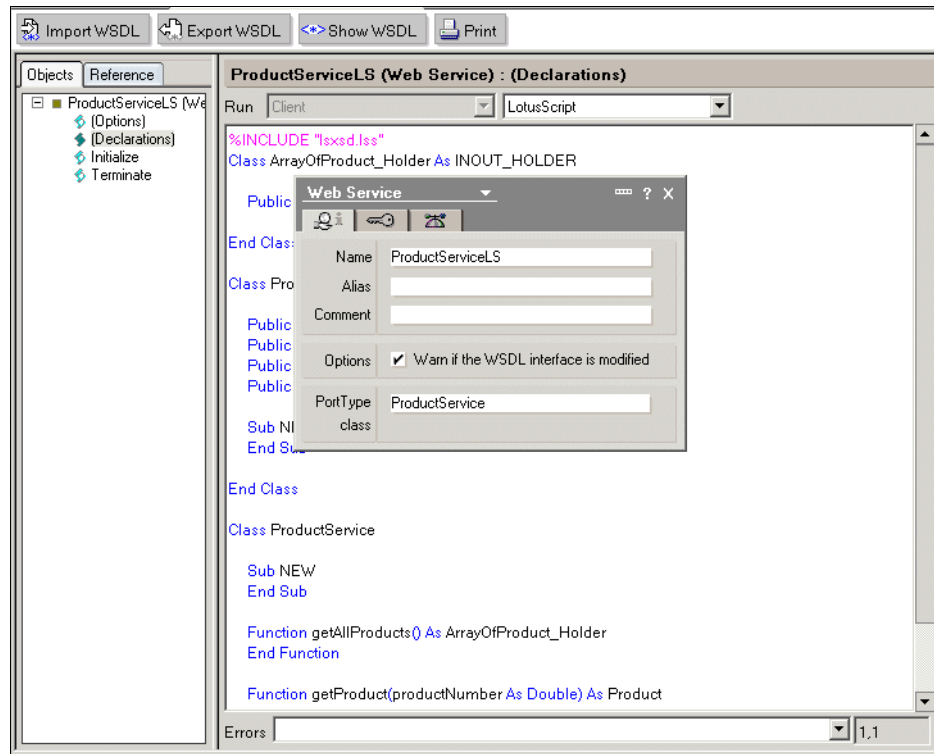


図3-15 WSDL 文書のインポート: 「Web サービス」プロパティ・ウィンドウ

例3-11 WSDL 文書のインポート後に生成されるスケルトン・コード

```
%INCLUDE "lsxsd.1ss"
Class ArrayOfProduct_Holder As INOUT_HOLDER

Public Value() As Product

End Class

Class Product

Public description As XSD_STRING
Public name As XSD_STRING
Public number As Long
Public price As Double

Sub NEW
End Sub

End Class

Class ProductService

Sub NEW
End Sub

Function getAllProducts() As ArrayOfProduct_Holder
End Function

Function getProduct(productNumber As Double) As Product
End Function
```

## 3.9 Web サービスでの例外およびエラー処理

操作で呼び出し側にエラーを戻すこともできます。WSDL 文書の操作に対し指定した `<wsdl:fault>` メッセージに対応する `fault` サブクラスを使用して戻すか、または `WS_FAULT` 基本クラス (LotusScript)、`lotus.domino.types.Fault` クラス、または `java.lang.Exception` クラス (Java) を直接使用して戻します。

**注:** LotusScript のサービス実装メソッドの障害 (つまり、`WS_FAULT` 基本クラスまたはそのサブクラス) は、メソッドの引数リストの末尾に記述する必要があります。

Java のサービス実装メソッドの障害は、従来の `Java throws` 節でメソッドに対し記述します。

LotusScript を使用して障害を処理する方法については、71 ページの 3.12、「LS2J を使用した Web サービスのコンシューム」を参照してください。

## 3.10 Web サービスのセキュリティ

ITSO Electronics アプリケーションに Web サービスを実装する場合、ホスト・システム、アプリケーション、およびネットワーク・インフラストラクチャー全体に対する脅威が、その Web サービスに対しても脅威となります。Web サービスを保護するには、一連の XML ベースのセキュリティ・メカニズムで、中間者の存在を許可する認証、役割ベースのアクセス制御、分散セキュリティ・ポリシーによる強制、およびメッセージ層のセキュリティに関連する問題を解決する必要があります。

現時点では、Web サービスのセキュリティに関して広く適用されている仕様はありません。したがって、開発者はこれらの機能を使用しないサービスを構築してもかまいません。また、その場その場でソリューションを開発してもかまいませんが、そのようなソリューションは相互運用性の問題を引き起こす可能性があります。

Web サービスの実装では、脅威またはリスクの程度により、2 地点間のセキュリティ・メカニズム、終端間のセキュリティ・メカニズムのうち的一方または両方が必要になる場合があります。従来からあるコネクション指向の 2 地点間セキュリティ・メカニズムでは、Web サービスの終端間セキュリティ要件を満たさない場合があります。ただし、セキュリティは、アクセスされることのリスクと対抗手段のコストのバランスです。実装のリスク許容度に従い、2 地点間でトランスポート・レベルのセキュリティを用意することで、十分なセキュリティ対抗手段となります。

ITSO Electronics アプリケーションのアーキテクチャーの視点から見ると、セキュリティに関連する基本概念は 3 つあります。1 つは保護する必要があるリソース、1 つはそれらのリソースを保護するメカニズム (つまり、ポリシーによる防御)、最後の 1 つは、それらのリソースに対する制約を記述した、マシンが処理できる文書であるポリシーです。セキュリティの考慮事項に関する詳細について、および Web サービスを実装することでの影響については、「*Web Service Architecture*」ガイドを参照してください。以下から入手できます。

<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>

Lotus Domino 7 の Web サービスには、エージェントと同じセキュリティ機能があります。Web サービスのセキュリティは、「Web サービス」プロパティ・ウィンドウの「セキュリティ」タブで設定します。以下のリストは、「セキュリティ」タブで使用できるオプションの説明です。

- ▶ 「Web ユーザーで実行」：Java または LotusScript コードが、Web サービスを呼び出すユーザーの有効なユーザー名で実行されます。
- ▶ 「代理で実行」：Java または LotusScript コードが、指定したユーザーの権限で実行されます。ただし、Web サービスの署名者が、サーバーにおける無制限の権限を所有している必要、つまりすべてのユーザーに代わって実行できる権限を所有している必要があります。
- ▶ 「実行時セキュリティレベルの設定」(以下のいずれかを選択):
  - 「制限された操作を許可しない」
  - 「制限された操作を許可する」
  - 「フルアドミニストレーター権限で制限された操作を許可する」
- ▶ 「Web サービスへのデフォルトアクセス」
  - 「読者以上すべて」。このオプションは選択またはクリアできます。
  - 「名前選択」。「読者以上すべて」をクリアした場合、デフォルト・アクセスを許可するユーザーを選択できます。
- ▶ 「パブリックアクセスユーザーの Web サービス使用を許可する」。データベース内の文書へのパブリック・アクセス権を所有するユーザーに対し、デフォルト・アクセスを許可します。

## 3.11 Java を使用した Web サービスのコンシューム

本書を執筆した時点では、Lotus Domino 7.0 は Web サービスを提供していますが、ネイティブではコンシュームしません。しかし、ITSO Electronics の第 2 のビジネス要件は、Lotus Domino の制約を超えて価格データにアクセスできるようにすることです。したがって、アプリケーションで Web サービスをコンシュームする必要があります。シナリオでは、ITSO Electronics は、Java を使用して Web サービスをコンシュームするために、オープン・ソース製品の Apache SOAP の使用を選択します。Apache SOAP (Simple Object Access Protocol) は、W3C に準拠して SOAP を実装しています。Apache SOAP は、IBM SOAP4J の実装をベースとしており、IBM SOAP4J の代わりに使用できます。後続のプロジェクトは Apache Axis です。Apache Axis は以下から入手できます。

<http://ws.apache.org/axis/>

シナリオで Web サービスのコンシュームを説明するために、ITSO Electronics アプリケーションの新しいデータベースを作成します。このデータベースの名前は WS Consumer です。このデータベースには、先に作成した複雑な Web サービスをコンシュームするエージェントを含めます。

**注：** Web サービスはどの Lotus Notes/Domino データベースにも追加できます。ただし、本書ではその目的のために、すべての Web サービスを、ITSO Electronics アプリケーションの個別のデータベース (webservices.nsf および wsconsumer.nsf) に追加します。

以下は、Web サービスをコンシュームするエージェントの作成手順です。

1. 以下から Apache SOAP の最新バージョンをダウンロードします。  
<http://ws.apache.org/soap/>
2. ダウンロードしたファイルを一時ディレクトリーに解凍します。このシナリオでは c:\\*soap です。
3. (Web Services データベース内の ) ProductService という名前の複雑な Web サービスを WSDL 文書にエクスポートします。エクスポートした WSDL 文書には、Web サービスをコンシュームするのに必要な情報がすべて含まれています。



4. テキスト・エディターで WSDL 文書を開きます。WSDL 文書には、例 3-12 に示されているように、サービスが戻す複雑なデータ型を定義する情報が含まれます。このケースでは、複雑なデータ型の名前は Product です。complexType 要素により、Product オブジェクトのプロパティー (description、name、number、および price) が示されています。

例 3-12 Product の WSDL 型

```
<wsdl:types>
  <schema targetNamespace="urn:DefaultNamespace" xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
    <complexType name="Product">
      <sequence>
        <element name="description" nillable="true" type="xsd:string"/>
        <element name="name" nillable="true" type="xsd:string"/>
        <element name="number" type="xsd:int"/>
        <element name="price" type="xsd:double"/>
      </sequence>
    </complexType>
    <complexType name="ArrayOfProduct">
      <complexContent>
        <restriction base="soapenc:Array">
          <attribute ref="soapenc:arrayType" wsdl:arrayType="impl:Product[]" />
        </restriction>
      </complexContent>
    </complexType>
  </schema>
</wsdl:types>
```

5. Lotus Domino Designer で新しい WS Consumer データベースを開き、「共有コード」→「エージェント」を選択し、「新規エージェント」をクリックします。
6. 図 3-16 に示されているように、エージェントの「名前」に ConsumeWS を設定し、実行の対象で「なし」を選択します。

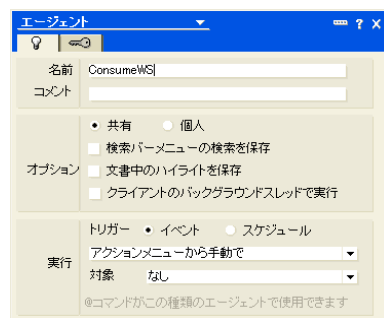


図 3-16 ConsumerWS エージェントのプロパティー・ウィンドウ

7. 「エージェント」プロパティー・ウィンドウを閉じて、「アクション」で「Java」を選択します。
8. 複雑なデータ型の Product を表す JavaBean クラスを、WSDL 文書の指定どおりに作成する必要があります。これを簡単に行うには、WSDL をインポートする機能を利用して JavaBean クラスを自動的に生成します。設計リストに戻り、「共有コード」→「Web サービス」を選択し、「新規 Web サービス」をクリックします。
9. 「Web サービス」プロパティー・ウィンドウを閉じて、「(Options)」が「Java」に設定されていることを確認します。「WSDL のインポート」をクリックし、先に作成したエクスポート済みの WSDL 文書を選択します。インポートにより JavaBeans™ クラスが自動的に作成されます。

10. Product.java の Java コードを選択しコピーします。
11. エージェントに戻り、「**新規クラス**」をクリックします。次に、69 ページの例 3-13 に示されているように、Untitled.java クラスをコピーした Product.java コードに置き換えます。
12. ConsumeWS エージェントを保存し、コードのコンパイルが成功したことを確認します。  
「**プロジェクトの編集**」をクリックします。図 3-17 に示されている「Java エージェントファイルの構成」が開きます。正しい SOAP LIB ライブラリ・ディレクトリーに移動します。この例の場合は c:\soap\soap-2\_3\_1\lib です。「**アーカイブ**」ファイル・タイプを選択し、「**ファイルの追加 / 置換**」をクリックし、「**OK**」をクリックします。

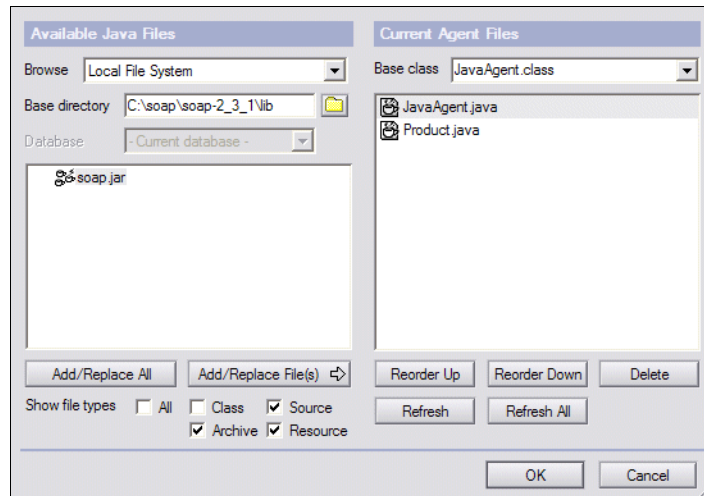


図3-17 「Java エージェントファイルの構成」ウィンドウ

13. デフォルトの JavaAgent.java コードを、Web サービスをコンシュームする独自のコードに置き換える必要があります。このシナリオでは、70 ページの例 3-14 のコードに置き換えます。JavaAgent.java クラスに URL 変数および soapAction 変数があることを確認します。これらの変数は、アプリケーションの場所を指している必要があります。
14. ConsumeWS エージェントを保存します。
15. 新しい Web サービスに戻り、保存せずに閉じます。
16. Lotus Notes クライアントを開き、WS Consumer データベースを選択します。「**アクション**」→「**ConsumeWS**」に移動し、新しく作成したエージェントを起動します。図 3-18 に示されているように、Web サービスがコンシュームする Product オブジェクトのコレクションがエージェントにより Java ウィンドウに表示されます。

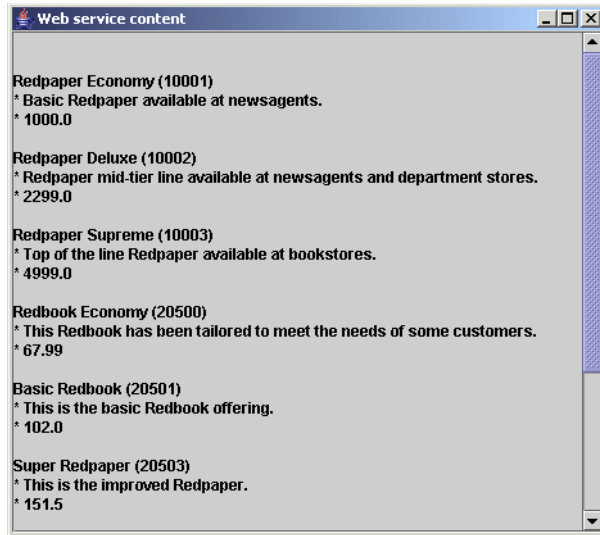


図3-18 Web サービスの結果を表示するJava ウィンドウ

このウィンドウが開いていない場合は、「ファイル」→「ツール」→「Java デバッグコンソールの表示」を選択してウィンドウを開き、情報を確認します。

#### 例3-13 Product クラスのコード

```
public class Product {
    private java.lang.String description;
    private java.lang.String name;
    private int number;
    private double price;

    public Product() {
    }

    public java.lang.String getDescription() {
        return description;
    }

    public void setDescription(java.lang.String description) {
        this.description = description;
    }

    public java.lang.String getName() {
        return name;
    }

    public void setName(java.lang.String name) {
        this.name = name;
    }

    public int getNumber() {
        return number;
    }

    public void setNumber(int number) {
        this.number = number;
    }

    public double getPrice() {
```

```

        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    private java.lang.Object __equalsCalc = null;
    public synchronized boolean equals(java.lang.Object obj) {
        if (!(obj instanceof Product)) return false;
        Product other = (Product) obj;
        if (obj == null) return false;
        if (this == obj) return true;
        if (__equalsCalc != null) {
            return (__equalsCalc == obj);
        }
        __equalsCalc = obj;
        boolean _equals;
        _equals = true &&
            ((this.description==null && other.getDescription()==null) ||
             (this.description!=null &&
              this.description.equals(other.getDescription()))) &&
            ((this.name==null && other.getName()==null) ||
             (this.name!=null &&
              this.name.equals(other.getName()))) &&
            this.number == other.getNumber() &&
            this.price == other.getPrice();
        __equalsCalc = null;
        return _equals;
    }

    private boolean __hashCodeCalc = false;
    public synchronized int hashCode() {
        if (__hashCodeCalc) {
            return 0;
        }
        __hashCodeCalc = true;
        int _hashCode = 1;
        if (getDescription() != null) {
            _hashCode += getDescription().hashCode();
        }
        if (getName() != null) {
            _hashCode += getName().hashCode();
        }
        _hashCode += getNumber();
        _hashCode += new Double(getPrice()).hashCode();
        __hashCodeCalc = false;
        return _hashCode;
    }
}

```

---

*例3-14 Java を使用してWeb サービスをコンシュームするためのコード*

---

```

import lotus.domino.*;
import java.io.*;
import java.util.*;
import java.net.*;
import org.w3c.dom.*;
import org.apache.soap.util.xml.*;
import org.apache.soap.*;
import org.apache.soap.encoding.*;
import org.apache.soap.encoding.soapenc.*;

```

```

import org.apache.soap.rpc.*;
import javax.swing.*;

public class JavaAgent extends AgentBase
{
    public void NotesMain()
    {
        try
        {
            Session session = getSession();
            AgentContext agentContext = session.getAgentContext();

            URL url = new URL("http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductService?WSDL");
            String soapAction = "http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductService?WSDL";

            Call call = new Call();
            BeanSerializer beanSerializer = new BeanSerializer();
            SOAPMappingRegistry smRegistry = new SOAPMappingRegistry();

            smRegistry.mapTypes(Constants.NS_URI_SOAP_ENC, new QName(
                "urn:DefaultNamespace", "Product"), Product.class, beanSerializer,
                beanSerializer);

            String targetNamespace = "urn:DefaultNamespace";
            call.setSOAPMappingRegistry(smRegistry);
            call.setTargetObjectURI(targetNamespace);
            call.setMethodName("getAllProducts");
            call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);

            Response resp = null;
            try
            {
                resp = call.invoke(url, soapAction);
            } catch (SOAPException e)
            {
                e.printStackTrace();
                System.err.println("SOAP Exception code: " + e.getFaultCode() + " message: " + e.getMessage());
                return;
            }

            if (!resp.generatedFault())
            {
                Parameter ret = resp.getReturnValue();
                Product[] products = (Product[]) ret.getValue();
                buildGUI(products);
            }
            else
            {
                Fault fault = resp.getFault();
                System.err.println("SOAP Fault, " + fault.getFaultCode() + " " + fault.getFaultString());
            }
        } catch (Exception e)
        {
            e.printStackTrace();
        }
    }

    private void buildGUI(Product[] products)
    {
        StringBuffer sb = new StringBuffer();
        String newLine = "<br>";
        for(int i = 0; i < products.length; i++)
        {
            sb.append(newLine + newLine);
            sb.append(products[i].getName() + " (" + products[i].getNumber() + ") " + newLine);
            sb.append(" * " + products[i].getDescription());
            sb.append(newLine + " * " + products[i].getPrice());
        }

        JFrame frame = new JFrame("Web service content");
        frame.getContentPane().add(new JScrollPane(new JLabel("<html>" + sb.toString() + "</html>")));
        frame.pack();
        frame.setVisible(true);
    }
}

```

---

## 3.12 LS2J を使用した Web サービスのコンシューム

このセクションでは、以下について説明します。

- ▶ LotusScript を使用して Web サービスをコンシュームする方法
- ▶ LS2J を使用して LotusScript エージェントから java コードを呼び出す方法
- ▶ Web サービスから送り返される例外またはエラーを処理する方法
- ▶ 認証と Web サービス

### 3.12.1 LS2J について

LS2J は、Java のデータ型から LotusScript のデータ型にデータを変換できるインターフェースで、LotusScript で Java オブジェクトのメソッドを実行できるようにします。LS2J を使用すると、LotusScript で Java オブジェクトを作成できますが、それらの Java オブジェクトが LotusScript 環境に対しネイティブであるかのように作成できます。

このシナリオでは、LotusScript エージェントを WS Consumer データベースに追加します。このエージェントは、スクリプト・ライブラリの Java コードを使用して Web サービスをコンシュームします。

コンシュームする Web サービスは、次の 2 つのうちのいずれかを実行できます。

- ▶ 現在のユーザーの名前を戻す
- ▶ Web サービスからクライアントにエラー・コードを送り返す方法を示す WS\_FAULT オブジェクトを戻す

### 3.12.2 Web サービスの作成

以下は、LS2J を使用する Web サービスの作成手順です。このセクションでは Web サービスの作成について説明し、以降のセクションでは Java スクリプト・ライブラリを作成する方法、および LS2J を介して LotusScript エージェントでそのスクリプト・ライブラリを使用する方法について説明します。以下の手順を実行します。

1. Lotus Domino Designer で Web Services データベースを開きます。
2. 「新規 Web サービス」ボタンをクリックします。
3. 言語が「LotusScript」に設定されていることを確認します。
4. 「(Declarations)」に移動し、例 3-15 に示されているコードを入力します。

例 3-15 LS2J Web サービスのコード

---

```
Dim session As NotesSession
Class UserInfo
    Function getUsername ( Fault1 As WS_FAULT)
        Set session = New NotesSession
        Dim doc As NotesDocument
        Set doc = session.DocumentContext

        If Cint(Rnd() * 10) > 5 Then
            getUsername = doc.remote_user(0)
        Else
            Call Fault1.setFault(True) ' required for fault activation
            Call Fault1.setFaultString("SUCCESS, We send a FAULT on purpose")
        End If

    End Function
End Class
```

---

5. Web サービスのプロパティ・ウィンドウを開き、例 3-19 に示されているように、「名前」および「ポートタイプ」クラスを UserInfo に設定します。

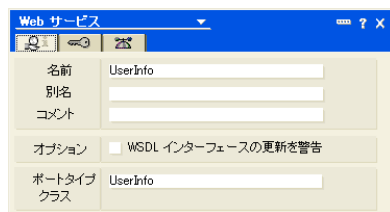


図 3-19 LS2J Web サービスの UserInfo プロパティのウィンドウ

6. Web サービスを保存して閉じます。

### 3.12.3 Java を使用して Web サービスをコンシュームするためのスクリプト・ライブラリの作成

次に、Web サービスをコンシュームし、それを LS2J を介して LotusScript エージェントでできるようにする Java クラスを含むスクリプト・ライブラリを作成する必要があります。Java クラスをセットアップするには、次の手順に従います。

1. Lotus Domino Designer で WS Consumer データベースを開きます。
2. ナビゲーション・ペインで「共有コード」をクリックし、「新規 Java ライブラリ」をクリックします。
3. 73 ページの図 3-20 に示されているように、新しく開いたウィンドウで「プロジェクトの編集」をクリックし、Apache の **soap.jar** ファイルを選択します。Apache の soap.jar ファイルがない場合は、66 ページの 3.11、「Java を使用した Web サービスのコンシューム」を参照して Apache SOAP のダウンロードに関する情報を確認してください。

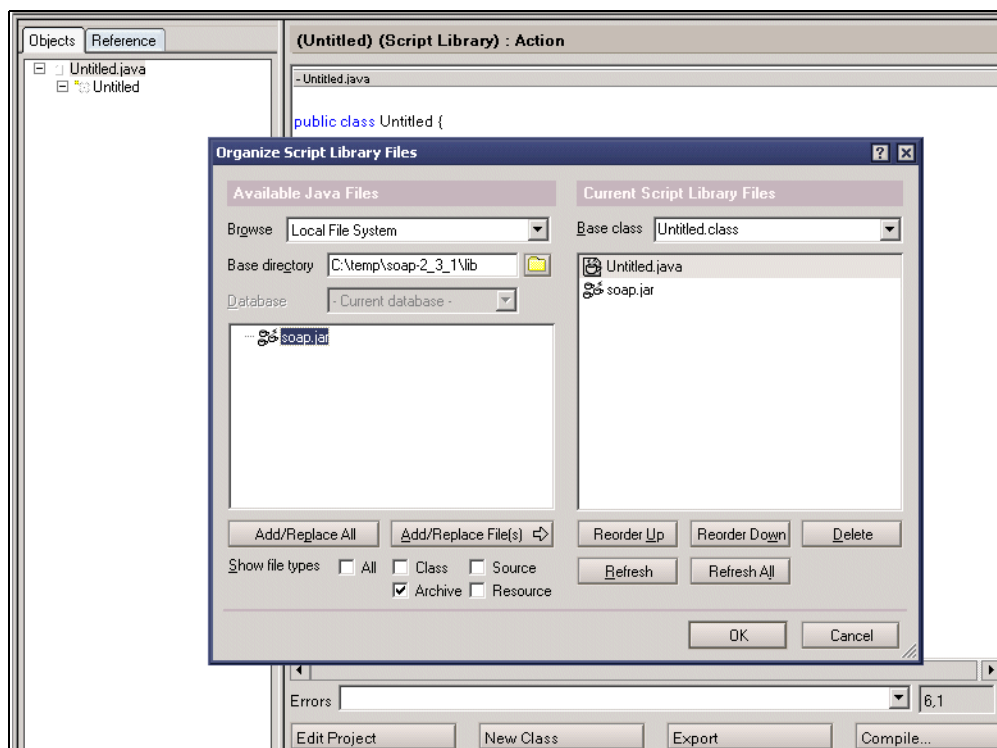


図 3-20 LS2J で使用する Java アーカイブのインポート

4. ConsumeWSLibrary という名前でスクリプト・ライブラリを保存します。

5. スクリプト・ライブラリのデフォルトのコードを 例 3-16 に示されているコードに置き換えます。

**重要：**例 3-16 にある次の行は、使用するデータベースの URL に置き換える必要があります。

```
URL url = new URL("http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductService?WSDL");
String soapAction = "http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/ProductService?WSDL";
```

例 3-16 LS2J を使用して Web サービスをコンシュームするための Java コード

```
import java.io.*;
import java.util.*;
import java.net.*;
import org.w3c.dom.*;
import org.apache.soap.util.xml.*;
import org.apache.soap.*;
import org.apache.soap.encoding.*;
import org.apache.soap.encoding.soapenc.*;
import org.apache.soap.rpc.*;
import org.apache.soap.transport.http.SOAPHTTPConnection;

public class WSConsumer
{
    public String getValue(String userName, String password)
    {
        try
        {
            URL url = new URL("http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/UserInfo?WSDL");
            String soapAction = "http://domino7appdev.cam.itso.ibm.com/itso/webservices.nsf/UserInfo?WSDL";

            Call call = new Call();
            SOAPHTTPConnection hc = new SOAPHTTPConnection();
            hc.setUserName(userName);
            hc.setPassword(password);
            call.setSOAPTransport(hc);

            SOAPMappingRegistry smRegistry = new SOAPMappingRegistry();
            String targetNamespace = "urn:DefaultNamespace";
            call.setSOAPMappingRegistry(smRegistry);
            call.setTargetObjectURI(targetNamespace);
            call.setMethodName("getUserName");
            call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);

            Response resp = null;
            try
            {
                resp = call.invoke(url, soapAction);
            } catch (SOAPException e)
            {
                e.printStackTrace();
                return ("SOAP Exception code: " + e.getFaultCode() + " message: " + e.getMessage());
            }

            if (!resp.generatedFault())
            {
                Parameter ret = resp.getReturnValue();
                return ret.getValue() + "";
            }
            else
            {
                Fault fault = resp.getFault();
                return ("SOAP Fault, " + fault.getFaultCode() + " " + fault.getFaultString());
            }
        } catch (Exception e)
        {
            e.printStackTrace();
        }
        return null;
    }
}
```



### 3.12.4 LotusScript エージェントの作成

最後に、LotusScript エージェントを作成し、新しく作成した Java スクリプト・ライブラリを LS2J を介して使用します。

1. 「ナビゲーション」ペインで、「共有コード」→「エージェント」を選択します。
2. 「新規エージェント」をクリックします。新しいエージェントのプロパティ・ウィンドウで、「名前」を LotusScriptConsumeWS に設定し、実行の対象を「なし」に設定します。プロパティ・ウィンドウを閉じます。
3. 言語を LotusScript に変更します。
4. 「Declarations」で、コードを 例 3-17 のコードに置き換えます。
5. 「Initialize」で、例 3-18 のコードに置き換えます。

注: value = myObject.getValue("The UserName", "The Password") という行には、有効なユーザー名およびパスワードを指定する必要があります。

例 3-17 LotusScript LS2J エージェントの宣言コード

```
Option Public
Use "ConsumeWSLibrary"
Use!sx "*javacon"
```

例 3-18 LotusScript LS2J エージェントの初期化コード

```
Sub Initialize
    Dim mySession As JavaSession
    Dim myClass As JavaClass
    Dim myObject As JavaObject

    Set mySession = New JavaSession ()
    Set myClass = mySession.GetClass("WSConsumer")
    Set myObject = myClass.CreateObject
    Dim value As String
    value = myObject.getValue("The UserName", "The Password")
    MsgBox value
End Sub
```

6. エージェントを保存して閉じます。
7. Lotus Notes クライアントで、新しく作成したエージェントを実行します。  
図 3-21 に示されているプロンプト・メッセージが表示された場合、正しいユーザー名およびパスワードが指定されていない可能性があります。

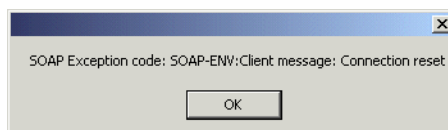


図 3-21 LS2J を使用して Web サービスをコンシュームする場合に表示されるエラー・メッセージ

エージェントが Web サービスをコンシュームできた場合、図 3-22 に示すようなプロンプト・メッセージが表示されます。

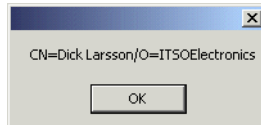


図3-22 LS2J を使用してコンシュームが成功した Web サービス

### 3.12.5 認証と Web サービス

Lotus Notes/Domino 7 では、Web サービスにエージェントと同じセキュリティ機能を用意しています。例 3-19 では、HTTP 基本認証により Web サービスが保護されます。

SOAPHTTPConnection クラスのインスタンスを Call オブジェクトに追加することで、Apache SOAP で SOAP 要求を使用してユーザー名およびパスワードを送信できます。

例3-19 Apache SOAP による基本認証

---

```
Call call = new Call();
SOAPHTTPConnection hc = new SOAPHTTPConnection();
hc.setUserName(userName);
hc.setPassword(password);
call.setSOAPTransport(hc);
```

---

## 3.13 どのスタイルの WSDL を使用するか

WSDL 文書には、Web サービスを表す XML コードが含まれます。WSDL バインディングは、メッセージング・プロトコル、特に SOAP メッセージング・プロトコルにどのようにサービスを結合するかを表します。WSDL SOAP バインディングは、リモート・プロシージャ・コール (RPC) スタイルのバインディング、または文書体裁のバインディングが可能です。また、SOAP バインディングはエンコードして使用することも、リテラルとして使用することも可能です。したがって、Lotus Domino アプリケーションの開発者は、次の 4 つのスタイル / 使用モデルのいずれかを選択できます。

- ▶ RPC/ エンコード
- ▶ RPC/ リテラル
- ▶ 文書 / リテラル
- ▶ ラップ

「RPC」という用語と「文書」という用語は紛らわしい場合があります。これは、これらの用語が、RPC プログラミング・モデルでは RPC スタイルを使用する必要がある、文書モデルまたはメッセージング・プログラミング・モデルでは文書体裁を使用する必要があるということを暗に示すからです。ただし、これはこのケースには当てはまりません。スタイルはプログラミング・モデルとは関係ないからです。スタイルは、単に WSDL バインディングを SOAP メッセージに変換する方法を決定するだけです。同様に、エンコードという用語とリテラルという用語も単に WSDL と SOAP のマッピングを意味するだけです。



## ITSO Electronics アプリケーションへの新しい設計要素の実装

Lotus Domino Designer 7 の新機能は、発展し続ける Web 標準とのより緊密な統合、その他の IBM ソフトウェア標準との相互運用性の強化、およびアプリケーション開発者とエンド・ユーザーの使いやすさの向上をテーマに構築されています。

この章では、Lotus Domino Designer 7 の新機能を使用して ITSO Electronics アプリケーションを拡張する方法を説明します。次の新機能に焦点を当てます。

- ▶ 右クリック・メニューのビュー・アクション
- ▶ 自動保存のフォームのプロパティ
- ▶ 共有列
- ▶ ビューおよびフォルダの拡張された実行時カスタマイズ
- ▶ 式言語に対する追加機能
- ▶ LotusScript、COM、および OLE クラスに対する追加機能
- ▶ Java および CORBA クラスに対する追加機能

## 4.1 右クリック・メニューのビュー・アクション

ビューおよびフォルダの右クリック・メニューにアプリケーション開発者がアクションを追加できるようになりました。右クリックの目的は、ショートカット、つまりユーザーが頻繁に使用する機能をすぐに実行する方法を提供することです。右クリックは、技術的にはアクション・バーのボタンをクリックするのと同じです。ビューの選択した文書に対して操作が行われます。ユーザー・インターフェースの観点では、右クリックの利点は、マウスを少し動かすだけで、文書を選択してその文書に対してアクションを実行できることです。右クリックは、ビューに表示されている情報に基づいて、単一の文書に対しアクションを実行する場合に便利です。例えば、返信または転送、頻繁に使用するフォルダへのファイルの追加、削除、後で注意を促すための強調表示などです。右クリック・メニューへのアクションの追加は、アクション・バーおよびアクション・メニューへアクションを組み込むことの代わりとして、またはそれと一緒に使用できます。新しいアクションと既存のアクションの両方を右クリック・メニューに追加できます。

ITSO Electronics アプリケーションでは、「Sales Activity By Customer」ビューに「Create New Sales Activity」アクション・ボタンがあります。アプリケーションを拡張しユーザーの生産性を向上させるには、図 4-1 に示されているように、アクションの「表示」オプションを更新し「右クリックメニューに含む」を選択します。

共有アクション	
名前	Create New Sales Activity
ラベル	
対象フレーム	
種類	ボタン
表示	<input checked="" type="checkbox"/> アクションバーにアクションを含む <input type="checkbox"/> アクションバーにアイコンのみを表示 <input type="checkbox"/> アクションボタンを右に揃える <input checked="" type="checkbox"/> アクションメニューにアクションを含む <input checked="" type="checkbox"/> 右クリックメニューに含む
アイコン	ローケーション 左
アイコン	なし • Notes • カスタム

図 4-1 「右クリックメニューに含む」オプション

次にビューを右クリックすると、79 ページの図 4-2 に示されているように、「Create New Sales Activity」アクションが右クリック・メニューに含まれています。また、その他のビュー・アクションも右クリック・メニューに追加されました。

**注:** 右クリック・メニューは、身体障害があるユーザーにとっては使用しやすくありません。右クリック・メニューで使用できるアクションは、アクション・メニューでもすべて使用できるようにする必要があります。

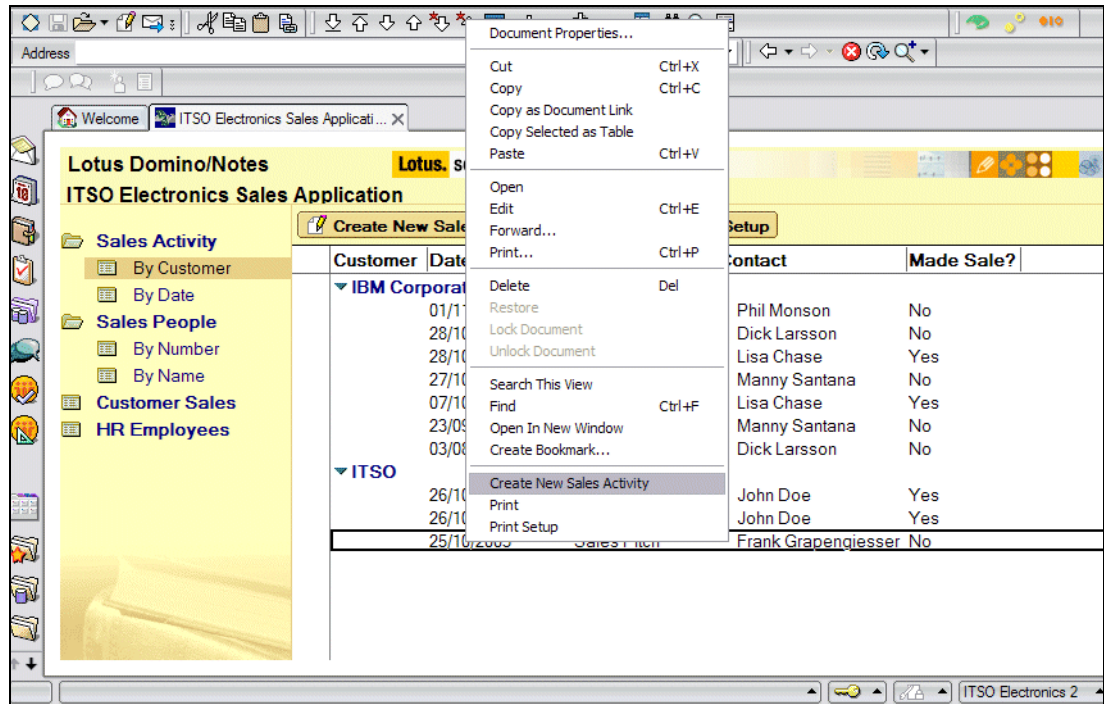


図4-2 右クリック・メニューに組み込まれたアクション

## 4.2 自動保存のフォームのプロパティ

自動保存は Lotus Notes クライアントの機能で、この機能により、ユーザーが指定した定期的間隔で Lotus Notes 文書がローカル・データベースに自動的に保存されます。ソフトウェアの障害、ワークステーションの障害、または電源異常が発生した場合、ユーザーは破損または停電の前に行った作業を回復できます。熱心なコンサルタントとして、ITSO Electronics 社には、この後のセクションの指示に従い、新しい自動保存機能を使用して Lotus Domino アプリケーションを保護するよう提言しました。

### 4.2.1 Lotus Notes クライアントでの自動保存の使用可能化

Lotus Notes 7 をインストールした時点、つまりデフォルトでは、自動保存は有効になっていません。自動保存を有効にするには、Lotus Notes クライアントのユーザープリファレンスでこのオプションを選択する必要があります。

1. 「ファイル」→「プリファレンス」→「ユーザー」を選択します。
2. 80 ページの図 4-3 に示されているように、「基本」パネルの「起動時オプション」で「常に自動保存 15 分」を有効にします。
3. (オプション) 自動保存の間隔を増加または減少させます。
4. 「OK」をクリックします。

**注:** デフォルトの自動保存の時間間隔は 15 分です。この時間間隔は 1 から 999 分の範囲で設定できます。

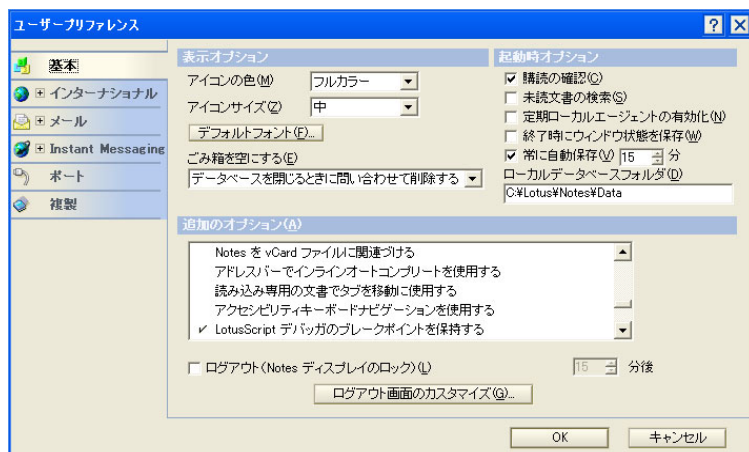


図 4-3 「ユーザープリファレンス」での自動保存の使用可能化

「ユーザープリファレンス」ウィンドウの自動保存オプションは、表 4-1 に示されている、Lotus Notes クライアントの NOTES.INI ファイルの 3 つの変数に対応します。

表 4-1 NOTES.INI の変数

NOTES.INI の変数	ユーザープリファレンス設定
AUTO_SAVE_ENABLE=n	n=0 の場合、自動保存は無効になります。
AUTO_SAVE_INTERVAL=<mins>	<mins> は、自動保存の時間間隔を設定します。
AUTO_SAVE_USER,<Abbreviated User Name>=<Database name relative to data directory>	自動保存データベースとの命名の競合を解決します。

## 4.2.2 フォームでの自動保存の使用可能化

Lotus Domino Designer 7 は、Lotus Notes アプリケーションのフォームに自動保存機能をサポートできる機能を追加することにより、新しい自動保存機能を活用します。ユーザーが、前のセクションで示したように Lotus Notes のプリファレンスで自動保存をアクティブにすると、自動保存が有効になったフォームを使用してユーザーが作成または編集した文書は、特殊なローカルの暗号化されたデータベースに保存されます。フォームの自動保存をアクティブにするには、81 ページの図 4-4 に示されているように、Lotus Domino Designer でフォームを開いて、「フォーム」プロパティ・ウィンドウで「自動保存を許可する」オプションを有効にします。Lotus Notes クライアントで自動保存が有効になると、ステータス・バーにどの時点で文書が自動保存されるかが示されます。

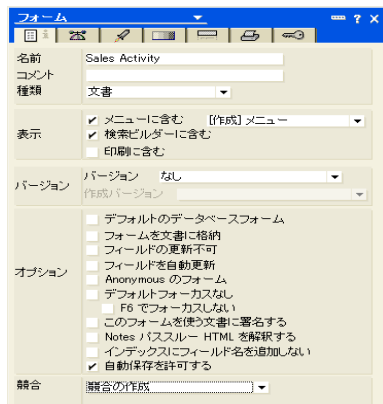


図4-4 「フォーム」プロパティ・ウィンドウの「自動保存を許可する」オプション

アプリケーション開発者は、アプリケーション・フォームの自動保存を有効にした場合、そのアプリケーションで自動保存が適切に動作するか確認する必要があります。多くの Lotus Notes/Domino アプリケーションには、Save イベントまたは Postsave イベントで項目を作成する、またはロード時に妥当性検証を実行するコードがあります。妥当性検証およびフォームのイベント・コードは文書の自動保存時には実行されないため、自動保存された文書をリカバリー後にロードできない場合があります。文書をリカバリー後に適切にロードできるようにするために、Queryopen イベントにコードを追加して、これらの特定の項目のデフォルト値を設定する必要があります。

\$AutosaveRecovered アイテムを使用すると、リカバリー対象の文書を処理できます。  
\$AutoSaveRecovered は、PostOpen イベント発生時までリカバリー対象の文書に対し有効で、このイベント後に削除されます。例えば、「ステータス」フィールドが「ドラフト」でない場合に編集モードで文書が開かれないようにする Queryopen コードがフォームに含まれる場合があります。ユーザーがステータスを「発行」に変更し、自動保存が実行され、その後ユーザーのコンピューターが破損した場合、ステータスに不適切な値があるように思われる場合でも、ユーザーは編集モードでその文書を開くことが可能である必要があります。次のコードをご覧ください。

```
Sub Queryopen(Source As Notesuidocument, Mode As Integer, Isnewdoc As Variant, Continue As Variant)
    Dim doc As NotesDocument
    If Mode And Not Isnewdoc Then
        Set doc = Source.Document
        If doc.Status(0) <> "Draft" Then
            Continue = False
            MessageBox "Only Draft documents may be edited.", _
                MB_ICONSTOP, "Edit Error"
        End If
    End If
End Sub
```

代わりに以下を記述します。

```
If doc.Status(0) <> "Draft" And Not doc.HasItem("$AutosaveRecovered ") Then
```

\$DontAutosave アイテムを使用して個々の文書に対して自動保存を無効にでき、その特定の文書では自動保存が実行されなくなります。\$DontAutosave を削除すると、再び文書を自動保存できます。



### 4.2.3 自動保存機能を使用して保存された文書のリカバリー

自動保存された文書は、起動直後または起動後の任意の時点でリカバリーできます。障害後または電源異常後の起動時に、ユーザー認証が行われた後、図 4-5 に示すように、未保存の作業をリカバリーするかどうか尋ねるプロンプトがユーザーに表示されます。

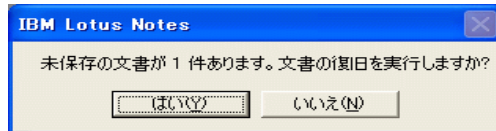


図 4-5 未保存の作業に関するプロンプト

ユーザーが「はい」をクリックすると、図 4-6 に示すように、「未保存文書の復旧」ウィンドウが開き、リカバリー可能なすべての文書がリスト表示されます。

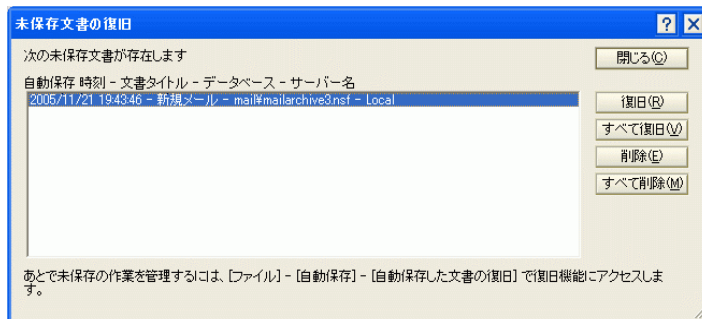


図 4-6 「未保存文書の復旧」ウィンドウ

表 4-2 は、図 4-6 に示されているウィンドウ内の 4 つのボタンそれぞれの機能の説明です。

表 4-2 「未保存文書の復旧」のオプション

オプション	説明
「復旧」	選択した文書をリカバリーし、リストからその文書を削除します。
「すべて復旧」	文書ごとにプロンプトを表示せずに、すべての文書をリカバリーします。
「削除」	自動保存データベースから選択した文書を削除します。
「すべて削除」	自動保存データベースからすべての文書を削除します。

ユーザーが「いいえ」をクリックした場合、「ファイル」→「自動保存」→「自動保存した文書の復旧」を選択することで、自動保存されている文書を後でリカバリーできます。この場合、先に説明したように「未保存文書の復旧」ウィンドウが開き、ユーザーは文書をリカバリーまたは削除できます。

文書のデータベースのレプリカが見つからない場合、文書はリカバリーできません。文書は自動保存データベースにそのままの形で残るので、後で再度リカバリーを試行できます。前回の自動保存以降に文書が変更された場合、プロンプトがユーザーに表示され、自動保存された文書を使用して上書きするか、または変更された文書を保持するかを尋ねられます。自動保存された文書のリカバリーをユーザーが選択すると、Lotus Notes でその文書が編集モードで開きます。リカバリー後、自動保存は自動的に文書をユーザーに代わって保存しません。自動保存データベースから文書がリカバリーされると、その文書は自動保存データ



ベースから削除されます。これは自動保存データベースのサイズを管理しやすいサイズに保持するためです。

#### 4.2.4 手動での自動保存の使用

**注:** このオプションは、フォームの自動保存を有効にした場合に限り使用できます。

ユーザーは、時には文書を編集の途中で即時に保存する必要があります。ユーザーは、「ファイル」→「自動保存」→「すぐに自動保存」を選択することで、現在の文書を自動保存データベースに保存することができます。

#### 4.2.5 自動保存データベースへのアクセス

自動保存データベースは、ユーザーのデータ・ディレクトリに配置されます。自動保存データベースは、作成されると、メール・ファイルと同様の命名規則を使用します。データベース名の先頭には \_ が付加されます。シナリオのアプリケーションでは、Manny Santana というユーザーが as\_msantana.nsf という自動保存データベース名を所持しています。この名前のデータベースが既に存在している場合は、「1」などを名前に追加して、自動保存データベース名が固有の名前になるようにします。

自動保存データベースの命名規則では、John Doe、Jane Doe、James Doe などのユーザー名の場合に競合が発生します。正しい自動保存データベースを現在のユーザーが使用することを確保するため、データベースが作成された後、次の項目がユーザーの NOTES.INI ファイルに追加されます。

AUTO\_SAVE\_USER,<Abbreviated User Name>=<Database name relative to data directory>

サンプルのアプリケーションでは、次の項目が Manny Santana の NOTES.INI ファイルに追加されます。

AUTO\_SAVE\_USER,Manny Santana/ITOSElectronics=as\_msantana.nsf

ユーザーが Lotus Notes を起動すると、NOTES.INI ファイルがチェックされ、自動保存データベースに対応する項目が存在するかどうかを確認されます。項目が存在する場合、その項目で指定されている名前のデータベースが自動保存データベースとして使用されます。指定されている自動保存データベースへのアクセス権がユーザーにない場合、ユーザーはエラーを受け取り、自動保存は無効になります。指定されたデータベースが存在しない場合、項目で指定されている名前を使用して新しいデータベースが作成されます。NOTES.INI ファイルにユーザーの名前の項目がない場合、NOTES.INI ファイルに項目が作成され、先に説明した命名規則を使用してデータベースが作成されます。

### 4.3 共有列

Lotus Domino Designer 7 ソフトウェアは、ビューおよびフォルダで使用する共有列を作成する機能を備えています。共有列は独自の設計要素を持ちますが、いったんビューに挿入すると、ビューの一部となります。共有列設計要素は、ビューまたはフォルダ内の非共有列と同じ列プロパティおよび外観を持ちます。さらに、共有列設計要素には共有列固有のプロパティがあります。84 ページの図 4-7 に示されている「共有列」プロパティ・ウィンドウで、共有列の名前 (必須)、別名 (オプション)、およびコメント (オプション) を指定します。

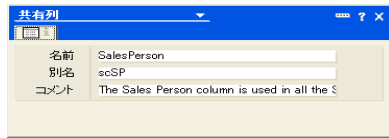


図4-7 「共有列」プロパティ・ウィンドウ

共有列を含むビューまたはフォルダは、Web および Lotus Notes からのアクセスに対して後方互換性があります。18 ページの 2.3、「Query ビュー」で説明したように、共有列は DB2 Query ビューで使用できます。共有列は、共有フィールドと同じく、マルチリンガル対応の要素ではありません。

共有列を使用すると、同じデータベースの複数のビューおよびフォルダに挿入する共通の列を作成できるので、同じ列を複数回作成する必要がなくなり、開発者の生産性が向上します。この新機能により、1 カ所でのみ変更を行うだけで、同じ列を使用しているすべてのビューにその変更を反映させることができます。

### 4.3.1 共有列の作成

新しい共有列を作成するには、「共有コード」の下で「列」を選択します。「新規共有列」をクリックするか、「作成」→「設計」→「共有列」を選択します。既存の共有列設計要素をコピーして貼り付けることもできます。

ビューにある既存の列の設計に基づいて新しい共有列を作成するには、ビューを開き、列を選択し、「作成」→「共有列としてコピー」を選択します。または、図 4-8 に示されているように、列を選択し、右クリックしてメニューにアクセスし、「共有列としてコピー」を選択します。

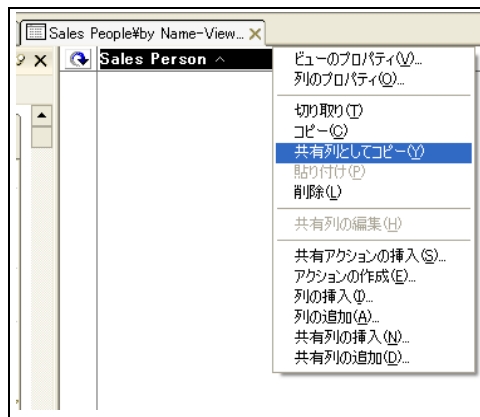


図4-8 右クリック・メニューを使用して新しい共有列を作成

84 ページの図 4-9 に示されているように、新しい共有列の名前を指定するためのウィンドウが開きます。このアクションでは、新しい共有列が作成され、既存のビューの列はそのままです。既存の列は新規の共有列からは完全に独立しています。したがって、新しい共有列から設計の変更を継承する場合、既存のビューの列を削除し、新しい共有列をそのビューに挿入して、その非共有列を置き換える必要があります。

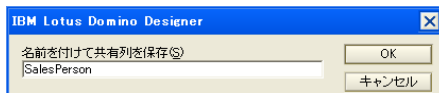


図4-9 新しい共有列への名前の指定

ITSO Electronics アプリケーションでは、「Sales People By Name」ビューに Sales Person 列があります。同じ情報を「Sales People By Number」ビューにも表示する必要があります。このため、Sales Person 列を使用して、同じ Sales Person という名前の新しい共有列を先に説明した方法で作成します。共有列の影響を受けるビューには、Lotus Notes クライアントの以前のリリースでアクセスできます。

### 4.3.2 共有列の使用

ビューで共有列を使用するには、ビューを開いて「作成」→「共有列の挿入」または「作成」→「共有列の追加」を選択します。図 4-10 に示されているように、表示されるリストから共有列を選択します。フォント、列の位置合わせなど、ビューでの共有列の表示プロパティをオーバーライドする場合は、「式のみを使用」オプションを選択します。表示プロパティをロックする場合は、このオプションは選択しません。挿入 / 追加された列のプログラム名（「プロパティ」→「詳細」タブ）は、「式のみを使用」の設定に関係なく、共有列からは継承されません。したがって、この値はいつでも変更できます。ただし、列の値（式）は変更できません。

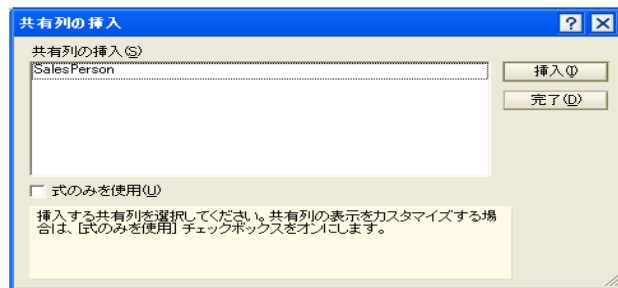


図 4-10 「共有列の挿入」 ウィンドウ

ITSO Electronics アプリケーションでは、「Sales People By Number」ビューで Sales Person 共有列を使用するために、Lotus Domino Designer でビューを開き、先に説明した手順を使用して共有列を挿入します。図 4-11 は、共有列を挿入して更新されたビューです。

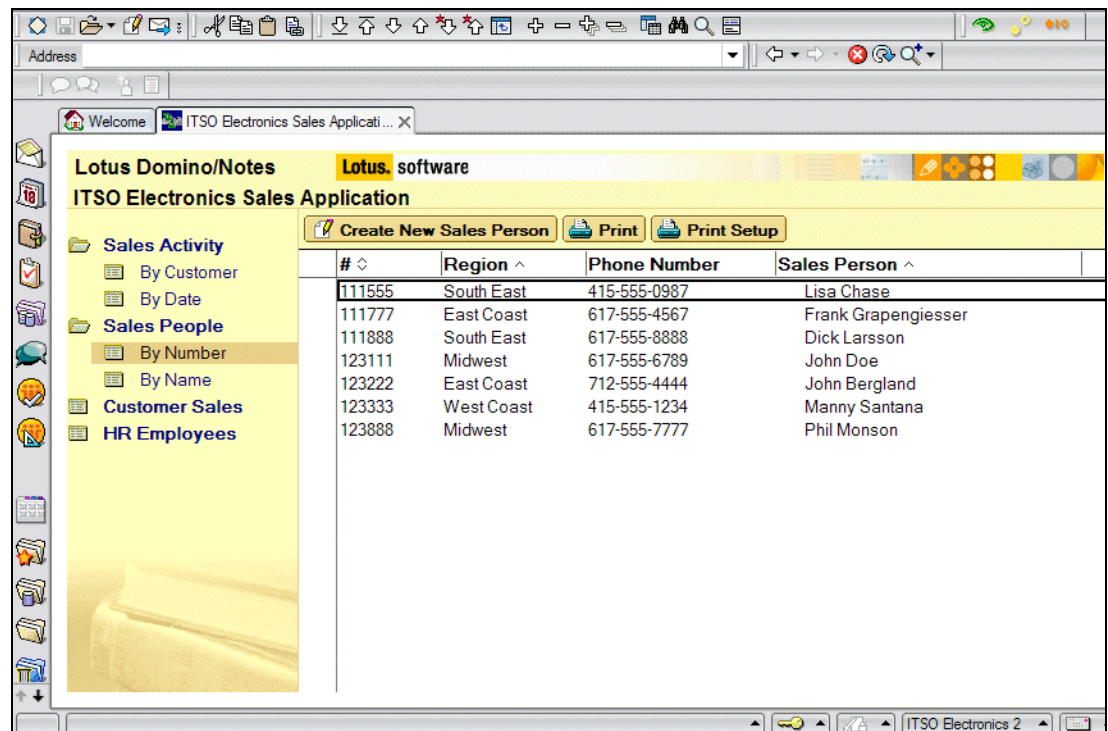


図4-11 共有列の Sales Person を挿入して更新された「Sales People By Number」ビュー

共有列の設計を編集するには、「共有コード」→「列」設計リストを開き、このリスト内から共有列を見つけます。ビューで共有列を選択し、「設計」→「共有列の編集」を選択することでも、共有列を編集できます。変更を保存する前に、共有列の定義に対して行った変更を、その共有列を使用しているビューに自動的に反映させるかどうか尋ねる図 4-12 のプロンプトがユーザーに表示されます。これらの更新されたビューは、現在のユーザー ID を使用して再保存されます。また「式のみを使用」が選択されている場合は、式のみ更新されます。

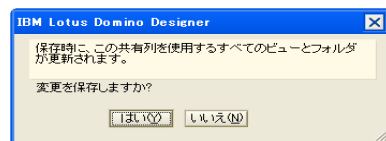


図4-12 共有列の保存

共有列を削除するには、図 4-13 に示されているように、共有列設計要素を選択し、「編集」→「削除」を選択するか、右クリックして「削除」を選択します。共有列を削除すると、その共有列を含むすべてのビューで、その共有列が非共有列に変更されます。

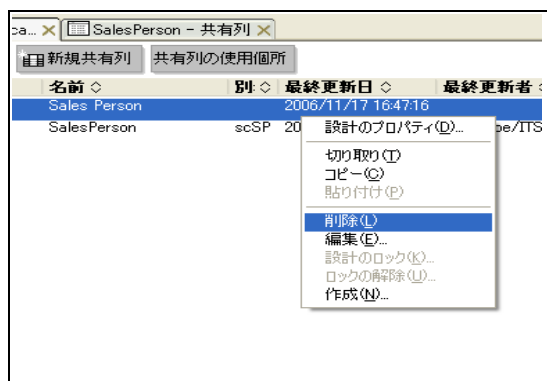


図 4-13 共有列の削除

「列」設計リストで「共有列の使用箇所」アクション・ボタンをクリックすると、選択した共有列をデータベース内のどのビューおよびフォルダが使用しているかを確認できます。ITSO Electronics アプリケーションでは、Lotus Domino Designer でデータベースを開き、「共有コード」→「列」を選択し、リストから「SalesPerson」共有列を選択し、「共有列の使用箇所」アクション・ボタンをクリックすると、図 4-14 に示されている情報が表示されます。

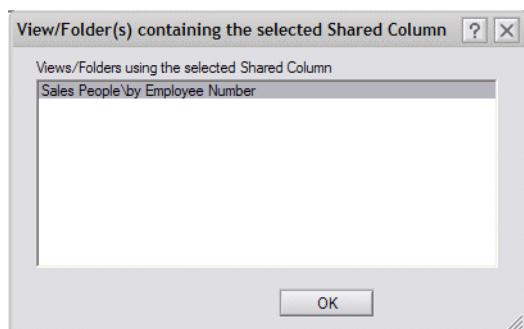


図 4-14 「共有列の使用箇所」メッセージ・ウィンドウ

## 4.4 ビューおよびフォルダの拡張された実行時カスタマイズ

開発者は、「ユーザー設定」と呼ばれる列プロパティを使用することで、データベース内のプロファイル文書を基に列の式を設計できます。Lotus Domino Designer ソフトウェアの前のバージョンでは、この機能の使用は、列のカラー・コーディングに制限されていました。Lotus Domino Designer 7 では、ビューおよびフォルダの列においてプロファイル文書のサポートが拡張され、アプリケーションでより多くの実行時オプションを使用できます。このことを示す良い例が、Lotus Notes 7 のメールテンプレートです。受信ボックス・フォルダには、メールの重要度を示す列と、カラー・コーディングを制御する非表示の列があります。これらの両方の列の内容および動作は、ユーザーのメールプリファレンスにより定義されます。88 ページの図 4-15 を参照してください。

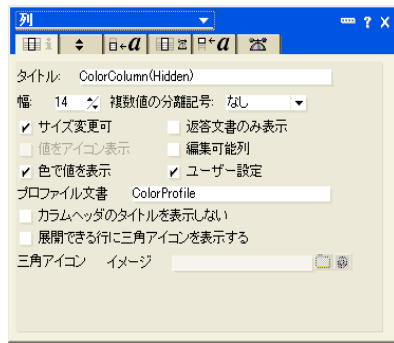


図4-15 ユーザー設定の「列」オプション

その月に重点を置く地域を示しさらに文書が現在のユーザーを参照しているかどうかを示す列を追加して、ITSO Electronics アプリケーションを拡張します。地域がその月の重点を置く地域の場合、スター型のアイコンが表示されます。青色の矢印は、文書が現在のユーザーを参照していることを示します。これら以外は、列には何も表示されません。ユーザー・プロファイル列を追加する前に、アプリケーションに SalesProfile という名前の既存のプロファイル文書があることを前提とします。ユーザー・プロファイル列で利用できる「共有」プロファイル文書は1つのみです。したがって、プロファイル文書を作成するときに、オプションの固有キー・パラメーターは使用しません。プロファイル文書の詳細については、IBM developerWorks の記事「*The hidden power of Profile documents*」を参照してください。以下で入手できます。

[http://www.ibm.com/developerworks/lotus/library/ls-Profile\\_documents/](http://www.ibm.com/developerworks/lotus/library/ls-Profile_documents/)

ITSO アプリケーションの「Sales Person By Name」ビューに新しいユーザー・プロファイル列を追加するには、次の手順に従います。

1. Lotus Domino Designer でビューを開き、「作成」→「列の挿入」をクリックして新しい列をビューに挿入します。
2. 89 ページの図 4-16 に示されているように、「列」プロパティ・ウィンドウを開き、次の値を確実に定義します。
  - 「値をアイコン表示」を選択します。これは、ユーザー・プロファイル列がその月の重点地域を表すアイコン、または現在のユーザーを参照していることを示すインディケーターを表示するためです。
  - 「ユーザー設定」を選択します。
  - データベースの既存のプロファイル文書 SalesProfile を、「プロファイル文書」フィールドに定義します。
  - 「列の値」の式として @UserName を使用します。

**注：**列の値の式は、@Username、@Created、@Random など非定数の式にする必要があります。非定数の式は、値が評価されないため、列の値の式として使用できません。“”、特殊なテキスト関数 (例えば @DocNumber) などの定数式を使用すると、ユーザー・プロファイル列は正しく表示されず、「競合文書」メッセージがビューの各文書に対して表示されます。

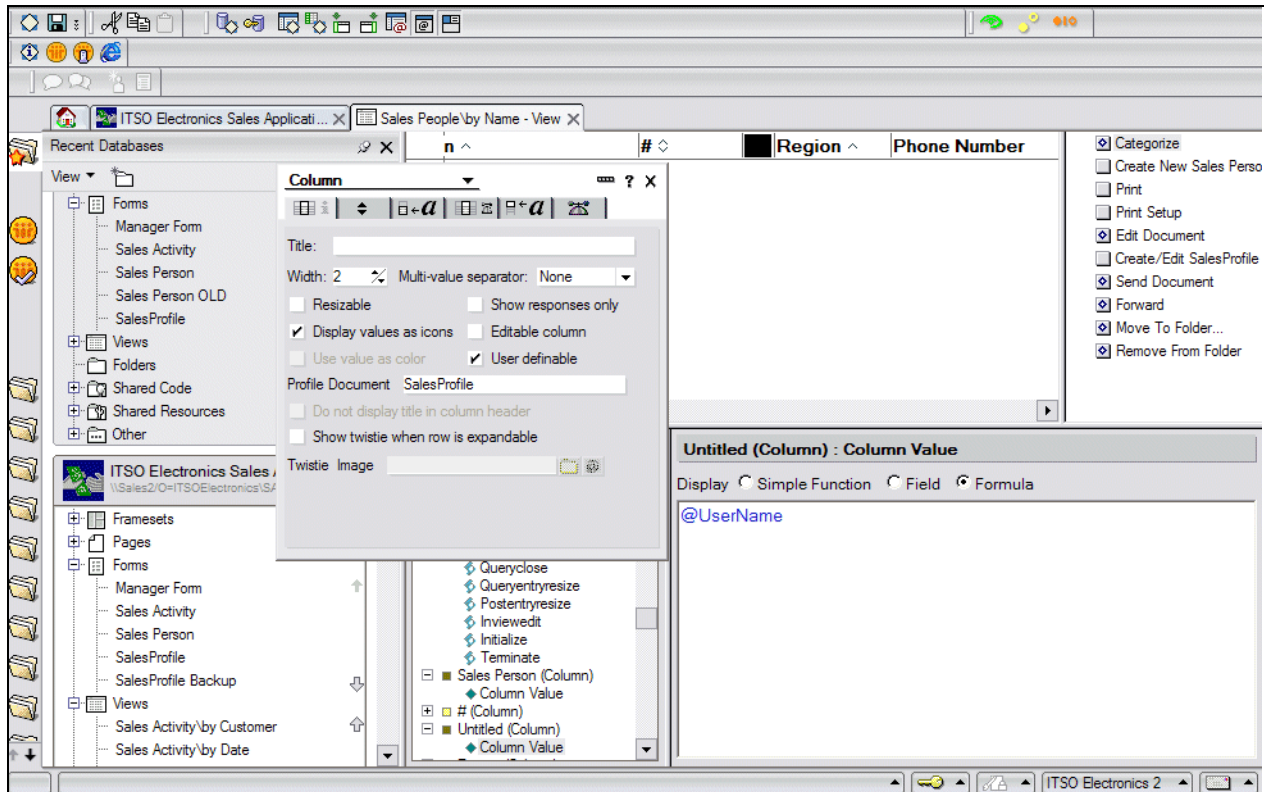


図4-16 ユーザー・プロフィール列の定義

- 図4-17に示されているように、「詳細」タブの「名前」フィールドにはプロフィール文書にあるフィールド名が含まれます。この例の場合は `$ToRegion` です。このプロフィール文書のフィールドに式が格納されていて、ビューの行が表示される時に評価されます。つまり、`$ToRegion` フィールドのフィールド値は式コードで、その出力は列に表示される値です。`$ToRegion` の値の式は式で、その出力はその式です。

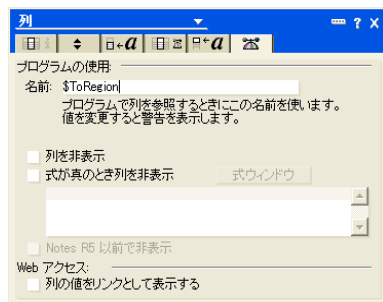


図4-17 「列」プロパティの「詳細」タブ



プロファイル文書を編集するエンド・ユーザーは、多くの場合、この式を記述できません。式を含む計算フィールド(この例では \$ToRegion)は、エンド・ユーザーに式の計算結果を示すために使用されます。この計算フィールドの式で、フィールドに格納する式を生成する必要があります。図 4-18 には、SalesProfile フォームの設計と、値の式を生成する \$ToRegion フィールドのフィールド値が示されています。

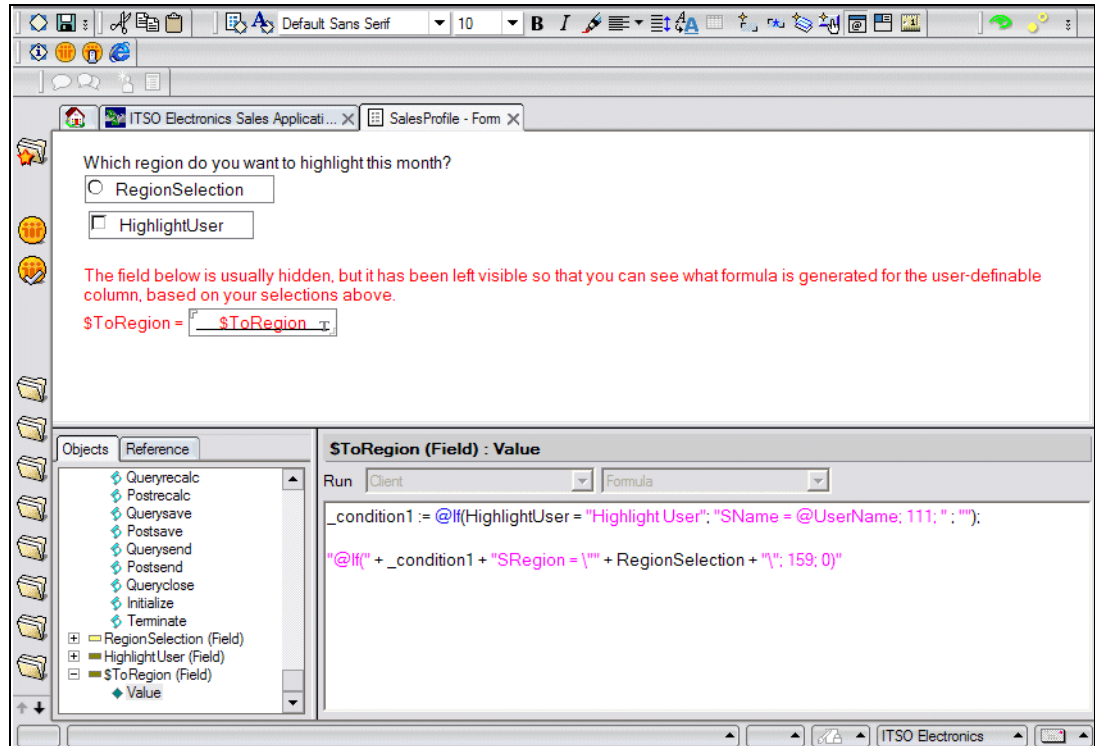


図 4-18 Lotus Domino Designer 内の SalesProfile フォーム



3. 図 4-19 は、SalesProfile プロファイル文書を示します。RegionSelection フィールドと HighlightUser フィールドに値を入力します。\$ToRegion フィールドの式により、ユーザー設定の列で使用される新しい式が計算されます。例えば、地域の選択が「**South East**」で「**Highlight User**」が選択されている場合、\$ToRegion の値は次のようになります。

@If(SName = @Username; 111; SRegion = "South East"; 159; 0)

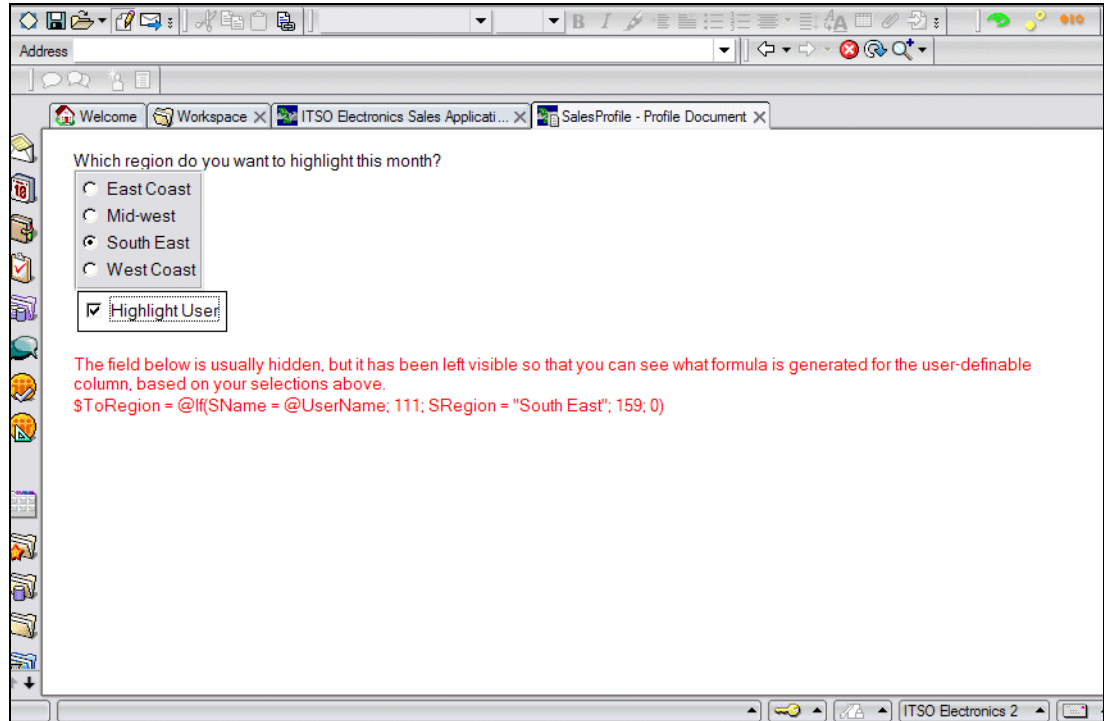


図 4-19 SalesProfile プロファイル文書

「ユーザー設定」オプションは、「色で値を表示」を選択した場合だけに限らず常に使用できます。このため、アイコン用のプロファイル文書列を使用したり、その他のタイプの列を使用したりできます。プロファイル文書を同じビュー内の複数の列に対して指定する場合、すべての列に対して同じプロファイル文書を指定する必要があります。このようにしないと、ビューを保存した時点で図 4-20 に示されているようなエラー・メッセージが表示されます。

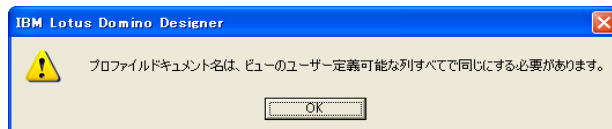


図 4-20 ユーザー・プロファイル列に関するエラー・メッセージ

ビューが保存された後、図 4-21 に示されているように、ユーザー・プロフィール列が「Sales People By Name」ビューに表示されます。

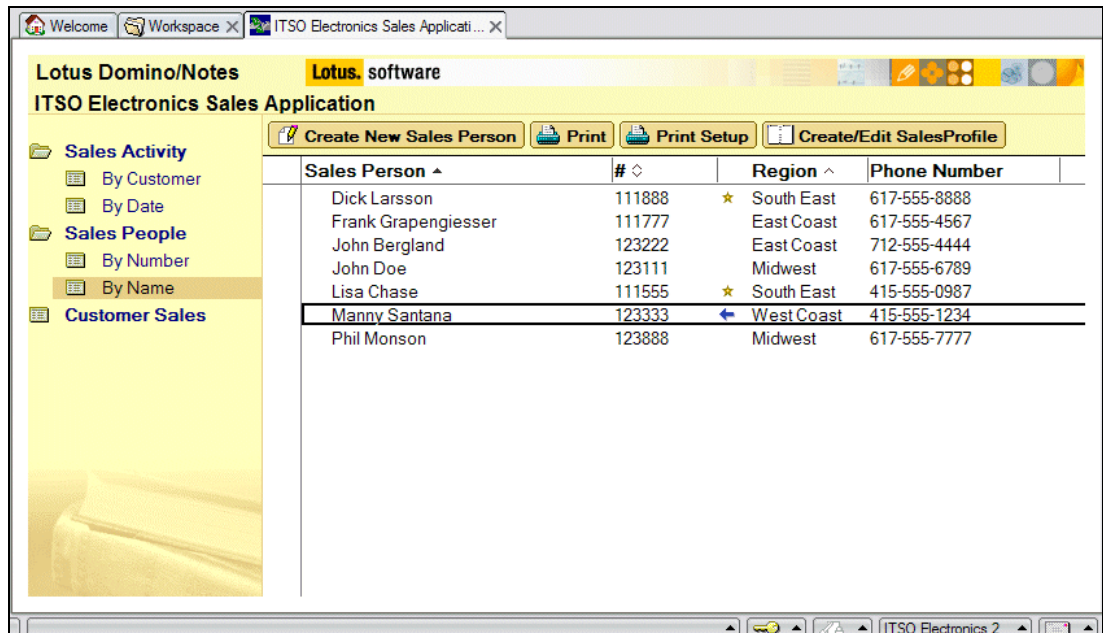


図4-21 アイコンを表示しているビュー内のユーザー・プロフィール列

**注：**計算フィールドの式では、引用符およびバックスラッシュ (¥ マーク) に注意してください。多くのマクロ式は引用符を、一部のマクロ式はバックスラッシュを含むため、プロフィール・フィールドの計算値式では、通常、引用符の内側に引用符 (またはバックスラッシュ) を含める必要があります、これが混乱の一般的な原因となっています。このため、ここで行ったように、開発中に限り計算フィールドを表示して、フィールドを編集したり、結果の式を確認したり、式の構文が正しいかどうか確認したりできるようにすることが良いことなのです。また、データを他のフィールドから式に挿入する場合は、そのデータに引用符またはバックスラッシュが含まれていないか注意し、  
**@ReplaceSubstring(fieldname; "¥¥": "¥"; "¥¥¥¥"; "¥¥¥¥")** を使用して、それらの引用符またはバックスラッシュを適切なストリング・リテラルに変換してください。

## 4.5 式言語に対する追加機能

Lotus Domino Designer 7 では、Lotus Notes の式言語が更新されました。新しく導入された @ 関数は、ポリシーのセキュリティ機能および ECL ロックをサポートしており、さらには IBM DB2 および IBM Workplace Client Technology™ の統合をサポートしています。選択した文書があるすべてのフォルダを検索して突き止める @ コマンドも新しく追加されました。

新しいコマンドおよび関数には、以下のものがあります。

- ▶ **@AdminECLIsLocked:** この新しい関数は、名前とアドレス・ブックの Administration ECL の現在の状態をチェックします。

- ▶ **@DB2Schema:** この新しい関数は、データベースの名前をテキスト・ストリングとして与えると、そのデータベースが db2nsf データベースの場合は、そのデータベースの DB2 スキーマを含むテキスト・ストリングを戻し、db2nsf データベースでない場合は空ストリングを戻します。この関数は、クエリー・ビューの SQL クエリー式で使用するると便利です。18 ページの 2.3、「Query ビュー」で説明しているように、クエリー・ビューでは、式を指定して必要なレコードを取得する SQL を生成する必要があります。SQL では、スキーマ名を含む完全修飾のテーブル名を使用する必要があります。設計を他のサーバーおよびアプリケーションに移植できるようにするために、スキーマ名をハードコーディングするのではなく、**@DB2Schema** 関数を使用することをお勧めします。
- ▶ **@IsDB2:** この新しい関数は、サーバーとファイル名、またはサーバーとレプリカ ID を与えると、指定したデータベースが DB2 データベースかどうかを示します。
- ▶ **@IsEmbeddedInsideWCT:** この新しい関数は、現在の Lotus Notes セッションのいずれかの部分が Workplace クライアント内に組み込まれているかどうかを示します。
- ▶ **@PolicyIsFieldLocked:** この新しい関数は、管理ポリシーによりフィールドがロックされているかどうか、変更できないかどうかを示します。
- ▶ **@Command([DiscoverFolders]):** この新しいコマンドは、「現在の文書を含むフォルダ」ウィンドウを表示します。

注 :**@IfError** 関数は Lotus Domino Designer 7.0 では廃止されましたが、後方互換性に関してはサポートされています。

これらの新しい関数およびコマンドの詳細については、「Lotus Domino Designer 7 Help」を参照してください。以下で入手できます。

[http://www.lotus.com/ldd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/ldd/doc/domino_notes/7.0/help7_designer.nsf)

## 4.6 LotusScript、COM、および OLE クラスに対する追加機能

LotusScript は更新され機能が向上されました。以下に更新箇所を列挙します。

- ▶ 新しいコード・プロファイル機能のサポート。96 ページの 5.1、「ITSO Electronics アプリケーションにおけるコード・プロファイル」の説明を参照してください。
- ▶ 文書収集機能の向上
- ▶ XML サポートの拡張
- ▶ Workplace Client Technology との統合のサポート

Lotus Domino Designer 7 の最大の特長は、LotusScript に追加された新しいプロパティです。NotesUIDocument.ModifiedSinceSaved プロパティにより、前回の保存以降、編集モードで文書に対し行われた変更が検出されます。

Lotus Domino Designer 7 の LotusScript の更新に関する詳細については、「Lotus Domino Designer 7 Help」を参照してください。以下で入手できます。

[http://www.lotus.com/ldd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/ldd/doc/domino_notes/7.0/help7_designer.nsf)

## 4.7 Java および CORBA クラスに対する追加機能

Lotus Domino Designer 7 ソフトウェアは、新しいバージョンの Java、Java Virtual Machine (JVM™) 1.4.2、および新しいバージョンの Extensible Markup Language (XML) パーサーをサポートしています。新しいチェック・ボックス機能および Java デバッグ設定を使用すること


で、Lotus Notes クライアントでリモート・デバッグを行うための Java エージェント、Web サービス、およびスクリプト・ライブラリを設定できます。

Lotus Domino Designer 7 では、次の機能が追加されています。

- ▶ Java クラスおよび CORBA クラス用の Sun™ コア Java 文書
- ▶ Sun Java™ 2 Platform、Technology Edition、V1.4.2 のサポート
- ▶ 新しい AdministrationProcess.approveHostedOrgStorageDeletion メソッド
- ▶ 新しい Database.getModifiedDocuments メソッド
- ▶ 新しい DocumentCollection.UntilTime プロパティ
- ▶ Java Platform Debugger Architecture (JPDA) のサポート

Lotus Domino Designer 7 の Java サポートおよび CORBA サポートの詳細については、「Lotus Domino Designer 7 Help」を参照してください。以下で入手できます。

[http://www.lotus.com/ldd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/ldd/doc/domino_notes/7.0/help7_designer.nsf)



## ITSO Electronics アプリケーションの診断およびトラブルシューティング

Lotus Domino Designer 7では、コード・プロファイル、高度な LotusScript デバッグ機能、リモート Java デバッグ機能の追加により拡張された、アプリケーション診断およびトラブルシューティング機能が導入されています。これらのデバッグ・ツールを追加することにより、ご使用の Lotus Domino アプリケーションのパフォーマンスを向上させることができます。ここでは、ITSO Electronics アプリケーションを例として使用し、Lotus Domino Designer デバッグ・ツールキットの以下のコンポーネントを検証します。

- ▶ コード・プロファイル：ITSO Electronics アプリケーションのパフォーマンスの最適化
- ▶ LotusScript デバッグ：ITSO Electronics アプリケーションにおける拡張された LotusScript デバッグ要素の活用
- ▶ Java デバッグ：ITSO Electronics アプリケーションにおけるリモート Java デバッグ

## 5.1 ITSO Electronics アプリケーションにおけるコード・プロファイル

Lotus Domino Designer 7では、コード・プロファイル・ツールである Profiler が追加されたことで、Lotus Notes/Domino アプリケーションのエージェント、Web サービス、スクリプト・ライブラリーにおけるメソッドの実行時間を Lotus Domino 開発者が測定できるようになり、強力なコード最適化が実現しました。Lotus Domino アプリケーション内でメソッドのパフォーマンスを測定できる機能は、開発者がコード内の非効率性を識別する上で役立ちます。さらに、最も実行時間がかかり、そのためパフォーマンスへの影響が最も著しいコードの部分の最適化に集中できるようになります。このセクションでは、ITSO Electronics アプリケーションの効率性を最大にするためのコード・プロファイルの使用方法について説明します。

Profiler は、Lotus Domino データベースなどの Lotus Domino オブジェクトの操作を含め、Java および LotusScript の両方でメソッドを検証および測定します。開発者は、エージェントまたは Web サービスでプロパティ・ウィンドウの「セキュリティ」タブをクリックし、「エージェントをプロファイル」または「Web サービスをプロファイル」を選択して、プロファイル作成を使用可能にすることができます。有効にしたエージェントまたは Web サービスを起動するたびにプロファイル作成が行われます。エージェントのプロファイル作成は、Lotus Notes クライアントおよび Lotus Domino サーバーの両方で実行されます。図 5-1 に示すように、「エージェント」プロパティ・ウィンドウの 2 番目のタブに表示されるプロファイル作成のオプションを有効にすると、次回にエージェントを起動したときに、プロファイル作成が実行されます。

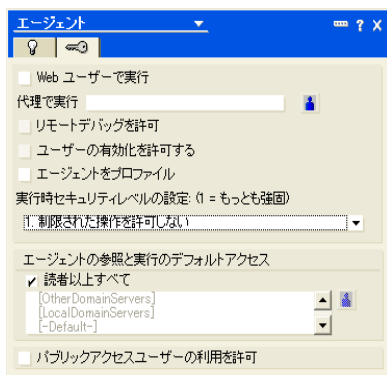


図 5-1 「エージェント」プロパティ

エージェントは、スケジュール・エージェントまたは Web エージェントのどちらとして実行するかにかかわらず、また「アクション」メニューから手動で実行している場合でも、プロファイルできます。プロファイル情報は、エージェントに関連付けられているデータベースのプロファイル文書に保管されます。プロファイル情報を表示するには、Lotus Domino Designer でプロファイルするエージェントを選択してから、「エージェント」→「プロファイル結果の表示」を選択します。各プロファイル文書の最上部に、エージェントの名前とプロファイルが実行された時刻が記録されます。経過時間はエージェントの実行時間の合計を示し、続いて測定時間の合計が示されます。これは通常、表示の便宜上端数を切り捨てた時刻値になるため、やや短い時間になります。例えば、ミリ秒未満の値はゼロとしてプロファイル文書に表示されます。プロファイル・テーブルには、呼び出された各 Lotus Domino Objects メソッドを表示する行があり、5 つの列で構成されています。

- ▶ クラス：「Session」、「Database」、「Document」のように正規化された名前で指定した Lotus Domino Objects クラスの名前。
- ▶ メソッド：「CurrentDatabase」、「AppendItemValue」、「Save」のように正規化された名前で指定した Lotus Domino Objects メソッドまたはプロパティの名前。

- ▶ 操作：プロパティーで実行する、Get または Set のいずれかのタイプの操作。
- ▶ コール：メソッドまたはプロパティーが呼び出された回数。
- ▶ 時間：呼び出しにかかった時間のミリ秒表示。“<” 記号は、所要時間が少ないために計算できないことを表します。

プロファイル・テーブルの情報は、実行の所要時間が最も多かったメソッドから降順でソートされ、表示されています。図 5-2 のプロファイル出力例は、このエージェントで最も時間を消費した部分が UpdateFTIndex メソッドであることを示しています。

ProfileExample Profile				
08/25/2005 01:39:53 PM EDT				
Elapsed time: 41059 msec				
Methods profiled: 6				
Total measured time: 37023 msec				
Class	Method	Operation	Calls	Time
Database	UpdateFTIndex		27	36113
Database	Open		27	910
Database	Title	Get	27	0
Database	FileName	Get	27	0
DbDirectory	GetNextDatabase		26	0
DbDirectory	GetFirstDatabase		1	0

図 5-2 コード・プロファイル結果

エージェントを実際に実行したときのデータを取り込むことで、コード・プロファイルにより開発者に有用な情報が提供され、コード実行におけるボトルネックが識別できます。開発者は、この情報を活用して実行時間が改善されるようにエージェントのロジックを修正でき、その結果ユーザー満足度と生産性が向上します。

## 5.2 ITSO Electronics アプリケーションにおける LotusScript デバッグ

Lotus Domino Designer 7 では、LotusScript デバッグ機能をより直感的に使用可能になり、このデバッグ能力が強化されています。開発者は、デバッグをメニュー・システムから開始できるだけでなく、アイコンで開始および終了できるようになりました。LotusScript デバッグが開始または終了していることを示すメッセージは、メッセージ・ウィンドウではなくステータス・バーに表示されます。このセクションでは、ITSO Electronics アプリケーションのデバッグで、この新しい LotusScript 診断機能をデモンストレーションします。

新しいデバッグ機能を使用して ITSO Electronics アプリケーションで LotusScript をデバッグするには、以下の手順に従います。

1. デバッガーを使用可能にするには、ツールバーの「**LotusScript のデバッグ**」アイコンをクリックします。98 ページの図 5-3 に示すように、ステータス行に「LotusScript のデバッグ開始」というメッセージが表示されます。



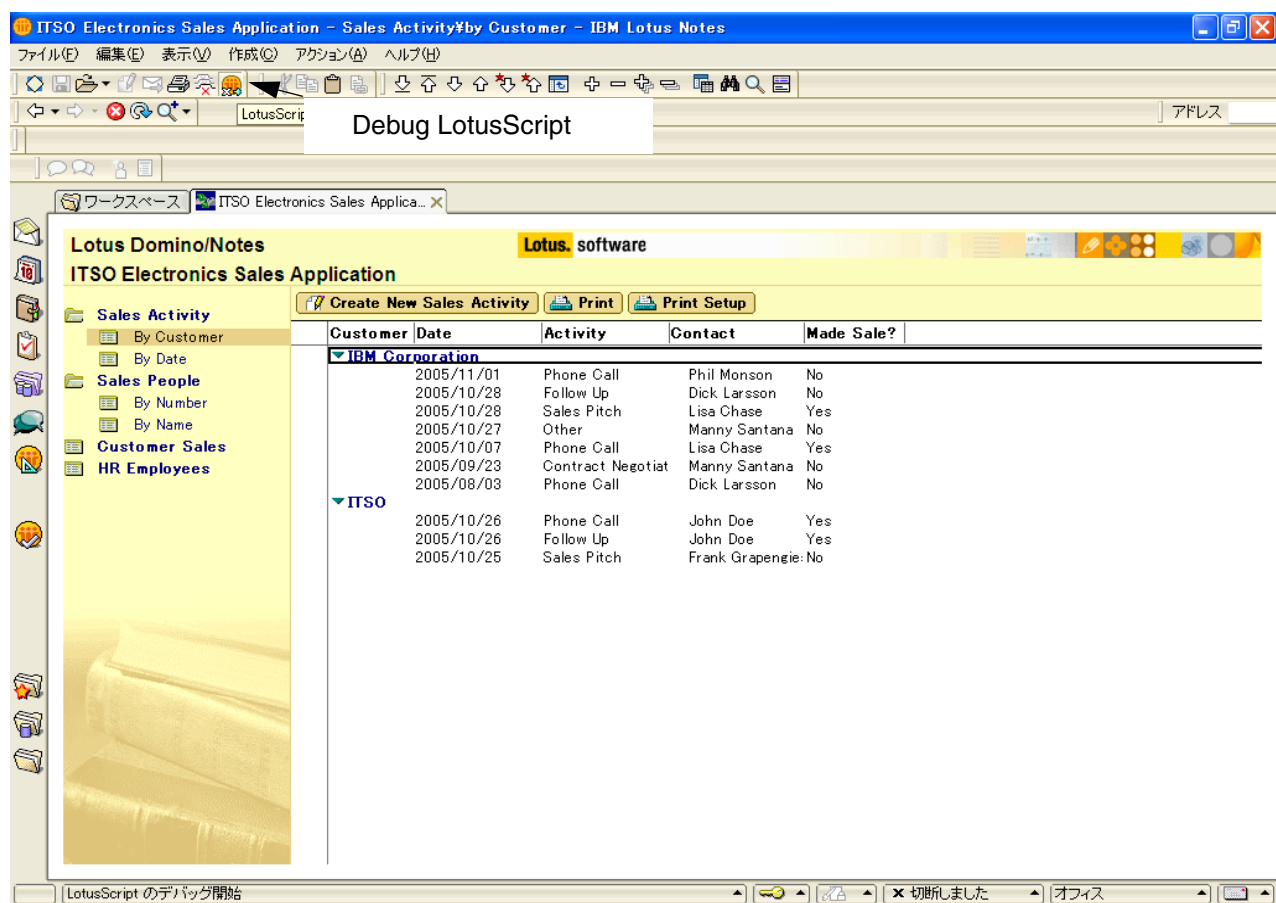


図 5-3 LotusScript デバッガーの開始

2. デバッガーを使用不可にするには、ツールバーの「**LotusScript のデバッグ**」アイコンをもう一度クリックします。99 ページの図 5-4 に示すように、ステータス行に「LotusScript のデバッグ終了」というメッセージが表示されます。



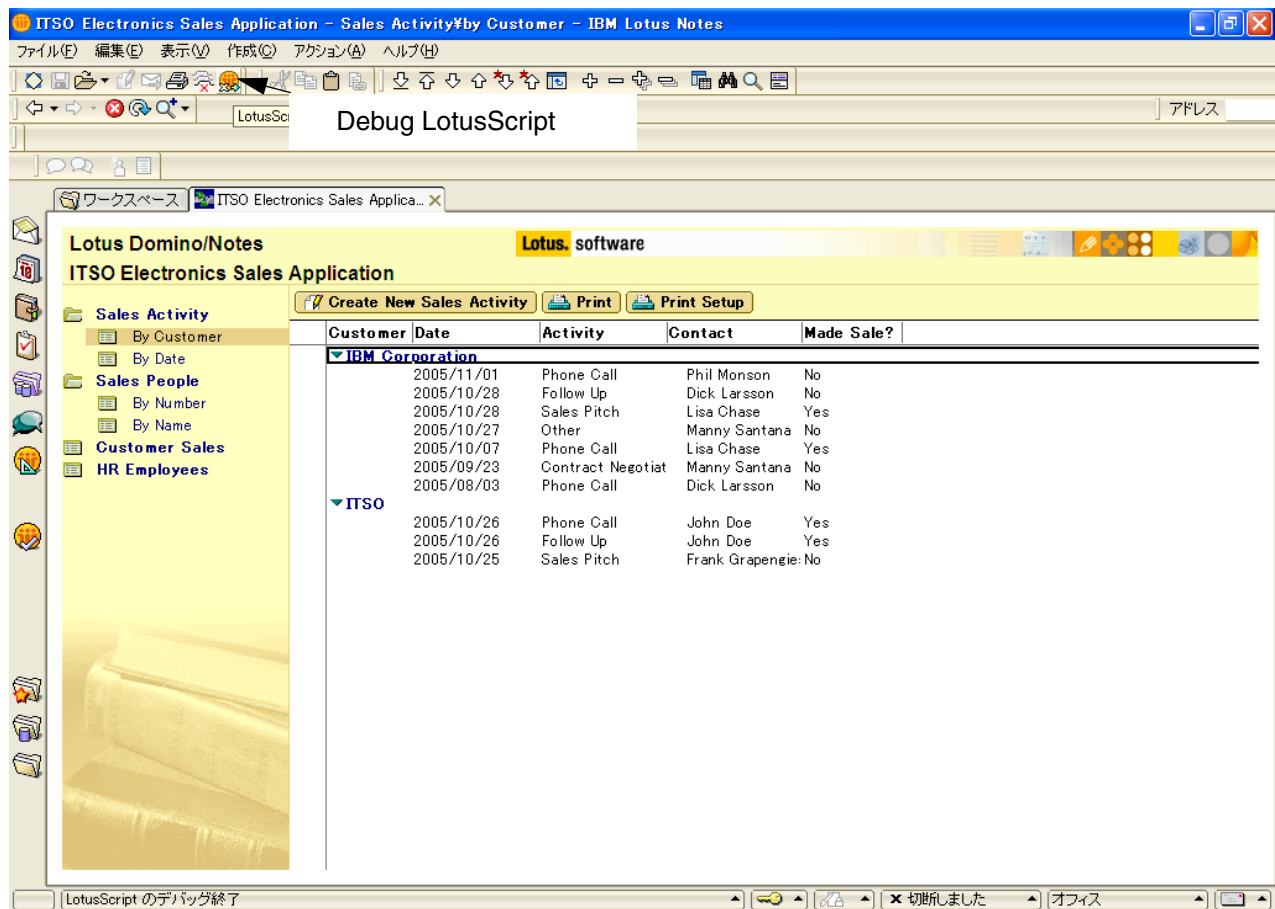


図 5-4 LotusScript デバッガーの停止

## 5.3 ITSO Electronics アプリケーションにおける Java デバッグ

Lotus Domino Designer 7 のデバッグ・ツールキットに追加された、恐らく最も強力な機能は、リモート Java デバッグです。Lotus Domino アプリケーション開発者は、Java リモート・デバッガーを使用することにより、Eclipse テクノロジーを基本とする Rational Application Developer などの Java Platform Debugger Architecture (JPDA) をサポートするデバッガー機能を備えた、Lotus Notes クライアントの Java Virtual Machine (JVM) の制御下で実行されている Java エージェント、Web プレビュー、およびスクリプト・ライブラリーをデバッグできます。Lotus Domino サーバー上で実行する Java コードはデバッグできませんが、JPDA により Java コードをローカルでデバッグするための多くの機能を使用できます。このセクションでは、Rational Application Developer を使用した ITSO Electronics アプリケーションのリモート Java デバッグをデモンストレーションします。

### 5.3.1 Lotus Notes クライアントでの Java デバッグの有効化

Lotus Notes クライアントが Java デバッグをサポートするのは、次のコンテキストです。各コンテキストで独自の JVM を使用している。呼び出しコードのコンテキストで実行するスクリプト・ライブラリーの Java コード。各コンテキストでは、一度に 1 人のユーザーのみデバッグを実行できます。

- ▶ 前景：「アクション」メニューからトリガーされたエージェントなど Lotus Notes クライアントで対話的に実行される Java コード。
- ▶ 背景：ローカルでスケジュールされるエージェントなど、タスク・ローダーの制御下で実行する Lotus Notes クライアントで実行される Java コード。
- ▶ Web プレビュー：フォーム上のアプレットなど、Lotus Domino Designer を使用するブラウザでプレビューされる Java コード

Lotus Notes クライアントで実行する ITSO Electronics アプリケーションで Java デバッグを有効化するには、以下の手順に従います。

1. 「ファイル」→「ツール」→「Java デバッグプリファレンス」を選択します。この操作により、図 5-5 に示すような「Java デバッグプリファレンス」ウィンドウが開きます。

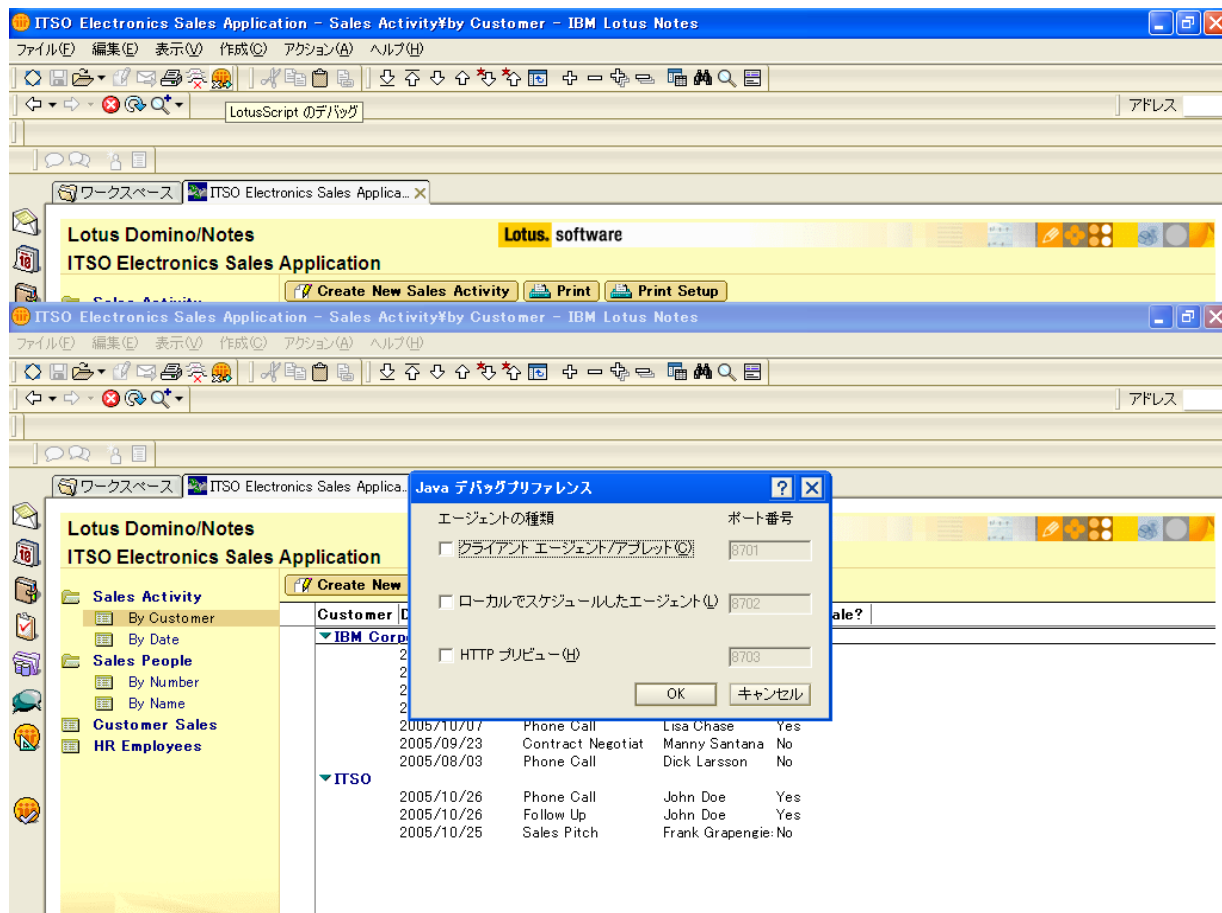


図 5-5 「Java デバッグプリファレンス」 ウィンドウ

2. デバッグを前景で使用可能にするには、「クライアント エージェント/アプレット」を選択し、101 ページの図 5-6 に示すように Lotus Notes およびデバッガー・コンピューターに接続するためのポート番号を指定します。使用不可にするには、このオプションをクリアします。

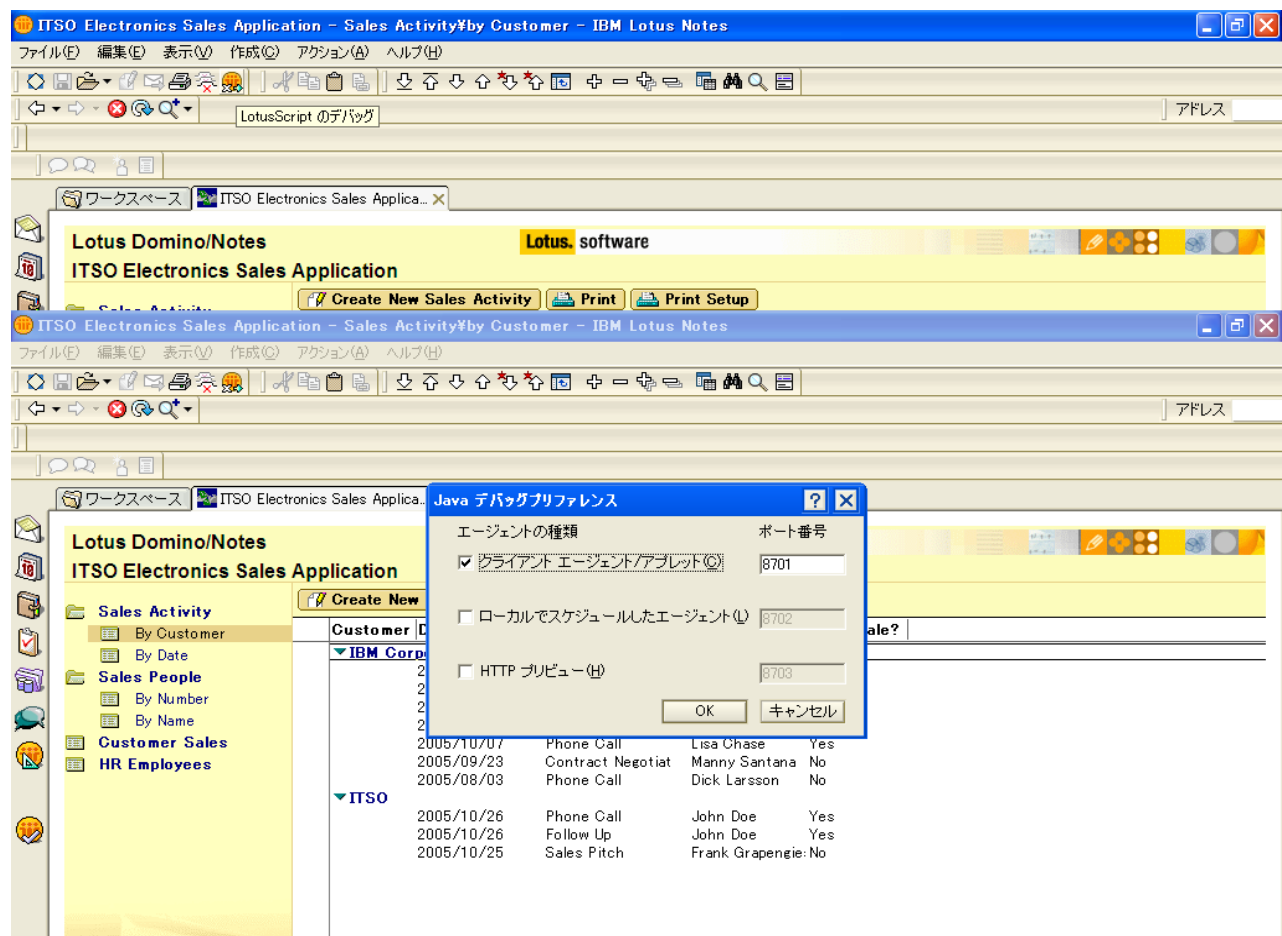


図 5-6 「クライアント エージェント/アプレット」を選択した「Java デバッグプリファレンス」

- Web プレビューのデバッグを使用可能にするには、「**Http プリビュー**」を選択して、Lotus Notes クライアントを実行するコンピュータに接続するためのポート番号を指定します。使用不可能にするには、このオプションをクリアします。

未使用のポート番号を選択してください。Windows マシンで使用中のポートを表示するには、コマンド行で次のコマンドを実行します。

```
> netstat -a
```

デフォルトでは Java デバッグは使用不可能に設定されています。

前景および背景のいずれかの設定を変更した場合は、Lotus Notes を再起動する必要があります。Web プレビューの設定を変更した場合も、プレビューを再起動する必要があります。

### 5.3.2 エージェント、Web サービス、またはスクリプト・ライブラリーで Java デバッグを使用可能にする

Java デバッグを使用可能にするには、エージェントまたは Web サービスのプロパティー・ウィンドウの「セキュリティ」タブに移動するか、スクリプト・ライブラリーのプロパティー・ウィンドウに移動してから、102 ページの図 5-7 に示すように「**デバッグ情報を含めて Java コードをコンパイルする**」を選択します。

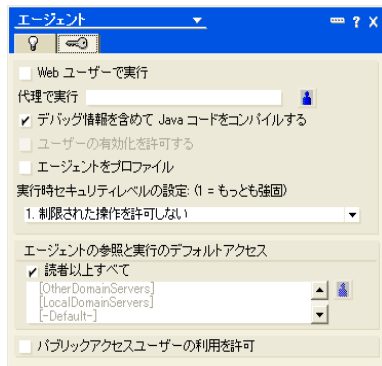


図5-7 「デバッグ情報を含めてJava コードをコンパイルする」オプション

Lotus Domino Designer 以外でデバッグする Java コードをコンパイルする場合は、**-g** オプションを指定してコンパイルします。エージェントまたは Web サービスの修正が完了したら、ご使用のデバッガーでアクセス可能なファイルにソース・コードをエクスポートしてください。

JPDA の詳細については、次を参照してください。

<http://java.sun.com/products/jpda>

### 5.3.3 デバッガーの JVM との接続

ここでは、Eclipse テクノロジーを基本とする Rational Application Developer ツールの使用を前提として説明します。他のデバッガーを使用する場合は、何らかの補間が必要です。以下の手順を実行します。

1. Lotus Notes クライアントを起動し、有効化した Java デバッグ機能でデバッグする、エージェントまたは Web サービスが含まれる ITSO Electronics アプリケーションを開きます。
2. デバッガーを開始します。

3. Java プロジェクトを作成し、図 5-8 に示すように、Java パースペクティブに切り替えます。「File」→「New」→「Project」を選択します。「Project」ウィンドウで「Java」を選択してプロジェクトの名前を入力し、「Finish」をクリックします。Java パースペクティブに切り替えます。もしくは既存のプロジェクトがある場合は、該当するプロジェクトを開きます。

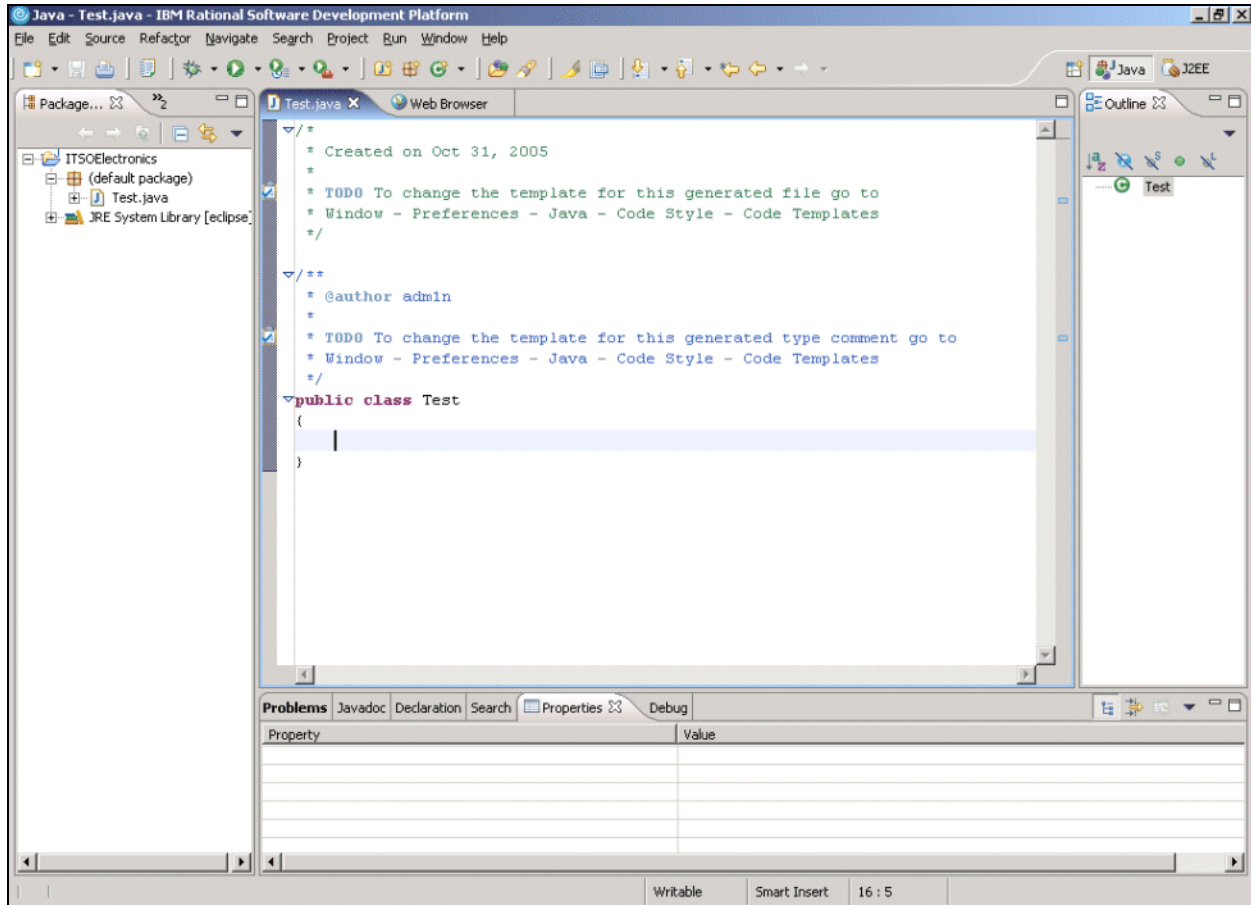


図 5-8 Java パースペクティブの表示

4. Java ソース・ファイルをインポートするには、左側のペインでプロジェクトのフォルダーを右クリックして、図 5-9 に示すように、「**Import**」を選択します。

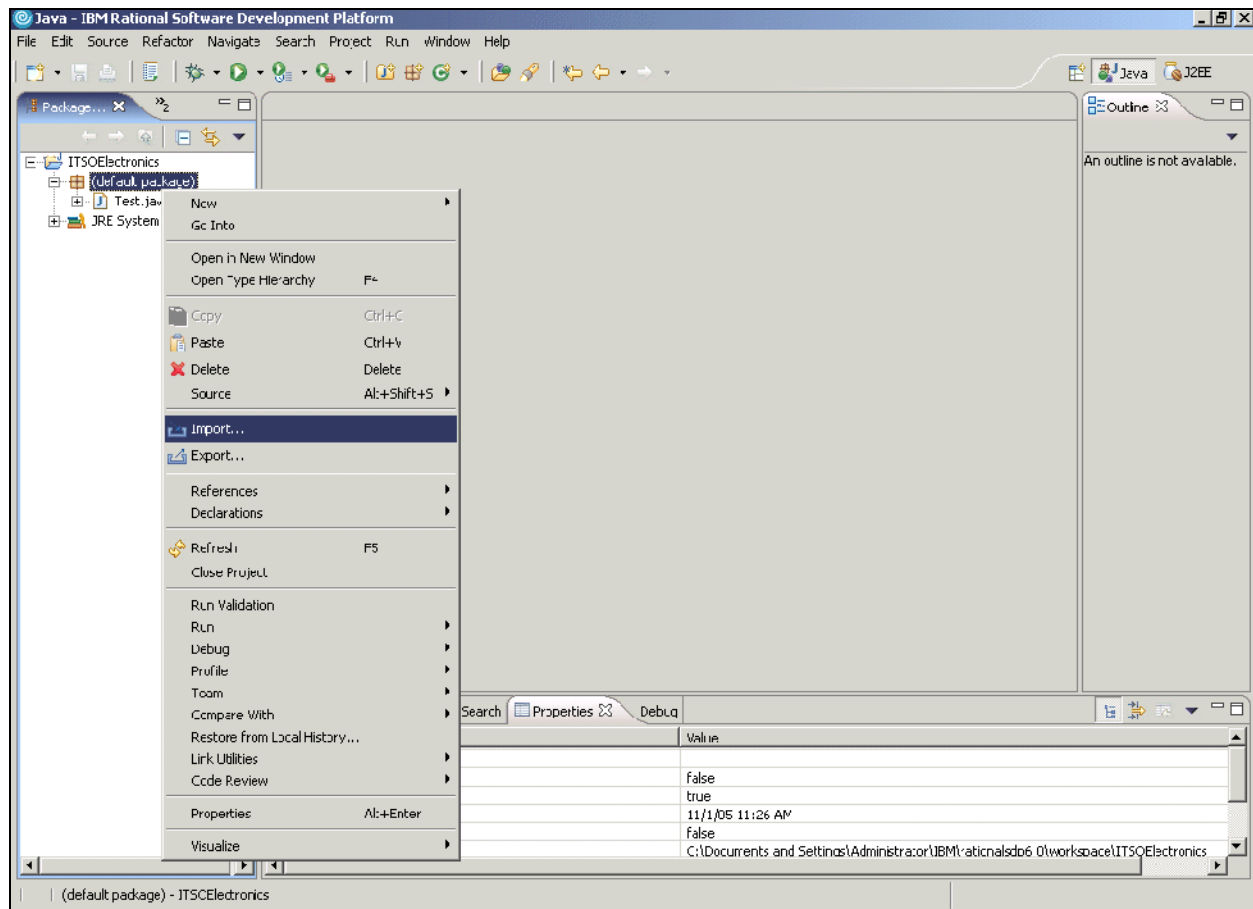


図 5-9 Java パースペクティブへのインポート

5. インポートするファイルのタイプを選択し、**Java** ソース・ファイルの場所を参照して選択します。図 5-10 のように、インポートしたファイルの保管先となるフォルダーを選択します。

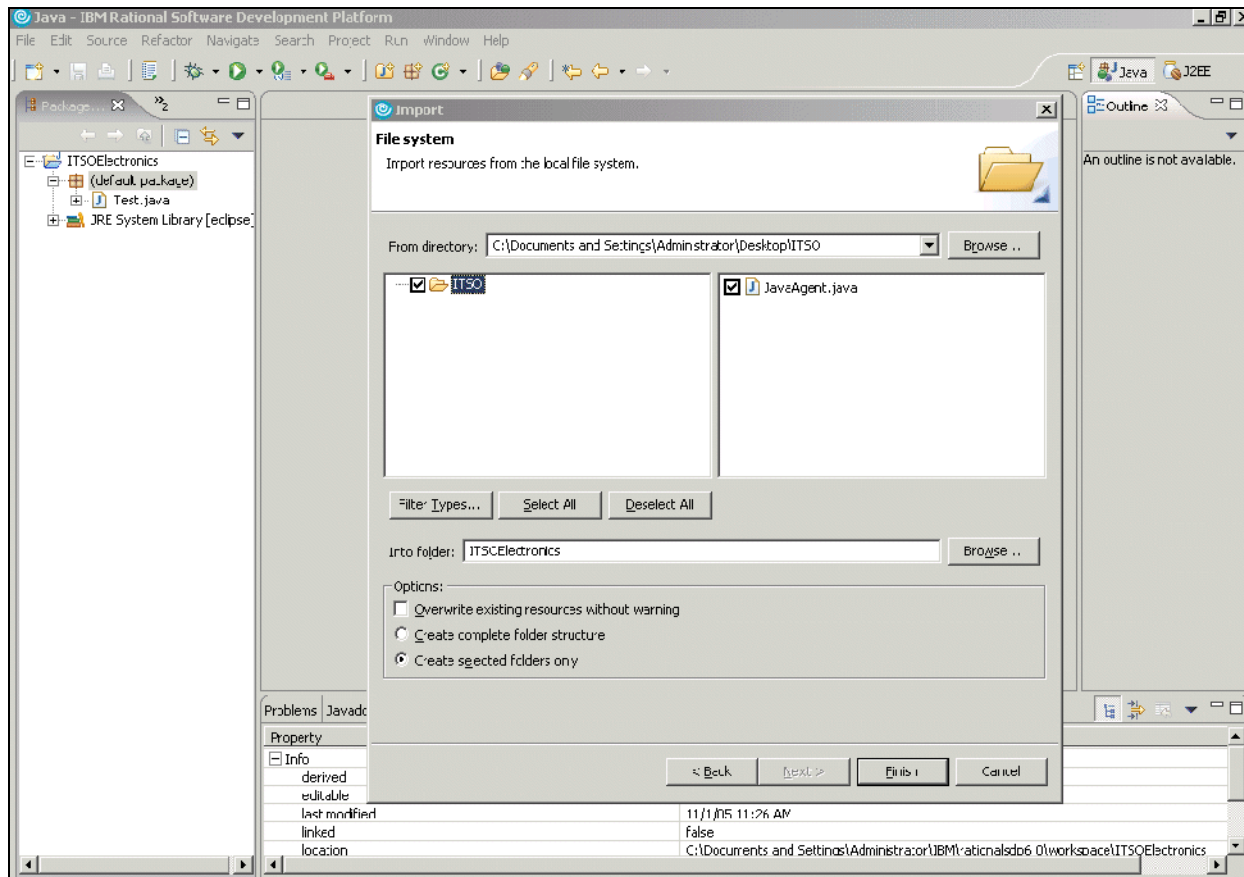


図 5-10 Java パースペクティブと「Import」ウィンドウ

6. 「**Finish**」をクリックしてファイルをインポートします。デバッガーで使用可能なのは、インポートされたソース・コードのみです。Java ソース・コードをインポートしていない場合は、実行スレッドを表示したり変数にアクセスしたりすることはできませんが、ソース・コードの表示やブレークポイントの設定は実行できません。図 5-11 を参照してください。

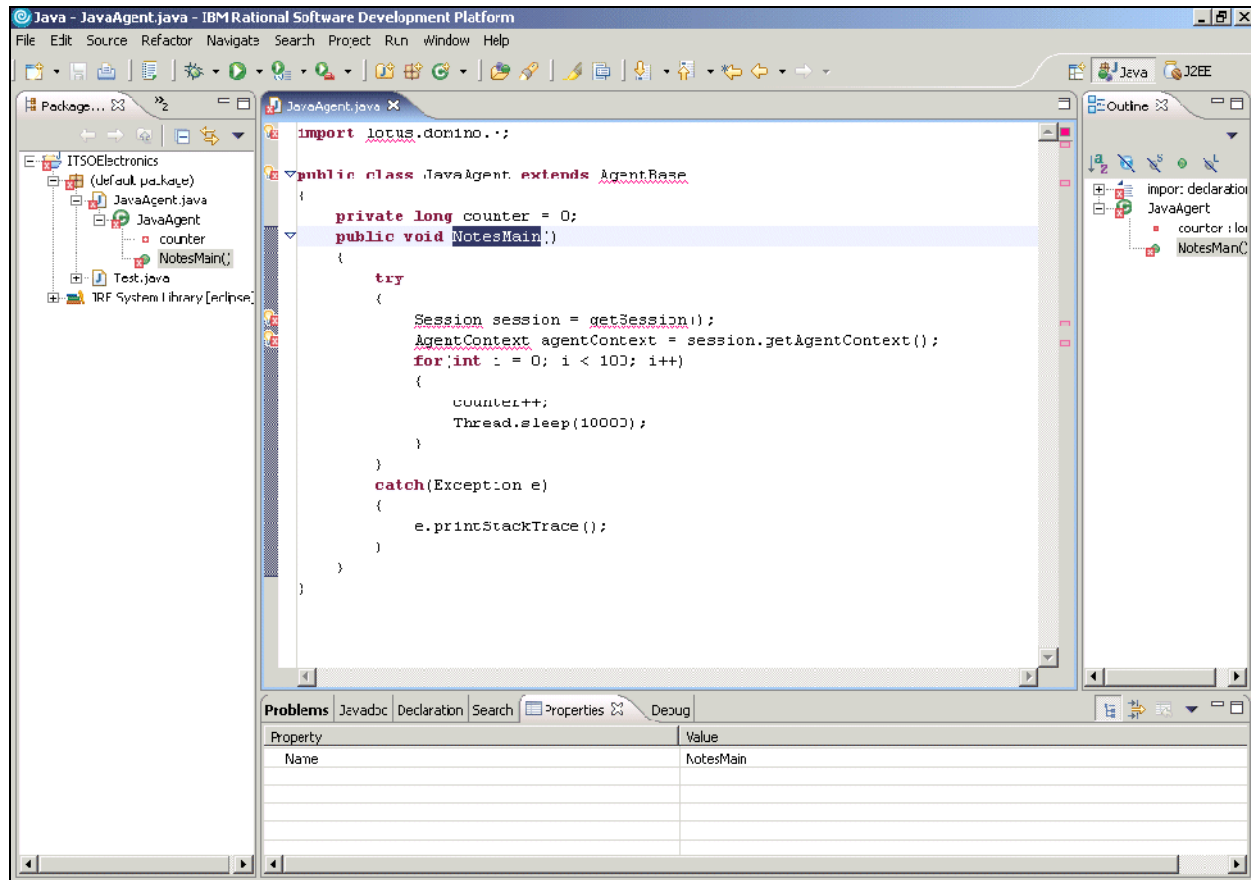


図 5-11 インポートされた Java ソース・コード

7. Lotus Notes で、デバッグするエージェントまたは Web サービスを開始します。



8. Rational Application Developer で、「実行」→「デバッグ」を選択し、図 5-12 に示すように、デバッグ構成の名前を入力します。「Connect」タブに移動します。プロジェクトを選択し、Lotus Notes コンピューターのホスト名またはアドレスを入力してから、Lotus Notes クライアントの Java デバッグ・ポート番号を入力します。Lotus Notes クライアントと Eclipse が同一マシン上にある場合は、ホスト・アドレスは、127.0.0.1 です。「Apply」をクリックしてから（初回または構成を変更した場合にのみ必要）、「Debug」をクリックしてデバッガーに接続します。

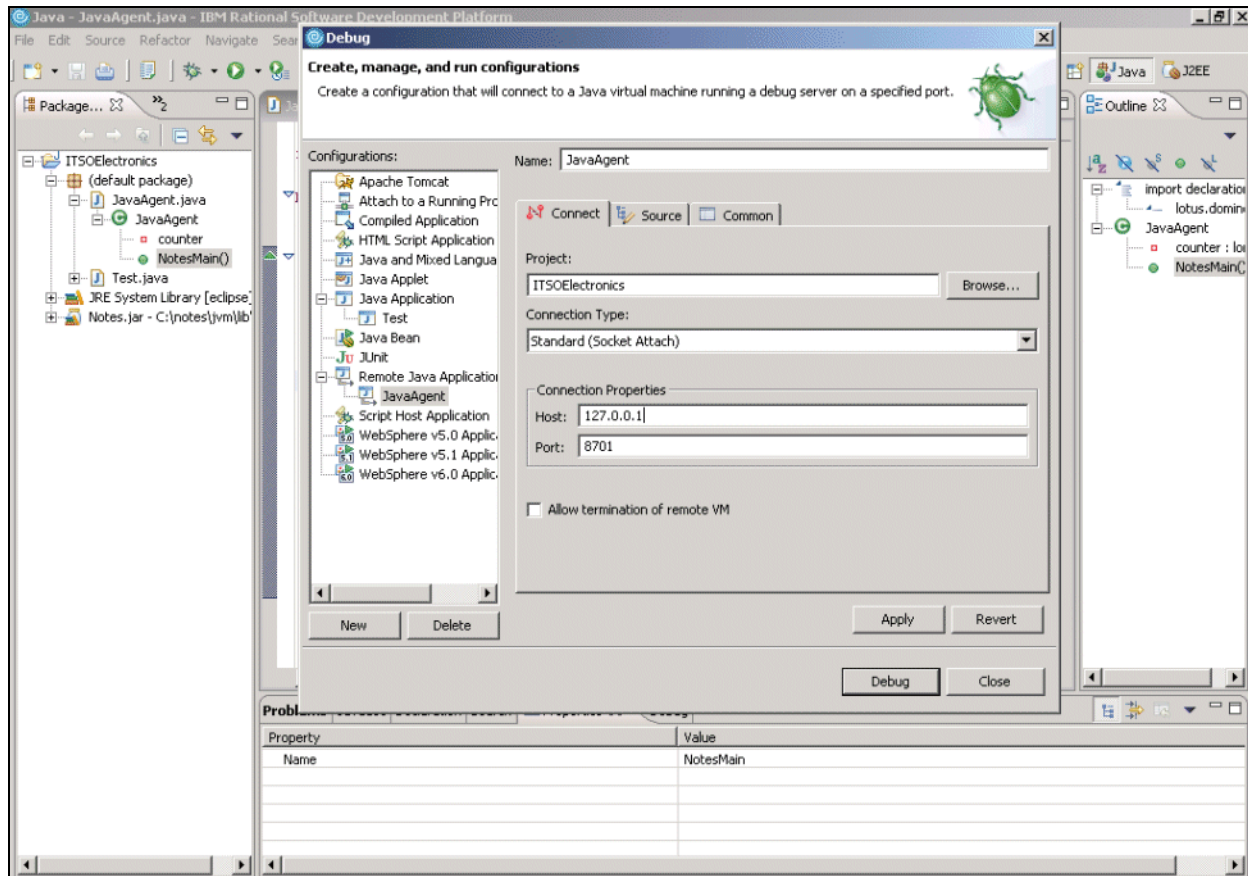


図 5-12 デバッグの接続パラメーターの設定

デバッガーに接続すると、JVM の実行スレッドが表示されます。次のようなエージェント・スレッドです。

Thread(AgentThread: JavaAgent)(Running)

エージェントのスレッドは、制御するために中断できます。この時点では、通常のデバッガー機能（ブレークポイントの設定、ステップ実行、継続、変数値の表示など）は使用可能です。

Java エージェントは、JVM にデバッガーを接続するのに十分な時間で実行する必要があります。手順の最初の段階で、例えば `sleep` ステートメントを記述したループを挿入するなどして実行を遅らせることができます。

Lotus Notes とデバッガーをそれぞれ別のコンピューターで使用していて「リモートの VM と接続できませんでした (Failed to connect to remote VM)」というエラー通知を受け取った場合は、指定したポートが別のプロセスで使用中心であると考えられます。別のポート番号を指定してから、Lotus Notes を再起動してください。

リモート Java デバッグの詳細については、次の URL で入手可能な「Lotus Domino Designer 7 Help」を参照してください。

[http://www.lotus.com/ldd/doc/domino\\_notes/7.0/help7\\_designer.nsf](http://www.lotus.com/ldd/doc/domino_notes/7.0/help7_designer.nsf)

#### 5.3.4 セキュリティー構成

ネットワークでの Java デバッグには、セキュリティー上の問題があります。Java コードをデバッグしていない場合、Java デバッグを有効化せずに Lotus Notes または Web プレビューを再起動してください。Java デバッグを使用不可にする方法については、99 ページの 5.3.1、「Lotus Notes クライアントでの Java デバッグの有効化」を参照してください。



A

## 追加の資料

この Redpaper で参照している追加の資料は、ここで説明する方法を使用してインターネットからダウンロードできます。

### Web 資料の場所

この Redpaper に関連する Web 資料は、IBM Redbooks の Web サーバーからインターネット経由でソフトコピーとして入手できます。Web ブラウザーで以下を指定してください。

<ftp://www.redbooks.ibm.com/redbooks/REDP4102>

または、次の場所にある IBM Redbooks の Web サイトへ移動してください。

[ibm.com/redbooks](http://ibm.com/redbooks)

「Additional materials」を選択し、Redpaper 資料番号 REDP4102 に対応するディレクトリを開きます。

### Web 資料の使用

この Redpaper に関連する追加の Web 資料には、ITSO Electronics Databases.zip ファイルが含まれます。

#### Web 資料の使い方

ワークステーションにサブディレクトリ (フォルダ) を作成し、Web 資料の ZIP ファイルの内容をそのフォルダ内に unzip します。次に、Lotus Domino データベースを、ローカル・ワークステーションまたは Lotus Domino サーバーにある Lotus Notes/Domino のデータ・ディレクトリ (例えば、c:\notes\data) にコピーします。



# 関連資料

本セクションでは、特にこの Redpaper で扱うトピックをさらに深く理解するために役立つような資料をリストしています。

## IBM Redbooks

以下の資料の注文に関する情報は、112 ページの「IBM Redbooks の入手方法」を参照してください。ここに記載されている一部の文書は、ソフトコピーのみの提供である場合があります。

*Domino Designer 6: 開発者用ハンドブック*、SG88-8519

*Lotus Domino 6.5.1 and Extended Products Integration Guide*、SG24-6357

*Using the Domino JSP Custom Tags: An Approach to Portalizing Domino Applications*、REDP-3902

*IBM Lotus Domino Application Portlet: Configuration and Tips*、REDP-3917

*Security Considerations in Notes and Domino 7*、REDP-4104

## オンライン・リソース

以下の Web サイトおよび URL も、詳細な情報ソースとして関係しています。

IBM Lotus Developer Domain: Lotus ソフトウェア・プラットフォームの技術情報ポータル

<http://www-06.ibm.com/jp/software/lotus/developer/>

Lotus Domino Administrator 7 Help

[http://www.lotus.com/1dd/doc/domino\\_notes/7.0/help7\\_admin.nsf/](http://www.lotus.com/1dd/doc/domino_notes/7.0/help7_admin.nsf/)

IBM Lotus Developer Domain: ドキュメント・ライブラリー

[http://www-06.ibm.com/jp/software/lotus/developer/1dd\\_doc/index.html](http://www-06.ibm.com/jp/software/lotus/developer/1dd_doc/index.html)

IBM Lotus 研修 / 技術者認定のサイト

<http://www-06.ibm.com/jp/software/lotus/techsupport/education.html>

DB2 インフォメーション・センター

<http://publib.boulder.ibm.com/infocenter/db2help/index.jsp>

Lotus Domino 7 Designer Help

[http://www.lotus.com/1dd/doc/domino\\_notes/7.0/help7\\_designer.nsf/](http://www.lotus.com/1dd/doc/domino_notes/7.0/help7_designer.nsf/)

IBM Lotus Developer Domain: *Lotus Notes/Domino 7 の自動保存のすべて*

<http://www-06.ibm.com/jp/software/lotus/developer/library/autosave.html>

IBM Lotus Developer Domain: *Lotus Notes/Domino 7 Web サービス*

[http://www-06.ibm.com/jp/software/lotus/developer/1dd\\_today/20040927\\_2.shtml](http://www-06.ibm.com/jp/software/lotus/developer/1dd_today/20040927_2.shtml)

IBM developerWorks: *Which style of WSDL should I use?*

<http://www.ibm.com/developerworks/webservices/library/ws-whichwsdl/>

IBM developerWorks: *The Hidden Power of Profile Documents*

[http://www.ibm.com/developerworks/lotus/library/ls-Profile\\_documents/](http://www.ibm.com/developerworks/lotus/library/ls-Profile_documents/)

Developing Applications with IBM Lotus Domino Enabled for DB2、E-learning Edition: Self Paced course、N7D400SP

<http://www.ibm.com/software/sw-lotus/services/education.nsf/81F04813DC9CE3C7852566DA004C5CB2/A6DF60493B0D2B13852570A7006C7E9B>

Introducing IBM DB2 Concepts course、N7D020

<https://education.lotus.com/rw/lewvschd.nsf/5FDC13DD71D1143185256575001E5133/E55E8003DCEC00FA8525701A006959C4>

データ型の説明

<http://www.w3.org/TR/xmlschema-2>

Web Service Architecture ガイド

<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211>

Apache Axis

<http://ws.apache.org/axis/>

Apache SOAP

<http://ws.apache.org/soap/>

Java Platform Debugger Architecture (JPDA)

<http://java.sun.com/products/jpda>

## IBM Redbooks の入手方法

以下の Web サイトから Redbooks のハードコピーまたは CD-ROM を注文したり、Redbooks、Redpaper、ヒント、ドラフト資料および追加資料を検索、表示、またはダウンロードしたりできます。

[ibm.com/redbooks](http://ibm.com/redbooks)

## IBM からのヘルプ

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)





# Lotus Domino 7 アプリケーションの開発

**DB2 統合によるストレージ機能およびリレーショナル機能の追加**

**Web サービスによる Lotus Domino アプリケーションの拡張および公開**

**ユーザビリティの改善と新しい設計ツールおよび診断ツール**

IBM Lotus Domino Designer ソフトウェアは、バージョン 7 で、Lotus Domino ソフトウェア・ベースのアプリケーションを開発するための最初のコラボレーション・アプリケーション開発ツールになるという期待に応えます。Lotus Domino Designer ソフトウェアは、IBM Workplace ファミリーの不可欠な部分として企業を支援し、企業が従業員の生産性を向上させ、Lotus Domino アプリケーションを迅速に構築および配置できるようにします。これにより、組織は変化する率優先的事業活動に素早く対応できます。新しい機能は、Web 標準との緊密な統合、IBM テクノロジーとのインターオペラビリティおよび使いやすさに焦点を当てています。

この IBM Redpaper では、Lotus Domino 7 の強力な新機能の使用方法を、サンプル・アプリケーションを拡張することで、アプリケーション開発者に説明します。

## INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

### 実用的な経験に基づく 技術情報の構築

IBM Redbook は IBM International Technical Support Organization によって作成されています。IBM の専門家、および世界中のカスタマーとパートナーが、現実的なシナリオに基づいてタイムリーな技術情報を作成します。ご使用の環境でより効率的に IT ソリューションをインプリメントするのに役立つ、具体的な提案を提供します。

詳細については、以下をご覧ください。  
[ibm.com/support/publications/jp/](http://ibm.com/support/publications/jp/)