

Novell exteNd Composer™ SAP Connect

4.2

ユーザガイド

www.novell.com



Novell®

保証と著作権

Copyright ©1999, 2000, 2001, 2002, 2003 SilverStream Software, LLC. All rights reserved.

SilverStream ソフトウェア製品は、SilverStream Software LLC により著作権とすべての権利が保留されています。

SilverStream は SilverStream Software, LLC の登録商標です。Novell は、Novell, Inc. の登録商標です。

ソフトウェアとマニュアルの所有権、および特許、著作権、およびそれに関連するその他のすべての財産権は常に、単独で排他的に SilverStream とそのライセンサーに保留され、当該所有権と矛盾するいかなる行為も行わないものとします。本ソフトウェアは、著作権法と国際条約規定で保護されています。ソフトウェアならびにそのマニュアルからすべての著作権に関する通知とその他の所有権に関する通知を削除してはならず、ソフトウェアとそのマニュアルのすべてのコピーまたは抜粋に当該通知を複製しなければなりません。本ソフトウェアのいかなる所有権も取得するものではありません。

Jakarta-Regexp Copyright ©1999 The Apache Software Foundation. All rights reserved. Ant Copyright ©1999 The Apache Software Foundation. All rights reserved. Xalan Copyright ©1999 The Apache Software Foundation. All rights reserved. Xerces Copyright ©1999-2000 The Apache Software Foundation. All rights reserved. Jakarta-Regexp, Ant, Xalan, Crimson, および Xerces ソフトウェアは、The Apache Software Foundation によりライセンスを付与され、Jakarta-Regexp, Ant, Xalan, Crimson, および Xerces のソースおよびバイナリ形式での再配布および使用は、変更のあるなしにかかわらず、以下の条件が満たされることを前提として許可されます。1. ソースコードの再配布に上記の著作権に関する通知、条件のリスト、および以下の権利放棄に関する通知が記載されていること。2. バイナリ形式の再配布では上記の著作権に関する通知、条件のリスト、および以下の権利放棄に関する通知がマニュアルまたは配布の際に提供されるその他の資料、あるいはその両方に記載されていること。3. エンドユーザの資料には、適宜、以下の通知を再配布の際に含めてください。「この製品には、Apache Software Foundation (<http://www.apache.org/>) により開発されたソフトウェアが含まれています」代わりに、この謝辞をソフトウェア自体に表示し、当該サードパーティに対する謝辞が通常表示される場所に表示することもできます。4. 「The Jakarta Project」、「Jakarta-Regexp」、「Xerces」、「Xalan」、「Ant」、および「Apache Software Foundation」は、書面による事前の許可なく、このソフトウェアから派生する製品を推薦したり、販売促進したりするのに使用してはなりません。書面による許可については、apache@apache.org <<mailto:apache@apache.org>> にお問い合わせください。5. 本ソフトウェアから派生する製品は「Apache」と呼ばれてはならず、「Apache」は The Apache Software Foundation の事前の書面による許可なくその名前に使用することはできません。本ソフトウェアは「現状のまま」提供され、いかなる明示的、暗黙の保証も行われるものではありません。販売可能性や特定の目的に対する適合性に対する暗黙の保証も行われません。いかなる場合でも、Apache Software Foundation またはその関係者はいかなる直接的、間接的、偶発的、特別な、免除的、または結果的な損害（代替品やサービスの調達、使用機会、データ、または利益の喪失、または業務の中断などを含む）についても、理論上責任がある場合でも、契約上の責任がある場合でも、厳密な責任、または瑕疵（怠慢などを含む）があった場合でも、ソフトウェアの使用の過程で生じ、当該損害の可能性を助言した場合であっても、責任を持ちません。

Copyright ©2000 Brett McLaughlin & Jason Hunter. All rights reserved. ソースおよびバイナリ形式での再配布および使用は、変更のあるなしにかかわらず、以下の条件が満たされることを前提として許可されます。1. ソースコードの再配布に上記の著作権に関する通知、条件のリスト、および以下の権利放棄に関する通知が記載されていること。2. バイナリ形式の再配布では上記の著作権に関する通知、条件のリスト、および以下の権利放棄に関する通知がマニュアルまたは配布の際に提供されるその他の資料、あるいはその両方に記載されていること。3. 「JDOM」という名前は、書面による事前の許可なく、このソフトウェアから派生する製品を推薦したり、販売促進したりするのに使用してはなりません。書面による許可については、license@jdom.org <<mailto:license@jdom.org>> にお問い合わせください。4. 本ソフトウェアから派生する製品は「JDOM」と呼ばれてはならず、「JDOM」は JDOM Project Management (pm@jdom.org) <<mailto:pm@jdom.org>> の事前の書面による許可なくその名前に使用することはできません。本ソフトウェアは「現状のまま」提供され、いかなる明示的、暗黙の保証も行われるものではありません。販売可能性や特定の目的に対する適合性に対する暗黙の保証も行われません。いかなる場合でも、Apache Software Foundation またはその関係者はいかなる直接的、間接的、偶発的、特別な、免除的、または結果的な損害（代替品やサービスの調達、使用機会、データ、または利益の喪失、または業務の中断などを含む）についても、理論上責任がある場合でも、契約上の責任がある場合でも、厳密な責任、または瑕疵（怠慢などを含む）があった場合でも、ソフトウェアの使用の過程で生じ、当該損害の可能性を助言した場合であっても、責任を持ちません。

Sun Microsystems, Inc. Sun, Sun Microsystems, Sun Logo Sun、Sun のロゴ、Sun Microsystems、JavaBeans、Enterprise JavaBeans、JavaServer Pages、Java Naming and Directory Interface、JDK、JDBC、Java、HotJava、HotJava Views、Visual Java、Solaris、NEO、Joe、Netra、NFS、ONC、ONC+、OpenWindows、PC-NFS、SNM、SunNet Manager、Solaris sunburst design、Solstice、SunCore、SolarNet、SunWeb、Sun Workstation、The Network Is The Computer、ToolTalk、Ultra、Ultracomputing、Ultraserver、Where The Network Is Going、SunWorkShop、XView、Java WorkShop、Java Coffee Cup のロゴ、Visual Java、および NetBeans は、米国およびその他の国の Sun Microsystems, Inc. の商標ならびに登録商標です。

Copyright ©2001 Extreme! Lab, Indiana University License. <http://www.extreme.indiana.edu>. 同社により許可が無料で、Indiana University ソフトウェアと関連する Indiana University のドキュメントファイル (「IU Software」) のコピーを取得したすべての人に、制限なく IU Software を取り扱うために付与されます。その際に、IU Software の使用、コピー、変更、マージ、公開、配布、サブライセンス、または販売、あるいはそれらのすべてに関する権利に制限はなく、IU Software が指定した人に以下の条件に基づき権利を付与します。上記の著作権に関する通知とその許可に関する通知は、IU Software のすべてのコピーおよび主要部分に含まれる必要があります。本 IU ソフトウェアは「現状のまま」提供され、いかなる明示的、暗黙の保証も行われるものではありません。販売可能性、特定の目的に対する適合性や権利侵害がないことに対する暗黙の保証も行われません。いかなる場合でも、作成者または著作権所有者は、契約上の責任がある場合でも、厳密な責任、または瑕疵 (怠慢などを含む) があつた場合でも、UI Software に関連して、または UI Software の使用やその他の取引の過程で生じた場合であっても、クレーム、損害、その他の責任について責任を持ちません。

本ソフトウェアは、著作権を持つ SSLava™ Toolkit の一部です。Copyright ©1996-1998 by Phaos Technology Corporation. All rights reserved.

Copyright © 1994-2002 W3C® (Massachusetts Institute of Technology, Institut National de Recherche Informatique et en Automatique, Keio University), all Rights Reserved. <http://www.w3.org/consortium/legal>. この W3C の成果物 (ソフトウェア、ドキュメント、またはその他の関連品目を含む) は、以下のライセンスの下で著作権所有者により提供されています。この成果物の取得、使用、またはコピー、あるいはそれらのすべてにより、ライセンシーは以下の条件を読み、理解し、遵守することに合意するものとします。本ソフトウェアとそのドキュメントの使用、コピー、変更、および配布は、変更のあるなしにかかわらず、いかなる目的でも無料または本契約で許可された使用料をもって許可されます。ただし、変更箇所を含む本ソフトウェアとドキュメントのすべてまたはその一部に以下のとおり記述することを前提とします。1. この通知の全文は、再配布物または派生物のユーザが見やすい場所に掲示しなければなりません。2. すべての前もって存在する知的所有権の放棄、通知、または条件。存在しない場合は、以下の形式の短い通知 (ハイパーテキストが望ましい、テキストでも良い) を再配布または派生コードの本文内で使用しなければなりません。「Copyright © [\$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. <http://www.w3.org/Consortium/Legal/>」3. W3C のファイルに変更または修正を加えた場合はその日付を含む通知。(コードが派生する場所への URI を示すことをお勧めします。) 本ソフトウェアは「現状のまま」提供され、いかなる明示的、暗黙の保証も行われるものではありません。販売可能性、特定の目的に対する適合性やサードパーティの特許、著作権、商標またはその他の権利を侵害しないことに対する暗黙の保証も行われません。著作権の所有者は本ソフトウェアまたはマニュアルの使用の結果生じる、直接的、間接的、特殊な、または結果的な損害に対していかなる責任も負いません。著作権所有者の名前および商標は、特別な書面による事前の承諾なしにソフトウェアに関する広告や広報に使用してはなりません。本ソフトウェアおよび関連する資料の著作権の所有権は常に、著作権所有者に帰属するものとします。

米国 Novell, Inc.
1800 South Novell Place
Provo, UT 85606

www.novell.com

SAP Connect ユーザガイド
2003 年 1 月
000-000000-000

オンラインマニュアル： この製品およびその他の Novell 製品のオンラインマニュアルや更新情報については、
www.novell.com/documentation を参照してください。

目次

このガイドについて	7
1 exteNd Composer および SAP Connect へようこそ	11
はじめに	11
exteNd Composer コネクタについて	12
SAP Connect とは	13
exteNd の SAP コンポーネントについて	16
SAP コンポーネントエディタを使用して作成できるアプリケーション	17
2 SAP コンポーネントエディタをお使いになる前に	19
SAP Java Connector (JCo) のインストール	19
SAP 接続リソースの作成	22
SAP 接続プールについて	26
コンポーネントに対する XML テンプレートの作成	30
3 SAP コンポーネントの作成	31
SAP コンポーネントを作成する前に	31
SAP コンポーネントエディタウィンドウについて	35
ネイティブ環境ペインについて	35
4 SAP アクションの実行	37
アクションについて	37
SAP Function アクション	38
SAP アクションモデル	44
Component with Connection アクション	52
SAP コンポーネントエディタでの他のアクションの使用	61
エラーおよびメッセージの処理	64
A テスト	65
アニメーションテストと配備テストの環境的相違	65
B ドキュメント管理	67
ドキュメント管理について	67
C SAP 用語集	69

このガイドについて

目的

このガイドでは、SAP コンポーネントエディタと呼ばれる exteNd Composer SAP Connect の使用方法について説明します。SAP コンポーネントエディタは、exteNd Composer で個別にインストールされるコンポーネントエディタです。

対象読者

このガイドの対象読者は、exteNd Composer を使用して、SAP アプリケーションを統合するサービスやコンポーネントを作成する開発者およびシステムインテグレータです。

前提条件

このガイドでは、読者が exteNd Composer の開発環境および配備オプションに精通しており、これらを使用したことがあるという前提で説明していきます。また、SAP の概念や SAP 環境を理解しておく必要もあります。

追加のドキュメント

Novell exteNd Composer の完全なマニュアルのセットは、Novell マニュアルの Web サイト (<http://www.novell.com/documentation-index/index.jsp>) を参照してください。

構成

このガイドは、次のように編成されています。

第 1 章「exteNd Composer および SAP へようこそ」では、SAP コンポーネントエディタの定義および概要について説明します。

第 2 章「SAP コンポーネントエディタをお使いになる前に」では、SAP コンポーネントを作成するために必要な準備について説明します。

第3章「SAP コンポーネントの作成」では、SAP コンポーネントエディタを使用した SAP コンポーネントの作成方法について説明します。

第4章「SAP アクションの実行」では、SAP Function アクションの使用方法について説明します。

第5章「接続プール」では、接続プールの作成方法および使用方法について説明します。

付録 A 「テスト」では、アニメーションテスト時と配備テスト時における環境的相違について説明します。

付録 B 「ドキュメント管理」では、SAP コンポーネントエディタ内でのドキュメント管理について説明します。

付録 C は、用語集です。

このガイドで使用する表記規則

このガイドで使用する表記規則は、次のとおりです。

手順での「太字」フォントは、次のアクション項目を示します。

- ◆ メニューの選択
- ◆ フォームの選択
- ◆ ダイアログボックス項目

太字の **Sans Serif** フォントは、次の項目に使用します。

- ◆ Uniform Resource Identifier
- ◆ ファイル名
- ◆ ディレクトリおよびパス名の一部

「斜体」フォントは、次の項目を示します。

- ◆ 入力する変数情報
- ◆ 新出の技術用語
- ◆ 他の Novell 出版物のタイトル

「モノスペース」フォントは、次の項目を示します。

- ◆ メソッド名
- ◆ コードの例
- ◆ システム入力
- ◆ オペレーティングシステムオブジェクト

1

exteNd Composer および SAP Connect へようこそ

はじめに

『Novell exteNd Composer SAP Connect ガイド』へようこそ。このガイドは、Composer の全機能 (Connect コンポーネントエディタを除く) の使用方法が詳しく説明されている『exteNd Composer ユーザガイド』に付属しています。『ユーザガイド』をご覧になっていない場合は、このガイドを使用する前に読んで内容を確認してください。

exteNd Composer には、SAP Connect など、Connect ごとに異なるコンポーネントエディタが用意されています。各コンポーネントエディタの特殊な機能は、これと同じような別のガイドで説明されています。

exteNd Composer を使用しており、コアコンポーネントエディタである XML Map コンポーネントエディタに精通している場合は、このガイドに従って SAP コンポーネントエディタを簡単に使用することができます。

作業を始める前に、まず SAP Connect を既存の exteNd Composer にインストールしておく必要があります。また、このコネクタで作成されたサービスを exteNd Composer Enterprise Server 環境で実行するには、このコネクタ用のサーバ側ソフトウェアが exteNd Composer Enterprise Server にインストールされている必要があります。

注記： このコンポーネントエディタを正しく使用するには、SAP 環境と、XML に対応させるアプリケーションに慣れ親しんでおく必要があります。

exteNd Composer コネクタについて

exteNd Composer は、単純なハブ & スポークアーキテクチャに基づいて構築されています。ハブは、XML ドキュメントを使用して要求を受け付け、XML に対応したアプリケーション上でこのようなドキュメントやインターフェースで変換プロセスを実行し、XML 応答ドキュメントを返す強力な XML 変換エンジンです。スポーク（つまりコネクタ）は、XML 対応でないデータのソースを「XML に対応させる」プラグインモジュールで、データをハブに送信して XML として処理します。これらのデータソースには、レガシー COBOL/VSAM で管理されている情報から、HTML ページに対するメッセージキューまで何でも使用できます。exteNd Composer コネクタは、情報ソースを XML に対応させるために各製品で使用されている統合方法に従って分類できます。統合方法は、インターネットベースのコンピュータアーキテクチャに対する現在のシステム設計において使用される主要な区分を反映したものです。exteNd Composer では、eBusiness のニーズとレガシーアプリケーションのアーキテクチャに応じて、ユーザインタフェースレベル、プログラム論理レベル、またはデータレベルでビジネスシステムを統合できます。

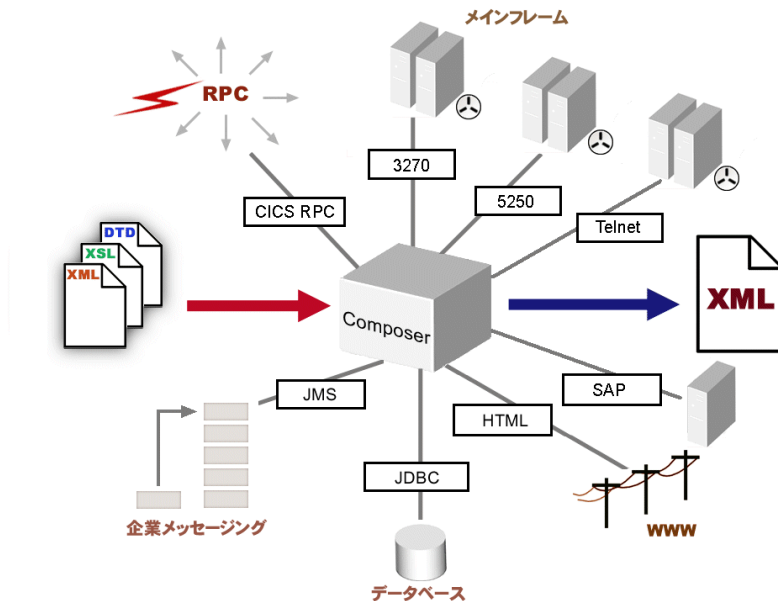


図 1-1

SAP Connect とは

SAP は、一連の統合モジュールを介して、コアビジネスプロセスに対して、企業リソースソフトウェアを提供します。これらのコアビジネスプロセスの例には、在庫、受注、受取勘定、支払勘定、生産スケジュール、および給与支払があります。

リモートファンクションコール (RFC: Remote Function Call) プロトコルは、SAP と外部コンポーネント間のすべての通信の中心部です。

ABAP (Advanced Business Application Programming) は、SAP プログラミング言語です。ABAP 関数モジュールは、RFC に対応している場合のみ、外部クライアントから呼び出すことができます。SAP R/3 には、RFC に対応した関数モジュール (RFM) が数多く含まれています。

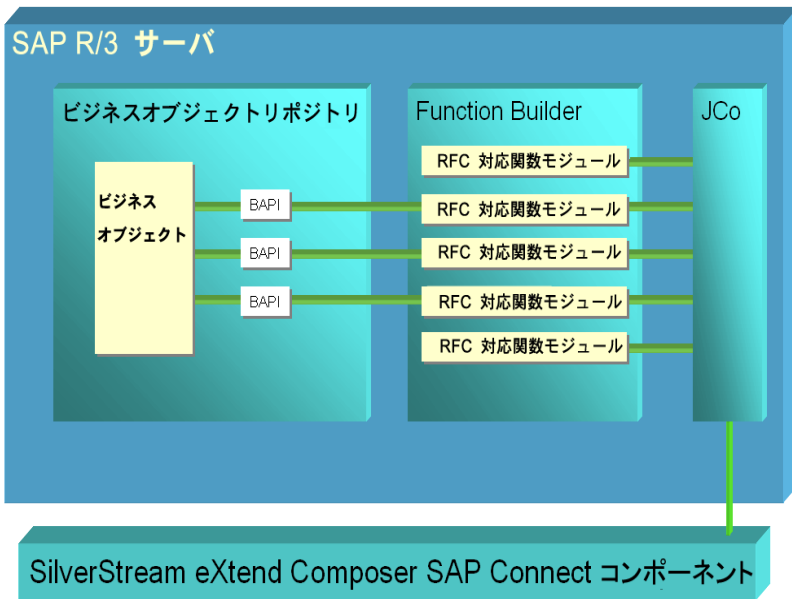
これらの RFM には、BAPI (Business Application Programming Interface) が含まれます。BAPI とは、追加規則 (『SAP BAPI Programming Guide』を参照) に従う RFM のことで、オブジェクトタイプのメソッドとして BOR (Business Object Repository) で定義されています。

ビジネスオブジェクトタイプは、SAP システムのベースとなるオブジェクト指向構造のことで、1 つのオブジェクト (たとえば、販売注文) は、1 つの業務詳細を表し、この詳細の関数やデータが含まれます。ビジネスオブジェクトは、ビジネスオブジェクトタイプのインスタンスです。

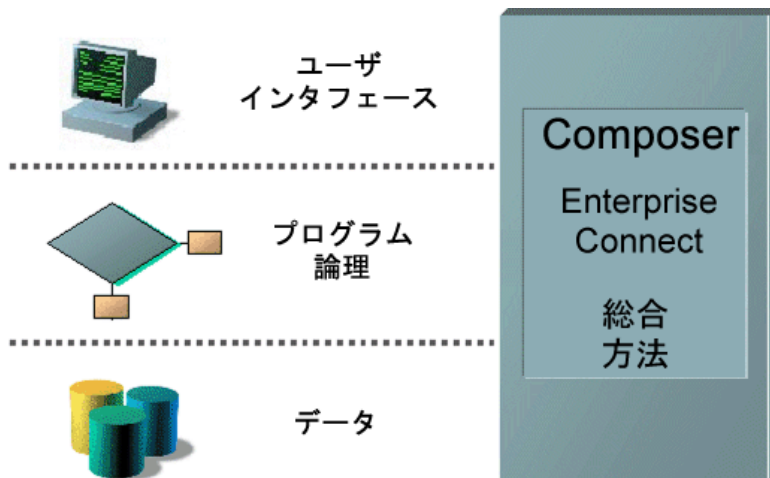
すべての SAP ビジネスオブジェクトタイプは、SAP R/3 BOR 内で定義および記述されます。BOR は、外部アプリケーション (たとえば、Composer 統合アプリケーション) から SAP ビジネスオブジェクトタイプにアクセスするときの中央アクセスポイントです。

BAPI では、外部アプリケーションから SAP ビジネスオブジェクトを呼び出すことができます。SAP Java コネクタ (JCo) は、BAPI へのアクセスを提供する Java API です。

このアーキテクチャは、次のモデルのとおりです。



SAP Connect では、SAP BAPI およびその他の RFC に対応した関数モジュール (RFM) への着信および発信呼び出しが可能です。exteNd Composer SAP Connect は、SAP JCo (Java コネクタ) を介して SAP 環境と統合することを目的として設計されました。SAP Connect コンポーネントは、SAP JCo ライブラリを使用して SAP RFM を呼び出して、SAP プログラムと通信します。Composer SAP Connect XML では、プログラム論理統合方法を使用した SAP R/3 アプリケーションが可能です。

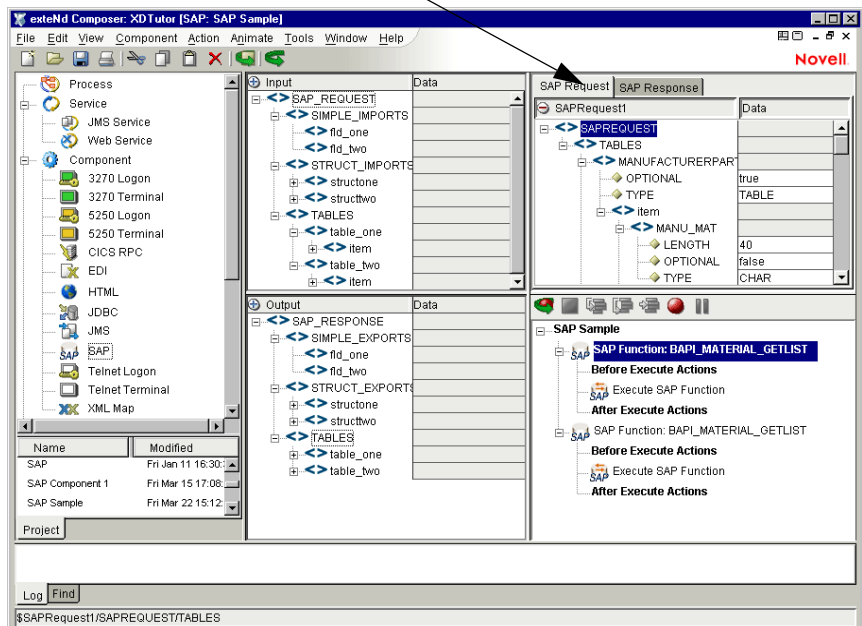


SAP Connect を使用すると、SAP アプリケーションやそれらのビジネス論理をインターネットプロセス、エクストラネットプロセス、またはイントラネットプロセスに対して利用可能にすることができます。SAP Function アクションでは、実行する SAP Function を検索および選択できます。目的の関数を検索して選択すると、データのマップ先となる [SAP Request] フィールド、およびデータのマップ元となる [SAP Response] フィールドを選択できます。これらの [SAP Request] フィールドおよび [SAP Response] フィールドは、ネイティブ環境ペイン (NEP) の [SAP Request] タブおよび [SAP Response] タブにそれぞれ表示されます。

選択した SAP 関数の要求データフィールドが NEP の [SAP Request] タブに表示されている場合、XML 要求ドキュメントから [SAP Request] フィールドにデータをマップできます。このデータは、SAP 関数の実行前にマップされます。

同様に、選択した SAP 関数の応答データフィールドが NEP の [SAP Response] タブに表示されている場合、[SAP Response] フィールドから XML 要求ドキュメントにデータをマップできます。このデータは、SAP 関数の実行後にマップされます。

ネイティブ環境ペインの[SAP Request]タブと[SAP Response]タブ



exteNd の SAP コンポーネントについて

SAP Connect は、exteNd Composer Service に追加できる SAP コンポーネントを作成します。XML Map コンポーネントと同様に、SAP コンポーネントは、2つの異なる XML テンプレート (つまり、要求 XML ドキュメントと応答 XML ドキュメント) 間でデータをマップ、変換、および転送するために設計されています。また、SAP BAPI またはその他の RFC に対応した関数モジュール (RFM) への着信、発信呼び出しが可能です。

任意のデータ交換操作と同様、SAP コンポーネントは、接続リソースに依存しています。接続リソースは、ホストアドレス、IP アドレス、SAP システム番号、SAP クライアント番号、SAP 言語、および接続のユーザ ID、パスワード、プーリングオプションに関する重要な情報を指定します。SAP 接続リソースを設定すると、要求データを処理し、応答データを返す SAP 関数を呼び出す SAP コンポーネントの設定に使用できます。

SAP コンポーネントでは、XML ドキュメントから SAP アプリケーションへのデータのマップや転送、および XML ドキュメントへのデータの配置などの単純なデータ操作を実行できます。また、マップなどの高度な操作も実行できます。SAP コンポーネントには XML Map コンポーネントの全機能が備わっており、XSL の処理、メールの送信、および HTTP プロトコルを使用した XML ドキュメントのポストと受信を実行できます。

接続を介した SAP コンポーネントおよびメインフレームのデータの関係は、次の図のとおりです。

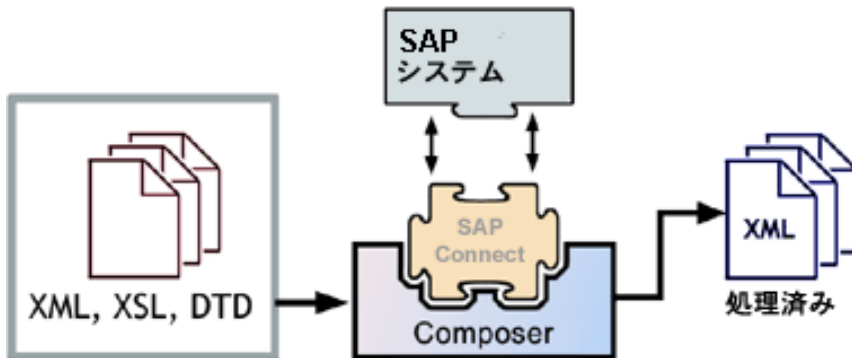


図 1-2

SAP コンポーネントエディタを使用して作成できるアプリケーション

SAP コンポーネントエディタを使用すると、作成している XML 統合を拡張して、SAP アプリケーションに含めることができます (詳細については、『**exteNd Composer ユーザガイド**』を参照してください)。たとえば、定期的に更新されるデータベースから製品の説明、画像、価格、および在庫情報を取得して Web ブラウザに表示するアプリケーションがあるとします。SAP コンポーネントエディタを使用すると、動作中のシステムから最新の製品情報を取得したり、データベースから静的情報 (画像など) を取得し、個別の情報ソースの情報をマージしてから、ユーザに対して表示できます。これにより、内部ユーザと外部ユーザの両方に同じ最新情報が提供されます。

2

SAP コンポーネントエディタをお使いになる前に

SAP Java Connector (JCo) のインストール

exteNd Composer SAP Connect は、SAP JCo (Java コネクタ) ライブラリを介して SAP 環境と統合することを目的として設計されています。特定のソフトウェアバージョンのサポートについては、『exteNd Composer SAP Connect ユーザガイド』のリリースノートを参照してください。

登録済みの SAP ユーザは、SAP サイト (<http://service.sap.com>) から SAP JCo ライブラリをダウンロードできます。この SAP サイトにアクセスするには、有効な SAP Service Marketplace ユーザ ID およびパスワードが必要です。

エラーメッセージ

注記： exteNd Composer SAP Connect をインストールするには、すでに SAP Java コネクタをインストールしている必要があります。

Composer SAP Connect をインストールする前に SAP Java コネクタをインストールしていない場合 (後の説明を参照)、次のダイアログボックスが表示されます。



また、ログには次のメッセージが記録されます。

```
java.lang.NoClassDefFoundError:  
com/sap/mw/jco/JCO$PoolChangedListener
```

exteNd Composer SAP Connect Developer

exteNd Composer SAP Connect Developer については、SAP の Windows NT、Windows 2000 への JCO インストール手順に従ってください。

- 1 SAP サイトから、jco-ntintel-1.1.04.zip を任意のディレクトリ {jco-install-path} にダウンロードします。
- 2 jco-ntintel-1.1.04.zip を任意のディレクトリ {jco-install-path} に解凍します。
- 3 DLL を extendComposer\Designer\bin ディレクトリに追加します。
- 4 jCO.jar を exteNdComposer\Designer\lib ディレクトリに追加します。
- 5 jCO.jar の <JAR> エントリを xConfig の SAP セクションに追加します。例：
<JAR>..\lib\jCO.jar</JAR>

Windows NT、Windows 2000 環境での exteNd Composer SAP Connect Server

exteNd Composer SAP Connect Server については、SAP の Windows NT、Windows 2000 への JCO インストール手順に従ってください。

- 1 SAP サイトから、jco-ntintel-1.1.04.zip を任意のディレクトリ {jco-install-path} にダウンロードします。

- 2 jco-ntintel-1.1.04.zip を任意のディレクトリ {jco-install-path} に解凍します。
- 3 実行時に見つけることができるように、DLL をパス上のアプリケーションサーバディレクトリツリー内に追加します。たとえば、アプリケーションサーバ bin ディレクトリです。
- 4 jCO.jar を Server の exteNdComposer\lib ディレクトリに追加します。
- 5 [PATH]\exteNdComposer\lib\jCO.jar をアプリケーションサーバの CLASSPATH に追加します。

Solaris 環境での exteNd Composer SAP Connect Server

exteNd Composer SAP Connect Server については、SAP の Solaris への JCO インストール手順に従ってください。

- 1 SAP サイトから、jco-sun-1.1.04.tgz を任意のディレクトリ {jco-install-path} にダウンロードします。
- 2 jco-sun-1.1.04.tgz または jco-sun-1.1.04.tar.z を任意のディレクトリ {jco-install-path} に展開します。
- 3 ランタイム時に見つけることができるように、SO を LD_LIBRARY_PATH 上のアプリケーションサーバディレクトリツリー内に追加します。
- 4 jCO.jar を Server の exteNdComposer\lib ディレクトリに追加します。
- 5 [PATH]\exteNdComposer\lib\jCO.jar を CLASSPATH 環境変数に追加します。

AIX 環境での exteNd Composer SAP Connect Server

exteNd Composer SAP Connect Server については、SAP の AIX への JCO インストール手順に従ってください。

- 1 SAP サイトから、jco-rs6000-1.1.04.tgz または jco-rs6000-1.1.04.tar.z を任意のディレクトリ {jco-install-path} にダウンロードします。
- 2 jco-rs6000-1.1.04.tgz または jco-rs6000-1.1.04.tar.z を任意のディレクトリ {jco-install-path} に展開します。
- 3 ランタイム時に見つけることができるように、SO を LIBPATH 上のアプリケーションサーバディレクトリツリー内に追加します。
- 4 jCO.jar を Server の exteNdComposer\lib ディレクトリに追加します。
- 5 [PATH]\exteNdComposer\lib\jCO.jar を CLASSPATH 環境変数に追加します。

HP-UX 環境での exteNd Composer SAP Connect Server

exteNd Composer SAP Connect Server については、SAP の HP-UX への JCO インストール手順に従ってください。

- 1 SAP サイトから、jco-hp_32-1.1.04.tgz または jco-hp_32-1.1.04.tar.z を任意のディレクトリ {jco-install-path} にダウンロードします。
- 2 jco-hp_32-1.1.04.tgz または jco-hp_32-1.1.04.tar.z を任意のディレクトリ {jco-install-path} に展開します。
- 3 実行時に見つけることができるように、SL を SHLIB_PATH 上のアプリケーションサーバディレクトリツリー内に追加します。
- 4 jCO.jar を Server の exteNdComposer/lib ディレクトリに追加します。
- 5 [PATH]exteNdComposer/lib/jCO.jar を CLASSPATH 環境変数に追加します。

SAP 接続リソースの作成

SAP コンポーネントを作成する前に、SQL システムにアクセスするための接続リソースを作成する必要があります。

SAP Connect を含む各コネクタでは、独自の接続リソースタイプが使用されます。接続リソース (JDBC、3270、5250、CICS RPC、JMS、HTML、Telnet、EDI など) では、接続される外部データソースに該当するさまざまなパラメータタイプが必要です。設定ウィザードの外観は、選択された接続リソースのタイプを反映するために動的に変化します。

接続リソースを一度作成すると、作成する複数の SAP コンポーネントに対して (それぞれに新しい接続を作成せずに) 再利用できます。また、接続リソースを作成すると、接続に関連するパラメータ値を制御する ECMAScript 式にデータフィールドをリンクできるという点で、ある程度自動設定できます。

定数駆動型および式駆動型の接続パラメータについて

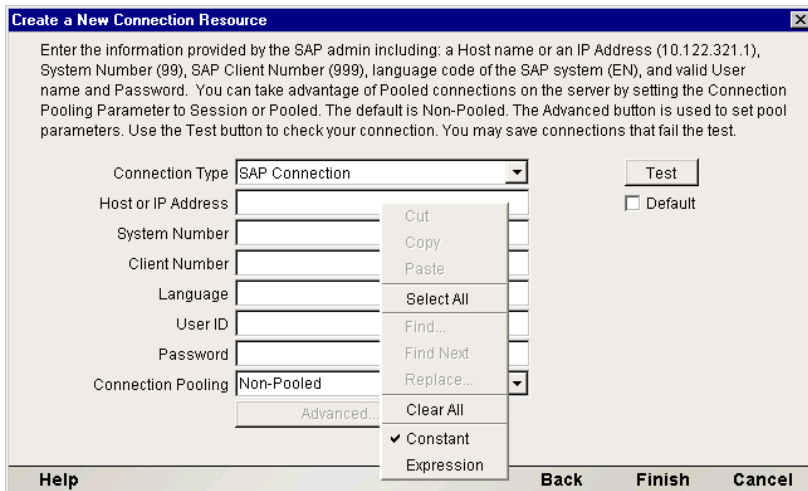
接続パラメータの値は、定数または式としての方法のうちいずれかを使用して指定できます。定数ベースのパラメータでは、接続が使用されるたびに [Connection] ダイアログボックスに入力した値を使用します。式ベースのパラメータでは、ランタイム時に接続が使用されるたびに異なる値となり得る、プログラムのな式を使用して値を設定できます。この場合、接続の動作は柔軟になり、接続の使用ごとにランタイム時の条件に対応できるようになります。

たとえば、SAP 接続における式駆動型のパラメータの非常に単純な使用の 1 つは、ユーザ ID とパスワードを PROJECT 変数 (例:

PROJECT.XPATH (“USERCONFIG/MyDeployUser”) として定義することです。このようにすると、プロジェクトを配備する際に、Deployment Wizard で PROJECT 変数を最終配備環境に適切な値に更新できます。それとは正反対に、アプリケーションサーバで Java ビジネスオブジェクトを照会するカスタムスクリプトを使って、使用するユーザ ID とパスワードを決定することもできます。

➤ 定数駆動型から式駆動型にパラメータを切り替える

- 1 変更するパラメータフィールドでマウスを右クリックします。
- 2 コンテキストメニューから **[Expression]** を選択すると、エディタボタンが表示されるか、または有効になります。
- 3 ボタンをクリックしてから、ランタイム時に有効なパラメータ値を返す式を作成します。



➤ SAP 接続リソースを作成する

- 1 **[File] > [New xObject] > [Resource] > [Connection]** の順に選択します。「Create a New Connection Resource」ウィザードが表示されます。

Create a New Connection Resource

A Connection resource is used to establish communications with an Connector data source or with a server using HTTP authentication. You need to create connections for each type of data source or each HTTP server you wish to communicate with. Enter a name and, optionally, a description for this Connection. The name will appear in the Composer Detail Pane and in choice lists when you are prompted for objects in Composer. The name may not contain the characters: \ / ? " < > . | Names are case insensitive.

Name:
SAPConnection

Description:
Purpose:
Input:
Output:
Remarks:

Help Back Next Cancel

- 2 接続オブジェクトの「名前」を入力します。
- 3 オプションとして、「説明」テキストを入力します。
- 4 [Next] をクリックします。

Create a New Connection Resource

Enter the information provided by the SAP admin including: a Host name or an IP Address (10.122.321.1), System Number (99), SAP Client Number (999), language code of the SAP system (EN), and valid User name and Password. You can take advantage of Pooled connections on the server by setting the Connection Pooling Parameter to Session or Pooled. The default is Non-Pooled. The Advanced button is used to set pool parameters. Use the Test button to check your connection. You may save connections that fail the test.

Connection Type: SAP Connection [Test]

Host or IP Address: 10.122.321.1 [Default]

System Number: 99

Client Number: 999

Language: EN

User ID: pryan

Password: *****

Connection Pooling: Non-Pooled [Advanced...]

Help Back Finish Cancel

- 5 [Connection Type]プルダウンメニューから[SAP Connection]を選択します。
- 6 「ホスト」の名前または「IP アドレス」を入力します。
- 7 「SAP システム番号」を入力します。
- 8 「SAP クライアント番号」を入力します。
- 9 「SAP 言語」(EN は English)を入力します。

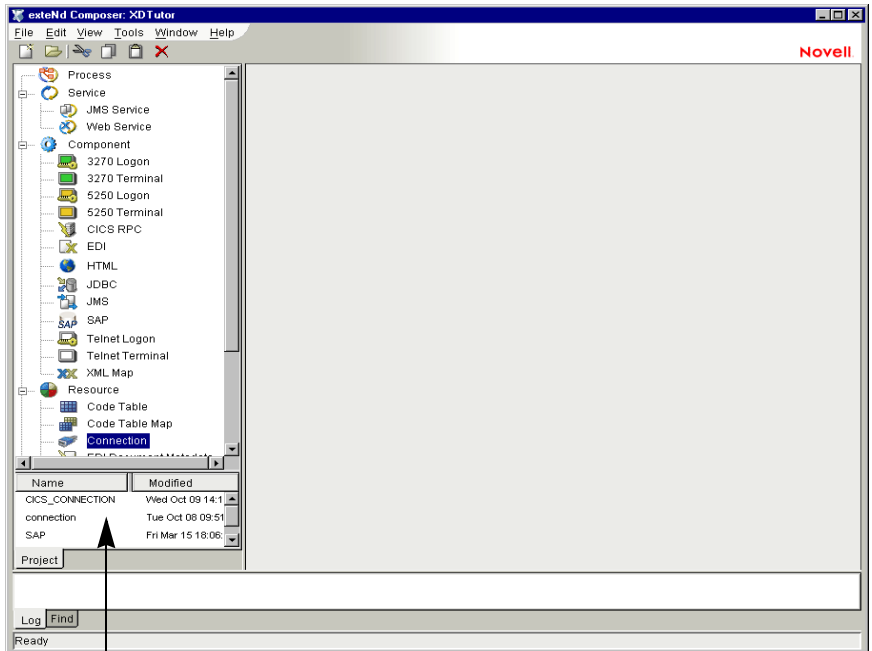
- 10** 有効な「**接続ユーザ ID**」を入力して、選択したデータベースにサインオンします。
- 11** 選択したデータベースに有効な「**接続パスワード**」を入力します。
- 12** [**Connection Pooling**] オプションを選択します。ドロップダウンリストで、ドロップダウンボックスから [**Non-Pooled**]、[**Session**]、または [**Pooled**] を選択します。
- ◆ [**Non-Pooled**] を選択すると、要求時に接続が作成されます。
 - ◆ [**Pooled**] を選択すると、接続がプールマネージャにより提供されます。
 - ◆ [**Session**] を選択すると、接続がセッション接続マネージャにより提供されます。

[**Connection Pooling**] に対して [**Pooled**] または [**Session**] オプションを選択すると、[**Advanced...**] ボタンを使用できるようになります。

[**Test**] をクリックして、接続コンポーネントから [**Pooled**] 接続を要求すると、接続が接続プールマネージャから取得および解放され、プールが削除されます。[**Test**] をクリックして、**SAP** コンポーネントから [**Pooled**] 接続を要求する、またはコンポーネントにより接続が要求されると、接続が接続プールマネージャから取得および解放されます。ただし、プールは削除されません。

[**Test**] をクリックして、接続コンポーネントから [**Session**] 接続を要求すると、接続が取得され、接続プールマネージャに解放されます。次にセッション接続が削除されます。[**Test**] をクリックして、**SAP** コンポーネントから [**Session**] 接続を要求する、またはコンポーネントにより接続が要求されると、接続が取得され、接続プールマネージャから解放されます。ただし、セッション接続マネージャは削除されません。

- 13** プロジェクトで作成した新しい **SAP** コンポーネントのデフォルトの接続として現在の接続を使用する場合は、[**Default**] チェックボックスをオンにします。
- 14** [**Test**] をクリックして、正常に接続されるかどうかを確認します。「**success**」または「**failure**」メッセージが、接続に対して表示されます。接続が失敗した場合でも、リソースの作成を続けることができます。
- 15** [**Finish**] をクリックします。新しく作成されたリソース接続オブジェクトが、**Composer** 接続リソースの詳細ペインに表示されます。



新しく作成したリソース

SAP 接続プールについて

Composer SAP Connect では、接続プール管理およびセッション接続管理の両方がサポートされています。

接続プール管理では、JCo プールマネージャにより JCo ライブラリを介して接続のプールへのアクセスが提供されます。

セッション接続管理では、リンク先の HTTP セッションの期間を保持する接続が提供されます。

セッション接続管理および接続プール管理の両方は、Composer SAP Connect Designer と Composer SAP Connect Server の両方でアクティブです。

セッション接続管理

セッション接続管理を使用すると、サーバ固有のタイムフレーム (サーバ要求タイムアウト) で複数のサーバ要求を作成するときのパフォーマンスが向上します。接続は、サーバ http 要求に基づいて維持されます。また、接続は、セッションが維持されている限り、セッション接続マネージャから使用できます。セッションがタイムアウトになると、セッション接続マネージャは削除されます。

セッション接続プールとの SAP 接続では、セッション接続マネージャとセッション接続は、接続を使用するコンポーネントが最初に実行されるときに作成されます。

セッション接続プールが SAP 接続リソースに選択された場合、[Advanced] ボタンを使用すると、[Session Info] ダイアログボックスが表示されます。[Session Info] ダイアログボックスでは、管理対象のセッション接続に対して [Keep Alive] および [Inactivity Lifetime] オプションを設定できます。

Session Info

Enter the values to use to keep the connection open when there is no activity. Keep Alive sets the interval after which we will ping the SAP system to keep the connection open. Inactivity Lifetime sets the time limit for an idle connection.

Keep Alive (minutes) 5

Inactivity Lifetime (minutes) 60

Help OK Cancel

5 分ごとの通信

60 分間アクティブな接続

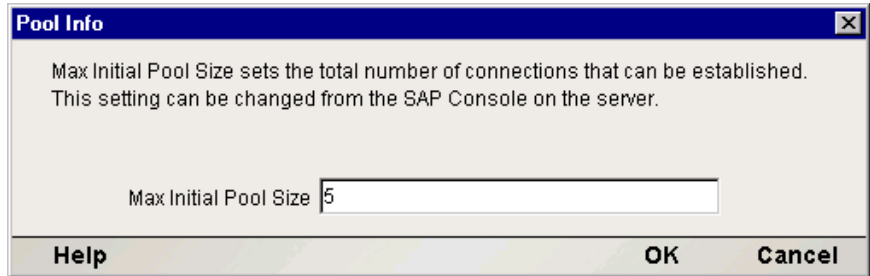
[Keep Alive] (デフォルトは 5) - 接続が維持されているかどうかを確認する間隔です。この期間が過ぎると、Composer では SAP システムが Ping され、接続が開かれたままの状態になります (つまり維持されます)。

[Inactivity Lifetime] (デフォルト 60) - HTTP セッションのタイムアウトを無効にするタイムアウト期間 (分単位) です。これは、アイドル接続に対する時間制限です。

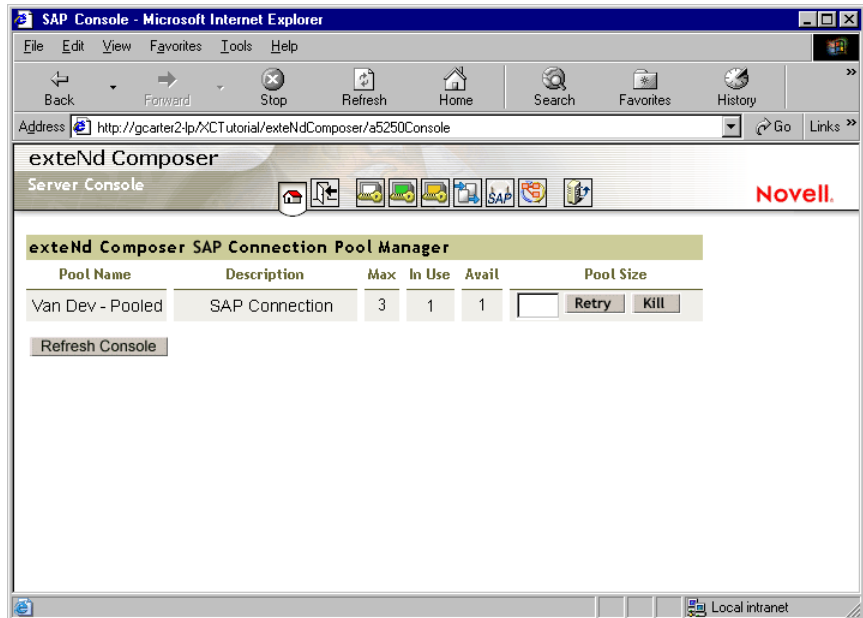
接続プール管理

[SAP Connection] ダイアログボックスでは、SAP 接続のサーバ接続ソースとして [Pooled] を指定すると、JCo Pool Manager を使用するプールは、接続を使用するコンポーネントが最初に作成される時、またはユーザが SAP コンソールからプールを事前に初期化したときに、接続プールマネージャによって作成されます。

SAP 接続リソースの [Connection Pooling] オプションとして [Pooling] を選択した場合、[Advanced] ボタンによって [Pool Info] ダイアログボックスが表示されます。[Pool Info] ダイアログボックスでは、接続プールの最大初期プールサイズを設定できます。



SAP コンソールから接続プールを管理します。



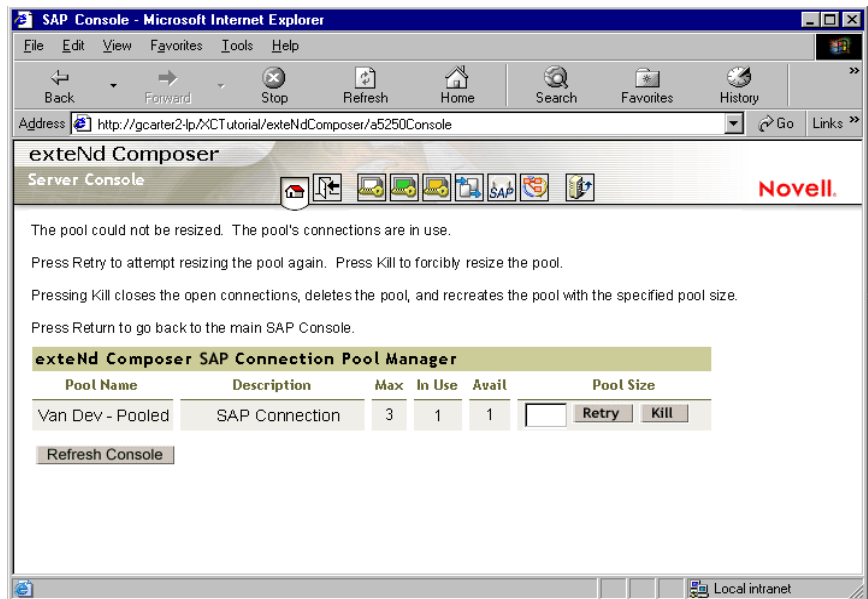
SAP コンソールでは、接続プールの管理に対する次の情報および通信を提供します。

- ◆ **[Pool Name]** - 完全な SAP 接続プール名
- ◆ **[In Use]** - 現在アプリケーションに割り当てられているプールの接続数
- ◆ **[Max Used]** - 使用されている接続のピーク数
- ◆ **[Available]** - 現在空いているプールの接続数
- ◆ **[Pool Size]** - プールの最大サイズ。このフィールドは、プールをリセットするときに編集できます。編集するには、新しいプールサイズを入力して、**[Apply]** を押します。

注記： プール接続 **[In Use]** は、アプリケーションに必要な数よりも必ず 1 つ多くなります。これは、リポジトリの接続に 1 つのプール接続が必要だからです。

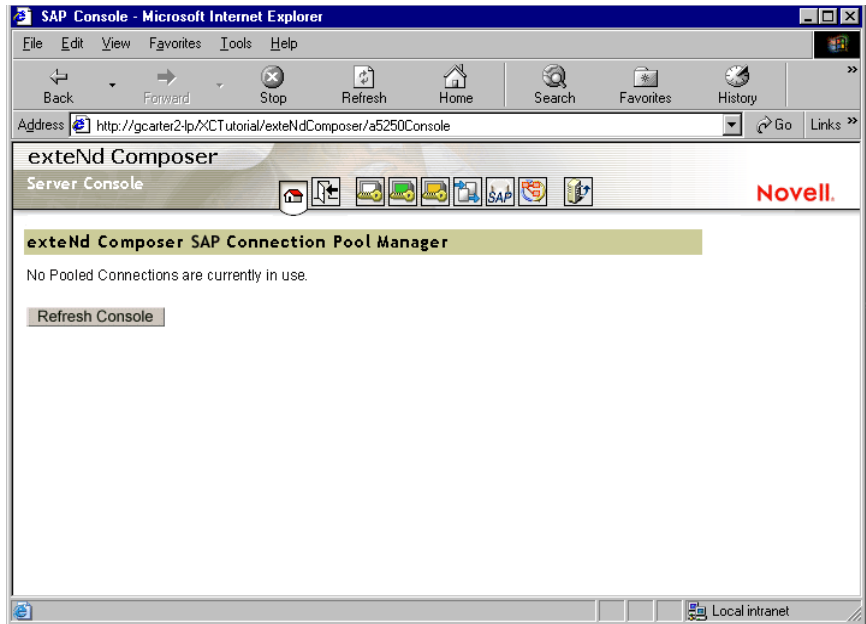
[Refresh Console] ボタンを押すと、表示されているプール情報を更新できます。

プールのサイズを変更しようとしたときに、接続が使用中の場合、次のような **[SAP Console]** ページが表示されます。プールのサイズ変更を再び行うには、**[Retry]** ボタンを押します。また、**[Kill]** ボタンを押して、プールのサイズを強制的に変更することもできます。強制的にサイズを変更すると、開いている接続が閉じ、プールが削除され、指定したサイズで再び作成されます。**[Return]** ボタンを押して、SAP のメインコンソールに戻ることもできます。



[Refresh] ボタンを押すと、表示されているプール情報を更新できます。

デフォルトでは、プールが使用されていないときに次の SAP コンソールが表示されます。



コンポーネントで接続が要求されたときに、プールが空の場合、要求は 30 秒後にタイムアウトになり、例外がスローされます。

設計時には、プールを定義する接続を編集して、接続をテストすると、プールを削除できます。また、接続を編集して、[Finish] を押しても、接続プールを削除できます。いずれの場合でも、プールは強制的に削除されます。

コンポーネント実行により作成されたプールには、[Pool Info] ダイアログボックスで設定されたプールサイズを使用する [Pool Size] があります。

コンポーネントに対する XML テンプレートの作成

接続リソースの他に、SAP コンポーネントでは、コンポーネント設計にサンプルコンポーネントを使用できるように、XML テンプレートをすでに作成している必要があります (詳細については、『Composer ユーザガイド』の第 5 章「XML テンプレートの作成」を参照してください)。

また、コンポーネント設計によって別の xObject リソース (カスタムスクリプトやコードテーブルマップなど) が要求される場合は、SAP コンポーネントを作成する前にこれらのリソースを作成することをお勧めします。詳細については、『Composer ユーザガイド』の「カスタムスクリプトの作成」を参照してください。

3

SAP コンポーネントの作成

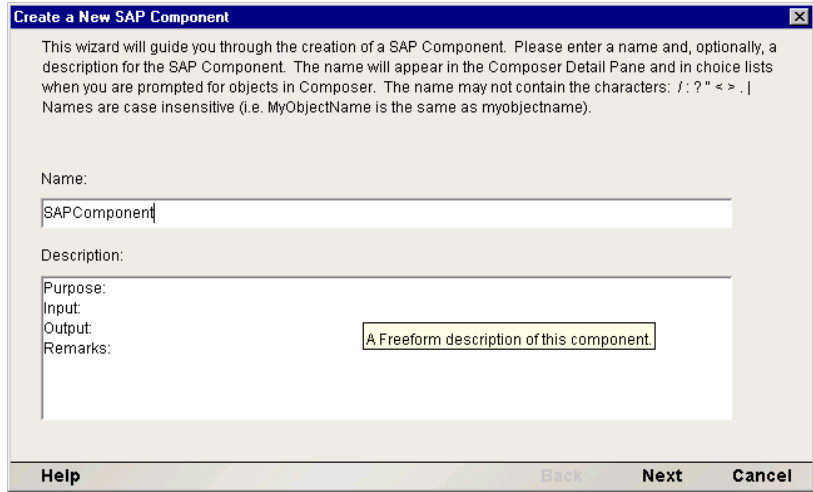
SAP コンポーネントを作成する前に

すべての Composer コンポーネントと同様に、SAP コンポーネントを作成する最初の手順は、必要な XML テンプレートを指定することです（詳細については、『Composer ユーザガイド』の「新しい XML テンプレートの作成」を参照してください）。XML テンプレートを指定すると、コンポーネントによって処理される入力および出力を表すテンプレートのサンプルドキュメントを使用して、コンポーネントを作成できます。

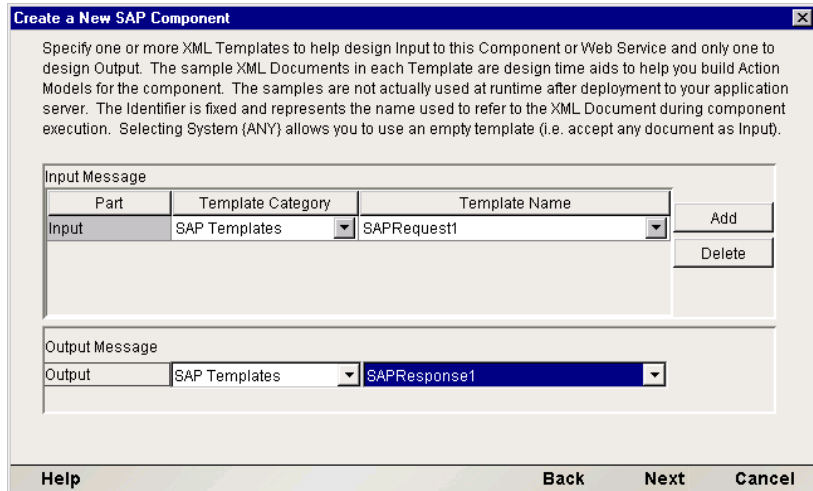
また、SAP コンポーネントの作成プロセスの一部として、SAP 接続を選択したり、新しい接続を作成したりする必要があります。接続は、事前に作成した場合、新しい SAP コンポーネントすべてに対して選択できるようになります。

➤ 新しい SAP コンポーネントを作成する

- 1 [File] > [New xObject] > [Component] > [SAP] の順に選択します。「Create a New SAP Component」ウィザードが表示されます。



- 2 新しい SAP コンポーネントの「名前」を入力します。
- 3 または、「説明」テキストを入力します。
- 4 [Next] をクリックします。New SAP Component ウィザードの XML テンプレートパネルが表示されます。



- 5 1つまたは複数の入力テンプレートを、次のように指定します。
 - ◆ デフォルトのカテゴリと異なる場合は、「**テンプレートカテゴリ**」を選択します。

- ◆ 選択した [Template Category] にある XML テンプレートのリストから [Template Name] を選択します。
 - ◆ 入力 XML テンプレートをさらに追加するには、[Add] をクリックして、手順 2 から 4 を繰り返します。
 - ◆ 入力 XML テンプレートを削除するには、エントリを選択して [Delete] をクリックします。
- 6 出力として XML テンプレートを選択します (名前が「Output」という出力 DOM が「1 つ」だけあることに注意してください)。
- 注記:** 出力テンプレートとして {System}{ANY} を選択すると、構造が含まれない出力 XML テンプレートを指定できます。たとえば、[Result Mapping] タブの [Custom] オプションを使用して最適化した出力 DOM を生成する場合、このように指定します (詳細については、『Composer ユーザガイド』の「テンプレートを 사용하지 않는 출력 DOM 의 作成」で XML Map コンポーネント의 作成에 関する 章을 参照してください)。
- 7 [Next] をクリックします。「Create a New SAP Component」ウィザード의 接続 情報 패널이 表示 されます。

The screenshot shows a dialog box titled "Create a New Connection Resource". It contains the following text and fields:

Enter the information provided by the SAP admin including: a Host name or an IP Address (10.122.321.1), System Number (99), SAP Client Number (999), language code of the SAP system (EN), and valid User name and Password. You can take advantage of Pooled connections on the server by setting the Connection Pooling Parameter to Session or Pooled. The default is Non-Pooled. The Advanced button is used to set pool parameters. Use the Test button to check your connection. You may save connections that fail the test.

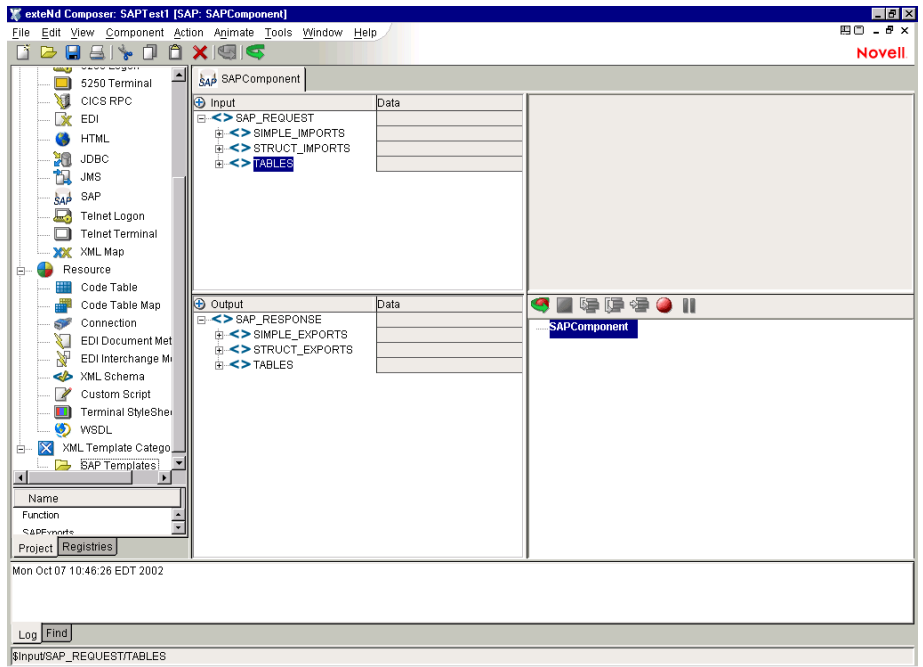
Fields and values:

- Connection Type: SAP Connection
- Host or IP Address: 172.168.0.128
- System Number: 00
- Client Number: 800
- Language: EN
- User ID: guest
- Password: ****
- Connection Pooling: Non-Pooled

Buttons: Test, Default, Advanced...

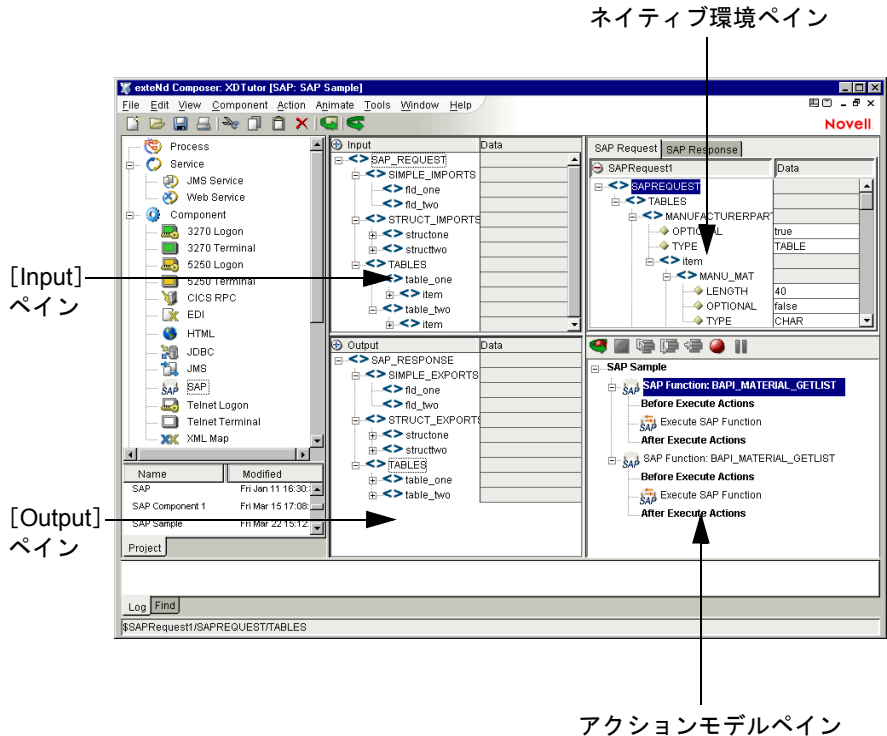
Footer: Help, Back, Finish, Cancel

- 8 プルダウンリストで「接続」タイプを選択します。SAP 接続の詳細については、「SAP 接続リソースの作成」を参照してください。
- 9 [Finish] をクリックします。コンポーネントが作成され、SAP コンポーネントエディタが表示されます。



SAP コンポーネントエディタウィンドウについて

SAP コンポーネントエディタには、XML Map コンポーネントエディタの機能がすべて含まれています。また、入力 XML ドキュメントと出力 XML ドキュメントのマッピングペインや、アクションモデルペインも含まれています。

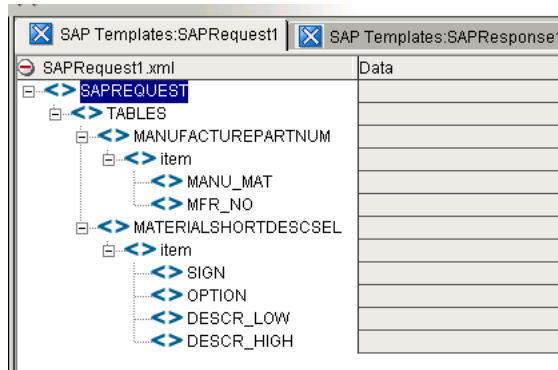


ネイティブ環境ペインについて

SAP コンポーネントエディタのネイティブ環境ペイン (最初はグレーです) には、SAP Function アクションがアクションモデルペインで選択されていると、[SAP Request] と [SAP Response] の 2 つのタブが表示されます。[SAP Request] タブと [SAP Response] タブは、それぞれ、SAP Function Action の [SAP Request] タブと [SAP Response] タブで [Use Mapping Model] チェックボックスがオンになっていると (デフォルト)、表示されます。したがって、SAP アクションには、[Request] タブだけがある場合、[Response] タブだけがある場合、または [Request] タブと [Response] タブの両方がある場合、あるいはどちらもない場合があります。

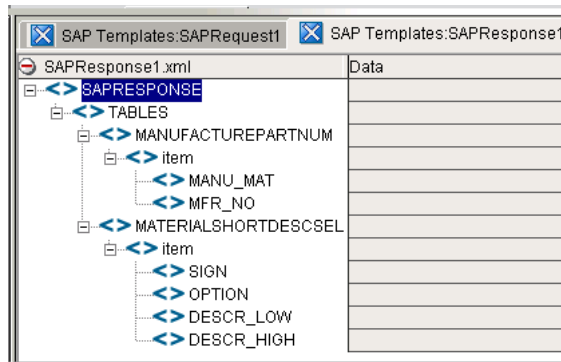
NEP の [SAP Request] タブ

ネイティブ環境ペインを初めて開くと、[SAP Request] タブが表示されます。[SAP Request] タブでは、入力 XML ドキュメント (またはその他に使用できる DOM) からデータをマップして、実行前の SAP 関数の要求に対する入力として使用できます。たとえば、入力 DOM から [SAP Request] の [CUSTOMID] フィールドに顧客 ID 番号をドラッグできます。SAP 関数を実行すると、顧客 ID に関連するデータが返されます。



NEP の [SAP Response] タブ

NEP の [SAP Response] タブでは、実行した SAP 関数の応答からデータをマップできます。データは、SAP 関数の [Response] フィールドから出力 XML ドキュメント (またはその他に使用できる Temp や MyDom など) にマップされます。



4

SAP アクションの実行

アクションについて

「アクション」は、プログラミングステートメントに類似しており、パラメータの形式で入力を受け付け、特定のタスクを実行します。『Composer ユーザガイド』のアクションに関する章を参照してください。

SAP コンポーネントエディタ内では、XML ドキュメントを処理したり、非 XML データソースと通信したりするための命令のセットが、「アクションモデル」の一部として作成されます。アクションモデルは、データベースと XML ドキュメント間でのすべてのデータマッピング、データ変換、データ転送、およびコンポーネントとサービス内でのデータ転送を実行します。

アクションモデルは、アクションのリストから構成されています。アクションモデル内のすべてのアクションは相互に機能します。たとえば、あるアクションモデルでは、請求書のデータをディスクから読み取り、データを在庫データベースから取得し、一時 XML ドキュメントに結果をマップして変換し、変換されたデータを出力 XML ドキュメントにマップする個々のアクションを含みます。

このアクションモデルの例は、いくつかの個々のアクションから構成されています。そのアクションは次のとおりです。

- ◆ ドキュメントを開き、SAP Function アクションを実行して、SAP データベースからデータを取得する
- ◆ 結果を一時 XML ドキュメントにマップする
- ◆ コードテーブルを使用して数値コードを変換し、結果を出力 XML ドキュメントにマップする

Composer SAP Connect では、Composer およびランタイム環境からの SAP の BAPI およびその他の RFC 可能な関数モジュール (RFM) への着信、発信呼び出しが可能です。Composer SAP Connect は、Java コネクタライブラリを使用して SAP の BAPI および RFM を呼び出します。

SAP Function アクション

SAP コンポーネントエディタには、exteNd Composer の XML Map コンポーネントエディタの重要な機能がすべて含まれています。SAP Function アクションのダイアログボックスは、[SAP Function]、[SAP Request]、および [SAP Response] の 3 つのタブで構成されています。

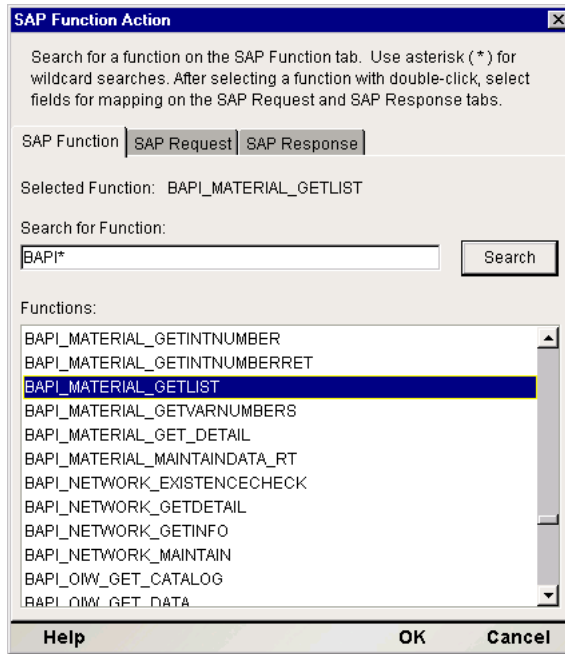
[SAP Function] タブ

[SAP Function] タブでは、アクション名が検証され、現在の接続先の SAP システムにこの関数が存在することが確認されます。関数名が有効な場合、[Function] テキストフィールドの [Search] に表示され、[Function] リストで選択できます。

入力された関数名が無効な場合、エントリ「No function found」が [Function] リストに表示されます。検証中に、通信エラーなど、別のタイプのエラーが発生した場合、エントリ「Error searching for function」が [Function] リストに表示されます。[SAP Request] タブおよび [SAP Response] タブは、[Function] リストで有効な関数名をダブルクリックすると、使用できるようになります。

➤ SAP Function アクションを追加する

- 1 アクションメニューから、[New Action] > [SAP Function] または RMB をクリックして、[Map Action pane] ペインで [New Action] > [SAP Function] をクリックします。
- 2 ダイアログボックスに表示される最初のタブは、[SAP Function] タブです。ダイアログボックスが表示されると、[SAP Request] タブと [SAP Response] タブは使用できなくなり、[Functions] リストは空になります。



- 3 [Search] ボタンをクリックして、関数の完全なリスト (次を参照) を取得します。検索結果にフィルタをかけるには、検索情報を指定します。指定するには、RFC* のように SAP 関数プリフィックスの後にワイルドカード文字 (*) を入力して、SAP 関数のタイプを検索します。検索文字列を入力した後、[Search] ボタンをクリックします。検索条件に一致する関数のリストが表示されます。スクロールバーを使用して関数を探し、ダブルクリックします。ワイルドカード文字は、リテラル文字列の任意の部分に使用できます (*FC、*FC*、RFC*)。

検索条件に一致する関数がない場合、1 つのエントリ「No function(s) found」が [Function] リストに表示されます。空の文字列を入力すると、検索では、デフォルトでワイルドカード文字が使用されます。

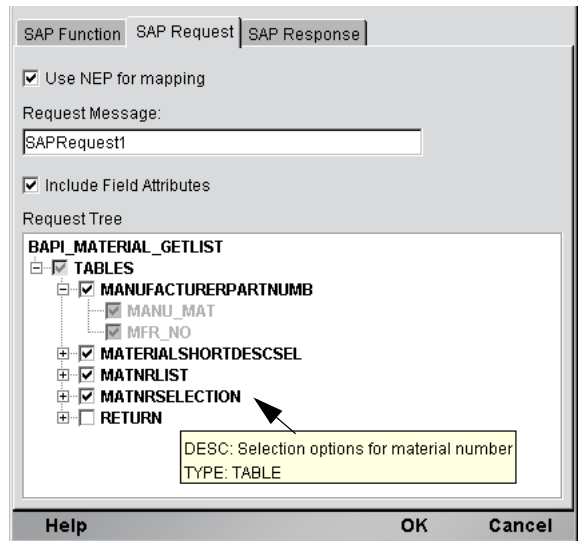
検索が成功すると、検索文字列が保存されます。保存された検索文字列は、その後ユーザが SAP Function アクションを作成するときに、関数のリストのフィルタとして使用されます。

検索が成功すると、ユーザは、ダブルクリックして SAP 関数を選択できます。SAP 関数を選択すると、関数ラベル「Selected」が設定され、[SAP Request] タブおよび [SAP Response] タブが使用可能になり、表示および編集できるようになります。

- 4 [SAP Request] タブをクリックします。

デフォルトでオンに
なっています →

デフォルトでオンに
なっています →



デフォルトでは、[Use NEP for mapping] チェックボックスはオンになっています。入力 XML ドキュメントからネイティブ環境ペイン内に表示される [SAP Request] フィールドにデータをマップする場合、マッピングオプションに [USE NEP] を選択します。[Use NEP for mapping] オプションを選択すると、[SAP Request] タブでその他すべてのコントロールが使用できるようになります。

入力 XML ドキュメントから [SAP Request] フィールドにデータをマップしない場合、チェックボックスをオフにして、[Use NEP for mapping] オプションの選択を解除します。この場合、ネイティブ環境ペインには [SAP Request] タブは表示されないため、Expression Editor で使用できるパブリッシュメソッドを使用する必要があります (以下の「Expression Editor」を参照)。[Use NEP for mapping] オプションが選択されている場合でも、Expression Editor のパブリッシュメソッドを使用できます。

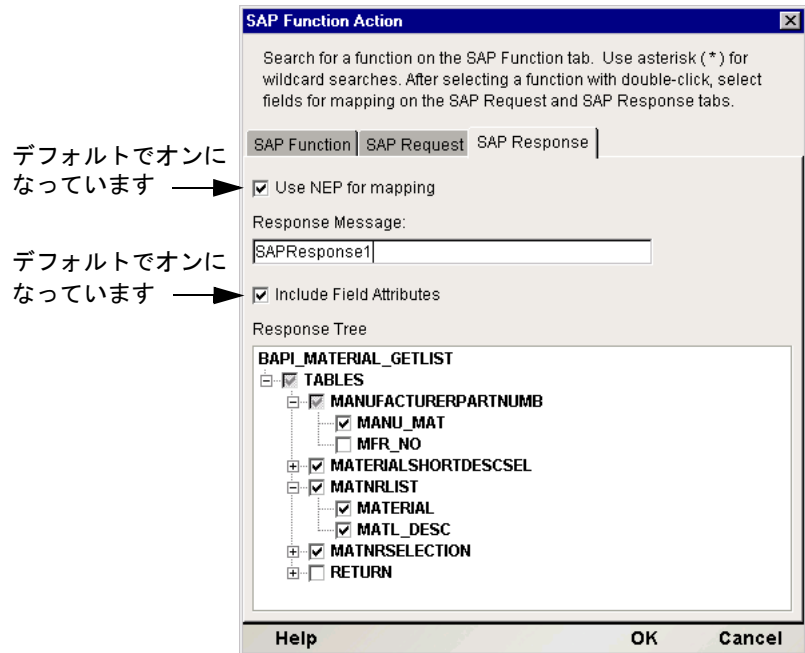
フィールド属性を追加する場合、[Include Field Attributes] チェックボックスをオンにする必要があります。[Use NEP for mapping] オプションおよび [Include Field Attributes] オプションの両方が選択されている場合、NEP の [SAP Request] タブ内に表示される SAP Request ドキュメントには、フィールド、構造、テーブルのそれぞれについてのメタデータが属性として含まれます。

[SAP Function] ダイアログボックスの [SAP Request Tree] では、デフォルトで事前を選択され、使用できない必須ノードが表示されます。ただし、必須でないノードは選択したり、選択を解除したりできます。

[SAP Request Tree] 内では、いくつかの (すべてではない) 必須ではないチャイルドノードが選択されている場合、ペアレントノードは選択された状態で、グレーの背景色で表示されます。必須ではないチャイルドコードが選択されていない場合、ペアレントノードは選択された状態で、通常の白い背景色で表示されます。

フィールド名にカーソルを置くと、そのフィールドの説明が表示されます。この説明は、ツールチップ内に表示されます (図を参照)。

5 [SAP Response] タブをクリックします。



デフォルトでは、[Use NEP mapping] チェックボックスはオンになっています。ネイティブ環境ペイン内に表示される [SAP Response] フィールドから出力 XML ドキュメントにデータをマップする場合、[Use NEP mapping] オプションを選択します。[Use NEP mapping] オプションを選択すると、[SAP Response] タブでその他すべてのコントロールが使用できるようになります。

ネイティブ環境ペイン内に表示される [SAP Response] フィールドから出力 XML ドキュメントにデータをマップしない場合、チェックボックスをオフにして [Use NEP mapping] オプションの選択を解除します。この場合、ネイティブ環境ペインには [SAP Response] タブは表示されないため、Expression Editor で使用できるパブリッシュメソッドを使用する必要があります (後の Expression Editor に関する節を参照)。**[Use NEP mapping]** オプションが選択されている場合でも、Expression Editor のパブリッシュメソッドを使用できます。

フィールド属性を追加する場合、**[Include Field Attributes]** チェックボックスをオンにする必要があります。**[Use NEP mapping]** オプションおよび **[Include Field Attributes]** オプションの両方が選択されている場合、NEP の [SAP Response] タブ内に表示される SAP Response ドキュメントには、フィールド、構造、テーブルのそれぞれについてのメタデータが属性として含まれます。

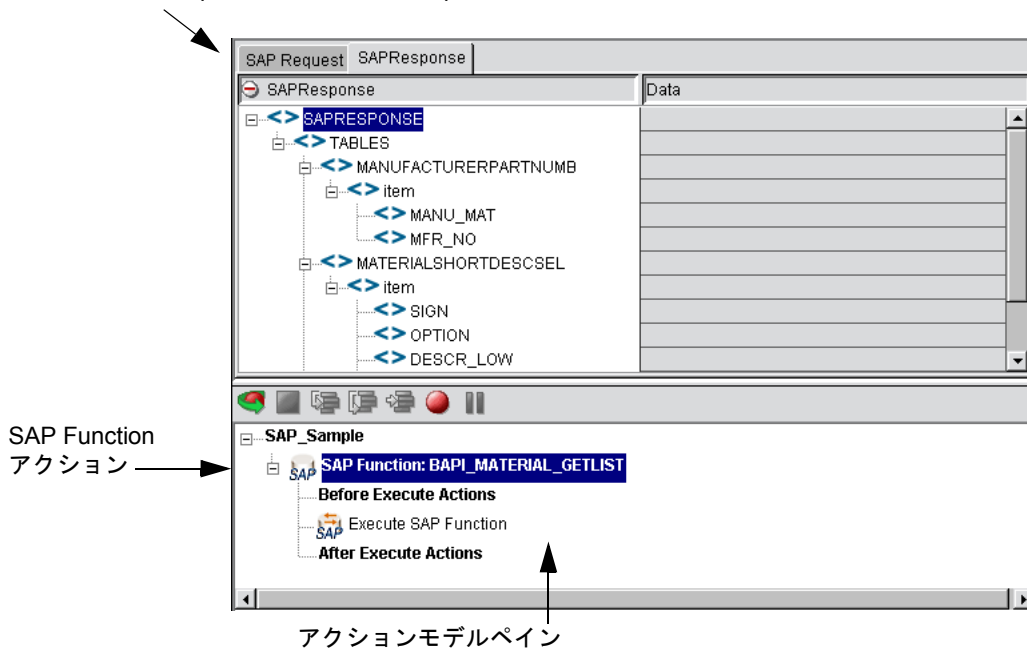
[SAP Function] ダイアログボックスの [SAP Response Tree] では、デフォルトで事前に選択され、使用できない必須ノードが表示されます。ただし、必須でないノードは選択したり、選択を解除したりできます。

[SAP Response Tree] 内では、いくつかの (すべてではない) 必須ではないチャイルドノードが選択されている場合、ペアレントノードは選択された状態で、グレーの背景色で表示されます。必須ではないチャイルドノードが選択されていない場合、ペアレントノードは選択された状態で、通常の白い背景色で表示されます。

フィールド名にカーソルを置くと、そのフィールドの説明が表示されます。

- 6 [SAP Function] アクションダイアログボックスの **[OK]** ボタンをクリックすると、エントリが検証され、アクションが保存されます。**[Use Mapping Model]** オプションが選択されている場合、[SAP Request] タブおよび [SAP Response] タブがネイティブ環境ペインに表示され、新しい SAP Function アクションがアクションモデルペインで自動的に作成されます。

NEP の[SAP Request]タブと[SAP Response]タブ



SAP Function アクション構造

アクションモデルで作成される構造は、次の4つの行で構成されます。

- 1 [SAP Function] - 選択された関数名。
- 2 [Before Execute Actions] - [SAP Request] のマッピングモデルドキュメントにデータをマップしたり、SAP 関数にデータを直接設定したりできます。
- 3 [Execute SAP Function] - [SAP Request] のマッピングモデルデータを関数に設定し、関数を実行し、[SAP Response] のマッピングモデルデータを関数から取得します。
- 4 [After Execute Actions] - [SAP Response] のマッピングモデルドキュメントからデータをマップしたり、SAP 関数からデータを直接取得したりできます。

SAP Function アクションの編集

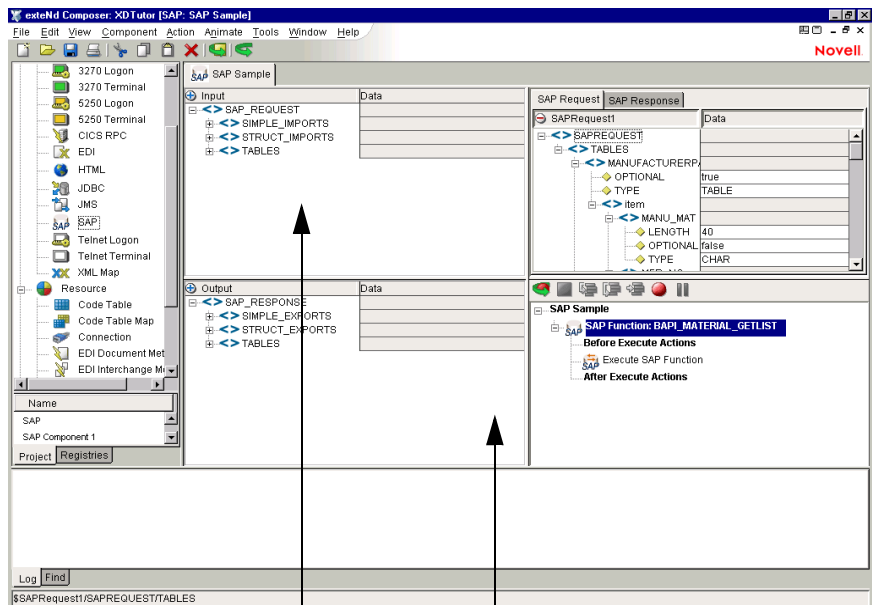
次の操作を実行して、既存の SAP Function アクションを編集します。

➤ SAP Function アクションを編集する

- 1 アクションのルートノードを選択し、マウスの右ボタンをクリックして [Edit] を選択するか、アクションのルートノードをダブルクリックします。
- 2 [Request] タブおよび [Response] タブを変更します。
- 3 [OK] をクリックして、変更内容を保存します。

SAP アクションモデル

SAP コンポーネントは、アクションモデル内の SAP Function アクション構造が自動的に作成される点で、他のコンポーネントとは異なります。この構造は、SAP 関数をアクションモデルに追加するときに作成されます。このガイドの第 3 章「SAP コンポーネントの作成」の手順に従って、SAP コンポーネントを作成できます。この例で示す SAP コンポーネントの作成では、SAP Request テンプレートおよび SAP Response テンプレートが入力および出力にそれぞれ選択されています。新しく作成した SAP コンポーネントは、[SAP Component Editor] ウィンドウに表示されます。



SAP Request
テンプレート

SAP Response
テンプレート

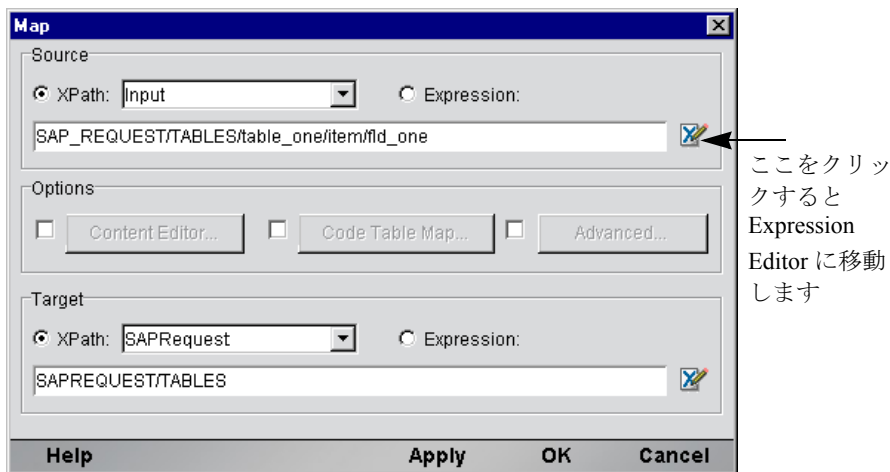
Before Execute Actions

[Before Execute Actions] - [SAP Request] のマッピングモデルドキュメントにデータをマップしたり、SAP 関数にデータを直接設定したりできます。この方法は、次の節で説明するようにいくつかあります。

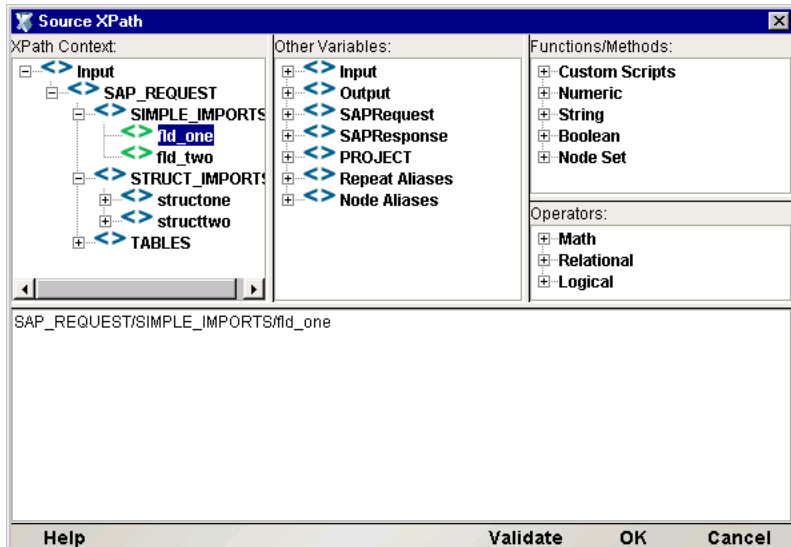
[SAP Request] タブへの XML ドキュメントの一部のマップ

XML ドキュメントの一部を [SAP Request] にマップするには、ネイティブ環境ペインの空の領域を ([SAP Request] タブが選択されている状態で) 右クリックします。コンテキストメニューが表示されます。

[Map...] コマンドを選択します。[Map] ダイアログボックスが表示されます。次の手順では、[Map] ダイアログボックスの [Source] フィールドと [Target] フィールドが初期化されます。

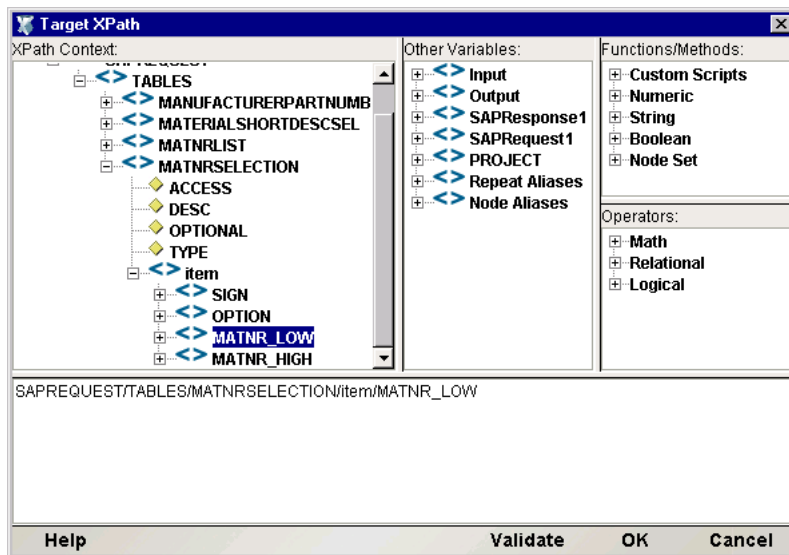


[Map] ダイアログボックスでは、[Input] が [Source] のデフォルトの DOM として表示され、[SAPRequest] が [Target] のデフォルトとして表示されます (プルダウンメニューを使用して、[Source] および [Target] で別の DOM を選択できます)。ソースとして使用する XPath フラグメントが分かっている場合、提供されているボックスに入力します。分からない場合、右側の青い Expression Editor アイコンをクリックします。Expression Editor アイコンをクリックすると、[Source] の [Expression Editor] ダイアログボックスが表示されます。

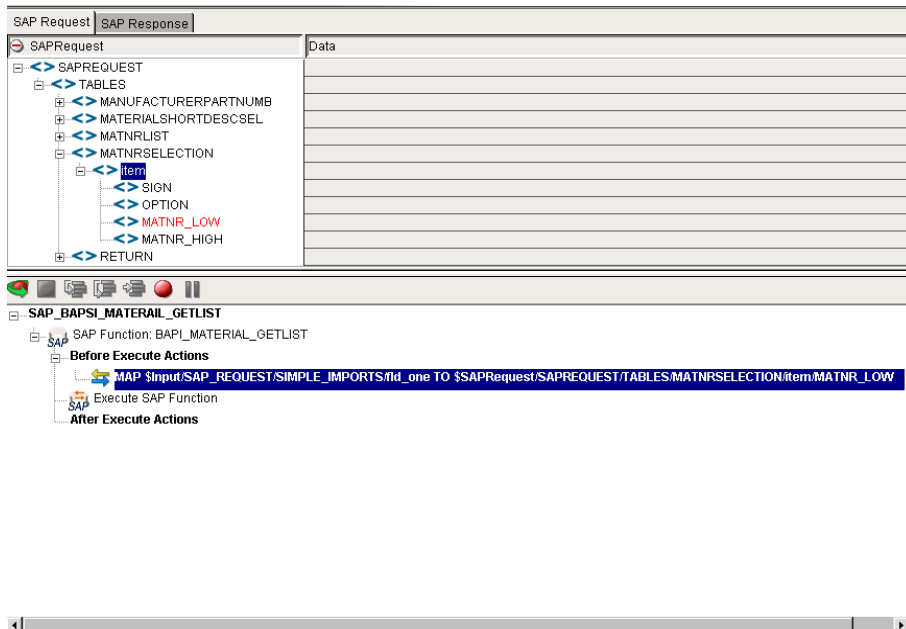


このダイアログボックスの上部にある選択リストを使用すると、ポイントしてクリックするだけで、XPath フラグメントまたは ECMAScript、あるいはその両方を作成できます。この場合、入力 DOM (左上) のツリービューを展開して、完全な入力ツリー構造を表示します。ツリーの項目をダブルクリックすると、ダイアログボックスの下部にその項目 (ツリーのその部分に対する XPath フラグメント) が自動的に表示されます。[OK] をクリックすると、[Map] ダイアログボックスの該当する箇所に XPath 情報が表示されます。

入力からメッセージ本文の MATNR_LOW の XPath 位置に情報をマップするには、[Map] ダイアログボックスの [Target] の部分に入力します。



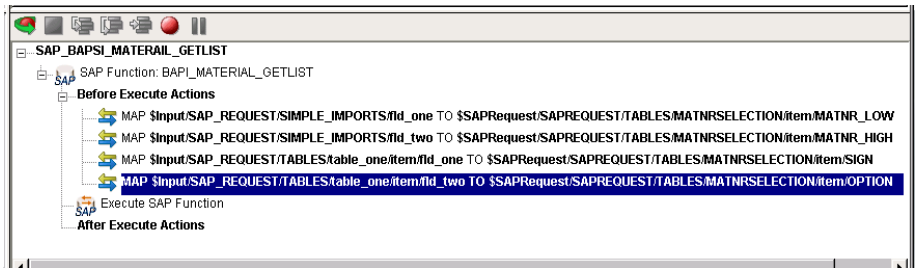
[OK] をクリックすると、[Map] ダイアログボックスが閉じ、SAP コンポーネントエディタのメインウィンドウでマップの結果を確認できるようになります。



Map アクションの追加

この手順は、メッセージ本文にデータを入力するたびに繰り返すことができます。または、Function アクションを (ECMAScript DOM メソッドと同時に) 使用して、プログラムによりメッセージ本文に XML ノードを作成することもできます。

「ドラッグアンドドロップ」を使用して、入力 DOM の一部から [SAP Request Document] (以下で説明する制約に従って) に、またはその逆にデータを直接マップできます。表示可能な DOM ペインで入力ノードをクリックしたまま、[SAP Request Document] の目的の位置までドラッグして、マウスボタンを放します。該当する Map アクションがアクションモデルに自動的に追加されます。



マップの制約



ノードは [SAP Request] タブの任意の分岐またはフィールドにドロップできます。ただし、ノードを属性にドロップすることはできません。ソースの値が、ターゲットフィールドノードのタイプに対して検証されない場合、「forbidden drag operation」記号が、マウスポインタとして左側に表示され、メッセージ「Invalid drop target」がステータスエリアに表示されます。ソースの値が、ターゲットフィールドノードの長さに対して長過ぎる場合、「forbidden drag operation」記号が表示され、メッセージ「Invalid drag length for this drop target」がステータスエリアに表示されます。

SAP Function アクションの実行

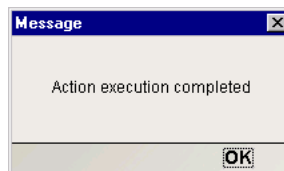
SAP Function アクションにより SAP Request DOM (存在する場合) が実行されると、SAP Request DOM がクリアされ、SAP Response DOM (存在する場合) が再ロードされます。次に、[Before Execute Actions] リストが実行されます。さらに、Execute SAP Function アクションが実行されます。ここでは、SAP Request マッピングモデルドキュメントマッピングが関数に設定され、その関数が実行され、SAP Response マッピングモデルドキュメントに関数の結果が設定されます。

SAP Function アクションのルートがアクションモデルの現在の (または選択されている) アクションの場合、メインツールバーの [Execute SAP Function] ボタンが使用できます。これ以外の場合、このボタンは使用できません。[SAP Execute Function] ボタンをクリックすると、SAP システムに接続され、SAP Function アクション内でアクションが実行されます。実行されると、SAP システムから切断されます。



The screenshot displays the SAP GUI interface during the execution of the BAPI_MATERIAL_GETLIST function. It is divided into several panels:

- Input:** Shows the function parameters. The 'TABLES' table is expanded, showing 'table_one' and 'table_two' with their respective fields.
- Output:** Shows the function results. The 'MATNRLIST' table is expanded, listing materials such as 'AM2-100 Shell', 'AM2-200 PRIMER', 'AM2-205 FILLER', and 'AM2-210 COLOR WHITE'.
- SAP Request / SAP Response:** Shows the internal data structures and tables used by the function.
- SAP Sample:** Shows the execution steps, including 'Before Execute Actions', 'Execute SAP Function', and 'After Execute Actions'.



実行が成功すると、メッセージダイアログボックスが表示されます。

[After Execute Actions] - [Response] タブからの Map アクションの追加

[After Execute Actions] : SAP Response マッピングモデルドキュメントからデータをマップしたり、SAP 関数からデータを直接取得したりできます。

The screenshot shows the exteNd Composer XDI Tutor interface with the following components:

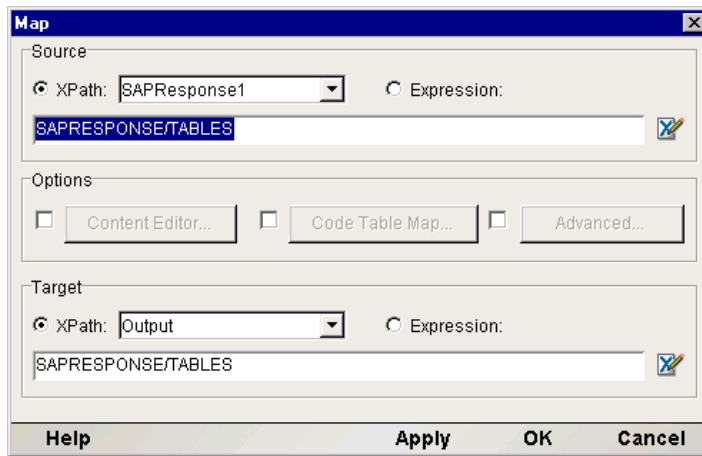
- Input Data:** A tree structure showing the SAP Request structure: SAP_REQUEST (SIMPLE_IMPORTS, STRUCT_IMPORTS, TABLES) and their respective data values.
- Output Data:** A tree structure showing the SAP Response structure: SAP_RESPONSE (TABLES, MATNRLIST) and their respective data values.
- Mapping Configuration:** A list of MAP actions under the 'After Execute Actions' section, showing the mapping from input nodes to output nodes.
- Sample Data:** A table showing the sample data for the SAP Response, including material names and descriptions.

SAP Response	Data
MATERIAL	AM2-100
MATL_DE8C	Shell
MATERIAL	AM2-200
MATL_DE8C	PRIMER
MATERIAL	AM2-205
MATL_DE8C	FILLER
MATERIAL	AM2-210
MATL_DE8C	COLOR WHITE
MATERIAL	AM2-220
MATL_DE8C	UNICOLOR BRIGHT RED

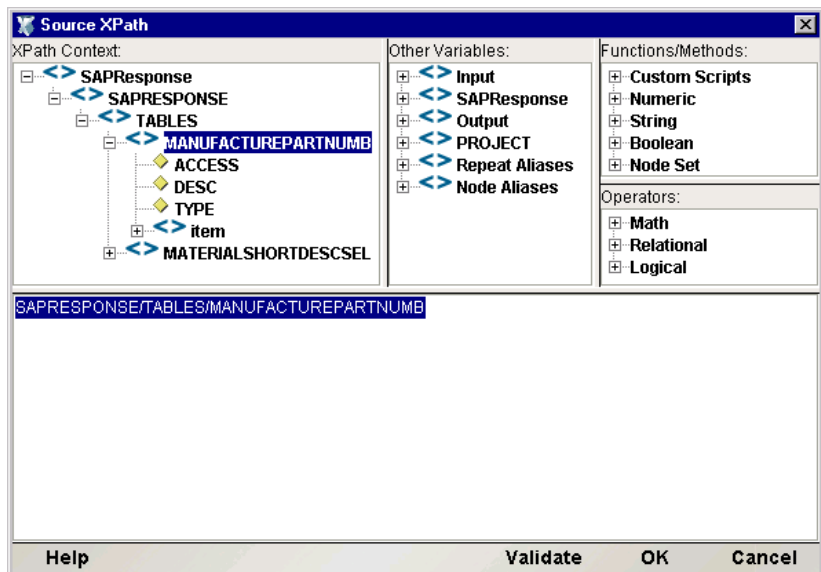
XML ドキュメントへの SAP Response の一部のマップ

[SAP Response] の一部を XML ドキュメントにマップするには、ネイティブ環境ペインの空の領域を ([SAP Response] タブが選択されている状態で) 右クリックします。コンテキストメニューが表示されます。

[Map...] コマンドを選択します。[Map] ダイアログボックスが表示されます。



[Map] ダイアログボックスでは、[SAP Response] が [Source] のデフォルトとして表示され、[Output] が [Target] のデフォルトの DOM として表示されます (プルダウンメニューを使用して、[Source] および [Target] で別の DOM を選択できます)。ソースとして使用する XPath フラグメントが分かっている場合、提供されているフィールドに入力します。分からない場合、右側の青い Expression Editor アイコンをクリックします。Expression Editor アイコンをクリックすると、[Source] の [Expression Editor] ダイアログボックスが表示されます。



このダイアログボックスの上部にある選択リストを使用すると、ポイントしてクリックするだけで、XPath フラグメントまたは ECMAScript、あるいはその両方を作成できます。この場合、[SAP Response Document] (左上) のツリービューを展開して、完全なツリー構造を表示します。ツリーの項目をダブルクリックすると、ダイアログボックスの下部にその項目 (ツリーのその部分に対する XPath フラグメント) が自動的に表示されます。[OK] をクリックすると、[Map] ダイアログボックスの該当する箇所に XPath 情報が表示されます。

Map アクションの追加

この手順は、出力 DOM にデータを入力するたびに繰り返すことができます。または、Function アクションを (ECMAScript DOM メソッドと同時に) 使用して、プログラムにより出力 DOM に XML ノードを作成することもできます。

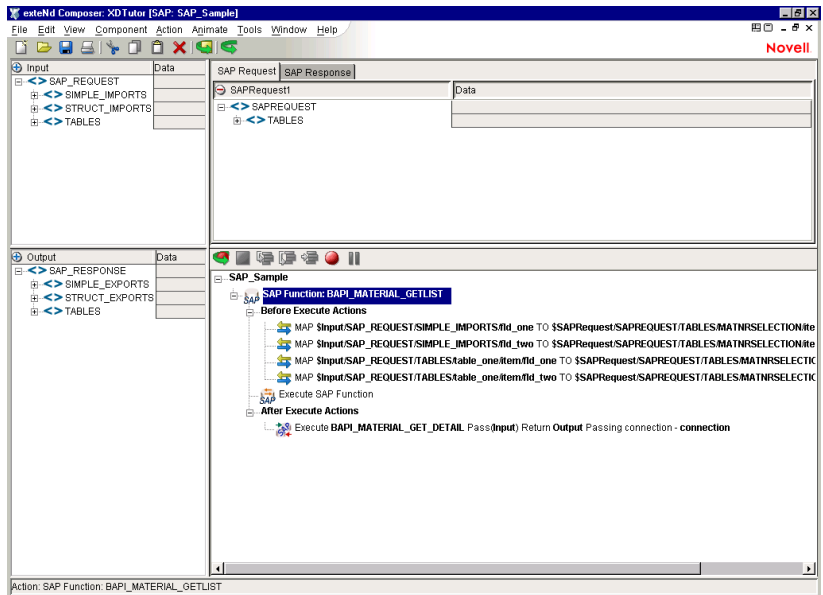
「ドラッグアンドドロップ」を使用して、[SAP Response Document] の一部から出力 DOM にデータを直接マップできます。[SAP Response Document] の任意のノードをクリックしたまま、出力 DOM の目的の位置までドラッグして、マウスボタンを放します。該当する Map アクションがアクションモデルに追加されます。

Component with Connection アクション

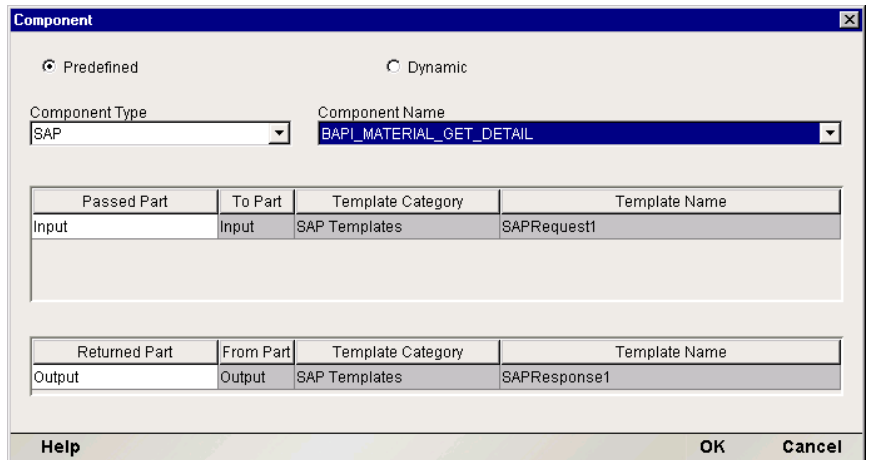
Component with Connection アクションでは、SAP コンポーネントから別のコンポーネントを呼び出し、同じ接続を共有できるので、このアクションは特有なアクションです。このアクションを使用すると、大きなコンポーネントをメイン SAP コンポーネントとサブコンポーネントに分けることができるので、アクションモデルを簡単に管理できます。メインコンポーネントとサブコンポーネントで接続を共有することにより、接続オーバーヘッドが大幅に削減されるので、実行時のパフォーマンスが向上します。このコマンドは、セッション接続を指定するときに特に役に立ちます。

➤ Component with Connection アクションを使用する

- 1 サブコンポーネントの呼び出しが可能な程度のメインコンポーネントの基本構造を作成および記録します。たとえば、サブコンポーネントには、「BAPI_MATERIAL_GET_DETAIL」という名前を付けます。

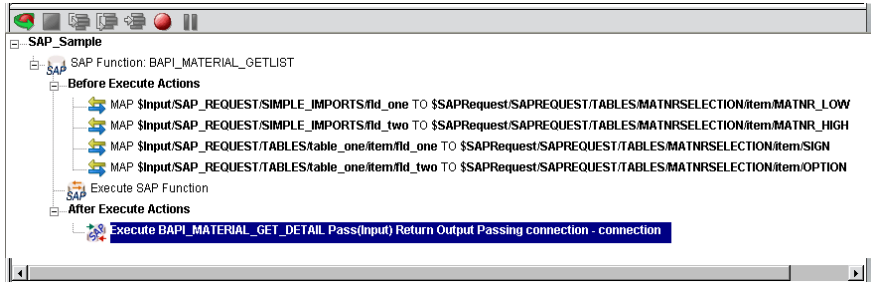


- 2 [Main] メニューから、またはマウスの右ボタンをクリックして、[New Action] > [Component/w connection] の順に選択します。次のダイアログボックスが表示されます。



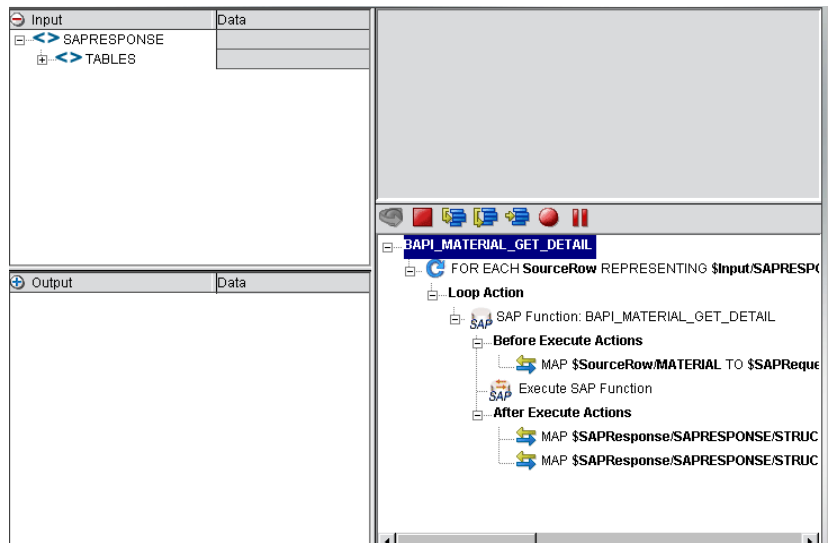
- 3 [Component Type] プルダウンリストから、コンポーネントタイプの名前を選択します。[Component Name] プルダウンリストから、コンポーネントの名前を選択します。

- 4 変更する必要がある場合に渡される ID をプルダウンリストから選択します。変更する必要がある場合に返される ID をプルダウンリストから選択します。[OK] をクリックします。
- 5 すると、アクションがマップペインに表示されます。

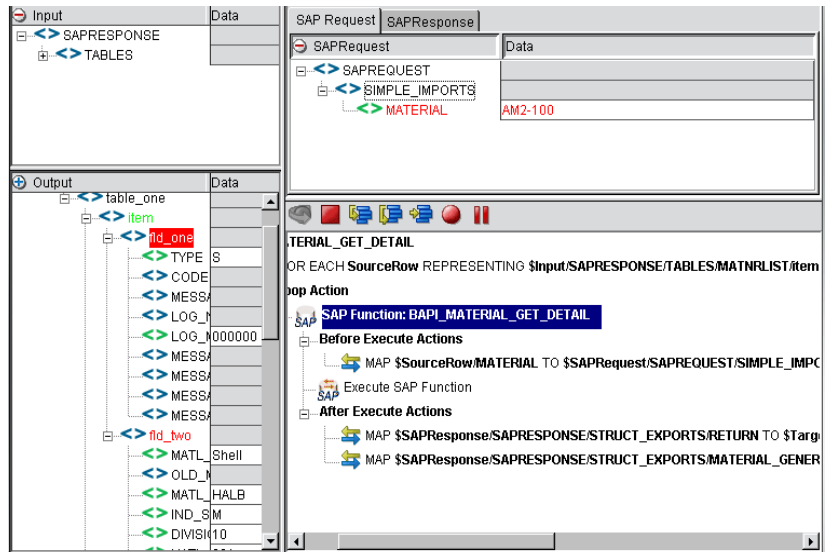


- 6 メインコンポーネントのアニメーションを実行して、Component with Connection アクションをステップインします。サブコンポーネントが開きます。画面が「BAPI MATERIAL GET DETAIL」という名前のコンポーネントにどのように変わるか確認します。
- 7 コンポーネントを作成したように、サブコンポーネントアクションモデルを作成します。

注記： この例の Repeat for Element アクションは、複数行を処理するように作成されています。

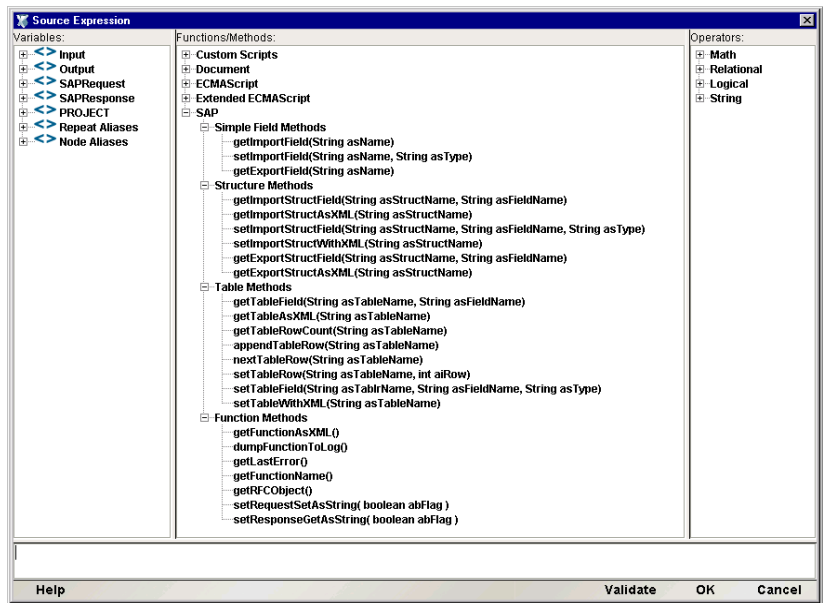


- 8 サブコンポーネントのアニメーションに実行して、ステップインを実行します。サブコンポーネントの結果が、[Output DOM] タブおよび [SAP Response] タブに表示されます。
- 9 コンポーネントおよびサブコンポーネントアクションモデルを保存します。



SAP 専用 Expression Builder プロパティ

SAP アクションモデルを作成するには、Expression Builder を使用して、データを SAP 関数に直接設定します。通常、これらの値を ECMAScript 式で反映できるようにすると便利です。Expression Builder 選択リスト ([Expression Editor] ウィンドウ上部) には、SAP 固有のメソッドが含まれます (次の図を参照)。



簡単なフィールドメソッド

オブジェクト `getImportField(String asName)` - [Simple Import] フィールドの値を名前で取得します。

`setImportField(String asName, String asType)` - [Simple Import] フィールドの値を名前とタイプで設定します。Map アクションの Target 式として使用されます。

オブジェクト `getExportField(String asName)` - [Simple Export] フィールドの値を名前で取得します。

構造メソッド

オブジェクト `getImportStructField(String asStructName, String asName)` - [Import Structure] フィールドの値を構造名とフィールド名で取得します。

ノード `getImportStructAsXML(String asStructName)` - [Import Structure] の名前を XML ノードとして取得します。

`setImportStructField(String asStructName, String asName, String asType)` - [Import Structure] フィールドの値を構造名、フィールド名、およびフィールドタイプで設定します。Map アクションの Target 式として使用されます。

`setImportStructWithXML(String asStructName)` - XML ノードに [Import Structure] の名前を設定します。

オブジェクト `getExportStructField(String asStructName, String asName)` - [Export Structure] フィールドの値を構造名とフィールド名で取得します。

ノード `getExportStructAsXML(String asStructName)` - [Export Structure] の名前を XML ノードとして取得します。

テーブルメソッド

オブジェクト `getTableField(String asTableName, String asName)` - [Table] フィールドの値をテーブル名とフィールド名で取得します。

ノード `getTableAsXML(String asTableName)` - [Table] の名前を XML ノードとして取得します。

`int getTableRowCount(String asTableName)` - 指定テーブルの行数を整数で取得します。

`appendTableRow(String asTableName)` - 指定テーブルの最後に行を追加します。

`nextTableRow(String asTableName)` - 指定テーブルの次の行にカーソルを移動します。

`setTableRow(String asTableName, int aiRow)` - 指定テーブルの指定行にカーソルを移動します。

`setTableField(String asTableName, String asName, String asType)` - [Table] フィールドの値をテーブル名、フィールド名、およびフィールドタイプで設定します。Map アクションの Target 式として使用されます。

`setTableWithXML(String asTableName)` - [Table] の名前を XML ノードで設定します。

関数メソッド

ノード `getFunctionAsXML()` - SAP 関数を表すノードを返します。実行前または実行後に使用できます。

`dumpFunctionToLog()` - `getFunctionAsXML()` を呼び出し、DATE 属性を関数名要素に追加し、Framework Factory `log()` メソッドを介してログにフォーマットを出力します。

文字列 `getLastError()` - 最新の実行エラーを返します。

文字列 `getFunctionName()` - 関数名を返します。

オブジェクト `getRFCObject()` - RFC オブジェクトを返します。JCo の場合、これは `JCo.Function` オブジェクトです。

setRequestSetAsString(boolean abFlag) - 実行前に関数への入力を文字列として設定するか、フィールドタイプに該当するオブジェクトに変換してから設定するかを決めます。デフォルトでは、このフラグは、ECMA ラッパで「True」です。たとえば、フラグが「True」で、フィールドタイプが DATE の場合、モデルドキュメントまたはパブリッシュセッタメソッドを介した入力は、「YYYYMMDD」形式の入力だと見なされます。フラグが「False」で、フィールドタイプが DATE で、入力がモデルドキュメントを介する場合、文字列が java Date() オブジェクトへの入力として使用されます。フラグが「False」で、フィールドタイプが DATE で、入力がパブリッシュセッタを介する場合、ユーザは、java Date() オブジェクトに直接渡すか、java Date() オブジェクトへの入力として使用される文字列に渡すことができます。

フラグが「False」のときのフィールドタイプ変換

BCD - java BigDecimal

BYTES - byte [] 配列。(現在)配列を渡すことはできません

CHAR - java String

DATE - java Date(YYYYMMDD パターンと

java Date() デフォルト文字列パターンの変換)

FLOAT - java Double

INT - java Integer

INT1 - java Integer

INT2 - java Integer

NUM - java String

XSTRING - byte [] 配列。(現在)配列を渡すことはできません

TIME - java Date

setResponseGetAsString(boolean abFlag) - 実行後に関数からの出力を文字列として取得するか、フィールドタイプに該当するオブジェクトとして設定するかを決めます。デフォルトでは、このフラグは、ECMA ラップで「True」です。たとえば、フラグが「True」で、フィールドタイプが DATE の場合、フィールドは、モデルドキュメントの場合でも、パブリッシュゲッタの場合でも、JCO.Field.getString() メソッドを使用して取得されます。フラグが「False」で、フィールドタイプが DATE で、入力がモデルドキュメントを介して取得される場合、フィールドは、JCO.Field.getObject() メソッドを使用して取得されてから、Objects.toString() メソッドを使用して文字列に変換されます。フラグが「False」で、フィールドタイプが DATE で、フィールドがパブリッシュゲッタメソッドを介して取得される場合、フィールドは、JCO.Field.getObject() メソッドを使用して取得されます。

フラグが「False」のときのフィールドタイプ変換

BCD - java String

BYTES - byte [] 配列 (応答モデルドキュメントに対する取得の場合、getString() が使用されます)

CHAR - java String

DATE - java Date

FLOAT - java Double

INT - java Integer

INT1 - java Integer

INT2 - java Integer

NUM - java String

XSTRING - byte [] 配列 (応答モデルドキュメントに対する取得の場合、getString() が使用されます)

TIME - java Date

テーブル行の処理

SAP テーブルを取得および設定する場合、テーブルに複数の行が含まれている可能性があるため、異なる論理が必要です。

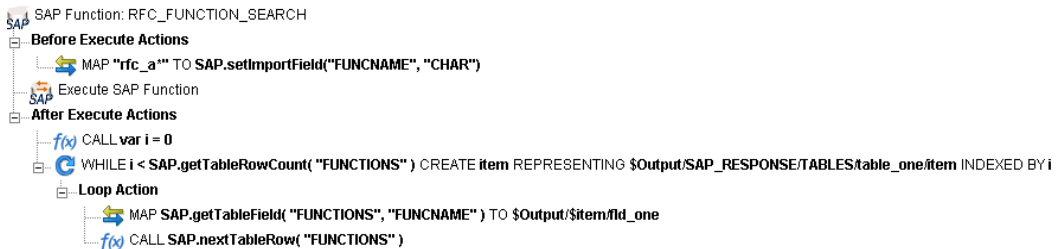
SAP Response モデルドキュメントを介してテーブル行を取得する

SAP Response ドキュメントの行は、<item> 要素で識別されます。複数の行を処理するには、SAP Function アクションの [After Execute Actions] リストで Repeat for Element アクションを作成します (図 8 の例を参照)。



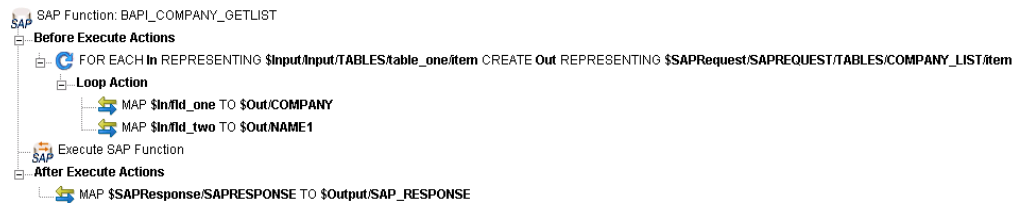
パブリッシュゲッターメソッドを介したテーブル行の取得

SAP.getTableRowCount(), SAP.getTableField(), および SAP.nextTableRow() を Repeat While アクションと併用すると、テーブルの項目を扱うことができます。



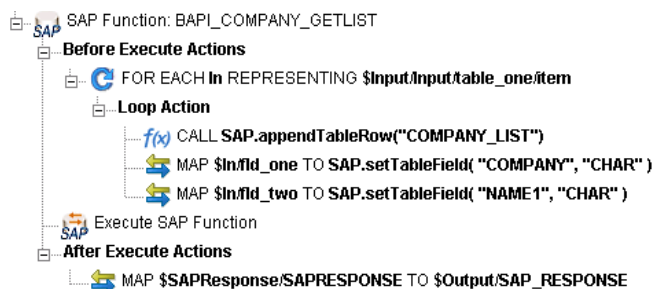
SAP Request モデルドキュメントを介したテーブル行の設定

SAP 関数を使用すると、ユーザは、テーブルを介して複数のレコードを入力できます。SAP Request モデルドキュメントを介して複数の行を挿入するには、[Before Execute Actions] リストで Repeat for Element アクションを作成します。



パブリッシュセッターメソッドを介したテーブル行の設定

SAP.appendTableRow() および SAP.setTableField() を Repeat for Element アクションと併用すると、テーブルに複数の行を挿入できます。



SAP コンポーネントエディタでの他のアクションの使用

Add SAP Function アクションに加えて、SAP コンポーネントエディタでは、その他すべてのアクションを使用できます。[Action] メニューには、基本的なアクションおよび高度なアクションの両方のリストが表示されます (次の表を参照)。

表 4-1

基本的なアクション	説明
Comment	アクションモデルを記録します。特に、アクションモデルに Decisions または Repeats、あるいはその両方が使用されている場合、コメントを使用して処理を明確にすることができます。
Component	別のコンポーネントを実行し、呼び出されたコンポーネントで受け渡すランタイム DOM を指定します。
Decision	指定した条件に基づいて、アクションの 2 つのセットから 1 つを実行できます。コンポーネントの実行で指定した条件がどのように解決されるかによって、True または False へのパスの分岐を処理します。
Declare Alias	XPath への任意のラベルを割り当て、操作を便利にすることができます。割り当てたラベルは、ランタイムまたはアニメーション時に完全な XPath に拡張されます。
Function	ECMAScript 関数または以前に作成したカスタムスクリプトのいずれかを実行します。カスタムスクリプトは、Composer のカスタムスクリプトリソースエディタを使用して作成できます (『Composer ユーザガイド』の第 9 章を参照してください)。

基本的なアクション	説明
Log	コンポーネントに指定されているさまざまなログファイルに情報を書き込みます。ログのタイプには、システム出力、システムログ、およびユーザログの3種類があります。
Map	要素のデータのあるXML DOMから別のXML DOMへ転送し、オプションで変換します。
Send Mail	コンポーネントの実行中、指定した電子メールアドレスに自動的に電子メールを送信します。
Switch	入力値と大文字小文字の値との一致に基づいて、プログラムの制御をアクションの特定のブロックに分岐させることができます。これは、長く、読み取りが困難なifまたはその他(Decisionアクション)のチェーンを排除するときを使用できる、基本的に便利なアクションです。

表 4-2 のアクションは、コンポーネントエディタの [Action] メニューで、[Advanced]、[Data Exchange and Repeat] の順にサブメニューを選択すると利用できます。

表 4-2

高度なアクション	説明
Apply Namespaces	NameSpace プリフィックスを上書きしたり、新しいNameSpace プリフィックスを宣言したり、またはNameSpace 全体を無視したりする方法を提供します。
Raise Error	条件を評価し、true となる場合は、ERROR と呼ばれるグローバル変数に式のコンテンツを記述します。単独で使用された場合は、例外をスローしてコンポーネントを停止し、サービスに制御を返します。Try On Error アクションの Execute 分岐内で使用された場合は、評価され、On Error 分岐でアクションに制御が渡されます。
Simultaneous Components	2 つまたはそれ以上のコンポーネントを同時に (つまり、マルチスレッド方式で) 実行できるようにします。
Transaction	非コンテナ管理サービスの一部として配備されるコンポーネントで <i>User Transaction</i> コマンド (開始、コミット、およびロールバックなど) を呼び出したり、コンテナ管理 EJB 配備の一部となるコンポーネントで <i>setRollbackOnly</i> を呼び出したりできます。

高度なアクション	説明
Try On Error	一連のアクションを実行することで、エラーを生成するアクションに応答します。Try On Error アクションは、本質的にエラートラップおよび解決を行うアクションです。
XSLT Transform	XSL ファイルの指示に従ってXMLファイルを変換します。出力は、一般的に Web ブラウザに XML ファイルを表示するために使用されます。

表 4-3

Data Exchange アクション	説明
UR_/File Read	XML でないファイル形式を Composer に読み込むことができます。
UR_/File Write	ファイルを XML 以外の別の形式で書き込むことができます。
WS Interchange	WSDL リソースで定義されたメッセージおよび操作を使用して Web サービスを実行します。
XML Interchange	外部 XML ドキュメントをコンポーネントの DOM に読み込んだり、外部 XML ドキュメントにコンポーネントの DOM を書き込んだりします。読み込み / 書き込みメソッドには、ファイル、FTP、HTTP、および HTTPS プロトコルを使用した Get、Put、Post、および Post with Response が含まれます。

表 4-4

Repeat アクション	説明
Break	Repeat for Element、Repeat for Group、または Repeat While ループの実行を停止し、ループ外で次のアクションの実行を続行します。
Continue	Repeat for Element、Repeat for Group、または Repeat While ループで現在のループ反復の実行を停止し、次の反復で同じループの一番上から続行します。
Declare Group	複数回発生する要素に基づきグループを作成して、グループに名前を付けることができます。グループは、Repeat for Group アクションで使用されます。

Repeat アクション	説明
Repeat for Element	DOM ツリーに指定した要素が発生するごとに 1 つまたは複数のアクションを繰り返します。Repeat for Element アクションでは、複数回発生する要素に基づき、ループを作成できます。
Repeat for Group	グループの各メンバーに対して 1 つまたは複数のアクションを繰り返します。Repeat For Group アクションでは、データを再作成して、データを集約計算できます。
Repeat While	ループを作成することで、1 つまたは複数のアクションを繰り返します。Repeat While アクションでは、処理ループを任意の有効な ECMAScript 式に基づかせることができます。

エラーおよびメッセージの処理

エラー処理は拡張されました。設計時に接続エラーが発生した場合、Composer によりその接続が解放されます。このため、ユーザは、コンポーネントを終了せずに問題を解決して、回復できます。

[SAP Function] ダイアログボックスでは、[Functions] リストに 3 つのエラーメッセージのいずれかが表示されます。

No function(s) found - RFC_FUNCTION_SEARCH で、条件に一致する RFC が見つかりませんでした。[Search] を押したときに発生することがあります。

Error searching for functions - 通信エラーまたはその他のシステムエラー。[Search] を押したときに発生することがあります。

Error getting metadata for function - 通信エラーまたはその他のシステムエラー。関数の選択時に発生することがあります。

A テスト

アニメーションテストと配備テストの環境的相違

Composer でのアニメーションテストと、配備テストには、大きな環境的相違があります。両方のタイプのテストは、作成したコンポーネントやサービスを適切に検証するために必要です。相違点については、次の表に詳しく説明します。

表 B-1

	Composer でのテスト	配備テスト
OS	Win98、WinNT、または Win 2000	WinNT、Sun Solaris、IBM AIX、または HP-UX
プラットフォーム	JRE (Java Runtime Environment).	フェイルオーバー、セキュリティ、接続 MGT などに対する JRE サポートを完全に備えたアプリケーションサーバ
コンポーネントまたはサーバの起動	Composer から直接的	サービストリガによってのみ (つまり、配備サーブレットまたは EJB)
xObject アクセス	ディスクファイルから	アプリケーションサーバ内の JAR ファイルから
ランタイムコンテキスト	個々のコンポーネント、またはサービス内で実行されているコンポーネントをテスト	常にサービス内から
サービス入力およびコンポーネント入力	入力ドキュメントは、ローカルマシンのサンプルXMLドキュメントだけでなく、他のサービスやコンポーネントからの DOM に由来することが多い	入力ドキュメントは、サービストリガ、または他のサービスやコンポーネントからの DOM を介して、サービスとコンポーネント内に渡される

	Composer でのテスト	配備テスト
次に対するプロジェクト変数: * ログファイルのパス * DTD URI * XSL URI * メール送信サーバ * XML 交換 URI の変更	通常はローカルマシン上の場所を指す (ただし、サーバや Web 上の場所でも可)	プロダクションサーバおよび Web 上の場所を指す
テストツール	Log アクション以外に、ダイアログボックス (ECMAScript alert() 関数) を使用してランタイム値を表示することも可能	ダイアログボックスの使用は不可
JDBC 接続	サーバ接続プールを使用しない - テストデータベースを使用している可能性がある	サーバ指定の接続プールを使用する - 製品データベースを使用している
HTTP 接続	ローカルマシンまたはテストサーバを指している可能性がある	テストサーバまたはプロダクションサーバを指している
SAP 接続	通常、システムのテスト、およびユーザ ID とパスワードのテストを指している	製品システム、および製品ユーザ ID とパスワードを指している

B

ドキュメント管理

ドキュメント管理について

各 SAP Function アクションには、2つのドキュメント名と別名が関連付けられています。ドキュメントは、SAP 要求ドキュメントと SAP 応答ドキュメントです。これらの別名は、[Use Mapping Model] が選択されているかどうかに関係なくアクションと保存されます。[SAP Component] では、これらの名前前のテーブルが維持されます。これにより、新しい固有名が生成され、既存の名前が使用されなくなります。

SAP Function アクションを削除しても、そのアクションに関連付けられている別名はテーブルから削除されません。これらの名前を再利用するには、コンポーネントを閉じて再び開くか、[Component] メニューから [Reload XML Documents] コマンドを実行する必要があります。

ドキュメントは、必要に応じてロードされます。SAP Function アクションをクリックすると、アクション内に保存されているモデル定義に基づいた別名で XML ドキュメントが作成され、作成されたドキュメントがドキュメントマネージャに保存され、必要なドキュメントがネイティブ環境ペインにロードされます。

ユーザが [Reload XML Documents] を実行すると、次のタスクが実行されます。

- ◆ 別名テーブルをクリアする
- ◆ 元に戻す / やり直しスタックをクリアする
- ◆ アクションツリーの各 SAP Function アクションに対して、別名テーブルの要求および応答エントリを追加する
- ◆ 要求をドキュメントマネージャに送り、これらの名前からドキュメントを削除する

別名を複数の SAP Function アクションで共有するには、SAP Function アクションをコピーして、貼り付けます。ただし、共有する場合、ネイティブ環境ペインと現在の SAP Function アクションを同期化する必要があります。各 SAP Function アクションには、同期化のため、別名テーブルに別名と保存されている固有の ID が割り当てられています。SAP Function アクションをクリックすると、ネイティブ環境ペインにより、各別名のエントリに対して、現在のアクションと関連付けられているかどうかチェックされます。関連付けられている場合、要求がドキュメントマネージャに送られます。関連付けられていない場合、再ロードフラグが「True」に設定された要求が、ドキュメントマネージャに送られ、SAP Function アクションの ID で別名テーブルのエントリが更新されます。

C

SAP 用語集

ALE

アプリケーションリンクの有効化

配布済みアプリケーション作成と操作、および同期通信と非同期通信によるアプリケーション統合をサポートします。ルーズリンクされた SAP アプリケーションで一貫したデータをビジネスコントロールのメッセージ交換に提供します。

BAPI

Business Application Programming Interface。

BOR

Business Object Repository。

IDoc

中間ドキュメント (Intermediate Document)。IDoc は、データを階層的に保存できる構造化データコンテナです。このデータフォーマットは、SAP の R/3 および mySAP.com 製品で生成されます。

JCo

Java コネクタ。

RPC

リモートファンクションコール (Remote Function Call)。

SAP Function アクション

SAP コンポーネントエディタの特別なアクションで、SAP ファンクションを検索および選択して、ネイティブ環境ペイン (NEP) に表示される [SAP Request] フィールドと [SAP Response] フィールドを選択できます。

XML スキーマ

XML ドキュメントで、インタフェースの呼び出しまたはその呼び出しの結果がどのように表されるか、またインタフェースリポジトリの各インタフェースに対してどのように保存されるかを定義します。

XML ビジネスドキュメント

リポジトリで定義される XML スキーマからインスタンス化される XML ドキュメント。

インタフェースリポジトリ

企業内通信に関するすべての SAP インタフェースが発行される場所。

構造化データ (BA またはレコード)

このデータは、パラメータ要素のサブ要素として構造のすべてのフィールドに入力される XML 要素で表されます。

コントロールレコード

IDoc の内容、構造、受信者、および状態についての情報を含みます。

スカラデータ

このデータは、パラメータ要素の内容として直接表されます。このデータは、テキスト形式で表されます。

ステータスレコード

IDoc の直前の処理手順を記述します。

データレコード

管理部分 (コントロールフィールド)、およびデータ部分 (セグメント) で構成されます。

テーブルデータ (内部 SAP テーブルまたは配列)

このデータは、パラメータ要素のサブ要素としてテーブルのすべての行に入力される XML 要素で表されます。

ネイティブ環境ペイン (NEP)

SAP コンポーネントエディタのペインで、XML ドキュメントデータをマップできる [SAP Request] フィールドと [SAP Response] フィールドが表示されます。

索引

数字

3270 22
5250 22

A

[After Execute Actions] - [Response] タブからの Map
アクションの追加 50
AIX 21
ALE 69
Apply Namespaces 62

B

BAPI 13, 69
[Before Execute Actions] 45
BOR 13
Break 63

C

CICS RPC 22
Comment 61
Component 61
Component with Connection アクション 52
Component with Connect アクションの使用 52
Continue 63

D

Data Exchange アクション 63
UR_/File Read 63
UR_/File Write 63
WS Interchange 63
XML Interchange 63
Decision 61
Declare Alias 61
Declare Group 63

E

ECMAScript 22, 46, 66
EDI 22
Expression Editor 45
exteNd Composer Service 16
exteNd コネクタ 12

F

forbidden drag operation 48
Function 61

H

HP-UX 21
HTML 22

I

[Inactivity Lifetime] 27
[Include Field Attributes] 40

J

Java コネクタ 13, 14
JCo 13, 14, 19, 57
JDBC 22
JMS 22

K

[Keep Alive] 27

L

Log 62

M

Map 62
Map アクション 48
[Map] コマンド 45

O

On Error 62
Override NameSpace 62

P

[Pool Info] 28

R

Raise Error 62
[Reload XML Documents] 67
Repeat for Element 54, 64
Repeat for Group 64
Repeat While 64
Repeat アクション 63
 Break 63
 Continue 63
 Declare Group 63
 Repeat for Element 64
 Repeat for Group 64
 Repeat While 64
RFC 対応関数モジュール 14
RFM 14

S

SAP Connect 13
SAP Function アクション 38
SAP Function アクション構造 43
SAP Function アクションの編集 43
[SAP Function] タブ 38
SAP R/3 14
[SAP Request] 38
[SAP Request] タブ 36
[SAP Request] タブへの XML ドキュメントの一部の
 マップ 45
[SAP Response] 38
[SAP Response] タブ 36
SAP アクションの実行 48
SAP アクションモデル 44
SAP コンソール 28
SAP コンポーネント 31
 概要 16

SAP コンポーネントエディタ
 アプリケーションの作成 17
SAP コンポーネントの作成 31
SAP 接続リソースの作成 22
SAP 専用 Expression Builder プロパティ 55
Send Mail 62
Simultaneous Component 62
Solaris 21
Switch 62

T

Telnet 22
Transaction 62
Try On Error 63

U

UR_/File Read 63
UR_/File Write 63

W

Windows 2000 20
Windows NT 20
WS Interchange 63

X

XML Interchange 63
XML テンプレート 30
XPath 45
XSLT Transform 63

あ

アクション 37, 61
アクションモデル 37
新しい xObject 31
アニメーションテストと配備テストの環境的
 相違 65
アプリケーションの作成 17

い

インポート 56

え

エラーおよびメッセージの処理 64

お

応答 15

か

カスタムスクリプト 30

関数メソッド 57

簡単なフィールドメソッド 56

き

基本的なアクション 61

comment 61

Decision 61

Function 61

Log 62

Map 62

Send Mail 62

Switch 62

コンポーネント 61

宣言 61

キュー 12

こ

構造メソッド 56

高度なアクション 62

Apply Namespaces 62

Raise Error 62

Simultaneous Component 62

Transaction 62

Try On Error 63

XSLT Transform 63

コードテーブルマップ 30

コンポーネントエディタ 35

さ

作成

SAP 接続リソース 22

し

式駆動型接続 22

照会 22

せ

セッション 25

セッション接続管理 26

接続プール 25, 26

接続プール管理 26, 28

接続リソース 16, 22

た

タイムアウト 30

て

定数駆動型および式駆動型の接続 22

定数駆動型接続 22

データ操作 16

テーブル行 59

テーブル行の処理 59

テーブル行の設定 60

テーブルメソッド 57

テスト 25

デフォルト 25

テンプレート 30, 32

と

ドキュメント管理 67

ね

ネイティブ環境ペイン 35, 67

ネームスペース 62

は

ハブとスポーク 12

ひ

ビジネスオブジェクトタイプ 13

ふ

フィールドタイプ変換 58

プール名 29

へ

別名 68

ま

マッピングモデルの使用 40

マップの制約 48

ゆ

ユーザ ID 16

よ

要求 15