

# Novell exteNd Composer™

## CICS RPC Connect

4.2

[www.novell.com](http://www.novell.com)

ユーザガイド



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) <<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) <<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) <<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、IU Software に関連して、または IU 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、ライセンサ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](http://www.novell.com)

CICS RPC 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](http://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
<b>1 exteNd Composer と CICS RPC へようこそ</b>	<b>9</b>
はじめに	9
CICS RPC Connect について	10
CICS RPC とは	10
exteNd の CICS RPC コンポーネントについて	12
CICS RPC コンポーネントエディタを使用して作成できるもの	13
<b>2 CICS RPC コンポーネントエディタをお使いになる前に</b>	<b>15</b>
サンプルトランザクション	15
CICS RPC 接続リソースの作成	16
接続リソースについて	16
パフォーマンスパラメータおよび接続リソース	16
定数駆動型および式駆動型の接続について	17
コードページサポートについて	20
コンポーネントに対する XML テンプレートの作成	21
<b>3 CICS RPC コンポーネントを作成する</b>	<b>23</b>
CICS RPC コンポーネントを作成する前に	23
CICS RPC コンポーネントエディタウィンドウについて	27
CICS RPC ネイティブ環境ペインについて	28
CICS RPC 専用メニューバー項目について	28
[自動マップコピーブック]	29
CICS RPC 専用コンテキストメニュー項目について	29
Auto Map を使用してコピーブックから入力サンプルドキュメントを作成	31
Auto Map を使用してコピーブックから出力サンプルドキュメントを作成	32
入力テンプレートおよび出力テンプレート手動で作成	33
<b>4 CICS RPC アクションの実行</b>	<b>35</b>
アクションについて	35
CICS RPC 専用アクションについて	35
CICS RPC 専用の Expression Builder 拡張	36
RPC コンポーネントエディタでの他のアクションの使用	38
エラーおよびメッセージの処理	40
ECI 接続エラーメッセージ	40
データ型変換エラーメッセージ	41

<b>A</b>	サポートされるデータ型の一覧	43
<b>B</b>	サポートされるデータ型の文字一覧	45
<b>C</b>	アニメーションテストと配備テストの環境的相違 テスト.....	47 47
<b>D</b>	Java でサポートされるコードページの一覧	49
<b>E</b>	CICS RPC 用語集	51

# このガイドについて

## 目的

このガイドでは、exteNd Composer CICS RPC Connect の設計時アプリケーションの一部である CICS RPC コンポーネントエディタの使用方法を説明します。

## 対象読者

このガイドは、CICS 対応コンポーネントを必要とするアプリケーションまたはサービスを構築するシステムアナリスト、プログラマ、およびその他の担当者を対象としています。

## 前提条件

このガイドでは、exteNd Composer の設計時環境および Composer のアプリケーション構築例についての予備知識が必要です。また、CICS RPC の概念についてすでに理解されていることが前提となります。

## 追加のドキュメント

Novell exteNd Director 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 \(http://www.novell.com/documentation-index/index.jsp\)](http://www.novell.com/documentation-index/index.jsp) を参照してください。





# 1

## exteNd Composer と CICS RPC へようこそ

### はじめに

『CICS RPC 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。『Composer 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。

Novell exteNd Composer には、CICS RPC と同様に 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、このガイドと類似する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 CICS RPC 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。

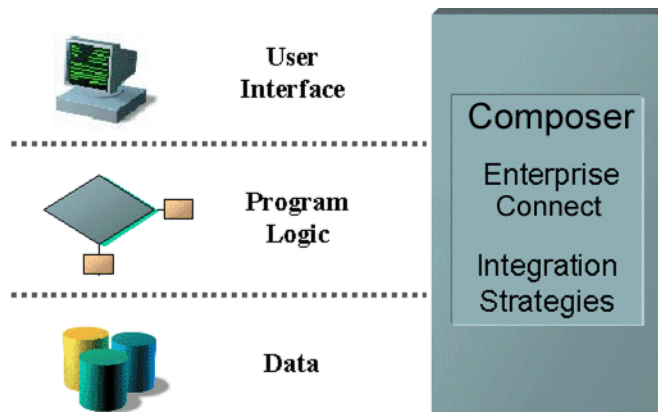
作業を始める前に、まず CICS RPC Connect を既存の exteNd Composer にインストールする必要があります。また、この Connect で作成されたサービスを Composer Enterprise Server 環境で実行するには、この Connect 用のサーバ側ソフトウェアが Novell exteNd Enterprise Server にインストールされている必要があります。

**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、インタフェースを操作するプログラムだけでなく、CICS の基礎と COBOL プログラム言語に関する知識が必要です。

# CICS RPC Connect について

Novell 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 Connect) 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/ アプリケーションから、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。

CICS RPC Connect 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、B2B 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。



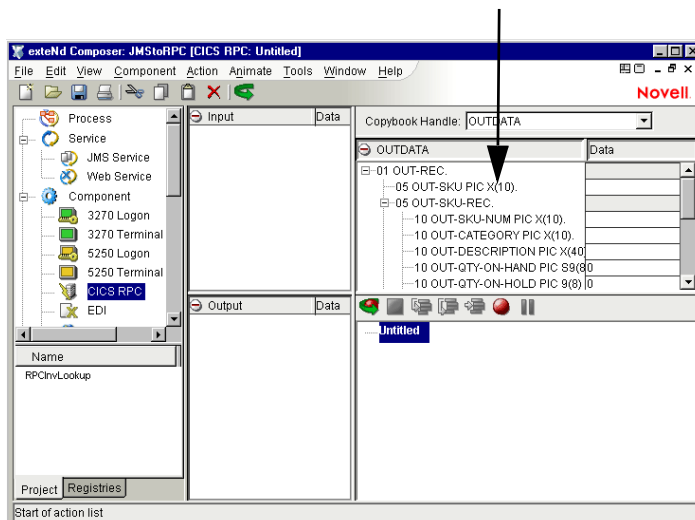
## CICS RPC とは

CICS RPC は、Customer Information Control System Remote Procedure Call の略語です。CICS RPC Connect XML では、ロジック統合ストラテジを使用してレガシーシステムデータを有効に活用できます。CICS RPC コンポーネントには、CICS の External Call Interface (ECI) を利用して CICS 管理プログラムとインタフェースを取り、通常 DFHCOMMAREA として参照される標準の通信エリアを使用している CICS リージョンで実行中のプログラムと直接データ交換ができる機能が備わっています。Novell exteNd は、コピーブックを介してアプリケーション内で定義されることが多いため通信エリアの表現として紹介される場合が多いです。

Novell exteNd CICS RPC Connect は、IBM の CICS Java Gateway を通じて CICS 環境で動作することを目的として設計されました。Java ゲートウェイは、ユーザのシステムアーキテクチャに応じて、CICS リージョンと同一のプラットフォームまたは別のプラットフォームのいずれにでも常駐することができます。CICS は、IBM のプロプライエタリな EBCDIC 文字エンコーディングスキーマを使用して IBM のホストシステムに常駐することの多い IBM トランザクション処理環境ですが、EBCDIC 以外の文字エンコーディングスキーマをサポートする他のプラットフォーム上の IBM TXSeries 製品ラインの下でも CICS を実行することができます。

CICS RPC Connect は、XML 変換ハブに受け渡されるデータインタフェースとして DFHCOMMAREA のコピーブックの表現を使用します。これらのコピーブックは、CICS RPC コンポーネントエディタのネイティブ環境画面に表示されます。ネイティブ環境画面では、データを変換、変形あるいは他の XML ドキュメントと同じように転送することができます。

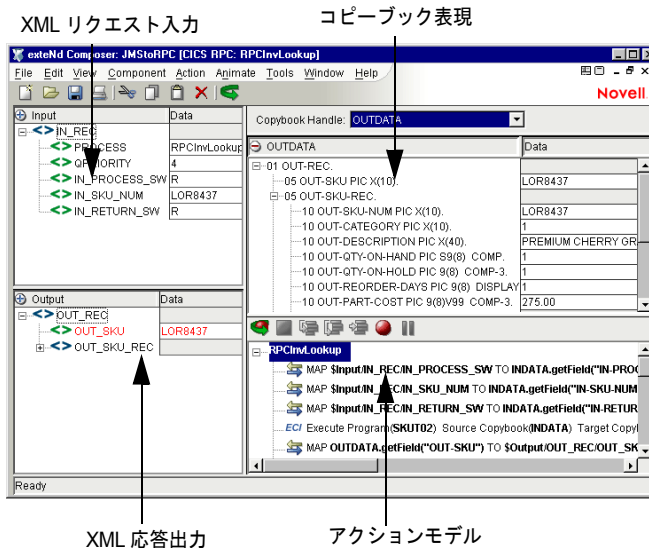
**RPC コピーブックインタフェースが  
ネイティブ環境ペインに表示されます。**



# exteNd の CICS RPC コンポーネントについて

XML Map コンポーネントと同様に、CICS RPC 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。ただし、CICS プログラムに接続して、DOM からの要素を使用してデータを処理し、結果を出力 DOM にマップします。その後、出力を Web サイトに表示するなど、組織に適切な方法で、出力 DOM に基づいて処置を取ることができます。本質的には、レガシーシステム自体を変更せずに、レガシーシステムからデータを取得したり、レガシーシステムにデータをプッシュすることが可能です。CICS RPC コンポーネントでは、1 つの XML ドキュメントから別の XML ドキュメントへ、あるいは XML ドキュメントからプログラムへ、データのマッピングや転送など、単純なデータ操作を実行できます。また、OCCURS 句を参照することで複数行のマップを実行するなどの高度な操作も実行できます。CICS RPC 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。

ECI 接続を介した CICS RPC コンポーネントとメインフレームのデータの関係を次の図に示します。



## CICS RPC コンポーネントエディタを使用して作成できるもの

CICS RPC コンポーネントエディタを使用すると、作成している XML 統合アプリケーションを拡張して、ECI 通信をサポートする CICS ビジネス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 ユーザーガイド』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、およびインベントリ情報を取得して 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。CICS RPC 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

## CICS RPC コンポーネントエディタをお使いになる前に

### サンプルトランザクション

本書全体を通して、SKUT02 と呼ばれるプロトタイプ的な COBOL プログラムを使用して CICS RPC Connect の優れた機能を説明します。SKUT02 は、インベントリシステムを照会する読み取り専用のオペレーションを実行するように設計されています。クエリは、3 つのパラメータを使って定義されます。最初のパラメータは、実行するオペレーションのタイプ (READ, UPDATE, など) を識別するためのプロセススイッチです。この目的のため、スイッチは常に R (READ の先頭文字) すなわち読み取りに設定されています。2 番目のパラメータは、固有の情報を取得するために特定の SKU またはプロダクト ID となります。3 番目のパラメータは、照会によって戻された情報がテキストメッセージあるいはインベントリレコードのいずれであるのかを指定するスイッチです。メッセージは、その項目が存在するのか、あるいは存在しないのかを示します。レコードはデータが含まれているか、あるいは空白のまま戻されます。

#### ➤ 新しい CICS RPC コンポーネントを作成するときに通常使用される手順

- 1 コピーブックとしてホストプログラム DFHCOMMAREA 定義のコピーを入手または作成し、これをローカルのコンピュータからアクセスできるようにします。
- 2 CICS RPC コンポーネントを作成します。
- 3 入力 XML テンプレートと複数のサンプルドキュメントを作成します (または、Auto Map 機能を使用してこれを作成することもできます)。
- 4 入力 DOM からコピーブックにデータを移動するための Map アクションを作成します。または、Auto Map 機能を使用してこれを作成することもできます。
- 5 Execute ECI アクションを作成します。
- 6 出力用に手順 3 から 4 を繰り返します。
- 7 コンポーネントをテストします。

# CICS RPC 接続リソースの作成

CICS RPC コンポーネントを作成する前に、ホストプログラ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。

## 接続リソースについて

CICS RPC Connect の接続リソースを作成する場合は、ECI CICS RPC 接続を選択します。ECI CICS RPC 接続は、CICS TCP プロトコルアプリケーションに接続したいときに使用します。この接続によって、ホスト上の TCP/IP ポートを介して CICS ゲートウェイに接続することができます。

## パフォーマンスパラメータおよび接続リソース

exteNd Composer を指定して CICS トランザクションゲートウェイを使用する場合、使用するアプリケーションに合わせてゲートウェイの調整をしておく必要があります。調整を行うためには、あらかじめ何人位のユーザが **Composer** サービスを同時に使用する可能性があるかを予測しておく必要があります。この予測によって、接続の調整をするのに役立つ 2 つのゲートウェイパラメータを設定できます。パラメータは、*initworker* と *initconnection* です。

*Initworker* は初期化された作業スレッドの数です。*Initconnection* は初期化された接続の数です。これら 2 つのパラメータは、ゲートウェイの起動時にコマンドラインパラメータとして、またはゲートウェイ 3.1.x バージョンの *ctg.ini* ファイルの中、あるいはゲートウェイ 3.0 バージョンのゲートウェイプロパティファイルの中に設定しておくことができます。

以上説明したとおり、これらのパラメータを適切に設定するためにはアプリケーションの負荷を正確に把握しておく必要があります。たとえば、50 人のユーザが同時にアプリケーションを使用することが予想される場合、*initworker* と *initconnection* の両方を 50 に設定します。コマンドを使って *.ini* ファイルにパラメータを設定する例を次に示します。

パラメータをコマンドラインパラメータとして設定するには、ゲートウェイの起動時に、スイッチを `-initconnect=50 -initworker=50` とします。

ゲートウェイのバージョンが 3.1.x の場合、*initconnect* パラメータと *initworker* パラメータは *ctg.ini* ファイルに設定されています。*ctg.ini* ファイルの最初の行を示した SECTION GATEWAY の例を次に示します。



```
SECTION GATEWAY
closetimeout=20000
ecigenericreplies=on
initconnect=50
initworker=50
maxconnect=100
maxworker=100
noinput=off
nonames=off
notime=off
tfile=ctg.trc
workertimeout=20000
```

```
protocol@tcp.handler=com.ibm.ctg.server.TCPHandler
```

```
protocol@tcp.parameters=connecttimeout=5000;idletimeout=600000;pingfrequency=600
00;port=2006;solinger=0;sotimeout=1000;
ENDSECTION
```

ゲートウェイのバージョンが 3.0.x の場合、*initconnect* パラメータと *initworker* パラメータは *Gateway.properties* ファイルに設定されます。*Gateway.properties* ファイルの最初の行を示した SECTION GATEWAY の例を次に示します。IBM ではバージョン 3.0 をサポートしなくなったため、バージョン 3.1 にアップグレードするようにお勧めします。

```
# Initial number of ConnectionManager threads
# initconnect=50
```

```
# Maximum number of ConnectionManager threads
# maxconnect=100
```

```
# Initial number of Worker threads
# initworker=50
```

```
# Maximum number of Worker threads
# maxworker=100
```

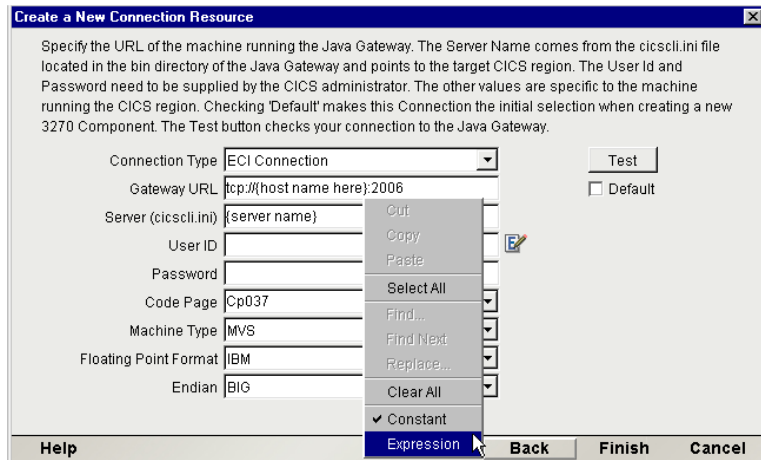
## 定数駆動型および式駆動型の接続について

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で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える、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、ECI CICS 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 (次を参照)。
- 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 ECMAScript 式を作成します。



➤ CICS RPC 接続リソースの作成

- 1 exteNd Composer の [File] メニューから、[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:

Description:

**注記：** または、Composer ウィンドウのカテゴリペインで **[Connection]** を選択し、マウスを右クリックした後で **[New]** を選択することもできます。

- 2 接続オブジェクトの名前を入力します。
- 3 オプションで、「説明」のテキストを入力します。
- 4 **[Next]** をクリックします。
- 5 **[Connection Type]** ドロップダウンメニューから **[ECI Connection]** を選択します。

**注記：** CICSアプリケーションに接続している場合は**[ECI Connection]**を使用します。

**Create a New Connection Resource**

Specify the URL of the machine running the Java Gateway. The Server Name comes from the cicscli.ini file located in the bin directory of the Java Gateway and points to the target CICS region. The User Id and Password need to be supplied by the CICS administrator. The other values are specific to the machine running the CICS region. Checking 'Default' makes this Connection the initial selection when creating a new 3270 Component. The Test button checks your connection to the Java Gateway.

Connection Type:

Gateway URL:   Default

Server (cicscli.ini):

User ID:

Password:

Code Page:

Machine Type:

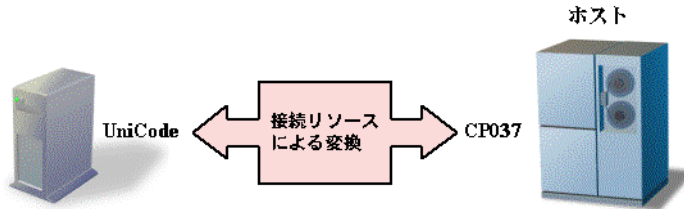
Floating Point Format:

Endian:

- 6 [Gateway URI] フィールドに、[ECI Connection] タイプを選択した場合は、接続先のコンピュータの物理 (IP) アドレスを入力します。例では、*tcp://TIGEREYE:2006* は物理コンピュータのエイリアスです。*tcp://* はプロトコル、*TIGEREYE* はホスト名、*2006* は CICS へのポートトランザクションゲートウェイです。
- 7 [Server (cicscli.ini)] フィールドにサーバ名、この例の場合は **CICS\_ECI** 入力します。
- 8 [User ID] フィールドと [Password] フィールドには、システム管理者から提供された名前とパスワードを入力します。これらのユーザ ID とパスワードは、別のクライアント初期化ファイルで定義されています。
- 9 [Code Page] フィールドは、ドロップダウンメニューからコードページを選択します。コードページは、リモートコンピュータのオペレーティングシステムの文字データ標準に依存します (たとえば、CP037 は EBCDIC 用、8859\_1 は ASCII 用)。
- 10 [Machine Type] フィールドは、ドロップダウンメニューから、ユーザの CICS Region/Server (MVS、OS2、NT、AIX) のターゲットプラットフォームを選択します。
- 11 [Floating Point Format] フィールドは、ドロップダウンメニューから、選択したシステムタイプに対応する名前、IBM または IEEE を選択します。
- 12 [Endian] フィールドは、ドロップダウンメニューから整数の最大/最小重みバイトの順序を選択します (メモリ内で最大重みバイトが最小重みバイトの前にくる場合は BIG、それ以外の場合は LITTLE を選択します)。
- 13 接続を CICS RPC コンポーネントのデフォルトの接続設定として保存したい場合は、[Default] チェックボックスをクリックします。

## コードページサポートについて

exteNd Composer 接続リソースのコードページサポートでは、exteNd Composer とその他のホストシステム間で送信された文字を変換する際に使用する文字エンコードスキームを指定できます。Novell exteNd では、Unicode 文字エンコード (Java および XML 標準) が使用されます。既存のレガシーホストシステムとその他のホストシステムでは、言語または使用に特定のさまざまな文字エンコードスキーム (つまり、コードページ) を使用します。したがって、互いに通信する場合は、これらのシステム間で文字エンコードを変換するためのメカニズムが必要となります。これは、そのシステムにアクセスするのに使用される接続リソースのホストシステムで使用されるコードページを指定することによって、exteNd Composer で処理されます。Java でサポートされるコードページのリストについては、付録 E を参照してください。



## コンポーネントに対する XML テンプレートの作成

CICS RPC に特有の機能として、コンポーネントメニューの自動マップ選択によって CICS RPC コンポーネントに関連付けられたコピーブックを基にして自動的に XML テンプレートとサンプルドキュメントを作成できます。この機能は、特にアプリケーションからの情報を複数のサービスで使用する場合、あるいは CICS プログラムが複数の用途で使用される場合（たとえば、すべてのタイプの情報アクセスつまり照会、更新、挿入、および削除を実行するプログラムなど）に使用します。

ただし、コンポーネントを設計する前にユーザ独自の XML テンプレートを作成することができます。（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）。このオプションは、コンポーネントでコピーブックの一部のデータだけを利用する場合に使用します。

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、CICS RPC 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が推奨されます。詳細については、『exteNd Composer ユーザガイド』を参照してください。



# 3

## CICS RPC コンポーネントを作成する

### CICS RPC コンポーネントを作成する前に

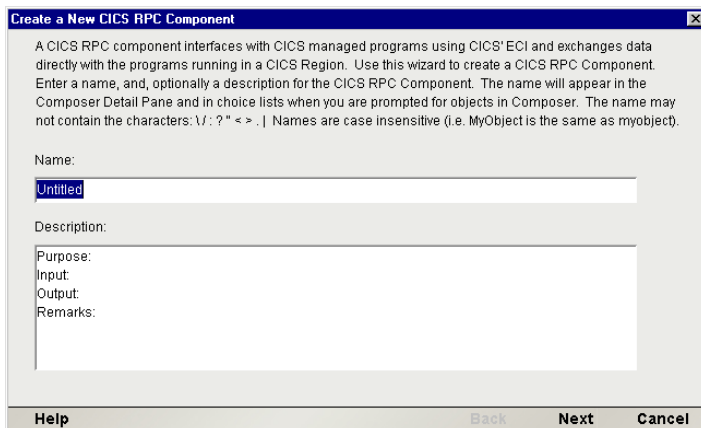
すべての Novell exteNd コンポーネントと同様に、CICS RPC コンポーネントを作成する最初の手順は、必要な XML テンプレートを指定することです ( 詳細については、『*Composer ユーザガイド*』の「*Creating a New XML Template*」を参照してください)。

Auto Map 機能を使用して XML テンプレートとサンプルドキュメントを作成する場合、最初の CICS RPC コンポーネントの作成時に入力および出力用のテンプレートとして [System] と [ANY] を選択します。

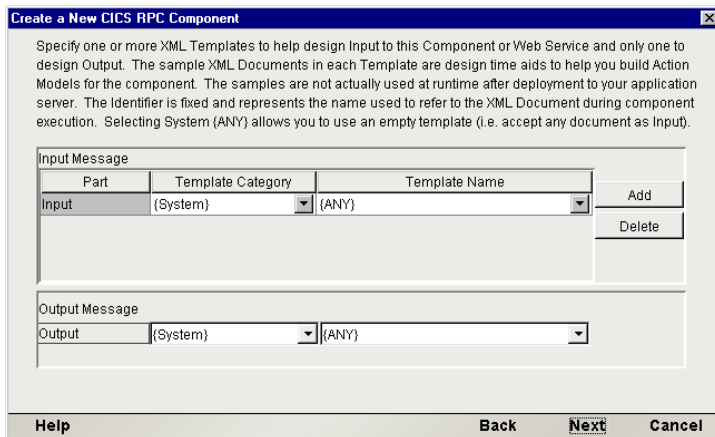
また、CICS RPC コンポーネントの作成プロセスの一部として、CICS RPC 接続を選択したり、新しい接続を作成したりできます。事前に接続を作成する場合、新しい CICS RPC コンポーネントすべてに対して選択できるようになります。

新しい CICS RPC コンポーネントを作成する

- 1 exteNd Composer で [File] > [New xObject] > [Component] > [CICS RPC] の順に選択します。すると、Create a New CICS RPC Component ウィザードが表示されます。



- 2 新しい CICS RPC コンポーネントの「名前」を入力します。
- 3 オプションで「説明」のテキストを入力します。
- 4 「次へ」をクリックします。New CICS RPC Component ウィザードの XML プロパティ情報パネルが表示されます。



- 5 入力テンプレートを次のように指定します。
  - a) Auto Map 機能を使用しない場合は、[Template Category] を選択します。
  - b) Auto Map 機能を使用しない場合は、選択した [Template Category] の XML テンプレートリストから [Template Name] を選択します。
  - c) 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。



- d) 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** 出力用に手順**[c]**から**[d]**を繰り返します。出力 DOM の名前は「Output」です。
- 7** ほとんどの新規 CICS RPC コンポーネントの場合、手順 **[b]** と **[d]** でデフォルト **[System]** を選択し Auto Map ツールを使用します。
- 8** **[Next]** をクリックします。すると、「Create a New CICS RPC Component」ウィザードの **[Connection Info]** パネルが表示されます。

The screenshot shows the 'Create a New CICS RPC Component' wizard window. The title bar reads 'Create a New CICS RPC Component'. The main text says: 'Specify which Connection you wish to use for this Component or Service. To change any connection parameters, you must change them in the Connection Resource object or create a new Connection Resource of the same type with different parameters.'

The form contains the following fields and controls:

- Connection: TigerEye Region=CICS\_ECI (dropdown menu)
- Gateway URL: tcp/TIGEREYE:2006 (text field)
- Server (cicscli.ini): CICS\_ECI (text field)
- User ID: gemlogic (text field)
- Password: \*\*\*\*\* (text field)
- Code Page: 8859\_1: ISO Latin alphabet No. 1 (dropdown menu)
- Machine Type: NT (dropdown menu)
- Floating Point Format: EEE (dropdown menu)
- Endian: LITTLE (dropdown menu)
- Buttons: Test, Help, Back, Next, Cancel

- 9** プルダウンリストで「**接続**」タイプを選択します。
- 10** **[Next]** をクリックします。すると、「Create a New CICS RPC Component」ウィザードの **[Connection Info]** パネルが表示されます。

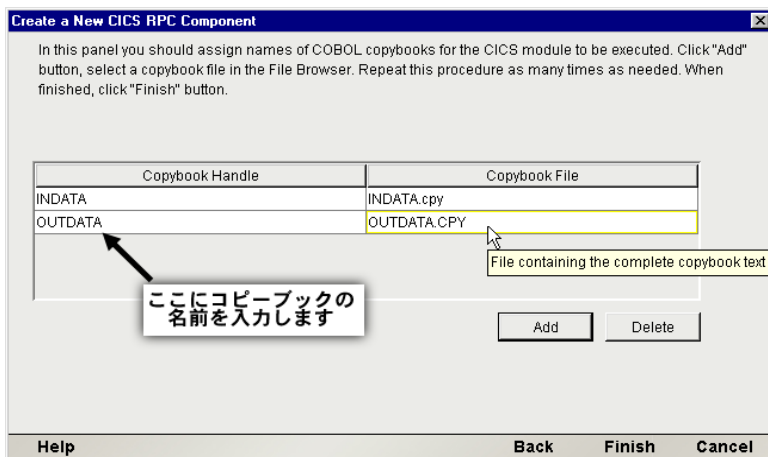
The screenshot shows the 'Create a New CICS RPC Component' wizard window. The title bar reads 'Create a New CICS RPC Component'. The main text says: 'In this panel you should assign names of COBOL copybooks for the CICS module to be executed. Click "Add" button, select a copybook file in the File Browser. Repeat this procedure as many times as needed. When finished, click "Finish" button.'

The form contains the following elements:

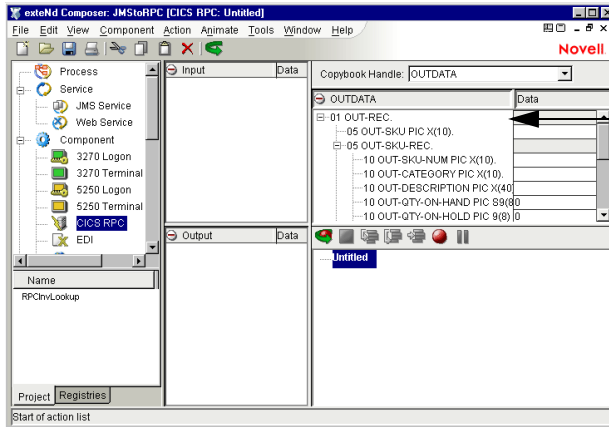
- A table with two columns: 'Copybook Handle' and 'Copybook File'. The table is currently empty.
- Buttons: Add, Delete, Help, Back, Finish, Cancel

- 11** [Add] ボタンをクリックします。実行するプログラムの COBOL コピーブックを検索および選択します。デフォルトでは、コンポーネント内部のハンドル名としてコピーブック名が割り当てられます。または、「Copybook Handle」列を上書き変更して任意の名前を割り当てることもできます。ファイルを削除するには、エントリを選択して [Delete] をクリックします。

**注記：** CICS RPC Component Editor は、プログラムの入出カインタフェースあるいは異なる入出力定義が必要なインタフェースの両方のインタフェースとしてコピーブックを使用する複数の ECI リクエストまたはホストプログラム、あるいはその両方を処理するよう設計されています。これらのスタイルはすべて Composer と完全な互換性があります。



- 12** [Finish] をクリックします。すると、コンポーネントが作成され、CICS RPC コンポーネントエディタが表示されます。

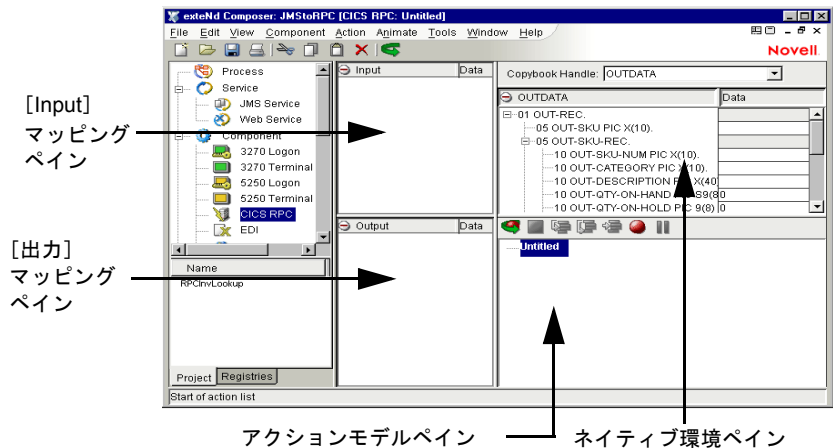


コピーブック  
説明

## CICS RPC コンポーネントエディタウィンドウについて

CICS RPC 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すべての機能が含まれます。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。

ただし、CICS RPC コンポーネントエディタにはすべての接続に共通なネイティブ環境ペインも含まれているという点で異なります。ネイティブ環境ペインには、CICS RPC コネクタに固有な初期環境ペインが表示されます。



## CICS RPC ネイティブ環境ペインについて

RPC ネイティブ環境ペインは、実際の COBOL コピーブックをシミュレートします。このペインからは、次のことを実行できます。

- ◆ 入力XMLドキュメント(または他の使用可能なDOM)からデータを取得し、そのデータを適切なコピーブックフィールドにドラッグして実行するプログラムの入力データとして使用します。
- ◆ ネイティブ環境ペインでマウスを右クリックすると、メニューオプションのリストにアクセスできます。ビューの選択では、データを表示するための3種類のオプションが用意されています。[Copybook] ビューまたは [Copybook + Data] (デフォルト) に加えて [Hex Data] を選択できます。メニューおよびツールバーの詳細については、(『*Composer User ユーザガイド*』を参照してください)。

**注記：** データは、入力 DOM および出力 DOM で編集されるため、このペインでは編集できません。

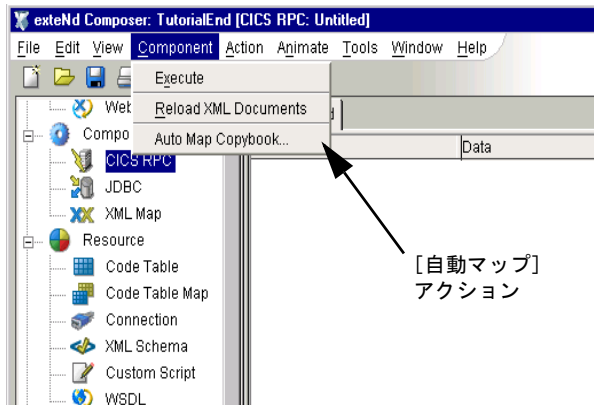
**注記：** コピーブックが単一データ記述子を指定しないで作られている場合 (たとえば、レベル2の項目がいくつかあり、レベル1の項目がない)、Composer はネイティブ環境ペインに表示されているコピーブックの最上位にレベル1のノードを作成します。

## CICS RPC 専用メニューバー項目について

CICS RPC Connect には、Composer では提供されていないアクションが含まれています。そのアクションの1つが Auto Map Copybook です。このアクションによって自動的に入力 / 出力ドキュメントを作成したり、マッピングアクションを作成したりできます。

## [自動マップコピーブック]

メニューバーから、[Component/Auto Map Copybook] を選択します。このオプションを選択すると、自動的に入力 / 出力ドキュメントを作成したり、マッピングアクションを作成したりしてデータをドキュメントとコピーブックの間で移動できます。または、生成されたドキュメントや Auto Map に基づいて XML テンプレートを作成できます。コピーブックから入力ドキュメントおよび出力ドキュメントを作成する詳細については、次の節を参照してください。



## CICS RPC 専用コンテキストメニュー項目について

ネイティブ環境ペイン内でマウスを右クリックするとコンテキストメニューが使用できます。一部のコマンドは、このペイン内の特定の品目について作業するときのみ使用できます。メニューの各項目とどの時点で有効にすることができるかについては、次の表で説明します。

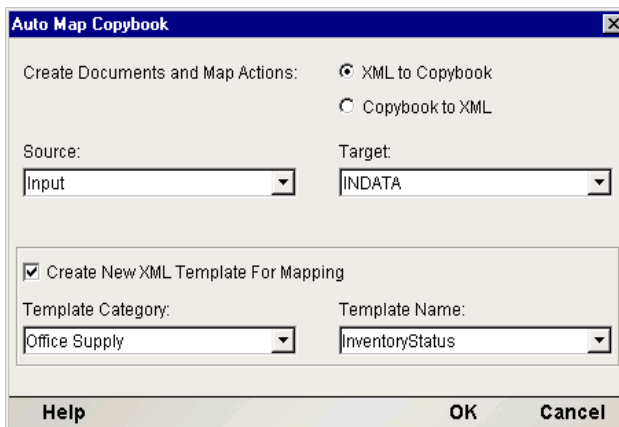
ネイティブペイン メニュー	[Description]
[Decision]	コピーブックの REDEFINE ステートメントが選択されていれば、有効に設定されています。 [Decision] をクリックすると、[Redefined] データ記述子を利用する時点を決断するための [Decision] 式を入力するよう求めるダイアログボックスが表示されます。
スレーブサーバの 場合は、	コピーブックの OCCURS ステートメントが選択されていれば、有効に設定されています。 [Repeat] をクリックすると、[Repeat] アクションのターゲットを指定するための情報を入力するよう求めるダイアログボックスが表示されます。

ネイティブペイン メニュー	[Description]
Map	入力ペインまたは出力ペインでアクティブです。ステートメントを選択して [Map] をクリックすると、情報を入力するよう求めるダイアログボックスが表示されます。
[View] - [Copybook]	ネイティブ環境ペインでアクティブになります。ステートメントを選択して [View/Copybook] をクリックすると、ペインにはコピーブックが表示されます。
[View] - [Copybook +Data]	ネイティブ環境ペインでアクティブとなり、デフォルトとして設定されています。ステートメントを選択して [View/Copybook + Data] をクリックすると、ペインにはコピーブックとコピーブックにマッピングされたあるいはプログラム実行の出力としてコピーブックに配置されたデータが表示されます。
[View Hex Data]	ネイティブ環境ペインでアクティブになります。ステートメントを選択して [View/Hex Data] をクリックすると、ペインにはコピーブックにマッピングされた 16 データあるいはプログラム実行の出力としてコピーブックに配置されたデータが 16 進数の形式で表示されます。
[Select Occurrence]	コピーブックの OCCURS ステートメントが選択されていれば、有効に設定されています。OCCURS 句に含まれているフィールドのデータの各オカレンスは表示されないため、[Select Occurrence] をクリックすると、フィールドのデータのどのオカレンスを表示するかを選択するよう求めるダイアログボックスが表示されます。1 から順に番号を入力してください。
[Expand Tree]	選択したデータ記述子の下にあるすべてのコピーブックノードが表示されます。
[Collapse Tree]	選択したデータ記述子の下にあるすべてのコピーブックノードを非表示にします。
[Copy]	[View/Hex Data] フォーマットで有効となります。コピーする一連のテキストを選択し、コピーして貼り付けることができます。
[Print Copybook]	コピーブックを印刷できます。
[Find]	すべてのデータビューのコピーブック内を検索できます。
[Find Next]	すべてのデータビューの [Find] ダイアログに入力されたテキストの次のオカレンスを検索できます。

# Auto Map を使用してコピーブックから入力サンプルドキュメントを作成

Auto Map Copybook を使用すると、ホストプログラムとのコピーブックインタフェースに基づいてデータの入力/出力XML サンプルドキュメントを作成できます。また、サンプルドキュメントとコピーブックの間のマッピングアクションを作成することもできます。または、サンプルドキュメントの XML テンプレートを作成し、再利用する場合に備えて常時保存しておくこともできます。

## ➤ 入力サンプルドキュメントを作成する



- 1 exteNd Composer から、[Component/Auto Map Copybook] を選択します。>すると、[Automap Copybook] ダイアログボックスが表示されます。
- 2 [XML to Copybook] を選択することは、入力ドキュメントおよびマッピングアクションを作成してデータをコピーブックに移動することを示します。
- 3 プルダウンリストで [Source (Input)] を選択します。ソースにはコンポーネントで使用できる DOM(入力、出力など)が含まれます。
- 4 ドロップダウンリストから [Target (Copybook name)] を選択します。ターゲットリストには、コンポーネントで利用可能なコピーブックが含まれています。
- 5 チェックボックスの [Create New XML Template For Mapping] をクリックすると、このコンポーネントおよび自分で作成する他の追加コンポーネントで利用するためにサンプルドキュメントを使用して常時 XML テンプレートを作成することを意味します。この機能を有効にすると、[Template Category] と [Template Name] がアクティブになります。
- 6 Select a デフォルトのカテゴリと異なる場合は、[Template Category] を選択するか、既存のエントリの上に [New Template Category Name] を上書きします。

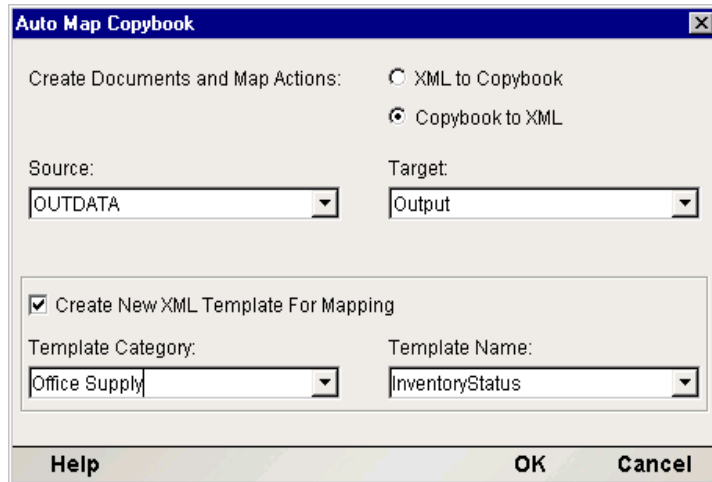
- 7 XML テンプレートのリストから [Template Category] を選択するか、既存のエントリの上に [New Template Category Name] を上書きします。
- 8 [OK] をクリックします。コンポーネントエディタウィンドウに、作成した入力テンプレートとアクションモデルペインに追加された Map アクションが表示されます。

**注記:** ホストプログラムで入力および出力用に単一のコピーブックのみを使用する場合は、引き続き Auto Map 機能を使用できますが、自動的に作成されてしまった入力とは関係のないアクションを削除できます。

## Auto Map を使用してコピーブックから出力サンプルドキュメントを作成

この画面では、出力ドキュメントを作成したり、マッピングアクションを作成したりしてデータをコピーブックとの間で移動できます。または、生成されたドキュメントや Auto Map コピーブックに基づいて XML テンプレートを作成できます。

**注記:** これらのアクションは通常「Execute ECI」アクションの後で実行されます（次の章を参照）。



### ➤ 新規の出力サンプルドキュメントを作成する

- 1 exteNd Composer から、[Component/Auto Map Copybook]> を選択します。すると、[Automap Copybook] ダイアログボックスが表示されます。
- 2 [Copybook to XML] を選択することは、出力ドキュメントおよびマッピングアクションを作成してホストプログラム出力データをコピーブックから出力ドキュメントに移動することを示します。



- 3 ドロップダウンリストから [Source(Copybook)] を選択します。ドロップダウンリストにはコピーブックのリストが含まれます。
- 4 プルダウンリストで [Target(Output)] を選択します。ターゲットリストには、コンポーネントで利用可能な DOMS が含まれています。
- 5 チェックボックスの [Create New XML Template For Mapping] をクリックすると、このコンポーネントおよび自分で作成する他の追加コンポーネントで利用するためにサンプルドキュメントを使用して常時 XML テンプレートを作成することを意味します。この機能を有効にすると、[Template Category] と [Template Name] がアクティブになります。
- 6 デフォルトのカテゴリと異なる場合は、[Template Category] を選択するか、既存のエントリの上に [New Template Category Name] を上書きします。
- 7 XML テンプレートのリストから [Template Category] を選択するか、既存のエントリの上に [New Template Category Name] を上書きします。
- 8 [OK] をクリックします。コンポーネントエディタウィンドウに、作成した出力テンプレートとアクションモデルペインに追加された Map アクションが表示されます。

**注記:** ホストプログラムで入力および出力用に単一のコピーブックのみを使用する場合は、引き続き Auto Map 機能を使用できますが、自動的に作成されてしまった出力とは関係のないアクションを削除できます。

## 入力テンプレートおよび出力テンプレート手動で作成

コピーブック内のDFHCOMMAREAから利用可能な一部のデータだけを使用したコンポーネントを作成する場合、手動でサンプルドキュメントを作成することができます。ネイティブペインの相互動作は、他のすべてのコンポーネントエディタと同じです。アプリケーションに応じて入力ドキュメントと出力ドキュメントのいずれか、あるいはその両方に対してこの方法を利用できます。

### ➤ 手動で XML テンプレートを作成する

- 1 既存の XML ドキュメントを検索するか、外部の XML エディタを使用してサンプルドキュメントを作成します。
- 2 新規の XML テンプレートを作成し、作成したドキュメントをサンプルとして追加します。
- 3 前と同じ方法でコンポーネントを作成しますが、入力/出力テンプレートとして [System] と [Any] を選択する代わりに、自分で作成したテンプレートの名前を選択します。
- 4 前と同じように、同じコピーブックを追加します。

- 5 すると Component Window が表示され、入力 DOM からコピーブックに要素をドラッグアンドドロップすると Map アクションが自動的に表示されます。を追加できます。ドラッグアンドドロップで自動マップ機能を使用する場合のマップ画面は次のように表示されます。

The screenshot shows a 'Map' dialog box with the following configuration:

- Source:** XPath: `input`, Expression: `OUTDATA.getField("OUT-SKU")`
- Options:** Content Editor..., Code Table Map..., Advanced... (all unchecked)
- Target:** XPath: `Output`, Expression: `OUT_REC/OUT_SKU`

**注記:** (詳細については、『Composer ユーザガイド』の「Creating a New XML Template」を参照してください)。

# 4

## CICS RPC 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。

CICS RPC 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、ホストと 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、データをホストインベントリ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します。

このアクションモデルの例は、いくつか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。

- ◆ 請求書のドキュメントを開き、CICS RPC コマンドを実行してホストデータベースから請求書のデータを取得する
- ◆ 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

### CICS RPC 専用アクションについて

CICS RPC には「Execute ECI」と呼ばれる固有のアクションがあります。メニューバーからこのアクションにアクセスするには、[Action Menu/New Action/Execute ECI] を順にクリックします。次のダイアログボックスが表示されます。

このダイアログの使用方法を次に示します。

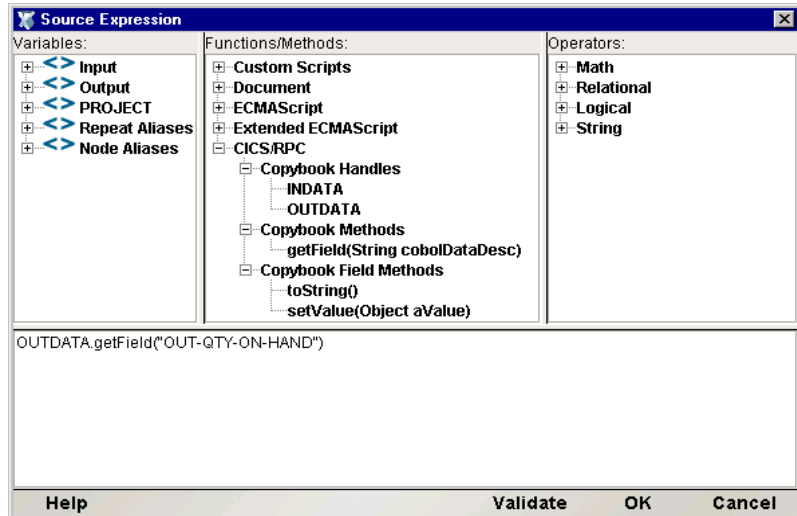
#### ▶ Execute ECI アクションを実行する

- 1 「ホストプログラム名」を入力します。この例では、SKUT02 がプログラム名です。
  - 2 チェックボックスの内部をクリックして CICS トランザクション ID のフィールドエントリをアクティブにします。このフィールドはオプションで、コンピュータまたはユーザの CICS 環境、あるいはその両方でホストプログラム名だけでなく トランザクション ID も必要な場合にのみ入力する必要があります。
- 注記：** 画面に代表的な トランザクション ID が入力されました。トランザクション ID を入力するためチェックボックスをクリックしたがそのフィールドに情報を入力できなかった場合、Execute ECI アクションを実行できません。
- 3 プルダウンリストで [Source Copybook] を選択します。この例では、INDATA が [Source Copybook] です。
  - 4 プルダウンリストで [Target Copybook] を選択します。この例では、OUTDATA が [Target Copybook] です。
  - 5 [OK] をクリックします。これでコンポーネントのアクションモデルに Execute ECI アクションが含まれました。

## CICS RPC 専用の Expression Builder 拡張

Expression Builder 拡張の目的は、DOM と Copybook の間でドラッグアンドドロップをして Map アクションを作成する際、自動的に実行される機能として ECMAScript プログラム以外の方法を提供することです。Expression アイコンをクリックすると、[Source] の [Expression] ダイアログボックスが表示されます。

これらの ECMAScript Extensions は、ユーザ独自の [Auto Map] をカスタマイズするときに利用できます。CICS RPC コンポーネントで使用される各コピーブックは、ECMAScript オブジェクトとして表されます。これらのコピーブックで使用可能なメソッドは次のとおりです。getField().getField メソッドは、Copybook Field オブジェクトを戻します。Copybook Field オブジェクトには次に示す 2 つのメソッドがあります。



- copybookField getField(String cobolData Desc)

Novell exteNd 拡張メソッド

CopybookField オブジェクトを戻します。

例:

コピーブック内で名前 cobolDataDesc が重複しているのを解決するには、次のように親 cobol Data Desc を参照します。

getField( 撤 ARTID IN INDATAi) は次のコピーブックに指定されています。

```
01COMMAREA
05 INDATA
10PARTID
05 OUTDATA
10PARTID
```

戻された Copybook Field オブジェクトには 2 つのメソッドがあります。

- String toString()

Novell exteNd 拡張メソッド

CopybookField オブジェクトの文字列値を戻します。

- void setValue(Object aValue)

Novell exteNd 拡張メソッド

CopybookField オブジェクトの値を設定します。

## RPC コンポーネントエディタでの他のアクションの使用

すでに説明したアクションに加えて、RPC 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)。

基本的なアクションを次に示します。

基本的なアクション	説明
コメント	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。
コンポーネント	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しするランタイム 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。
機能	ECMAScript スクリプト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
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。次に示す 3 つのログタイプがあります。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に使用できる、基本的に便利なアクションです。

高度なアクションは、コンポーネントエディタの [Action] メニューで、[Advanced] の順にサブメニューを選択すると利用できます。アクションモデルで右クリックすると、コンテキストメニューと同じコマンドが表示されます。

高度なアクションは、次のものから構成されます。

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(真)の場合は、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 配備の一部のコンポーネントで <i>setRollbackOnly</i> を呼び出したりできます。
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。

次の表は、Data Exchange アクションの一覧です。

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。

Data Exchange アクション	説明
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。

Repeat アクションには、任意のアクションモデルで使用する制御フロー構成要素が含まれています。

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 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

この節では、アニメーションツールの実行時に発生する可能性がある一般的なエラーメッセージについて説明します。

### ECI 接続エラーメッセージ

例外メッセージ



- dt002001: Error Executing Component. ECI Exception  
java.io.IOException:CCL6651E: ゲートウェイに接続できません  
[address = tiger,port = 2006]  
[java.net.UnknownHostException: tiger]
- ネストしたメッセージ: rtCICSRPC000301:ECIException java.io.IOException:  
CCL6651E: ゲートウェイに接続できません  
[address = tiger, port = 2006]  
[java.net.UnknownHostException: tiger]
- ネストしたメッセージ: CCL6651E: ゲートウェイに接続できません  
[address = tiger, port = 2006]  
[java.net.UnknownHostException: tiger]

**注記:** このエラーは、ECI 接続用に使用されたホスト名が正しくないか、または実行中の環境に存在していない、あるいはその両方の場合があることをユーザに示しています。問題を解決するには、接続ダイアログのパラメータ、特にホスト名 IP アドレスを確認します。

## データ型変換エラーメッセージ

### 例外メッセージ

- dt002701: Error trying to map 'TESTALL.getField("ESIGNED")'
- ネストしたメッセージ: rt001002:\*\* Error evaluating  
`TESTALL.getField("ESIGNED").setValue(Input.XPath("COMMAREA/EXT  
ERNAL  
DECIMAL/ESIGNED"))`
- ネストしたメッセージ: Runtime error Error in Java method set Value

例外により発生:

Error converting source data type to the selected copybook field.

**注記:** これは、ユーザが COBOL コピーブックのターゲットフィールドのデータ型に準拠していない値を入力したときに出力される、もっとも代表的なエラーメッセージです。例をあげると、COMP、COMP-3、あるいは 9999999 などの数値データタイプのフィールドに、LOR8437 のような値が入力された場合です。



# A

## サポートされるデータ型の一覧

**注記：** CICS RPC の PICTURE 句でサポートされる COBOL データ型の一覧は次のとおりです。

- ◆ 英字項目
- ◆ 数値項目
- ◆ 数字編集項目
- ◆ 英数字項目
- ◆ 英数字編集項目
- ◆ 指数以外の外部浮動小数点項目



# B

## サポートされるデータ型の文字一覧

CICS RPCの PICTURE 句でサポートされるデータ型の文字の一覧は次のとおりです。

PICTURE シンボルには次のものが含まれます。

A B E P S V X Z 9 0 / , . + - C R D B \* \$

これらの文字は、データ型を編集するときに使用されます。これらの文字の意味については、COBOL の参照マニュアルに明記されています。



# C

## アニメーションテストと配備テストの環境的相違

### テスト

CICS RPC でのテストと、配備テストには、大きな環境的相違があります。考慮すべき最も重要な点は、設計時のテスト用に使用していた CICS サーバとの接続ポイントは、必ずしも運用環境で使用するものと同じではないということです。さらに、すべてのコンポーネントに対して適用される基本的な環境的相違はすべて注意する必要があります。詳細については、『exteNd Composer ユーザガイド』の「環境的相違」を参照してください。





# D

## Java でサポートされるコードページの一覧

Novell exteNd Composer の文字エンコーディング変換機能は、使用されている Java VM と密接な関係があります。サポートされる文字エンコーディングは、Java 2 プラットフォームの実装によって異なります。最新のエンコーディングの詳細については、Web サイト <http://java.sun.com> を参照してください。

Sun's Java 2 Runtime Environment for Windows には、米国版と国際版の 2 種類があります。国際版 (**lib\i18n.jar** ファイルを含む) では、すべての文字エンコーディングがサポートされます。



# E

## CICS RPC 用語集

### CICS

Customer Information Control Systemの略。IBM ライセンステレプロセッシングサブシステムで、端末からのオンライン入力を受け取り、ユーザの要求に応じてプログラムを実行し、さらにバッファ、ストレージ、ファイルの入出力管理を行います。

### CICS RPC

CICS RPCは、Customer Information Control System Remote Procedure Callを表します。

### COBOL

Common Business Oriented Language の略で、主としてレガシーアプリケーションで使用されています。

### COMMAREA (DFHCOMMAREA)

通信領域を表します。これはコマンドレベルの機能で、1つのトランザクション内の2つのプログラム間で情報を交換したり、同一の端末からの2つのトランザクション間で情報を交換したりします。

### EBCDIC

IBM メーンフレームで使用される照合シーケンス。EBCDIC 照合シーケンスでは、小文字、大文字そして数字の順に表示されます。

### Endian

コンピュータメモリに一連のバイトを格納する順序を説明するために使用される用語。ビッグエンディアン (Big-endian) は、シーケンス内の最上位バイトがストレージの下位アドレスに最初に格納されます。リトルエンディアン (Little-endian) は、シーケンス内の最下位バイトがストレージの下位アドレスに最初に格納されます。

## コピーブック

プログラムまたはコピーブックで個別に定義された COBOL データ記述子から構成されるレコード構造 ( コピーブックを他のコピーブック内部に含めることができます )。コピーブックを他のプログラムまたはコピーブックで再利用することもできます。

## ゲートウェイパラメータ

接続のチューニングを行うには、次の 2 つのパラメータがあります。パラメータは、*Initworker* と *Initconnections* です。*Initworker* は初期化された作業スレッドの数です。*Initconnections* は初期化された接続の数です。

## ネイティブ環境ペイン

クエリーを発行する際に実際の CICS 環境をシミュレートする CICS RPC コンポーネントエディタ内のペイン。

# 索引

## 数字

- 16 進データ 28
- 16 進数形式 30

## A

- [Auto Map Copybook] 31
- Auto Map を使用してコピーブックから入力ドキュメントを作成 31

## C

- CICS 18, 51
- cicscli.ini 20
- CICS Java ゲートウェイ 11
- CICS RPC Connect 10
- External Call Interface 10
- CICS RPC コンポーネント  
概要 12
- CICS RPC コンポーネントエディタ 13
- CICS RPC 専用メニューバー項目 28
- CICS コンポーネント 23
- CICS コンポーネントエディタウィンドウ 27
- COBOL 43, 51
  - コピーブック 26

## D

- Data Exchange アクション 39
- DFHCOMMAREA 10, 11, 15

## E

- EBCDIC 11, 20
- ECI CICS RPC 接続 16
- ECI Connection Error Messages 41
- ECI 接続エラーメッセージ 40
- ECI 接続タイプ 20
- ECMAScript 36
- Endian 20, 51

- Expression Builder 36
- Expression Builder 拡張 36
- eXtend CICS RPC Connect 10
- eXtend Composer 9

## G

- getField(). 37

## I

- Initconnection 16
- Initworker 16

## N

- Novell 9

## O

- OCCURS 12, 29
- [OCCURS] 30

## R

- REDEFINE 29
- Repeat アクション 40

## S

- SQL アクションの実行  
定義 51

## T

- TCP/IP ポート 16

## X

- XML テンプレート  
新規作成 31
- XPath() メソッド 18

アクション 35  
専用 35

## え

エラーおよびメッセージ 40  
エラーおよびメッセージの処理 40

## か

カスタムスクリプト 21

## き

基本的なアクション 38

## く

クエリ / 結果マッピングペイン 28, 32

## こ

高度なアクション 39  
コードテーブルマップ 21  
コードページ 20  
コードページサポート 20  
コピーブック  
XML 32  
コンポーネントに対する XML テンプレートの  
作成 21

## し

自動マップ 21  
[自動マップコピーブック] 29  
手動での XML テンプレート 33

## せ

接続プール  
定義 51, 52  
接続リソース 16

## て

定数駆動型および式駆動型の接続 17  
データ型変換エラーメッセージ 41  
テンプレート  
[分類] 24  
名前 24

## ね

ネイティブ環境ペイン 52, 27

## は

パフォーマンスパラメータおよび接続リソース 16  
パラメータ 15  
式ベース 17  
定数駆動型 17

## ふ

複数の ECI リクエスト 26  
浮動小数点フォーマット 20

## ま

マッピング 31

## め

メール 12

## も

文字エンコード 20

## れ

例外メッセージ 40

