

Novell exteNd Composer™ HTML 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、責任をもちません。

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、責任をもちません。

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もちません。

本ソフトウェアは、著作権をもつ 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 © [Sdate-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、商標またはそのた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

HTML 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 HTML User Interface へようこそ	9
はじめに	9
exteNd Connect について	10
HTML Connect とは	11
Composer の HTML コンポーネントの概要	12
SSL (Secure Sockets Layer) の概要	13
クッキーの概要	13
クッキーサポートの概要	14
フレームの概要	14
JavaScript の概要	15
HTML User Interface 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	15
2 HTML コンポーネントエディタをお使いになる前に	17
サンプルトランザクション	17
HTML コンポーネントの作成に使用される一般的な手順	17
設計時環境をセットアップする	18
適した HTTP ハンドラを選択する	18
プロキシサーバの設定	19
HTML の接続リソースについて	22
HTTP 認証のタイプ	22
HTTP 基本認証リソース	23
HTTP ダイジェスト認証の接続リソース	26
HTTP NTLM 認証の接続リソース	26
コンポーネント用 XML テンプレートの作成	28
3 HTML コンポーネントの作成	29
HTML コンポーネントを作成する前に	29
HTML コンポーネントエディタウィンドウについて	31
HTML ネイティブ環境ペインについて	32
ScreenDoc DOM について	32
HTML 固有のメニューバー項目について	35
HTML 固有のボタンについて	36
4 HTML アクションの実行	37
アクションについて	37
HTML セッションの記録	38
フレームを使用した HTML セッションの記録	43
HTML アクションの編集	46
以前に記録したアクションモデルの編集	48

既存のアクションの変更.....	49
新しいアクションの追加.....	51
アクションの削除.....	54
HTML コンポーネントの実行.....	54
アニメーションツールの使用.....	55
HTML コンポーネントエディタでの他のアクションの使用.....	61
XML Interchange アクションの使用.....	63
パフォーマンス.....	65
エラーおよびメッセージの処理.....	66
A デジタル証明書	69
B テスト	75
アニメーションテストと配備テストの環境的相違.....	75
C HTTP ステータスコード	77
詳細コードの意味.....	77
D フォームフィールド値をインタラクティブに変更した場合に作成されたアクション	81
E 記録済みの Function アクションで使用される内部スクリプト	83
F HTML 用語集	87
G 予約語	91

このガイドについて

目的

このガイドでは、exteNd Composer HTML Connect の設計時アプリケーションの一部である HTML 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、セキュリティ保護されていない Web ページ、または HTTP 基本認証、ダイジェスト認証、または NTLM のセキュリティプロシージャを使用してセキュリティ保護されている Web ページにアクセスできる Web ページ「スクリーンスクレーパ」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。HTML、HTTP セッションの動作、HTTP のエラーメッセージ、および関連する概念についても理解していることが必要です。

追加のドキュメント

Novell exteNd Director に関する完全なドキュメンテーションについては、次の Novell マニュアルの Web サイトを参照してください。

<http://www.novell.com/documentation-index/index.jsp>

1

exteNd Composer および HTML User Interface へようこそ

はじめに

『Connect for HTML ユーザガイド』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、HTML 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。

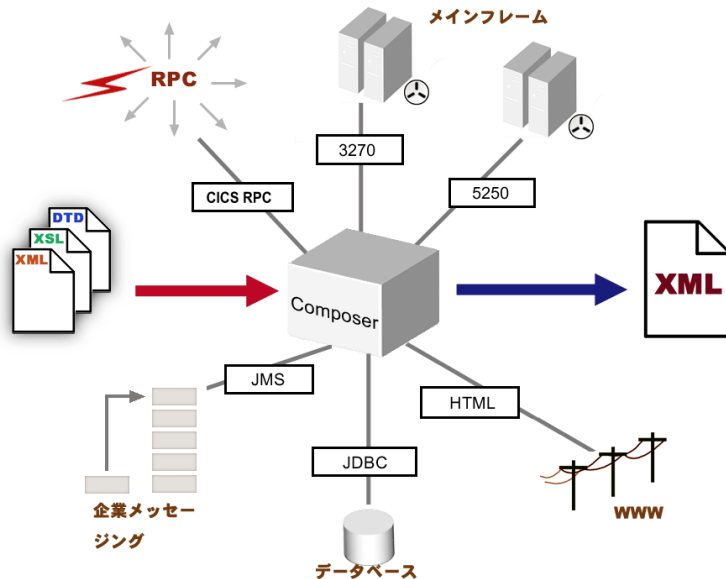
exteNd Composer を使用しており、コアコンポーネントエディタである XML Map コンポーネントエディタに精通している場合は、このガイドに従って HTML コンポーネントエディタを簡単に使用することができます。

作業を始める前に、まず HTML 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、このコネクタのサーバ側ソフトウェアが Composer Enterprise Server にインストールされている必要があります。

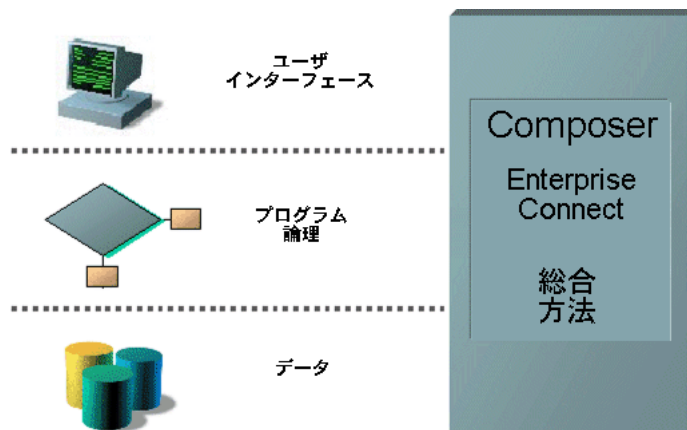
注記： このコンポーネントエディタを正しく使用するには、HTML 環境と、XML に対応させるアプリケーションに慣れ親しんでおく必要があります。

exteNd 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 に対応したその他のサービスを使用するか、使用せずに) このようなドキュメント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、次に示すようにレガシー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。



Composer 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。(次の図を参照)。Novell exteNd 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。

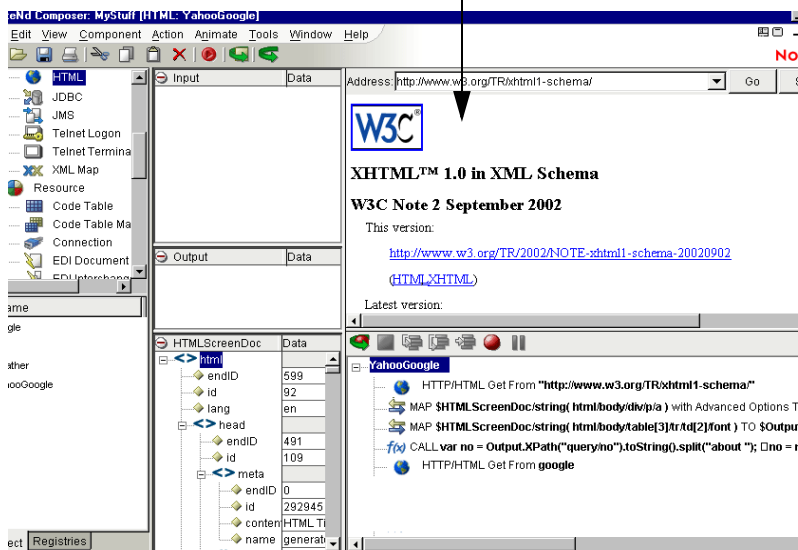


HTML Connect とは

Composer の HTML Connect は、後で B2B サービスとして再生するために、HTML 情報ストリームにフッキングすることにより、ユーザーインターフェース統合を使用して標準 Web サイトとのレコードナビゲーションと統合が可能になります。HTML(Hyper Text Markup Language) という用語は、World Wide Web で公開されるドキュメントに使用されます。HTML Connect は、読みやすく、時には構文上に誤りのある HTML 情報を明確にレイヤされた XHTML ドキュメントに変換する Tidy というユーティリティを使用します。

注記: HTML Connect は、JavaScript(エラーメッセージなしで)、Flash、アプレットのみならず ActiveX コントロールも処理しません。

HTML ページは、ネイティブ環境ペインに表示される



Composer の HTML コンポーネントの概要

XML Map コンポーネントと同様に、XML 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、インターネットまたはエクストラネットアプリケーションに接続して、Web ページにより表示し、DOM 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、統合アプリケーションに適切な方法で、出力 DOM に基づいて処置を取ることができます。

HTML コンポーネントでは、XML ドキュメントから HTML 形式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。HTML 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。

SSL (Secure Sockets Layer) の概要

HTML Connect は exteNd Composer とターゲットの Web サイトの間の安全な接続を行います。このセキュリティは、クライアントおよびサーバ側のデジタル認証をサポートする SSL3 (Secure Socket Layer Support 3) Composer を使用して達成されます。HTML Connect は HTTP 基本認証接続を使用して、インターネットのサイトと対話するセキュリティをセットアップします。SSL プロトコルは、ブラウザとサーバ間の認証され、暗号化された通信に許可されます。このセキュリティを有効にするために、Composer アプリケーションは、Web サイトにアクセスし、Web サイトにより要求されるデジタル証明書を処理できなければなりません。このことは、証明書が Web サーバ (Novell、WebLogic、または Websphere) だけでなく HTML Connect にインストールされなければならないことを意味します。このために、Composer は 2 つの方法をもっています。サーバ側の証明書の場合、HTML Connect は、Designer/lib ディレクトリの jar ファイル *xerootca.jar* に 98 個の業界標準デジタル証明書が入れられて出荷されます。

含まれていないサーバ側の証明書が必要な場合は、証明書の保守の方法についてアプリケーションサーバの資料を参照してください。サポートされている証明書のリストと Composer に新しいサーバ側証明書を追加する方法については、付録 B も参照してください。クライアント側の証明書の場合、Composer は DER の暗号化されたバイナリの x509 証明書をサポートします。証明書を関連付けるために、Composer は HTTP 基本認証接続リソースでの証明書の指定を許可します。

クッキーの概要

クッキーはブラウザが格納するテキストファイルです。クッキーの主な目的の 1 つは、ホストに対してユーザ (HTTP クライアント) を識別することです。たとえば、Web サイトのページ間で移動するときにログオンステータスを維持したり、Web サイトのショッピングカートを使用してショッピングするときに最新のセッションを維持したりする場合です。クッキーは、接続のクライアント側の情報を格納および検索するためにサーバ側が使用できる一般的なメカニズムです。クッキーには 2 つの種類があります。セッションの期間中のみアクティブなセッションクッキーおよびコンピュータのディスクに保管され、そのサイトにアクセスするたびに選択可能になる持続的クッキーです。

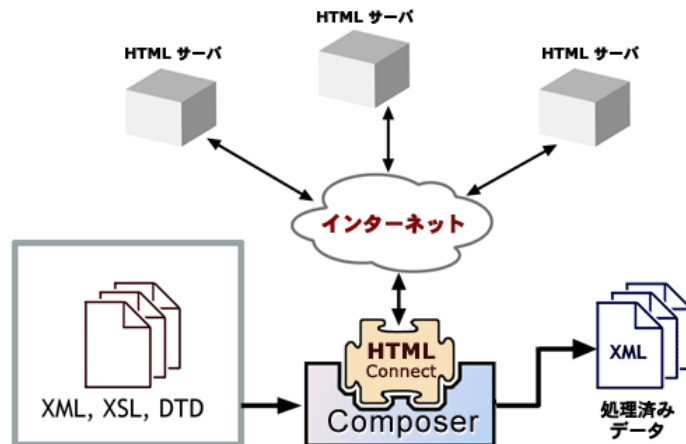
クッキーサポートの概要

HTML Connect はセッションクッキーのみをサポートします。つまり、サイトにより要求される場合、HTML Connect はコンポーネントがアクティブになるたびにクッキーを作成します。クッキーは一度セットされ、コンポーネントの実行が終了すると消滅します。テスト中に、コンポーネントのアニメーションまたは実行を再開すると、実行が始まる前にクッキーが破棄され、コンポーネントの実行のたびに新しいセッションとして取り扱われます。持続的クッキーはサポートされておらず、クライアント側のクッキーの配備もサポートされていません。

フレームの概要

このバージョンの HTML Connect はフレームをサポートしています。HTMLScreenDoc には、ロードされたページの DOM が含まれます。HTML パネルをクリックするか、情報をフレームの1つの入力制御にドラッグアンドドロップすると、HTMLScreenDoc にこのフレームの DOM が表示されます。フレームがこのとき変更されると、新しい SetFrame アクションが作成され、Map アクションの DOM に記録されます。

フレームをもつページを発見すると、(ページの生の HTML ソースコードの検査を通して) 対象のターゲットフレームに関連付ける URI を決め、ネイティブ環境ペインのアドレス行の URI を使用することをお勧めします。HTML コードのソースコードを表示するには、メニューバーから [View]、[HTML Text] の順に選択します。ページの FRAMESET タグを検索し、各フレームのターゲット URI を判別します。



JavaScript の概要

HTML Connect は Web ページ内でのスクリプト (JScript または JavaScript) の実行をサポートしていません。HTML Component Editor で多くのスクリプトを含むページを表示したり、操作したりすることはできますが、イベントハンドラにスクリプトが添付されたページ要素をクリックしようとした場合、またはスクリプトが別の URI にリダイレクトしようとした場合、HTML Connect はスクリプトを実行しません。代わりに、66 ページに示されたような画面が表示されます。

注記： 設計時に、どのターゲットページ、およびどのページのどの要素が、JavaScript が埋め込まれているために問題を発生させる可能性があるかを (テストを通じて) 判別し、これらのページまたは要素、あるいはその両方を避けるようにしてください。

HTML User Interface 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

HTML User Interface 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、HTML ベースの操作をサポートするビジネスアプリケーションをどれでも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。HTML コンポーネントエディタを使用すると、動作中の Web サイト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

HTML 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

説明を目的として、このドキュメントでは、サンプル中で1つのトランザクションについて取り扱います。ある Web サイトに移動し、SKU 番号をフォームに入力して問い合わせます。オプションを変更して、Web サイトの画面からさまざまな情報を取得します。HTML 形式の Web ページの詳細は、ScreenDoc DOM というオブジェクトにある Web ページの XHTML 表現を表示することにより、見ることができます。選択操作およびその結果は、アクションペインで見られます。サービスを保存する前に、アクションを変更したり編集したりすることもできます。

HTML コンポーネントの作成に使用される一般的な手順

HTML コンポーネントを作成する方法はいくつもありますが、単純な HTML コンポーネントの作成に使用される一般的な手順は次のとおりです。

- 1 HTML トランザクションでの入出力に必要な XML テンプレートを作成します。
- 2 セキュリティ保護されたサイトにアクセスする場合は、必要なセキュリティ情報をすべて盛り込んでいる接続リソースを作成します。
- 3 HTML コンポーネントを作成します。
- 4 記録モードを入力し、キャプチャしたい情報のある Web ページに移動します。
- 5 情報を入力 DOM から Web ページにドラッグアンドドロップして、フォーム操作を実行します。次に、Web ページから返された結果を、出力 DOM にドラッグアンドドロップします。
- 6 必要に応じてアクションを編集します。
- 7 コンポーネントを実行して、テストします。
- 8 コンポーネントを保存します。

設計時環境をセットアップする

Composer 環境が Web ページにアクセスして表示できるように、適切にセットアップされていることを確認してから、HTML コンポーネントを設計してください。そのためには、適した HTTP ハンドラを選択し、必要なら、プロキシを設定します。

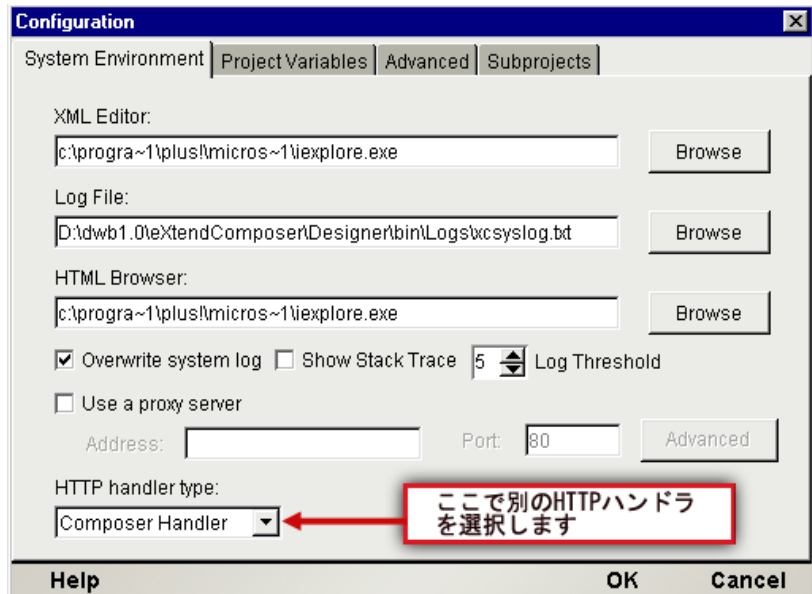
適した HTTP ハンドラを選択する

Novell exteNd Composer では、HTML コンポーネントをテストするために、別の HTTP ハンドラを選択して設計時に使用できます。これは、Composer の [Native Environment] ペインで HTML を表示する仕方に、影響することがあります。(つまり、設計時に HTML エディタを使っていると、メモリエラーまたは表示トラブルが発生する場合は、次に説明しているとおおり、別の HTTP ハンドラを選択しなければならないことがあります。)

注記： これは、設計時のみにおける問題です。サービスがサーバ上で移動し始めるとすぐに、アプリケーションサーバが HTTP ハンドラを自動的に選択するので、画像表示がコンポーネントのパフォーマンスを左右する要因になることはありません。この設定は、配備には影響しません。

➤ HTTP ハンドラを選択する

- 1 Composer の [Tools] メニューで [Configuration] を選択して、[Configuration] ダイアログボックスを表示します (次を参照)。



- 2 [System Environment] ペインで、[HTTP handler type] というプルダウンメニューに行きます。HTML コンポーネントの設計時テスト用に、[HTTP Handler] または [Default JRE Handler] を選択します。設計時に [Native Environment] ペインで画像が正しく表示されない場合は、ここから別のタイプを選択してみてください。
- 3 [OK] をクリックしてダイアログボックスを閉じます。

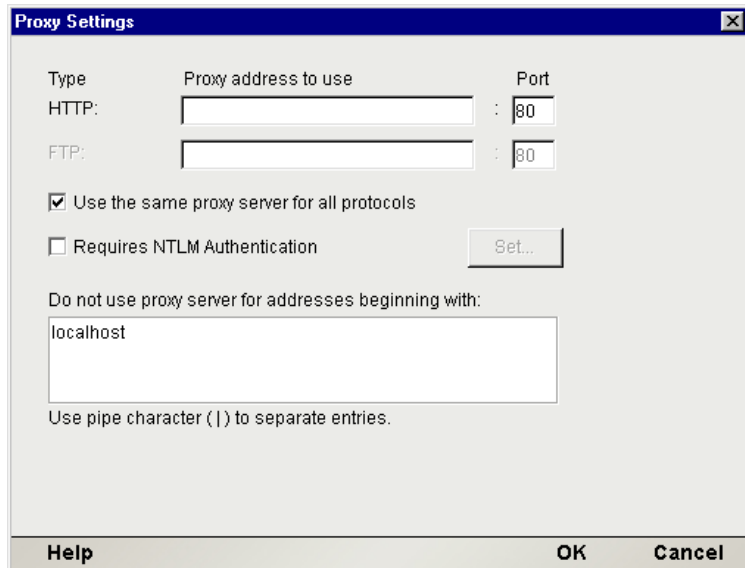
プロキシサーバの設定

Web ユーザが Web サイトにアクセスするには、プロキシサーバを経由することになっている場合は、Composer がプロキシを「通り抜け」られるように、プロキシを設定する必要があります。

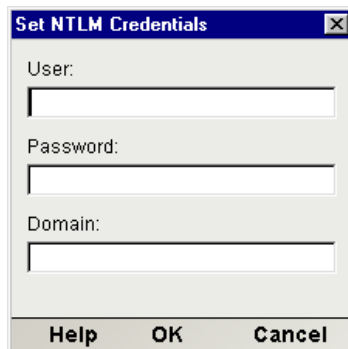
注記：これは、設計時および実行時の問題です。次の詳しい説明を参照してください。

➤ プロキシサーバ設定を設定する

- 1 Composer の [Tools] メニューで [Configuration] を選択して、[Configuration] ダイアログボックスを表示します (前を参照)。
- 2 [Use a proxy server] というチェックボックスをオンにします。ダイアログボックスの右端にある [Advanced] ボタンの他に、[Address] および [Port] のテキストフィールドも有効になります。
- 3 [Address] テキストフィールドに、プロキシサーバのIPアドレスを入力します。
- 4 [Port] フィールドに、該当するポート番号を入力します。
- 5 追加設定を表示するには、[Advanced] ボタンをクリックします。新しいダイアログボックスが表示されます。



- 6 FTPアクセス用に、HTTPアクセスとは異なるプロキシアドレスまたはポート番号、あるいはその両方を指定したい場合は、**[Use the same proxy server for all protocols]** チェックボックスが、オフになっていることを確認します。(オフにしないと、**[FTP]** テキストフィールドは無効のままです。)次に、FTPプロキシの IP アドレスおよびポート情報を、それぞれのテキストフィールドに入力します。
- 7 お使いのプロキシサーバが、サービスの利用に NTLM 認証を要求している場合は、**[Requires NTLM Authentication]** チェックボックスをオンにします。
- 8 このチェックボックスをオンにしなかった場合は、手順 13 に進みます。オンにした場合は、**[Set]** ボタンが有効になります。このボタンをクリックします。新しいダイアログボックスが表示されます。



- 9 [User] フィールドに、認証用に発行されたユーザ名を入力します。
- 10 [Password] フィールドに、パスワードを入力します。
- 11 [Domain] フィールドに、この認証手続きを適用する領域の名前を入力します。
- 12 [OK] をクリックして、ダイアログボックスを閉じます。
- 13 認証ハンドシェイク手続きから除外したいドメインがある場合は、[Proxy Settings] ダイアログボックスの [Do not use proxy server for addresses beginning with] というテキスト領域に、そのドメイン名を入力します。(よくあるのは、localhost または他の社内のテストドメイン、あるいはその両方を除外するケースです。) ドメインが複数あれば、パイプ文字で区切って入力できます。
- 14 最後に、[OK] をクリックして、ダイアログボックスを閉じます。

実行時向けのプロキシ設定

HTML コンポーネントが、いったん配備された後でも、プロキシを「通り抜け」なければならない場合は、プロキシ設定をランタイム環境にも適用する必要があります。

[Configuration] ダイアログボックスで入力するプロキシ設定は、**xconfig.xml** というファイル (Composer の **bin** ディレクトリにあります) に格納されています。このプロキシ設定が、配備されたコンポーネントでもアクティブな状態を保つには、プロキシ情報が Composer Enterprise Server の **xconfig.xml** ファイルに含まれていることを確認する必要があります。テキストエディタで設計側の **xconfig** ファイルを開き、PROXYSERVERINFO 要素を探します。次のようなデータセクションが見つかるはずです。

```
<PROXYSERVERINFO>
  <USEPROXYSERVER Desc="If on, the additional PROXY options
are enabled (valid values are on | off)">on</USEPROXYSERVER>
  <HTTPPROXYHOST Desc=" For Doc I/O, HTTP Actions etc., if
network uses a proxy enter name
here.">127.7.7.7</HTTPPROXYHOST>
  <HTTPPROXYPORT Desc="Port number HTTPPROXYHOST listens
on.">8008</HTTPPROXYPORT>
  <HTTPNONPROXYHOSTS Desc="List of hosts that do not
require a Proxy.Each hostname must be seperated by a pipe
&apos;|&apos;.">localhost</HTTPNONPROXYHOSTS>
  <FTPProxyHOST Desc=" For Doc I/O, HTTP Actions etc., if
network uses a proxy enter name here.">127.7.7.7</FTPProxyHOST>
  <FTPProxyPORT Desc="Port number FTPProxyHOST listens
on.">8008</FTPProxyPORT>
  <NTLMCREDENTIALS>
    <PROXYNTLMPROTECTED>on</PROXYNTLMPROTECTED>
    <NTLMUSER>MikeM</NTLMUSER>
    <NTLMPWD>ABYZsjbDcOk=
```

```
</NTLMPWD>
<NTLMDOMAIN>Argonaut</NTLMDOMAIN>
</NTLMCREDENTIALS>
</PROXYSERVERINFO>
```

このセクションを、お使いのサーバ側の **xconfig.xml** ファイルにコピーします。

HTML の接続リソースについて

Enterprise Connect for HTML は、他のタイプのコネクタとは異なり、サイトにアクセスするための特別なセキュリティ要件がない限り、コネクタ使用のための特別な接続リソースを作成する必要はありません。HTML コンポーネント用の特別な接続リソースを作成する必要が生じるのは、1 つ以上の HTML アクションが、HTTPS、ダイジェスト HTTP 認証、または NTLM (NT LAN Manager) を介して、セキュリティ保護された Web ページにアクセスする場合だけです。その場合、アクセスする予定のセキュリティ保護されたサイトごとに、(後で説明しているとおり) HTTP 接続リソースを 1 つ以上設定する必要があります。次に、アクションモデルで、適切な接続リソースを各 HTML アクションに割り当てます。(接続リソースを HTML アクションに関連付ける手順の詳細は、第 4 章の 43 ページで説明しています。)

HTTP 認証のタイプ

「セキュアな」HTTP 通信では、基本認証、ダイジェスト認証、および NTLM (NT LAN Manager) という 3 タイプの主要な認証方式が使用されています。Composer は、これら 3 種類の認証プロトコルに応じて、3 タイプの HTTP 接続リソースを用意しています。

基本認証

Web ブラウザなどの HTTP クライアントが、セキュリティで保護された領域の一部となっているページを要求すると、サーバは *401 Unauthorized* ステータスコードを返し、応答に [WWW-Authenticate] ヘッダフィールドを含めます。このヘッダフィールドには、その要求ページに適用可能な認証チャレンジが、少なくとも 1 つ含まれていなければなりません。これを受けてクライアントは、もう 1 度要求を出します。今回は、自分のアカウント情報を入力した [Authentication] ヘッダフィールドが含まれています。サーバは、そのアカウント情報を受け入れると、要求ページを返します。受け入れない場合は、また *401 Unauthorized* 応答を返して、認証が失敗したことをクライアントに伝えます。

この方式が抱える弱点の 1 つは、ユーザのアカウント情報が「平文で」伝送されるので、盗聴者に横取りされやすいことです。その結果、

ダイジェスト認証

ダイジェスト認証は、基本認証と同じように動作しますが、ユーザのアカウント情報が暗号化されてから HTTP で送信される点が異なっています。もちろん、公衆回線を介して送信する前にパスワードを単に暗号化しただけでは、安全性はあまり向上しません。暗号化されたパスワードでも、不正なプログラムまたは悪意のあるハッカーが盗聴し、後で再使用できるからです。これを防ぐには、所定のクライアントホストセッションに関連付けできる、一意のトランザクション ID 値を、ダイジェスト認証に対応したホストが生成する必要があります。しかも、この一意の nonce(ランダムストリング) と呼ばれるトランザクションごとの値は、クライアントに伝送されてから、認証チャレンジの一部として、ホストに返送されなければなりません。nonce の意義は、それが正確に 1 度しか使用できないことにあります。ハッカーが、盗んだユーザのアカウント情報と nonce のセットを再使用しようとしても、認証は失敗します。ホストが、その nonce はすでに 1 度使用されている、と認識するからです。nonce の生成およびその使用状況の追跡は、ホストの責任です。したがって、HTTP ダイジェスト認証の接続リソースを設定する際、この点に関しては、クライアント側は何の心配も要りません。

NTLM 認証

NTLM (NT LAN Manager) 認証は、Microsoft 独自の認証プロトコルです。まず、ハッシュ化されたキーをサーバに伝送します。サーバはそのキーを再ハッシュして新しい値を生成します。この値は、nonce(ランダムストリング) が追加されて、クライアントに戻されます。クライアントの HTML Connect Guid が、この新たに再ハッシュされたキーを再送します。クライアント側とサーバ側のハッシュアルゴリズムは、異なっています。しかも nonce が関係しているので、この方式は再試行攻撃に対して比較的セキュアです。NTLM 認証は、ローカルプロキシレベルの他に、接続先の Web サイトまたはリモートホストでもよく利用されています。NTLM 認証で保護されたプロキシを介してアクセスするつもりなら、前に節「プロキシサーバの設定」で説明したとおり、[Tools] メニューから表示される [Configuration] ダイアログで、NTLM オプションを設定する必要があります。NTLM 認証を発行する Web サイトにアクセスする場合は、後で説明しているとおり、HTTP NTLM 認証の接続リソースを設定する必要があります。

HTTP 基本認証リソース

「HTTP 基本認証」の接続タイプは、セキュリティ保護されたサイトにアクセスするために、セキュリティ証明書とパスワード情報を指定する手段として用意されています。一般に、アクセスするセキュリティ保護されたサイトごとに、HTTP 基本認証の接続リソースを 1 つ作成します。その後、適切な接続リソースを、43 ページの説明どおりに、セキュリティ保護されているサイトにアクセスする必要がある各 HTML アクションに割り当てます。

➤ セキュリティ保護されたサイト用の HTML 接続リソースを作成する

- 1 Composer の [File] メニューから、[New xObject]、[Resource]、[Connection] の順に選択します。

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。

すると、[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:
Untitled

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」のテキストを入力します。
- 4 [Next] をクリックします。
- 5 [Connection Type] プルダウンメニューから、[HTTP Basic Authentication Connection] を選択します。

- 6 「ユーザ ID」および「パスワード」を入力します。これらは、接続の確立中に実際には送信されず、単にここで定義されているにすぎません (パスワードの暗号化は行われます)。ユーザは、ECMAScript 式からユーザ ID 変数およびパスワード変数にアクセスして、画面内の値として、ユーザ ID とパスワードをマップできるようにします。このため、パスワードが誰かに表示されることは決してありません。
- 7 接続先のサイトが、クライアント側の証明書を要求している場合は、次の手順を実行します。
 - [Browse] ボタンをクリックし、このサービス接続で使用する証明書ファイルを選択して、「クライアント証明書」を選択します。
 - [Browse] ボタンをクリックし、セキュリティ用のクライアントキーファイルを選択して、「クライアントプライベートキー」を選択します。
 - ここまでの手順に関して、デジタル証明書の詳細については、このマニュアルの「付録 A」を参照してください。
 - 「プライベートキー」の「パスワード」を入力します。プライベートキーにより、クライアントプライベートキーの所有者のセキュリティレベルが、さらに向上します。
- 8 「接続タイムアウト」の値を秒単位で入力します。この値は、コンポーネントが、Web ページがダウンロードされ始めるのを待つ時間の最大値です。この指定時間内に、接続が確立されなかったりページがダウンロードされなかった場合は、例外が発生します。
- 9 この時点以降、[HTML Action setup] ダイアログボックス (43 ページ参照) のプルダウンリストに、この接続リソースが最初に表示されるようにしたい場合は、[Default] チェックボックスをオンにします。
- 10 [Finish] をクリックします。これで、接続リソースが作成されました。

HTTP ダイジェスト認証の接続リソース

「HTTP ダイジェスト認証」の接続タイプは、ダイジェスト認証（前を参照）を使用している、セキュリティ保護されたサイトにアクセスするために、ユーザがユーザ名、パスワード、および（オプションとして）セキュリティ証明書を指定する手段として用意されています。一般に、アクセスするセキュリティ保護されたサイトごとに、HTTP ダイジェスト認証の接続リソースを1つ作成します。コンポーネントの作成時に、適切な接続リソースを、36 ページの説明どおりに、セキュリティ保護されているサイトにアクセスする必要がある各HTMLアクションに割り当てます。

➤ HTTP ダイジェスト認証の接続リソースを作成する

前に説明した、「HTTP 基本認証の接続リソース」での手順を参照してください。表示されるダイアログボックスは、「基本」および「ダイジェスト」という語句以外は、すべて同じです。

HTTP NTLM 認証の接続リソース

「HTTP NTLM 認証」の接続タイプは、NTLM 認証を使用している、セキュリティ保護されたサイトにアクセスするために必要な、アカウント情報を指定する手段として用意されています。

一般に、アクセスするセキュリティ保護されたサイトごとに、HTTP NTLM 認証の接続リソースを1つ作成します。コンポーネントの作成時に、適切な接続リソースを、36 ページの説明どおりに、セキュリティ保護されているサイトにアクセスする必要がある各 HTML アクションに割り当てます。

➤ HTTP NTLM 認証の接続リソースを作成する

- 1 Composer の [File] メニューから、[New xObject]、[Resource]、[Connection] の順に選択します。

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。

すると、[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:

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」のテキストを入力します。
- 4 [Next] をクリックします。
- 5 [Connection Type] プルダウンメニューから、[HTTP NTLM Authentication Connection] を選択します。

Create a New Connection Resource

For HTTP Servers requiring NTLM authentication, supply a User ID, Password and Domain. If the connection requires Client Certificates and/or a Client Private Key use the browse to select the resource. Use the right mouse button to create a conditional expression for a connection parameter. Checking 'Default' makes this Connection Resource the initial selection when requesting an HTTP connection

Connection Type:

User ID:

Password:

Domain:

Client Certificate:

Client Private Key:

Private Key Password:

Connection Timeout (sec):

Default

Help **Back** **Finish** **Cancel**

- 6 「ユーザ ID」および「パスワード」を入力します。
- 7 サイトの「ドメイン」名を入力します（“http://www.domain.com”のように）。
- 8 接続先のサイトが、クライアント側の証明書を要求している場合は、次の手順を実行します。

- **[Browse]** ボタンをクリックし、このサービス接続で使用する証明書ファイルを選択して、「**クライアント証明書**」を選択します。

- **[Browse]** ボタンをクリックし、セキュリティ用のクライアントキーファイルを選択して、「**クライアントプライベートキー**」を選択します。

注記: ここまでの手順に関して、デジタル証明書の詳細については、このマニュアルの「付録 A」を参照してください。

- 「**プライベートキー**」の「**パスワード**」を入力します。プライベートキーにより、クライアントプライベートキーの所有者のセキュリティレベルが、さらに向上します。

- 9 「**接続タイムアウト**」の値を秒単位で入力します。この値は、コンポーネントが、Web ページがダウンロードされ始めるのを待つ時間の最大値です。この指定時間内に、接続が確立されなかったりページがダウンロードされなかった場合は、例外が発生します。
- 10 この時点以降、**[HTML Action setup]** ダイアログボックス (36 ページ) のプルダウンリストに、この接続リソースが最初に表示されるようにしたい場合は、**[Default]** チェックボックスをオンにします。
- 11 **[Finish]** をクリックします。これで、接続リソースが作成されました。

コンポーネント用 XML テンプレートの作成

接続リソースの他に、HTML 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、あらかじめ XML テンプレートを作成しておかなければならないこともあります。詳細については、『*exteNd 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。

3

HTML コンポーネントの作成

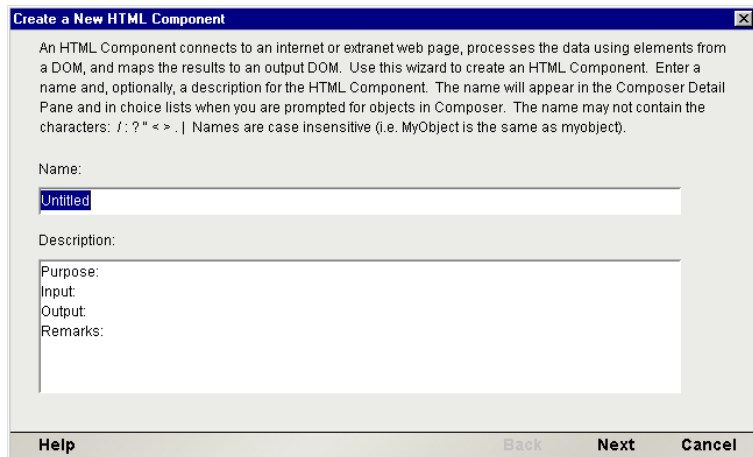
HTML コンポーネントを作成する前に

すべての exteNd コンポーネントと同様、HTML 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。

▶ 新しい HTML コンポーネントを作成する

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



- 2 [Name] に、新しい HTML コンポーネントの名前を入力します。

- 3 オプションで、**[Description]** に説明のテキストを入力します。
- 4 **[Next]** をクリックします。Create a New HTML 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。

Specify one or more XML Templates to help design Input to this Component or Web Service and only one to design Output. The sample XML Documents in each Template are design time aids to help you build Action Models for the component. The samples are not actually used at runtime after deployment to your application server. The Identifier is fixed and represents the name used to refer to the XML Document during component execution. Selecting System {ANY} allows you to use an empty template (i.e. accept any document as Input).

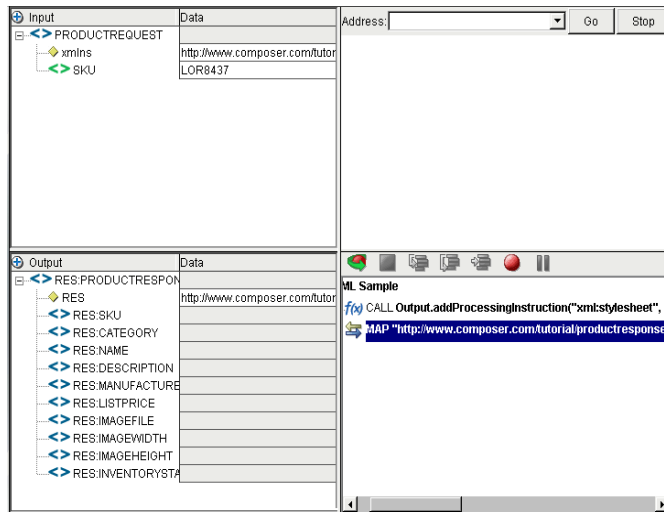
Part	Template Category	Template Name	
Input	Office Supply	ProductRequest	Add
			Delete

Part	Template Name
Output	ProductResponse

Help Back Finish Cancel

- 5 入力テンプレートおよび出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、**[Template Category]** を選択します。この例では、入力および出力に対して **[Office Supply]** を選択します。
 - ◆ 選択した **[Template Category]** に対するテンプレート名を **[Template Name]** の XML テンプレートのリストから選択します。この例では、入力テンプレートに対して **[Product Request]** を選択します。他の入力 XML テンプレートを追加するには、**[Add]** をクリックして、**[Template Category]** および **[Template Name]** をそれぞれ選択します。
 - ◆ 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 出力としての XML テンプレートを選択します。この例では、出力テンプレートは **[Product Request]** です (出力 DOM の名前は「Output」です)。

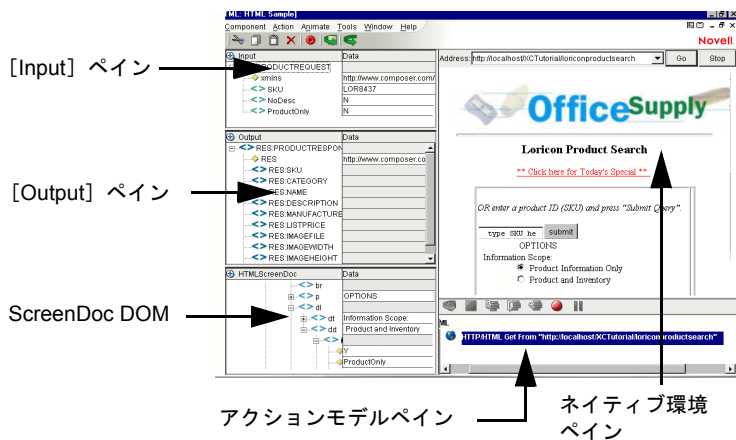
注記： 出力テンプレートとして {System}{ANY} を選択すると、構造が含まれない入力 XML テンプレートまたは出力 XML テンプレートを指定できます。詳細については、『Composer ユーザガイド』の「テンプレートを使用しない出力 DOM の作成」を参照してください。
- 7 **[Finish]** をクリックします。コンポーネントが作成され、HTML コンポーネントエディタが表示されます (次を参照)。



HTML コンポーネントエディタウィンドウについて

HTML 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。HTML Connect では、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、アクションペインも含まれています。

ただし、従来の機能との大きな違いが 2 つあります。1 つ目は、HTML コンポーネントエディタにはすべての Connect に共通するネイティブ環境ペインも含まれているという点です。エディタにはアドレスラインに加え、[Go] および [Stop] という 2 つのボタンが上部に備わった Web ブラウザウィンドウが含まれます。2 つ目は、ScreenDoc という XML DOM がコンポーネントエディタウィンドウに追加されているという点です。この DOM では Web サイトから受信した各画面の XML ドキュメントが表示され、コンポーネント内でマップアクションを参照および作成することができます。さらに、ScreenDoc DOM は Expression Builder で使用でき、ユーザは画面フィールドを簡単に参照できます。



HTML ネイティブ環境ペインについて

HTML ネイティブ環境ペインから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) からデータをマップして、HTML 画面フィールドまたは HTML フォームの入力として使用する。たとえば、SKU 番号を入力 DOM から HTML 問い合わせフォームのパーツフィールドにドラッグして、そのパーツ番号に関連付けられているデータ (説明や価格など) を返すことができます。
- ◆ HTML 画面からデータをマップして、出力 XML ドキュメント (または、Temp、MyDom などの他の使用可能な DOM) に配置する。
- ◆ SSL3 およびデジタル証明書を使用してセキュリティ保護された Web サイトを操作する B2B サービスを作成する。

JavaScript を実行できない点を除き、HTML ネイティブ環境ペイン機能は、Web ブラウザとまったく同じように動作し、同じタイプのタスクを実行できます。

ScreenDoc DOM について

ScreenDoc DOM は、ネイティブ環境ペインのブラウザで表示されている現在の Web ページの XML ドキュメント表記 (実際には、Tidy による Web ページ処理後の XHTML) です。この画面に対するすべてのマップアクション (ドラッグアンドドロップなど) は、実際には ScreenDoc DOM の要素を参照します。任意のアプリケーション画面を XML ドキュメントとして扱うと同時に、表示および参照できます。

動作

コンポーネントで Web ページを受信するたびに、同時に次のような複数の状況が発生します。

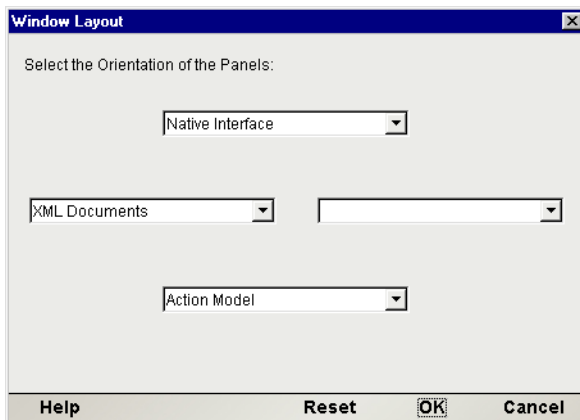
- ◆ ネイティブ環境ペインに元の HTML Web ページが表示される。
- ◆ 元の HTML が Tidy によって処理され、Web ページの XHTML バージョンが作成された後、ScreenDoc DOM に表示される。他の XML ドキュメントを使用する場合と同様に、任意のアクションを ScreenDoc DOM で使用できます。
- ◆ ホストに送信されたパラメータに加え、受信した Web ページの URI を含む HTML アクションが作成される (記録モード時のみ)。

Web ページを XHTML として表示すると、ScreenDoc DOM が非常に大きくなります。これは、主に非表示フィールドの検出、フィールドやパラメータフィールド値の確認、および便利な場合には Composer のドラッグアンドドロップ機能を使用した ScreenDoc DOM から出力 DOM へのマップに使用されます。

注記： 通常、ScreenDoc DOM にマップするよりも、ドラッグアンドドロップを使用してネイティブ環境ペインとの間で直接マップした方が、より迅速で効率的です。ScreenDoc DOM の XHTML には、フォント、テーブル属性などに関連する属性を含め、Web ページのあらゆる詳細な情報が表示されます。不正なデータがマップされないように、可能な場合はドラッグアンドドロップを使用してネイティブ環境ペインとの間で直接ドラッグしてください。

➤ ScreenDoc DOM を表示する

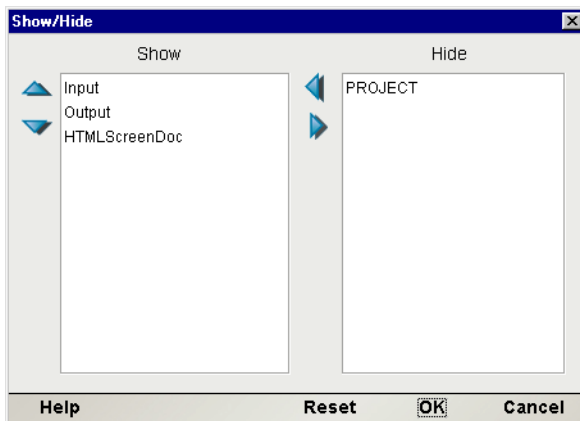
- 1 コンポーネントウィンドウのメニューから、[View]、[Window Layout] の順に選択します。
- 2 [Window Layout] ダイアログボックスが表示され、ウィンドウ内でパネルの配置を調節できます。4 つの異なるフィールドのドロップダウンリストの矢印を使用して、ペインの配置を選択します。



- 3 [OK] をクリックして、ダイアログボックスを閉じます。設定を変更する場合は、[Reset] をクリックします。

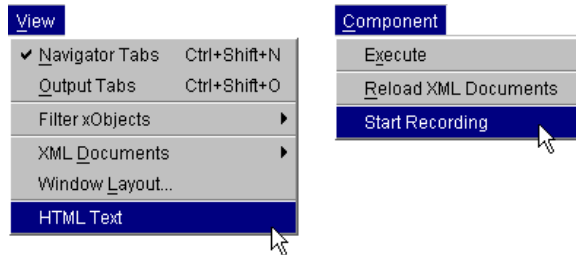
➤ **コンポーネントエディタでXMLドキュメントのビューを整理する**

- 1 [View]、[XML Documents]、[Show/Hide] の順に選択します。
- 2 矢印ボタンを使用すると、[Invisible] 列から [Visible] 列に (またはその逆にも) ペインを移動できます。また、表示の選択項目が画面に表示される順序を選択することもできます。
- 3 [OK] をクリックして、設定を保存します。設定を変更する場合は、[Reset] をクリックします。



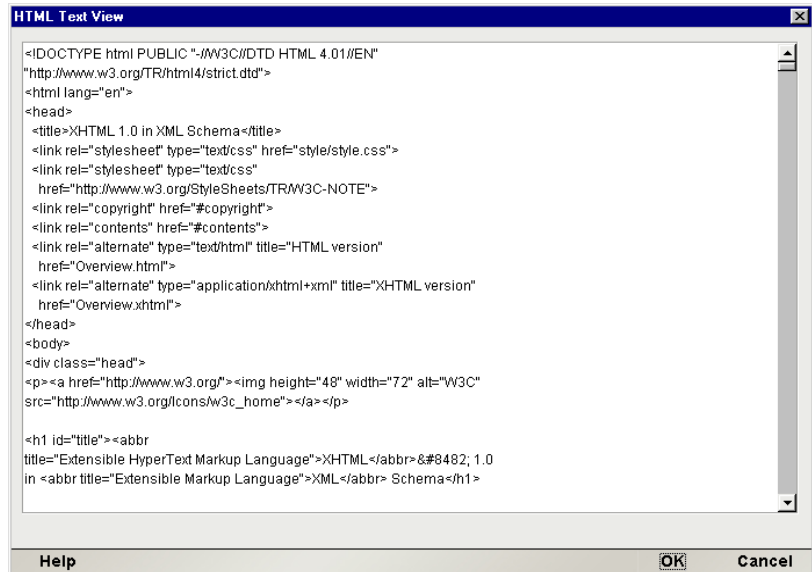
HTML 固有のメニューバー項目について

HTML Connect の編集環境を使用する場合、Composer のメインメニューに HTML Connect 固有の特定のコマンドが表示されます。



[View] メニュー

[HTML Text] 翼 I ティブ環境ペインに現在表示されているページの HTML コード (ソースコード) を表示するウィンドウを開きます。これは、Netscape または Internet Explorer における [表示]、[ソース] の操作と類似しています。



[Component] メニュー

[Start Recording] — ネイティブ環境ペインでリアルタイムに操作しながら Composer で動的にアクションを作成します。

記録を停止するには、[Record] ボタンをクリックします (次の節を参照)。ボタンにより動作を切り替えられます。

HTML 固有のボタンについて

[レコード]

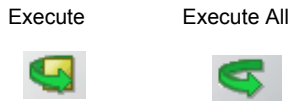
HTML Connect では、コンポーネントエディタのツールバーに [Record] という追加のツールボタンがあります。[Record] ボタンを使用すると、HTML 画面を操作する際に新しいアクションがアクションモデルに自動的に作成されます。



レコード

[Execute] および [Execute All]

[Record] ボタンの横に、[Execute] および [Execute All] ボタンがあります。これらのボタンを使用すると、コンポーネントで選択したアクションまたはアクションモデル全体をそれぞれ実行できます。



Execute および Execute

4

HTML 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。

HTML 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、Web サイトの画面と 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。

HTML Connect には、HTML アクションおよび Set Frame アクションという基本的な 2 つのアクションがあります。HTML アクションは URI にアクセスし、URI アクションのパラメータをエンコードします。HTML アクションには、HTTP Get および HTTP Post が含まれます。これらのアクションタイプは、データを要求するメソッドと呼ばれます。HTTP Get とは、指定の form-data 形式で構成されるクエリを使用してファイルを取得するメソッドです。データは URL エンコードされ、文字列に変換され、その後クエリ文字列として送信されます。

Web ページには複数のフレームを設定することができます。また、各フレームには独自のドキュメントを設定できます。フレームのコンテキストを設定するために自動生成される Set Frame アクションを使用すると、選択したフレームのコンテキストを設定できるため、DOM にこのフレームのコンテンツが反映されます。

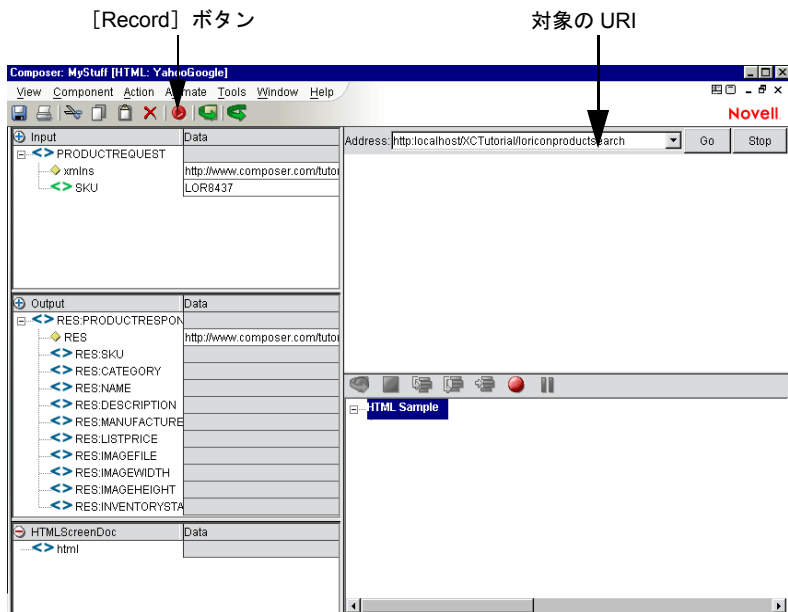
HTTP Post は、特定のヘッダを使用して指定の URI にデータをポストする別のメソッドタイプです。HTTP Post では、処理するための大量の情報がサーバに送信されます。ただし、この処理は Web ページの埋め込み形式に依存します。データは最初に URL エンコードされ、次に使用されている形式の文字列に変換されます。アクションに対する応答としては、「リダイレクト」が考えられます。リダイレクトは自動的に処理されるため、ユーザからの操作は必要ありません。

HTML セッションの記録

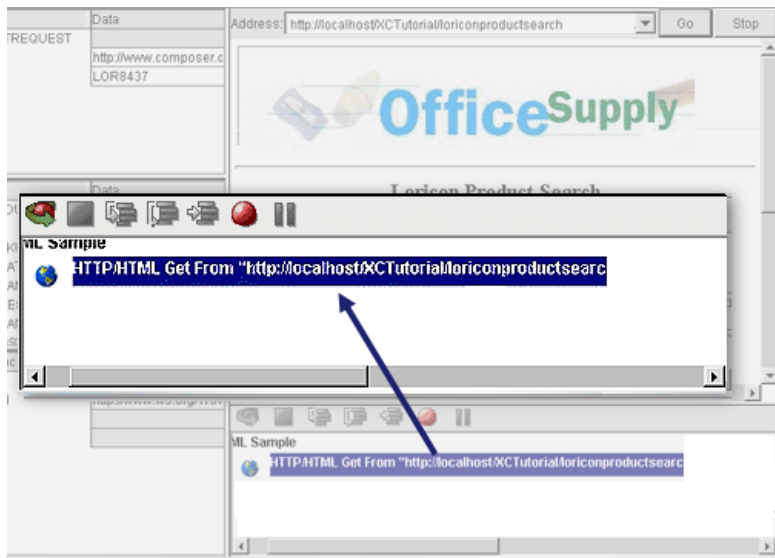
HTML コンポーネントは、大部分のアクションモデルが自動的に作成されるため、他のコンポーネントとは異なります。これは、ネイティブ環境ペインで Web サイトからのライブセッションと通信し、Composer の「記録」機能を使用してアクティビティをアクションのセットとしてアクションモデルに記録するためです。他のコンポーネントでは、ユーザは、マップ、変換、およびタスクの転送を実行するためのアクションを、アクションモデルで手動で作成する必要があります。HTML コンポーネントを作成する場合、アクションモデルペインの HTML アクション、Map アクション、および Function アクションの組み合わせとしてキャプチャされる、Web サイトからまたは Web サイトへの要求および応答を基本的に記録します。

➤ シンプルな HTML セッションを記録する

- 1 このガイドの第3章の「HTML コンポーネントの作成方法」の手順に従って、HTML コンポーネントを作成します。この例で示す HTML コンポーネントの作成では、Office Supply Product Request テンプレートおよび Product Response テンプレートが入力および出力に対してそれぞれ選択されています。新しく作成した HTML コンポーネントは、HTML コンポーネントエディタウィンドウに表示されます。
- 2 [URI] フィールドにアドレスを入力します。例で使用されているアドレスは、**<http://localhost/XCTutorial/loriconproductsearch>** です。[GO] ボタンをクリックするか、キーボードで **Enter** キーを押します。

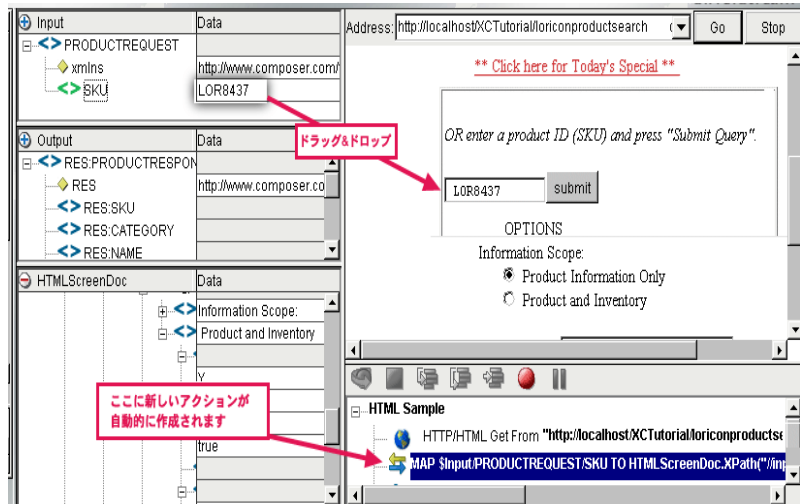


- 3 [Record] ボタンをクリックすると、現在の Web ページがアクションモデルで HTML アクションとしてキャプチャされます (次を参照)。

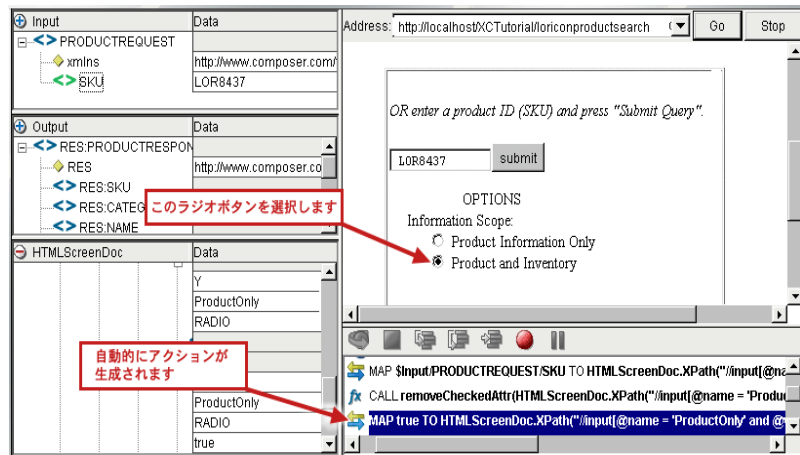


HTML アクションは自動的にアクションモデルに追加されます。

- 4 SKU 情報「LOR8437」を入力 DOM からネイティブ環境ペインに表示されているフォームのエントリーフィールドにドラッグし、フィールド内にドロップします。Map アクションはアクションペインで作成されます。



- 5 ネイティブ環境ペインで SKU を入力した後、デフォルトのオプションの選択を解除して、[Product and Inventory] ラジオボタンをオンにします。



注記： ネイティブ環境ペインでフォームを操作するとき新しく選択操作を行うと、アクションペインに Map アクションが自動的に作成されます。フォーム内で以前の選択を選択解除すると、アクションペインに Function アクションが作成されます。たとえばこの例では、デフォルトのラジオボタンを選択解除して、[Product and Inventory] ラジオボタンを選択し、オンにしています。この場合、アクションペインには両方のタイプのアクション、Function アクションおよび Map アクションが作成されます。HTML ビジュアルコントロールの結果として作成されるアクションのリストについては、「付録 D」を参照してください。

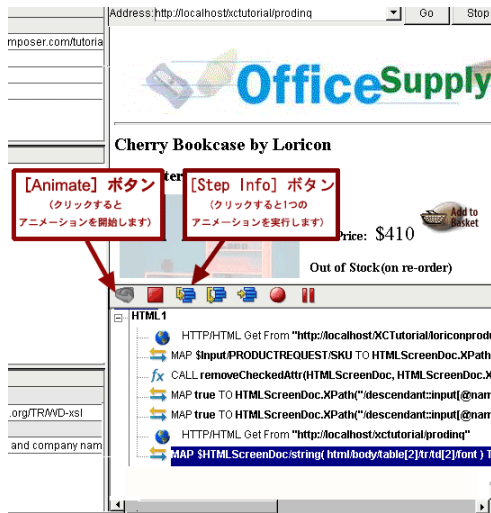
- 6** [Omit Product Description] のチェックボックスをオンにします。

注記： ネイティブ環境ペインのフォームコントロールを操作するに従って、アクションモデルに新しいアクションが作成されます。この例では、チェックボックスをオンにして、[Omit Product Description] を選択しています。

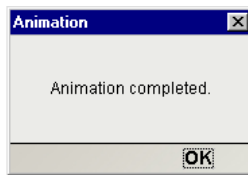
- 7** [Submit] ボタンをクリックします。要求された検索情報が表示された Web ページが開きます。
- 8** 検索結果画面から出力 DOM に、要素をドラッグアンドドロップします。たとえば、「\$410」を出力 DOM の [LISTPRICE] フィールドにドラッグします。ドラッグアンドドロップするデータは赤で表示されます。
- 9** 必要に応じて、データ要素を検索結果画面から出力 DOM の該当するフィールドにドラッグアンドドロップする操作を繰り返します。要素を検索結果画面から出力 DOM にドラッグするたびに、アクションモデルペインにアクションが記録されます。

注記： 各アクションが行われるたびに、ネイティブ環境で Web ページをミラーリングする HTML ScreenDoc DOM は更新されます。[HTMLScreenDoc] ツリーを展開すると、詳細を確認できます。

- 10** [Record] ボタンをクリックして、記録モードをオフにします。
- 11** [Animate] アイコンをクリックしてアニメーションを開始し、[Step Into] アイコンを繰り返しクリックすることによってアクションペインでアクションを実行します。(次の図を参照してください)。



- 12 この処理が完了すると、小さなダイアログボックスが表示されます。[OK] をクリックします。



次に表示される画面では、出力 DOM には Web ページから要求された「\$410」のみが含まれています。

Input	Data
PRODUCTREQUEST	
xmIns	http://www.composer.com/tutorial
SKU	LOR8437
NoDesc	N
ProductOnly	N

Output	Data
RESPONSE	
LISTPRICE	\$410

HTMLScreenDoc	Data
html	
xmIns:xsl	http://www.w3.org/TR/RWD-xsl
head	
body	Other product and company nam

Address: Go Stop

OfficeSupply

Cherry Bookcase by Loricon

You entered SKU: 8437-lor

Our Price: \$410

Out of Stock (on re-order)

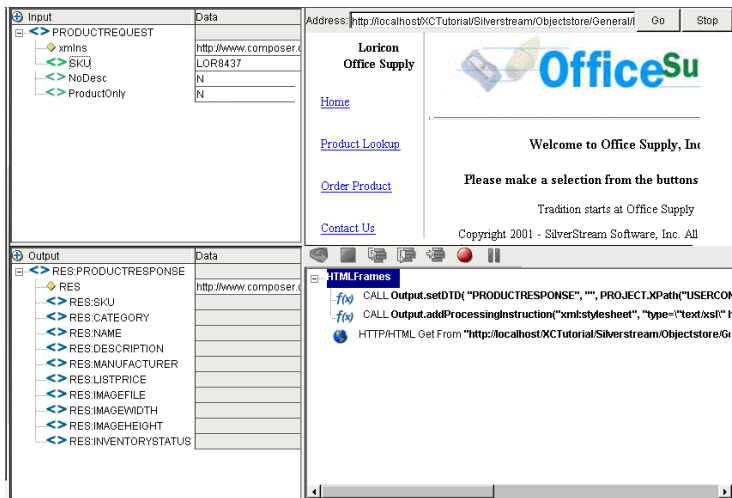
HTML 1
HTTP/HTML Get From "http://localhost/xctutorial/oriconproducts"
MAP \$input.PRODUCTREQUEST/SKU TO HTMLScreenDoc.XPath("/dk
CALL removeCheckedAttr(HTMLScreenDoc, HTMLScreenDoc.XPat
MAP true TO HTMLScreenDoc.XPath("descendant::form[1]/input[6
CALL removeCheckedAttr(HTMLScreenDoc, HTMLScreenDoc.XPat
MAP true TO HTMLScreenDoc.XPath("descendant::form[1]/input[6
MAP true TO HTMLScreenDoc.XPath("descendant::form[1]/input[6
HTTP/HTML Get From "http://localhost/xctutorial/prodinq"
MAP \$HTMLScreenDoc.html/body/table[2]/tr[td[2]]font TO \$Output.PI

フレームを使用した HTML セッションの記録

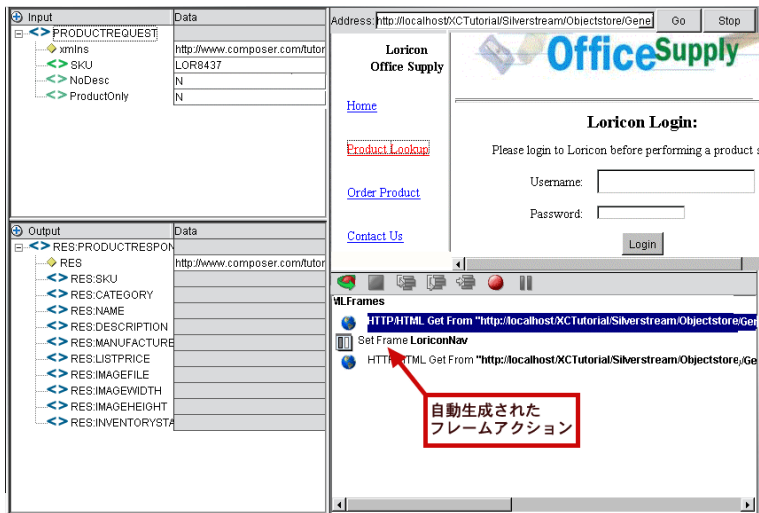
使用している Web ページに複数のフレームが含まれている場合、フレーム間をクリックしたり、データを DOM にドラッグアンドドロップしたりすると HTML Connect によって Map Screen アクションに Set Frame アクションが作成および記録されます。HTML Screen Doc には、このフレームの DOM が表示されます。

▶ フレームを含むページを使用した HTML セッションを記録する

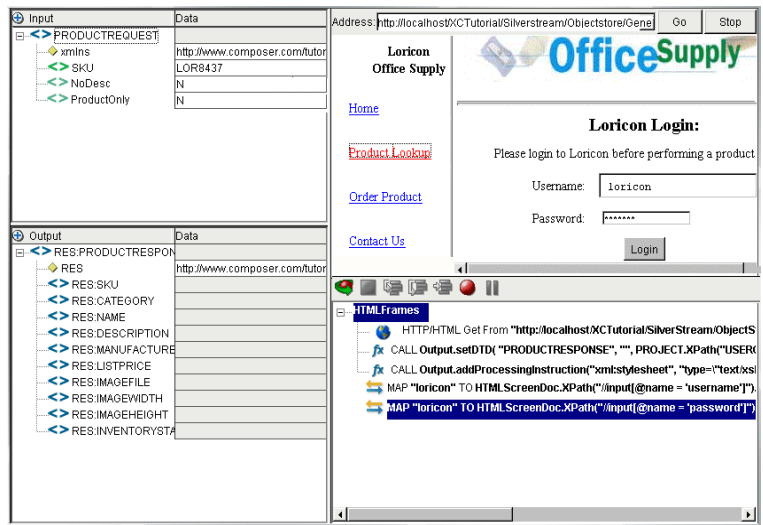
- このガイドの第 3 章の「HTML コンポーネントの作成方法」の手順に従って、HTML コンポーネントを作成します。この例では HTML コンポーネントを作成する場合、入力と出力に対してそれぞれシンプルなテンプレートが使用されています。新しく作成した HTML コンポーネントは、HTML コンポーネントエディタウィンドウに表示されます。
- [URI] フィールドにアドレスを入力します。この例で使用されているアドレスは、<http://localhost/XCTutorial/Silverstream/Objectstore/General/LoriconHome.html> です。
- [GO] ボタンをクリックするか、キーボードで **Enter** キーを押します。
- [Record] ボタンをクリックすると、現在の Web ページがアクションモデルで HTML アクションとしてキャプチャされます。



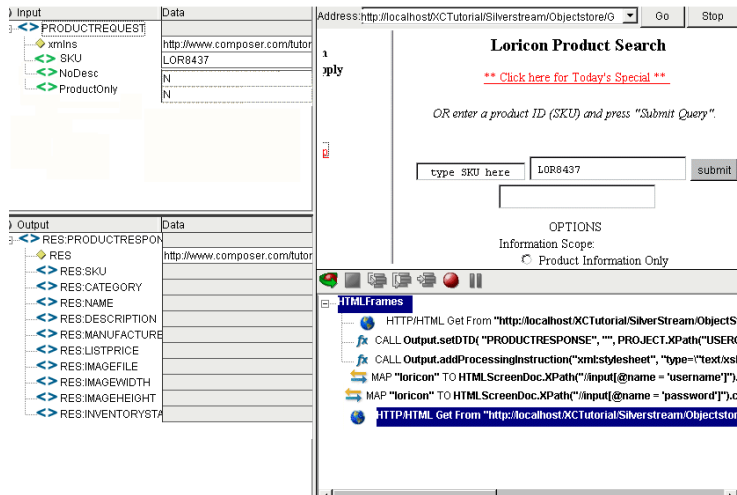
この例では、[Product Lookup] をクリックします。Map アクションペインでは、Set Frame アクションが作成され、異なるフレームに入っていることが示されています。次に、その後続くアクションが記録されます。



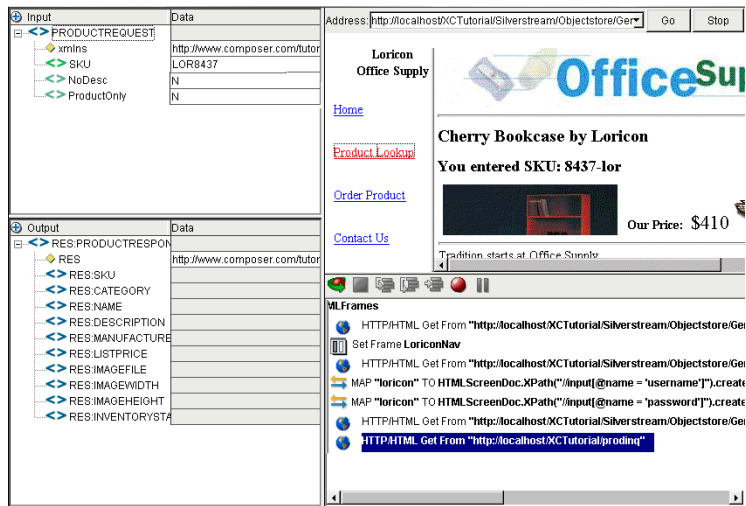
5 選択した後で、ユーザ名とパスワードをを求めるログイン画面が表示されます。



- 6 ユーザ名とパスワードを入力します。この例では、ユーザ名もパスワードも共に「loricon」です。[Login] ボタンをクリックします。マップアクションペインにアクションが作成されていることが分かります。
- 7 SKUを入力し、[Submit] ボタンをクリックします。この例では、SKUには LOR8437 を使用します。



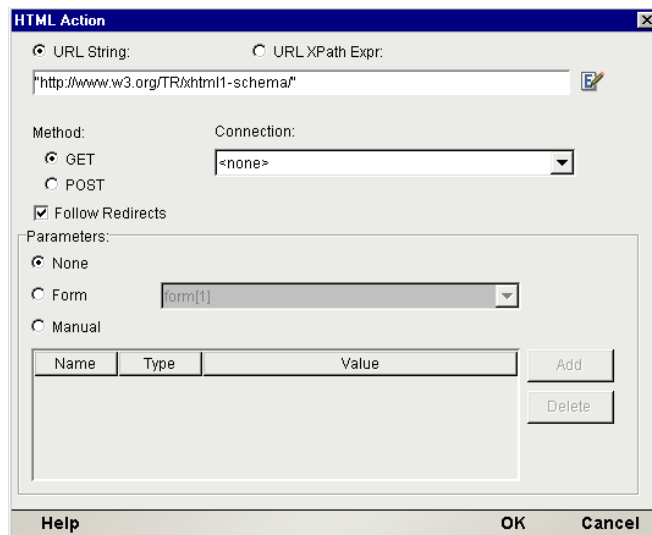
- 8 [Submit] ボタンをクリックすると、要求に対する応答がナビゲーションペインに表示されます (次を参照)。



HTML アクションの編集

➤ HTML アクションでリンクおよびパラメータを編集する

- 1 アクションモデルで既存の HTML アクションをダブルクリックします。
[HTML Action] ダイアログボックスが表示されます (次を参照)。



- 2 移動先の Web サイト画面のアドレスを [URI] フィールドに入力します。文字列を入力するか、[Expression] アイコンをクリックして目的の URI を指定する ECMAScript 式を作成します。

注記： URI を直接入力する場合 (前の図を参照)、URI を二重引用符で囲んでください。

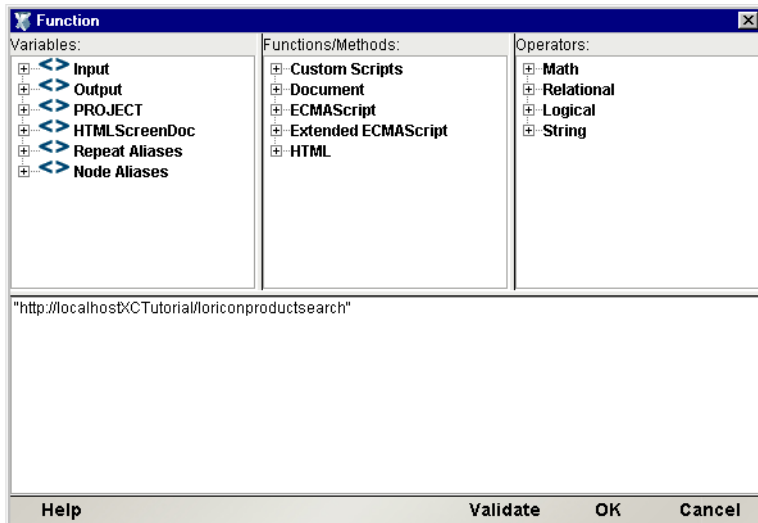
- 3 [GET] ラジオボタンまたは [POST] ラジオボタンをオンにして、メソッドタイプを選択します。通常は、[GET] を選択します。
- 4 デフォルトにより HTML アクションでランタイム時にリダイレクトを適用する場合は、[Redirect] チェックボックスをオンにします。

注記： ここでは、リダイレクトは別の URI への Java スクリプト以外のリダイレクトを指します。HTML ページに埋め込まれた Java スクリプトは、(このバージョンの Connect では) `exteNd` によっては認識されないため、Java スクリプトのリダイレクトは適用されません。

- 5 問題の Web ページが保護されているページである場合 (つまり、URI が「https:」で始まる場合)、プルダウンリストから [Connection] を選択します。選択できる接続リソースが見つからない場合、第 2 章で HTTP 基本認証接続リソースを作成する方法を参照してください。新しい接続リソースを作成すると、この接続リソースはプルダウンリストに表示されます。

- 6 [Parameters] セクションには、[None]、[Form]、および [Manual] という 3 つのラジオボタンが含まれています。いずれかのラジオボタンを選択します。

- ◆ [None] を選択すると、パラメータは選択されません。
- ◆ [Form] を選択すると、名前別に表示されたフォームをプルダウンリストから選択できます。リストには、HTML ページに埋め込まれている各フォームが表示されます。フォームを選択すると、そのフォームに含まれているフィールドがダイアログボックス下部のテーブルに表示されます (名前、タイプ、および値別に表示)。
- ◆ [Manual] を選択すると、任意のフォームの新しいフォームフィールドに名前、タイプ、および値を追加できます。また、選択したフォームから任意のフィールドを削除することもできます (これらの変更によって ScreenDoc DOM は適切に更新されます)。[ADD] ボタンを選択した後で、新しいフィールドの名前、タイプ、および値の情報を入力します。[DELETE] ボタンを使用すると、必要に応じてフィールドをリストから削除できます。[Value] 領域をクリックすると、[Expression] アイコンが表示されます。アイコンをクリックすると、[Expression Builder] 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では、選択リスト項目を使用して簡単に式を作成できます。



注記： ECMAScript 文字列は、二重引用符で囲む必要があります。

- 7 終了すると [OK] をクリックして、[HTML Action] ダイアログに戻ります。
- 8 [OK] をクリックして、[HTML Action] ダイアログを閉じます。
- 9 コンポーネントを保存します。

以前に記録したアクションモデルの編集

以前に記録したアクションモデルの編集が必要となる場合があります。他のコンポーネントの編集とは異なり、HTML コンポーネントの編集には、さらに注意が必要です。HTML コンポーネントを実行すると、コンポーネントが適切に動作するために、特定の Web ページおよびデータが表示されるようなアクションのシーケンスが繰り返されます。そのため、コンポーネントを編集する際には、アクションモデルのシーケンスが以前に記録した Web ページの実行シーケンスと矛盾しないように注意する必要があります。

一般的に、正常に編集を行うには、次の推奨事項が適用されます。

- ◆ HTML アクションを切り取ってコピーしたり、アクションモデル内の別の場所に貼り付けたりしないでください。
- ◆ HTMLScreenDoc DOM と通信する個々の Map アクションをコピーしたり貼り付けたりした後は、各アクションのチェックと編集を慎重に行ってください。
- ◆ Composer のドラッグアンドドロップ機能を使用して、画面と動作する新しい Map アクションを追加します。

アクションを編集または追加する場合の最も安全な手順は、アクションモデルで所定の行までアニメーション表示した後、一時停止し、記録モードに戻ります。これにより、アクションモデルが適切な ScreenDoc DOM または特定の ScreenDoc DOM 内のフィールド、あるいはその両方と同期が取れなくなることを防ぐことができます。

既存のアクションの変更

次の手順では、以前に記録したセッションで既存のアクションを変更する方法を説明します。

▶ 以前に記録されたアクションモデルで、既存のアクションを変更する

- 1 編集する対象となる、以前に記録されたアクションモデルを含むコンポーネントを開きます。HTML コンポーネントエディタウィンドウにコンポーネントが表示されます。

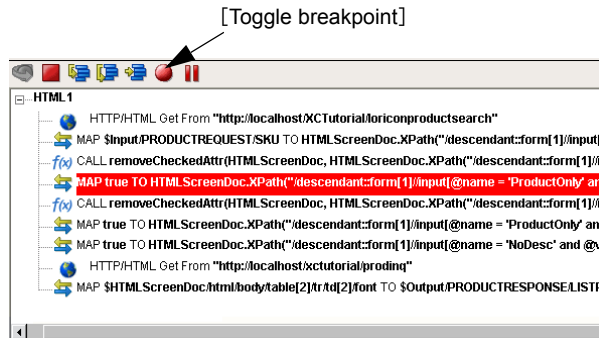
The screenshot displays the Action Model Editor interface. On the left, there are three panels: 'Input', 'Output', and 'HTMLScreenDoc'. The 'Input' panel shows a 'PRODUCTREQUEST' action with fields for 'xmlns', 'SKU' (LOR8437), 'NoDesc', and 'ProductOnly'. The 'Output' panel shows a 'RES:PRODUCTRESPONSE' action with a 'LISTPRICE' field containing the value '\$410'. The 'HTMLScreenDoc' panel shows the DOM structure of the page, including 'html', 'xmlns:xsl', 'head', and 'body'. On the right, a browser window shows the 'OfficeSupply' website with a 'Cherry Bookcase by Loricon' product. The product page displays 'You entered SKU: 8437-lor', 'Our Price: \$410', and 'Out of Stock (on re-order)'. Below the browser window, the 'HTML 1' panel shows the recorded actions, including 'HTTP/HTML Get From', 'MAP \$input.PRODUCTREQUEST/SKU TO HTMLScreenDoc.XPath()', 'CALL removeCheckedAttr()', 'MAP true TO HTMLScreenDoc.XPath()', and 'HTTP/HTML Get From'.

- 2 アクションモデルで、編集を行うアクションに移動し、選択します。

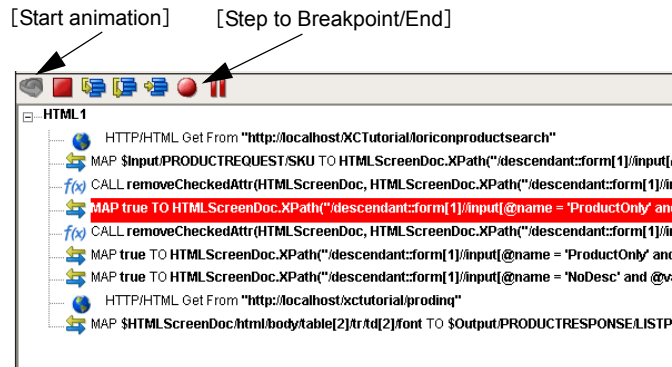
The screenshot shows the 'HTML 1' panel in the Action Model Editor. It contains a list of recorded actions:

- HTTP/HTML Get From "http://localhost/XCTutorial/Loricon/productsearch"
- MAP \$input.PRODUCTREQUEST/SKU TO HTMLScreenDoc.XPath()/"descendant::form[1]/input[@name='SKU']"
- CALL removeCheckedAttr(HTMLScreenDoc, HTMLScreenDoc.XPath()/"descendant::form[1]/input[@name='SKU']")
- MAP true TO HTMLScreenDoc.XPath()/"descendant::form[1]/input[@name='SKU']"** (highlighted)
- MAP true TO HTMLScreenDoc.XPath()/"descendant::form[1]/input[@name='SKU']"
- HTTP/HTML Get From "http://localhost/xctutorial/prodinq"
- MAP \$HTMLScreenDoc.html/body/table[2]/tr/td[2]/font TO \$Output.PRODUCT

- 3 [Toggle Breakpoint] ボタンをクリックします (または、<F2> を押します)。選択したアクションが赤色になります。



- 4 [Start Animation] ボタンをクリックします。アニメーションツールが有効になります。



- 5 [Run to Breakpoint/End] ボタンをクリックします。アクションモデルは、最初から手順 3 で設定したブレーポイントまでのすべてのアクションを実行します (次の図を参照)。
- 6 アニメーションツールバーで、[Pause] ボタンをクリックします。
- 7 コンポーネントエディタのツールバーで、[Record] ボタンをクリックします。
- 8 アクションモデルに追加するアクションを実行します。
- 9 [Record] ボタンをクリックして、記録モードをオフにします。
- 10 [File]、[Save] の順に選択するか、コンポーネントエディタのツールバーで [Save] ボタンをクリックします。
- 11 「アニメーションツールの使用」の指示に従って、コンポーネントをテストします。

新しいアクションの追加

次の手順では、以前に記録したセッションで新しいアクションを追加する方法を説明します。

▶ 新しいアクションを以前に記録したアクションモデルに追加する

- 1 アクションを追加する対象となる、以前に記録したアクションモデルを含むコンポーネントを開きます。HTML コンポーネントエディタウィンドウにコンポーネントが表示されます。

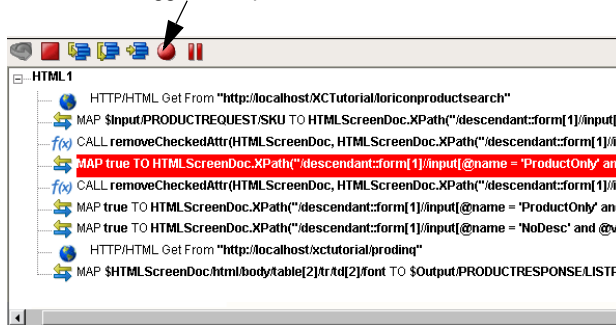
The screenshot displays the Selenium IDE interface. On the left, the 'HTMLScreenDoc' component is selected, showing its structure: 'html' (containing 'xmlns:xsl', 'head', and 'body') and 'Data' (containing 'http://www.w3.org/TR/RWD-xsl' and 'Other product and company nam'). The 'Output' pane shows the 'RES:PRODUCTRESPONSE' component with a 'LISTPRICE' action that has a value of '\$410'. A callout box highlights the 'LISTPRICE' action and its value. The 'Data' pane shows the 'PRODUCTREQUEST' component with values: 'xmlns' (http://www.composer.com/tutorial), 'SKU' (LOR8437), 'NoDesc' (N), and 'ProductOnly' (N). On the right, a browser window shows the 'OfficeSupply' website with a 'Cherry Bookcase by Loricon' product. The price is '\$410' and it is 'Out of Stock (on re-order)'. The browser's address bar shows 'http://localhost:xtutorial/proding'.

- 2 アクションモデルで、追加を行うアクションに移動し、選択します。

The screenshot displays the Selenium IDE interface. The 'HTML 1' component is selected, showing its structure: 'HTTP/HTML Get From "http://localhost/xc/tutorial/loricon/productsearch"', 'MAP \$input/PRODUCTREQUEST/SKU TO HTMLScreenDoc.XPath("/descend:)', 'CALL removeCheckedAttr(HTMLScreenDoc, HTMLScreenDoc.XPath("/descend:)', 'MAP true TO HTMLScreenDoc.XPath("/descendant::form[1]/input[@name =', 'MAP true TO HTMLScreenDoc.XPath("/descendant::form[1]/input[@name =', 'HTTP/HTML Get From "http://localhost/xc/tutorial/proding"', and 'MAP \$HTMLScreenDoc.html/body/table[2]/tr/td[2]/font TO \$Output:PRODUC'. The 'CALL removeCheckedAttr' action is highlighted in blue, indicating it is selected.

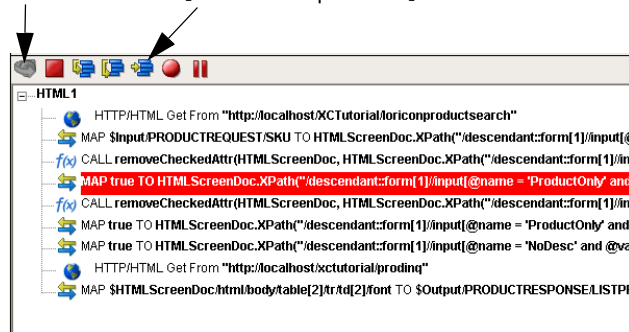
- 3 [Toggle Breakpoint] ボタンをクリックします (または、<F2> を押します)。選択したアクションが赤色になります。

[Toggle breakpoint]

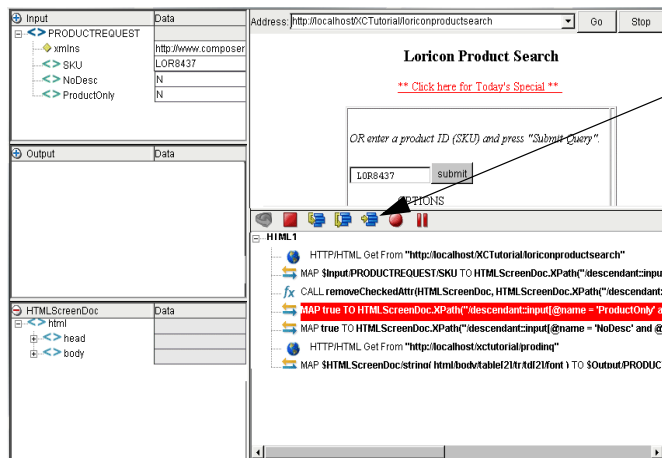


- 4 [Start Animation] ボタンをクリックします。アニメーションツールが有効になります。

[Start animation] [Run to Breakpoint/End]

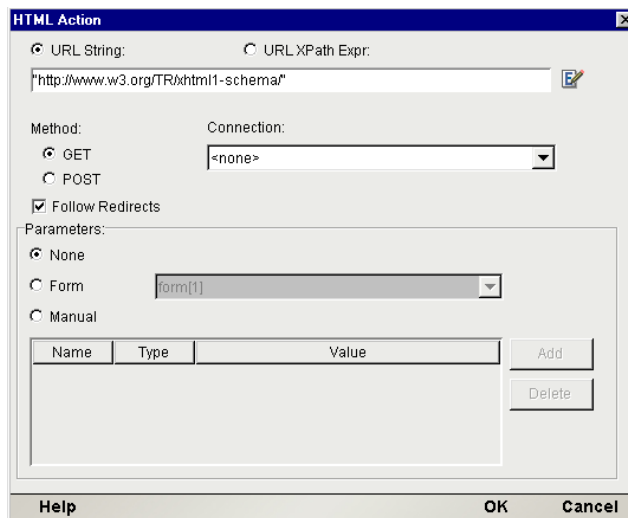


- 5 [Run to Breakpoint/End] ボタンをクリックします。アクションモデルは、最初から手順 3 で設定したブレーポイントまでのすべてのアクションを実行します (次の図を参照)。



[Run to breakpoint end]

- 6 アニメーションツールバーで、[Pause] ボタンをクリックします。
- 7 コンポーネントエディタのツールバーで、[Record] ボタンをクリックします。
- 8 [Action]、[New Action]、[Http/Html Action] から、アクションの行とマウスの右ボタンを選択します。[HTML Action] ダイアログボックスが表示されます。この章の「リンクおよびパラメータの編集」の詳細については、前のセクションを参照してください。



- 9 画面の必須フィールドを入力した後で、[OK] ボタンをクリックします。選択した行のすぐ下に新しいアクションが追加されます。
- 10 [Record] ボタンをクリックして、記録モードをオフにします。

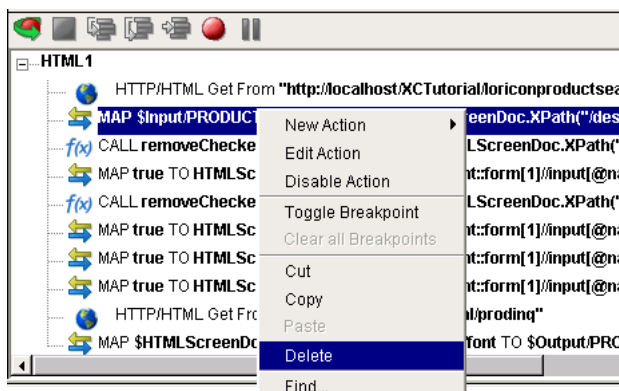
- 11 [File]、[Save] の順に選択するか、コンポーネントエディタのツールバーで [Save] ボタンをクリックします。
- 12 「アニメーションツールの使用」の指示に従って、コンポーネントをテストします。
- 13 選択した行のすぐ後に新しいアクションが追加されます。

アクションの削除

次の手順では、以前に記録したセッションでアクションを削除する方法を説明します。

➤ 以前に記録したアクションモデルでアクションを削除する

- 1 削除するアクションの行を選択して、マウスを右クリックし、メニューから [Delete] を選択します。行を選択して、キーボードの <Delete> ボタンを押すこともできます。

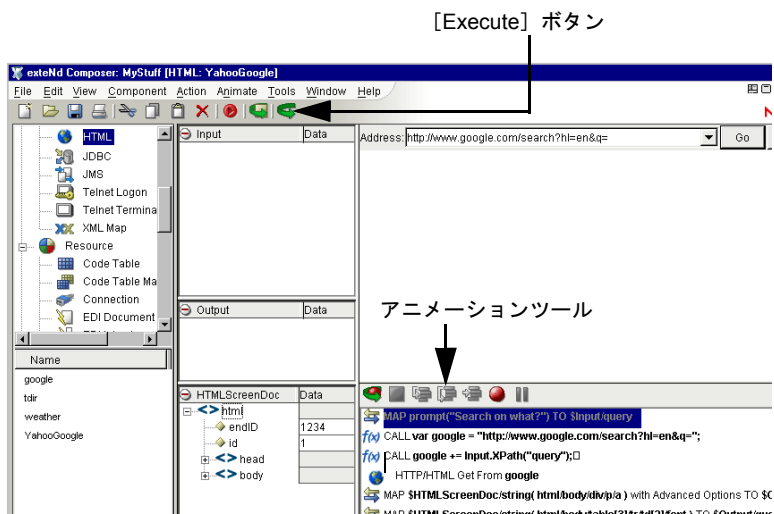


HTML コンポーネントの実行

Composer には、コンポーネントをテストできるアニメーションツールが含まれています。HTML コンポーネントエディタのツールバーには、[Execute] ボタンがあり、このボタンを使用するとアクションモデル全体を実行して、コンポーネントが意図したとおりに動作するか検証できます。

➤ HTML コンポーネントを実行する

- 1 HTML コンポーネントを開きます。HTML コンポーネントエディタウィンドウが表示されます。



- 2 [Execute] ボタンを選択します。アクションモデルのアクションが実行され、アクションの完了時にメッセージが表示されます。



- 3 [OK] をクリックします。
- 4 [View] メニューから、[Expand XML Documents] を選択します。これにより、XML ドキュメントのペアレント、チャイルド、データ要素などがすべて展開され、実行されたコンポーネントの結果を確認できます。XML ドキュメントを展開しないと、指定したデータが、HTML 環境から出力 DOM に移動しているかどうかを確認できません。

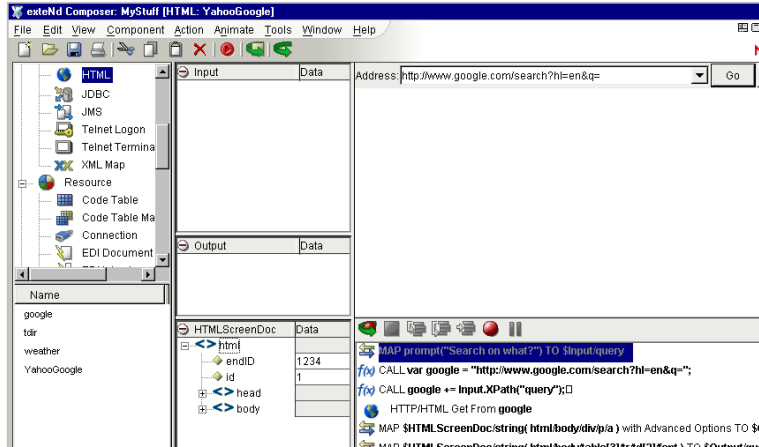
アニメーションツールの使用

アクションモデルには、1 つまたは複数のブレイクポイントを設定することにより、アクションモデルの特定のセクションをテストできるアニメーションツールがあります。これらのツールを使用すると、適切に動作するアクションをすべて実行して、問題の生じたアクションで停止してから、問題のアクションを1 つずつトラブルシューティングすることができます。

注記： アニメーションツール機能の簡単な例は、次のとおりです。すべてのアニメーションツールおよびその機能の詳細については、『exteNd Composer ユーザガイド』を参照してください。

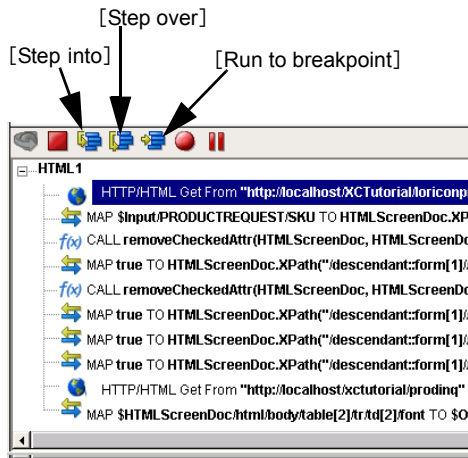
➤ アニメーションを実行する

- 1 HTML コンポーネントを開きます。HTML コンポーネントエディタウィンドウにコンポーネントが表示されます。

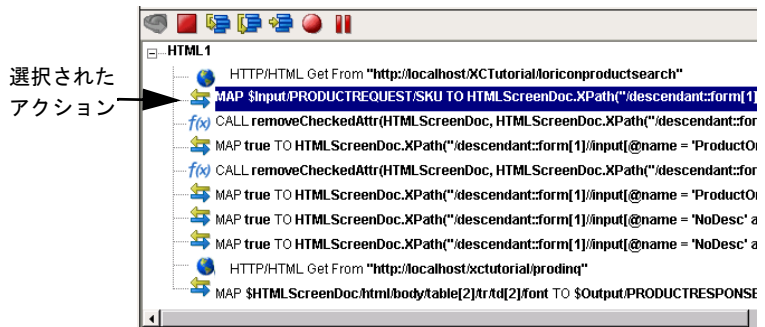


注記： アニメーションモードと記録モードは、コンポーネントでは互いに排他的なモードです。アニメーション中に記録を行うには、アニメーションを一時停止または停止してから、記録モードをオンにする必要があります。

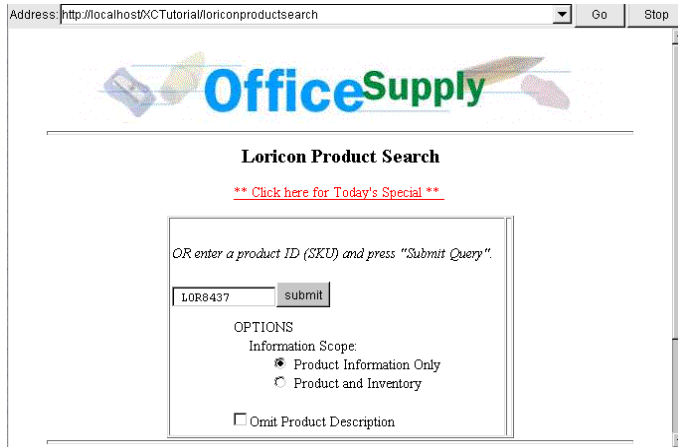
- 2 アクションモデルのツールバーで [Start Animation] ボタンをクリックするか、またはキーボードの <F5> キーを押します。ツールバーのすべてのツールが有効になります。



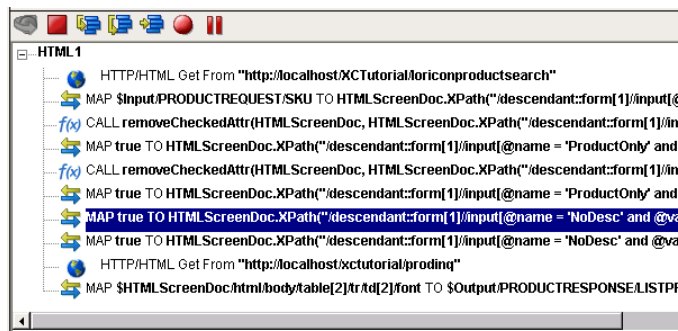
- 3 [Step Into] ボタンをクリックします。HTML アクションが選択されます。
 [Step Into] ボタンをもう一度クリックします。最初の Map アクションが選択されます。



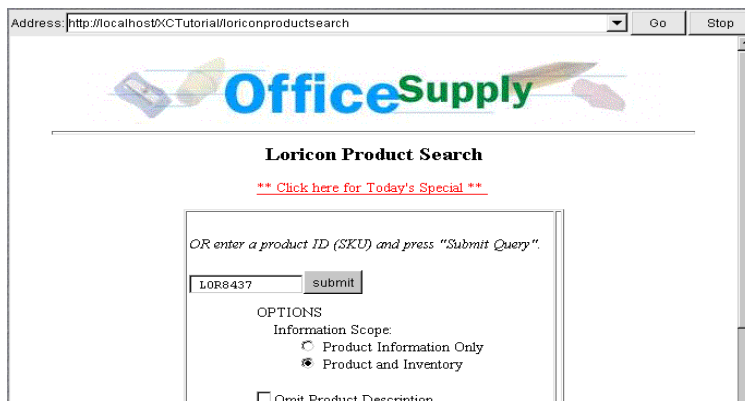
- 4 [Step Into] ボタンをもう一度クリックします。ネイティブ環境ペインでは、SKUが入力されています。



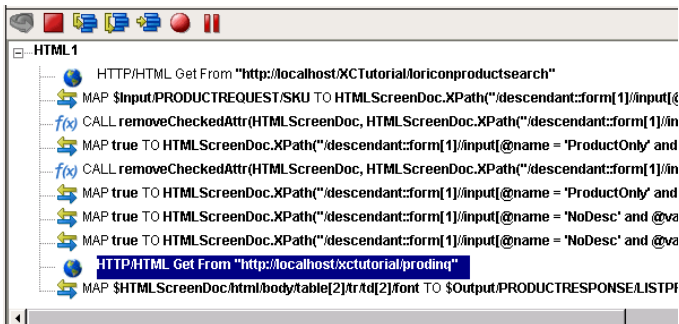
- 5 [Step Into] ボタンをもう一度クリックします。次に、Map アクションに到達するまで [Step Into] をもう一度クリックします。



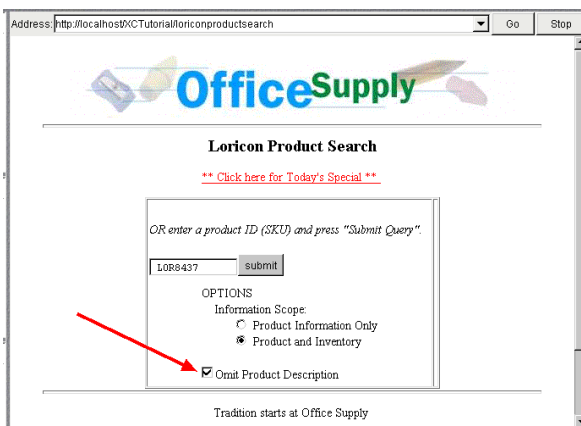
- 6 ネイティブ環境ペインでは、ラジオボタンが変更しています。



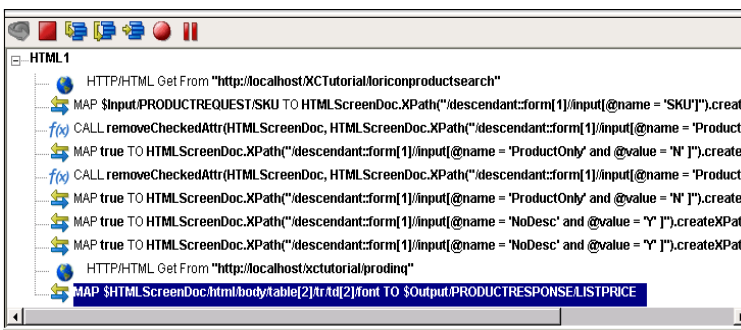
[Step Into] ボタンをもう一度クリックします。



7 ネイティブ環境ペインでは、[Omit Product Description] にチェックマークの選択が表示されています。



8 [Step Into] ボタンをもう一度クリックします。アクションモデルでは、選択されたアクションは HTML アクションです。

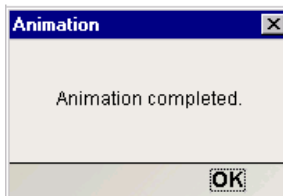


- 9 [Step Into] ボタンをもう一度クリックします。次のアクションにステップインすると、ネイティブ環境ペインには次の Web ページ画面が反映されます。

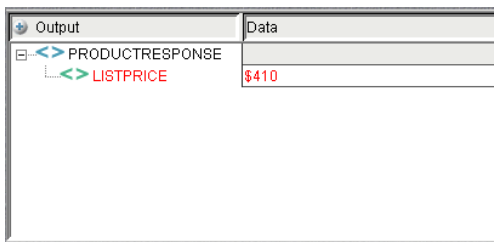


- 10 [Step Into] ボタンをもう一度クリックします。

- 11 完了すると、次のメッセージが表示されます。



- 12 [OK] をクリックします。出力 DOM には、[ListPrice] フィールドにドラッグアンドドロップによる情報「\$410」が表示されています。次の図を参照してください。



HTML コンポーネントエディタでの他のアクションの使用

Map アクションに加えて、HTML コンポーネントエディタでは、その他すべての標準コンポーネント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	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 Decision または Repeat、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。
Function	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。ログのタイプには、システム出力、システムログ、およびユーザーログの 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	入力値とケース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/else (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。

次の表の高度なアクションは、Composer の [Actions] メニューまたはアクションモデルのコンテキストメニューから利用できます。

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ことができます。
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 (<i>begin</i> 、 <i>commit</i> 、および <i>rollback</i>) 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。
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。

次に説明されている Data Exchange アクションでは、I/O 処理がアクションとして実行されます。

Data Exchange アクション	説明 ^
URL/File Read	XML 以外のファイル形式を Composer に読み込むことができます。
URL/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、File、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 メソッドが含まれます。

次に説明されている Repeat アクションでは、制御フロー構成要素のループが実行されます。

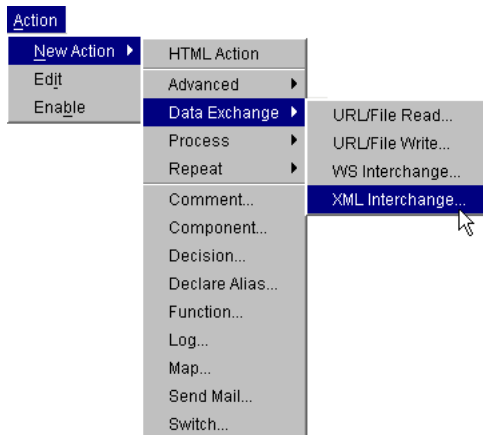
Repeat アクション	説明
Break	Repeat for Element、Repeat for Group、または Repeat While ループの実行を停止し、ループ外で次のアクションの実行を続行します。
Continue	Repeat for Element、Repeat for Group、または Repeat While ループで現在のループ反復の実行を停止し、次の反復で同じループの一番上から続行します。
Declare Group	複数回発生する要素に基づいてグループを作成し、グループに名前を付けることができます。グループは、Repeat for Group アクションで使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

XML Interchange アクションの使用

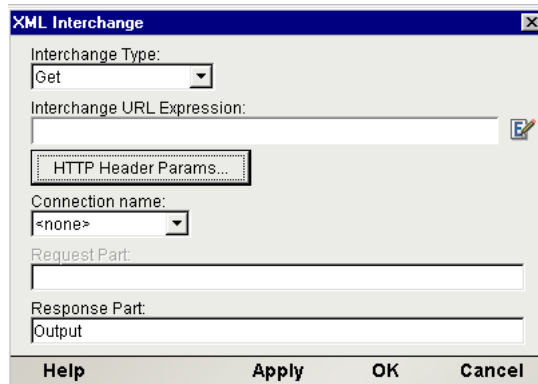
XML Interchange アクションを使用すると、特定の URI から外部 XML ドキュメントを DOM のコンポーネントに読み書きできます。

➤ XML Interchange アクションを使用する

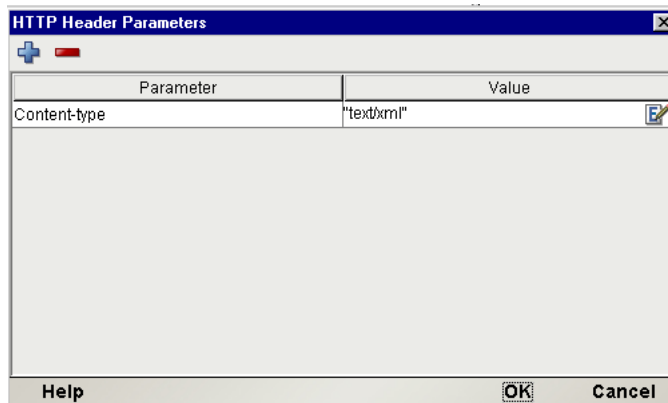
- 1 新しい XML Map コンポーネントを作成し、メインメニューバーで **[Action]**、**[New Action]**、**[Advanced]**、**[XML Interchange]** の順に選択します。



- 2 ダイアログボックスが表示されます。



- 3 ドロップダウンリストから、[Interchange Type] を選択します。[Get]、[Put]、[Post]、または [Post with response] を選択できます。
- 4 [Interchange URL Expression] に入力するか、[Source Expression] アイコンをクリックします。
- 5 ヘッダにパラメータおよび値を追加または削除する場合は、[HTTP Header Params] ボタンをクリックします。次のダイアログボックスが表示されます。

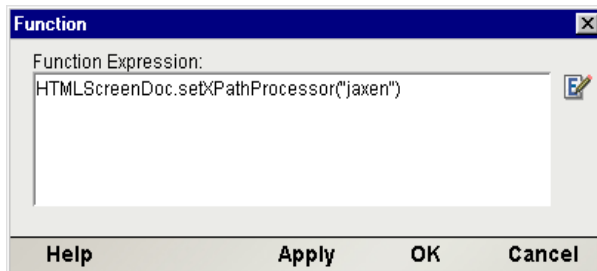


- 6 行を追加するにはプラス記号アイコン(+)をクリックします。行を削除するにはマイナス記号アイコン(-)をクリックします。[Source Expression] アイコンをクリックして、ヘッダの値の式を入力します。
- 7 [OK] ボタンをクリックして、[HTTP Header Parameters] ダイアログボックスに戻ります。[OK] をもう一度クリックすると、[XML Interchange] ダイアログボックスに戻ります。
- 8 ドロップダウンリストから [Connection Name] を選択します。
- 9 [Document Handle] フィールドは、デフォルトの値「Output」に変更します。この [Document Handle] の名前は、[Interchange Type] フィールドでの選択項目に応じて変更します。
- 10 [Post Response Document Handle] は、[Interchange Type] ドロップダウンリストから [Post Response] を選択した場合のみアクティブになります。
- 11 [OK] をクリックすると、Map アクションが選択項目に応じて作成されます。

パフォーマンス

HTML ページの DOM 表現が大きく複雑である場合、XPath 評価にはかなりの時間を要する可能性があります。XPath 評価で費やされる時間は、ページがどの程度複雑であるかという問題だけではなく、使用される XPath プロセッサによって大幅に異なります。exteNd Composer HTML Connect によって使用されるデフォルトの XPath エンジンである Xalan プロセッサでは深さ優先、ノードウォーキング (depth-first node-walking) アルゴリズムが使用され、小サイズから中サイズであまり複雑でない HTML ページに対しては好ましいパフォーマンスが得られます。反対に、Jaxen XPath プロセッサでは幅優先 (breadth-first) アルゴリズムが使用され、複雑で大きい DOM に対してより良いパフォーマンスが得られます。

exteNd Composer HTML Connect では、使用を希望する XPath エンジンを選択できます。プロセッサを変更するには、対象の DOM オブジェクトに対して（通常は `HTMLScreenDoc.setXPathProcessor()`）を呼び出します。引数文字列には、「xalan」（デフォルト）または「jaxen」を指定します。たとえば、新しい Function アクションを作成して、[Function] ダイアログボックスに式を入力します（次を参照）。



負荷テストまたはパフォーマンスの調整を行なう場合、それぞれの XPath プロセッサを試すことによって、使用する特定のアプリケーションに対してどちらのプロセッサが適しているかを確認してください。

注記： いずれかの XPath プロセッサを使用してアプリケーションを実行している場合にはバグが存在するものの、もう一方のプロセッサでは問題がない場合、お使いのアプリケーションが指定の順序で (XPath 式から) 返される DOM ノードに依存していることが原因であると十分に考えられます。ノードリストを返す XPath 式を使用する場合、ノードリスト内のノードの順序は予測不可能であることに注意してください。この動作は、すべてのプロセッサを通じて見られる標準の XPath の動作です。お使いのアプリケーションが特定の順序で返されるノードオブジェクトに依存する場合はこのような問題が生じることに注意してください。

エラーおよびメッセージの処理

ページの取得または解析に関わるエラーが生じると、ネイティブ環境ペインに短いエラーメッセージが表示されます。たとえば、過度に JavaScript を使用するサイトを誤って入力した場合、次のようなメッセージがネイティブ環境ペインに表示される可能性があります。

Web サイトにアニメーション表示される GIF ファイルが含まれている場合、Connect の動作が遅くなるか、より多くのメモリが必要となるか、あるいはその両方の状況が生じる可能性があります。これは、設計時のみにおける問題です。



Get 操作が正常に行なわれない場合、HTTP ステータスコードは ECMAScript を通じて `HTMLStatusCode` プロパティで利用できます。また、`HTMLStatusMsg` というプロパティにはエラー状況に関する情報を示す文字列が含まれています。これらのプロパティのコンテンツは、`Log` アクションを使用してログメッセージで取得できます (`Log` アクションの詳細については、『*Composer ガイド*』の「“The Log Action”」のセクションを参照してください)。

注記： HTTP ステータスコードのリストについては、「付録 C」を参照してください。

A

デジタル証明書

HTML Connect では、jar ファイル *xerootca.jar* において 98 種類のデジタル証明書がサポートされています。jar ファイルに証明書を追加する必要がある場合、次の表で説明されている手順を使用します。

exteNd でサポートされている認証証明書
ABA.ECOM Root CA
Baltimore EZ by DST
Belgacom E-Trust Primary CA
CA Data
Certiposte Classe A Personne
Certiposte Serveur
Certisign - Autoridade Certificadora - AC2
Certisign - Autoridade Certificadora - AC4
Certisign - Autoridade Certificadora - AC1S
Certisign - Autoridade Certificadora - AC3S
Class 1 Primary CA
Class 1 Public Primary Certification Authority - G2
Class 1 Public Primary Certification Authority - G2_2
Class 1 Public Primary Certification Authority
Class 1 Public Primary Certification Authority_2
Class 2Primary CA

exteNd でサポートされている認証証明書

Class 2 Public Primary Certification Authority - G2

Class 2 Public Primary Certification Authority - G2_2

Class 2 Public Primary Certification Authority - G2

Class 2 Public Primary Certification Authority- G2_2

Class 3 Primary CA

Class 3 Public Primary Certification Authority - G2

Class 3 Public Primary Certification Authority - G2_2

Class 3 Public Primary Certification Authority

Class 3 Public Primary Certification Authority_2

Class 3P Primary CA

Class 3TS Primary CA

Class 4 Public Primary Certificate Authority - G2

Class 4 Public Primary Certificate Authority - G2_2

Commercial Certification Authority

Deutsche Telekom Root CA 1

Deutsche Telekom Root CA 2

DST (ANX Network) CA

DST (NRF) RootCA

DST (UPS) RootCA

DST RootCA X1

DST RootCA X2

DST-Entrust GTI CA

DSTCA E1

DSTCA E2

Entrust.net Secure Server Certification Authority

Equifax Secure eBusiness CA

Equifax Secure Global eBusiness CA

FESTE, Public Notary Certs

exteNd でサポートされている認証証明書

FESTE, Verified Certs

First Data Digital Certificates Inc. Certification Authority

FNMT Clase 2 CA

getcacert

GlobalSign Root CA

GTE CyberTrust Global Root

GTE CyberTrust Root

GTE CyberTrust Root_2

IPS SERVIDORES

KeyWitness 2048 Root

Microsoft Authenticode(tm) Root Authority

Microsoft Corporation

Microsoft Root Authority

NetLock Expressz (Class C) Tanusitvanykiado

NetLock Kozjegyzoi (Class A) Tanusitvanykiado

NetLock Uzleti (Class B) Tanusitvanykiado

PTT Post Root CA

Secure Server Certification Authority

SecureNet CA Class A

SecureNet CA Class B

SecureNet CA Root

SecureNet CA SGC Root

SecureSign RootCA1

SecureSign RootCA2

SecureSign RootCA3

SERVICIOS DE CERTIFICACION - A.N.C.

SIA Secure Client CA

SIA Secure Server CA

exteNd でサポートされている認証証明書

Swisskey Root CA

TC TrustCenter Class 1 CA

TC TrustCenter Class 2 CA

TC TrustCenter Class 3 CA

TC TrustCenter Class 4 CA

TC TrustCenter TimeStamping CA

Thawte Personal Basic CA

Thawte Personal Freemail CA

Thawte Personal Premium CA

Thawte Premium Server CA

Thawte Server CA

Thawte Timestamping CA

UTN - STATCorp SGC

UTN - USERFirst-Client Authentication and Email

UTN - USERFirst-Hardware

UTN - USERFirst-Network Application

UTN - USERFirst-Object

ValiCert Class 1 Policy Validation Authority

ValiCert Class 2 Policy Validation Authority

ValiCert Class 3 Policy Validation Authority

VeriSign Commercial Software Publishers CA

VeriSign Commercial Software Publishers CA_2

VeriSign IndividualSoftware Publishers CA

VeriSign IndividualSoftware Publishers CA_2

VeriSign, Inc.

Xcert EZ by DST

➤ **信頼できる他の認証局を exteNd に追加する**

- 1** WinZip ユーティリティプログラムで、**Designer/lib/xcroota.jar** ファイルを開きます。
- 2** [Add] をクリックして、ファイルを追加するディレクトリを参照します。
- 3** **xcroota.jar** ファイル内でリストに表示されているその他のファイルにファイルを追加します。WinZip プログラムを閉じます。
- 4** サーバを再起動します。

B

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では、かなり環境的な違いがあります。これらのテストはいずれも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

比較項目	Composer でのテスト	配備テスト
HTML ページでのイメージ	ネイティブ環境ペイン内でレンダリングされます。	レンダリングされません。
HTTP ハンドラ	[Tools] の [Configuration] からハンドラを割り当てることができます(注: 設計およびテスト用には Composer Handler を使用します)。	アプリケーションサーバにより自動的に選択されます。
エラー処理	ネイティブ環境ペインにエラーメッセージが表示されます。	Log アクションにより HTTP ステータスコードをキャプチャできます。
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します(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 上の場所を指します。

比較項目	Composer でのテスト	配備テスト
テストツール	Log アクションの他に、ECMAScript の <i>alert()</i> 関数によりダイアログボックスを使用してランタイム値を表示できます。	ダイアログボックスは使用できません。

C

HTTP ステータスコード

HTTP ステータスコードは、特定の HTTP 要求の結果を要約する 3 桁の整数から成る結果コードです。これらのコードを次に要約します。ステータスコードの最初の 1 桁は応答クラスを定義します。最後の 2 桁は特定の「カテゴリ」上の意味をもちません。

コードの最初の 1 桁に基づくステータスコードには 5 クラスあります。

1xx	情報	要求を受信しました、プロセスを続行します
2xx	成功	要求を正常に受信し、理解し、受け付けました
3xx	リダイレクション	要求を完了するためにさらにアクションが必要です
4xx	クライアントエラー	要求に不正な構文が含まれているか、処理できません
5xx	サーバエラー	サーバは明白に有効な要求を処理できませんでした

詳細コードの意味

- 100 続行
- 101 プロトコルの切り替え
- 200 OK
- 201 作成日
- 202 受諾
- 203 非認定情報
- 204 コンテンツなし
- 205 コンテンツのリセット

- 206 部分コンテンツ
- 300 複数選択肢
- 301 完全に移動
- 302 検出
- 303 その他を参照
- 304 変更なし
- 305 プロキシーを使用
- 307 一時的にリダイレクト
- 400 不正要求
- 401 未許可
- 402 支払いが必要
- 403 禁止
- 404 未検出
- 405 メソッド未許可
- 406 受諾不可
- 407 プロキシー認証が必要
- 408 要求タイムアウト
- 409 重複
- 410 ダウン
- 411 長さが必要
- 412 事前条件失敗
- 413 要求エンティティが大きすぎる
- 414 要求 URI が大きすぎる
- 415 未サポートメディアタイプ
- 416 要求範囲を満たすことができない
- 417 予想が失敗した
- 500 内部サーバエラー
- 501 未実装
- 502 不正なゲートウェイ

503 サービス使用不可

504 ゲートウェイタイムアウト

505HTTP バージョン未サポート

拡張コード=3桁理由フレーズ =*`<テキスト、CR、LF を除外>`

注記： HTTP ステータスコードは広い範囲にわたります。HTTP アプリケーションは、すべての登録ステータスコードの意味を理解することが望ましいことは明白ですが、必ずしも理解することを必要とはしません。

D

フォームフィールド値をインタラクティブに変更した場合に作成されたアクション

記録モードを使用して、ネイティブ環境ペインに現在表示されている Web ページ内でフォームフィールドを操作する場合、適切なアクションがリアルタイムで自動的にアクションモデルに追加されます。たとえば、HTML ページでテキストをテキストフィールドに入力すると、Map アクションが自動的に生成されます。さまざまなフィールドタイプを操作した場合に、アクションモデルで自動生成されるアクションのタイプは、次の表のとおりです。

フィールドタイプおよびユーザーによる操作	アクションモデルで作成されるアクション
ユーザーがテキストフィールドにテキストを入力（またはドラッグ）する	Map アクション： MAP string TO HTMLScreenDoc.XPath(path)
ユーザーがチェックボックスをオンにする	Map アクション。例： MAP true TO HTMLScreenDoc.XPath("//input[@name = 'safe' and @value = 'on']).createXPath("/@checked")
ユーザーがチェックボックスをオフにする	Function アクション。例： CALL removeCheckedAttr(HTMLScreenDoc.XPath("//input[@name = 'safe']));
ユーザーがラジオボタンをオンにする	Map アクション。例： HTMLScreenDoc.XPath("//input[@name = 'safe' and @value = 'on']).createXPath("/@checked")

フィールドタイプおよび ユーザによる操作	アクションモデルで作成されるアクション
ユーザがラジオボタン をオフにする	Function アクション。例： CALL removeCheckedAttr(HTMLScreenDoc.XPath("//input[@name = 'safe']"));
ユーザがドロップダウン リストから項目を選 択する	Function アクション。例： updateSelectedAttr(HTMLScreenDoc.XPath("/descendant::select[@name = 'num']/option"), '20');
ユーザがリストボッ クスで項目を切り替 える	Function アクション： CALL toggleListSelectedAttr(DOMElement, string, boolean) 注記： ソースコードとその使用法、および関連カスタム ECMAScript 関数については、次の付録を参照してください。
ユーザが [Submit] ボ タンをクリックする	Mapアクション、およびその後HTTPポスト(通常)。例： CALL HTMLScreenDoc.XPath("//input[@name = 'Mode' and @value = 'SEARCH']").createXPath("/@checked") HTTP/HTML Post From "http://www.xyz.com/search.php"

E

記録済みの Function アクションで使用される内部スクリプト

```
// 次に説明する ECMAScript の関数は、ユーザが HTML フォームによる
// 表示の制御について特定の選択を行う場合、HTML Connect で
// 作成された Function アクションから実行されます。
/*****/
// 関数名： removeSelectedAttr(nodelist)
// 説明： 選択された属性を制御から削除します。
// ノードリスト： 必要。属性を削除するノードのリストです。
// 返す値： なし
// 注： DOM 要素の getAttribute() および removeAttribute() メソッドを
//      使用します。
/*****/
function removeSelectedAttr(nodelist)
{
    for(var elem in nodelist)
    {
        var attr = elem.getAttribute("selected");
        if(attr != null)
            elem.removeAttribute("selected");
    }
}

/*****/
// 関数名： updateSelectedAttr(nodelist, asRExpr)
// 説明： コンボボックスに対するリストの選択を更新するメソッド。コンボボックスでは
```

```

//1つの値だけ選択できます。このメソッドでは渡される式が比較され、渡されたパラメータと
// 属性が一致する1つの要素に対して選択された属性が
// 作成されます。選択された属性が存在する場合、
// 残りの項目に対して選択された属性は削除されます。
ノードリスト：必要。オプションノードのリストです。
// asRExpr：必要。右側の式です。
// 返す値：なし
注：DOM要素のgetAttribute() および removeAttribute() メソッドを使用します。
/*****
/
function updateSelectedAttr(nodelist, asRExpr)
{
    for(var elem in nodelist)
    {
// ノード値を取得します。exteNdが呼び返されます。
        var lVal1 =
Packages.com.sssw.b2b.rt.GNVXMLDocument.getNodeStringValue(elem);

        // 比較の前に式の両側にある
        // 空白を削除します。
        var lVal = new java.lang.String(lVal1);
        lVal = lVal.trim();
        var rVal = new java.lang.String(asRExpr);
        var rVal = rVal.trim();
        if(lVal == rVal)
            elem.setAttribute("selected", "selected");
        else
        {
            var attr = elem.getAttribute("selected");
            if(attr != null)
                elem.removeAttribute("selected");
        }
    }
}

```

```

/*****/
// 関数名 : removeCheckedAttr(nodelist)
// 説明 : ラジオボタン、チェックボックス、プッシュボタンなど
// 「checked」属性を削除するメソッド。
ノードリスト : (必要) オフにする必要がある
// 「checked」属性のノードのリスト。
// 返す値 : なし
注 : DOM 要素の getAttribute() および
//           removeAttribute() メソッドを使用します。
/*****/
function removeCheckedAttr(nodelist)
{
    for(var elem in nodelist)
    {
        var attr = elem.getAttribute("checked");
        if(attr != null)
            elem.removeAttribute("checked");
    }
}

*****/
// 関数名 : toggleListSelectedAttr(elem, asValue, abAddAttr)
// 説明 : ブール演算の渡されるパラメータである abAddAttr が
//           True (真) の場合は複数選択可能な
//           リストボックスの選択した属性を作成し、abAddAttr が False (偽)
//           の場合は
//           選択した属性削除するメソッド。
// elem:      作成または削除される「selected」属性の
//           //DOM 要素。
// asValue:   「オプション」の子要素に一致する値。
ブール演算子。True (真) の場合、asValue と
//           値が一致する「option」要素に対し
// 「selected」属性を作成し、

```

```

//                フラグが False ( 偽 ) の場合は
// 「selected」 属性を削除します。
// 返す値： なし
注： DOM 要素の getAttribute() および
//                removeAttribute() メソッドを使用します。
/*****/
function toggleListSelectedAttr(elem, asValue, abAddAttr)
{
    // 「option」 名ですべての子要素を検索します。
    var lOptionList = elem.getElementsByTagName("option");
    for(var child in lOptionList)
    {
        // 子に対する値の属性を取得します。
        var lsChildAttrVal = child.getAttribute("value");
        if(lsChildAttrVal == asValue)
        {
            // abAddAttr が True ( 真 ) の場合、渡される値と値が一致する子に対して
            // 属性が作成され、そうでない場合は削除されます。
            if(abAddAttr == true)
            {
                child.setAttribute("selected", "selected");
            }
            else
            {
                child.removeAttribute("selected");
            }
            break;
        }
    }
}

```

F

HTML 用語集

HTML

Hyper Text Markup Language

HTTP

HyperText Transfer Protocol。

HTTP GET

ターゲット URI に追加される URlencoded フォームデータを使用して指定された URI でファイルを取得する方法。

HTTP POST

指定された URI にデータを投函する方法 (通常は、サーバの CGI スクリプトをポイントします)。

HTTPS

HTTP の「安全な」バージョン。

Map Screen アクション

新しいデータを受信するアクションモデルの場所を示す特別なエディット可能でないアクション。このスクリーンと対話する任意のアクションは、アクションモデルの Map アクションの Screen アクションラインに従属する位置に配置します。

ScreenDoc

XML ドキュメントとして現在の HTML ページを表す HTML Component Editor で表示できる特別な DOM。

SSL

Secure Socket Layer Support

TIDY

読みにくいHTMLを明確にレイヤされたマークアップテキストに変換するために使用されるユーティリティ。

URI

Uniform Resource Locator。URI は、Web リソースの場所だけではなく、通信するためにしようするプロトコルを表示します。

Web ブラウザ

HTML ドキュメントの実際の内容が表示される表示ウィンドウ。

XHTML

XML 標準の最低の「フォーム表示」要件を満たす HTML。

クッキー

状態情報を格納するテキストファイルまたは文字列。一部のクッキーはHTTPセッションのみを対象とし (クライアントアプリケーションがシャットダウンすると存在しなくなる)、他のクッキーは記憶デバイスに書き込まれることにより *持続的* になります。

デジタル証明書

Web サーバとの安全な通信を可能にします。

フィールド

フォームに含まれるデータの単位。フィールドは画面上に表示するラベル、データの項目、ある種の対話ウィジェットです。非表示のフィールドもあります (その場合はビジュアルインタフェースはない)。各フィールドには、表示方法と領域を変更できるかどうかを決める固有の属性があります。

フレーム

一部の HTML ページはドラッグ可能な分割線で区切られたペインで表示されます。これらのペインはフレームと呼ばれます。FRAMESET タグは、フレームのグループを作成するために使用されます。各フレームには、固有の FRAME タグがあります。フレームを含む HTML ページのソースコードには、個々のフレームの詳細な HTML コードは含まれません。代わりに、FRAME 要素の SRC 属性は、フレームの適切な URI にブラウザを導きます。

ネイティブ環境ペイン

実際の HTML 端末セッションのエミュレーションを表示する、HTML コンポーネントエディタ内のペイン。

リダイレクション

クライアントを新しい宛先 URI に送る試み。リダイレクションは、HTML ページの `<META>` エLEMENTの `Refresh` ディレクティブ、(応答の HTTP ヘッダの) サーバにより発行される特別な `Refresh` ディレクティブ、または HTML ページ内のスクリプト (一般的な JavaScript) の結果として発生することがあります。

G

予約語

次の用語は、`exteNd Composer HTML Connect` の予約語であるため、ユーザ作成ラベルまたは変数名では使用しないでください。

- `Get`
- `Post`
- `Put`
- `Redirect`
- `HTML`
- `ScreenDoc`
- `FrameSet`
- `theComponent`

索引

A

ActiveX 11
[ADD] ボタン 47

C

Certisign 69
[Component] メニュー 35
CyberTrust 71

D

Data Exchange アクション 62
[DELETE] ボタン 47
[Document Handle] 65

E

ECMAScript 25
 エラートラップ 67
eXtend Connects
 概要 10

F

Function アクション 41, 82

H

HTML 87
HTMLScreenDoc 41
HTMLStatusCode プロパティ 67
HTMLStatusMsg 67
HTML アクションの編集 43, 46
HTML 固有のボタンについて 36
HTML 固有のメニューバー項目について 35
HTML コンポーネントの実行 54

HTML コンポーネント、作成 29
HTML 接続リソース 22
HTTP GET 87
HTTP POST 87
HTTPS 22
HTTP ステータスコード 77
HTTP ハンドラ 18
HTTP ハンドラタイプ 19

J

JavaScript 15, 66
Jaxen 65
JScript 15

M

Map Screen アクション、定義 87
Map アクション 41

N

NetLock 71

P

Post、HTTP 37

R

[Redirect] チェックボックス 47
removeCheckedAttr() 関数 85
removeSelectedAttr() 関数 83
Repeat アクション 63
[Run to Breakpoint/End] 50, 52

S

ScreenDoc DOM 32
ScreenDoc、定義 87

SecureNet 71
Secure Socket Layer サポート 13
setXPathProcessor() メソッド 66
SSL 13, 87
SSL3 32
[Step into] 58

T

TIDY 11, 88
[Toggle Breakpoint] 50, 51
toggleListSelectedAttr() 関数 86

U

updateSelectedAttr() 関数 84
URL 88
URL アクション 37

V

ValiCert 72
VeriSign 72
[View] メニュー 35

W

Web ブラウザ 88
[Window Layout] 33

X

Xalan 65
xcrootca.jar 13
XHTML 11, 33, 88
XML Interchange アクションの使用 63
XML テンプレート 12
XML ドキュメントの展開 55
XPath 66
XPath プロセッサ (Xalan/Jaxen) 65
XSL 75

あ

アクション

記録済みの Function アクションで使用される内部スクリプト 83

概要 37

基本

Comment 61
Component 61
Decision 61
Function 61
Log 61
Map 61
Send Mail 61

基本および高度の使用 61

フォームフィールド値変更時の作成 81

アクションの削除 54

[Action] メニュー 61

アクションモデル 37

以前の記録の編集 48

アクション、HTML 以外 61

新しいアクションの追加 51

アニメーション

実行 56

ツールの使用 55

アニメーションテストおよび配備テスト 75

え

エラーおよびメッセージ 66

か

カスタム ECMAScript 関数 83

環境の相違 75

き

既存のアクションの変更 49

基本的なアクション 61

Switch 61

記録 38

編集 48

く

クッキー 13, 14, 88
 持続的 13
 セッション 13
クライアント証明書 25

こ

高度なアクション 61, 62
コンポーネントエディタ 31

さ

サンプルトランザクション 17

し

持続的クッキー 14
自動生成アクション 81
詳細コードの意味 77
証明書 23, 69

せ

セッションクッキー 14
接続リソース 22

そ

ソースの表示 (HTML ダンプ) 14

た

正しく表示されない画像 19

て

デジタル証明書 13, 69, 88

と

ドラッグアンドドロップ 33, 41

トランザクション、サンプル 17

に

認証 23

ね

ネイティブ環境ペイン、定義 88

の

ノードオブジェクト 66
ノードの順序 66
ノードの順序、ノードリスト 66

は

パスワード 25
パフォーマンス 65
パラメータ 47

ふ

フィールド、ScreenDoc DOM への追加 47
フィールド、定義 88
負荷テスト 66
複数フレーム 43
複数フレームを使用した HTML セッションを
 記録する 43
プライベートキー 25
フレーム 14
フレームを使用した HTML セッションの記録 43

め

メソッドタイプ 47

り

リダイレクト 89
リンクおよびパラメータ、編集 46

