

Novell Identity Manager の Designer

3.5.1

September 28, 2007

www.novell.com

IDENTITY MANAGER ユーザアプリケーション
3.5.1: 設計ガイド



Novell®

保証と著作権

米国 Novell, Inc. およびノベル株式会社は、この文書の内容または使用について、いかなる保証、表明または約束も行っていません。また文書の商品性、および特定の目的への適合性について、いかなる黙示の保証も否認し、排除します。また、本書の内容は予告なく変更されることがあります。

米国 Novell, Inc. およびノベル株式会社は、すべてのノベル製ソフトウェアについて、いかなる保証、表明または約束も行っていません。またノベル製ソフトウェアの商品性、および特定の目的への適合性について、いかなる黙示の保証も否認し、排除します。米国 Novell, Inc. およびノベル株式会社は、ノベル製ソフトウェアの内容を変更する権利を常に留保します。

本契約の下で提供される製品または技術情報はすべて、米国の輸出規制および他国の商法の制限を受けます。お客様は、すべての輸出規制を遵守し、製品の輸出、再輸出、または輸入に必要なすべての許可または等級を取得するものとします。お客様は、現在の米国の輸出除外リストに掲載されている企業、および米国の輸出管理規定で指定された輸出禁止国またはテロリスト国に本製品を輸出または再輸出しないものとします。お客様は、取引対象製品を、禁止されている核兵器、ミサイル、または生物化学兵器を最終目的として使用しないものとします。ノベル製ソフトウェアの輸出に関する詳細については、www.novell.com/info/exports/ を参照してください。弊社は、お客様が必要な輸出承認を取得しなかったことに対し如何なる責任も負わないものとします。

Copyright © 2007 Novell, Inc. All rights reserved. 本ドキュメントの一部または全体を無断で複写・転載することは、その形態を問わず禁じます。

米国 Novell, Inc. は、本文書に記載されている製品に統合されている技術に関する知的所有権を保有します。これらの知的所有権は、<http://www.novell.com/company/legal/patents/> に記載されている 1 つ以上の米国特許、および米国およびその他の国における 1 つ以上の特許または出願中の特許を含む場合があります。

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

オンラインマニュアル: 本製品とその他の Novell 製品のオンラインヘルプにアクセスする場合や、アップデート版入手する場合は、www.novell.com&documentation をご覧ください。

Novell の商標

Novell は、Novell, Inc. の米国およびその他の国における登録商標です。
SUSE は、Novell, Inc. の米国およびその他の国における登録商標です。

サードパーティ資料

サードパーティの商標は、それぞれの所有者に属します。

目次

このガイドについて	7
1 ユーザアプリケーション設計ツールの概要	9
1.1 プロビジョニングビューについて	9
1.2 ディレクトリ抽象化レイヤエディタについて	10
1.3 プロビジョニング要求定義エディタについて	10
1.4 ECMA式ビルダについて	10
1.5 プロビジョニングチームエディタについて	11
1.6 プロジェクトのドキュメント生成	11
2 プロビジョニングビューの操作	13
2.1 プロビジョニングプロジェクトのセットアップ	13
2.1.1 ユーザアプリケーションドライバの作成	13
2.1.2 電子メール通知テンプレートについて	17
2.2 プロビジョニングビューにアクセスする	18
2.3 プロビジョニングビューの初期設定の設定	19
2.4 プロビジョニングオブジェクトのインポート	23
2.4.1 ドライバ環境設定ファイルからのインポート	23
2.4.2 識別ポールトからのインポート	24
2.5 プロビジョニングオブジェクトのエクスポート	24
2.5.1 ドライバ環境設定ファイルへのエクスポート	24
2.6 プロビジョニングオブジェクトの検証	25
2.7 プロビジョニングオブジェクトの展開	26
2.7.1 プロビジョニングオブジェクトの展開	26
2.7.2 展開された変更のテスト	27
2.8 プロビジョニングオブジェクトの比較	27
2.9 ロケールおよびローカリゼーションリソースグループの指定	28
2.10 表示ラベルのローカライズ	33
2.10.1 サポートされている言語	34
2.10.2 ディレクトリ抽象化レイヤの表示ラベルのローカライズ	34
2.10.3 ローカライズされたラベルのエクスポートとインポート	34
3 ディレクトリ抽象化レイヤの設定	39
3.1 ディレクトリ抽象化レイヤについて	39
3.1.1 ユーザアプリケーションのデータニーズの分析	39
3.1.2 ディレクトリ抽象化レイヤエディタについて	40
3.1.3 ディレクトリ抽象化レイヤエディタファイルについて	42
3.2 エンティティおよび属性の操作	43
3.2.1 エンティティおよび属性について	44
3.2.2 エンティティの追加	45
3.2.3 属性の追加	49
3.2.4 スキーマ要素リストの更新	51
3.3 リストの操作	51
3.4 クエリの操作	54
3.5 関係の操作	58
3.6 環境設定の操作	61

3.7	ディレクトリ抽象化レイヤプロパティのリファレンス	62
3.7.1	エンティティのプロパティ	62
3.7.2	属性のプロパティ	66
3.7.3	クエリのプロパティ	75
3.7.4	関係のプロパティ	76
4	プロビジョニング要求定義エディタの操作	79
4.1	プロビジョニング要求定義エディタについて	79
4.1.1	プロビジョニング要求定義エディタを Identity Manager アーキテクチャに適合させる方法	79
4.1.2	プロビジョニングおよびワークフローの例	81
4.2	プロビジョニング要求定義を作成する場合の基本ステップ	85
4.3	ワークフロー作成のガイドライン	86
4.3.1	アクティビティのルール	87
4.3.2	フローパスのルール	87
4.3.3	ワークフローデータについての理解	89
4.4	インストールされているテンプレートでの作業	93
4.5	ワークフローのデバッグ	95
5	プロビジョニング要求定義の作成	97
5.1	ウィザードおよび [概要] タブについて	97
5.2	ウィザードによるプロビジョニング要求定義の作成	99
5.2.1	テンプレートの使用	100
5.2.2	独自のプロビジョニング要求定義の作成	102
5.3	[概要] タブでの基本設定の変更	103
6	プロビジョニング要求定義のフォームの作成	107
6.1	フォームについて	107
6.1.1	フォームコントロールのデータバインディングについて	109
6.1.2	フォームおよびイベントについて	109
6.2	[フォーム] タブについて	110
6.2.1	[フォームの選択] について	111
6.2.2	[フォームコントロール] について	113
6.3	フォームの作成	115
6.3.1	新しいフォームの作成	115
6.3.2	フォームコントロールとアクションの追加	116
6.3.3	イベントの定義	118
6.3.4	[スクリプト] タブの使用	121
6.4	アクションのリファレンス	123
6.5	フォームコントロールのリファレンス	125
6.5.1	ユーザ入力コメントのコントロール	127
6.5.2	一般的なフォームコントロールのプロパティ	127
6.5.3	CheckBoxPickList	128
6.5.4	DatePicker	129
6.5.5	DateTimePicker	130
6.5.6	DNContainer	131
6.5.7	DNDisplay	132
6.5.8	DNLookup	133
6.5.9	DNMaker	137
6.5.10	DNQuery	138
6.5.11	Global List	139
6.5.12	HTML	140
6.5.13	MVCheckbox	140

6.5.14	MVEditor	141
6.5.15	PickList	146
6.5.16	スタティックリスト	148
6.5.17	テキスト	149
6.5.18	テキスト領域	150
6.5.19	役職	150
6.5.20	TrueFalseRadioButtons	150
6.5.21	TrueFalseSelectBox	151
6.6	識別名の操作	151
6.6.1	DN のフォーマット	151
6.6.2	オブジェクトセレクタの操作	152
6.7	フォームでの DAL クエリの使用	154
6.8	フォームの印刷	159
6.9	フォームへの直接アクセス	160

7 プロビジョニング要求定義のワークフローの作成 163

7.1	[ワークフロー] タブについて	163
7.1.1	キャンバス	164
7.1.2	パレット	165
7.1.3	ビュー	167
7.2	ワークフローへの動作の追加	167
7.2.1	動作の一般プロパティの設定	168
7.2.2	データ項目マッピングの定義	169
7.2.3	電子メール通知設定の定義	170
7.3	フローパスの追加	171
7.4	フローパスの設定	172
7.5	承認動作の宛先の指定	174
7.5.1	有効な宛先式	175
7.5.2	宛先と承認者タイプとの関係	175
7.6	1つのワークフローインスタンスによる複数の個人のプロビジョニング	179
7.6.1	ワークフローを使用するための基本手順	179
7.6.2	チームマネージャが使用するワークフローの設定	181
7.7	エンティティ動作を使用した作業	181
7.7.1	エンティティの追加または変更	181
7.7.2	エンティティ動作を使用したエンティティの削除	182
7.7.3	エンティティ動作を使用した属性または値の削除	182
7.8	識別名参照の移植可能化	183
7.9	デジタル署名サポートの設定	183
7.9.1	デジタル署名のワークフロープロパティ	184
7.9.2	署名宣言の作成	185

8 ワークフローアクティビティの参照 187

8.1	開始アクティビティ	187
8.1.1	プロパティ	188
8.1.2	データ項目マッピング	189
8.1.3	電子メール通知	190
8.2	承認アクティビティ	190
8.2.1	プロパティ	191
8.2.2	データ項目マッピング	197
8.2.3	電子メール通知	198
8.3	ログアクティビティ	199
8.3.1	プロパティ	200
8.3.2	データ項目マッピング	200
8.3.3	電子メール通知	201

8.4	プランチアクティビティ	201
8.4.1	プロパティ	201
8.4.2	データ項目マッピング	201
8.4.3	電子メール通知	201
8.5	マージアクティビティ	201
8.5.1	プロパティ	202
8.5.2	データ項目マッピング	202
8.5.3	電子メール通知	202
8.6	条件アクティビティ	202
8.6.1	プロパティ	202
8.6.2	データ項目マッピング	203
8.6.3	電子メール通知	203
8.7	マッピングアクティビティ	203
8.7.1	プロパティ	203
8.7.2	データ項目マッピング	204
8.7.3	電子メール通知	204
8.8	ワークフローステータス	204
8.8.1	プロパティ	204
8.8.2	データ項目マッピング	205
8.8.3	電子メール通知	205
8.9	電子メールアクティビティ	205
8.9.1	プロパティ	205
8.9.2	データ項目マッピング	206
8.9.3	電子メール通知	206
8.10	完了アクティビティ	207
8.10.1	プロパティ	208
8.10.2	データ項目マッピング	208
8.10.3	電子メール通知	208
8.11	統合アクティビティ	209
8.11.1	プロパティ	210
8.11.2	データ項目マッピング	210
8.11.3	電子メール通知	211
8.12	エンタイトルメントアクティビティ	211
8.12.1	プロパティ	212
8.12.2	データ項目マッピング	212
8.12.3	電子メール通知	213
8.13	エンティティアクティビティ	213
8.13.1	プロパティ	214
8.13.2	データ項目マッピング	214
8.13.3	電子メール通知	216
9	統合アクティビティの操作	217
9.1	統合アクティビティについて	217
9.2	統合アクティビティの追加	217
9.3	統合アクティビティとの間のデータの移動	219
9.4	統合アクティビティエディタインターフェースの使用	222
9.4.1	XML ビュー	223
9.4.2	アクションモデル	227
9.4.3	WSDL エディタ	234
9.4.4	メッセージ	235
9.4.5	アクションモデルのコードの再生成	235
9.4.6	アクションモデルへのアクションの追加	235
9.5	アクション	236
9.5.1	詳細	236
9.5.2	データ交換	243

9.5.3	繰り返し	248
9.5.4	コメント	255
9.5.5	決定	256
9.5.6	関数	257
9.5.7	ログ	258
9.5.8	マップ	260

10 ECMA 式の使用 269

10.1	ECMA 式ビルダについて	269
10.1.1	ECMAScript について	270
10.1.2	ECMAScript の機能	270
10.1.3	ECMA 式ビルダの使用	270
10.1.4	Java の統合について	276
10.1.5	XPath の統合について	277
10.1.6	パフォーマンスについて	279
10.2	ECMAScript の例	279
10.2.1	一般的な例	280
10.2.2	フローデータの例	280
10.2.3	フォームコントロールの例	280
10.2.4	エラー処理	281
10.3	ECMAScript API	282
10.3.1	フォームアクションスクリプトメソッド	282
10.3.2	DOM メソッド	292
10.3.3	ECMAScript Core	318
10.3.4	Global 関数	337
10.3.5	IDVault 関数	337

11 プロビジョニングチームの設定 339

11.1	チームの概要	339
11.1.1	プロビジョニングチーム要求について	340
11.1.2	チームを使ったダイレクトレポートの管理	341
11.2	プロビジョニングチームの管理	341
11.2.1	プロビジョニングチームの作成	341
11.2.2	プロビジョニングチームの削除	348
11.2.3	ダイレクトレポートを管理するチームの作成	348

このガイドについて

本ガイドでは、Designer を使用して、ユーザアプリケーションコンポーネントを作成する方法について説明します。プロジェクトビュー、ディレクトリ抽出化レイヤエディタ、およびプロジェクト要求定義エディタの使用方法について説明しています。

- ◆ 9 ページの第 1 章「ユーザアプリケーション設計ツールの概要」
- ◆ 13 ページの第 2 章「プロジェクトビューの操作」
- ◆ 39 ページの第 3 章「ディレクトリ抽象化レイヤの設定」
- ◆ 79 ページの第 4 章「プロジェクト要求定義エディタの操作」
- ◆ 97 ページの第 5 章「プロジェクト要求定義の作成」
- ◆ 107 ページの第 6 章「プロジェクト要求定義のフォームの作成」
- ◆ 163 ページの第 7 章「プロジェクト要求定義のワークフローの作成」
- ◆ 187 ページの第 8 章「ワークフローアクティビティの参照」
- ◆ 217 ページの第 9 章「統合アクティビティの操作」
- ◆ 269 ページの第 10 章「ECMA 式の使用」
- ◆ 339 ページの第 11 章「プロジェクトチームの設定」

対象読者

本ガイドは、Identity Manager 上で実行するワークフローベースのプロジェクトの作成を担当している設計者を対象にしています。

フィードバック

本マニュアルおよびこの製品に含まれているその他のマニュアルについて、皆様のご意見やご要望をお寄せください。オンラインマニュアルの各ページの下部にあるユーザコメント機能を使用するか www.novell.com/documentation/feedback.html にアクセスしてコメントを記入してください。

追加のマニュアル

Identity Manager の他の機能に関するマニュアルについては、[Identity Manager のマニュアルの Web サイト](http://www.novell.com/documentation/idm) (<http://www.novell.com/documentation/idm>) を参照してください。

マニュアルの表記規則

Novell® のマニュアルでは、手順に含まれる複数の操作および相互参照パス内の項目を分けるために、大なり記号 (>) を使用しています。

商標記号 (®, ™ など) は、Novell の商標を示します。アスタリスク (*) は、サードパーティの商標を示します。

パス名の表記に円記号 () を使用するプラットフォームとスラッシュ (/) を使用するプラットフォームがありますが、このマニュアルでは円記号を使用します。Linux* または UNIX* などのようにスラッシュを使用するプラットフォームの場合は、必要に応じて円記号をスラッシュに置き換えてください。

ユーザアプリケーション設計ツール の概要

1

この項では、ユーザアプリケーションの設計に使用できるツールの概要を説明します。主なトピックは次のとおりです。

- ◆ 9 ページのセクション 1.1 「プロビジョニングビューについて」
- ◆ 10 ページのセクション 1.2 「ディレクトリ抽象化レイヤエディタについて」
- ◆ 10 ページのセクション 1.3 「プロビジョニング要求定義エディタについて」
- ◆ 10 ページのセクション 1.4 「ECMA 式ビルダについて」
- ◆ 11 ページのセクション 1.5 「プロビジョニングチームエディタについて」
- ◆ 11 ページのセクション 1.6 「プロジェクトのドキュメント生成」

1.1 プロビジョニングビューについて

プロビジョニングビューでは、常に Designer のプロビジョニング機能にアクセスできます。プロビジョニングビューを使用して、プロビジョニングオブジェクトに関する次のアクションを実行します。

- ◆ ユーザアプリケーションコンポーネントを作成および操作できる次のエディタにアクセスする。
 - ◆ ディレクトリ抽象化レイヤエディタ。
 - ◆ プロビジョニング要求定義エディタ。
 - ◆ プロビジョニングチームエディタ。

プロビジョニングビューの項目をダブルクリックすると、その項目用のエディタが開きます。

- ◆ 次のようにオブジェクト定義を操作する。
 - ◆ 識別ボールトまたはローカルファイルシステムのオブジェクト定義をインポートおよびエクスポートする。
 - ◆ ローカルオブジェクト定義を検証する。
 - ◆ 識別ボールトにオブジェクト定義を展開する。
 - ◆ ローカルファイルシステム上のオブジェクトと識別ボールトのオブジェクトとを比較する。
- ◆ ユーザアプリケーションドライバのサポートされるロケールとデフォルトのロケールについて次のことを定義する。
 - ◆ ローカライズ用の表示ラベルおよびその他のユーザアプリケーション文字列をインポートおよびエクスポートする。
 - ◆ ローカライズ用のカスタムリソースグループ（フィールドのローカライズでのみ使用される）を定義する。

1.2 ディレクトリ抽象化レイヤエディタについて

ディレクトリ抽象化レイヤエディタでは、ディレクトリ抽象化レイヤを定義できます。ディレクトリ抽象化レイヤエディタを使用して、ユーザアプリケーションの動作を変更します。次のような操作を実行できます。

- ◆ 新しいエンティティ（識別ボールトオブジェクト）を追加する。
- ◆ エンティティの属性セットを定義する。
- ◆ リストのコンテンツを指定する。
- ◆ エンティティ間の関係をモデル化する。
- ◆ エンティティ間の自動ルックアップを定義する。
- ◆ LDAP 検索を、要求フォームおよび承認フォームから実行できるクエリとして定義する。

1.3 プロビジョニング要求定義エディタについて

プロビジョニング要求定義エディタでは、さまざまな Eclipse* ベース設計ツールを使用して、カスタムプロビジョニング要求定義を作成できます。プロビジョニング要求定義エディタを使用して、次の操作を実行します。

- ◆ プロビジョニング要求の基本特性を定義する。
- ◆ 関連付けるワークフローを設計する。
- ◆ 要求フォームおよび承認フォームを定義する。
- ◆ アクティビティおよびフローパスを設定する。

1.4 ECMA 式ビルダについて

Designer には、ワークフローデータを参照および変更するスクリプト式を作成するための ECMAScript インタープリタおよび式エディタが組み込まれています。たとえば、スクリプトを使用して次の操作を実行できます。

- ◆ フローデータ要素の下に、ワークフローで必要な新しいデータ項目を作成する。
- ◆ データに対する文字列操作、日付操作、数値演算操作、リレーションナル操作、連結操作、論理操作などの基本操作を実行する。
- ◆ より高度なデータ操作用に標準またはカスタム Java* クラスを呼び出す。
- ◆ ランタイムコントロール式を使用して次の操作を実行する。
 - ◆ フォームフィールドラベルを変更または上書きする。
 - ◆ フォームフィールドデータを初期化する。
 - ◆ 電子メールのアドレスと内容をカスタマイズする。
 - ◆ エンタイトルメントの付与 / 取り消し権およびパラメータを設定する。
 - ◆ 過去のアクティビティデータを評価し、条件アクティビティを使用して条件付きでワークフローパスをたどる。
 - ◆ 単一のログアクティビティを使用して条件付きでトリガされる複数のログメッセージを作成する。

1.5 プロビジョニングチームエディタについて

プロビジョニングチームエディタでは、ユーザアプリケーションの【要求と承認】タブでチームとして作業できる一連のユーザを定義できます。チーム定義では、このチームに関連付けられたプロビジョニング要求と承認タスクを管理できるユーザを決定します。チーム定義は、チームマネージャ、チームメンバー、およびチームオプションのリストで構成されます。さらに、チームが作業できる一連のプロビジョニング要求オブジェクトも定義できます。

1.6 プロジェクトのドキュメント生成

Designer では、作成した Designer プロジェクト用のカスタマイズされたドキュメントをすばやく生成できるドキュメントジェネレータを利用できます。独自のドキュメントスタイルを定義できますが、Designer にはデフォルトのプロビジョニングスタイルが用意されています。デフォルトのプロビジョニングスタイルには、ユーザアプリケーションのセクションが含まれています。セクションの内容を次に示します。

ディレクトリ抽象化レイヤのドキュメントには、次のセクションがあります。

- ◆ エンティティ：アクセスプロパティ、補助クラス、および LDAP クラスが含まれます。
- ◆ グローバルリスト：キーおよび表示ラベルが含まれます。
- ◆ クエリ：クエリのキー、パラメータ、および条件が含まれます。
- ◆ 関係：キー、親キー、親属性、子キー、および子属性が含まれます。
- ◆ 設定：デフォルトエンティティキー、デフォルトロケール、およびコンテナクラスが含まれます。

プロビジョニング要求定義のドキュメントは、次の内容で構成されます。

- ◆ 定義のカテゴリ、ステータス、および電子メール通知を含むテーブル。
- ◆ ワークフローの構造のイメージ。
- ◆ アクティビティごとに 1 つのセクション。このセクションには、アクティビティまたは式（アクティビティタイプでサポートされている場合）のデータマッピングを一覧表示するテーブルが含まれます。
- ◆ フォームごとに 1 つのセクション。

プロビジョニングチームのドキュメントは、次の内容で構成されます。

- ◆ \95\5c 示名
- ◆ チームメンバー
- ◆ 要求のタイプとスコープ
- ◆ マネージャの許可

プロジェクトビューオブジェクトの操作

2

この項では、プロジェクトビューオブジェクトの使用の詳細について説明します。主なトピックは次のとおりです。

- ◆ 13 ページのセクション 2.1 「プロジェクトビューオブジェクトのセットアップ」
- ◆ 18 ページのセクション 2.2 「プロジェクトビューオブジェクトにアクセスする」
- ◆ 19 ページのセクション 2.3 「プロジェクトビューオブジェクトの初期設定の設定」
- ◆ 23 ページのセクション 2.4 「プロジェクトビューオブジェクトのインポート」
- ◆ 24 ページのセクション 2.5 「プロジェクトビューオブジェクトのエクスポート」
- ◆ 25 ページのセクション 2.6 「プロジェクトビューオブジェクトの検証」
- ◆ 26 ページのセクション 2.7 「プロジェクトビューオブジェクトの展開」
- ◆ 27 ページのセクション 2.8 「プロジェクトビューオブジェクトの比較」
- ◆ 28 ページのセクション 2.9 「ロケールおよびローカリゼーションリソースグループの指定」
- ◆ 33 ページのセクション 2.10 「表示ラベルのローカライズ」

2.1 プロジェクトビューオブジェクトのセットアップ

プロジェクトビューオブジェクトは、ユーザアプリケーションドライバが含まれるプロジェクトでのみ使用できます。Identity Manager プロジェクトをセットアップし (『Designer for Identity Manager 3.5: 管理ガイド』の「★ Creating a Project ★」を参照)、プロジェクトの識別ボルトとドライバセットを設定した後、ユーザアプリケーションドライバを追加して設定します。

2.1.1 ユーザアプリケーションドライバの作成

- 1 プロジェクトビューでプロジェクトを拡張します。



- 2 [システムモデル] をダブルクリックします。

- 3 次のいずれかの方法で、新しいドライバのドライバ環境設定ページにアクセスします。

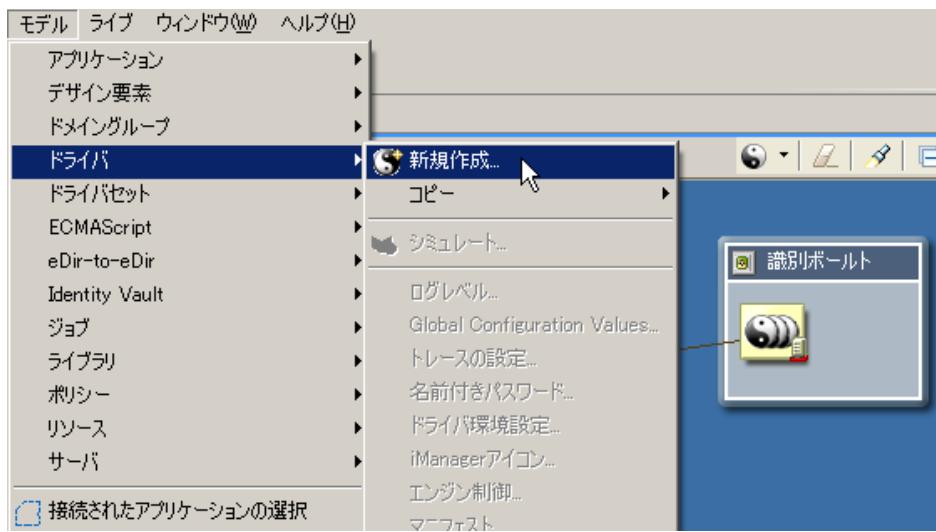
- ◆ プロジェクトのドライバセットを右クリックし、[新規] > [ドライバ] の順にクリックします。



- ◆ [パレット] の [プロビジョニング] をクリックし、[ユーザアプリケーション] アイコンをキャンバスにドラッグします。



- ◆ プロジェクトのドライバセットをクリックし、[モデル] > [ドライバ] > [新規] の順に選択します。



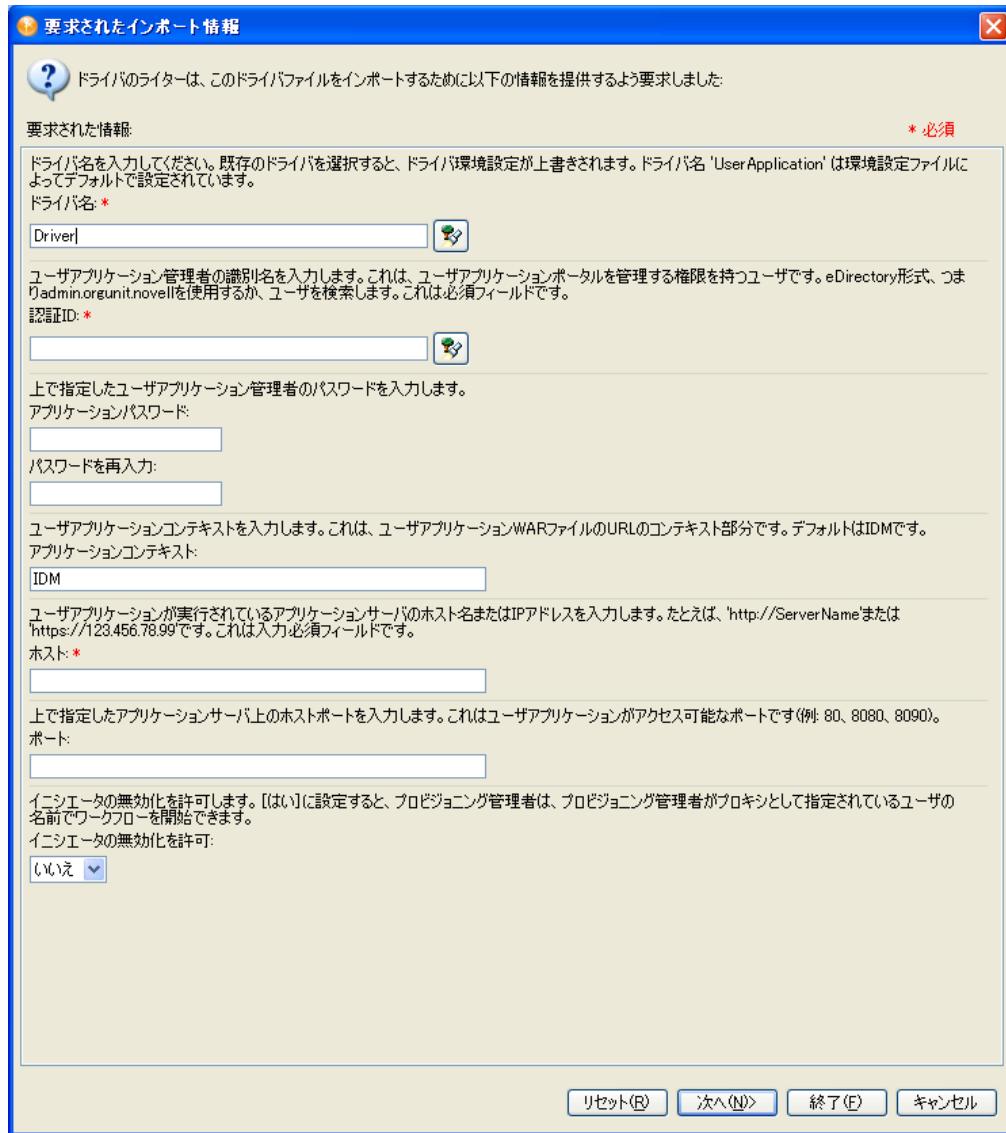
ドライバ環境設定ウィザードが表示されます。

- 4 次のいずれかのドライバ環境設定を選択します。

ドライバ設定	説明
ユーザアプリケーション 3.0.1	バージョン 3.0.1 のユーザアプリケーションドライバを作成します。
ユーザアプリケーション 3.5	バージョン 3.5 のユーザアプリケーションドライバを作成します。
ユーザアプリケーション 3.5.1	バージョン 3.5.1 のユーザアプリケーションドライバを作成します。

この手順では、すべてのバージョンの設定方法について説明しており、特定バージョンに固有のフィールドの場合はその旨を記載しています。

次のダイアログボックスが起動されます。



5 次のようにフィールドに値を入力します。

プロパティ	指定する内容
ドライバ名	既存のユーザーアプリケーションドライバ（ユーザーアプリケーションのインストール時に指定されたドライバ）の名前、または新しいユーザーアプリケーションドライバの名前。
認証 ID	ユーザーアプリケーションの管理者の DN。
アプリケーションパスワード / パスワードの再入力	ユーザーアプリケーションの管理者（前の項目）のパスワード。
アプリケーションコンテキスト	ユーザーアプリケーションのコンテキストの名前（たとえば、IDM など）。

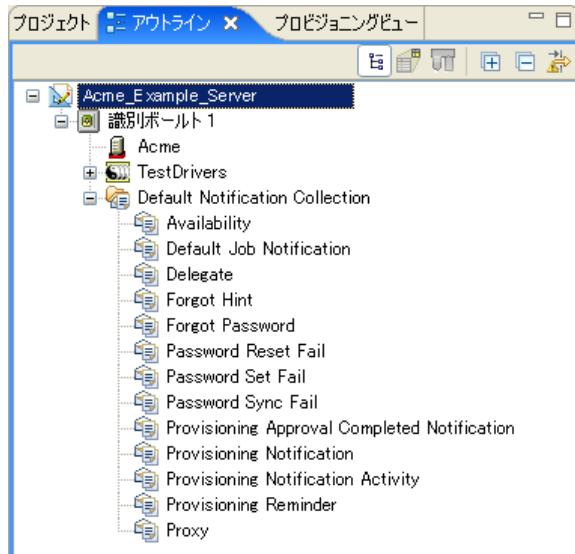
プロパティ	指定する内容
ホスト	<p>Identity Manager ユーザアプリケーションが展開されたアプリケーションサーバのホスト名または IP アドレス。この情報は次のように使用されます。</p> <ul style="list-style-type: none"> ◆ アプリケーションサーバでワークフローを起動し、ワークフローに接続してアクセスする（終了、リトラクトなど）。 ◆ キャッシュされたデータ定義を更新する。
ポート	ホスト（前の項目）のポート。
イニシエータの無効化を許可	バージョン 3.5 およびバージョン 3.5.1 のユーザーアプリケーションに適用されます。
	<p>このプロパティは、自動的に開始されるワークフローに適用されます。自動的に開始されるワークフローは、通常は管理 ID によって開始されます。このプロパティで「[はい]」を選択すると、自動開始ワークフローを別のユーザー ID で開始できます。詳細については、『<i>Identity Manager</i> ユーザアプリケーション：管理ガイド』を参照してください。</p>

6 [完了] をクリックします。

2.1.2 電子メール通知テンプレートについて

Identity Manager には、電子メール通知テンプレートの標準セットが含まれています（『*Identity Manager 3.5 ユーザアプリケーション：管理ガイド*』の「★ Working with E-Mail Templates ★」を参照）。ユーザアプリケーションドライバを作成すると、標準セットから失われている電子メールテンプレートはすべて置き換えられます。ただし、既存の電子メール通知テンプレート（以前のバージョンの *Identity Manager* のテンプレートなど）は更新されません。既存のテンプレートを新しいテンプレートに置き換えるには、次の手順に従います。

1 アウトラインビューを拡張します。



- 2 [デフォルト通知コレクション] で、置き換える電子メール通知テンプレートを削除します。
- 3 [デフォルト通知コレクション] を右クリックし、[テンプレートの更新] をクリックします。
新しいユーザアプリケーションドライバを作成せずに電子メール通知テンプレートを更新する場合はいつでも、このコマンドを使用できます。
- 4 電子メール通知テンプレートを識別ボルトに展開するには、[デフォルト通知コレクション] を右クリックし、[ライブ] > [展開] の順にクリックします。

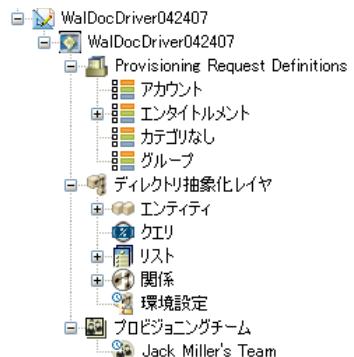
2.2 プロビジョニングビューにアクセスする

プロビジョニングビューには、次のいずれかの方法でアクセスできます。

- ◆ [ウィンドウ] > [ビューの表示] > [プロビジョニングビュー] の順にクリックします。
- ◆ [モーダル] ウィンドウで、[ユーザアプリケーション] を右クリックし、[ビューの表示] > [プロビジョニングビュー] の順にクリックします。
- ◆ アウトラインビューで、[ユーザアプリケーション] を右クリックし、[プロビジョニングビューの表示] をクリックします。
- ◆ FastView のツールバーから [プロビジョニングビュー] を選択します。

プロビジョニングビューが開くと、同じワークスペース内にあるすべてのプロビジョニングプロジェクトが表示されます。

図 2-1 プロビジョニングビューの例



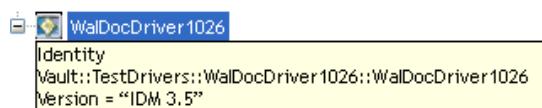
プロビジョニングビューには、オブジェクトのステータスを示すアイコンが表示されます。表 2-1 で、アイコンについて説明しています。

表 2-1 プロビジョニングビューのステータスアイコン

アイコン	説明
🕒	ローカルオブジェクトが変更されたことを示します。
⚠	ローカルオブジェクトに検証警告が含まれていることを示します。
✗	ローカルオブジェクトに検証エラーが含まれていることを示します。

ドライバ情報 :ユーザアプリケーションドライバのアイコンには、プロジェクトの識別ボールト名、ドライバセット、ドライバ名、バージョンを表示するツールチップが含まれています。図 2-2 は、バージョン 3.5 のユーザアプリケーションドライバのツールチップ例を示しています。

図 2-2 ユーザアプリケーションドライバのツールチップ



ヒント :表示されるはずのユーザアプリケーションが表示されない場合、プロジェクトが壊れている可能性があります。プロジェクトが壊れている場合は、作成し直す必要があります。

2.3 プロビジョニングビューの初期設定の設定

初期設定を設定することにより、プロビジョニングビューのいくつかの動作をカスタマイズできます。初期設定ページにアクセスするには、[ウィンドウ] > [初期設定] > [プロビジョニング] の順にクリックします。初期設定には次の項目があります。

表 2-2 プロビジョニングビューの初期設定

初期設定の カテゴリ	設定	説明
一般	ユーザアプリケーション環境設定の削除時にメッセージを表示します	これを選択した場合、モーダルからユーザアプリケーションを削除するときに、削除操作の一部としてディスク上のプロジェクトを削除するかどうかの確認を求められます。プロジェクトを削除しない場合、ユーザアプリケーションが削除されても、プロジェクトはディスク上に残されます。
	すべての [削除の確認] ダイアログで、識別ポールトからの削除をデフォルトとして設定します	プロビジョニングビューまたはディレクトリ抽象化レイヤエディタでオブジェクトを削除する場合、削除を確認するメッセージが表示されます。確認ダイアログボックスの「[展開時に識別ポールト内のオブジェクトを削除します] チェックボックスがデフォルトで選択されるかどうかが、この設定によって決まります。
	新規ユーザアプリケーションが作成またはインポートされたらプロジェクトを表示します	この設定を選択すると、チェックボックスは選択され、デフォルトで識別ポールトオブジェクトが削除されます。ローカルオブジェクトは常に削除されます。
	サポートされているロケールの表示名を検証	新しいユーザアプリケーションドライバを作成したとき、または既存のユーザアプリケーションドライバをインポートしたときにプロビジョニングビューが起動されるようにしたい場合、このオプションを選択します。
	サポートされているロケールの表示名を検証	次のことを確認するために表示名が検証されるようにしたい場合、このオプションを選択します。 <ul style="list-style-type: none"> ◆ ロケール内で表示名が一意であること。 ◆ ロケールごとに表示名が指定されている（空白ではない）こと。
	プロビジョニングビューでカテゴリを表示	ディレクトリ抽象化レイヤエディタ、プロビジョニング要求エディタ、またはプロビジョニングチームエディタを使用して定義された表示名に適用されます。
	すべてのローカライズされた電子メールテンプレートを表示	このオプションを選択すると、プロビジョニング要求定義がカテゴリ別に整理されて表示されます。[概要] パネルでカテゴリを指定します。カテゴリは、ディレクトリ抽象化レイヤで定義されるプロジェクトカテゴリリストで定義されます。
	すべてのローカライズされた電子メールテンプレートを表示	このオプションを選択すると、ローカライズされたすべての電子メールテンプレートが選択可能オプションとして [電子メール通知] タブに表示されます。電子メールテンプレートの名前には、Java 言語コードが付加されます。たとえば、cn=Provisioning Notification Activity_es, cn=Default Notification Collection,cn=security は、これがこのテンプレートのスペイン語バージョンであることを示します。
	ローカライズされたテンプレートを選択すると、ユーザのデフォルト言語に関係なく、そのテンプレートの言語が使用されます。デフォルトのテンプレート（ローカルコードがないテンプレート）を選択すると、電子メールには、ユーザのデフォルト言語（サポートされている言語の場合）が使用されます。	ローカライズされたテンプレートを選択すると、ユーザのデフォルト言語に関係なく、そのテンプレートの言語が使用されます。デフォルトのテンプレート（ローカルコードがないテンプレート）を選択すると、電子メールには、ユーザのデフォルト言語（サポートされている言語の場合）が使用されます。
識別ポールトの接続タイムアウト (ミリ秒)	Designer が識別ポールトに接続するための時間 (ミリ秒単位) です。設定時間が短かすぎると、プロビジョニング要求定義でトラスティ権を設定しようとするとき、または ECMA 式ビルダで識別ポールトにアクセスしようとするときに、エラーが発生することがあります。	Designer が識別ポールトに接続するための時間 (ミリ秒単位) です。設定時間が短かすぎると、プロビジョニング要求定義でトラスティ権を設定しようとするとき、または ECMA 式ビルダで識別ポールトにアクセスしようとするときに、エラーが発生することがあります。

初期設定の カテゴリ	設定	説明
インポート / 展開	インポート	オブジェクトが識別ボールトから削除されている場合、インポート時にローカルオブジェクトを削除します: このオプションを選択すると、識別ボールトのオブジェクトが削除されている場合、対応するローカルオブジェクトが削除されます。これにより、識別ボールトとローカルファイルを同期させることができます。ローカルファイルを残しておきたい場合は、このオプションを選択解除します。
インポート / 展開	インポート	ファイルからのインポート時にランタイム環境設定を上書きするかどうかメッセージを表示します: テスト環境からドライバをインポートして運用環境に展開したい場合、このオプションを選択します。ユーザーアプリケーションドライバのランタイムは、Designerでアクセスできないドライバに保存されているオブジェクトに依存します。それらのオブジェクトを含まないドライバを展開すると、正しく動作しません。ドライバをインポートし、変更を加えて、再び同じドライバセットに展開する場合、そのドライバにはすでにランタイム環境設定オブジェクトがあるので、このオプションを選択解除します。
インポート / 展開	展開	検証エラーがあつてもオブジェクトを展開できるようにします: 検証チェックに失敗したオブジェクトを展開したい場合、このオプションを選択します。展開時に、Designerは25ページのセクション2.6「プロビジョニングオブジェクトの検証」で説明されている検証ルールに従って、展開しようとしている定義を検証します。検証に失敗した定義が展開されないようにする場合、このオプションを選択解除します。
マイグレーション	識別ボールトスキーマの変更に関する警告を表示する	警告: 検証に失敗したオブジェクトを展開すると、ユーザーアプリケーションのランタイム時にエラーが発生することがあります。
常に(未展開の)ユーザーアプリケーションドライバを展開する		[移行] コマンドの選択時に、移行されたドライバを展開するには(新しい機能のサポートに必要な)スキーマ変更を行う必要があることを警告するダイアログボックスが表示されます。スキーマを更新していない場合、更新が完了するまでは、移行をキャンセルする必要があります。[移行] コマンドの選択時にこの警告が表示されないようにする場合は、このオプションを選択解除します。
エディタが閉じられることを示す警告を表示する		この設定は、ディレクトリに展開されていないユーザーアプリケーションドライバ(たとえば、ドライバ環境設定ファイルからインポートしたユーザーアプリケーションドライバなど)だけに関係します。展開されていないユーザーアプリケーションドライバを移行するときに、ドライバを展開するよう促す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を表示せずに、ユーザーアプリケーションドライバが常に展開されるようにしたい場合、[常に(未展開の)ユーザーアプリケーションドライバを展開する]を選択します。
		[移行] コマンドの選択時に、すべてのエディタが閉じられることを警告するダイアログボックスが表示されます。[移行] コマンドを選択するたびにこの警告が表示されないようにしたい場合、このオプションを選択します。

初期設定の カテゴリ	設定	説明
検証マスク	検証マスクテーブル	<p>このダイアログボックスを使用して、テキストコントロールなどのフォームコントロールで使用できる検証マスクを定義します。検証マスクは正規表現なので、正規表現構文に従う必要があります。</p> <p>Designer には、検証マスクのデフォルトセットが用意されています。フォームコントロールのプロパティシートに検証マスクが表示されない場合、[デフォルトに戻す] をクリックして [適用] をクリックすると、デフォルトの検証マスクを使用できます。</p>
Workflows	フォームテンプレート	<p>このダイアログボックスを使用して、既存のフォームテンプレートを削除またはプレビューします。</p> <p>図の初期設定</p> <p>[動作IDの表示] : プロビジョニング要求定義エディタの [ワークフロー] タブに、フロー内の各動作の動作 ID を表示したい場合、この設定を選択します。動作 ID は、ECMA 式ビルダで使用され、ユーザーアプリケーションのエラーログに書き込まれます。</p> <p>[フローパスタイプを表示する] : プロビジョニング要求定義の [ワークフロー] タブに、フロー内の各動作のフローパスタイプを表示したい場合、この設定を選択します。フローパスタイプは、ECMA 式ビルダで使用され、ユーザーアプリケーションのエラーログに書き込まれます。</p>

2.4 プロビジョニングオブジェクトのインポート

プロビジョニングビューのインポート機能では、次の場所からプロビジョニングオブジェクトをインポートできます。

- ◆ ドライバ環境設定ファイル
- ◆ 識別ポールト

この機能は、既存プロジェクトからの 1 つ以上の定義に基づいて新しいプロジェクトを開始する場合や、同じプロジェクトで作業する他の開発者と定義を共有したい場合に役立ちます。

注: 識別ポールトまたはドライバセットの展開コンテキストを変更したときは、インポートを実行する前に、プロジェクトを保存する必要があります。変更を保存していない場合、インポート操作では引き続き古い展開コンテキストが使用されます。

2.4.1 ドライバ環境設定ファイルからのインポート

ドライバ環境設定ファイルからオブジェクトをインポートするには:

- 1 プロビジョニングビューを開きます。
- 2 インポートしたいオブジェクトのタイプを表すルートノードを選択します。
- 3 コンテナを右クリックし、[ファイルからインポート] をクリックします。[OK] をクリックして、インポート操作（同じ名前の既存定義を上書きする可能性がある）を確認します。

- 4 インポートしたいドライバ環境設定ファイルの名前を指定し、[OK] をクリックします。

2.4.2 識別ポートからのインポート

- 1 プロビジョニングビューを開き、定義のインポート先のコンテナを選択します。
特定のプロビジョニングオブジェクトをインポートするには、プロビジョニングビューでそのノードを選択します。特定タイプのすべてのオブジェクトをインポートするには、そのタイプを表すルートノードを選択します。
- 2 コンテナを右クリックし、次のいずれかをクリックします。
 - ◆ 指定したオブジェクトとその子をインポートする場合は [オブジェクトのインポート]。
 - ◆ 選択したコンテナのすべてのオブジェクトをインポートする場合は [すべてインポート]。

識別ポート資格情報の入力を求められたら、その情報を入力し、[OK] をクリックします。

注: プロビジョニングチームの場合、[オブジェクトのインポート] ではチームオブジェクトだけがインポートされます。[チーム要求のインポート] では、関連付けられているすべてのチーム要求オブジェクトがインポートされます。

- 3 インポートしたい識別ポートコンテナまたはオブジェクトに移動し、[OK] をクリックします。
- 4 [インポート概要] ページを確認し、続行するかどうかを決めます。インポートを完了する場合は [インポート] をクリックし、インポートを中止する場合は [キャンセル] をクリックします。[インポート] をクリックすると、インポート操作が実行され、完了した操作の概要が表示されます。

2.5 プロビジョニングオブジェクトのエクスポート

プロビジョニングビューのエクスポート機能では、コンテンツを再作成せずに、プロジェクト間でプロジェクトコンポーネントを移動できます。また、プロジェクトのクローンを作成することもできます。クローンを使用して、プロビジョニングオブジェクト（およびその子）を XML ベースのドライバ環境設定ファイルにエクスポートできます。エクスポートしたファイルを [ファイルからインポート] 機能の入力元として使用すると、自分のプロビジョニングプロジェクトのコンテンツを他の開発者と容易に共有できます。

2.5.1 ドライバ環境設定ファイルへのエクスポート

- 1 プロビジョニングビューを開き、エクスポートする定義が含まれているオブジェクトを選択します。
特定のプロビジョニングオブジェクトをエクスポートするには、プロビジョニングビューでそのノードを選択します。特定タイプのすべてのオブジェクトをエクスポートするには、そのタイプを表すルートノードを選択します。
- 2 コンテナまたはオブジェクトを右クリックし、[ファイルにエクスポート] をクリックします。
- 3 生成するファイルの名前と場所を指定して、[OK] をクリックします。

ファイルのデフォルト名には、ファイルの内容が反映されます。たとえば、リストをエクスポートする場合、ファイルのデフォルト名は lists.xml になります。必要に応じて名前を変更できます。

2.6 プロビジョニングオブジェクトの検証

検証機能では、ローカルファイルシステム上のプロビジョニングオブジェクトを、展開する前に検証できます。検証では、Designer のプロジェクトチェックが実行され、プロジェクトチェックビューに結果が表示されます。

ディレクトリ抽象化レイヤオブジェクトの場合、Designer は次の検証を行います。

- ◆ XML が適切に作成されており、エンティティ、属性、リスト、関係などに必要なエレメントを定義するスキーマに準拠していることを検証します。
- ◆ すべてのエンティティを確認し、他のエンティティやグローバルリストへの参照が有効であることを確認します。
たとえば、エンティティとその属性を検証する場合、[エンティティの編集]、[DNLookup]、および [詳細エンティティ] の参照を経由する他のエンティティへの参照がすべて存在するかどうかを検証プログラムが確認します。
- ◆ 各エンティティに属性が少なくとも 1 つ定義されていることを確認します。
- ◆ 各ローカルリストおよびグローバルリストに項目が少なくとも 1 つ含まれていることを確認します。

プロビジョニング要求定義の場合、Designer は次の検証を行います。

- ◆ すべてのプロビジョニング要求定義に要求フォームと承認フォームが少なくとも 1 つずつ設定されていることを確認します。
- ◆ 条件動作に、true のアウトバウンドフローパスと false のアウトバウンドフローパスの両方が設定されていることを確認します。
- ◆ DirXML-Entitlement-DN のエンタイトルメント動作のデータ項目マッピングが有効であることを確認します。
- ◆ ユーザ動作の [最終タイムアウトアクション] プロパティに、その動作に対応するフローパスリンクがあることを確認します。たとえば、[最終タイムアウトアクション] が [却下済み] の場合は、却下済みリンクが必要です。
- ◆ ブランチ動作およびマージ動作では、ワークフローでブランチ動作とマージ動作に同じ番号が設定されていることを確認します。さらに、ブランチ動作から分岐するすべてのパスが 1 つのマージ動作にマージされていること、すべてのマージ動作にブランチ動作が 1 つずつ設定されていること、およびすべてのマージ動作に branch-activity-id 属性が設定されていることも確認します。
- ◆ スタティックリストキーに、10 進数データタイプの正しいデータが含まれていることを確認します。

プロビジョニングチームオブジェクトの場合、Designer は次の検証を行います。

- ◆ チームのマネージャとメンバーが定義されていることを検証します。
- ◆ チームのチーム要求が指定されていることを検証します。
- ◆ 要求スコープがカテゴリの場合は、チーム要求が実際にカテゴリを参照することを検証します。

ユーザアプリケーションドライバのロケールの場合、Designer は、ロケールに説明と表示名が含まれていることを確認します。初期設定を設定することにより、各ロケールの表示名の検証を無効にできます。詳細については、[19 ページのセクション 2.3 「プロジェクトビューの初期設定の設定」](#) を参照してください。

オブジェクトを検証するには、次のいずれかの操作を実行します。

- ◆ プロビジョニングビューでは、ノードを右クリックし、[検証] をクリックします。
- ◆ ディレクトリ抽象化レイヤエディタでは、エディタのツールバーの [抽象化レイヤの検証] をクリックするか、または Designer のメニューから [DAL] > [検証] の順に選択します。
- ◆ プロビジョニング要求定義エディタでは、Designer のメニューから [PRD] > [検証] の順に選択します。
- ◆ プロビジョニングチームエディタでは、Designer のメニューから [チームエディタ] > [検証] の順に選択します。

注：検証では、識別ボールトにオブジェクトが存在するかどうかの確認は行われません。

2.7 プロビジョニングオブジェクトの展開

プロビジョニングビューの展開機能では、指定したユーザアプリケーションドライバにプロビジョニングオブジェクトを展開します。設計環境でプロビジョニングオブジェクトに加えた変更を Identity Manager ユーザアプリケーションに反映させるには、変更を展開する必要があります。プロビジョニングビューでは、コンテナとその子すべて（全エンティティ、全リストなど）を展開できます。または、1つのプロビジョニングオブジェクトだけ（1つのリスト要素など）を展開することもできます。展開する項目を選択すると、その項目が識別ボールト内の同じ項目と比較されます。両項目が等しい場合、展開は中止されます。両項目が異なっている場合、両項目が表示され、ユーザは展開を続行またはキャンセルできます。

バージョン 3.5 のユーザアプリケーションドライバを展開するときに、必要な 3.5 のスキーマ変更が識別ボールトに含まれていない場合、プロビジョニングオブジェクトは展開されず、[展開結果] ダイアログボックスにエラーメッセージが表示されます。これは、3.5 のドライバが 3.0.1 の識別ボールトに展開されないようにするためにです。

注：識別ボールトまたはドライバセットの展開コンテキストを変更したときは、展開を実行する前に、プロジェクトを保存する必要があります。変更を保存していない場合、展開操作では引き続き古い展開コンテキストが使用されます。

2.7.1 プロビジョニングオブジェクトの展開

1 変更を保存します。

未保存の変更がオブジェクトに含まれている場合、未保存の定義が表示され、保存するよう促すメッセージが表示されます。保存しなくても、オブジェクトは展開されますが、未保存の変更は展開されません。変更を保存しないよう選択した場合でも、展開はキャンセルされません。

2 プロビジョニングビューを開き、展開するオブジェクトを右クリックして、[展開] または [すべて展開] をクリックします。

特定のプロビジョニングオブジェクトを展開するには、プロビジョニングビューでのノードを選択します。特定タイプのすべてのオブジェクトを展開するには、そのタイプを表すルートノードを選択します。

識別ポールト資格情報を入力するよう求められ（必要な場合）、オブジェクトが検証され、プロジェクトチェックビューにメッセージが表示されます。

検証に失敗したプロビジョニングオブジェクトが含まれているドライバを展開すると、展開の初期設定にかかわらず、ドライバは展開されますが、無効なオブジェクトは展開されません。[展開結果] ダイアログボックスにエラーが表示されます。

検証エラーが含まれているプロビジョニングオブジェクトを展開すると、定義された初期設定に基づいて展開が実行され、プロジェクトチェックビューにエラーが表示されます。

プロビジョニング要求定義の展開に関するヒント

- プロビジョニング要求定義の展開中に、動作に関連するエラーが検出された場合、動作 ID によってエラーが発生した動作が特定されます。ただし、ユーザインターフェースには、デフォルトでは動作名で表示されます。エラーメッセージで動作を特定しやすくするには、プロビジョニング要求定義を展開する前に、動作 ID の表示を有効にします。動作 ID の表示を有効にするには、ワークフローキャンバスを右クリックし、[動作ID の表示] をクリックします。
- エンタイトルメントプロビジョニング動作でプレースホルダ式の置き換えに失敗した場合は、一般エラーが発生します。この場合は、エラーを修正してから、プロビジョニング要求定義を再度展開します。
- 設計時に式は評価されないので、ランタイム時に解決する必要があるエンタイトルメントで式を使用すると、警告が表示されることがあります。これは致命的なエラーではないため、展開は行われます。
- [概要] タブで [ステータス] が [アクティブ] になっていることを確認します。
- 同じ CN を持つプロビジョニング要求定義がすでに識別ポールトに存在する場合、展開概要に相違点が表示されます。展開を実行する前に、相違点を確認できます。

2.7.2 展開された変更のテスト

Designer からユーザアプリケーションにアクセスして、展開内容を表示またはテストできます。Designer からユーザアプリケーションにアクセスするには、次の手順に従います。

- 1 [ツール] > [ユーザアプリケーションのアクセス] の順にクリックします。
- 2 表示したいユーザアプリケーションに関連付けられているプロジェクトとユーザアプリケーションドライバのコンテナを選択し、[OK] をクリックします。

プロジェクトに対して定義されたドライバ環境設定情報を使用して接続が確立されます。[ウィンドウ] > [初期設定] > [一般] > [Web Browser] で指定されたブラウザ設定が使用されます。

2.8 プロビジョニングオブジェクトの比較

プロビジョニングビューの比較機能では、ローカルファイルシステム内のプロビジョニングオブジェクトと、展開されたユーザアプリケーションドライバで実行されるプロビジョニングオブジェクトの相違点を表示できます。相違点が検出された場合、その相違点に対してどのように対処するかを指定できます。相違点は無視または調整できます。

注: 識別ボールトまたはドライバセットの展開コンテキストを変更したときは、比較を実行する前に、プロジェクトを保存する必要があります。変更を保存していない場合、比較操作では引き続き古い展開コンテキストが使用されます。

プロビジョニングオブジェクトを比較する：

- 1 プロビジョニングビューでコンテナまたはオブジェクトを右クリックし、[比較] をクリックします。
- 2 識別ボールト資格情報の入力を求められたら、その情報を入力し、[OK] をクリックします。

比較の結果が表示されます。デフォルトでは相違点だけが表示されますが、[相違点のみ表示] を選択解除すると、完全な比較情報を表示できます。

注: プロビジョニングチームの場合は、コンテナを選択して、プロビジョニング要求とプロビジョニングチームオブジェクトを比較する必要があります。特定のチームを選択した場合は、プロビジョニングチームオブジェクトだけが比較されます。

- 3 相違点がある場合、次のいずれかの操作を選択します。

状態の調整	説明
調整しない	定義を一切変更しません。
Designer の更新	識別ボールトから定義をインポートします。
eDirectory の更新	Designer から識別ボールトに定義を展開します。
親による調整	情報の提供のみを行います。親オブジェクトのいずれかがすでに調整中であるどうかを示します。通常は無効になっており、Designer または識別ボールトで親オブジェクトがすでに調整中である場合にのみ設定されます。

2.9 ロケールおよびローカリゼーションリソースグループの指定

Designer を使用して次の項目を定義できます。

- ◆ ユーザアプリケーションドライバのデフォルトのロケール。ユーザのロケールと一致するロケールが見つからない場合にコンテンツの表示に使用されるロケールです。
- ◆ ユーザアプリケーションドライバがサポートするロケール。
- ◆ 特定フィールドのローカライズ用のカスタムローカリゼーションリソースグループ。

注: ロケールおよびローカリゼーションリソースグループを指定する機能は、バージョン 3.5.1（およびそれ以上）のユーザアプリケーションドライバでのみ使用できます。

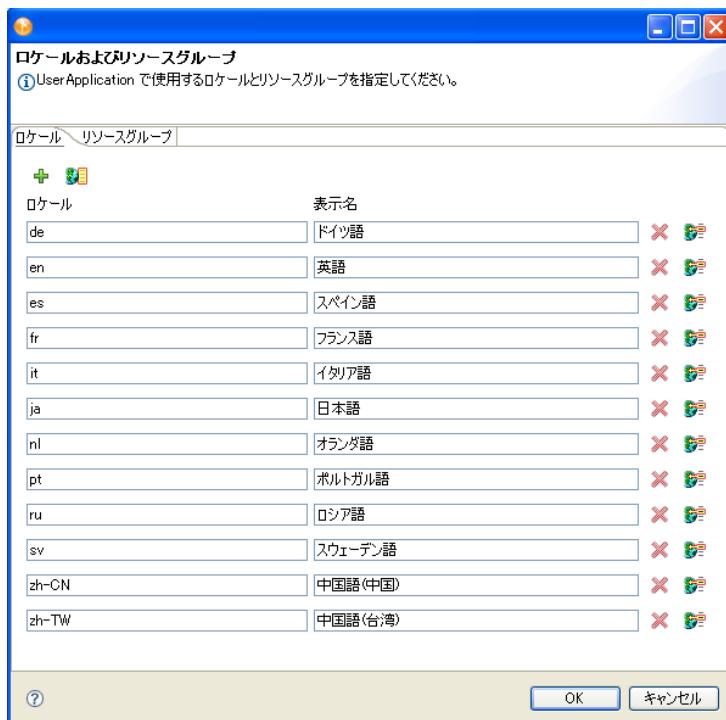
ユーザアプリケーションドライバのデフォルトロケールを指定する：

- 1 プロビジョニングビューでユーザアプリケーションドライバを右クリックし、[環境設定] > [デフォルトロケール] の順にクリックします。

- 2** ドロップダウンリストからロケールを選択し、[OK] をクリックします。目的のロケールがリストに表示されない場合は、[ロケール] ダイアログボックスを使用してそのロケールを追加する必要があります。

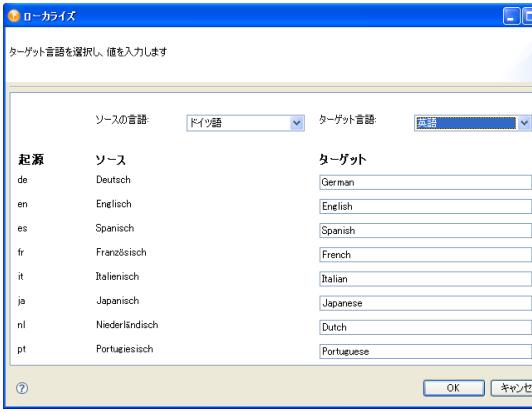
ユーザアプリケーションドライバでサポートされるロケールを定義する：

- 1** プロビジョニングビューでユーザアプリケーションドライバを右クリックし、[環境設定] > [ロケール] の順にクリックします。[サポートされるロケール] ダイアログボックスが表示されます。



- 2** 次のようにフィールドに値を入力します。

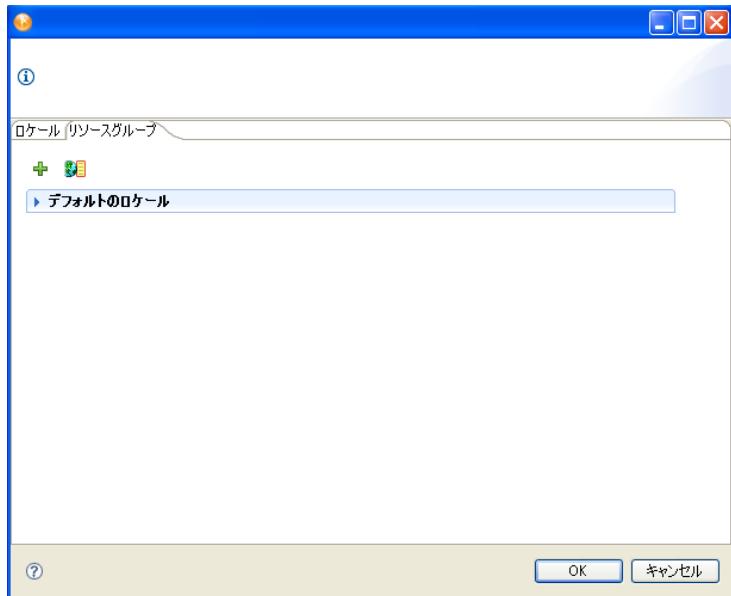
ボタン	説明
	ユーザアプリケーションドライバでサポートされるロケールのリストに新しいロケールを追加します。[追加] ボタンをクリックすると、言語と国の入力を求められます。リストから選択するか、または値を入力できます。言語は必須で、ロケールとして表示されます。国は必須ではありません。
	選択したロケールをサポート対象リストから削除します。選択したロケールのファイルは削除されません。

ボタン	説明																											
	<p>サポートされるすべてのロケールを 1 つのダイアログボックスでローカライズできます。このボタンをクリックすると、次のダイアログボックスが表示されます。</p>  <p>The dialog box is titled "ローカライズ" (Localization). It has dropdown menus for "ソースの言語" (Source Language) set to "ドイツ語" (German) and "ターゲット言語" (Target Language) set to "英語" (English). Below these are two columns of language mappings:</p> <table border="1"> <thead> <tr> <th>起原</th> <th>ソース</th> <th>ターゲット</th> </tr> </thead> <tbody> <tr> <td>de</td> <td>Deutsch</td> <td>German</td> </tr> <tr> <td>en</td> <td>Englisch</td> <td>English</td> </tr> <tr> <td>es</td> <td>Spanisch</td> <td>Spanish</td> </tr> <tr> <td>fr</td> <td>Französisch</td> <td>French</td> </tr> <tr> <td>it</td> <td>Italienisch</td> <td>Italian</td> </tr> <tr> <td>ja</td> <td>Japanisch</td> <td>Japanese</td> </tr> <tr> <td>nl</td> <td>Niederländisch</td> <td>Dutch</td> </tr> <tr> <td>pt</td> <td>Portugiesisch</td> <td>Portuguese</td> </tr> </tbody> </table> <p>At the bottom are "OK" and "キャンセル" (Cancel) buttons.</p>	起原	ソース	ターゲット	de	Deutsch	German	en	Englisch	English	es	Spanisch	Spanish	fr	Französisch	French	it	Italienisch	Italian	ja	Japanisch	Japanese	nl	Niederländisch	Dutch	pt	Portugiesisch	Portuguese
起原	ソース	ターゲット																										
de	Deutsch	German																										
en	Englisch	English																										
es	Spanisch	Spanish																										
fr	Französisch	French																										
it	Italienisch	Italian																										
ja	Japanisch	Japanese																										
nl	Niederländisch	Dutch																										
pt	Portugiesisch	Portuguese																										
	<p>サポートされるロケールの名前を翻訳するソース言語を選択できます。サポートされるロケールの名前を翻訳するターゲット言語を選択します。[ターゲット] フィールドに翻訳を入力します。</p> <p>選択したロケールの表示ラベルをローカライズします。この手順は、33 ページのセクション 2.10 「表示ラベルのローカライズ」 で説明されている手順と同じです。</p>																											

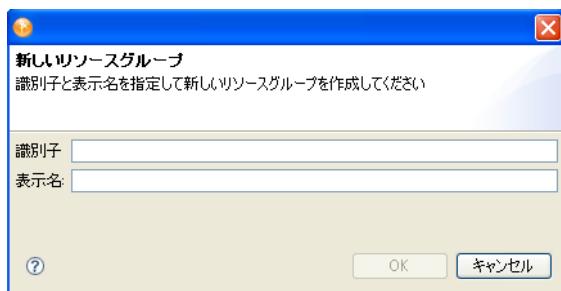
3 [OK] をクリックします。

カスタムローカリゼーションリソースグループを作成する：

- 1 プロビジョニングビューでユーザアプリケーションドライバを右クリックし、[環境設定] > [ロケール] の順にクリックします。[リソースグループ] タブに移動します。



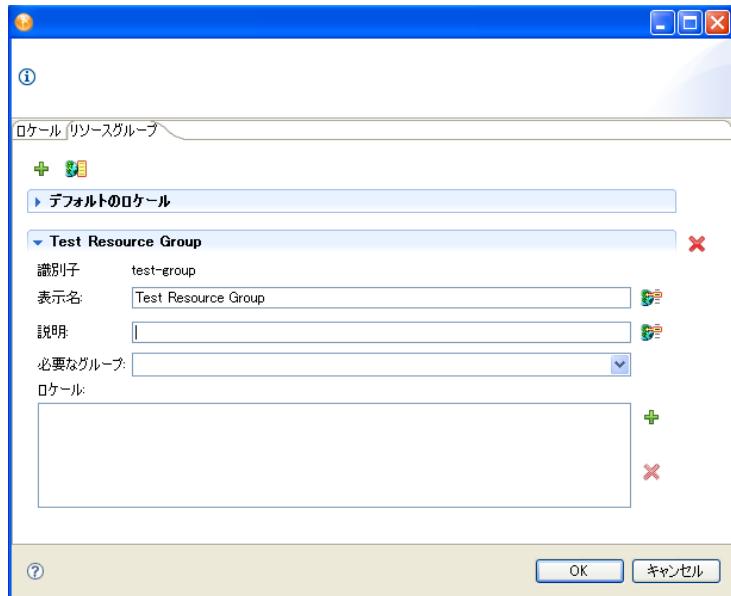
2 をクリックします。



3 次のようにフィールドに値を入力します。

フィールド	説明
Identifier	ローカリゼーションリソースグループの識別に使用される固有名です。
表示名	Designer で表示される、このグループに対応する名前です。

4 [OK] をクリックします。[リソースグループ] プロパティダイアログボックスが表示されます。



5 次のようにフィールドに値を入力します。

ボタン	説明
	新しいローカリゼーションリソースグループを追加します。
	現在のローカリゼーションリソースグループを削除します。
	ローカリゼーションリソースグループを1つのダイアログボックスでローカライズできます。
説明	このエントリの説明テキストを入力します。
必要なグループ	他のローカリゼーションリソースグループへの従属を指定します。 ローカリゼーションリソースグループは、標準以外の言語のリソースバンドルを作成する場合に使用されます。必要なグループは、『★ <i>Identity Manager 3.5.1 User Application: Localization Toolkit Guide</i> ★』に定義されています。ツールキットの詳細については、Novell の販売代理店にお問い合わせください。
ロケール	このローカリゼーションリソースグループのローカライズ先のロケールを追加または削除します。ロケールがリストにない場合は、「 29 ページの「ユーザアプリケーションドライバでサポートされるロケールを定義する:」 」を参照してください。

2.10 表示ラベルのローカライズ

Designer には、プロビジョニングオブジェクトで定義された表示ラベルを簡単にローカライズする方法が用意されています。図 2-3 のボタンが表示されている場合はいつでも、ローカライズ文字列値を指定できます。

図 2-3 [ストリングのローカライズ] ボタン



このボタンをクリックすると、図 2-4 に示されている [ローカリゼーション] ダイアログボックスが表示されます。

図 2-4 [ローカリゼーション] ダイアログボックス

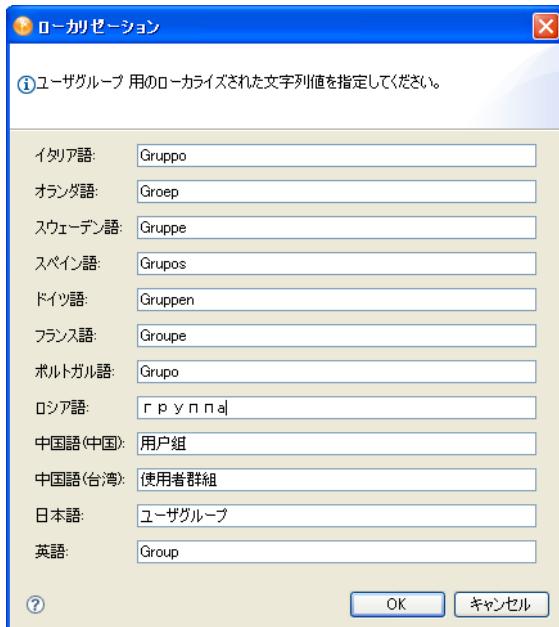


表 2-3 ローカライズ可能なオブジェクト

Designer ツール	説明
ディレクトリ抽象化レイヤエディタ	<ul style="list-style-type: none">エンティティおよび属性の表示ラベル関係名グローバルリストおよびローカルリストの項目クエリ表示ラベルおよびパラメータ表示ラベル
プロビジョニング要求定義エディタ	<ul style="list-style-type: none">ユーザーに表示される動作プロパティユーザーに表示されるフォームプロパティ
プロビジョニングチームエディタ	<ul style="list-style-type: none">プロビジョニングチームの表示名および説明

ユーザアプリケーションのデフォルト言語の表示ラベルは必ず入力してください。入力しないと、次のランタイムエラーが発生します。「リソースリゾルバ com.novell.soa.common.i18n.LocalizedMapResolver が <locale> のデフォルトのロケールのリソースを返しませんでした。デフォルトのロケールにはリソースが存在している必要があります。」

ロケール環境設定は、ドライバの AppConfig.AppDfs.locale-configuration XMLData 属性の <default-locale> 要素に保存されます。

2.10.1 サポートされている言語

[ローカリゼーション] ダイアログボックスに表示されているどの言語にでも表示ラベルをローカライズできます。サポートされている言語は、ユーザアプリケーションドライバで定義されます。サポートされるロケールのリストを変更するには、[28 ページのセクション 2.9 「ロケールおよびローカリゼーションリソースグループの指定」](#) で説明されている手順に従います。

2.10.2 ディレクトリ抽象化レイヤの表示ラベルのローカライズ

ディレクトリ抽象化レイヤエディタでは、複数の方法で抽象化レイヤ定義をローカライズできます。ローカライズ用のダイアログボックスには、次の方法でアクセスできます。

表 2-4 ローカライズ用のダイアログボックスにアクセスする方法

ローカライズするテキスト	アクション
ディレクトリ抽象化レイヤでローカライズ可能なすべての項目	[DAL] > [グローバルローカリゼーションの設定] の順にクリックします。 または エディタのツールバーで [グローバルローカリゼーションの設定] をクリックし、[ターゲット言語] を選択してから、[ターゲット] フィールドにローカライズテキストを入力します。
特定のエンティティ、関係、またはリスト	ツリービューで、ローカライズするオブジェクトを右クリックして [ローカライズ] をクリックし、[ターゲット言語] を選択してから、[ターゲット] フィールドにローカライズテキストを入力します。
単一の表示ラベル	特定のエンティティまたは属性を選択し、プロパティペインの [表示ラベル] の隣にある [表示ラベルのローカライズ] をクリックします。

2.10.3 ローカライズされたラベルのエクスポートとインポート

Designer には、ユーザアプリケーションプロジェクトのすべての表示ラベルを XML ファイルまたはプロパティファイルにエクスポートするためのウィザードが用意されています。これらのファイルをローカライズして、再びインポートできます。ドライバ全体、すべてのディレクトリ抽象化レイヤまたはすべてのプロビジョニング要求定義、または单一オブジェクトを一度にエクスポートできます。

表示ラベルをエクスポートする：

- 1 ローカリゼーションデータのエクスポートウィザードを起動するには、プロジェクトビューでコンテナノードまたはオブジェクトを右クリックします。
- 2 [ローカライズ] > [ローカリゼーションデータのエクスポート] の順にクリックします。



- 3 次のようにフィールドに値を入力します。

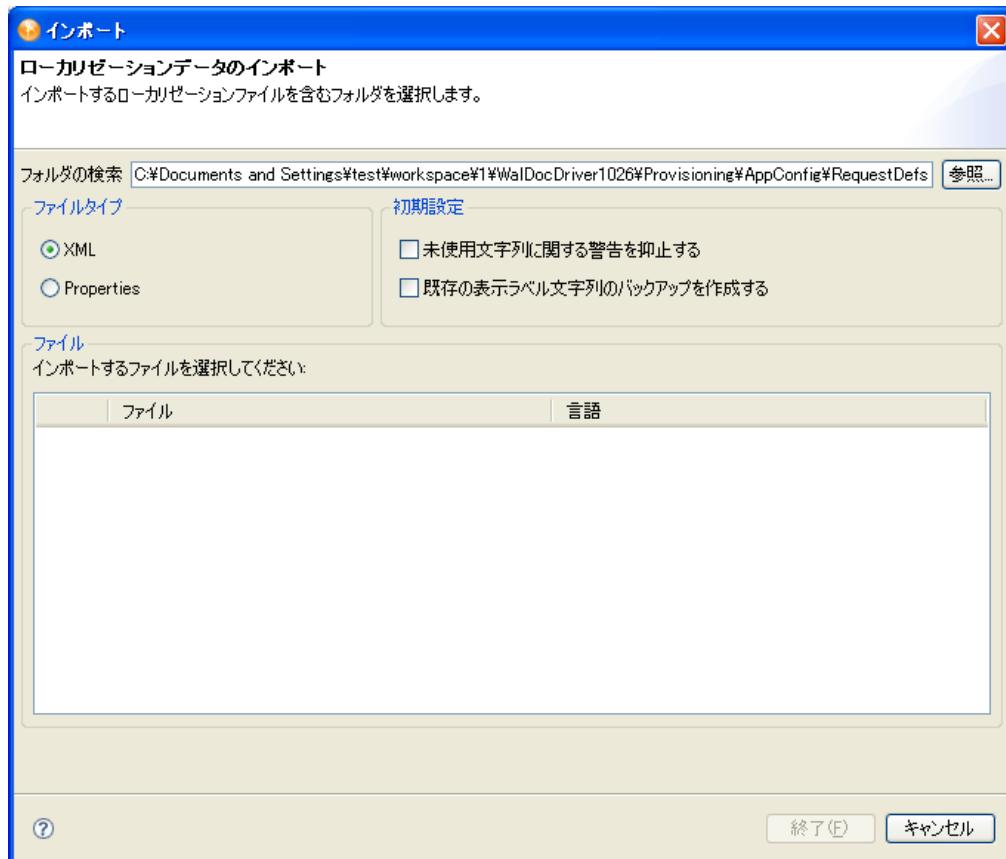
フィールド	説明
フォルダに格納する	エクスポートされたファイルを書き込むローカルフォルダの名前を指定します。
生成されたファイルのプレフィックス	生成されたファイルのプレフィックスを指定します。どのプロジェクトのファイルであるかが分かるように、名前付け方法を決めます。
ファイルタイプ	エンコードまたは形式に応じて [XML] または [プロパティ] を選択します。XML ファイルは UTF-8 でエンコードされます。プロパティファイルでは Unicode* が使用されます。

フィールド	説明
エクスポートする言語を選択する	ローカライズ先の言語を選択します。表示ラベルキーを格納するファイルがその言語で生成されます。正しいユーザアプリケーションプロジェクトにインポートできるように、ローカリゼーションをこのファイルに正しい形式で追加する必要があります。
既存ファイルを上書きする前にプロンプトする	このオプションを選択すると、ターゲットディレクトリに同じ名前のファイルがある場合、既存ファイルを上書きする前に、メッセージが表示されます。

- 4 [完了] をクリックします。エクスポート操作の結果を示すメッセージが表示されます。

ローカライズされたファイルをインポートする：

- 1 ローカリゼーションデータのインポートウィザードを起動するには、プロジェクトビューでコンテナノードまたはオブジェクトを右クリックします。
- 2 [ローカライズ] > [ローカリゼーションデータのインポート] の順にクリックします。



- 3 次のようにフィールドに値を入力します。

フィールド	説明
フォルダの検索	インポートするファイルが保存されているフォルダの場所を指定します。
ファイルタイプ	インポートするファイルが XML 形式の場合は [XML] を選択します。
	インポートするファイルがプロパティ形式の場合は [プロパティ] を選択します。
初期設定	ウィザードで警告メッセージが表示されないようにする場合は、[未使用文字列に関する警告を抑止する] を選択します。
	インポートする前に既存文字列のバックアップが作成されるようにする場合は、[既存の表示ラベル文字列のバックアップを作成する] を選択します。元に戻す必要が生じた場合に役立ちます。
ファイル	インポートするファイルを選択します。このテーブルには、前の手順で指定したフォルダの場所とファイルタイプに基づいたファイルが表示されます。空白の場合、指定したタイプのファイルはターゲットフォルダに存在しません。ウィザードは、ファイル名から言語を判別しようとします。ファイル名から判別できない場合、デフォルト言語は英語になります。
	ウィザードが推定した言語が間違っている場合は、[言語] 列を変更できます。ウィザードは、指定された言語を反映してファイル名を変更し、対応する言語に表示名をインポートします。

- 4 [完了] をクリックしてインポートを完了します。結果を示すステータスダイアログボックスが表示されます。結果には、ファイルの読み取りに関するエラーや、一致対象が見つからなかったため使用されていない表示ラベルキーに関する警告が含まれます。

ディレクトリ抽象化レイヤの設定

3

この項では、ディレクトリ抽象化レイヤの設定について説明します。主なトピックは次のとおりです。

- ◆ 39 ページのセクション 3.1 「ディレクトリ抽象化レイヤについて」
- ◆ 43 ページのセクション 3.2 「エンティティおよび属性の操作」
- ◆ 51 ページのセクション 3.3 「リストの操作」
- ◆ 54 ページのセクション 3.4 「クエリの操作」
- ◆ 58 ページのセクション 3.5 「関係の操作」
- ◆ 61 ページのセクション 3.6 「環境設定の操作」
- ◆ 62 ページのセクション 3.7 「ディレクトリ抽象化レイヤプロパティのリファレンス」

3.1 ディレクトリ抽象化レイヤについて

ディレクトリ抽象化レイヤは、ユーザアプリケーションの識別ポールトの論理ビューを定義する、XMLベースのファイルセットです。ユーザアプリケーションは、ディレクトリ抽象化レイヤ定義を使用して、次のことを決定します。

- ◆ ユーザアプリケーションで表示または変更できる識別ポールトオブジェクトおよび属性。
- ◆ ユーザアプリケーションで識別ポールトデータを表示する方法。
- ◆ ユーザアプリケーションで表示できる関係。
- ◆ ユーザアプリケーションで表示できるプロビジョニング要求カテゴリ、電子メール通知タイプ、および委任関係。

ユーザアプリケーションには、機能するために必要なエンティティ、関係、およびリストのデフォルトセットが用意されていますが、新しいディレクトリ抽象化レイヤを追加したり、既存のディレクトリ抽象化レイヤを変更したりすることで、ビジネスニーズに合わせてユーザアプリケーションをカスタマイズできます。ディレクトリ抽象化レイヤエディタ（40 ページのセクション 3.1.2 「ディレクトリ抽象化レイヤエディタについて」で説明）を使用して、ディレクトリ抽象化レイヤの内容を定義します。

3.1.1 ユーザアプリケーションのデータニーズの分析

ディレクトリ抽象化レイヤオブジェクトに変更を加える前に、ユーザアプリケーションで識別ポールトデータを表示する方法を分析します。次の事項を検討してください。

- ◆ ユーザアプリケーションで使用できるようにする識別ポールト要素。
たとえば、ユーザが検索および表示できるオブジェクトを検討し、リストアップします。このリストを、抽象化レイヤ定義の基本セットと比較し、新しいオブジェクトを追加する必要があるかどうかを判断します。
- ◆ 識別ポールトスキーマの構造。カスタム拡張子および補助クラスの有無。

- ◆ データの構造。
 - ◆ 必須のデータとオプションのデータ。
 - ◆ 適切な検証ルール。
 - ◆ オブジェクト間の関係 (DN 参照)。
 - ◆ 属性の定義方法 (たとえば、電話番号を表す属性は、自宅、オフィス、および携帯電話の電話番号など複数の値が含まれることがあります)
- ◆ データの表示対象。ユーザーアプリケーションを公開サイトとして使用できるようにするか、プライベートサイトにするか。

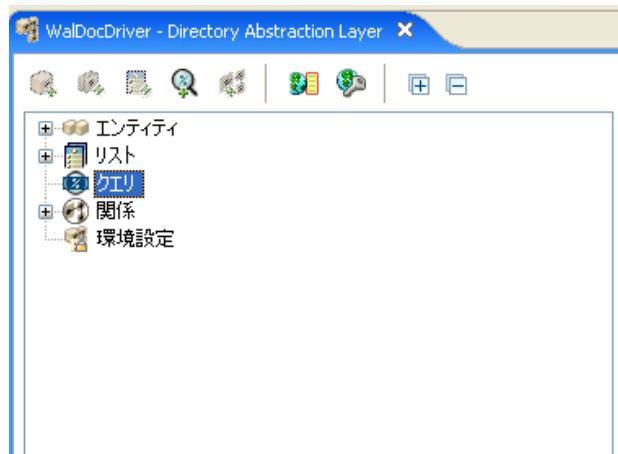
データニーズに関するこれらの情報を使用して、識別ポールトオブジェクトを抽象化レイヤエンティティにマップします。

3.1.2 ディレクトリ抽象化レイヤエディタについて

ディレクトリ抽象化レイヤエディタは、ディレクトリ抽象化レイヤファイルを定義するためのグラフィカルツールです。ユーザーアプリケーションドライバを Identity Manager プロジェクトに追加して設定ウィザードを実行すると、Designer によってディレクトリ抽象化レイヤの初期ファイルセットが作成されます。設定ウィザードを実行しなければ、初期ファイルは作成されません。作成された基本ファイルは、ディレクトリ抽象化レイヤエディタの起動時に表示されます。

ディレクトリ抽象化レイヤエディタを起動する

- 1 プロビジョニングビューを開いて、[ディレクトリ抽象化レイヤ] ノードをダブルクリックします。
[エンティティ]、[リスト]、[クエリ]、[関係]、および [環境設定] ノードを含むディレクトリ抽象化レイヤツリーが表示されます。



ノード	説明
エンティティ	<p>エンティティは、ユーザアプリケーションで使用できる識別ポールトオブジェクトを表します。エンティティには2つの種類があります。</p> <ul style="list-style-type: none"> ◆ スキーマからマップされるエンティティ：このエンティティは、ユーザアプリケーションを介してユーザに直接公開される識別ポールトオブジェクトを表します。ユーザは通常、これらのエンティティの属性を作成、検索、および変更できます。 ◆ LDAP関係を表すエンティティ：DNルックアップとも呼ばれます。これらのエンティティはインデックス検索を表し、ユーザアプリケーションで特定タイプの属性をサポートするために使用されます。DNルックアップエンティティは、LDAPオブジェクト間の関係に関する情報を提供します。DNルックアップエンティティは、次の目的で使用されます。 <ul style="list-style-type: none"> ◆ 組織図ポートレットで関係を判断する。 ◆ 検索リスト、作成、および詳細の各ポートレットで選択リストやDNコンテキストを提供する。 ◆ プロビジョニング要求定義エディタを使用して定義するワークフローの要求および承認のフローフォームで使用する。
リスト	<p>グローバルリストの内容を定義します。グローバルリストは、</p> <ul style="list-style-type: none"> ◆ 1つの属性に関連付けられています。ユーザアプリケーションで、ドロップダウンリストとして属性値が表示されます。 ◆ リソース要求カテゴリを表示するために使用されます。
クエリ	ワークフローフォームから実行できるLDAP検索条件を定義できます。
関係	スキーマベースのエンティティ間の階層関係をマップできます。ユーザアプリケーションで <i>[Identityセルフサービス]</i> タブの「組織チャート」アクションを実行するとき、およびiManagerでプロビジョニングチームを定義するときに使用されます。
環境設定	一般の設定パラメータです。

- 2 左側のペインを使用して、ディレクトリ抽象化レイヤノードをナビゲートします。左側のペインで項目を選択すると、選択した項目のプロパティが右側のペインに表示されます。

3 右側のペインを使用して、選択した項目のプロパティを定義します。プロパティの詳細については、「[62 ページのセクション 3.7 「ディレクトリ抽象化レイヤプロパティのリファレンス」](#)」を参照してください。

次の表は、[ディレクトリ抽象化レイヤ] ツールバーについて説明しています。

表 3-1 [ディレクトリ抽象化レイヤ] ツールバー

ツールバーボタン	説明
	エンティティの追加ウィザードを起動します。
	属性の追加ウィザードを起動します。
	新規リストウィザードを起動します。
	新しいクエリウィザードを起動します。
	新しい関係ウィザードを起動します。
	[グローバルアクセス修飾子の設定] ダイアログボックスを起動します。
	[グローバルローカリゼーションの設定] ダイアログボックスを起動します。
	ディレクトリ抽象化レイヤツリーを展開および縮小します。

3.1.3 ディレクトリ抽象化レイヤエディタファイルについて

作業するディレクトリ抽象化レイヤファイルは、Designer プロジェクトの Provisioning\ AppConfig\ DirectoryModel ディレクトリに保存されます。ファイル名はオブジェクトキーから派生します。

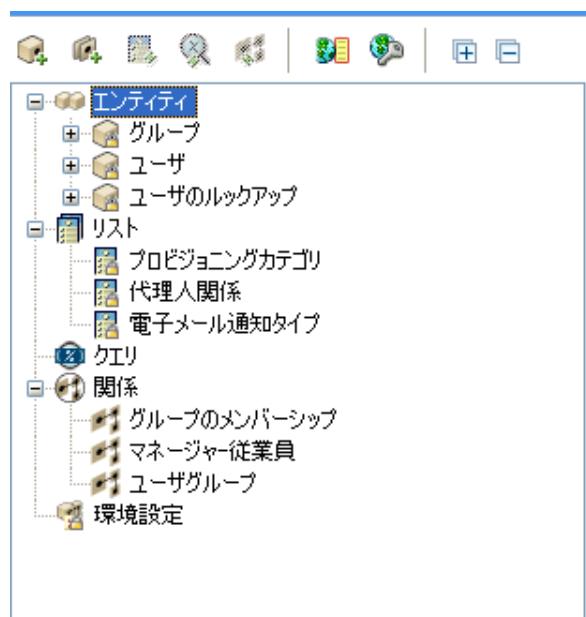
表 3-2 ローカルのディレクトリ抽象化レイヤディレクトリ

ディレクトリ名	説明
ChoiceDefs	グローバルリストを定義するファイルが含まれます。ファイルには拡張子 <code>choice</code> が付きます。
EntityDefs	エンティティおよび属性を定義するファイルが含まれます。ファイルには拡張子 <code>entity</code> が付きます。
QueryDefs	クエリを定義するファイルが含まれます。ファイルには拡張子 <code>query</code> が付きます。
RelationshipDefs	組織図ポートレット、および iManager でのプロジェクトチーム設定で使用できる関係を定義するファイルが含まれます。ファイルには拡張子 <code>relation</code> が付きます。

Designerによって、各プロジェクトのディレクトリ抽象化レイヤファイルの基本セットが作成されます。同じファイルセットが、ユーザアプリケーションのインストール時にユーザアプリケーションドライバに展開されます。

Identity Manager ユーザアプリケーションをカスタマイズするには、ディレクトリ抽象化レイヤオブジェクトを変更し、変更内容をユーザアプリケーションドライバに展開します。エンティティ、属性、リスト、および関係の中には、ユーザアプリケーションが正常に機能するために必須のものがあります。エディタでは、削除できない定義は鍵アイコン付きで表示されます。次の図を見ると、[グループ]、[ユーザ]、または [ユーザのロックアップ] エンティティは削除できないことがわかります。

図3-1 DAL ユーザアプリケーションのデフォルトのエンティティ、リスト、および関係



1つのプロジェクトに複数のユーザアプリケーションドライバを定義すると、複数の AppConfig フォルダが作成され、AppConfig、AppConfig1、AppConfig2 のようにフォルダ名が付けられます。

3.2 エンティティおよび属性の操作

識別ボルトの内容に基づいてオブジェクトとその属性を追加することにより、ユーザアプリケーションをカスタマイズできます。これを行うには、新しいエンティティと属性をディレクトリ抽象化レイヤに追加し、追加したエンティティと属性をユーザアプリケーションドライバに展開します。

デフォルトでインストールされたエンティティファイルを変更するには、「45 ページのセクション 3.2.2 「エンティティの追加」」および「49 ページのセクション 3.2.3 「属性の追加」」を参照してください。すでに展開されたプロジェクトのエンティティファイル、または別の開発者が定義したファイルセットを変更するには、まず、対象ファイルを設計環境にインポートする必要があります。ファイルのインポートについては、「23 ページのセクション 2.4 「プロジェクトのインポート」」を参照してください。

3.2.1 エンティティおよび属性について

Identity Manager ユーザアプリケーションでユーザが検索、表示、編集できるようにする識別ボールトオブジェクトはすべて、ディレクトリ抽象化レイヤのエンティティとして定義する必要があります。たとえば、ユーザアプリケーションで `inetOrgPerson` 識別ボールトオブジェクトを使用するには、そのためのエンティティ定義を作成する必要があります。論理的には次の 2 種類のエンティティがあります（ただし、作成方法は同じです）。

- ◆ **スキーマからマップされるエンティティ**：このエンティティは、識別ボールトに存在し、ユーザアプリケーションでユーザに直接公開されるオブジェクトを表します。この種類のエンティティを定義する場合、ユーザに利用してもらうすべての属性を公開します。この種類のエンティティの例には、「ユーザ」、「グループ」などがあります。異なる種類のユーザに異なる属性セットを公開する場合は、同じオブジェクトに対して複数のエンティティ定義を作成することもできます。詳細については、[44 ページの「1 つのオブジェクトに対する複数のエンティティ定義の作成」](#) を参照してください。
- ◆ **LDAP 関係を表すエンティティ**：この種類のエンティティは、*DNLookup* と呼ばれ、次の目的でユーザアプリケーションに使用されます。
 - ◆ 関連するエンティティでの DN 検索の結果をリストに入力する
 - ◆ 更新や削除が行われた場合、複数の DN 参照属性間の参照整合性を維持する

DNLookup をサポートするエンティティは、組織図ポートレットで関係を判断するために使用されます。また、検索ポートレット、作成ポートレット、および詳細ポートレットでポップアップ選択リストや DN コンテキストを表示するためにも使用されます。この種類のエンティティの例には、「ユーザのルックアップ」エンティティがあります。詳細については、[71 ページの「属性および DNLookup プロパティ」](#) を参照してください。

1 つのオブジェクトに対する複数のエンティティ定義の作成

同じ識別ボールトオブジェクトを表しながら異なる方法でデータを表示する、複数のエンティティ定義を作成できます。複数のエンティティ定義で、それぞれに異なる属性を定義できます。または、すべてに同じ属性を定義し、それぞれに異なるアクセスプロパティを指定して、属性の検索、表示、編集、または非表示の方法を制御することもできます。

注：結果セットで特定のエンティティを非表示にするフィルタを定義する方法もあります。

これで、ユーザインターフェースの異なる部分に異なるエンティティ定義を使用できるようになります。たとえば、1 つは公開サイト用、もう 1 つは社内サイト用に従業員のディレクトリを作成する場合を考えてみましょう。公開サイトでは従業員の姓名および電話番号を記載し、社内サイトでは役職、マネージャなどの追加情報も含めることにします。その方法を次に示します。

- 1 2 つのエンティティ定義を作成します（異なるキーを使用します）。

どちらのエンティティ定義も同じ識別ボールトオブジェクトを公開しますが、1 つのエンティティ定義キーは公開従業員情報、もう 1 つのエンティティ定義キーは社内従業員情報です。

- 2 それぞれのエンティティ定義で、異なる属性セットを定義します。1 つは公開従業員情報用、もう 1 つは社内従業員情報用です。

3 Identity Manager ユーザアプリケーションの [ポータルの管理] タブで、公開ページ用と社内ページ用のポートレットインスタンスをそれぞれ作成します。

ポートレットインスタンスの作成の詳細については、『*Identity Manager 3.5 ユーザアプリケーション：管理ガイド*』の「★ Portlet Reference ★」を参照してください。

3.2.2 エンティティの追加

エンティティを追加するには、次の手順で説明するようにエンティティの追加ウィザードを使用するか、またはツールバーの [エンティティの追加] ボタンをクリックします。

注：[エンティティの追加] ボタンを使用する場合、作成するエンティティのオブジェクトクラスを選択するよう促すメッセージが表示されます。必須属性は自動的にエンティティに追加されます。[属性の追加] ダイアログボックスを使用して、エンティティの定義を完了します。

エンティティの追加ウィザードを使用してエンティティを追加する

1 次のいずれかの方法でエンティティの追加ウィザードを起動します。

Designer のメニューから起動する場合

- ◆ [ファイル] > [新規] > [プロビジョニング] の順にクリックします。[ディレクトリ抽象化レイヤエンティティ] を選択し、[次へ] をクリックします。

プロビジョニングビューから起動する場合：

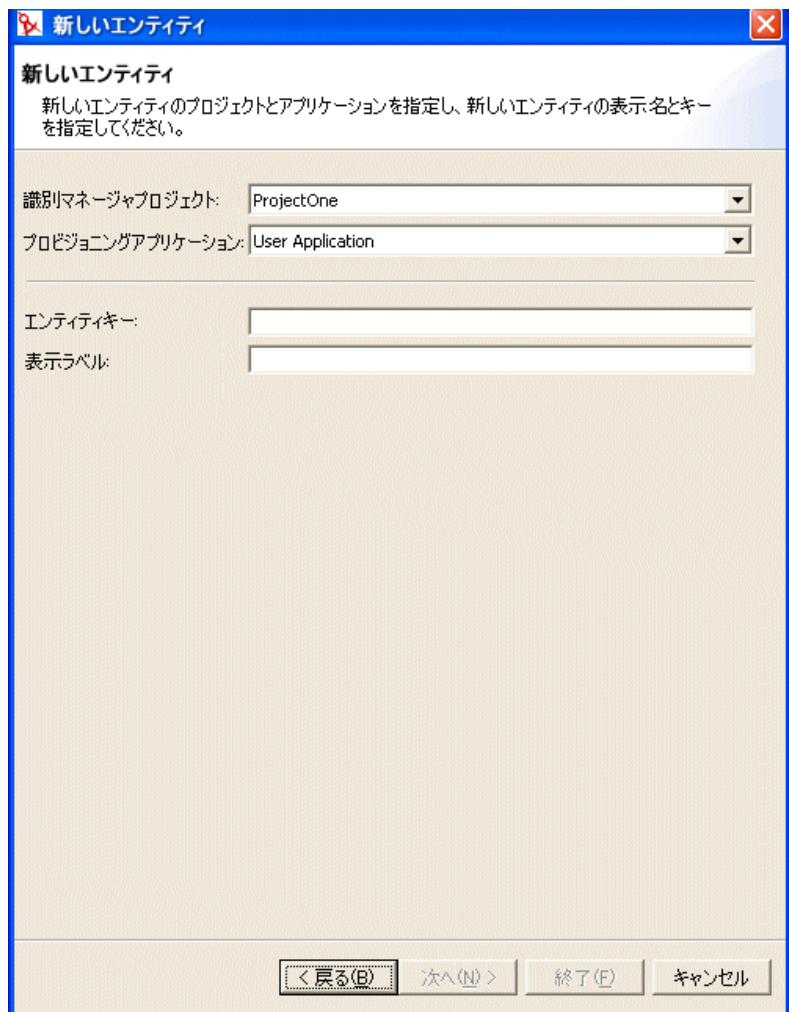
- ◆ [エンティティ] ノードを右クリックし、[新規] をクリックします。

ディレクトリ抽象化レイヤエディタから起動する場合

- ◆ [DAL] > [新規] > [エンティティ] の順にクリックします。
または
- ◆ [エンティティ] ノードを右クリックし、[新しいエンティティ - 属性ウィザード] をクリックします。

[新しいエンティティ] ダイアログボックスが表示されます。

注：[ファイル] メニューから起動した場合、次に示されているように、追加のフィールドがダイアログボックスに表示されます。



2 次のようにフィールドに値を入力します。

フィールド	説明
<i>Identity Manager</i> プロジェクトおよびプロビジョニングアプリケーション	エンティティおよび属性を追加する <i>Identity Manager</i> プロジェクトおよびプロビジョニングアプリケーション。
	注: これらのフィールドは、[ファイル] メニューからウィザードを起動したときに表示されます。
エンティティキー	エンティティの固有識別子です。
表示ラベル	ユーザアプリケーションでこのエンティティが表示されるときに使用される文字列です。このラベルはローカライズできます。詳細については、33 ページのセクション 2.10 「表示ラベルのローカライズ」を参照してください。

3 [次へ] をクリックします。[新しいエンティティ] ダイアログボックスが表示されます。



- 4 このエンティティのオブジェクトクラスを選択し、[エンティティで使用可能な属性]リストで、使用する属性をダブルクリックして追加します。必須属性は、[オブジェクトクラス]の選択時に追加されており、[エンティティで選択された属性]リストから削除できません。

ヒント: エンティティのオブジェクトクラスが [オブジェクトクラスの選択] リストに表示されないときは、[51 ページの「スキーマ要素リストの更新」](#)で説明されている手順に従って、Designer のローカルスキーマファイルを更新してください。

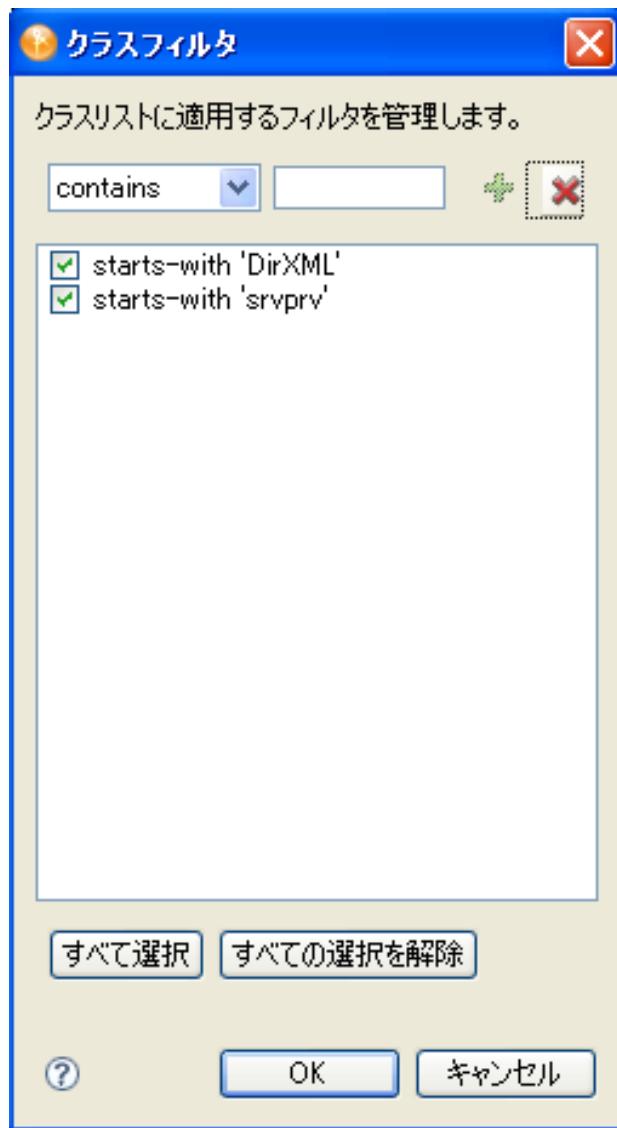
- 5 [完了] をクリックします。

編集用のプロパティページが表示されます。詳細については、[62 ページの「エンティティのプロパティ」](#)を参照してください。ユーザーアプリケーションでエンティティを使用できるようにするには、そのエンティティを展開する必要があります。

オブジェクトクラスリストのフィルタ

フィルタを追加することにより、[新しいエンティティ] ダイアログボックスに表示されるオブジェクトクラスを制限できます。フィルタを追加するには、以下の手順に従います。

- 1 [フィルタの環境設定] をクリックして [クラスフィルタ] ダイアログボックスを起動します。



デフォルトでは、クラスフィルタは適用されません。[クラスフィルタ] ダイアログボックスには、2つの事前定義フィルタ ([starts-with "DirXML"] および [starts-with "srvprv"]) が表示されています。2つとも有効にするには、[すべて選択] をクリックし、[OK] をクリックします。2つのフィルタが直ちにオブジェクトクラスリストに適用されます。これらのフィルタは、選択解除しない限り、適用されたままになります。

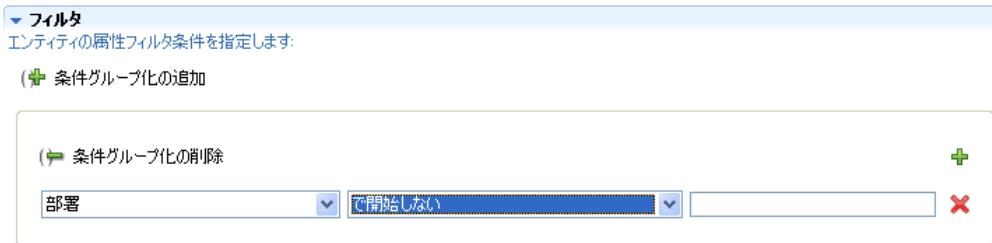
- 2 ボタンの使用方法は次のとおりです。

ボタン	説明
	文字列の比較演算子（[次を含む]、[次で始まる]、[次で終わる]など）の1つを選択し、比較対象の文字列を入力します。
	フィルタを追加します。フィルタ比較値を定義した場合に有効になります。
	選択されたフィルタを削除します。
	すべてのフィルタを使用する場合、このオプションをクリックします。定義されたすべてのフィルタが選択されます。
	定義されたすべてのフィルタを選択解除する場合は、このオプションをクリックします。このオプションを選択すると、フィルタは適用されません。

エンティティフィルタの追加

指定されたエンティティに対して返されるエントリを制限するには、エンティティフィルタを定義します。属性に基づいて、指定した別の値と比較するフィルタを定義します。たとえば、リージョン属性が「北東」のエントリだけをユーザエンティティに含めるようにするフィルタを作成できます。

- 1 [条件グループ化の追加] をクリックします。



- 2 左のドロップダウンリストで、属性を選択します。
- 3 中央のドロップダウンリストで、比較演算子を選択します。
- 4 右のフィールドで、比較する値を指定します。
- 5 複数の条件グループ化を指定する場合は、この手順を繰り返します。条件グループ化では、使用する条件をそれぞれ指定し、指定した条件を論理演算子 (and、or) で結合します。

条件は、定義されている順に評価されます。

3.2.3 属性の追加

- 1 エンティティを選択します。
- 2 次のいずれかを実行して属性を追加します。
 - エンティティを右クリックし、[属性の追加] を選択します。

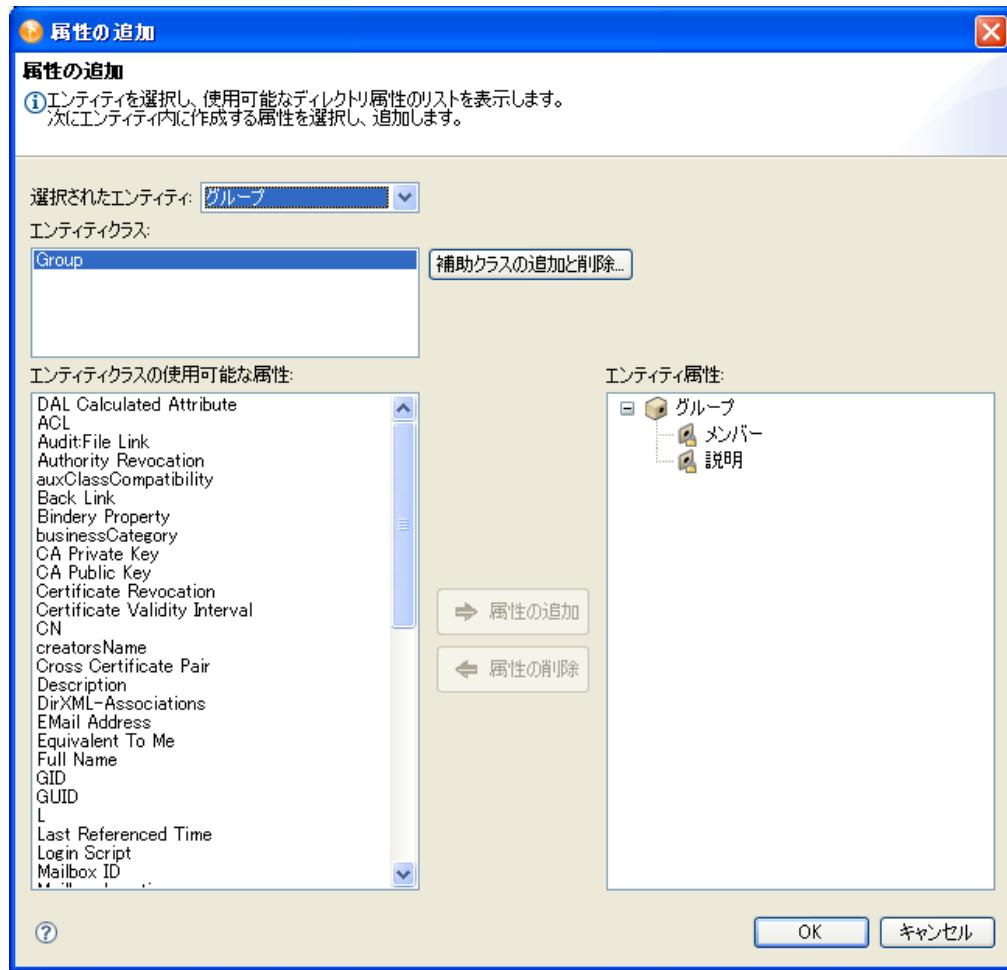
または

- ・ [属性の追加] ボタンをクリックします。

または

- ・ [DAL] > [新規] > [属性] の順にクリックします。

エンティティに追加する属性が含まれているエンティティクラスを選択するよう促すメッセージが表示されます。目的の属性が含まれるクラスを追加する必要がある場合は、補助クラスを追加（および削除）することもできます。



- 3 [エンティティクラスの使用可能な属性] リストで、目的の属性をダブルクリックして追加します。

LDAP 動作属性は、ディレクトリ抽象化レイヤエディタおよびユーザアプリケーションでサポートされています。ただし、動作属性を追加した場合、[編集]、[必須]、および [非表示] プロパティは `false` に設定され、無効になるので、これらのプロパティ値は変更できません。

ヒント: 追加したい属性が [エンティティクラスの使用可能な属性] リストに表示されない場合は、[51 ページの「スキーマ要素リストの更新」](#) の手順に従って、Designer のローカルスキーマファイルを更新してください。

- 4 [OK] をクリックします。編集用のプロパティページが表示されます。

詳細については、「[66 ページの「属性のプロパティ」](#)」を参照してください。ユーザーアプリケーションで属性を使用できるようにするには、その属性を展開する必要があります。

DAL 計算される属性の追加

式から派生する属性を作成できます。たとえば、2つ以上の属性を連結して、1つの計算値を作成できます。式は ECMAScript と互換性があり、ECMA 262 言語仕様に準拠します。

制限: この属性タイプは、識別ポールト内の特定の属性にマップされないため、更新、削除、複数値設定、必須化、検索は行えません。

計算される属性を作成する

- 1 [49 ページのセクション 3.2.3 「属性の追加」](#) の手順に従って属性を追加し、[エンティティクラスの使用可能な属性] リストで、[DAL 計算された属性] を選択していることを確認します。

次の制限付きで属性が追加されます。

表 3-3 計算される属性のプロパティ

プロパティ名	説明
式	ECMA 式ビルダを起動するには、[ECMAScript 式のビルド] をクリックします。ECMA 式ビルダの使用方法の詳細については、「 269 ページの第 10 章「ECMA 式の使用」 」を参照してください。

3.2.4 スキーマ要素リストの更新

- 1 Identity Manager プロジェクトが開いた状態で、識別ポールトを右クリックし、[ライズ] > [スキーマのインポート] の順にクリックします。
- 2 [eDirectory からインポート] を選択し、eDirectory ホストの仕様を入力します。
- 3 [次へ] をクリックします。
- 4 インポートするクラスおよび属性を選択し、[完了] をクリックします。

3.3 リストの操作

リストノードでは、グローバルリストの内容を定義できます。その後、属性の制御タイプをグローバルリストとして定義できます。編集する属性をユーザーアプリケーションで表示するときに、ドロップダウンリストにグローバルリストの内容が表示され、ユーザはそこから選択できます。デフォルトで、ディレクトリ抽象化レイヤには、表 3-4 に示されたグローバルリストが含まれています。

表3-4 ディレクトリ抽象化レイヤのデフォルトのグローバルリスト

リスト名	説明
代理人関係	関係による代理人割り当てを作成するときに選択できる関係を定義します。このリストの内容は、ドロップダウンリストに表示されます。値にできるのは、ユーザエンティティのDN属性だけです。
電子メール通知タイプ	ユーザがリソース要求の代理/委任処理に関わっているときに受信する電子メール通知のタイプを表します。タイプはロックされています。
警告: タイプの値を編集しないでください。	
この値は、ユーザエンティティの優先通知属性で使用されます。	
プロビジョニングカテゴリ	プロビジョニング対象リソース(エンタイトルメント)およびプロビジョニング要求を整理する一連のカテゴリを定義します。このリストのカテゴリは次のように <code>\95\5c</code> 示されます。
<ul style="list-style-type: none">◆ Designer: プロビジョニング要求定義エディタプラグイン◆ iManager: プロビジョニング要求設定プラグイン◆ ユーザアプリケーション: [要求および承認]タブ	

注: これらのリストは削除できず、リストのキー値を変更することもできません。電子メール通知タイプ以外では、項目の追加と削除、および既存の値とラベルの変更を実行できます。

新しいグローバルリストを作成する

- 1 次のいずれかの方法で【新規リスト】ウィザードを起動します。

Designer のメニューから起動する場合

- ◆ 【ファイル】>【新規】>【プロビジョニング】の順にクリックし、【ディレクトリ抽象化レイヤリスト】を選択して、【次へ】をクリックします。
【ファイル】メニューから起動すると、他の方法で起動した場合には表示されないフィールドがダイアログボックスに表示されます。
- ◆ 【DAL】>【新規】>【リスト】の順にクリックします。

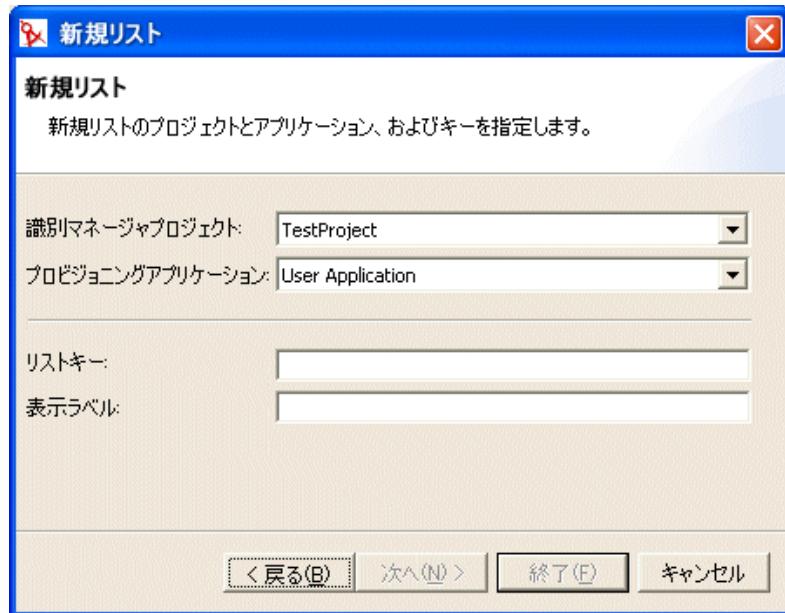
プロビジョニングビューから起動する場合

- ◆ 【リスト】ノードを右クリックし、【新規】をクリックします。

ディレクトリ抽出化層エディタから起動する場合

- ◆ 【新規リスト】をクリックします。
- ◆ 【リスト】ノードを右クリックし、【リストの追加】を選択します。

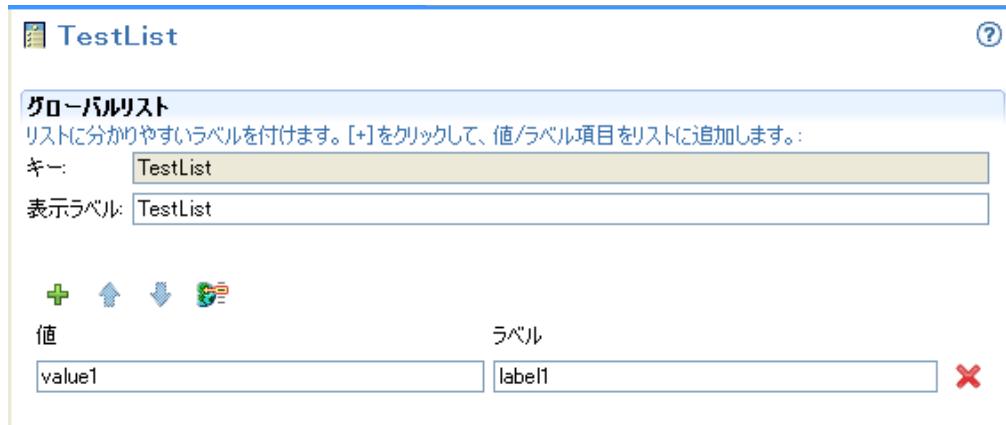
【新規リスト】ダイアログボックスが表示されます。



2 次のようにフィールドに値を入力します。

フィールド	説明
<i>Identity Manager</i> プロジェクトおよびプロビジョニングアプリケーション	リストを追加する <i>Identity Manager</i> プロジェクトおよびプロビジョニングアプリケーションを選択します。
	注: これらのフィールドは、[ファイル] メニューからウィザードを起動したときに表示されます。
リストキー	リストの固有の識別子です。
表示ラベル	ユーザアプリケーションにリストが表示されるときに使用される文字列です。このラベルはローカライズできます。詳細については、 33 ページのセクション 2.10 「表示ラベルのローカライズ」 を参照してください。

3 [完了] をクリックします。編集用の、グローバルリストのプロパティページが表示されます。



4 次のようにフィールドに値を入力します。

フィールド	説明
表示ラベル	リストの名前です。この名前が Designer に表示されます。
ラベル	ユーザアプリケーションに表示するリスト項目のテキストです。
値	識別ポルトに保存されるリスト項目の値です。使用できる文字は、英字、数字、およびアンダースコア (_) です。

次の表では、ウィザードのボタンについて説明しています。

ボタン	説明
	新しい値を追加します。
	リストの行を上方向または下方向に移動します。この順序が、ユーザアプリケーションでラベルが表示される順序になります。
	[ローカリゼーション] ダイアログボックスを表示します。このダイアログボックスの使用の詳細については、「 33 ページのセクション 2.10 「表示ラベルのローカライズ」 」を参照してください。
	行を削除します。

5 プロジェクトを保存します。

6 プロジェクトを展開して、ユーザアプリケーションで使用できるようにします。

3.4 クエリの操作

[クエリ] ノードでは、よく使用される LDAP 検索を定義できます。定義した検索は、DNQuery コントロールを使用するか、または globalQuery() メソッドを呼び出すことによって、要求フォームまたは承認フォームから実行できます。クエリを定義するには、ディレクトリ抽象化レイヤエンティティ、検索ルート、取得する行数、およびソースエン

ティティの取得条件を指定します。条件（「LastName に s が含まれる場合」など）はハーデコード化できます。または、要求フォームまたは承認フォームでユーザが入力するパラメータを 1 つ以上指定することもできます。

クエリを作成する

- 1 次のいずれかの方法で、新しいクエリ ウィザードを起動します。

Designer のメニューから起動する場合

- ◆ [ファイル] > [新規] > [プロビジョニング] の順にクリックします。[ディレクトリ抽象化レイヤクエリ] を選択し、[次へ] をクリックします。
- ◆ [DAL] > [新規] > [クエリ] の順にクリックします。

プロビジョニングビューから起動する場合

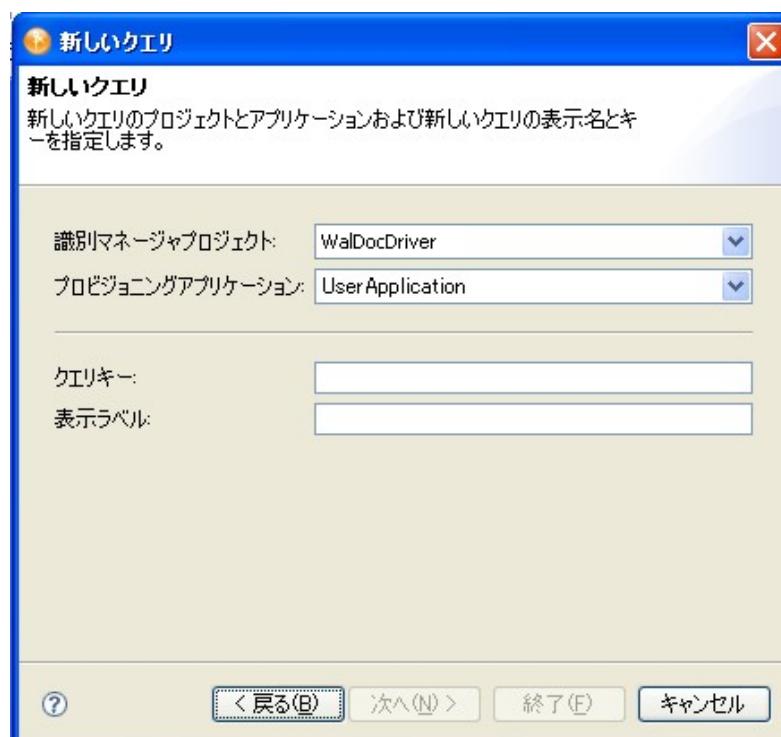
- ◆ [クエリ] を右クリックし、[追加] をクリックします。

ディレクトリ抽出化層エディタから起動する場合

- ◆ [クエリの追加] ボタンをクリックします。
- ◆ [クエリ] を右クリックし、[クエリの追加] をクリックします。

[新しいクエリ] ダイアログボックスが表示されます。

注：[ファイル] メニューから起動すると、他の方法で起動した場合には表示されないフィールドがダイアログボックスに表示されます。



- 2 次のようにフィールドに値を入力します。

フィールド	操作
Identity Manager プロジェクトおよびプロビジョニングアプリケーション	適切な Identity Manager プロジェクトおよびプロビジョニングアプリケーションを選択します。
	注: このフィールドは、[ファイル] メニューからクエリを作成したときに表示されます。
クエリキー	クエリキーの固有値を入力します。この値は、式ビルダでクエリを識別するために使用されます。
表示ラベル	ディレクトリ抽象化レイヤエディタおよびプロビジョニングビューに表示する文字列を入力します。この値は、式ビルダには表示されません。

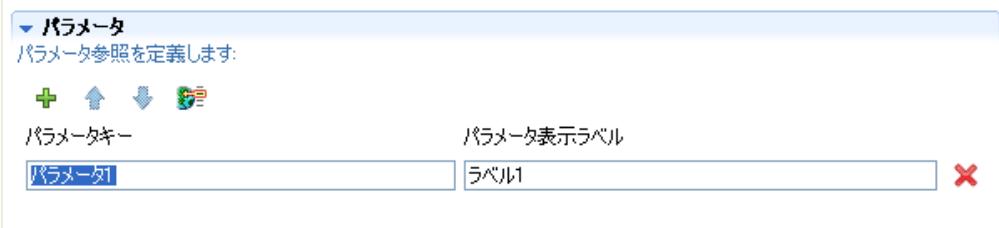
3 [完了] をクリックします。

クエリが作成され、編集用のプロパティページが開きます。

4 [クエリエンティティ] を選択します。使用したいエンティティが表示されない場合は、[エンティティ] ノードでそのエンティティが定義されているか確認してください。

5 [パラメータ] セクションで、クエリのパラメータを 1 つ以上定義します。パラメータを追加する

5a [行の追加] をクリックします。

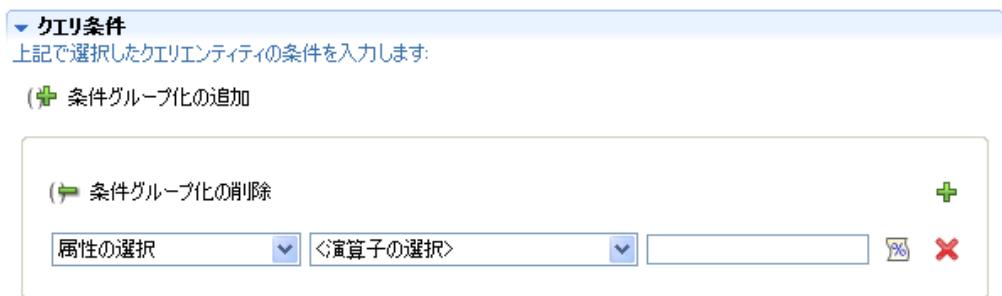


5b パラメータの固有キーと表示ラベルを指定します。フォームで globalQuery() メソッドを呼び出すときに、このキーを渡します。globalQuery() の詳細については、「[291 ページの「globalQuery \(fieldname, key, param\)」](#)」を参照してください。

5c 別のパラメータを追加する場合は、これらの手順を繰り返します。

6 クエリをさらに絞り込む場合は、[クエリ条件] を追加します。

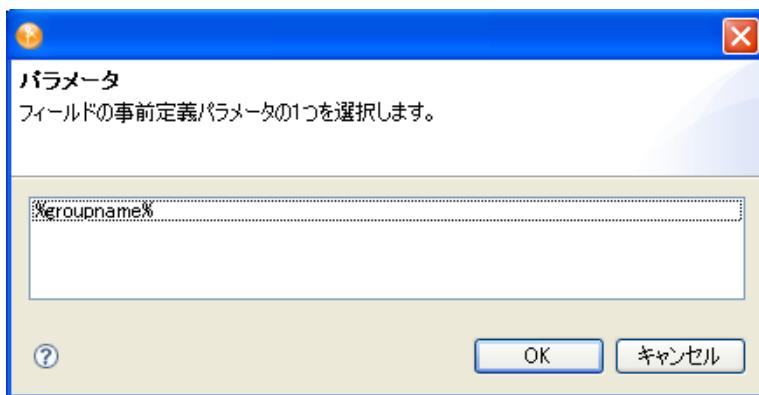
6a [条件グループ化の追加] をクリックします（[条件グループ化の追加] は、[クエリエンティティ] が選択されていないと有効になりません）。



6b 左のドロップダウンリストで、属性を選択します。このドロップダウンリストの属性は、[クエリエンティティ] で選択されたエンティティの属性です。

6c 中央のドロップダウンリストで、選択した属性に対して実行する比較演算子を選択します。

6d 右のテキストボックスで、選択した属性と比較する値を入力します。[事前定義パラメータ] をクリックして [事前定義パラメータ] ダイアログボックスを起動すると、変数名を選択できます。



2つ以上の属性または条件でフィルタする必要があるクエリで、条件を評価する順序を制御したい場合、複数の条件または条件グループを定義できます。条件グ

ループ化では、使用する条件をそれぞれ指定し、指定した条件を論理演算子 (and、or) で結合します。

- 7 複数の条件グループ化を指定するには、[条件グループ化の追加] をクリックし、各ドロップダウンリストボックスから選択します。
- 8 選択したエンティティに対してすでに定義されている内容よりもさらに検索を絞り込みたい場合は、クエリの LDAP 検索プロパティを定義します。エンティティの検索ルートとは異なり、クエリの検索ルートでは事前定義パラメータの使用はサポートされません。詳細については、[75 ページのセクション 3.7.3 「クエリのプロパティ」](#) を参照してください。
- 9 [保存] をクリックします。
- 10 クエリを展開し、ユーザアプリケーションで使用できるようにします。

3.5 関係の操作

[関係] ノードでは、ディレクトリ抽象化レイヤで定義されたエンティティ間の関係を定義できます。定義した関係は、ユーザアプリケーションで組織図を使用するとき、および iManager でチーム内のチームメンバーを定義する場合に使用されます。

定義する関係には、類似したエンティティ間（ユーザとユーザなど）の関係と、類似していないエンティティ間（ユーザとデバイスなど）の関係があります。関係をさらに絞り込むための条件を定義できます。たとえば、マネージャと従業員の関係をすべて表示し、次に特定の地域の従業員のみに絞り込んで表示したり、東部地域に配置されている特定の部長の部下全員を表示したりするような条件を作成できます。

ユーザアプリケーションでは、デフォルトで次の関係が定義されています。

- ◆ [グループのメンバーシップ] (組織図のみ)
- ◆ [マネージャ - 従業員] (組織図およびチーム管理)
- ◆ [ユーザグループ] (組織図のみ)

ソースエンティティとターゲットエンティティがどちらも InetOrgPerson オブジェクトに関連付けられている場合、関係はチーム管理でのみ使用できます。

関係を正常に展開するには、関係のコンポーネントすべて（エンティティと属性）がすでに展開されている必要があります。

- 1 次のいずれかの方法で新しい関係を作成します。

Designer のメニューから始める場合

- ◆ [ファイル] > [新規] > [プロビジョニング] の順にクリックします。[ディレクトリ抽象化レイヤ関係] を選択し、[次へ] をクリックします。
- ◆ [DAL] > [新規] > [関係] の順にクリックします。

プロビジョニングビューから始める場合

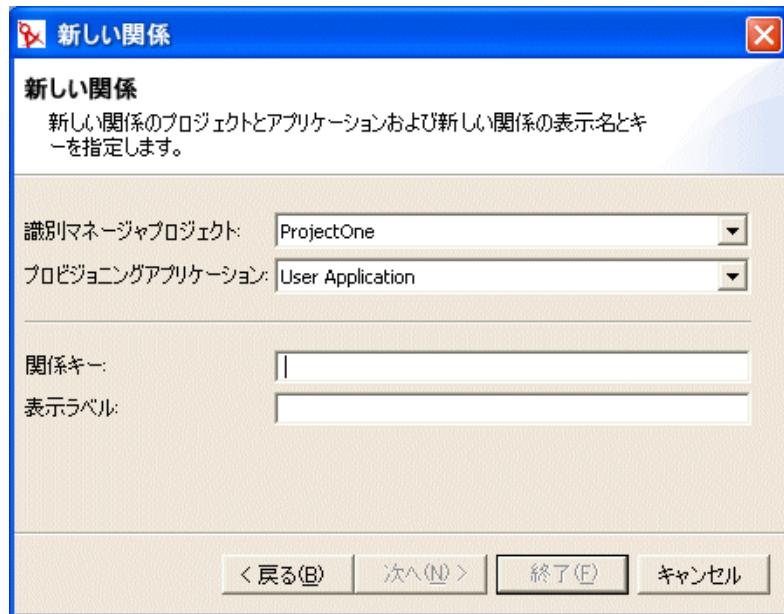
- ◆ [関係] を右クリックし、[追加] をクリックします。

ディレクトリ抽象化レイヤエディタから始める場合

- ◆ [関係の追加] ボタンをクリックします。
- ◆ [関係] を右クリックし、[関係の追加] をクリックします。

[新しい関係] ダイアログボックスが表示されます。

注: [ファイル] メニューから起動すると、他の方法で起動した場合には表示されないフィールドがダイアログボックスに表示されます。

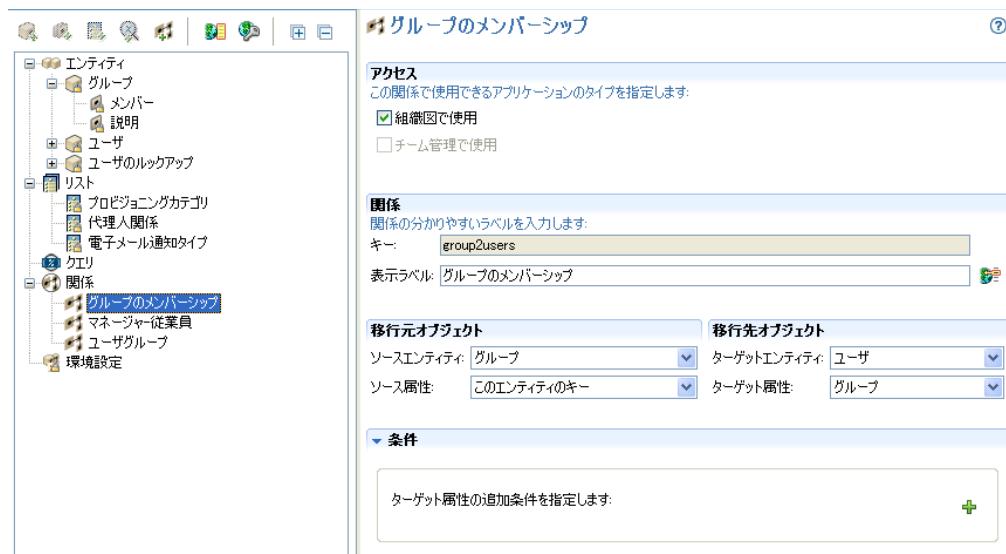


2 次のようにフィールドに値を入力します。

フィールド	操作
Identity Manager プロジェクトおよびプロビジョニングアプリケーション	適切な Identity Manager プロジェクトおよびプロビジョニングアプリケーションを選択します。 注: このフィールドは、[ファイル] メニューから関係を作成したときに表示されます。
関係キー	関係キーの固有の値を入力します。
表示ラベル	ユーザアプリケーションに関係を表示するときに使用する文字列を入力します。

3 [完了] をクリックします。

関係が作成され、編集用のプロパティページが開きます。



プロパティの定義については、「[76 ページのセクション 3.7.4 「関係のプロパティ」](#)」を参照してください。

関係を削除する

- 削除したい関係を右クリックし、[削除] をクリックします。

関係条件を追加する

- [行の追加] をクリックします。
- 左のドロップダウンリストで、属性を選択します。このドロップダウンリストの属性は、[ターゲットオブジェクト] で選択されたエンティティの属性です。
- 中央のドロップダウンリストで、演算子を選択します。
- 右のテキストボックスで比較値を指定して、条件を完成させます。例：

マネージャー従業員(region)

アクセス
この関係で使用できるアプリケーションのタイプを指定します:

組織図で使用
チーム管理で使用

階層式関係を有効にする
0 階層の最高レベル

関係
関係の分かりやすいラベルを入力します:
キー: user2users
表示ラベル: マネージャー従業員(region)

移行元オブジェクト **移行先オブジェクト**

ソースエンティティ: ユーザー ターゲットエンティティ: ユーザー
ソース属性: このエンティティのキー ターゲット属性: マネージャー

条件

ターゲット属性の追加条件を指定します:

部署 より小さいか等しい Latin

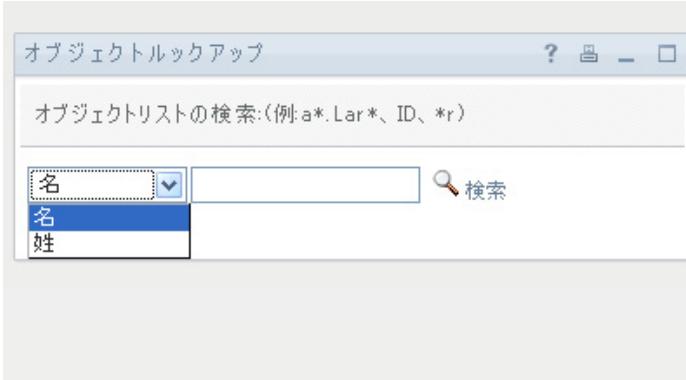
2つ以上の属性または条件でフィルタする条件を作成し、論理演算子 (and、or) を使用してそれらの属性を結合できます。条件は、定義されている順に評価されます。

3.6 環境設定の操作

[環境設定] ノードでは、ユーザーアプリケーションの一般的な環境設定プロパティを設定できます。

表3-5 環境設定

フィールド	説明
デフォルト「マイプロファイル」エンティティ	ユーザが、ユーザインターフェースの「マイプロファイル」をクリックしたときに表示されるエンティティを定義します。
デフォルト LDAP 名前付け属性	このフィールドは、オブジェクトクラスがユーザー（または LDAP inetOrgPerson）のエンティティだけを表示するよう制限されています。
デフォルト LDAP 名前付け属性	エンティティの「 ネーミング属性の作成 」が定義されていない場合のデフォルトの LDAP 名前付け属性を定義します。

フィールド	説明
デフォルトチーム管理属性	チームメンバーのルックアップに使用される属性です。ユーザアプリケーションの [マイチームの作業] ページおよび [マイチームの設定] ページで、ユーザは検索アイコンをクリックしてチームメンバーを検索できます。例：
	<p>チームメンバー: <input type="text"/> </p> <p>ここで指定した属性が検索で表示されます。次に例を示します。</p>  <p>これらの設定は、チームマネージャが実行するルックアップのみに影響します。ユーザアプリケーション管理者には、[名] および [姓] のみが表示されます。</p> <p>コンテナクラス</p> <p>[ユーザまたはグループの作成] アクションに、コンテナクラスの選択リストの内容を提供します。ユーザは選択リストから、新しく作成されたオブジェクトを保存するコンテナを選択します。</p>

3.7 ディレクトリ抽象化レイヤプロパティのリフレンス

この項では、次の抽象化レイヤノードのプロパティの定義について説明します。

- ◆ [62 ページのセクション 3.7.1 「エンティティのプロパティ」](#)
- ◆ [66 ページのセクション 3.7.2 「属性のプロパティ」](#)
- ◆ [75 ページのセクション 3.7.3 「クエリのプロパティ」](#)
- ◆ [76 ページのセクション 3.7.4 「関係のプロパティ」](#)

3.7.1 エンティティのプロパティ

エンティティには次の種類のプロパティを設定できます。

- ◆ [63 ページの「エンティティのアクセスプロパティ」](#)
- ◆ [63 ページの「エンティティの一般プロパティ」](#)
- ◆ [63 ページの「エンティティの補助プロパティ」](#)
- ◆ [64 ページの「エンティティの検索プロパティ」](#)

- ◆ 65 ページの 「エンティティの作成プロパティ」
- ◆ 65 ページの 「エンティティのパスワード管理プロパティ」
- ◆ 66 ページの 「事前定義パラメータの使用」

エンティティのアクセスプロパティ

アクセスプロパティは、ユーザアプリケーションがエンティティと対話する方法を制御します。

注: アクセスプロパティには、[DAL] > [グローバルアクセスの設定] の順にクリックしてアクセスすることもできます。

表 3-6 エンティティのアクセスプロパティ

プロパティ名	説明
作成	選択した場合、ユーザアプリケーションでこのオブジェクトを作成できます。
編集	選択しない場合、基になる ACL に関係なく、ユーザアプリケーションではこのオブジェクトを変更できません。 選択した場合、このオブジェクトは場合によって編集できます。識別ポートの ACL を使用して編集可能かどうかを決定します。
表示	選択した場合、ユーザアプリケーションでこのオブジェクトを表示できます。
削除	選択した場合、ユーザアプリケーションでこのオブジェクトを削除できます。

エンティティの一般プロパティ

表 3-7 エンティティの一般プロパティ

プロパティ名	説明
キー	エンティティの固有識別子です。このオブジェクトをユーザアプリケーションで参照する方法を定義します。エンティティの作成時に定義され、エンティティの作成後は変更できません。
表示ラベル	ユーザインターフェースでオブジェクトを \'95\5c 示す方法を定義します。
クラス名	eDirectory オブジェクトクラスの名前です。
LDAP 名	LDAP オブジェクトクラスの名前です。
検索に含める	選択した場合、ユーザアプリケーションでこのエンティティを検索できます。識別ポートレット（エンティティ検索リストやエンティティ組織図など）の クエリで使用されるエンティティを選択（True に設定）する必要があります。

エンティティの補助プロパティ

表 3-8 エンティティの補助プロパティ

プロパティ名	説明
補助クラス	<p>このエンティティの補助クラスのリストです。補助クラスは存在しない場合もあります。補助クラスを追加する場合、次の項目を定義するよう指示されます。</p> <ul style="list-style-type: none">◆ 補助クラス。使用できるクラスのリストから選択します。◆ 検索可能にするかどうか。検索可能 (True) に設定すると、ディレクトリ抽象化レイヤ関係に関わる LDAP 検索にフィルタが適用されます。たとえば、ユーザエンティティに補助クラスを追加し、その補助クラスを検索可能に設定した場合、マネージャと従業員の関係を使用した組織図には、その補助クラスを持つ従業員だけが表示されます。◆ 常に追加するかどうか。True に設定 (選択) した場合、ユーザアプリケーションでエンティティが変更されると、オブジェクトクラスが自動的に追加されます。変更には、作成操作と更新操作が含まれます。False に設定すると、この補助クラスに関連付けられている属性が変更された場合にのみ、オブジェクトクラスが追加されます。

エンティティの検索プロパティ

表 3-9 エンティティの検索プロパティ

プロパティ名	説明
検索コンテナ	<p>検索が開始される LDAP ノードまたはコンテナの識別名 (検索ルート)。例:</p> <p>ou=sample, o=ourOrg</p> <p>識別ボルトをブラウズしてコンテナを選択できます。または、66 ページの「事前定義パラメータの使用」で説明されている事前定義パラメータの 1つを使用できます。</p>
検索スコープ	<p>検索ルートから始まる検索の範囲を指定します。次の値があります。</p> <p><デフォルト>: この検索スコープは、[コンテナとサブコンテナ] を選択した場合と同じです。</p> <p>コンテナ: 検索ルート DN と検索ルートレベルのすべてのエントリが検索されます。</p> <p>コンテナとサブコンテナ: 検索ルート DN とすべてのサブコンテナが検索されます。これは [<デフォルト>] を選択した場合と同じです。</p> <p>オブジェクト: 指定したオブジェクトに検索範囲を限定します。この検索は、指定したオブジェクトが存在しているか確認するために使用されます。</p>
検索の制限時間 [ms]	値をミリ秒単位で指定します。制限時間を指定しない場合は 0 を指定します。

プロパティ名	説明
最大検索エントリ	検索で返される検索結果エントリの最大数を指定します。ランタイム設定を使用する場合は 0 を指定します。推奨値：100～200 の範囲に設定すると最も効率的です。1000 より高い値は設定しないでください。
自動クエリの実行	選択した場合、エンティティの自動クエリが実行され、結果が選択可能リストに表示されます。大量のデータが返される場合は、ユーザが長い結果セットをスクロールしなければならなくなるため、このオプションを選択しないでください。
	選択しない場合、ユーザがエンティティクエリの検索条件を指定できます。結果は選択可能リストに表示されます。

エンティティの作成プロパティ

表3-10 エンティティの作成プロパティ

プロパティ名	定義
コンテナの作成	このタイプの新しいエンティティが作成されるコンテナの名前。 識別ポルトをブラウズしてコンテナを選択できます。または、66 ページの「事前定義パラメータの使用」で説明されている事前定義パラメータの 1 つを使用できます。
ネーミング属性の作成	この値を指定していない場合は、作成ポートレットで、新しいオブジェクトのコンテナを使用するよう促すメッセージがユーザに表示されます。ポートレットではエンティティ定義で指定された検索ルートが基準として使用され、ユーザは検索ルートからドリルダウンできます。エンティティ定義で検索ルートが指定されていない場合は、ユーザアプリケーションのインストール時に指定されたルート DN が使用されます。
LDAP 属性	エンティティの名前付け属性です。これは相対識別名 (RDN) です。エンティティでアクセスパラメータ Create が選択されている場合だけ、この値が必要です。
名前付きラベルの作成	[ネーミング属性の作成] の LDAP 属性です。 ユーザアプリケーションで [ネーミング属性の作成] に表示される表示ラベルです。

エンティティのパスワード管理プロパティ

表3-11 エンティティのパスワード管理プロパティ

プロパティ名	定義
エンティティを作成するときにはパスワードが必要です。	パスワード属性が必要な場合、この属性を True に設定（選択）して、作成ポートレットでユーザにパスワードを要求します。パスワードが必要な場合、ワークフローでこのエンティティを作成できません。

事前定義パラメータの使用

ディレクトリ抽出化層エディタでは、特定の値に対して事前定義パラメータを使用できます。

表 3-12 事前定義パラメータ

事前定義パラメータ	説明
%driver-root%	プロビジョニングドライバの DN を表します。この値はインストール時、あるいはその後の設定時にユーザーアプリケーションの設定の中で指定されます。これはユーザーアプリケーションのルーム設定に保存されます。
%user-root%	ユーザーコンテナ DN を表します。この値はインストール時、あるいはその後の設定時にユーザーアプリケーションの設定の中で指定されます。これはユーザーアプリケーションのルーム設定に保存されます。
%group-root%	グループコンテナ DN を表します。この値は、インストール時、あるいはその後の設定時にユーザーアプリケーションの設定の中で指定されます。これはユーザーアプリケーションのルーム設定に保存されます。

3.7.2 属性のプロパティ

属性には次の種類のプロパティを設定できます。

- ◆ 66 ページの 「属性のアクセスプロパティ」
- ◆ 68 ページの 「属性の一般プロパティ」
- ◆ 68 ページの 「属性のデフォルト値プロパティ」
- ◆ 68 ページの 「属性の UI 制御プロパティ」
- ◆ 71 ページの 「属性および DNLookup プロパティ」

属性のアクセスプロパティ

注: あるエンティティのすべての属性の属性アクセスを設定するには、[DAL] > [属性アクセスの設定] の順にクリックするか、またはエンティティを右クリックし、[属性アクセスの設定] をクリックします。

表 3-13 属性のアクセスプロパティ

名前	説明
編集	選択した場合、この属性をユーザーアプリケーションで編集および変更できます。このプロパティを選択 (True に設定) しても、基になる識別ボルトの ACL や有効な権利で編集が禁止されている場合は、この属性を編集できないことがあります。
有効にする	選択していない場合、ユーザーアプリケーションでこの属性を使用できなくなります。ファイルからエントリを削除するのと同じです。

名前	説明
非表示	<p>ユーザアプリケーションの [非表示] チェックボックスを有効にするか無効にするかを制御します。ユーザは [非表示] チェックボックスを使用して、属性（ユーザの写真など）をアプリケーションで表示するかどうかを制御できます。</p> <p>選択していない場合、この属性に対して [非表示] チェックボックスが無効になるため、ユーザはこの属性を非表示にできません。</p> <p>選択した場合、ユーザアプリケーションで [非表示] チェックボックスが有効になります。ただし、ログインユーザは次の条件を満たしている必要があります。</p> <ul style="list-style-type: none"> ◆ 属性の所有者、またはユーザアプリケーション管理者であること。 ◆ 識別ポールトの <code>srprvHideAttributes</code> 属性を更新するトラスティ権を持っていること。 <p>これらの条件が満たされていなければ、この設定が選択（True に設定）されても [非表示] チェックボックスはユーザインターフェースで無効になります。</p> <hr/> <p>ヒント: 画像を含む属性を非表示にした場合でも、ブラウザのキャッシュが更新されるまで画像が表示されることがあります。</p>
複数値	<p>この属性が複数の値を持つかを指定します（電話番号など）。</p> <p>選択した場合、属性は複数の値を持つことができます。</p>
読み込み	<p>このオプションを選択した場合、ユーザアプリケーションでこの属性を問い合わせることができます。大部分の属性でこのプロパティを選択（True に設定）する必要がありますが、パスワードなど一部の属性では選択解除する必要があります。</p>
必須	<p>このオプションを選択した場合、この属性は入力が必須になります。</p>
検索	<p>このオプションを選択した場合、ユーザアプリケーションでこの属性を検索できます。識別ポートレット（エンティティ検索リストやエンティティ組織図など）、または要求フォームや承認フォームでクエリに使用される属性を選択する必要があります。</p> <hr/> <p>ヒント: 検索で使用される属性も eDirectory でインデックス化すると、検索が速くなります。</p>
表示	<p>このオプションを選択した場合、ユーザアプリケーションでこの属性を表示できます。大部分の属性でこのプロパティを選択する必要がありますが、パスワードなど一部の属性では選択解除する必要があります。要求フォームまたは承認フォームで指定する場合は、選択する必要があります。</p>

属性の一般プロパティ

表 3-14 属性の一般プロパティ

プロパティ名	説明
キー	属性の固有の識別子です。
表示ラベル	ユーザアプリケーションで表示されるラベルです。
属性名	この属性の eDirectory 名。
LDAP 名	この属性の LDAP 名。

属性のデフォルト値プロパティ

この値は、作成識別ポートレットまたはワークフローでオブジェクトが作成されるときに使用されます。デフォルト値は、リテラルまたは ECMAScript 式で表すことができます。デフォルト値を、計算される属性の一部として使用することはできません。ECMAScript 式で定義した場合、式はランタイム時に解決されます。リテラルと式の両方を定義した場合、式が優先されます。

ヒント: 作成ポートレットでデフォルト値を表示したい場合は、表示可能アクセスプロパティを True に設定（選択）する必要があります。ユーザが値を変更できるようにしたい場合は、編集可能プロパティを True に設定する必要があります。

属性の UI 制御プロパティ

表 3-15 属性の UI 制御プロパティ

プロパティ名	説明
データタイプ	次のリストからデータタイプを選択します。 <ul style="list-style-type: none">◆ バイナリ◆ ブール◆ DN◆ 整数◆ LocalizedString◆ 文字列◆ [時刻]

プロパティ名	説明
フォーマットタイプ	ユーザアプリケーションでデータをフォーマットするために使用されます。次のフォーマットタイプがあります。

- ◆ なし
- ◆ AOL IM
- ◆ 電子メール
- ◆ Groupwise IM
- ◆ 画像
- ◆ 電話番号
- ◆ Yahoo IM
- ◆ 画像 URL
- ◆ 日付
- ◆ 日時

フォーマットタイプはデータタイプに依存しています。たとえば、時間のデータタイプは、「日付」と「日時」のフォーマットにのみ関連付けられます。

プロパティ名	説明
--------	----

制御タイプ	<p>次のタイプがあります。</p> <p>DNLookup: この属性が DN 参照を含むことを定義します。次の場合に使用します。</p> <ul style="list-style-type: none"> 関連するエンティティでの DN 検索の結果をリストに入力する。 更新や削除が行われた場合、複数の DN 参照属性間の参照整合性を維持する。 オブジェクトセレクタダイアログボックスでこの属性を使用する。オブジェクトセレクタは、詳細ポートレットなどの特定の識別ポートレットで使用されます。また、プロビジョニング要求フォームおよび承認フォームに対して定義できるフォームコントロールでも使用できます。
-------	--

ユーザーアプリケーションはこの情報を使用して特別なユーザインターフェース要素（オブジェクトセレクタなど）を生成し、DNLookup 定義に基づいて最適化された検索を実行します。

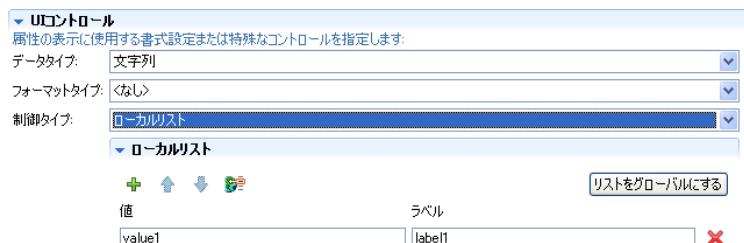
このプロパティの定義の詳細については、「[71 ページの「属性および DNLookup プロパティ」](#)」を参照してください。要求フォームと承認フォームのオブジェクトセレクタダイアログボックスの詳細については、「[152 ページのセクション 6.6.2 「オブジェクトセレクタの操作」](#)」を参照してください。

グローバルリスト: この属性をドロップダウンリストとして表示します。リストの内容はこの属性定義以外のファイルで定義されます。選択したリストのグローバルリストエディタにアクセスするには、[一覧を表示] をクリックします。

詳細については、[51 ページのセクション 3.3 「リストの操作」](#) を参照してください。

ローカルリスト: この属性をドロップダウンリストとして表示します。リストの内容はこの属性で定義されます。ローカルリストを定義するには、次の手順を実行します。

- 属性が選択された状態で、[制御タイプ] を [ローカルリスト] に設定します。



- ボタンを使用して、リストの項目を追加または削除します。リスト内の項目の位置を変更するには、上下の矢印ボタンを使用します。
[値] 列で、識別ボルトに書き込む値を入力します。使用できる文字は、英字、数字、およびアンダースコア (_) です。
- [ラベル] 列で、ユーザインターフェースに表示するテキストを入力します。

範囲: [範囲] の制御タイプ（整数データタイプ）で、ユーザが入力できる値を連続した一定の範囲内に限定します。範囲の開始値と終了値を定義します。

属性および DNLookup プロパティ

属性を DNLookup 制御タイプとして定義すると、次のようにになります。

- ◆ この属性をオブジェクトセレクタダイアログボックスで使用すると、ユーザは、この属性を検索するときに、可能な値のリストから選択できます。
- ◆ ユーザアプリケーションでこの属性が作成、入力、または削除されると、ユーザの操作（作成、削除、更新）に基づいて関連するエンティティの属性が適切に更新され、参照整合性が維持されます。

オブジェクトセレクタの DNLookup

特定の属性の DNLookup 表示プロパティで、ユーザアプリケーションのオブジェクトセレクタの内容を定義します。オブジェクトセレクタは、Identity セルフサービスポートレット、およびワークフローの要求フォームと承認フォームで表示されます。ユーザはこれを使用して、DN（ユーザやグループなど）を表すオブジェクトを簡単に検索および選択できます。オブジェクトセレクタには属性のドロップダウンリストが表示されます。ユーザは、このリストから属性を選択して、その属性の検索条件を入力できます。次の例では、ユーザはグループの説明によってグループを検索します。

図3-2 オブジェクトセレクタの例



ユーザが選択した検索の結果は次のようにになります。

図3-3 オブジェクトセレクタの結果例



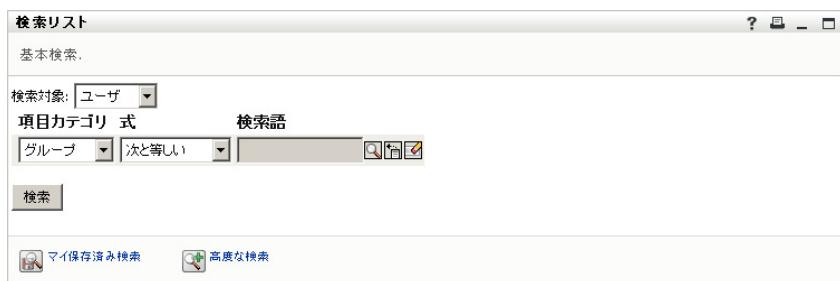
DNLookup 表示プロパティでは、オブジェクトセレクタの内容および結果セットを制御します。オブジェクトセレクタが、前に示したように表示されるのは、このオブジェクトセレクタがユーザエンティティのグループ属性に基づいていたためです。グループ属性は、次に示すように、DNLookup 制御タイプとして定義されます。

図3-4 グループのDNLookup の定義



この定義では、識別ポートレットでグループのリストをユーザに提供する方法も制御します。たとえば、あるユーザが、あるグループ内のユーザを見つけるためにディレクトリ検索を行う場合に、そのグループ名が不明であるとします。このユーザは、次のように、検索対象として [ユーザ] を選択し、検索条件として [グループ] を選択します。

図3-5 検索条件



メンバー属性はユーザエンティティの DNLookup であるため、[ルックアップ] アイコンが表示されます。ユーザがこのアイコンを選択すると、グループの候補リストが表示されます。

図3-6 オブジェクトのルックアップ

The screenshot shows a web-based object lookup interface titled 'オブジェクトルックアップ' (Object Lookup). The URL in the address bar is 'http://192.168.5.74:8080 - オブジェクトルックアップ'. The interface includes a search bar with placeholder text 'オブジェクトリストの検索: (例: a*, Lar*, ID, *r)' and a search button. Below the search bar, there is a '説明' (Description) dropdown and a list of objects. The list contains the following items:

- Accounting
- Executive Management
- Human Resources
- Improve Customer Service task force
- Information Technology
- Marketing
- Sales

At the bottom of the list, it says '1 - 7 / 7'.

ユーザがグループを選択するときに、リストからグループを選択すると、そのグループのすべてのメンバーが表示されます。

注：[自動クエリの実行] プロパティが選択されていない (False に設定されている) 場合、最初にユーザに表示されるオブジェクトセレクタには何も入力されておらず、ユーザは選択条件を入力する必要があります。前の例は、[自動クエリの実行] プロパティが選択 (True に設定) されている場合に表示されるオブジェクトセレクタを示しています。

参照整合性のための DNLookup

LDAP ではグループ関係を両方向にマップできるため、更新や同期のための DNLookup は重要です。たとえば、データは次のように設定されているかもしれません。

- ◆ ユーザオブジェクトにグループ属性が含まれる。グループ属性は、複数の値を持ち、ユーザが属するすべてのグループを一覧表示します。
- ◆ グループオブジェクトにユーザ属性が含まれる。グループ属性は、複数の値を持ち、グループに属するすべてのユーザを一覧表示します。

つまり、ユーザオブジェクトではユーザが属しているすべてのグループを示す属性を持つことができます。また、グループオブジェクトではグループ内のすべてのメンバーを含む DN 属性を持つことができます。

ユーザが更新を要求した場合、ユーザアプリケーションは関係を遵守し、ターゲット属性とソース属性を同期させる必要があります。DNLookup では、同期させる必要がある両方の属性を指定します。この手法を使用して、グループ構造のオブジェクトだけではなく、関連性のあるすべてのオブジェクトを同期させることができます。この種類の DNLookup 制御タイプを作成するときは、「*DNLookup 関係整合性プロパティ*」で説明されている DNLookup の詳細プロパティを指定します。

DNLookup プロパティのリファレンス

表 3-16 DNLookup の表示プロパティ

プロパティ名	説明
ルックアップエンティティ	検索するエンティティの名前です。たとえば、[ユーザ] エンティティに [マネージャ] の属性が含まれているとします。このフィールドに入力するには、どのユーザがマネージャかを知っている必要があります。
ルックアップ属性	検索の完了時に表示する属性を 1 つ以上選択します。
自動クエリの実行	ルックアップ属性の表示方法を定義します。 <ul style="list-style-type: none">◆ このオプションを選択した場合、フォームまたはポートレットで、エンティティの自動クエリが実行され、結果が選択可能リストに表示されます。大量のデータが返される可能性がある場合は、ユーザが長い結果セットをスクロールしなければならなくなるため、このオプションを選択することはお勧めしません。◆ このオプションを選択しない場合、ユーザがエンティティクエリの検索条件を指定できます。結果は選択可能リストに表示されます。

表 3-17 DNLookup の詳細プロパティ

プロパティ名	説明
詳細エンティティ	ユーザーアプリケーションでユーザがハイパーテキストリンクをクリックして詳しい情報を要求した場合に、詳細を表示するエンティティのキーです。DNLookup を定義すると、識別ポートレットでハイパーテキストリンクを表示できるようになります。ユーザはこれを使用してリンク先のオブジェクトの詳細を表示できます。

DNLookup 関係整合性プロパティは、グループとグループメンバーなど 2 つのオブジェクト間のデータを同期させるために使用されます。

表 3-18 DNLookup の関係整合性プロパティ

プロパティ名	説明
更新するソース属性	更新する属性の名前。属性には [更新するターゲット属性] への DN 参照が含まれている必要があります。これは 2 つの異なるオブジェクトの属性を同期させる場合に必要となります。
更新するターゲット属性	[更新するソース属性] と一緒に更新が必要な属性の名前。これは LDAP 属性名です。これは、2 つの異なるオブジェクトの属性を同期させる場合に必要です。属性に DN 参照が含まれている必要があります。
必要なターゲット補助クラス（存在する場合）	[更新するターゲット属性] を含む補助クラスの名前。

3.7.3 クエリのプロパティ

次の種類のクエリのプロパティを設定できます。

- ◆ 75 ページの「クエリの一般プロパティ」
- ◆ 76 ページの「クエリパラメータのプロパティ」
- ◆ 76 ページの「クエリ検索のプロパティ」

クエリの一般プロパティ

表 3-19 クエリの一般プロパティ

プロパティ名	説明
キー	クエリキーの固有値。この値は、式ビルダでクエリを識別するために使用されますこのキーは、クエリの作成時に指定されます。エンティティの作成後は変更できません。
クエリエンティティ	ドロップダウンリストからエンティティを選択します。結果として得られる LDAP 検索は、このエンティティに対するものです。

プロパティ名	説明
表示ラベル	ディレクトリ抽象化レイヤエディタおよびプロビジョニングビューに表示する文字列を入力します。この値は、式ビルダには表示されません。

クエリパラメータのプロパティ

表 3-20 クエリパラメータのプロパティ

プロパティ名	説明
パラメータキー	キーの固有識別子。フォームで <code>globalQuery()</code> メソッドを呼び出すときに、このキーを渡します。
パラメータ表示ラベル	キーを識別するラベル。

クエリ検索のプロパティ

空白のままにすると、クエリ検索プロパティのデフォルト設定は、選択されたエンティティで指定されている検索プロパティになります。エンティティすでに定義されている検索スコープをさらに絞り込む場合に、クエリ検索プロパティを指定します。クエリの検索プロパティの事前定義パラメータ（%user-root% など）は指定できません。

表 3-21 クエリ検索のプロパティ

プロパティ名	説明
検索ルート	クエリで定義された LDAP 検索を開始する、LDAP ツリー内の場所を指定します。
検索スコープ	検索ルートから始まる検索の範囲を指定します。次の値があります。 ＜デフォルト＞: この検索スコープは、[コンテナとサブコンテナ] を選択した場合と同じです。 コンテナ: 検索ルート DN と検索ルートレベルのすべてのエントリが検索されます。 コンテナとサブコンテナ: 検索ルート DN とすべてのサブコンテナが検索されます。これは＜デフォルト＞を選択した場合と同じです。 オブジェクト: 指定したオブジェクトに検索範囲を限定します。この検索は、指定したオブジェクトが存在しているか確認するために使用されます。
最大検索エントリ	検索で返される検索結果エントリの最大数を指定します。ランタイム設定を使用する場合は 0 を指定します。推奨値: 100 ~ 200 の範囲に設定すると最も効率的です。1000 より高い値は設定しないでください。

3.7.4 関係のプロパティ

次の関係プロパティがあります。

- ◆ 77 ページの 「関係のアクセスプロパティ」

- ◆ 77 ページの 「関係のプロパティ」

関係のアクセスプロパティ

表3-22 関係のアクセスプロパティ

プロパティ名	説明
組織図で使用	選択した場合、組織図ポートレットでこの関係を使用できます。
チーム管理で使用	選択した場合、iManager でプロジェクトマネージャーを定義する場合にこの関係を使用できます。 たとえば、マネージャと従業員の関係で [チーム管理で使用] が選択されている場合、プロジェクトマネージャーの管理者は、この関係を使用して、チームマネージャにレポートするすべてのユーザをチームメンバーとして定義できます。 [階層式関係を有効にする] が選択されている場合、組織内のいくつかのレベルをチームに含めることができます。 [階層の最高レベル] でレベル数を指定できます。

関係のプロパティ

表3-23 関係のプロパティ

プロパティ名	説明
キー	関係の固有の識別子で、読み込み専用です。 ヒント: この値は組織図ポートレットの初期設定シートで指定します。
表示ラベル	ユーザアプリケーションでこの関係を表示するときに使用する名前を指定します。たとえば、詳細ポートレットでユーザが [組織図の選択] をクリックすると、この値が表示されます。 表示ラベルのテキストの翻訳を入力するには、[ローカライズ] をクリックします。
ソースエンティティ	ドロップダウンリストからエンティティを選択します。 選択したエンティティは、組織図の階層で親オブジェクトまたはソースオブジェクトになります。マネージャと従業員の関係では、[ソースエンティティ] は [ユーザ] です。グループとメンバーの関係では、[ソースエンティティ] は [グループ] です。 ディレクトリ抽象化レイヤ要件: このリストのエンティティはディレクトリ抽象化レイヤで定義されたエンティティのサブセットです。ソースエンティティでは、表示アクセスプロパティが選択 (True に設定) されている必要があります。

プロパティ名	説明
ソース属性	<p>ドロップダウンリストから属性を選択します。</p> <p>この属性は、対応するターゲットエンティティの検索に使用されます。この属性値がターゲットエンティティの属性に含まれる対応した値と一致する場合（後の「ターゲット属性」を参照）、関係を確立できます。</p> <p>ディレクトリ抽象化レイヤ要件：属性リストは、選択したソースエンティティの属性を使用して入力されます。リストには、検索可能および読み込み可能なすべての属性が含まれます。</p>
ターゲットエンティティ	<p>階層内の子オブジェクトのエンティティを選択します。マネージャと従業員の関係では、ユーザです。</p> <p>このエンティティには、ソース属性に関係した属性が含まれている必要があります。</p>
ターゲット属性	<p>ソース属性に一致する属性を選択します。</p> <p>これは、対応するソースエンティティの検索に使用される、ターゲットエンティティの属性です。この属性値がソースエンティティに含まれる対応した値と一致する場合（前の「ソース属性」を参照）、関係を確立できます。</p>
注： 組織図ポートレットでは、ダイナミックグループは完全にはサポートされません。ダイナミックグループをソースエンティティとして定義することはできませんが、ターゲットエンティティとして定義することはできます。	

プロビジョニング要求定義エディタ の操作

4

この項では、プロビジョニング要求定義エディタを使用する際の一般的なガイドラインについて説明します。主なトピックは次のとおりです。

- ◆ 79 ページのセクション 4.1 「プロビジョニング要求定義エディタについて」
- ◆ 85 ページのセクション 4.2 「プロビジョニング要求定義を作成する場合の基本ステップ」
- ◆ 86 ページのセクション 4.3 「ワークフロー作成のガイドライン」
- ◆ 93 ページのセクション 4.4 「インストールされているテンプレートでの作業」
- ◆ 95 ページのセクション 4.5 「ワークフローのデバッグ」

4.1 プロビジョニング要求定義エディタについて

プロビジョニング要求定義エディタを使用することにより、豊富な Eclipse ベースの設計ツールを使用してカスタムプロビジョニング要求定義を作成できます。プロビジョニング要求定義エディタでは、プロビジョニング要求の基本特性の定義、関連するワークフローの定義、初期要求フォームと承認フォームのモデル作成を行えます。

Identity Manager には、定義の作成に使用できるプロビジョニング要求テンプレートのセットが用意されています。テンプレートはいくつかの一般的なワークフロー設計パターンのモデルとしています。しかし、ワークフローの動作をすべて制御できるようにするために、プロビジョニング要求定義を独自に作成することもできます。

注：テンプレート使用の詳細については、「93 ページのセクション 4.4 「インストールされているテンプレートでの作業」」を参照してください。

4.1.1 プロビジョニング要求定義エディタを Identity Manager アーキテクチャに適合させる方法

ワークフローベースのプロビジョニングで、Identity Manager の主要な機能で、組織のセキュアリソースへのユーザアクセスを管理するプロセスです。このようなリソースには、ユーザアカウント、コンピュータ、データベースなどのデジタルエンティティが含まれます。プロビジョニングされたリソースは Identity Manager エンタイトルメントまたはディレクトリ抽象化レイヤのエンティティにマップされます。

Identity Manager は広い範囲のプロビジョニング要求に対応しています。プロビジョニング要求は、組織リソースへのアクセスの許可または取り消しのための、ユーザまたはシステムのアクションです。プロビジョニング要求は、Identity Manager ユーザアプリケーションからエンドユーザが直接開始することもできますし、識別ポールト (eDirectory™) で発生するイベントに対応して間接的に開始することもできます。

プロビジョニング要求に対して、組織内の 1 人以上の個人による承認が必要な場合、ワークフローが開始されます。このワークフローでは、要求の処理に必要な承認が調整されます。1 人の個人からの承認を必要とするプロビジョニング要求もあれば、複数の個人から

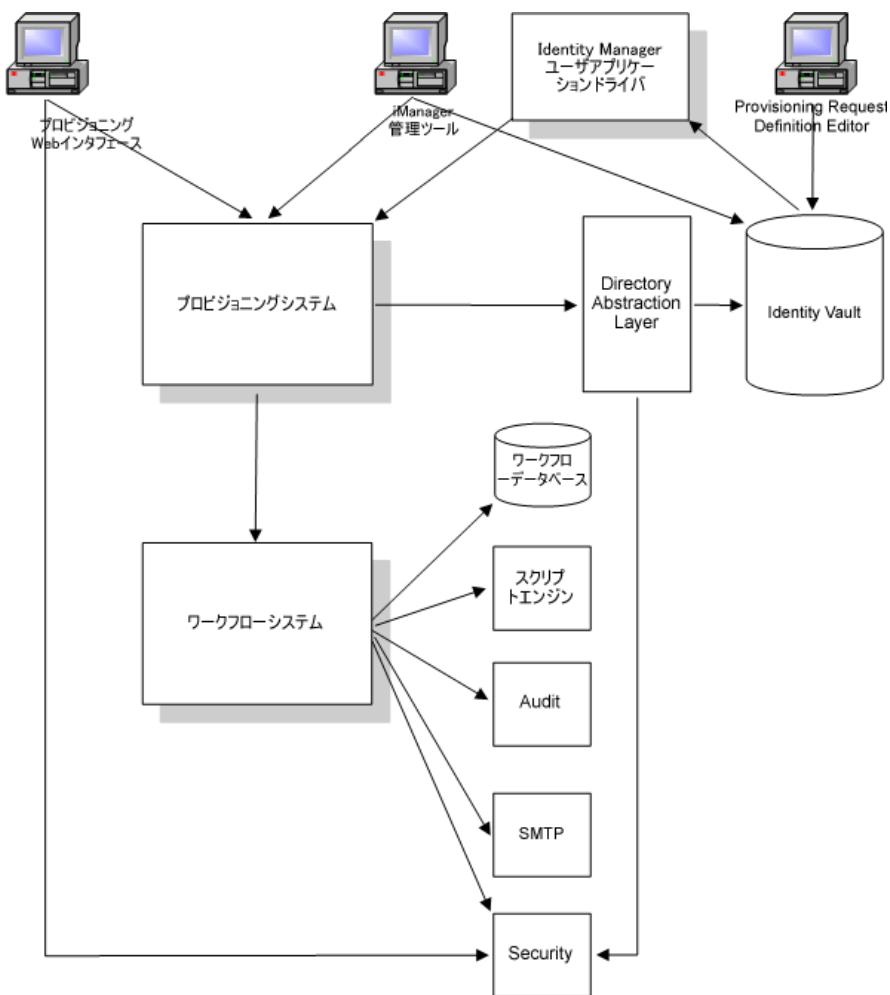
の承認を必要とするプロビジョニング要求もあります。場合によっては、承認なしに実行できる要求もあります。

承認手順が順次実行される連続方式の処理を必要とするワークフローもあれば、並行処理をサポートするワークフローもあります。プロビジョニング要求を定義する場合、ワークフローが、連続処理または並行処理のどちらをサポートするかを指定します。

プロビジョニング要求を設定するには、リソースをワークフローにバインドするプロビジョニング要求定義を作成します。Identity Manager にはプロビジョニング要求定義エディタが用意されていて、プロビジョニング要求とそれに関連付けられているワークフローのすべての動作を制御できます。Identity Manager には iManager プラグインのセットも含まれています。このプラグインを使用してすでに展開されているプロビジョニング要求定義をカスタマイズすることもできます。iManager ツールを使用して、プロビジョニング要求定義の動作に小さな変更を加えたり、処理中のワークフローを管理したりできます。

次の図は、プロビジョニング要求定義エディタが Identity Manager によって組み込まれたワークフローベースのプロビジョニングシステムにどのように適合するかを示しています。

図4-1 プロビジョニング要求定義エディタとワークフローアーキテクチャ



4.1.2 プロビジョニングおよびワークフローの例

IT システムのアカウントを必要とするユーザがいると想定します。アカウントを設定するには、ユーザは、Identity Manager ユーザアプリケーションを使用して要求を開始します。この要求によりワークフローが開始され、ワークフローは承認プロセスを調整します。必要な承認が付与されると、要求が実行されます。このプロセスは 4 つの基本的なステップからなっています。

- ◆ 81 ページの 「ステップ 1: 要求の開始」
- ◆ 81 ページの 「ステップ 2: 要求の承認」
- ◆ 85 ページの 「ステップ 3: 要求の実行」
- ◆ 85 ページの 「ステップ 4: ワークフローの完了」

ステップ 1: 要求の開始

Identity Manager ユーザアプリケーションで、ユーザはリソースリストをカテゴリで参照し、プロビジョニングを行うリソースを 1 つ選択します。識別ポルトで、選択したプロビジョニング対象リソースがプロビジョニング要求定義に関連付けられます。プロビジョニング要求定義は、プロビジョニングシステムで最も重要なオブジェクトです。プロビジョニング要求定義は、プロビジョニング対象リソースをワークフローにバインドし、ワークフロープロセスをエンドユーザに明らかにする手段としての役割を果たします。プロビジョニング要求定義は、初期要求形式を表示したり、初期要求に続くフロー開始したりするために必要なすべての情報を提供します。

この例では、ユーザは新規アカウントリソースを選択します。ユーザが要求を開始すると、Web アプリケーションは、プロビジョニングシステムから関連する初期要求データの初期要求形式および `description` 属性を取得します。すると、プロビジョニング要求定義からこれらのオブジェクトを取得します。

プロビジョニング要求が開始されると、プロビジョニングシステムによってイニシエータと受信者が追跡されます。イニシエータとは、要求を作成した人物です。受信者とは、作成された要求の宛先の人物です。場合によっては、イニシエータおよび受信者が同一人物ということもあります。

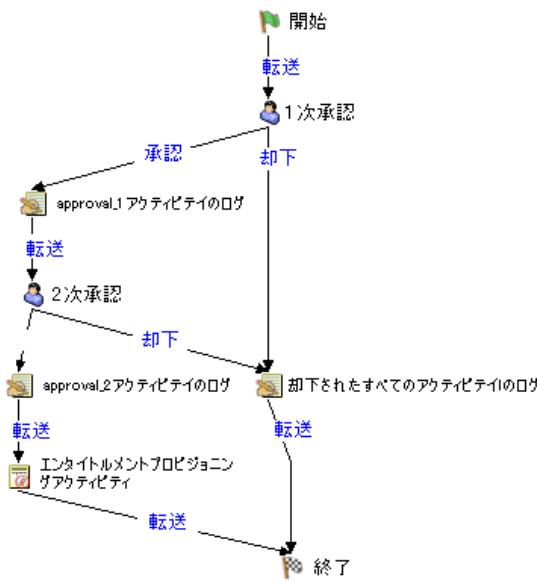
各プロビジョニング要求には、それに関連付けられた操作があります。ユーザがリソースを許可するか取り消しするかは、操作により指定されます。

ステップ 2: 要求の承認

ユーザが要求を開始すると、プロビジョニングシステムはワークフロープロセスを開始します。ワークフロープロセスは、承認を調整します。この例では、2 つのレベルの承認が必要です。1 つはユーザのマネージャからの承認で、もう 1 つはマネージャのスーパーバイザからの承認です。承認がワークフロー内のユーザに拒否された場合、フローは終了し、要求は拒否されます。

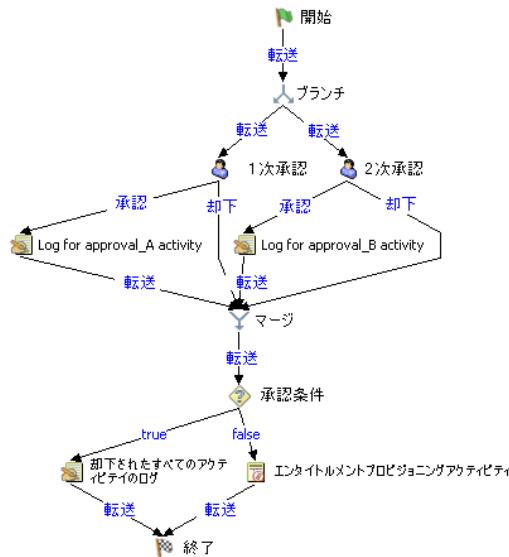
ワークフローは承認を順次に、または並行に処理できます。順次ワークフローでは、次の図に示されているように、1 つの承認タスクを処理してから、次の承認タスクを開始する必要があります。

図4-2 2つの承認の順次ワークフロー



並行ワークフローでは、次の図に示されているように、ユーザが複数の承認タスクを同時に処理できます。

図4-3 2つの承認の並行ワークフロー



注：表示ラベル（1次承認、2次承認など）はアプリケーション要件に合わせて簡単に変更できます。並行フローの場合、順次処理を意味しないラベルを指定する必要があります。たとえば、One of Three Parallel Approvals、Two of Three Parallel Approvalsなどのラベルを割り当てる場合です。

ワークフロー定義は、次の表に示されているコンポーネントで構成されています。

表4-1 ワークフロー定義コンポーネント

プロセスコンポーネント	説明
アクティビティ	<p>アクティビティとは、タスクを表すオブジェクトのことです。アクティビティは情報をユーザに提示し、ユーザの操作に応答できます。ユーザに表示されないバックグラウンド機能を行うこともできます。</p> <p>ワークフローダイアグラムでは、アクティビティは四角形で表されています。</p> <p>Identity Manager ユーザアプリケーションでは、承認プロセスを処理するアクティビティは、タスクと呼ばれます。[My Work] アクショングループの [マイタスク] をクリックすると、エンドユーザは、自分のキューにあるタスクのリストを確認できます。特定のタスクのためにどのワークフローアクティビティが処理されているかを確認するには、タスクを選択してから、[Task Detail] フォームの [View Comment History] リンクをクリックします。</p> <p>特定のプロビジョニング要求のためにどのワークフローアクティビティが処理されているかを確認するには、[My Requests] をクリックして要求を選択してから、[Request Detail] フォームの [View Comment and Flow History] リンクをクリックします。</p> <p>[マイタスク] および [マイリクエスト] の各アクションの詳細については、『Identity Manager 3.5.1 ユーザアプリケーション：ユーザガイド』を参照してください。</p>
フローパス	<p>ワークフローでアクティビティどうしを結んでいるのがフローパスです。フローパスは、2つのアクティビティ間でたどられるパスを示します。</p> <p>1つのアクティビティは複数の着信フローパスと複数の発信フローパスを持つことができます。1つのアクティビティから複数の発信フローパスが分岐している場合、アクティビティの結果により、選択されるフローパスが決定されることがほとんどです。結果とは、アクティビティによって実行された処理の最終結果のことです。たとえば、承認アクティビティは、ユーザが実行するアクションにより、承認という結果になる場合と、拒否という結果になる場合があります。</p> <p>ワークフローの図では、フローパスは矢印で示されています。</p>

開始アクティビティ：ワークフロープロセスは、開始アクティビティの実行により開始されます。アクティビティはユーザに対して初期要求フォームを表示します。ユーザが初期要求データを指定すると、アクティビティはこのデータを使用して作業ドキュメントを初期化します。開始アクティビティはイニシエータ、受信者などの複数のシステム値をバインドし、これらの値をスクリプト式で使用できるようにします。

承認アクティビティ：開始アクティビティの実行が終了すると、ワークフローシステムはフロー内の1次承認アクティビティの処理に進みます。承認アクティビティは承認者に電子メールを送信し、注意を向ける必要があることを通知します。ユーザがタスクを要求すると、承認アクティビティは承認フォームを表示し、このフォームを使用してユーザは要求を実行できます。81ページの「**ステップ 2: 要求の承認**」に示したワークフローの例では、[1次承認] および [2次承認] が承認アクティビティの例になります。承認アクティビティの表示ラベルは、使用する国の要件を満たすためにローカライズできます。

承認アクティビティの結果には 5 種類あり、それぞれはアクティビティからの異なるフロー パスで示されています。

- ◆ Approved
- ◆ Denied
- ◆ Refused
- ◆ エラー
- ◆ タイムアウト

注: 「エラー」と「タイムアウト」の結果は、ユーザがどのアクションも実行しなかった場合にも発生します。

ユーザが要求を承認すると、ワークフローは承認済みフロー パスをたどってフロー内の次のアクティビティに進みます。これ以上承認が必要ない場合は、リソースがプロビジョニングされます。ユーザが要求を拒否すると、ワークフローは拒否されたフロー パスをたどってフロー内の次のアクティビティに進みます。または、(ユーザが組織マネージャまたはユーザアプリケーション管理者である場合は) ユーザがタスクを再割り当てすることができます。すると、ワーク項目は別のユーザのキューに送られます。

承認アクティビティが割り当てる人を「宛先」と呼びます。アクティビティの宛先に、割り当てるタスクを電子メールで通知することもできます。アクティビティに関連付けられた作業を実行する場合、宛先は電子メールにある URL をクリックし、ワーカリスト (キュー) にあるタスクを見つけ、タスクを要求します。

宛先ユーザは指定時間内に承認アクティビティに応答する必要があります。応答しないとアクティビティでタイムアウトが発生します。通常、タイムアウト間隔は、ユーザが応答できるだけの十分な時間を時間単位または日単位で指定します。

アクティビティのタイムアウトが発生した場合、そのアクティビティに指定された再試行回数により、ワークフロープロセスはアクティビティを再試行します。場合によっては、ワークフロープロセスは、タイムアウトの発生したアクティビティをエスカレートするよう設定されます。この場合、このアクティビティは新しい宛先 (たとえばユーザのマネージャ) に再割り当され、新しい宛先ユーザがこのアクティビティの作業を完了できるようにします。最後の再試行でタイムアウトが発生した場合、ワークフローの設定内容に従い、そのアクティビティは承認済みまたは却下済みとしてマークされます。

ログアクティビティ: ログアクティビティはログにメッセージを書き込むシステムアクティビティです。ワークフロープロセスの状態に関する情報をログとして記録するとき、ワークフローシステムは Novell® Auditとのやり取りを行います。ワークフローは、その処理中、発生したさまざまなイベントについての情報を記録します。ログデータは、Novell Audit レポートツールを使用して確認できます。

ブランチアクティビティとマージアクティビティ: 並行処理をサポートするワークフローでは、ブランチアクティビティにより、2人のユーザが1つの作業項目の異なる領域を同時に作業できます。ユーザが作業を完了すると、マージアクティビティにより、フローに合流するブランチが同期化されます。

条件アクティビティ: ワークフロープロセスの実行中、次の作業を決定するために、結果のテストとチェックが実行されます。条件アクティビティがこの機能を提供しています。条件アクティビティは、スクリプト式を使用して評価する条件を定義します。[81 ページの「ステップ 2: 要求の承認」](#)に示したワークフローの例では、「[承認条件]」が条件アクティビティの例です。

条件アクティビティは、3種類の取りうる結果または終了パスをサポートします。

- ◆ True
- ◆ False
- ◆ エラー

ステップ 3: 要求の実行

プロビジョニング要求が承認されると、ワークフローシステムはプロビジョニングのステップを開始できます。この時点で、制御はプロビジョニングシステムに戻ります。

プロビジョニング要求を実行するにあたり、プロビジョニングシステムは、Identity Manager エンタイトルメントを実行することも、eDirectory オブジェクトおよびその属性を直接操作することもできます。これらのアクションはエンタイトルメントアクティビティまたはエンティティアクティビティのどちらかによって実行されます。

エンタイトルメントアクティビティ：エンタイトルメントアクティビティは、エンタイトルメントの許可または取り消しによってプロビジョニング要求を実行します。通常必要な承認すべてが与えられない限り、エンタイトルメントアクティビティは実行されません。

エンティティアクティビティ：エンティティアクティビティは、eDirectory オブジェクトおよびその属性を直接操作することにより、プロビジョニング要求を実行します。通常必要な承認すべてが与えられない限り、エンティティアクティビティは実行されません。

ステップ 4: ワークフローの完了

他のすべてのアクティビティが終了すると、ワークフローは完了アクティビティを実行します。

完了アクティビティ：完了アクティビティは、ワークフローの最後のアクティビティです。フロー内のすべてのアクティビティが完了し、フローの最終結果が有効な場合、完了アクティビティが実行できます。完了アクティビティは最後の電子メール通知を送信して、ワークフローの完了を参加者に知らせます。

4.2 プロビジョニング要求定義を作成する場合の基本ステップ

プロビジョニング要求を定義する方法を次の表に示します。

表4-2 プロビジョニング要求を定義する場合の基本ステップ

タスク	アクション	詳細情報
ステップ 1: ウィザードを使用してプロビジョニング要求定義を作成	プロビジョニング要求に名前を付け、その基本特性を定義します。次に、要求の作成にテンプレートを使用するかどうかを指定します。 プロビジョニング要求定義は、ワークスペース内の Provisioning\ AppConfig\ RequestDefs ディレクトリにローカルに保存されます。	詳細については、97 ページの第 5 章「プロビジョニング要求定義の作成」を参照してください。
ステップ 2: フォームの作成	ワークフローの初期要求および承認フォームを作成します。先にフォームを作成することにより、ユーザインターフェースが正しいことを確認してから、実装の詳細に進むことができます。 フォームフィールドをアプリケーションデータにマッピングするプロセスを大幅に単純化できます。	詳細については、107 ページの第 6 章「プロビジョニング要求定義のフォームの作成」を参照してください。
ステップ 3: ワークフローダイアグラムの作成	ワークフローダイアグラムにアクティビティを追加して、各アクティビティをフローパスで接続します。	詳細については、「163 ページの第 7 章「プロビジョニング要求定義のワークフローの作成」」を参照してください。
ステップ 4: アクティビティとフローパスの設定	アクティビティのプロパティ、データ項目マッピング、メール通知の設定を指定します。次に、フローパスのセマンティクスを定義します。	詳細については、「187 ページの第 8 章「ワークフローアクティビティの参照」」を参照してください。

4.3 ワークフロー作成のガイドライン

適格なワークフローを作成するには、アクティビティとフローパス追加のルールを理解する必要があります。ワークフローデータを操作する方法を理解する必要があります。次のトピックを参照してください。

- ◆ 87 ページのセクション 4.3.1 「アクティビティのルール」
- ◆ 87 ページのセクション 4.3.2 「フローパスのルール」
- ◆ 89 ページのセクション 4.3.3 「ワークフローデータについての理解」

注: プロビジョニング要求定義は、展開前に検証することができます。詳細については、25 ページのセクション 2.6 「プロビジョニングオブジェクトの検証」を参照してください。

4.3.1 アクティビティのルール

ワークフローにアクティビティを追加するときに、次のルールに従います：

- ◆ 1つのワークフローには開始アクティビティと完了アクティビティをそれぞれ1つだけ設定できます。
- ◆ 1つのワークフローは次のアクティビティタイプを0個、またはそれ以上持つことができます。

承認アクティビティ
ログアクティビティ
ブランチアクティビティ
マージアクティビティ
条件アクティビティ
マッピングアクティビティ
ワークフローステータス
電子メールアクティビティ
統合
エンタイトルメント
エンティティ

- ◆ 各ブランチアクティビティには対応するマージアクティビティが必要です。
- ◆ プロビジョニングのステップが実行されていることを確かめるために、ワークフローに少なくとも1つのエンタイトルメントアクティビティまたはエンティティアクティビティが必要です。

4.3.2 フローパスのルール

ワークフローにフローパスを追加するときには、次のルールに従います：

- ◆ 開始アクティビティ以外のすべてのアクティビティには、1つ以上の着信フローパスを設定できます。開始アクティビティには着信フローパスを設定できません。
- ◆ 完了アクティビティには発信フローパスを設定できません。
- ◆ 開始アクティビティに設定できる発信フローパスは1つだけです。フローパスのタイプは転送でなければなりません。
- ◆ 承認アクティビティには1つから5つまでの発信フローパスを設定できます。有効なフローパスのタイプは、承認、却下済み、拒否済み、タイムアウト、エラーです。ランタイム時には、1つのフローパスだけが実行されます。
- ◆ エンタイトルメント、エンティティ、ログ、マージの各アクティビティに設定できる発信フローパスは1つだけです。フローパスのタイプは転送でなければなりません。
- ◆ 条件アクティビティには2つまたは3つの発信フローパスを設定できます。有効なフローパスのタイプは、True、False、エラーです。TrueとFalseのフローパスは必須で、エラーのフローパスはオプションです。
- ◆ ブランチアクティビティには1つ以上の発信フローパスを設定できます。それぞれのフローパスのタイプは転送でなければなりません。ランタイム時には、すべてのフローパスが実行されます。

着信および発信フローパスをアクティビティに追加するときのルールを、次の表にまとめて示します。

表4-3 各アクティビティに設定できるフローパスの数

アクティビティ	着信パス	発信パス
開始	0	1 (常に転送)
承認	1 以上	1 ~ 5 (承認、却下済み、拒否済み、タイムアウト、エラー)
ログ	1 以上	1 (常に転送)
ブランチ	1 以上	1 以上
マージ	1 以上	1 (常に転送)
条件	1 以上	2 ~ 3 (True と False は必須、エラーはオプション)
マッピング	1 以上	1
ワークフローステータス	1 以上	1 (常に転送)
電子メール	1 以上	1 (常に転送)
終了	1 以上	0
統合	1 以上	1 ~ 4 (成功、タイムアウト、エラー、フォルト)
エンタイトルメント	1 以上	1 (常に転送)
Entity (エンティティ)	1 以上	1 (常に転送)

ソースまたはターゲットとして設定できるアクティビティのタイプを、使用可能なフローパスタイプごとに、次の表にまとめて示します。

表4-4 アクティビティで使用可能なフローパスタイプ

アクティビティ	転送	Approved	Denied	Refused	タイムアウト	True	False	エラー	成功	フォルト
開始	ソース									
承認	ターゲット	ソース / ターゲット	ソース / ターゲット	ソース / ターゲット	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ソース / ターゲット	
ログ	ターゲット	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	
ブランチ	ターゲット	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	
マージ	ターゲット	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	

アクティビティ	転送	Approved	Denied	Refused	タイムアウト	True	False	エラー	成功	フォルト
条件	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ソース / ターゲット	ソース / ターゲット	ソース / ターゲット	ソース / ターゲット	ソース / ターゲット
マッピング	ソース	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット
ワークフローステータス	ソース / ターゲット									
電子メール	ソース / ターゲット									
完了	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット
統合	ソース / ターゲット				ソース / ターゲット			ソース / ターゲット	ソース	ソース
エンタリトルメント	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット
エンティティ	ソース / ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット	ターゲット

4.3.3 ワークフローデータについての理解

ワークフローの作成時には、プロビジョニングアプリケーションのニーズに合わせて、ワークフローデータを操作できます。

ワークフローでは、プロセスに関する情報を管理するために、1つのプロセスオブジェクトを使用します。ワークフロー内のアクティビティごとに別個のアクティビティオブジェクトが作成されて、ユーザとのやりとりを担う各アクティビティのフォームデータが保持されます。

フォーム上の各ユーザインターフェースコントロール（テキストフィールド、ドロップダウンリストなど）に関連付けられているデータオブジェクトは、対応するアクティビティ（開始アクティビティまたは承認アクティビティ）を実行する直前に変更することができます。アクティビティを実行した直後に、このデータを取得することもできます。制御が次のアクティビティに渡されると、フォームコントロールデータは使用できなくなります。そのため、ワークフローにはフローデータと呼ばれる特別なオブジェクトが用意されており、このオブジェクトを使用して独自のデータ項目を定義できるようになっています。このオブジェクトに独自の変数を追加して、フォームデータを含む、ワークフローにとって重要な情報が失われないように追跡することができます。

ワークフローデータのカテゴリを、次の表にまとめて示します。

表 4-5 ワークフローデータのカテゴリ

データオブジェクト	使用期間	編集可能	作成者
プロセス	ワークフロー	いいえ	システム
アクティビティ	ワークフロー	いいえ	システム
アクティビティフォーム	アクティビティ	はい	システムおよびワークフロー設計者
フローデータ	ワークフロー	はい	ワークフロー設計者

注: ワークフロー設計者とは、Designer でワークフローを作成するユーザのことです。

各オブジェクトタイプの変数を、次の表に示します。

表 4-6 ワークフロー内のデータ変数

オブジェクト	変数	説明
プロセス	approvalStatus	このプロセスの現在の状態。
	category	要求を開始したユーザによって選択されたプロジェクト（エンタイトルメントなど）。
	container dn	インストール時にユーザアプリケーションに対して定義されたコンテナの識別名。
	description	プロジェクト要求定義の説明。
	group container dn	インストール時にユーザアプリケーションに対して定義されたグループコンテナの識別名。
	id	プロジェクト要求定義の固有 IDVault ID (CN)。
	initiator	要求を開始したユーザの識別名。
	locale	現在のロケール。
	name	ワークフロープロセスの名前。
	provisioning driver dn	インストール時にユーザアプリケーションに対して定義されたプロジェクトドライバの識別名。
	recipient	プロジェクト対象リソースのターゲットの識別名。
	user container dn	インストール時にユーザアプリケーションに対して定義されたユーザコンテナの識別名。
	requestID	プロジェクト要求の ID。
	timestamp	プロセスが開始された時刻。

オブジェクト	変数	説明
<i>approval-activity-name</i>	action	ユーザが実行したアクション。
	addressee	承認アクティビティの現在の宛先。
	name	アクティビティの名前。
	timestamp	アクティビティがワークリストのキューに入った時刻。
	user	現在のアクティビティに関連付けられているユーザ。
	workId	現在のワークフローアクティビティのシステム生成固有ID。
<i>form-name</i>	custom-form-controls	ユーザによりフォームに追加されたユーザインタフェースコントロール。
フローデータ	custom-variables	ワークフローに必要なデータを保持するために作成されたカスタム変数。 インストール済みのテンプレートの1つを使用してワークフローを作成する場合は、フローデータオブジェクトに初期要求フォームの reason フィールドからコピーされたテキストを含む reason という変数が設定できます。

これらのオブジェクトは ECMAScript 式で参照できます。ワークフローのスクリプト式は、フロー内でアップストリームにバインドされているデータ項目を常に参照できます。しかし、ワークフロー式はダウンストリームで作成されるデータ項目を参照できず（その時点で該当するデータ項目は存在しないため）、並行処理をサポートするフローで他のブランチにバインドされているデータも参照できません（このブランチが現在のアクティビティと同時に実行されている可能性があるため）。

新しいデータ項目の作成

開始または承認アクティビティの [データ項目マッピング] タブに後動作ターゲット式を指定することにより、フローデータオブジェクトに新しいデータ項目を作成できます。[ターゲット式] 列に新しいデータ項目の名前を指定すると、変数が自動的に作成されます。その結果、このアクティビティの後に実行したアクティビティが新しいデータ項目にアクセスできるようになります。

たとえば、*[reason]* というフォームフィールドをターゲット式 `flowdata.myReason` にマップします。このとき変数 `myReason` は、ワークフロー内で後に実行されるすべてのアクティビティが使用できる新しいデータ項目になります。

データ項目の変更

開始または承認アクティビティの [データ項目マッピング] タブに前動作の式を指定することにより、データ項目を変更できます。たとえば、ドル記号を価格にプリprendするために、次のソース式を *[Price]* というターゲットフォームフィールドにマップします。

```
"$" + flowdata.get('cost')
```

フォームが表示されると、Price データが次のように示されます：

```
$xx.xx
```

別の例として、基本コストに税金を加えて総コストを計算します。そのためには、次のソース式を [TotalCost] というターゲットフォームフィールドにマップします：

```
Number(flowdata.get('cost')) + Number(flowdata.get('tax'))
```

複雑なデータ項目マッピングの処理

フローデータオブジェクト内のすべてのデータは XML 形式で保持されるため、データ項目を階層的に作成することもできます。たとえば、ユーザが買掛金勘定と売掛金勘定の 2 つの内部システムへのアクセスを要求できるワークフローフォームがあるとします。このフォームには、他のフィールドとともに [Acct_Pay] と [Acct_Rec] という名前の 2 つの Yes/No フィールドがあります。後動作のデータ項目マッピングで、次のように 2 つのマッピングを作成できます。

表 4-7 複雑なデータ項目マッピングの例

ソースフォームフィールド	ターゲット式
Acct_Pay	flowdata.SystemAccess/AcctPay
Acct_Rec	flowdata.SystemAccess/AcctRec

その結果、SystemAccess という名前の XML 要素と、AcctPay および AcctRec という名前の 2 つの子要素が作成されます。このようにしてデータを構造化する 1 つの理由は、多くのフォームとデータ項目が含まれている複雑なワークフローで、データの編成と管理をよりわかりやすくすることにあります。この階層データからデータを取得するには、次の構文を使用します：

```
flowdata.get('SystemAccess/AcctPay').
```

ECMAScript 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

フォームコントロールデータのフローデータへの移動

作成したフォームコントロール (DN の表示以外) はすべて自動的に、フォームを使用するアクティビティの [データ項目マッピング] タブの前動作、後動作の式で使用可能になります。たとえば、AACTIVITY のフォーム AFORM 上にあるコントロール ACONTROL のユーザ入力データを後続のアクティビティで利用できるようにします。そのためには、ワークフローで AACTIVITY を選択し、[データ項目マッピング] タブを選択し、[後動作マッピング] ラジオボタンをクリックします。それから、ソースフォームフィールド ACONTROL の横に、次の形式でターゲット式を入力します：

```
flowdata.my_ACONTROL
```

ワークフロー内の後続のすべてのアクティビティは、次のような前動作ソース式を使用して、このデータにアクセスできるようになります：

```
flowdata.get('my_ACONTROL')  
flowdata.getObject('my_ACONTROL')
```

フローデータのフォームコントロールへの移動

フローデータ値をフォームコントロールに移動することもできます。最もシンプルなケースは、単一のテキスト値をフォームコントロールに移動する場合です。前に示した例で、ACONTROL が 1 つのテキスト入力フィールドであるとします。この場合、このフィール

ドを ZACTIVITY というアクティビティの別のテキスト入力フィールドに移動するには、ワークフローで ZACTIVITY を選択し、[データ項目マッピング] タブを選択し、[前動作マッピング] ラジオボタンをクリックします。それから、ターゲットフォームフィールドの横に、次のソース式を入力します：

```
flowdata.my_ACONTROL
```

より複雑なフォームコントロールデータ（複数値 DN コントロールなど）を別のフォームコントロールに移動するには、`getObject()` 式構文を使用できます。たとえば、ACONTROL が MultiValue DN コントロールであれば、次のソース式を使用することができます：

```
flowdata.getObject('my_ACONTROL')
```

データをフォームコントロールに移動する場合には、タイプの制約に注意する必要があります。たとえば、テキストベースのデータを数値コントロールに移動しようとしたり、プール値を DN コントロールに移動しようしたりすべきではありません。

4.4 インストールされているテンプレートでの作業

Identity Manager には、事前設定済みのプロビジョニング要求定義とワークフローのセットが用意されています。これらをテンプレートとして使用し、独自のプロビジョニングシステムを構築することができます。システムを設定するには、インストールされているテンプレートに基づいて新しいオブジェクトを定義してから、これらのオブジェクトを組織のニーズに合うようにカスタマイズします。

インストールされているテンプレートを使用して、要求を実行するのに必要な承認ステップの数を指定できます。0 から 5 個までの承認ステップを必要とするプロビジョニング要求を設定できます。

順次処理または並行処理のどちらをサポートするか、および処理中にワークフローのタイムアウトが発生した場合に要求を承認するか拒否するかも指定できます。

Identity Manager に含まれているテンプレートを、次の表に一覧で示します。

表 4-8 事前設定済みのプロビジョニング要求定義とワークフロー

テンプレート	説明
Self Provision Approval	承認なしにプロビジョニング要求を実行できるようにします。
One Step Approval (Timeout Approves)	プロビジョニング要求の実行に、1段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。
2 ステップ順次承認（タイムアウト時に承認）	プロビジョニング要求の実行に、2段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。 このテンプレートは順次処理をサポートします。

テンプレート	説明
Three Step Sequential Approval (Timeout Approves)	<p>プロビジョニング要求の実行に、3段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは順次処理をサポートします。</p>
4ステップ順次承認（タイムアウト時に承認）	<p>プロビジョニング要求の実行に、4段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは順次処理をサポートします。</p>
5ステップ順次承認（タイムアウト時に承認）	<p>プロビジョニング要求の実行に、5段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは順次処理をサポートします。</p>
1ステップ承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、1段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p>
2ステップ順次承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、2段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは順次処理をサポートします。</p>
3ステップ順次承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、3段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは順次処理をサポートします。</p>
4ステップ順次承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、4段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは順次処理をサポートします。</p>
5ステップ順次承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、5段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは順次処理をサポートします。</p>
2ステップ並行承認（タイムアウト時に承認）	<p>プロビジョニング要求の実行に、2段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは並行処理をサポートします。</p>

テンプレート	説明
Three Step Parallel Approval (Timeout Approves)	<p>プロビジョニング要求の実行に、3段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは並行処理をサポートします。</p>
4ステップ並行承認（タイムアウト時に承認）	<p>プロビジョニング要求の実行に、4段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは並行処理をサポートします。</p>
5ステップ並行承認（タイムアウト時に承認）	<p>プロビジョニング要求の実行に、5段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、アクティビティは要求を承認し、作業項目は次のアクティビティに送られます。</p> <p>このテンプレートは並行処理をサポートします。</p>
2ステップ並行承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、2段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは並行処理をサポートします。</p>
3ステップ並行承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、3段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは並行処理をサポートします。</p>
4ステップ並行承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、4段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは並行処理をサポートします。</p>
5ステップ並行承認（タイムアウト時に拒否）	<p>プロビジョニング要求の実行に、5段階の承認を必要とします。アクティビティのタイムアウトが発生した場合、ワークフローは要求を拒否します。</p> <p>このテンプレートは並行処理をサポートします。</p>

4.5 ワークフローのデバッグ

ワークフローのテスト時には、フローに使用している変数の値を確認することが必要になる場合があります。確認する方法はいくつかあります。その1つは、ログアクティビティを使用して、確認する必要のある変数を含むメッセージを表示するという方法です。ログアクティビティの設定が完了した後に、コンソールに表示されるメッセージを確認できます。ログアクティビティでは、メッセージプロパティにスクリプト式を使用して、必要な値を取得できます。たとえば、フローデータオブジェクトに定義されている変数の値を含むメッセージをログに記録するには、次の式を使用できます：

```
flowdata.get('my_variable')
```

ログアクティビティの使用に関する詳細については、「[199 ページのセクション 8.3 「ログアクティビティ」](#)」を参照してください。

別 の方法は、ワークフローデータベースを調べて、ワークフローがアクティビティを次々と実行するにつれて、フロー データ オブジェクトに関連付けられているデータがどのように変化しているかを確認することです。[afdocument] テーブルを表示すると、このデータを確認できます。

最後の方法は、デバッグ処理中に使用できます。ワークフローシステム (com.novell.soa.af.impl)、ユーザ アプリケーションのプロビジョニング要求コンポーネント (com.novell.srpr.apwa)、サーバ側スクリプトの評価 (com.novell.soa.script) に関連付けられているログレベルを変更するというものです。この方法を実行すると必要以上の情報が生成されますが、その情報が役立つ場合もあります。ログレベルを変更するには、ユーザ アプリケーションの [管理] タブにあるログページに移動します。

プロビジョニング要求定義の作成

5

この項では、プロビジョニング要求定義の作成について説明します。主なトピックは次のとおりです。

- ◆ 97 ページのセクション 5.1 「ウィザードおよび [概要] タブについて」
- ◆ 99 ページのセクション 5.2 「ウィザードによるプロビジョニング要求定義の作成」
- ◆ 103 ページのセクション 5.3 「[概要] タブでの基本設定の変更」

5.1 ウィザードおよび [概要] タブについて

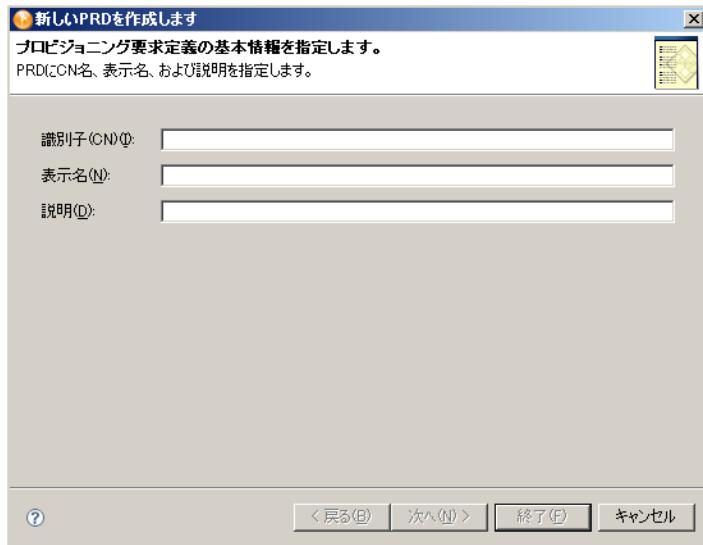
プロビジョニング要求定義は、次の 3 つの基本手順で作成します。

- ◆ 新しい PRD の作成ウィザードを使用して、プロビジョニング要求定義に関する基本情報（プロビジョニング要求定義の名前、属するカテゴリ、アクセスできるユーザなど）を作成します。プロビジョニング要求定義の基本情報を作成すると、[概要] タブに基本情報が表示されます。
- ◆ [フォーム] タブを使用して、ワークフローの参加者とやりとりするためのフォームを作成します。
- ◆ [ワークフロー] タブを使用して、ワークフローを設計します。

プロビジョニング要求定義を追加するには：

- 1 次のいずれかの方法で新しい PRD の作成ウィザードを起動します。
 - ◆ プロビジョニングビューで、[プロビジョニング要求定義] ノードを右クリックし、[新規] をクリックします。
 - ◆ プロビジョニングビューで、ユーザアプリケーションまたはプロビジョニング要求コンテナをクリックし、[挿入] [プロビジョニング要求定義] の順にクリックします。
 - ◆ [ファイル] [新規] [プロビジョニング] の順にクリックします。
 - ◆ [プロビジョニング要求定義] を選択し、[次へ] をクリックします。

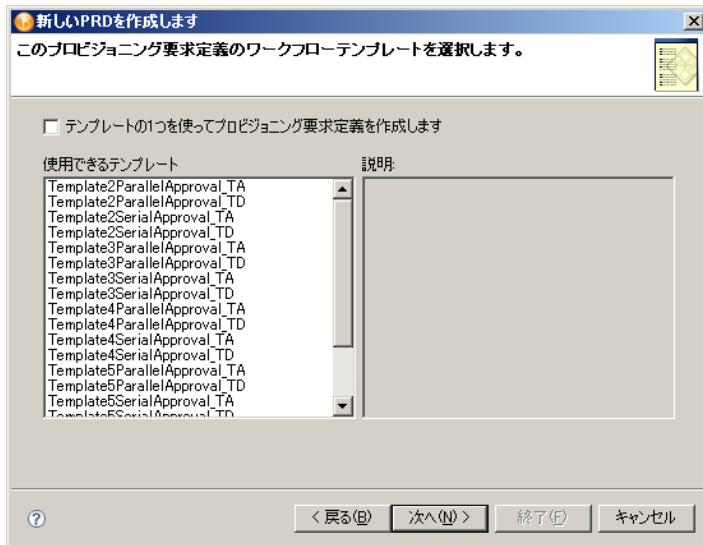
新しい PRD の作成ウィザードの最初のページが表示されます。



2 次のようにフィールドに値を入力します。

フィールド	説明
識別子 (CN)	プロビジョニング要求定義の CN (一般名) 識別子です。
表示名	プロビジョニング要求定義の表示名です。これは、プロビジョニングビューに表示される名前です。
説明	プロビジョニング要求定義の説明です。

3 [次へ] をクリックします。ウィザードの次のページが表示されます。

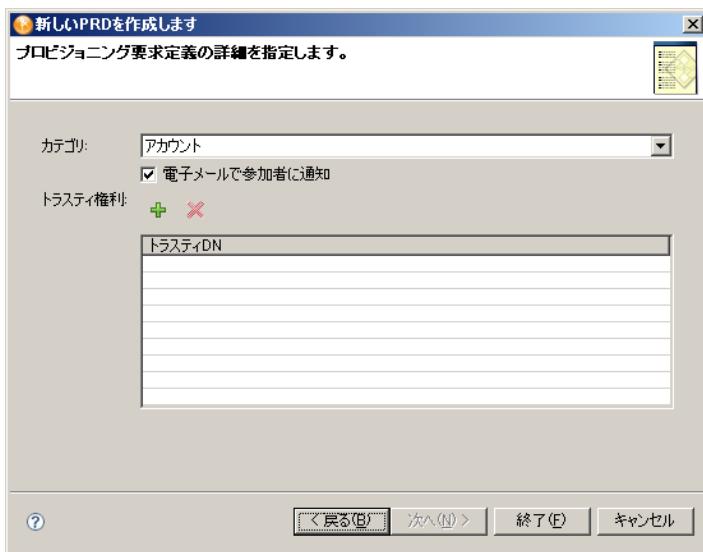


新しいプロビジョニング要求定義をテンプレートに基づいて作成できます。または、独自のプロビジョニング要求定義を一から作成することもできます。ウィザードの次のパネルを使用して、このプロビジョニング要求定義をテンプレートに基づいて作成するかどうかを指定します。

4 次のいずれかの手順を実行します。

- ◆ プロビジョニング要求定義をテンプレートに基づいて作成する場合は、[テンプレートの 1 つを使ってプロビジョニング要求定義を作成します] を選択し、[使用できるテンプレート] リストから、使用するテンプレート (TemplateSingleApproval_TA など) を選択して、[次へ] をクリックします。
- ◆ 独自のプロビジョニング要求定義を一から作成する場合は、[次へ] をクリックします。

ウィザードの次のパネルを使用して、展開後のプロビジョニング要求定義にアクセスできるトラスティ（ユーザ）を指定します。



5 トラスティを追加するには、プラス (+) アイコンをクリックします。

Designer にはトラスティを選択するために識別ポールトを参照するパネルが表示されます。個人のトラスティまたはグループを選択できます。

識別ポールトに接続できない場合は、[トランザクション DN] フィールドにトランザクション DN を直接入力できます。

6 トラスティを選択し、[OK] をクリックします。

画面は元のパネルに戻ります。別のトラスティを追加する場合は、前の手順を繰り返します。

7 トラスティの追加が完了したら、[完了] をクリックします。

[概要] タブに [プロビジョニング要求定義の詳細] パネルが表示されます（「[103 ページのセクション 5.3 「\[概要\] タブでの基本設定の変更」](#)」を参照してください）。

5.2 ウィザードによるプロビジョニング要求定義の作成

テンプレートを使用してプロビジョニング要求定義を作成するか、または独自の定義を一から作成できます。既存のテンプレートを使用して新しいプロビジョニング要求定義を作成することをお勧めします。テンプレートを使用すると時間を節約でき、既存のプロビジョニング要求定義に必要な変更だけを加えられます。ただし、実行したい新しい作業に

似た既存のプロビジョニング要求定義がない場合は、新しいプロビジョニング要求定義を一から作成できます。

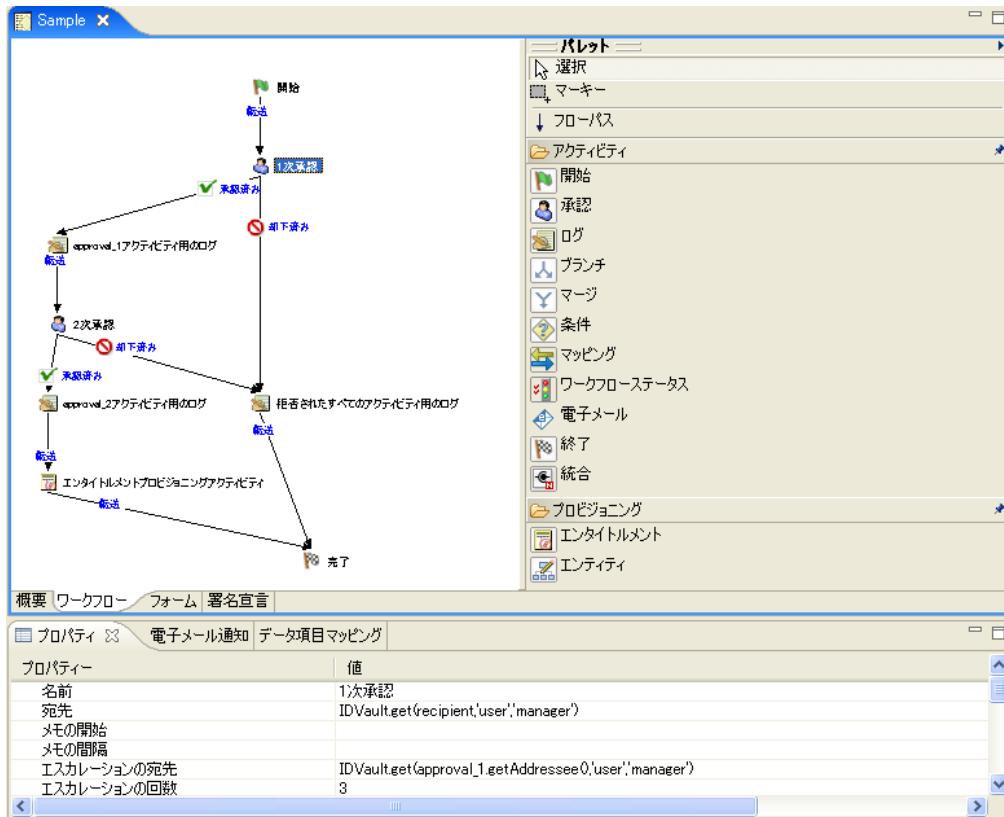
ここでは次のトピックについて説明します。

- ◆ 100 ページのセクション 5.2.1 「テンプレートの使用」
- ◆ 102 ページのセクション 5.2.2 「独自のプロビジョニング要求定義の作成」

5.2.1 テンプレートの使用

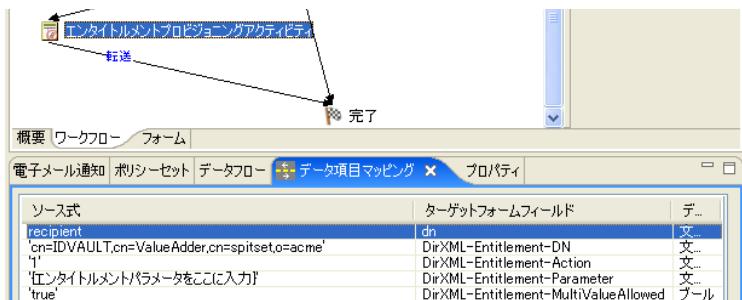
テンプレートを使用してプロビジョニング要求定義を作成するには：

- 1 新しいプロビジョニング要求定義の基本情報を作成します（「97 ページのセクション 5.1 「ウィザードおよび [概要] タブについて」」を参照してください）。手順 99 ページのステップ 4 で、[テンプレートの 1 つを使ってプロビジョニング要求定義を作成します] を選択し、使用するテンプレートを選択します。完了すると、新しいプロビジョニング要求定義の [概要] タブが表示されます。
- 2 [ワークフロー] タブをクリックします。ワークフロービューが表示されます。



プロビジョニング要求定義テンプレートに含まれているデフォルト値のいくつかを、環境に応じてカスタマイズする必要があります。たとえば、「エンタイトルメントプロビジョニングアクティビティ」には、いくつかのデータ項目マッピングプロパティのプレースホルダ値が含まれています。それらのプレースホルダ値を、このプロビジョニング要求の実際の値に置き換える必要があります。

- 3 [エンタイトルメントプロビジョニングアクティビティ] をクリックし、[データ項目マッピング] タブをクリックします。



- 4 [ターゲット] フィールドにある *[DirXML-Entitlement-DN]* の [ソース式] フィールド内をダブルクリックし、フィールド内に表示されたボタンをクリックして ECMA Expression Builder を表示します。
ECMA Expression Builder については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。
- 5 ECMA Expression Builder を使用して、プレースホルダ式を、このプロビジョニング要求でプロビジョニングするエンタイトルメントを指定する式に置き換えます。
- 6 *[DirXML-Entitlement-Parameter]* の [ソース式] フィールドのプレースホルダ式を置き換えます。
- 7 [フォーム] タブをクリックし、プロビジョニング要求のフォームをニーズに合わせてカスタマイズします。
[「107 ページの第 6 章「プロビジョニング要求定義のフォームの作成」」](#) を参照してください。テンプレートには、事前定義された要求フォームと承認フォームが含まれています。別のフォームを追加したり、フォームコントロールを追加または削除したりできます。
- 8 [ワークフロー] タブをクリックし、ワークフローのプロパティをニーズに合わせてカスタマイズします。
[「163 ページの第 7 章「プロビジョニング要求定義のワークフローの作成」」](#) および [「187 ページの第 8 章「ワークフローアクティビティの参照」」](#) を参照してください。

テンプレートとフロー戦略

デフォルトでは、プロビジョニング要求定義テンプレートのフロー戦略には *[単一のフロー]* が使用されます。テンプレートでは、受信者はユーザ DN であると想定されます。フロー戦略を変更する場合は、テンプレートを変更する必要があります。フロー戦略を変更する場合、次のように操作します。

- *[メンバごとのフロー]* に変更する場合は、要求フォームから受信者の参照を削除します。
- *[単一のフロープロビジョニングメンバ]* に変更する場合は、要求フォームから受信者を削除し、1 次承認アクティビティの宛先を決定するロジックを追加します。テンプレートでは受信者はユーザ DN であると想定されるので、受信者がユーザ DN かグループ DN かを確認する必要があります。DN がグループであるかどうかを確認するには、`IDVault isGroup (String dn)` メソッドを使用できます。受信者がグループ DN である場合、承認アクティビティの宛先を割り当てるロジックを提供する必要があります。

5.2.2 独自のプロビジョニング要求定義の作成

可能な場合は、既存のテンプレートを使用して（または既存のプロビジョニング要求定義を新しい名前で保存して）、新しいプロビジョニング要求定義を作成してください。テンプレートを使用すると時間を節約でき、既存のプロビジョニング要求定義に必要な変更だけを加えられます。ただし、実行する新しい作業に似た既存のプロビジョニング要求定義がない場合は、独自のプロビジョニング要求定義を一から作成する必要があります。その場合でも、他のワークフローのフォームを再利用すれば、時間と労力を節約できます。

プロビジョニング要求定義を作成するには：

- 1 新しいプロビジョニング要求定義の基本情報を作成します（「[97 ページのセクション 5.1 「ウィザードおよび \[概要\] タブについて」](#)」を参照してください）。手順 [99 ページのステップ 4](#) で、[テンプレートの 1 つを使ってプロビジョニング要求定義を作成します] を選択しません。テンプレートも選択しないでください。完了すると、新しいプロビジョニング要求の [概要] タブが表示されます。
- 2 プロビジョニング要求定義のフォームを作成します。プロビジョニング要求定義の基本情報を作成したら、次に、プロビジョニング要求ユーザに表示するフォームを作成します。ワークフロートポロジを作成する前にフォームを定義することが重要です。そうすれば、アクティビティの作成時に、各アクティビティのデータバインディングが自動的に設定されます。

フォームには、次の 2 つのタイプがあります。

要求：リソースを要求するユーザが、要求する項目や機能を指定するために使用します。ワークフローごとに 1 つの要求フォームを定義できます。要求フォームは常に、開始アクティビティと関連付けられます。

承認：プロビジョニング要求を受信するユーザが、プロビジョニング要求に対して承認、拒否、またはコメントするために使用します。1 つ以上の承認フォームを定義できます。フォームごとに承認アクティビティを関連付ける必要があります。承認アクティビティのプロパティを使用して、承認フォームと承認アクティビティを関連付けます。

フォームを作成するには、「[115 ページのセクション 6.3 「フォームの作成」](#)」を参照してください。

- 3 [ワークフロー] タブをクリックし、ワークフロートポロジを作成します。
ワークフロートポロジを作成するには、アクティビティを作成して目的のワークフローパターンに関連付け、アクティビティ間のフローパスにルールを割り当てます。ワークフロートポロジの作成については、「[163 ページの第 7 章「プロビジョニング要求定義のワークフローの作成」](#)」を参照してください。
- 4 ワークフローアクティビティごとに詳細（プロパティ、データ項目マッピング、電子メール通知など）を指定します。
ワークフローアクティビティの詳細を指定するには、「[187 ページの第 8 章「ワークフローアクティビティの参照」](#)」を参照してください。
- 5 ワークフローアクティビティ間のフローパスを設定します。
フローパスを設定するには、「[172 ページのセクション 7.4 「フローパスの設定」](#)」を参照してください。

5.3 [概要] タブでの基本設定の変更

[概要] タブを使用して、プロビジョニング要求定義に関する基本情報（プロビジョニング要求定義の名前、属するカテゴリ、アクセスできるユーザなど）を定義します。

図5-1 [概要] タブ



次の表で、[概要] タブで設定できるプロパティについて説明しています。

表5-1 概要プロパティ

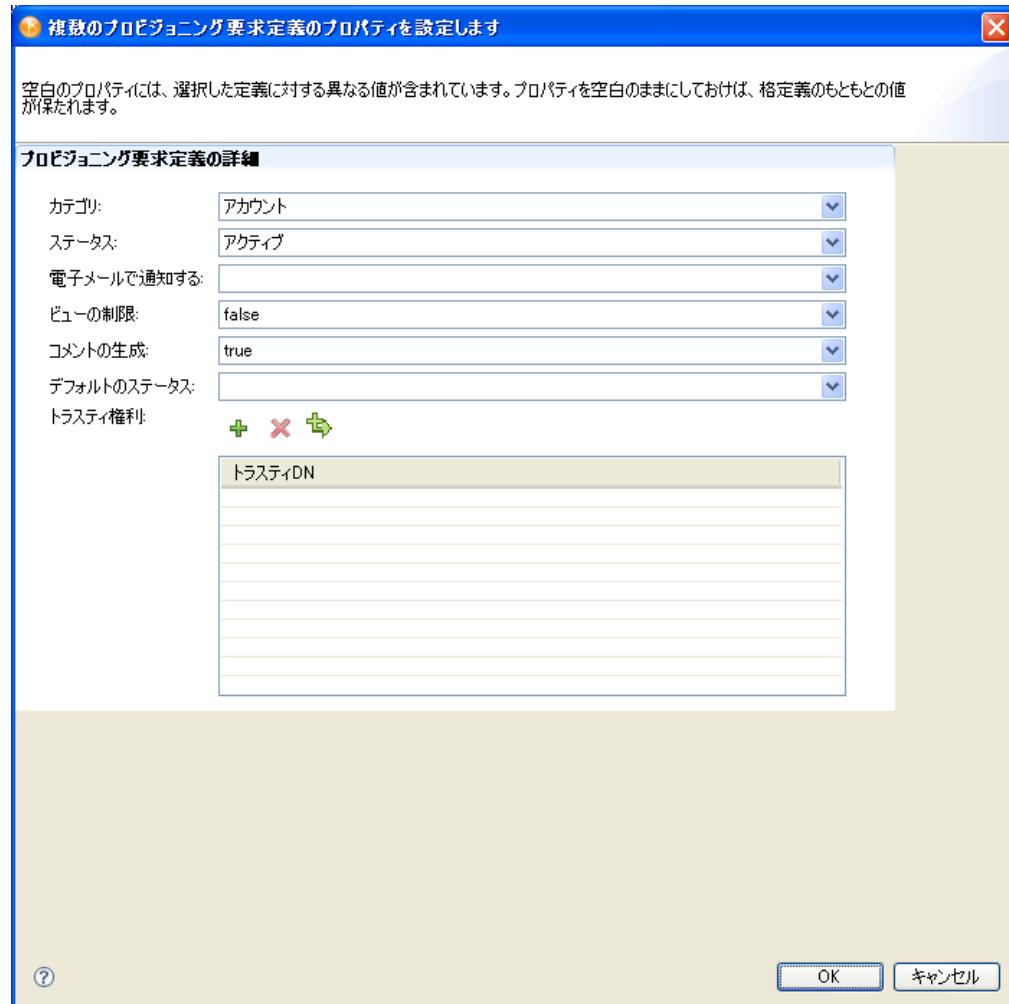
フィールド	説明
識別子 (CN)	プロビジョニング要求定義の CN (一般名) が表示されます。CN は変更できません。
表示名	プロビジョニング要求定義の表示名を指定します。これは、Designer および Identity Manager でユーザに表示される名前です。
説明	プロビジョニング要求定義の説明を指定します。
カテゴリ	ディレクトリ抽象化レイヤで定義されたプロビジョニングカテゴリのリストから、プロビジョニング要求定義が属するカテゴリを指定します。プロビジョニングビューでは、プロビジョニング要求定義がカテゴリ別に表示されます。

フィールド	説明
ステータス	<p>プロビジョニング要求定義のステータスを指定します。</p> <p>アクティブ: プロビジョニング要求定義をユーザアプリケーションで使用できるようにする場合は、このオプションを選択します。</p> <p>Inactive: プロビジョニング要求定義をユーザアプリケーションで一時的に使用できないようにする場合は、このオプションを選択します。プロビジョニング要求定義を開発および展開するユーザの役割を、プロビジョニング要求定義をアクティビ化するユーザから切り離しておきたい場合に、このオプションを使用できます。たとえば、開発者は [非アクティブ] のステータスでプロビジョニング要求定義を展開し、管理者が [アクティブ] へのステータス変更を担当します。</p> <p>テンプレート: このプロビジョニング要求定義を、他のプロビジョニング要求定義のベースとして使用する場合は、このオプションを選択します。テンプレートは、ユーザアプリケーションでは使用できません。</p> <p>Retired: プロビジョニング要求定義をユーザアプリケーションで今後使用できないようにする場合は、このオプションを選択します（その後も、プロビジョニング要求定義のステータスはいつでも変更できます）。このステータスは、使用しなくなったプロビジョニング要求定義の履歴レコードを保持するために用意されています。</p>
フロー戦略	<p>プロビジョニング要求定義のフロー戦略を指定します。</p> <p>単一のフロー: この戦略では、受信者ごとに 1 つのワークフローを使用できます。</p> <p>メンバごとのフロー: この戦略では、受信者をグループ DN にできます。この戦略を選択すると、ユーザアプリケーションでは、グループのメンバーごとにワークフローインスタンスが開始されます。各ワークフローは個別に承認または拒否できます。たとえば、Human Resources という受信者のプロビジョニング要求定義があるとします。Human Resources グループのメンバーは ablake と kchester です。ユーザアプリケーションにより、Human ResourcesDN のプロビジョニングが開始されます。プロビジョニングインターフェースでは、ablake 用と kchester 用に、2 つのワークフローインスタンスが開始されます。</p> <p>単一のフロープロビジョニングメンバ: この戦略では、受信者をグループ DN にできます。この戦略を選択すると、ユーザアプリケーションでは、グループに対して 1 つのワークフローが開始されます。1 つのワークフロー内で、複数のプロビジョニングステップ（メンバーごとに 1 つ）が生成されます。このワークフローは、個々のメンバー別ではなく、グループ全体として承認または拒否されます。</p>
電子メールで参加者に通知	<p>保留中の承認タスクについて承認者に電子メールで通知するかどうか、およびワークフローの完了をイニシエータに電子メールで通知するかどうかを指定します。[電子メールによる参加者への通知] チェックボックスがオンになっていない場合、ユーザは、タスクに関する通知を確認するために、ユーザアプリケーションで [要求および承認] タブを確認する必要があります。</p> <p>電子メールテンプレートの選択および電子メールテンプレートパラメータのカスタマイズについては、「207 ページのセクション 8.10 「完了アクティビティ」」を参照してください。</p>
ビューの制限	<p>ユーザアプリケーションの [マイ要求] でユーザが表示できるタスクのリストを、ユーザ自身が開始したタスクのみに制限します。デフォルトの動作 ([ビューの制限] が選択されていない) では、ユーザは、自分が開始した要求や、自分が受信者になっている要求をすべて表示できます。</p>

フィールド	説明
コメントの生成	ワークフローエンジンがワークフローの進捗に応じてコメントを生成するように指定します。このコメントを表示するには、次のいずれかをクリックします。
	<ul style="list-style-type: none"> ◆ ユーザアプリケーション内の [要求詳細] フォームの [コメントとフロー履歴の参照] ◆ ユーザアプリケーション内の [タスク詳細] フォームの [コメント履歴の表示]
デフォルトの完了ステータスを「承認」に設定	選択した場合、プロビジョニング要求のデフォルトの完了ステータスは「承認」に、選択しない場合は「拒否」に設定されます。この機能は、プロビジョニングアクティビティ（エンタイトルメントまたはエンティティ）が含まれていないプロビジョニング要求で役に立ちます。このパラメータの値は、プロビジョニングアクティビティまたはワークフローステータスアクティビティを使用して完了ステータスを明示的に設定することによって上書きできます。
トラスティ権利	このプロビジョニング要求定義を使用できるユーザまたはグループを指定します。

これらのアクセスプロパティを、1つ以上のプロビジョニング要求定義の [概要] パネルで使用できるように設定できます。

- 1 プロビジョニングビューが開いた状態で、1つ以上のプロビジョニング要求を選択し、右クリックして [プロパティ] を選択します。次の画面が表示されます。



- 2 選択したすべてのプロビジョニング要求定義でプロパティを変更するには、そのプロパティを変更します。
- 3 プロパティの 1 つに値が含まれていない場合、対応するフィールドは空白になります。そのフィールドを変更する場合、選択した値は、選択したすべてのプロビジョニング要求定義に適用されます。
- 4 トラスティについては、選択したプロビジョニング要求定義の値を削除、追加、またはマージできます。値をマージするには、<> を選択します。この操作により、選択したプロビジョニング要求定義のすべてのトラスティが、選択したすべてのプロビジョニング要求定義に追加されます。

プロビジョニング要求定義のフォームの作成

6

この項では、ユーザアプリケーションの要求および承認フォームの作成とカスタマイズについて詳しく説明します。主なトピックは次のとおりです。

- ◆ 107 ページのセクション 6.1 「フォームについて」
- ◆ 110 ページのセクション 6.2 「[フォーム] タブについて」
- ◆ 115 ページのセクション 6.3 「フォームの作成」
- ◆ 123 ページのセクション 6.4 「アクションのリファレンス」
- ◆ 125 ページのセクション 6.5 「フォームコントロールのリファレンス」
- ◆ 151 ページのセクション 6.6 「識別名の操作」
- ◆ 154 ページのセクション 6.7 「フォームでの DAL クエリの使用」
- ◆ 159 ページのセクション 6.8 「フォームの印刷」
- ◆ 160 ページのセクション 6.9 「フォームへの直接アクセス」

6.1 フォームについて

ユーザはフォームを使用して、リソースを要求し、リソース要求を承認し、タスクを処理できます。フォームは、ユーザがユーザアプリケーションの「要求および承認」タブの「マイ作業」カテゴリでアクションを選択するときに使用できます。URL から直接使用することもできます。この手法の詳細については、「160 ページのセクション 6.9 「フォームへの直接アクセス」」を参照してください。

図 6-1 はリソース要求フォームの一例です。フォームの上部には、要求の詳細を表示した読み込み専用領域があります（承認フォームでは承認の詳細）。下部にある「[フォームの詳細]」セクションで、ユーザはリソース要求（または承認）に情報を入力し、何らかのアクションを行います。

図6-1 フォームのサンプル

The screenshot shows the Novell Identity Manager interface. The top navigation bar includes 'Novell® Identity Manager', 'Welcome, Allison', 'Identityセルフサービス', '要求と承認' (selected), 'Logout' (ログアウト), and 'Help' (ヘルプ). A red 'N' logo is in the top right corner.

The left sidebar has sections: 'マイ作業' (My Work), 'マイタスク' (My Tasks) with a sub-item '・要求リソース' (Resource Request), 'マイ要求' (My Requests), 'マイ設定' (My Settings) with sub-items '代理人モードの開始' (Start Agent Mode), '移動可能状況の設定' (Set Mobility Status), 'マイ代理人割り当て' (Assignments to Me), and 'マイ委任割り当て' (Delegations to Me).

The main content area has a '要求リソース' (Resource Request) section with the message: 'ステップ3/3:リソース要求を確認し、終了します。* - は入力必須です。' (Step 3/3: Confirm resource request, finish. * - is required input). It lists: 'リソース: タイトル変更要求', '受信者: Allison Blake', 'リソースカテゴリー: アカウント', and '説明: ユーザがタイトル変更を要求することを許可します。マネージャの承認が必要です。' (User allows title change. Manager approval is required).

A 'フォームの詳細' (Form Details) section contains a sub-section '1ステップ承認(タイムアウト時に承認)' (One-step approval (Approval at timeout)) with fields: '受信者: Allison Blake', '現在のタイトル: 新しいタイトル', '新しいタイトルの要求:', '要求の理由: *' (Reason for request: *), and buttons '送信' (Send) and 'キャンセル' (Cancel).

プロビジョニング要求定義エディタで [フォーム] タブを使用して、ユーザアプリケーションの要求フォームと承認フォームにある [フォームの詳細] セクションの見た目と動作を定義します。

要求フォームについて

プロビジョニング要求定義の要求フォームを1つ作成できます。要求フォームは、ワークフローの開始アクティビティに関連付けられています。

図6-2 リソース要求フォームのサンプル

This screenshot is identical to Figure 6-1, showing the 'Resource Request' form for title change. The layout, fields, and overall structure are the same, demonstrating how the form appears to users.

承認フォームについて

プロビジョニング要求定義の承認フォームは複数定義できますが、承認アクティビティごとに1つのフォームに限られます。承認アクティビティに関連付ける承認フォームは、承認アクティビティのプロパティで指定できます。[フォーム] タブまたは承認アクティビティプロパティシートから承認フォームを作成できます。

図6-3 リソース承認フォームのサンプル

リソース: タイトル変更要求 受信者: Allison Blake
要求者: Allison Blake タスク: 単一承認
次の日時以降 タイムアウト 日時:
キューブにある 06/30/2006 12:32:18 PM 03/17/2026 12:32:18 PM
割り当て先: Margo MacKenzie 引き受け者:

単一承認
該当するボタンを選択して要求を承認または却下してください。

要求者:	Allison Blake	受信者:	Allison Blake
要求日:	06/30/2006		
理由:	テスト		
現在の従業員タイトル:	新しいタイトル		
タイトル変更の要求:	マネジングアシスタント		

コメント:

6.1.1 フォームコントロールのデータバインディングについて

フォームに対して定義するフィールドはすべて、[データ項目マッピング] プロパティシートでのデータバインディングに自動的に使用できるようになります。フォームフィールドを初期化またはデータをプリロードする前動作マッピングと、変更したフォームデータをフローデータというワークフロードキュメントに移動する後動作マッピングの2つのバインディング（つまりマッピング）を、フォームフィールドごとに行えます。これらのデータ項目バインディングとバインディングに使用するスクリプト式は、クライアントブラウザへ送信してユーザに表示する前に、フォームの準備作業としてアプリケーションサーバ上で実行します。前動作データ項目マッピングとフローデータドキュメントを処理する式は、通常、前の承認データを現在の承認に移動したり、フィールドのデフォルト値を設定するために使用されます。データ項目マッピングの詳細については、「[169 ページのセクション7.2.2 「データ項目マッピングの定義」](#)」を参照してください。

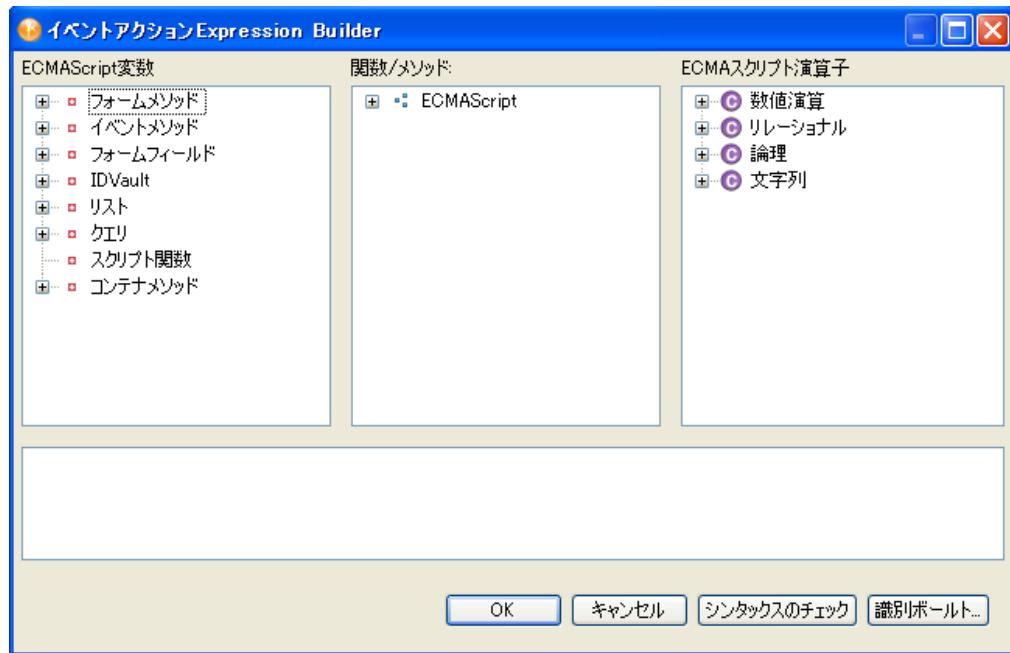
一部のフォームコントロールを使用して、ワークフローデータ以外のデータソースから値を初期化できます。たとえば、一部のリストコントロールでは、コントロールのプロパティとして初期値を指定できます。初期値の定義の詳細については、「[125 ページのセクション6.5 「フォームコントロールのリファレンス」](#)」を参照してください。

6.1.2 フォームおよびイベントについて

Designer では、フォームコントロールの `onLoad` イベント、`onChange` イベント、またはカスタムイベントで実行するアクションスクリプトを定義できます。各フォームコントロールは [イベント] プロパティをサポートしており、このプロパティでイベントのスクリプトを指定します。定義するスクリプトはイベントレベルスコープを持ち、ユーザのクライアントコンピュータのブラウザで実行します。

[イベント] プロパティから、フォームとデータを参照し変更するスクリプト式を作成できる、Designer のイベントアクション Expression Builder にアクセスできます。フォームコントロールイベントスクリプトはクライアントブラウザで実行するので、フローデータドキュメントにアクセスできません。ディレクトリ抽象化レイヤクエリにはアクセスできます。

図 6-4 イベントアクション Expression Builder

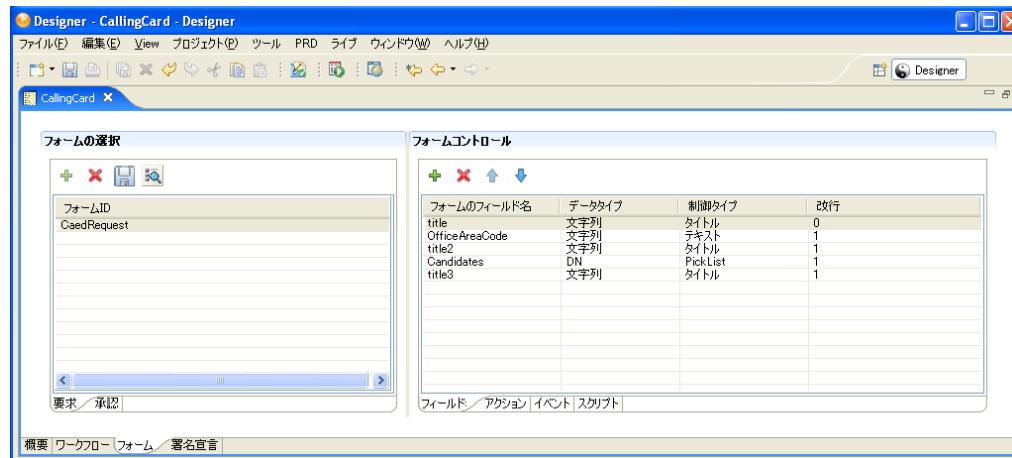


イベントアクション Expression Builder は、フォームアクションメソッドにもアクセスできます（左列を参照）。この列からは、ディレクトリ抽象化レイヤクエリオブジェクトとともにフォームアクションスクリプト API にもアクセスできます。フォームアクションスクリプト API は JavaScript* で作成されているので、条件、ループ、およびユーザ定義関数を追加できます。フォームアクション API の詳細については、「[282 ページのセクション 10.3.1 「フォームアクションスクリプトメソッド」](#)」を参照してください。JavaScript ライブラリをインポートまたは組み込むには、[フォームコントロール] エリアの [スクリプト] タブを使用します。詳細については、[121 ページのセクション 6.3.4 「\[スクリプト\] タブの使用」](#) を参照してください。

6.2 [フォーム] タブについて

要求フォームと承認フォームの見た目と動作を定義するには、プロビジョニング要求定義エディタの [フォーム] タブを使用します。

図6-5 [フォーム] タブ



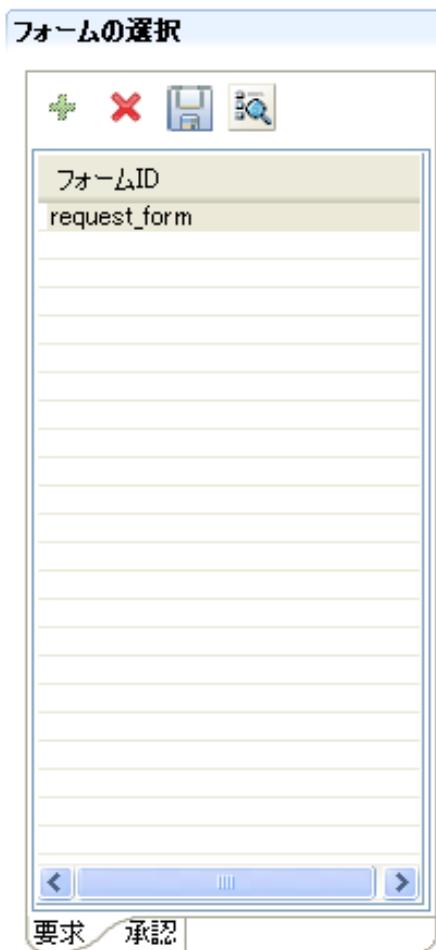
[フォーム] タブには、[フォームの選択] セクション ([フォームの選択] についてで説明) と [フォームコントロール] セクション ([フォームコントロール] についてで説明) があります。

- ◆ 111 ページのセクション 6.2.1 「[フォームの選択] について」
- ◆ 113 ページのセクション 6.2.2 「[フォームコントロール] について」

6.2.1 [フォームの選択] について

フォームを作成、削除、またはプレビューしたり、フォームテンプレートを作成したりするには、[フォームの選択] セクションを使用します。操作するフォームのタイプに応じて、[要求] タブか [承認] タブをクリックします。

図6-6 フォームの選択



[フォームの選択] ツールバーには次のオプションがあります。

表6-1 [フォームの選択] ツールバーのオプション

ボタン	説明
	クリックすると、[新規フォーム] ウィザードが起動します。
	クリックすると、既存のフォームが削除されます。
	クリックすると、テンプレートとしてフォームが保存されます。保存しておくと、このテンプレートに基づいて他のフォームを作成できます。フォームは XML ドキュメントとしてプロジェクトディレクトリに保存されます。 テンプレートは、テンプレートを作成したプロジェクト内でのみ使用できます。

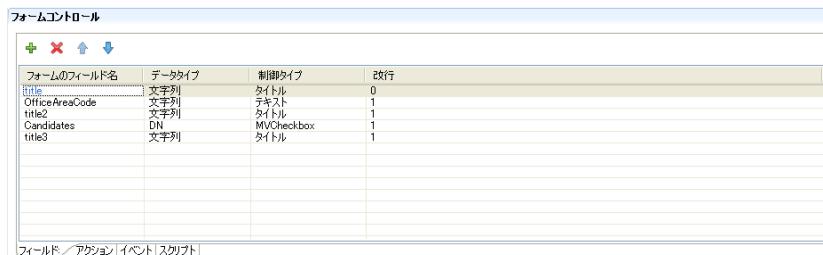
ボタン	説明
	クリックすると、フォームがプレビューされます。

既存のテンプレートからプロビジョニング要求定義を作成する場合、テンプレートにフォームが関連付けられていれば、[フォームコントロール] セクションにそのフォームが表示されます。[フォームコントロール] セクションを使用して、フォームインスタンスを変更できます。

6.2.2 [フォームコントロール] について

フォームの見た目と動作を定義または変更するには、[フォームコントロール] セクションを使用します。

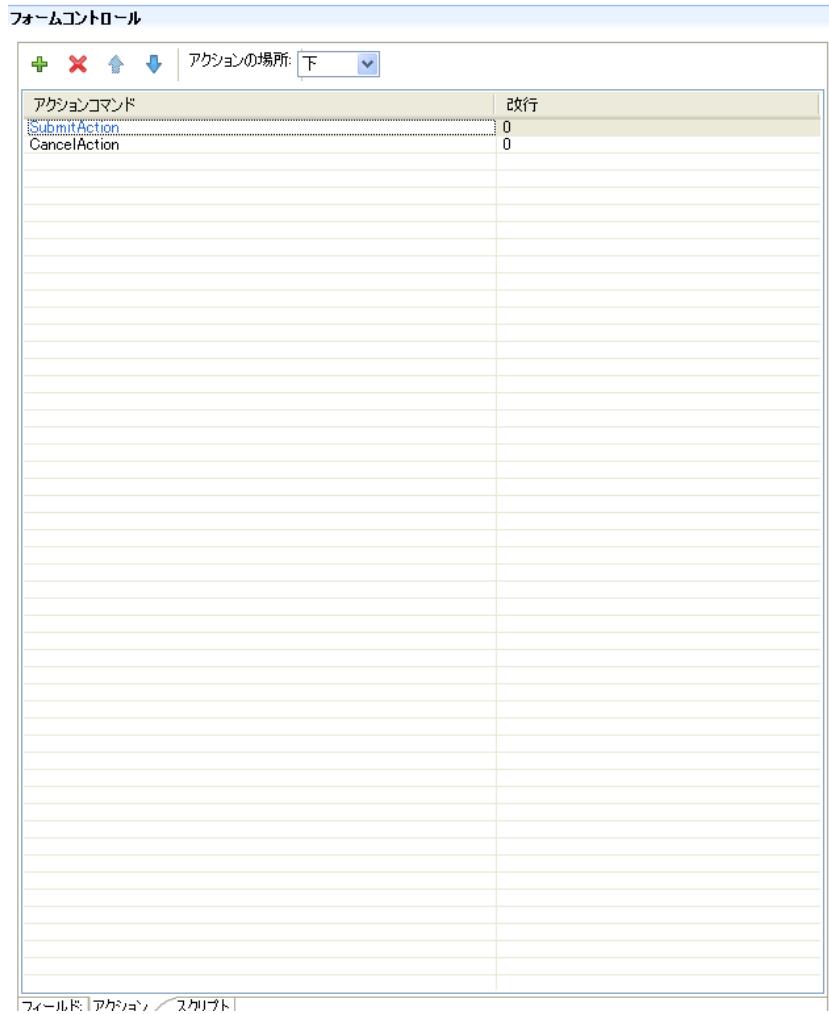
- ◆ [フィールド] タブ：データタイプ、コントロールタイプ、およびフォームでのコントロールのレイアウト順序を追加、削除、および変更できます。



フォームのフィールド名	データタイプ	制御タイプ	改行
Title	文字列	タイトル	0
OfficeAreaCode	文字列	テキスト	1
title2	文字列	タイトル	1
Candidates	DN	MVCheckbox	1
title3	文字列	タイトル	1

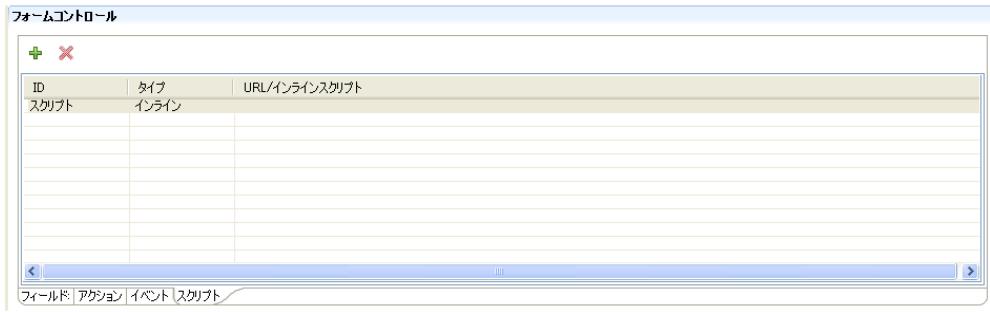
コントロールの追加の詳細については、「[115 ページのセクション 6.3 「フォームの作成」](#)」を参照してください。個別のフォームコントロールの詳細については、「[125 ページのセクション 6.5 「フォームコントロールのリファレンス」](#)」を参照してください。

- ◆ [アクション] タブ：ユーザがフォームで実行できるアクションを定義できます。[アクション] ツールバーを使用して、アクションとフォームでのアクションのレイアウト順序を追加、削除、および変更します。

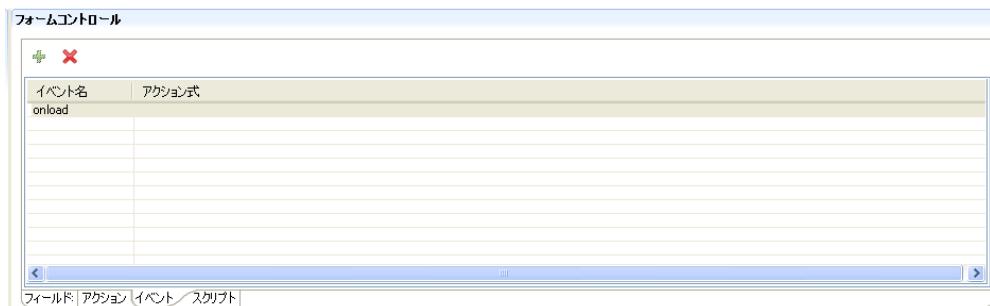


サポートするアクションの詳細については、「[123 ページのセクション 6.4 「アクションのリファレンス」](#)」を参照してください。

- ・[スクリプト] タブ：[スクリプト] タブを使用して、外部 JavaScript ファイルへのコールを定義したり、フォーム定義の一部として格納される JavaScript スクリプトを作成します。このタブを使用してスクリプトを作成した場合、このスクリプトはアクションスクリプト *Expression Builder* で使用できるようになり、どのフォームコントロールイベントからでもコールできます。これらのスクリプトは、イベントスコープではなくページスコープを持ちます。[スクリプト] タブの使用の詳細については「[121 ページのセクション 6.3.4 「\[スクリプト\] タブの使用」](#)」を参照してください。



- ◆ イベント



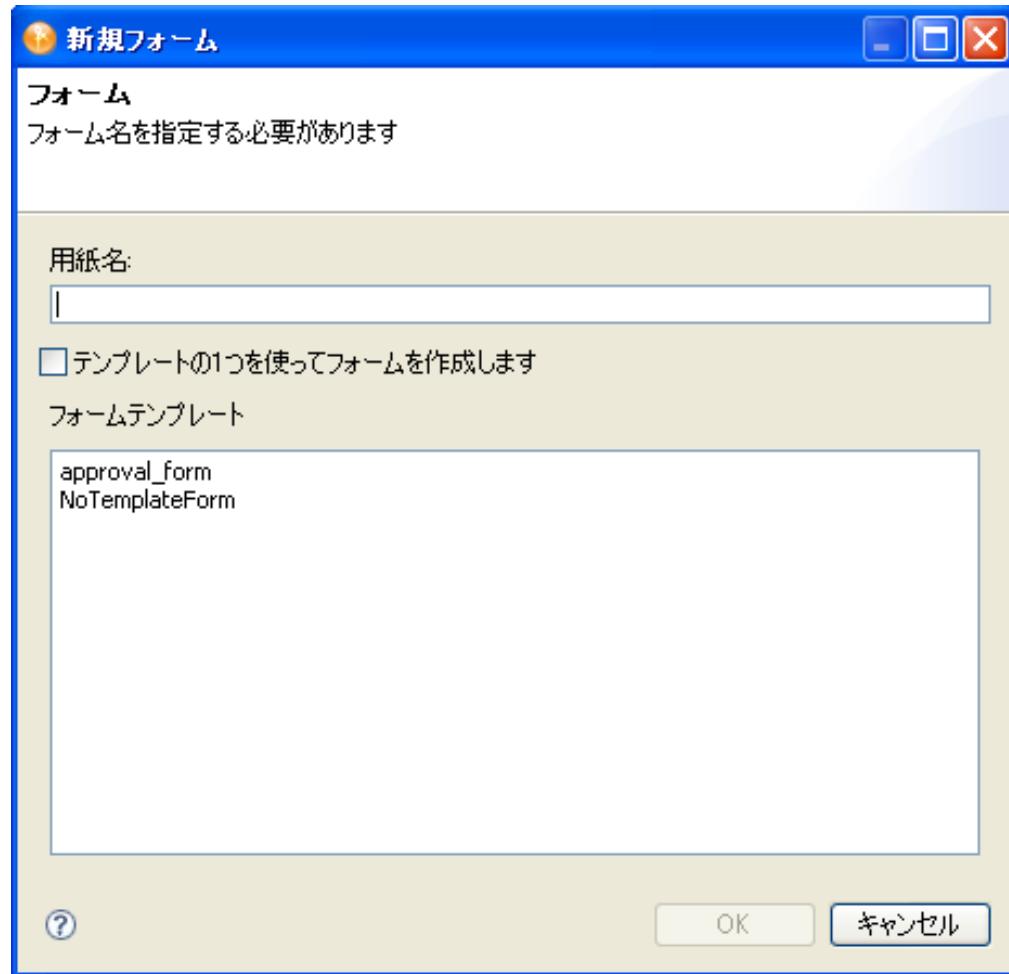
6.3 フォームの作成

この項では、新しいフォームを作成し、コントロールをそのフォームに追加する方法について説明します。次のトピックについて説明します。

- ◆ 115 ページのセクション 6.3.1 「新しいフォームの作成」
- ◆ 116 ページのセクション 6.3.2 「フォームコントロールとアクションの追加」
- ◆ 118 ページのセクション 6.3.3 「イベントの定義」
- ◆ 121 ページのセクション 6.3.4 「[スクリプト] タブの使用」

6.3.1 新しいフォームの作成

- 1 プロビジョニング要求定義エディタを開いて、[フォーム] タブをクリックします。
- 2 ページの [フォームの選択] セクションで [追加] をクリックして、[新規フォーム] ウィザードにアクセスします。



3 次のようにフィールドに値を入力します。

フィールド	説明
フォーム名	Designer で表示するフォームの名前を入力します。
テンプレートの1つを使ってフォームを作成します	既存のテンプレートに基づいて新しいフォームを作成する場合は、このオプションを選択し、フォームテンプレートリストからいずれかのフォームを選択します。

4 [OK] をクリックしてフォームを保存するか、[キャンセル] をクリックして保存せずに終了します。

6.3.2 フォームコントロールとアクションの追加

[フォームコントロール] セクションを使用して、フォームのコンテンツとレイアウトを定義します。

注：Designer は、フォームの上から下、左から右へフォームを配置します。コントロール間で間隔を空けるには、改行を使用します。

コントロールをフォームに追加する

- 1 [追加] をクリックします。Designer は、Field というコントロールをフォームの最下部に追加します。
複数のコントロールを同じ名前でフォームに追加した場合、コントロール名の最後に一意の番号が Designer によって追加されます。
- 2 コントロールの次のプロパティを定義します。

フィールド	説明
名前	フォームのフィールド フィールドの固有名です。この名前は次の場所で使用されます。
ド名	<ul style="list-style-type: none">◆ [ワークフロー] タブの [データ項目マッピング] ダイアログボックス。◆ [ECMA expression builder] ダイアログボックス。◆ プロビジョニング要求定義ファイル内の内部 XML リファレンス。
データマッピング	[データ項目マッピング] ダイアログボックスと [ECMA expression builder] ダイアログボックスで混同しないように、フォームフィールドに使用する命名規則を考慮してください。たとえば、要求フォームと承認フォームの両方に、Reason というフィールドが含まれているとします。データマッピングを実行しているときに作業しているフィールドを明確にするために、使用しているフォームの名前をフィールド名の先頭に付けることができます。一方の Reason フィールドの名前を Req_Reason、もう一方のフィールドの名前を App_Reason することができます。
データタイプ	フィールドのデータタイプです。データタイプにより、有効なコントロールタイプと実行する検証のタイプが決まります。
Control Type	データの表示または編集に使用する視覚コントロールのタイプです。選択リストの項目は、選択したデータタイプに基づいてフィルタされます。
改行	コントロールの後に挿入する行数を定義します。

フォームフィールドコントロールには、[データ項目マッピング] プロパティシートも [電子メール通知] プロパティシートもありません。

- 3 コントロールごとに、[プロパティ] タブ ([ウィンドウ] > [ビューの表示] > [プロパティ] の順に選択) でそれぞれのプロパティを指定します。詳細については、[125 ページのセクション 6.5 「フォームコントロールのリファレンス」](#) を参照してください。
- 4 [アクション] タブをクリックして、ユーザがフォームで行える作業を定義します。たとえば、ユーザがフォームを送信したりキャンセルしたりできるようにするアクションを追加できます。
要求フォームには最低限、SubmitAction が必要になります。SubmitAction がない場合、要求は処理されません。CancelAction もすべてのフォームに設定することをお勧めします。各承認フォームには、最低 1 つのアクションを定義している必要があります。
- 5 [アクション] ページで、[追加] をクリックして、新しいアクションを追加します。次のようにフィールドに値を入力します。

フィールド	説明
アクションの場所	フォームに追加するアクションボタンの場所を選択します。 下フォームの最下部にアクションボタンを配置します。(デフォルト.) [上] : フォームの最上部にアクションボタンを配置します。 [上と下] : フォームの最上部と最下部にボタンを配置します。
アクションコマンド	ボタンのアクションを選択します。 詳細については、 123 ページのセクション 6.4 「アクションのリファレンス」 を参照してください。
改行	アクションボタンの後に挿入する行数を定義します。

6 フォームを保存します。

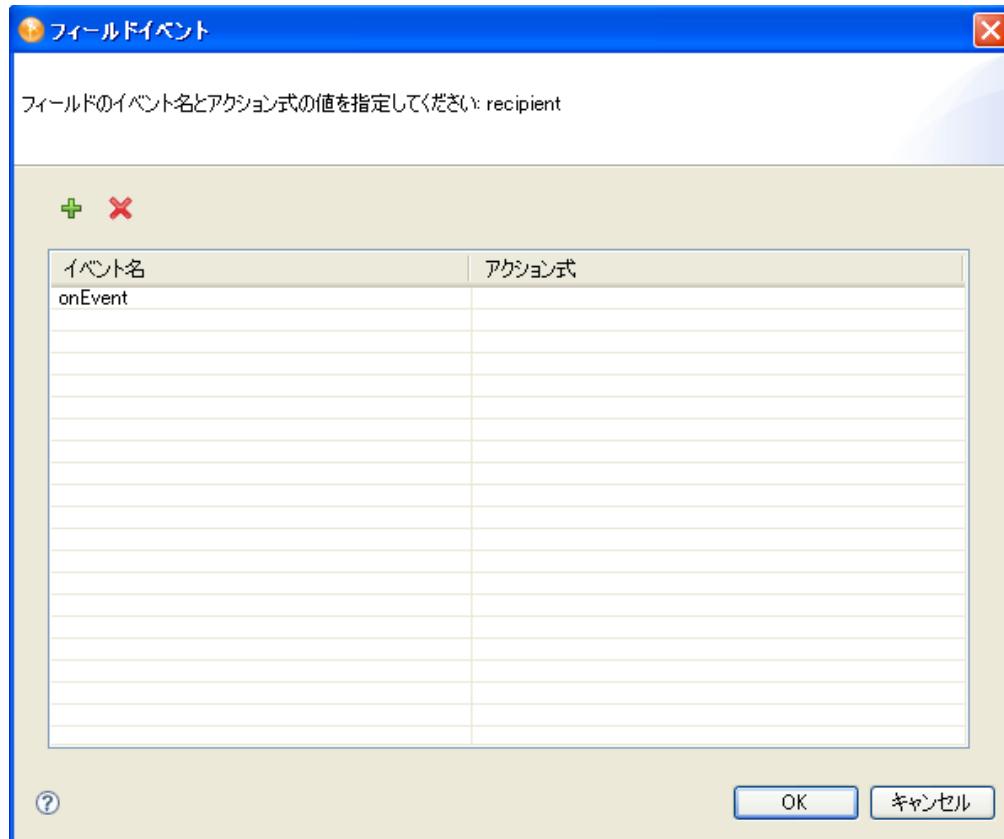
6.3.3 イベントの定義

イベントハンドラに添付するスクリプトは、ブラウザウィンドウではなく適切なコントロールにスコープされます。

- ◆ [118 ページの「イベントの定義」](#)
- ◆ [119 ページの「カスタムイベントの作成」](#)

イベントの定義

- 1 イベントを定義しプロパティシートを開くフォームコントロールを選択します。
- 2 イベントプロパティに移動して、イベントを追加します。 Designer により、*onEvent* というデフォルトイベント名で行が追加されます。



- 3 [イベント名] フィールドをクリックし、*onchange* または *onload* イベントを選択します。他のイベントを追加する方法について詳しくは、「119 ページの「カスタムイベントの作成」」を参照してください。
- 4 [アクション式] フィールドをクリックします。このフィールドにスクリプトを直接入力することも、ボタンをクリックしてイベントアクション *Expression Builder* にアクセスすることもできます。
- 5 アクションスクリプトを定義し、構文を確認して、[OK] をクリックします。この手順を繰り返して、イベントをこのコントロールに追加します。

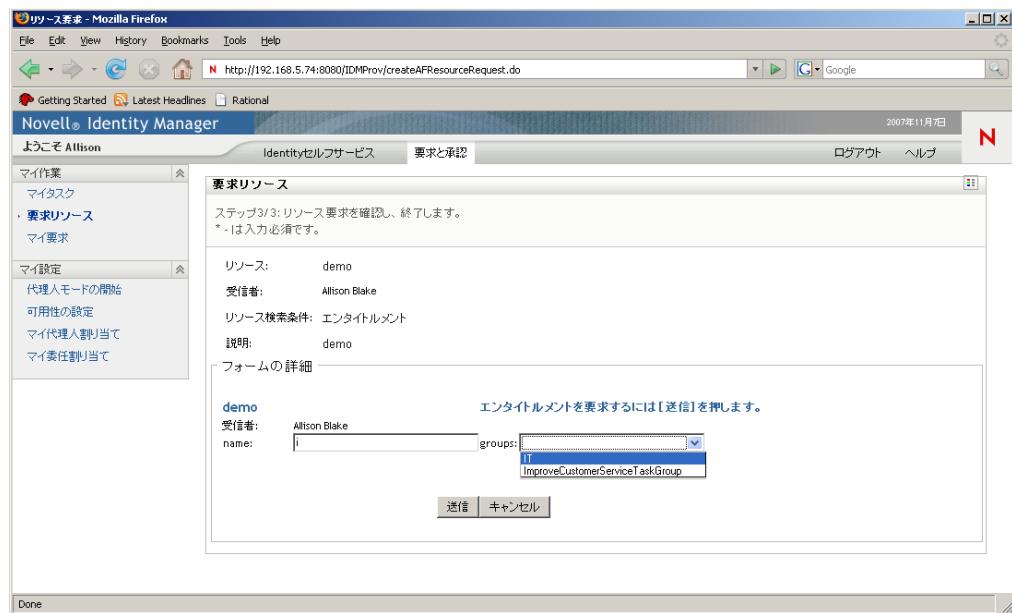
onChange および *onLoad* イベントの詳細については、127 ページのセクション 6.5.2 「一般的なフォームコントロールのプロパティ」に記したイベントプロパティの説明を参照してください。

カスタムイベントの作成

フォーム上で、他のコントロールに条件またはユーザアクションを通知する独自のイベントを作成できます。[イベント] プロパティを使用してイベントを作成します。イベントには任意の名前を付けることができます。*fireEvent ()* メソッドを使用しイベントの名前を渡して、イベントを明示的に開始する必要があります。

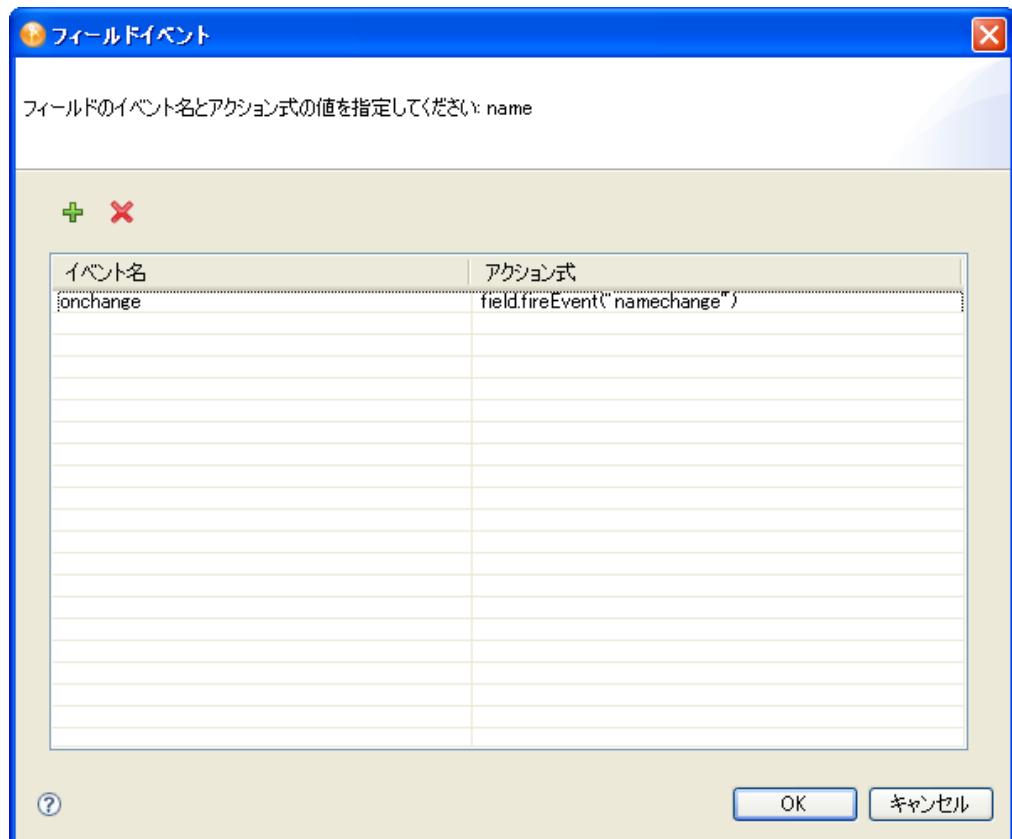
ユーザが入力した値に一致するグループだけを返すクエリを、[グループ] コンテナで実行することもできます。図 6-7 に示した例では、ユーザは名前フィールドに値を入力します。ユーザがタブで次のフィールドに移動すると、ドロップダウンリストのコンテンツは、*namechange* カスタムイベントで開始したクエリから取り込まれます。

図 6-7 ユーザアプリケーションのランタイムカスタムイベントのサンプル



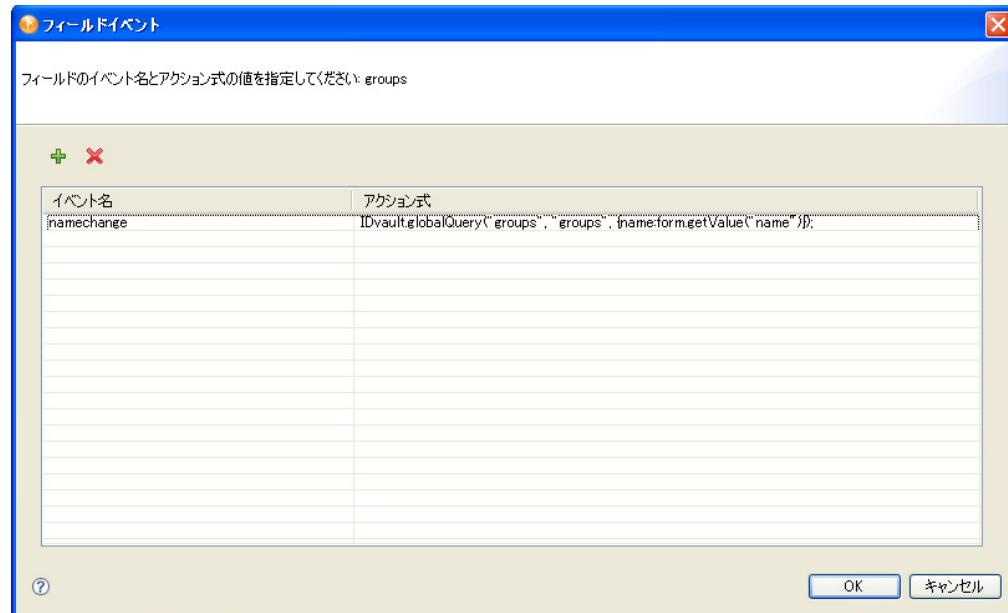
【名前】フィールドは、*onchange* イベントで *namechange* イベントを開始する [イベント] プロパティを定義します。この定義を図 6-8 に示します。

図 6-8 *field.FireEvent()* メソッドのサンプル



namechange イベントは、*groups* というクエリを実行する式を含みます。

図 6-9 カスタムイベント定義の例



クエリの使用の詳細については、「154 ページのセクション 6.7 「フォームでの DAL クエリの使用」」を参照してください。

6.3.4 [スクリプト] タブの使用

[スクリプト] タブを使用して、ページレベルスコープを持つスクリプトを定義します。ページレベルスコープとは、スクリプトがページロード時にロードされ、フォームの存続中使用できることを意味します。表 6-2 で説明するいずれかの方法でスクリプトを用意できます。

表 6-2 スクリプトタイプ

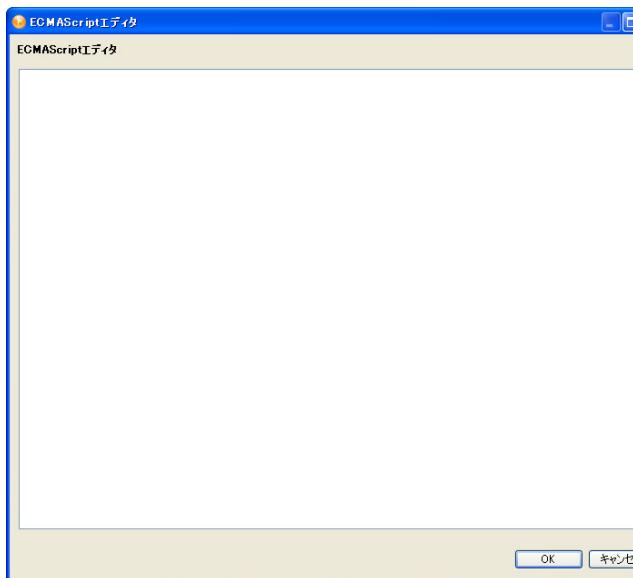
スクリプトタイプ	説明
外部	スクリプトは、与えられた URL を使用して参照することによってページに組み込まれます。スクリプトブロックは、たとえば <>script type="text/javascript" scr="http://some.server/custom.js" > のようになります。custom.js ファイルがフォームのロード時にインポートされます。
インライン	スクリプトは、フォームの <script> ブロックに直接挿入されます。

これらのスクリプトはページのロード時にロードされるので、フォームコントロールとそれに関連付けられたイベントハンドラスクリプトは、ページがロードされるときに、スコープに入っていません。ページレベルスクリプトとイベントレベルスクリプト間で依存関係をコード化しないようにしてください。ただし、イベントレベルスクリプト内からページレベルスクリプトをコールできます。

外部 JavaScript ファイルへのリンクを追加する

- 1 [スクリプト] タブを開いて、[追加] をクリックします。
- 2 次のようにフィールドに入力します。

フィールド	説明
ID	わかりやすい名前を指定します。この値はイベントアクション Expression Builder で表示されます。
タイプ	デフォルト値である [外部] をそのまま使用します。
URL/ インラインスクリプト	フィールド内をクリックすると、[スクリプト URL エディタ] ボタンが右側に表示されます。このボタンをクリックしてエディタを表示します。



.js ファイルへの URL を入力し、[OK] をクリックして、[スクリプト] タブに戻ります。外部リンクを追加すると、フォームを表示するページに次の行が生成されます。

```
<script src=" someURL.com/script.js"/>
```

インラインスクリプトを作成する

- 1 [スクリプト] タブを開いて、[追加] をクリックします。
- 2 次のようにフィールドに入力します。

フィールド	説明
ID	わかりやすい名前を指定します。この値はイベントアクション Expression Builder で表示されます。
タイプ	値を [インライン] に設定します。

フィールド	説明
<i>URL/ インラインスクリプト</i>	<p>ECMAScript エディタを使用して JavaScript を編集します。エディタにアクセスするには、前に指定した ID と同じ名前のタブをクリックします。このタブは、[署名宣言] タブの隣にあります。</p> <p>インラインスクリプトの場合、次の行がページに挿入されます。</p> <pre><script> 入力したスクリプト </script></pre>

インラインスクリプトと外部スクリプトはどちらもページのロード時に実行されますが、ページがコントロールをロードする前に実行されます。また、特にフォームコントロールイベントでコールされたときにも実行されます。

6.4 アクションのリファレンス

この項では、フォームに追加できるアクションについて説明します。アクションはボタンとして実装されます。ボタンごとにカスタム表示ラベルを指定できます。

表 6-3 有効なアクション

アクション名	フォームタイプ	説明
<i>ApprovalAction</i>	承認	<p>承認アクティビティが完了し、「承認済み」フローパスに従って次の動作に移ります。このアクションを使用する場合、[読み込み専用の場合に隠す] フォームプロパティを True に設定する必要があります。設定しない場合、フォームを展開するときに検証に失敗します。</p> <p>ヒント : <i>ApprovalAction</i> では、フォームに関連付けられた承認アクティビティから出力される承認済みフローパスが必要になります。</p>
<i>CancelAction</i>	要求および承認	<p>要求フォームの場合、キャンセルによって、ユーザは要求リソース検索条件フォームに戻ります。承認フォームの場合、キャンセルによって、ユーザは [マイタスク] リストに戻ります。</p>

アクション名	フォームタイプ	説明																
<i>CommentAction</i>	承認フォーム	デフォルトラベルが「コメント履歴の表示」に設定されたボタンを生成します。このボタンをクリックすると、ワークフローの開始から現時点に至る各動作の処理履歴を表示した「コメント」ダイアログボックスが開きます。次の例に示すように、表示されるデータには、日付、動作名、ユーザ、およびコメントが含まれます。																
		 <p>The screenshot shows a Windows Internet Explorer window titled 'コメント - Windows Internet Explorer'. The URL is http://192.168.5.74:8080/IDMProv/getAPComments.do?value(apwaDetailId=3b1c24bb464241119c35a4d76c001a5f&value(apwaDetailId=3b1c24bb464241119c35a4d76c001a5f). The main content area is a table titled 'タスクコント' (Task Control) with columns '日付' (Date), 'アクティビティ' (Activity), 'ユーザ' (User), and 'コメント' (Comment). The table contains four rows of data:</p> <table border="1"> <thead> <tr> <th>日付</th> <th>アクティビティ</th> <th>ユーザ</th> <th>コメント</th> </tr> </thead> <tbody> <tr> <td>13/11/2007 18:30:13</td> <td>開始</td> <td>IDMProv</td> <td>ワークフローが開始されました</td> </tr> <tr> <td>13/11/2007 18:30:13</td> <td>開始</td> <td>IDMProv</td> <td>転送:</td> </tr> <tr> <td>13/11/2007 18:30:13</td> <td>单一承認</td> <td>IDMProv</td> <td>割り当て</td> </tr> </tbody> </table> <p>Below the table, there are two checkboxes: 'ユーザコメントの表示' (Display user comments) and 'システムコントロールの表示' (Display system control). At the bottom are '更新' (Update) and '閉じる' (Close) buttons.</p>	日付	アクティビティ	ユーザ	コメント	13/11/2007 18:30:13	開始	IDMProv	ワークフローが開始されました	13/11/2007 18:30:13	開始	IDMProv	転送:	13/11/2007 18:30:13	单一承認	IDMProv	割り当て
日付	アクティビティ	ユーザ	コメント															
13/11/2007 18:30:13	開始	IDMProv	ワークフローが開始されました															
13/11/2007 18:30:13	開始	IDMProv	転送:															
13/11/2007 18:30:13	单一承認	IDMProv	割り当て															
		コメントは、 UpdateAction （後で説明）によって更新され、ワークフローデータベースに保持されます。																
		注: このアクションを含むすべてのフォームで、 <i>apwaComment</i> というフィールドも含む必要があります。																
<i>DenyAction</i>	承認	承認アクティビティが完了し、拒否フローパスに進みます。このアクションを使用する場合、「[読み込み専用の場合に隠す]」フォームプロパティを True に設定する必要があります。設定しない場合、フォームを展開するときに検証に失敗します。																
		ヒント: <i>DenyAction</i> では、フォームに関連付けられた承認アクティビティから出力される拒否フローパスが必要になります。																
<i>RefusalAction</i>	承認	承認アクティビティが完了し、拒否済みフローパスに進みます。このアクションを使用する場合、「[読み込み専用の場合に隠す]」フォームプロパティを True に設定する必要があります。設定しない場合、展開するときに検証に失敗します。																
		ヒント: <i>RefusalAction</i> では、フォームに関連付けられた承認アクティビティから出力される拒否フローパスが必要になります。																
<i>SubmitAction</i>	要求および承認	ワークフローが開始し、転送フロータイプを実行します。ワークフローは、ユーザ入力データをすべてワークフロー内の次の動作に渡します。																

アクション名	フォームタイプ	説明
<i>UpdateAction</i>	承認	承認アクティビティがユーザコメントをワークフローデータベースに書き込みます。通常、[apwaComment] フォームフィールドに関連付けられたテキスト領域があります。ユーザがこのフィールドにテキストを入力してこのアクションをクリックすると、テキストはワークフローデータベースの afcomment テーブルに保持されます。コメントは、CommentAction（前に説明）を使用して取得および表示できます。次の例では、テキスト領域と更新アクションボタン（ラベルは UpdateAction）を示します。

单一承認
該当するボタンを選択して要求を承認または却下してください。

要求元: Jane Brown
要求日: 12/11/2007
理由:
コメント:

注: フォームには、apwaComment というフィールドを含める必要があります。含めない場合、プロビジョニング要求定義が検証に失敗します。

apwaComment の詳細については、「[127 ページの「ユーザ入力コメントのコントロール」](#)」を参照してください。

次の表では、アクションに対して設定できるプロパティについて説明します。

表 6-4 アクションのプロパティ

プロパティ名	説明
表示ラベル	ボタンに表示するテキストを指定します。
表示可能	True の場合、ランタイム時にアクションを表示することを指定します。
エラー時にブロック	True の場合、いずれかのフォームのコントロールが検証に失敗した場合に、アクションをブロックすることを指定します。これは SubmitAction の場合に推奨されます。
	アクションボタンがデータを送信する場合には、False に設定しないでください。設定した場合、無効なデータが送信され、予期しない結果を招くことがあります。
読み込み専用の場合に隠す	True の場合、フォームが読み込み専用のときにアクションを隠すことを指定します。ユーザが最初に要求せずにタスクを開いた場合、フォームは読み込み専用になります。フォームに ApprovalAction、DenyAction、または RefusalAction が含まれている場合、このプロパティは True に設定する必要があります。False に設定した場合、検証エラーが発生し、フォームを展開できません。

6.5 フォームコントロールのリファレンス

この項では、フォームに追加できるコントロールについて説明します。

表 6-5 コントロールタイプとサポートするデータタイプ

コントロールタイプ	データタイプ						
	プール	日付	10進	DN	整数	文字列	時間
CheckBoxPickList	X		X	X	X	X	
DatePicker		X					
DateTimePicker							X
DNContainer					X		
DNDisplay					X		
DNLookup					X		
DNMaker					X		
DNQuery					X		
Global List							X
HTML							X
MVCheckbox		X	X	X	X		
MVEditor		X	X	X	X		
PickList		X	X	X	X		
スタティックリスト	X		X		X	X	
テキスト			X	X	X	X	
テキスト領域							X
役職							X
TrueFalseRadioButtons	X						
TrueFalseSelectBox	X						

- ◆ 127 ページのセクション 6.5.1 「ユーザ入力コメントのコントロール」
- ◆ 127 ページのセクション 6.5.2 「一般的なフォームコントロールのプロパティ」
- ◆ 128 ページのセクション 6.5.3 「CheckBoxPickList」
- ◆ 129 ページのセクション 6.5.4 「DatePicker」
- ◆ 130 ページのセクション 6.5.5 「DateTimePicker」
- ◆ 131 ページのセクション 6.5.6 「DNContainer」
- ◆ 132 ページのセクション 6.5.7 「DNDisplay」
- ◆ 133 ページのセクション 6.5.8 「DNLookup」
- ◆ 137 ページのセクション 6.5.9 「DNMaker」
- ◆ 138 ページのセクション 6.5.10 「DNQuery」
- ◆ 139 ページのセクション 6.5.11 「Global List」
- ◆ 140 ページのセクション 6.5.12 「HTML」

- ◆ 140 ページのセクション 6.5.13 「MVCheckbox」
- ◆ 141 ページのセクション 6.5.14 「MVEditor」
- ◆ 146 ページのセクション 6.5.15 「PickList」
- ◆ 148 ページのセクション 6.5.16 「スタティックリスト」
- ◆ 149 ページのセクション 6.5.17 「テキスト」
- ◆ 150 ページのセクション 6.5.18 「テキスト領域」
- ◆ 150 ページのセクション 6.5.19 「役職」
- ◆ 150 ページのセクション 6.5.20 「TrueFalseRadioButtons」
- ◆ 151 ページのセクション 6.5.21 「TrueFalseSelectBox」

6.5.1 ユーザ入力コメントのコントロール

Designer では、ユーザがワークフローにコメントを追加したり、以前に入力したコメントを表示できるようにするフォームを追加するための特別な内部コントロールをサポートします。コメントは、*CommentAction* または *UpdateAction* を使用するフォームで必要になります。コメントはワークフローデータの一部ではないので、フローデータオブジェクトからはアクセスできません。コメントは、ワークフローデータベースの afcomment テーブルに格納された特別なデータ項目です。コメントは、afprocess テーブル内に requestid を示す行がある間、保持されます。

ユーザコメントをサポートするフォームを作成する

- 1 コントロールをフォームに追加します。データタイプとして「コメント」を選択します。フォームフィールド名は自動的に「apwaComment」と定義され、コントロールタイプは「TextArea」になります。单一のフォームには、1つのコメントフィールドしか含められません。
- 2 *CommentAction* または *UpdateAction* をフォームに追加します。

詳細については、123 ページのセクション 6.4 「アクションのリファレンス」を参照してください。

6.5.2 一般的なフォームコントロールのプロパティ

次の表のプロパティは、各コントロールで使用できます。

表 6-6 一般プロパティ

プロパティ名	説明
表示ラベル	コントロールの識別に表示されるラベルを指定します。ローカライズできます。
編集可能	コントロールが編集可能 (True) かどうかを指定します。これ以外の場合、読み込み専用として表示されます。

プロパティ名	説明
イベント	コントロールのイベントを指定します。指定できる値は次のとおりです。 <ul style="list-style-type: none"> ◆ OnChange: 次のいずれかの場合に開始します。 <ul style="list-style-type: none"> ◆ onload 直後。 ◆ 別のスクリプトがコントロールの値を変更した場合。 ◆ ユーザがコントロールに関連付けられたデータ値への変更をコミットした場合。これは、ユーザがコントロールからタブで次に移動したとき、またはそれ以外では、コントロールからフォーカスを外したときに発生します。たとえば、ユーザがコントロール（[テキスト]、[TextArea]、[DatePicker]などのテキスト入力ベースのコントロール）からタブで移動するとき、またはユーザが別のエントリ選択肢（[PickList]、[MVCheckbox]、[StaticList]などの選択ベースのコントロール）を選択したときに、発生します。 ◆ onLoad: コントロールの onload イベントは、コントロールが最初にページにロードされたときに1回だけ開始されます。初期値を設定したり、エントリを事前選択するために使用できます。ただし、コントロールが一定の順序でロードされるという保証はありません。
複数値	これは読み込み専用のプロパティです。コントロールが複数値属性をサポートする（True）かどうかを指定します。
必須	コントロールでユーザ入力が必要（True）かどうかを指定します。
ツールチップ	コントロールのツールチップのテキストを指定します。ローカライズできます。
表示可能	ユーザインターフェースでコントロールを表示する（True）かどうかを指定します。

Sort Order(ソート順序)

リストベースのコントロールはアルファベット順にコンテンツをソートします。DNベースのリストの場合、[式の表示] プロパティ結果に基づいたアルファベット順にソートされます。他のタイプではすべて、表示ラベルに基づいてソートされます。

6.5.3 CheckBoxPickList

[CheckBoxPickList] コントロールを使用して、ユーザが、チェックボックスとして表示され動的に生成される選択リストから、1つ以上の値を表示および選択できるようにします。

関連付けられたデータタイプが識別ボルトから取得した DN である場合、チェックボックスラベルを完全修飾 DN として表示したり、[式の表示] プロパティを使用して代わりに表示する属性を指定できます。

図 6-10 [CheckBoxPickList] コントロールのサンプル

CheckBoxPickList: Asia Europe North America South America

表6-7 [CheckboxPickList] のプロパティ

プロパティ名	説明
DN 式ルックアップのエンティティキー	識別ポールトから取得した DN をこのコントロールに取り込み、この値をわかりやすく表示しようと考える場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定する必要があります。
式の表示	識別ポールトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。選択するエンティティは、ディレクトリ抽象化レイヤの [表示] プロパティを True に設定しており、識別ポールトから取得している DN のエンティティである必要があります。
複数選択の許可	[DN 式ルックアップのエンティティキー] を指定するときに必要です。チェックボックスラベルとして表示する属性を選択します。たとえば、ユーザエンティティの [名] 属性と [姓] 属性を表示するには、「FirstName LastName」のように式を構築します。
ソートエントリ	属性のディレクトリ抽象化レイヤの [表示]、[読み取り]、[検索]、および [必須] プロパティは、True に設定する必要があります。

6.5.4 DatePicker

日付および時刻の表示と入力にこのコントロールを使用します。このコントロールを使用すると、ユーザは、ポップアップカレンダから日付を選択するか、テキストフィールドに日付を入力するかを選択できます。ランタイム時にフォームは、ユーザのロケールおよびタイムゾーンの形式を使用して日付を自動的に検証します。ユーザが間違った形式を入力した場合、フォームにはエラーメッセージが表示されます。[DatePicker] コントロールのツールチップに、有効な日付形式が表示されます。デフォルトの [DatePicker] コントロールは次のようにになります。

図6-11 [DatePicker] コントロールのサンプル



[日付選択の表示] プロパティが True の場合、フォームには日付フィールドがボタンとともに表示されます。ユーザがボタンをクリックすると、日付を選択するカレンダがフォームに表示されます。カレンダポップアップを次に示します。

図6-12 [カレンダ] コントロールのサンプル



表6-8 [DatePicker] コントロールのプロパティ

プロパティ名	説明
日付時刻インジケーター	<code>False</code> に設定した場合、カレンダポップアップには時刻が表示されません。
曜日のヘッダー	値を一重引用符で囲んだカンマ区切りのリストがカレンダポップアップに表示され、曜日が示されます。この値はローカライズできます。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
月の名前	値を一重引用符で囲んだカンマ区切りのリストがカレンダポップアップに表示され、月の名前が示されます。この値はローカライズできます。
日付選択の表示	<code>True</code> に設定した場合、カレンダポップアップが表示されます。 <code>False</code> に設定した場合、カレンダポップアップは表示されません。ユーザは、ロケールに応じた適切な形式を使用して、テキストフィールドに日付を入力する必要があります。

6.5.5 DateTimePicker

[時間] データタイプの日付と時刻の表示と入力に、このコントロールを使用します。ユーザは、ポップアップカレンダから日付と時刻を選択するか、テキストフィールドに値を入力できます。ランタイム時に、フォームでは、ユーザのロケールおよびタイムゾーンの形式で、日付と時刻が自動的に検証されます。ユーザが間違った形式を入力した場合、フォームにはエラーメッセージが表示されます。[DateTimePicker] のツールチップに、有効な日付形式が表示されます。デフォルトのコントロールは次のようになります。

図6-13 [DateTimePicker] コントロールのサンプル

DateTimePicker:

[日付選択の表示] プロパティを `True` に設定した場合、フォームには、テキストフィールドとカレンダボタンが表示されます。ユーザがカレンダボタンをクリックすると、ユーザ

が日付と時刻の値を選択する〔カレンダ〕コントロールがフォームに表示されます。カレンダポップアップを次に示します。

図 6-14 [DateTimePicker] カレンダコントロール



表 6-9 [DateTimePicker] コントロールのプロパティ

プロパティ名	説明
曜日のヘッダー	値を一重引用符で囲んだカンマ区切りのリストがカレンダポップアップに表示され、曜日が示されます。この値はローカライズできます。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
isDateTime	False に設定した場合、カレンダポップアップには時刻が表示されません。
月の名前	値を一重引用符で囲んだカンマ区切りのリストがカレンダポップアップに表示され、月の名前が示されます。この値はローカライズできます。
日付選択の表示	True に設定した場合、カレンダポップアップが表示されます。False に設定した場合、カレンダポップアップは表示されません。ユーザは、テキストフィールドに日付を入力するときに、ロケールに応じた形式で入力する必要があります。

6.5.6 DNContainer

このコントロールを使用して、指定したルートコンテナ内からコンテナオブジェクトをユーザが選択できるようにします。このコントロールを使用して、ユーザを特定のコンテナのサブツリーに制限できます。これは、専用バージョンの [DNLookup] コントロールです。

図 6-15 ルートコンテナを指定した [DNContainer] コントロール



表 6-10 [DNContainer] コントロールのプロパティ

プロパティ名	説明
オブジェクトのロックアップに使用されるエンティティキー	ドロップダウンリストからエンティティを選択します。選択したエンティティにより、指定したエンティティのコンテナ内にあるオブジェクトをロックアップできるユーザの権限が制限されます。エンティティキーとルートコンテナを指定した場合、エンティティキーが優先されます。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
ルートコンテナ	ユーザが [オブジェクトセレクタ] ボタンをクリックしたときにロックアップするルートコンテナを指定します。
クリアボタンの表示	True に設定した場合、フォームには [フィールドのリセット] ボタンが表示されます。
オブジェクト履歴ボタンの表示	True に設定した場合、フォームには [履歴の表示] ボタンが表示されます。
オブジェクトセレクタボタンの表示	True に設定した場合、フォームには [オブジェクトセレクタ] ボタンが表示されます。

6.5.7 DNDisplay

このコントロールを使用して、読み込み専用 DN を表示します。フローデータからコントロールを取り込みます。コントロールは、設定したプロパティに応じて、完全な DN を表示することも、DN に関連付けられた属性のセットを表示することもできます。

図 6-16 [DNDisplay] のサンプル



図 6-17 [式の表示] を指定した [DNDisplay] のサンプル



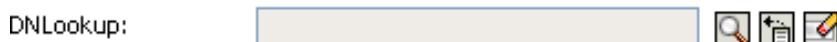
表6-11 [DNDisplay] コントロールのプロパティ

プロパティ名	説明
式の表示	<p>完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN をマスクする場合は、式ビルダを起動して、リストから目的の属性を選択します。(最初に [DN 式ルックアップのエンティティキー] を指定する必要があります)。</p> <p>たとえば、ユーザエンティティの [名] 属性と [姓] 属性を表示するには、「FirstName LastName」のように式を構築します。</p> <p>ディレクトリ抽象化レイヤで、属性の [表示]、[読み取り]、[検索]、および [必須] プロパティが、True に設定されていることを確認してください。詳細については、66 ページのセクション 3.7.2 「属性のプロパティ」 を参照してください。</p>
DN 式ルックアップのエンティティキー	<p>識別ポールトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN または CN をマスクする場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定します。</p> <p>選択するエンティティは次の条件を満たしている必要があります。</p> <ul style="list-style-type: none">◆ ディレクトリ抽象化レイヤの [表示] プロパティが True に設定されている。◆ 作業している DN のエンティティである。 <p>詳細については、151 ページのセクション 6.6 「識別名の操作」 を参照してください。</p>

6.5.8 DNLookup

このコントロールを使用して、ユーザが、識別ポールトから DN を検索し取得できるようになります。フローデータからの DN でコントロールを初期化できます。ユーザが検索できるエンティティおよびコンテナと、DN の形式を制御するようにプロパティを設定します。

図6-18 [DNLookup] コントロールのサンプル



[DNLookup] コントロールに関連付けられたボタンについて、表 6-12 で説明します。

表6-12 [DNLookup] コントロールのボタン

ボタン	説明
	[オブジェクトルックアップ] ダイアログボックスを開きます。[オブジェクトセレクタのタイプ] プロパティを介してダイアログボックスにコンテナを表示するか、オブジェクトを表示するかを定義します。次に、オブジェクトルックアップの例を示します。
	ドロップダウンリストに表示される属性（前の例の [説明]）は、ディレクトリ抽象化レイヤで指定されます。このボタンが使用できるかどうかは、[オブジェクトセレクタボタンの表示] プロパティで制御されます。
	履歴を表示します。ユーザは、検索したオブジェクトの履歴を表示できます。このリストから選択することも、コンテンツを消去することもできます。このボタンが使用できるかどうかは、[オブジェクト履歴ボタンの表示] プロパティで制御されます。
	フィールドをリセットします。フィールドのコンテンツを削除します。このボタンが使用できるかどうかは、[クリアボタンの表示] プロパティで制御されます。

表6-13 [DNLookup] コントロールのプロパティ

プロパティ名	説明
式の表示	<p>このプロパティは、フローデータからコントロールを初期化するときにのみ適用されます。完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN をマスクする場合は、式ビルダを起動して、リストから目的の属性を選択します。(最初に [DN 式ルックアップのエンティティキー] を指定する必要があります)。</p> <p>たとえば、ユーザエンティティの [名] 属性と [姓] 属性を表示するには、「FirstName LastName」のように式を構築します。</p> <p>ディレクトリ抽象化レイヤで、属性の [表示]、[読み取り]、[検索]、および [必須] プロパティが、True に設定されていることを確認してください。詳細については、66 ページのセクション 3.7.2 「属性のプロパティ」 を参照してください。</p>
DN 式ルックアップのエンティティキー	<p>このプロパティは、フローデータからコントロールを初期化するときにのみ適用されます。識別ボルトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN または CN をマスクする場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定します。</p> <p>選択するエンティティは次の条件を満たしている必要があります。</p> <ul style="list-style-type: none"> ◆ ディレクトリ抽象化レイヤの [表示] プロパティが True に設定されている。 ◆ 作業している DN のエンティティである。 <p>参照先、151 ページのセクション 6.6 「識別名の操作」。</p>

プロパティ名	説明
オブジェクトセレクタのタイプ	[オブジェクトセレクタ] ダイアログボックスが、オブジェクトルックアップを実行するか、コンテナルックアップを実行するかを決定します。次に、オブジェクトルックアップの例を示します。
<i>paramlist:</i>	[オブジェクトセレクタ] ダイアログでオブジェクトルックアップを実行します。[オブジェクトのルックアップ] 使用されるエンティティキー] プロパティを介してルックアップ条件を指定します。
<i>container:</i>	[オブジェクトセレクタ] ダイアログで 1 つ以上のコンテナを選択対象として表示します。検索対象のコンテナは、[検索コンテナ] プロパティによって決まります。このプロパティは、[オブジェクトのルックアップ] 使用されるエンティティキー] プロパティで名前が付けられたエンティティに対して、ディレクトリ抽象化レイヤで指定されます。たとえば、[オブジェクトのルックアップ] 使用されるエンティティキー] プロパティが「グループ」である場合、検索コンテナはデフォルトで「%group-root%」に設定されます。検索コンテナが使用されない場合、ユーザーアプリケーションのインストール中に指定された検索ルートが使用されます。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
クリアボタンの表示	True に設定した場合、フォームには [フィールドのリセット] ボタンが表示されます。
オブジェクト履歴ボタンの表示	True に設定した場合、フォームには [履歴の表示] ボタンが表示されます。
オブジェクトセレクタボタンの表示	True に設定した場合、フォームには [オブジェクトセレクタ] ボタンが表示されます。

6.5.9 DNMaker

このコントロールを使用して、ユーザが、名前付け値を指定しコンテナを選択することによって、DN 値を構築できるようにします。

図 6-19 [DNMaker] コントロールのサンプル



表 6-14 [DNMaker] コントロールのボタン

ボタン	説明
	次のようなコンテナ検索のオブジェクトセレクタが起動します。
	コンテナ検索ルートは、[オブジェクトルックアップ]に使用されるエンティティプロパティで指定したエンティティに対して定義されます。このボタンが使用できるかどうかは、[オブジェクトセレクタボタンの表示] プロパティで制御されます。
	履歴の表示ユーザは、検索したオブジェクトの履歴を表示できます。このリストから選択することも、コンテンツを消去することもできます。このボタンが使用できるかどうかは、[オブジェクト履歴ボタンの表示] プロパティで制御されます。
	フィールドのリセットフィールドのコンテンツを削除します。このボタンが使用できるかどうかは、[クリアボタンの表示] プロパティで制御されます。

表 6-15 [DNMaker] コントロールのプロパティ

プロパティ	説明
オブジェクトのルックアップに使用されるエンティティキー	必須フィールドです。ドロップダウンリストからエンティティを選択します。これにより、ユーザが [オブジェクトセレクタ] ボタンをクリックしたときに開始する検索が決まります。
エンティティキーとルートコンテナ	エンティティキーとルートコンテナを指定した場合、エンティティキーが優先されます。
名前付け属性	名前付け属性は最終 DN を構築するために使用されます。この値は、ユーザに対する追加ヒントとしてコントロールの表示ラベルの隣に表示されます。
ルートコンテナ	ユーザが [オブジェクトセレクタ] ボタンをクリックしたときにルックアップするルートコンテナを指定します。ルートコンテナを指定しない場合、ユーザアプリケーションは、「検索コンテナ」というディレクトリ抽象化レイヤのプロパティ内にあるエンティティ用のコンテナを使用します。指定したエンティティに対して検索コンテナが指定されていない場合、ユーザアプリケーションのインストール中に指定されたルートコンテナ DN が使用されます。エンティティキーとルートコンテナを指定した場合、エンティティキーが優先されます。
クリアボタンの表示	True に設定した場合、フォームには [フィールドのリセット] ボタンが表示されます。
オブジェクト履歴ボタンの表示	True に設定した場合、フォームには [履歴の表示] ボタンが表示されます。
オブジェクトセレクタボタンの表示	True に設定した場合、フォームには [オブジェクトセレクタ] ボタンが表示されます。

6.5.10 DNQuery

[DNQuery] は、専用バージョンの [DNLookup] コントロールです。[DNQuery] は [DNLookup] と同様に、ユーザが識別ポールトから DN を検索し取得できるようにします。ただし、[DNQuery] では、オブジェクトセレクタのコンテンツは、プロパティからではなく、ディレクトリ抽象化レイヤのクエリオブジェクトの結果から取得できます。

表 6-16 [DNQuery] コントロールのプロパティ

プロパティ名	説明
DAL グローバルクエリキー	実行する DAL クエリオブジェクトのキーを指定します。イベントアクション Expression Builder から選択できます。 DAL クエリの使用の詳細については、「 154 ページのセクション 6.7 「フォームでの DAL クエリの使用」 」を参照してください。DAL クエリの定義の詳細については、「 54 ページのセクション 3.4 「クエリの操作」 」を参照してください。
DAL グローバルクエリパラメータ	クエリパラメータの値を指定します。たとえば、これは文字列「Sales」を「groupname」というクエリパラメータに渡します。 <pre>(function (){return {"groupname":"Sales"}})();</pre>

プロパティ名	説明
式の表示	データ項目マッピング値から初期データをコントロールに取り込むときに、このプロパティを使用して、表示する属性を指定します。
DN 式ルックアップのエンティティキー	<p>このプロパティは、フローデータからコントロールを初期化するときにのみ適用されます。識別ポールトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN または CN をマスクする場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定します。</p> <p>選択するエンティティは次の条件を満たしている必要があります。</p> <ul style="list-style-type: none"> ◆ ディレクトリ抽象化レイヤの [表示] プロパティが True に設定されている。 ◆ 作業している DN のエンティティである。
	参照先、 151 ページのセクション 6.6 「識別名の操作」 。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
クリアボタンの表示	True に設定した場合、フォームには [フィールドのリセット] ボタンが表示されます。
オブジェクト履歴ボタンの表示	True に設定した場合、フォームには [履歴の表示] ボタンが表示されます。DAL グローバルクエリを指定した場合、[オブジェクト履歴] ボタンは表示されません。
オブジェクトセレクタボタンの表示	True に設定した場合、フォームには [オブジェクトセレクタ] ボタンが表示されます。

6.5.11 Global List

このコントロールを使用して、ユーザがドロップダウンリストから単一のエントリを選択できるようにします。リストのコンテンツは、ディレクトリ抽象化レイヤのグローバルリスト要素で定義されます。

図 6-20 [グローバルリスト] コントロールのサンプル



表 6-17 [グローバルリスト] のプロパティ

プロパティ名	説明
DAL グローバルリストキー	グローバルリストの固有識別子を指定します。これは、ディレクトリ抽象化レイヤで指定したキーに一致する必要があります。

グローバルリストの詳細については、「[51 ページのセクション 3.3 「リストの操作」](#)」を参照してください。

6.5.12 HTML

このコントロールを使用して、[フォームの詳細] に HTML フラグメントを追加します。追加するには、[HTML コンテンツ] プロパティで HTML フラグメントを指定します。また、フォームコントロールでイベントを介して、条件付きで HTML フラグメントを追加することもできます。どちらの場合でも、(function() { return "<yourTag yourAttr='your attr value' />"; })(); などの匿名関数を使用して、HTML を指定します。

例：

```
(function(){ return "<table bgcolor='#C0C0C0'><th colspan='3' align='center'>Table Header Goes Here</th><tr><td>Value 1.1</td><td>Value 1.2</td></tr><tr><td>Value 2.1</td><td>Value 2.2</td></tr></table>"; })()
```

6.5.13 MVCheckbox

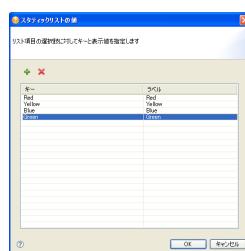
このコントロールを使用して、ラベル付きのチェックボックスのセットを表示します。[リスト項目] プロパティを使用して、ラベルとそれに関連付けられた値を指定します。次に、[MVCheckbox] コントロールのサンプルを示します。

図 6-21 [MVCheckbox] コントロールのサンプル

MVCheckboxControl: Blue Green Red Yellow

表 6-18 [MVCheckbox] コントロールのプロパティ

プロパティ名	説明
リスト項目	チェックボックスのラベルと値を構成する一連の静的な値を定義できます。[リストプロパティ] ボタンをクリックすると、次に示すリスト値のダイアログボックスが開きます。



ヒント: このコントロールからユーザ入力値を取得するには、`flowdata.get()` ではなく `flowdata.getObject()` を使用します。`flowdata.get()` を使用した場合、最初の値しか取得されません。

値の事前選択の詳細については、「[280 ページのセクション 10.2.3 「フォームコントロールの例」](#)」を参照してください。

6.5.14 MVEditor

このコントロールを使用して、ユーザが、ドロップダウンリストボックス内の複数の値を表示、編集、または追加できるようにします。識別ポールトから動的にデータをロードすることも、ユーザが値を入力できるようにすることもできます。

コントロールの見た目は、コントロールのデータタイプと指定したプロパティに応じて異なります。たとえば、データタイプが DN である場合、DN に関連した特定の属性を表示するプロパティを設定できます。また、[オブジェクトのルックアップに使用されるエンティティキー] プロパティを設定して、ユーザが値を検索し選択できるようにする [オブジェクトセレクタ] ボタンを有効にすることもできます。

実行する DAL グローバルクエリを指定したり、オブジェクトセレクタを操作するルート DN を指定できるようにするプロパティもあります。

図 6-22 オブジェクトルックアッププロパティを設定した [MVEditor] のサンプル



表6-19 オブジェクトセレクタプロパティを設定した [MVEditor] コントロールのボタン

ボタン	説明
	オブジェクトセレクタと呼ばれる検索ダイアログボックスを開きます。[オブジェクトセレクタ] ダイアログボックスは次のようにになります。
	ユーザは、リストから値を選択してコントロールに取り込むことができます。ドロップダウンリストに表示される属性（前の例の「説明」）は、ディレクトリ抽象化レイヤで指定されます。属性の [UIControl] プロパティで指定します。参照先 68 ページの「属性の UI 制御プロパティ」 。このボタンが使用できるかどうかは、[オブジェクトセレクタボタンの表示] プロパティで制御されます。
	履歴の表示ユーザは、検索したオブジェクトの履歴を表示できます。このリストから選択することも、コンテンツを消去することもできます。このボタンが使用できるかどうかは、[オブジェクト履歴ボタンの表示] プロパティで制御されます。
	フィールドのリセットフィールドのコンテンツを削除します。このボタンが使用できるかどうかは、[クリアボタンの表示] プロパティで制御されます。

オブジェクトルックアッププロパティを設定しない場合、[MVEditor] は単純な編集コントロールを表示します。

図6-23 オブジェクトルックアッププロパティを設定しない [MVEditor] のサンプル



表6-20 [MVEditor] コントロールのボタン

ボタン	説明
	リストの最後に項目を追加します。

ボタン	説明
	選択したリスト項目を削除します。
	選択したリスト項目を編集します。

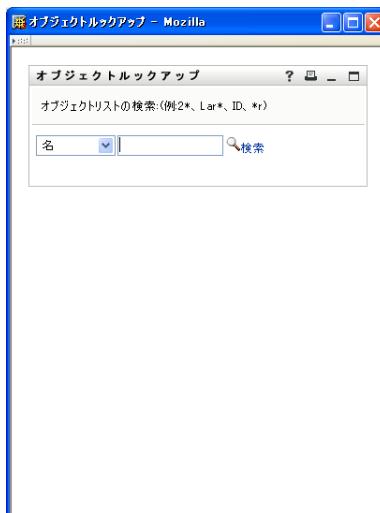
ヒント : [MVEditor] コントロールの [編集可能] プロパティが **False** に設定されている場合、このコントロールは読み込み専用であり、フォームには [MVEditor] コントロールのボタンは表示されません。

表 6-21 [MVEditor] コントロールのプロパティ

プロパティ名	説明
追加データのエントリテキストフィールド	True に設定し、單一行のデータがある場合（データは DN ではない）、コントロールはデータ入力テキストフィールドを表示します。テキストフィールドは空欄か、1 つの値しか入っていない場合に表示されます。それ以外の場合は、ドロップダウンリストが表示されます。複数行のデータがある場合は、常にドロップダウンリストが表示されます。
DAL グローバルクエリ	指定するグローバルクエリの結果をコントロールに取り込む場合に、この値を指定します。キー名を指定します。イベントアクション Expression Builder から選択できます。フォームでクエリを使用する方法について詳しくは、「 154 ページのセクション 6.7 「フォームでの DAL クエリの使用」 」を参照してください。クエリの定義の詳細については、「 54 ページのセクション 3.4 「クエリの操作」 」を参照してください。
DAL グローバルクエリパラメータ	クエリパラメータの値を指定します。たとえば、これは文字列「 Sales 」を「groupname」というクエリパラメータに渡します。
式の表示	(function (){return {"groupname":"Sales"}})(); 完全な DN または CN 値を表示する場合は、この値を空白にしておきます。 属性を代わりに表示することによって DN または CN をマスクする場合は、式ビルダを起動して、リストから目的の属性を選択します。（最初に [DN 式ルックアップのエンティティキー】を指定する必要があります）。 たとえば、ユーザエンティティの [名] 属性と [姓] 属性を表示するには、「 FirstName LastName 」のように式を構築します。 ディレクトリ抽象化レイヤで、属性の [表示]、[読み取り]、[検索]、および [必須] プロパティが、 True に設定されていることを確認してください。詳細については、「 66 ページのセクション 3.7.2 「属性のプロパティ」 」を参照してください。

プロパティ名	説明
一意性の強制	ユーザ入力のリスト項目が一意になるように強制します。
DN 式ルックアップのエンティティキー	識別ポールトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。
	属性を代わりに表示することによって DN または CN をマスクする場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定します。
	選択するエンティティは次の条件を満たしている必要があります。
	<ul style="list-style-type: none"> ◆ ディレクトリ抽象化レイヤの [表示] プロパティが True に設定されている。 ◆ 識別ポールトから取得している DN を持つエンティティである。
	詳細については、 151 ページのセクション 6.6 「識別名の操作」 を参照してください。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
大文字と小文字を無視する	True に設定した場合、一意性を強制しているときに大文字と小文字を無視します。
下限(番号の場合のみ)	最小の整数または 10 進数値です。
最大長	文字列値の最大文字数です。この値に達すると、コントロールは入力をブロックします。
最短長	文字列値の最小文字数です。コントロールは、少なくともこの文字数をユーザが入力していることを検証します。
表示される行数	コントロールによって表示される行数です。これは、取得または表示されるレコード数ではなく、コントロールの縦サイズになります。この数値を 10 に設定し、表示するレコードが 5 つしかない場合でも、コントロールサイズは 10 行のままでです。
	行数は 1 または 3 以上に設定できます。2 には設定できません。ブラウザでスクロールボタンを表示するために必要なスペースが取れないためです。2 に設定した場合、Designer は [プロジェクトチェック] ビューに警告を表示し、3 に設定しなおします。
番号のみ	True に設定した場合、番号だけを入力できます。

プロパティ名	説明
オブジェクトセレクタのタイプ	[オブジェクトセレクタ] ダイアログボックスが、オブジェクトルックアップを実行するか、コンテナルックアップを実行するかを決定します。次に、オブジェクトルックアップの例を示します。



paramlist: [オブジェクトセレクタ] ダイアログボックスでオブジェクトルックアップを実行します。[オブジェクトのルックアップ]に使用されるエンティティキー] プロパティを介してルックアップ条件を指定します。

container: オブジェクトセレクタで1つ以上のコンテナを選択対象として表示します。検索対象のコンテナは、[検索コンテナ] プロパティによって決定されます。このプロパティは、[オブジェクトのルックアップ]に使用されるエンティティキー] プロパティで名前が付けられたエンティティに対して、ディレクトリ抽象化レイヤで指定されます。たとえば、[オブジェクトのルックアップ]に使用されるエンティティキー] プロパティが「グループ」である場合、検索コンテナはデフォルトで「%group-root%」に設定されます。検索コンテナが使用されない場合、ユーザーアプリケーションのインストール中に指定された検索ルートが使用されます。

解決DN

False に設定した場合、**式の表示**ではなく DN が表示されます。多数の DN が返されると予想され、パフォーマンスが重要な場合には、この使用を検討してください。

ルートコンテナ

ユーザが [オブジェクトセレクタ] ボタンをクリックしたときにルックアップするルートコンテナを指定します。エンティティキーとルートコンテナを指定した場合、エンティティキーが優先されます。

オブジェクト履歴ボタンの表示

True に設定した場合、[オブジェクトの履歴] ボタンがコントロールの隣に表示されます。

オブジェクトセレクタボタンの表示

True に設定した場合、[オブジェクトセレクタ] ボタンがコントロールの隣に表示されます。

プロパティ名	説明
ソートエントリ	True に設定した場合、昇順に結果をソートします。詳細については、 128 ページの「Sort Order(ソート順序)」 を参照してください。
上限(番号の場合のみ)	ユーザが入力できる最大の数値です。
ヒント : このコントロールからユーザ入力値を取得するには、 <code>flowdata.get()</code> ではなく <code>flowdata.getObject()</code> を使用します。 <code>flowdata.get()</code> を使用した場合、最初の値しか取得されません。	

項目の事前選択の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

6.5.15 PickList

[PickList] コントロールを使用して、ユーザが、動的に生成される選択リストから、1つ以上の値を表示および選択できるようにします。リスト項目は、識別ボルトから取得した DN または CN 値です。完全な DN または CN を表示することも、[PickList] プロパティを使用して代わりに表示する属性を指定することもできます。

図 6-24 DN をマスクしない [PickList] コントロールのサンプル

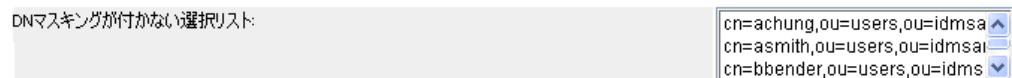


図 6-25 DN をマスクした [PickList] コントロールのサンプル

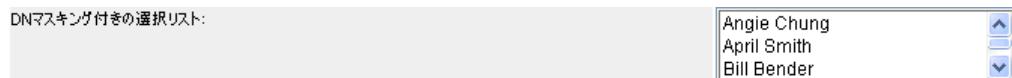


表 6-22 [PickList] コントロールのプロパティ

プロパティ名	説明
複数選択の許可	True に設定した場合、ユーザは、プラットフォーム固有の複数選択キーを使用して、複数のリスト値を選択できます。
	True に設定した場合、コントロールは、 表示される行数 プロパティで指定した値に関係なく、最低 3 行で表示されます。この値が False である場合、 表示される行数 プロパティが使用されます。

プロパティ名	説明
式の表示	<p>完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN または CN をフォーマットする場合は、式ビルダを起動して、リストから目的の属性を選択します。(最初に [DN 式ルックアップのエンティティキー] を指定する必要があります)。</p> <p>たとえば、ユーザエンティティの [名] 属性と [姓] 属性を表示するには、「FirstName LastName」のように式を構築します。</p> <p>ディレクトリ抽象化レイヤで、属性の [表示]、[読み取り]、[検索]、および [必須] プロパティが、True に設定されていることを確認してください。詳細については、66 ページのセクション 3.7.2 「属性のプロパティ」 を参照してください。</p>
DN 式ルックアップのエンティティキー	<p>識別ボールトから取得した完全な DN または CN 値を表示する場合は、この値を空白にしておきます。</p> <p>属性を代わりに表示することによって DN または CN をマスクする場合は、ドロップダウンリストからエンティティを選択し、[式の表示] プロパティで属性のセットを指定します。</p> <p>選択するエンティティは次の条件を満たしている必要があります。</p> <ul style="list-style-type: none"> ◆ ディレクトリ抽象化レイヤの [表示] プロパティが True に設定されている。 ◆ 識別ボールトから取得している DN を持つエンティティである。
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
表示される行数	コントロールによって表示される行数です。これは、取得または表示されるレコード数ではなく、コントロールの縦サイズになります。この数値を 10 に設定し、表示するレコードが 5 つしかない場合でも、コントロールサイズは 10 行のままでです。
	表示される行数は 複数選択の許可 設定に関連します。[複数選択の許可] が True に設定されている場合は、表示される行数は常に 3 (以上) になります。[複数選択の許可] が False に設定されている場合は、行数を 1 か 3 以上に設定できます。2 には設定できません。ブラウザでスクロールボタンを表示するために必要なスペースが取れないためです。2 に設定した場合、Designer は [プロジェクトチェック] ビューに警告を表示し、3 に設定しなおします。

プロパティ名	説明
ソートエントリ	True に設定した場合、昇順に結果をソートします。詳細については、128 ページの「Sort Order(ソート順序)」を参照してください。
ヒント: このコントロールからユーザ入力値を取得するには、flowdata.get() ではなく flowdata.getObject() を使用します。flowdata.get() を使用した場合、最初の値しか取得されません。	
事前選択されたオプションを使用したコントロールの表示の詳細については、「280 ページのセクション 10.2.3 「フォームコントロールの例」」を参照してください。	

6.5.16 スタティックリスト

このコントロールを使用して、ユーザが単一の項目を選択できるドロップダウンリストに項目のリストを表示します。リスト項目は静的であり、プロビジョニング要求定義に格納されます。このフィールドを「必須」に設定していない場合、「選択するにはここをクリックします」というテキストが表示されます。

図 6-26 [スタティックリスト] コントロールのサンプル

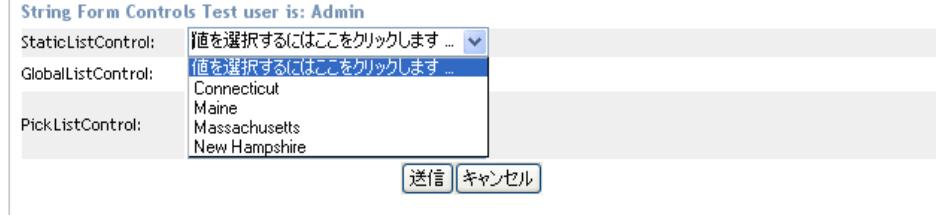
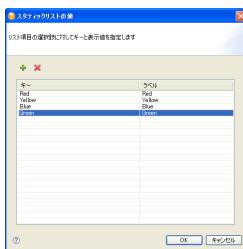


表 6-23 [スタティックリスト] のプロパティ

プロパティ名	説明
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。

プロパティ名	説明
リスト項目	スタティックリストを構成する一連のラベルと値を定義できます。[リストプロパティ] ボタンをクリックすると、次に示すリスト値のダイアログボックスが開きます。



[追加] をクリックして、リスト項目を追加します。各リスト項目には、一意のキーが必要です。ダイアログボックスは、新しいリスト項目を挿入したときに、自動的に一意のキーを生成します。キーワードをクリックして、変更できます。空白キー (NULL) は有効なので、空白キーと空白ラベルを持つリスト項目を設定できます。表示されるラベルは、デフォルトの言語に対して定義されたものです。

6.5.17 テキスト

データ表示またはユーザ入力に [テキスト] コントロールを使用します。コントロールのデータタイプに応じてユーザ入力が検証されます。

図 6-27 [テキスト] コントロールのサンプル



表 6-24 [テキスト] コントロールのプロパティ

プロパティ名	説明
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。
下限(番号の場合のみ)	10 進数か整数値に設定できる下限の番号です。
最大長	文字列値の最大長です。この長さに達した場合、入力をブロックします。
最短長	文字列値の最短長です。ユーザが少なくともこの長さで文字列を入力していることを検証します。
許可される文字数	ユーザが入力できる文字数を指定します。これは フィールド幅(ピクセル単位) に関連しています。
上限(番号の場合のみ)	10 進数か整数値に設定できる最大の番号です。

プロパティ名	説明
検証マスク(正規表現)	フィールドのデータの検証に使用される式です。Designer にはデフォルトで、デフォルトセットの検証マスクが用意されています。[ウインドウ] > [初期設定] > [プロジェクト] > [検証マスク] の順に選択して、これらのマスクを有効にする必要があります。詳細については、19 ページのセクション 2.3 「プロジェクトビューの初期設定の設定」を参照してください。

6.5.18 テキスト領域

このコントロールを使用して、複数行データを表示したり入力を受け付けます。ユーザは、プラットフォームの複数選択キーコンビネーションを使用して、複数行のデータを選択できます。

図 6-28 [テキスト領域] コントロールのサンプル



表 6-25 [テキスト領域] コントロールのプロパティ

プロパティ名	説明
表示される桁数	コントロールの表示幅、つまり文字数幅です。
表示される行数	同時に表示される行数です。

6.5.19 役職

この読み込み専用コントロールを使用して、フォームにラベルを付けたり、指示を記入します。

表 6-26 [タイトル] コントロールのプロパティ

プロパティ名	説明
署名されたフォームド False に設定し、フォームが署名済みフォーム（デジタル署名を使用）であるキュメントにタイトル 場合、[タイトル] コントロールは表示されません。	を表示します
フォントサイズ	小、中、または大を選択します。
スタイルクラス	フォントスタイル（太字など）と色をパレットから選択します。

6.5.20 TrueFalseRadioButtons

このコントロールを使用して、True または False の選択肢をラジオボタンのセットとして表示します。

図 6-29 [TrueFalseRadioButtons] コントロールのサンプル

TrueFalseRadioButtons: False True

このコントロールにはカスタムプロパティはありません。

6.5.21 TrueFalseSelectBox

このコントロールを使用して、ドロップダウンリストに True または False の選択肢を表示します。フィールドが必須でない場合にのみ、「値を選択するにはここをクリックします」というテキストが表示されます。

図 6-30 [TrueFalseSelectBox] コントロールのサンプル

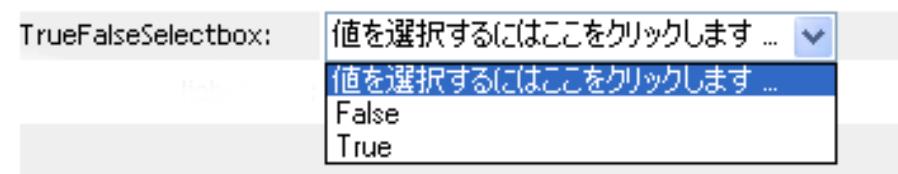


表 6-27 [TrueFalseSelectBox] のプロパティ

プロパティ名	説明
フィールド幅(ピクセル単位)	このフィールドを使用して、フォームでのフィールドの表示幅を設定します。デフォルトは 200 ピクセルです。

6.6 識別名の操作

次のコントロールは、識別名(DN)専用のサポートを提供します。

- ◆ DNDisplay
- ◆ DNLookup
- ◆ DNMaker
- ◆ MVEditor
- ◆ PickList

この項では、次の専用サポートについて説明します。

- ◆ 151 ページのセクション 6.6.1 「DN のフォーマット」
- ◆ 152 ページのセクション 6.6.2 「オブジェクトセレクタの操作」

6.6.1 DN のフォーマット

DN 値がある場合、DN を表示することも、その DN に関連した属性のセットを表示することができます。たとえば、コントロールにユーザエンティティの DN を表示する場合、

エンティティの [名] 属性と [姓] 属性を代わりに表示できます。この機能をサポートするコントロールは、[DNDisplay]、[DNLookup]、[MVEditor]、[Picklist] です。

コントロールの [式の表示] プロパティで表示する属性を定義します。この [式の表示] は、属性キーを属性値に置き換えることにより、ランタイム時に解決されます。

6.6.2 オブジェクトセレクタの操作

場合によっては、ユーザが候補の値のリストから DN を検索したり選択できるようにすることもできます。[オブジェクトセレクタ] ダイアログボックス（[オブジェクトルックアップ] ダイアログボックスとも呼ばれる）が、この機能を提供します。[オブジェクトセレクタ] ダイアログボックスのコンテンツは、フォームコントロールのプロパティ（「表 6-28」を参照）と、どのように DAL プロパティを定義しているか（「152 ページの「[DNLookup] コントロールタイプの定義とオブジェクトセレクタのコンテンツ」」を参照）によって制御されます。

オブジェクトセレクタは、必須および検索可能なディレクトリ抽象化レイヤアクセスプロパティが True に設定されている、文字列または DNLookup データタイプの属性だけをサポートします。

表 6-28 [オブジェクトセレクタ] ダイアログボックスを定義するプロパティ

プロパティ	説明
オブジェクトのルックアップに使用されるエンティティキー	これは、検索または表示する DN を持つディレクトリ抽象化レイヤエンティティを示すキーです。このフィールドは必須です。
オブジェクトセレクタのタイプ	<i>paramlist:</i> [オブジェクトセレクタ] ダイアログボックスでオブジェクトルックアップを実行します。[オブジェクトのルックアップに使用されるエンティティキー] プロパティを介してルックアップ条件を指定します。
オブジェクトセレクタボタンの表示	<i>container:</i> [オブジェクトセレクタ] ダイアログボックスで 1 つ以上のコンテナを選択対象として表示します。検索対象のコンテナは、[検索コンテナ] プロパティによって決定されます。このプロパティは、[オブジェクトのルックアップに使用されるエンティティキー] プロパティで名前が付けられたエンティティに対して、ディレクトリ抽象化レイヤで指定されます。たとえば、[オブジェクトのルックアップに使用されるエンティティキー] プロパティが「グループ」である場合、検索コンテナはデフォルトで '%group-root%' に設定されます。検索コンテナが使用されない場合、ユーザアプリケーションのインストール中に指定された検索ルートが使用されます。
オブジェクトセレクタボタンの表示	True に設定した場合、[オブジェクトセレクタ] ボタンがコントロールに表示されます。設定しない場合は、表示されません。

[DNLookup] コントロールタイプの定義とオブジェクトセレクタのコンテンツ

[オブジェクトのルックアップに使用されるエンティティキー] を指定した場合、オブジェクトセレクタのコンテンツは、属性の [DNLookup] コントロールタイプの定義（ディレクトリ抽象化レイヤ内）により定義されます。たとえば、ユーザエンティティを

オブジェクトルックアップに指定し、マネージャを属性に指定した場合、オブジェクトセレクタはマネージャの [DNLookup] コントロールタイプ定義を使用してルックアップ条件を決定するので、ユーザは [名] 属性と [姓] 属性で検索できます。マネージャエンティティの [DNLookup] 定義を図 6-31 に示します。

図 6-31 ユーザの [DNLookup] プロパティ定義に対する [マネージャ] 属性

▼ UIコントロール
属性の表示に使用する書式設定または特殊なコントロールを指定します:

データタイプ: DN

フォーマットタイプ: <なし>

制御タイプ: DNLookup

▼ DNLookup表示
ルックアップ操作で表示するエンティティと属性を選択します:

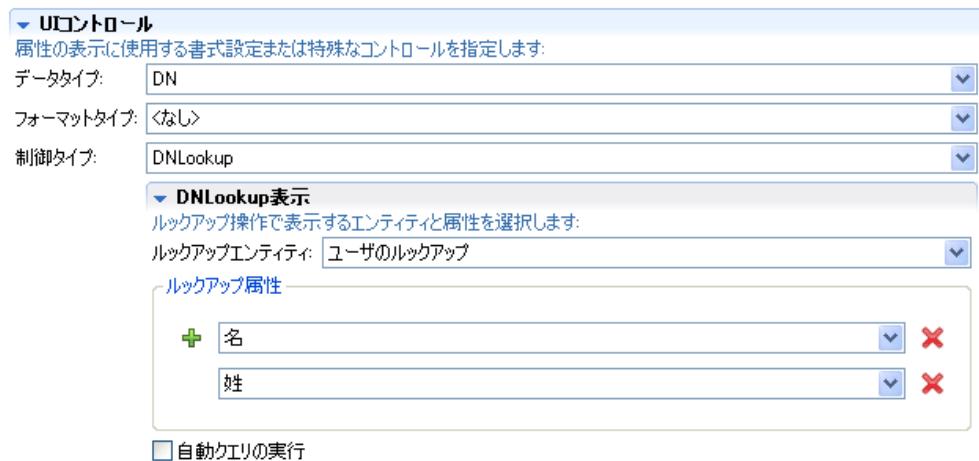
ルックアップエンティティ: ユーザのルックアップ

ルックアップ属性

+ 名

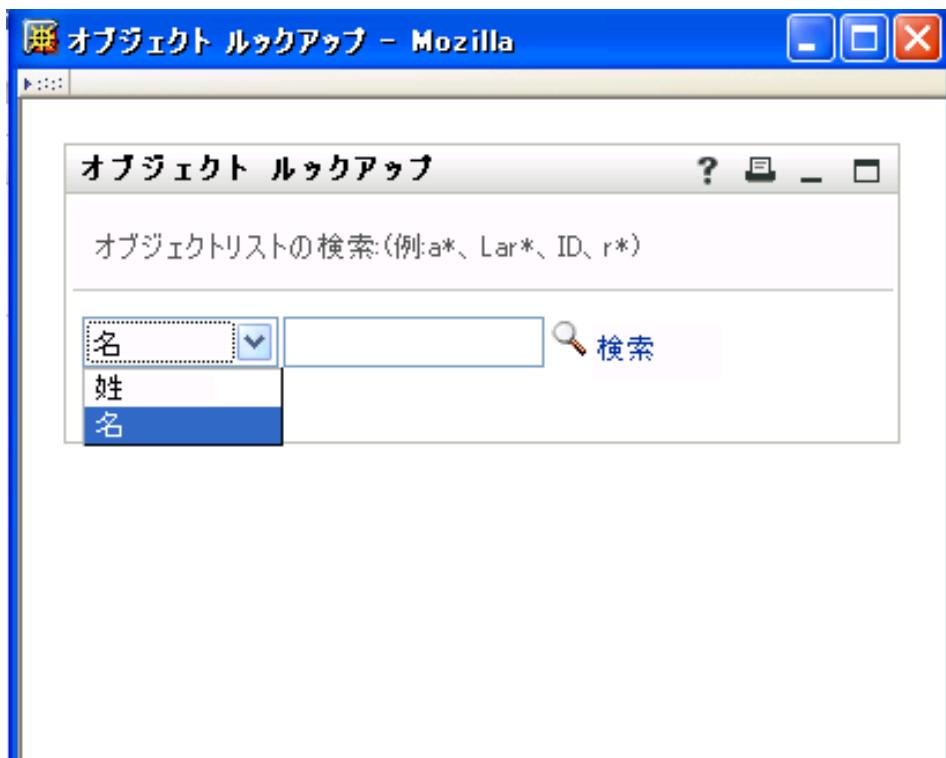
姓

自動クリアの実行



結果のオブジェクトセレクタを図 6-32 に示します。

図 6-32 オブジェクトセレクタのサンプル



[ルックアップ] 属性を変更することによって、オブジェクトセレクタが使用する属性を変更できます。オブジェクトセレクタでその他の属性を許可する

- 1 目的の属性が、ルックアップエンティティとして指定されたエンティティに対して定義されているかどうかを特定します。(この例では、マネージャルックアップです。)
- 2 必要な属性がルックアップエンティティで使用できる場合、ルックアップ属性に追加できます。[検索] および [読み取り] プロパティを必ず True に設定してください。設定しないと、[オブジェクトセレクタ] ダイアログボックスに表示されません。
- 3 ルックアップエンティティに対してまだ属性が存在していない場合、次の手順を行う必要があります。
 - ◆ ルックアップエンティティに属性を追加します。たとえば、前の例のようなマネージャルックアップに別の属性を表示するには、マネージャルックアップエンティティにその属性を追加します。詳細については、[49 ページのセクション 3.2.3 「属性の追加」](#) を参照してください。
 - ◆ [DNLookup] 定義に属性を追加します。
 - ◆ 変更した定義を展開します。この例では、[マネージャ] 属性の定義を変更したので、マネージャルックアップエンティティ (定義に新しい属性を追加した場合) およびユーザエンティティを再展開しました。
 - ◆ アプリケーションサーバの DirectoryAbstractionLayerDefinitions キャッシュをリフレッシュします。

6.7 フォームでの DAL クエリの使用

ディレクトリ抽象化レイヤで定義されたクエリオブジェクトでは、LDAP 検索を事前定義して、その後ワークフローフォームからこの検索を実行できます。この項では、クエリを定義し、フォームでそれを使用する方法について説明します。

特定の従業員にテレフォンカードを配布したいと考えており、自宅で勤務し、その自宅がローカルオフィスの市外局番外にある従業員だけに配布しようと考えているとします。従業員が次の作業を行うワークフローフォームを作成します。

- ◆ カードを受け取る資格があることを確認する。
- ◆ 資格がある場合、カードの要求を送信する。

フォームでは、ユーザは、自身のローカルオフィスの市外局番を入力し、その市外局番に基づいて、カードを受ける資格があるユーザのリストを確認できます。ランタイム時のフォームを図 6-33 に示します。

図6-33 テレフォンカード要求フォームのサンプル

The screenshot shows a Mozilla Firefox browser window with the title '要求リソース - Mozilla Firefox'. The URL in the address bar is <http://localhost:8080>IDM/createAFRResourceRequest.do>. The page is titled 'Novell® Identity Manager' and displays a 'Welcome Bill' message. On the left, there's a sidebar with links like 'マイ作業', 'マイスク', '要求リソース', and 'マイ要求'. The main content area is titled '要求リソース' and contains the following information:

ステップ3/3リソース要求を確認し、終了します。
* - は入力必須です。

リソース:	コーリングカード
受信者:	Bill Brown
リソース検索条件:	アカウント
説明:	コーリングカード

フォームの詳細

ローカル オフィスのエリア コード以外の場所への通話に対してコーリング カードを取得
オフィス エリア コード: 212

コーリング カードはローカル オフィスのエリア コード以外の場所に住む自宅勤務の従業員用です

候補:

Allison Blake (555) 555-1222
Angie Chung (555) 555-1208
Anthony Palari (555) 555-1202

上記のリストで名前を確認できない場合は、コーリングカードを取得する資格がありません。[キャンセル]を押してください。

送信 キャンセル

[Candidates Picklist] コントロールのデータは、図 6-34 に示すように定義されたクエリの結果から取り込まれます。

図 6-34 テレフォンカードクエリの定義

Employees Outside Local Area Code

一般
クエリの分かりやすいラベルを入力します:
キー: EEOOutsideLocalAreaCode
表示ラベル: Employees Outside Local Area Code
クエリエンティティ: ユーザ

▼ パラメータ
パラメータ参照を定義します:
+ ↑ ↓ 🔍
パラメータキー: AreaCodeParam パラメータ表示ラベル: Area Code ✎

▼ クエリ条件
上記で選択したクエリエンティティの条件を入力します:
(+ 条件グループ化の追加)

(- 条件グループ化の削除)
名: 少ない %AreaCodeParam% ✎

▼ 検索
クエリの検索制約を入力します:
検索ルート: OU=idmsample-alh,O=novell 🔎
検索スコープ: <デフォルト>
最大検索エントリ: 0 (0=ランタイム設定を使用する)

クエリは、ユーザが入力した市外局番の单一の入力パラメータ (*AreaCodeParam*) を取得します。クエリは続いて、ユーザエンティティ (idmsample-alh コンテナ内) を検索し、電話番号が *AreaCodeParam* に入力した同じ値で始まらないユーザを返します。

フォームには、[OfficeAreaCode] という入力フィールドがあります。これは、ユーザがローカルオフィスの市外局番を入力するテキストフィールドです。[OfficeAreaCode] のプロパティを図 6-35 に示します。

図 6-35 [OfficeAreaCode] のプロパティ

The screenshot shows the 'Form Control' dialog with the following details:

フォームのフィールド名	データタイプ	制御タイプ	改行
title	文字列	タイトル	1
OfficeAreaCode	文字列	テキスト	1
title2	文字列	タイトル	1
Candidates	DN	PickList	1
title3	文字列	タイトル	1

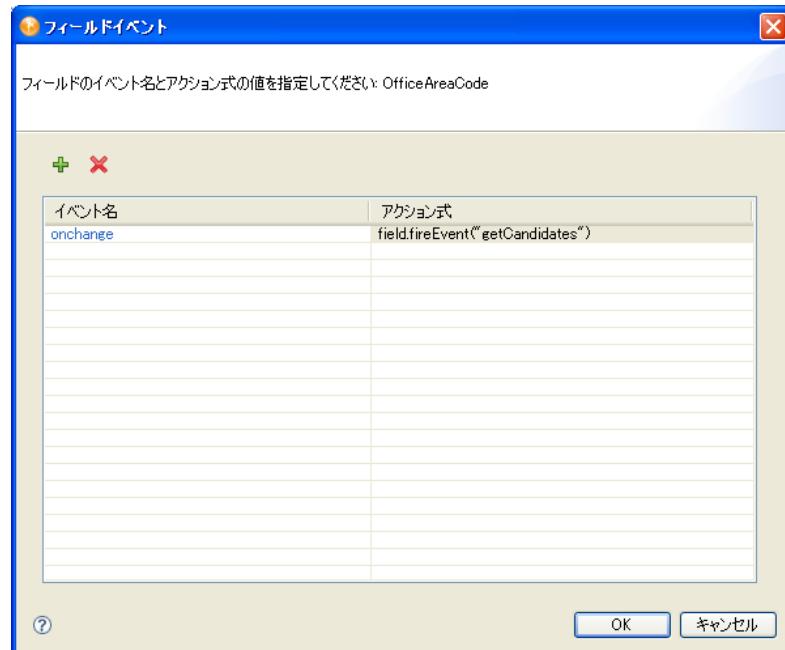
Properties panel (Properties tab selected):

プロパティー	値
表示ラベル	OfficeAreaCode
編集可能	True
必須	False
表示可能	True
複数値	False
イベント	onchange
ツールチップ	
フィールド幅(ピクセル単位)	200
下限(番号の場合のみ)	
許可される文字数	40
検証マスク(正規表現)	
最大長	3
最短長	3
上限(番号の場合のみ)	

Buttons at the bottom: フィールド | アクション | スクリプト |

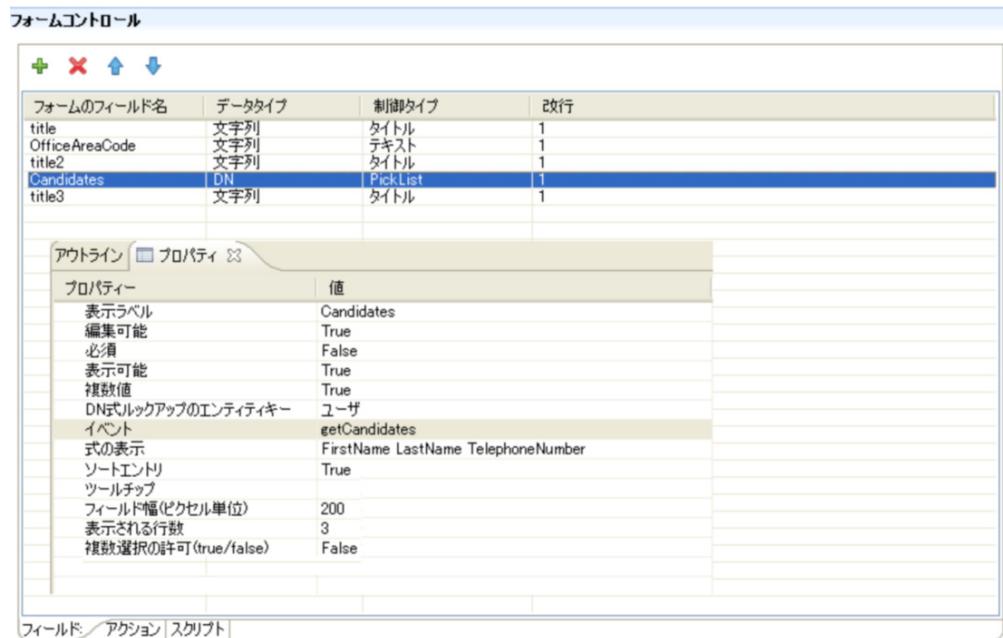
[テキスト] コントロールが *onchange* イベントを定義することに注意してください。
onchange イベントは、ユーザが [テキスト] コントロールからタブで移動すると開始します。*onchange* イベントは、図 6-36 に示すように *getCandidates* カスタムイベントを開始します。

図 6-36 OnChange イベントのプロパティ



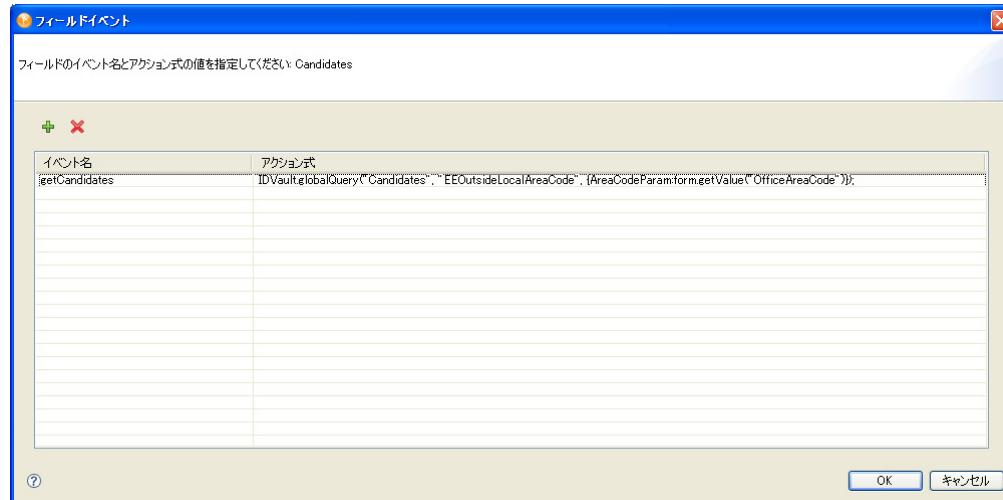
getCandidates イベントは、[Candidates Picklist] コントロールでプロパティとして定義されます。

図 6-37 [Candidates PickList] のプロパティ



イベントが開始されると、*getCandidates* イベントは、*globalQuery()* メソッド（「図 6-38」を参照）をコールするアクション式を実行します。このメソッドは、[Candidates PickList] コントロールの値に、*EEOoutsideLocalAreaCode*（156 ページの図 6-34 で定義）というクエリの結果を取り込みます。[OfficeAreaCode] テキストフィールドの値をクエリの入力パラメータ *AreaCodeParam* として渡します。

図 6-38 GetCandidates イベント



6.8 フォームの印刷

JavaScript を使用して [印刷] ボタンを要求または承認フォームに追加できます。フォームを印刷する機能を追加するには、JavaScript ライブラリ PrintForm.js への URL を追加します。ライブラリは次の URL にあるユーザーアプリケーション WAR 内にあります。/javascript/JUICE/form/PrintForm.js。オプションで次のことを実行できます。

- ◆ フォームの onload イベントまたはいずれかのフィールドの onload イベントに、次の要素を追加することによって、(データが検証された後) 送信を行うときに印刷フォームを開きます。

```
var action = "SubmitAction"  
var pf = new PrintForm(action);  
form.interceptAction("SubmitAction", "around",  
    pf.printFormInterceptor, pf);
```

図 6-39 [印刷] ボタンのある [印刷] フォームのサンプル

The screenshot shows a JUICE form window with the title '1ステップ承認(タイムアウト時に承認)'. On the right side, there is a note: 'エンタイトルメントを
要求するには
[送信]を押します'. The form contains the following fields:

- 受信者: Admin idmsample-pamela
- 現在のタイトル:
- 新しいタイトル: * **[印刷]**
- 要求の理由: * テストです

At the bottom are two buttons: **送信** and **キャンセル**.

承認アクションで印刷するには、コードは次のようにになります。

```
var action = "ApprovalAction";  
var pf = new PrintForm(action);  
form.interceptAction("ApprovalAction", "around",  
    pf.printFormInterceptor, pf);
```

- ◆ フィールドの onload イベントにこれを追加することによって、フォーム上にあるいずれかのフィールドの隣に [印刷] ボタンを追加します。

```
var ctrl = JUICE.UICtrlUtil.getControl(field.getName()); var btn =  
JUICE.UICtrlUtil.addButton(ctrl, "printid", "Print", "Print",  
"javascript:var p = new  
PrintForm();p.printFormAfterValidation(\"printid\");");
```

図 6-40 [印刷] ボタンの例

1ステップ承認(タイムアウト時に承認)

受信者: Admin idmsample-pamela

現在のタイトル:

新しいタイトル: * 販売担当

要求の理由: * テストです

エンタイトルメントを要求するには[送信]を押しま。

[印刷] [送信] [キャンセル]

6.9 フォームへの直接アクセス

ユーザがフォームに直接アクセスできるようにする場合、ユーザアプリケーションの WAR ファイル内でいくつかの変更を行い、WAR ファイルを再展開する必要があります。次に、設定手順について説明します。

- 1 WEB-INF/struts-config.xml ファイルで、次のアクションステートメント内で「roles="authUser"」と指定します。

```
<action path="/getAFResourceList" scope="request"  
name="apwaListForm"  
type="com.novell.srvprv.apwa.actions.GetResourceList"  
roles="authUser">
```

- 2 また、WEB-INF/struts-config.xml ファイルで、次のアクションステートメント内で「roles="guest, authUser"」と指定します。

```
<action path="/createAFResourceRequest" scope="request"  
name="apwaDetail"  
type="com.novell.srvprv.apwa.actions.ProcessResourceRequest"  
roles="guest, authUser">
```

- 3 また、WEB-INF/struts-config.xml ファイルで、次のアクションマッピングを追加します。

```
<action path="/getGuestResourceList" scope="request"  
name="apwaListForm"  
type="com.novell.srvprv.apwa.actions.GetResourceList"  
roles="guest">      <forward name="success"  
path="tiles.guestResourceList"/>      <forward  
name="displaySelector" path="tiles.guestSelector"/>    </action>
```

- 4 WEB-INF/tiles-defs.xml ファイルで、次の定義を追加します。

```
<definition name="tiles.guestSelector"  
extends=".unauthenticatedLayout">  <put name="title"  
value="JSP_TITLE_SELECTOR"/>  <put name="body" value="/jsps/tiles/  
selector.jsp"/></definition><definition  
name="tiles.guestResourceList" extends=".unauthenticatedLayout">  
<put name="title" value="JSP_TITLE_RESOURCE_LIST"/>  <put  
name="body" value="/jsps/tiles/resourceList.jsp"/></definition>
```

- 5 WAR ファイルを再展開します。

フォームに直接アクセスするための URL では、次の形式を使用する必要があります。

`http://localhost:8080/IDMProv/
getAFResourceList.do?apwaLeftNavItem=JSP_MENU_RESOURCE_REQUEST_CLASSIC
&wfdn=PRDEFDN`

パラメータ *PRDEFDN* は、要求定義の DN で置き換える必要があります。このパラメータを含んだ完全な例は次のとおりです。

```
http://localhost:8080/IDMProv/
getAFResourceList.do?apwaLeftNavItem=JSP_MENU_RESOURCE_REQUEST_CLASSIC
&wfdn=cn=TestPRDEF,cn=RequestDefs,cn=AppConfig,cn=Test-
Antelope,cn=TestDrivers,o=novell
```


プロビジョニング要求定義のワークフローの作成

7

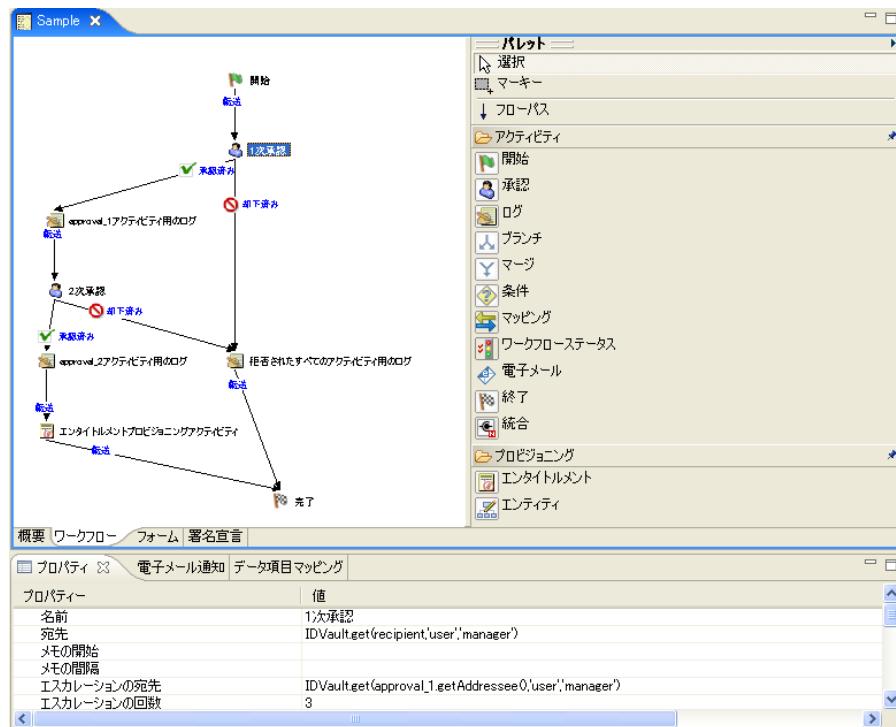
この節では、プロビジョニング要求定義のワークフローを作成する方法について詳しく説明します。主なトピックは次のとおりです。

- ◆ 163 ページのセクション 7.1 「[ワークフロー] タブについて」
- ◆ 167 ページのセクション 7.2 「ワークフローへの動作の追加」
- ◆ 171 ページのセクション 7.3 「フローパスの追加」
- ◆ 172 ページのセクション 7.4 「フローパスの設定」
- ◆ 174 ページのセクション 7.5 「承認動作の宛先の指定」
- ◆ 179 ページのセクション 7.6 「1 つのワークフローインスタンスによる複数の個人のプロビジョニング」
- ◆ 181 ページのセクション 7.7 「エンティティ動作を使用した作業」
- ◆ 183 ページのセクション 7.8 「識別名参照の移植可能化」
- ◆ 183 ページのセクション 7.9 「デジタル署名サポートの設定」

7.1 [ワークフロー] タブについて

[ワークフロー] タブを使用して、[ワークフロー] ページを表示します。[ワークフロー] ページを使用して、プロビジョニング要求定義のワークフローの動作を定義します。[ワークフロー] ページは、キャンバス、パレット、および関連ビューから構成されています。

図 7-1 [ワークフロー] ページ



7.1.1 キャンバス

キャンバスには、ワークフロー内のアクティビティが図で表示されます。テンプレートを使用せずに新しいプロビジョニング要求定義を作成する場合は、開始動作と完了動作を除いてキャンバスには何もありません。

キャンバスの任意の場所を右クリックすると、メニューが表示されます。メニューに含まれるコマンドは次のとおりです。

表 7-1 [ワークフロー] メニュー

項目	説明
削除	選択した動作またはフローパスを削除します。
動作ID の表示	ワークフローエディタで動作名を表示するか、動作 ID を表示するかを切り替えます。動作 ID はシステムで定義されており、編集できません。ただし、動作に関連したエラーが検証中に検出された場合、Designer は、エラーが発生した動作を動作 ID で特定します。この場合は、キャンバス上でその動作を見つけるために、動作 ID の表示に切り替えます。「ウィンドウ」>「初期設定」>「プロビジョニング」>「ワークフロー」>「[図の初期設定]」>「動作ID の表示」の順に選択することによって、動作名と動作 ID のどちらをデフォルトで表示するかを指定できます。

項目	説明
フローパスタイプを表示する	フローパスタイプ（転送、承認済み、却下済みなど）の表示のオンとオフを切り替えます。[フローパスタイプを表示する] をオンにすると、フローパスタイプを示すラベルが各フローパス上に表示されます。
プロパティを表示する	選択した動作の [プロパティ] ビューが表示されます。動作が選択されていない場合、ワークフロー自体の [プロパティ] ビューが表示されます。
データ項目マッピングを表示する	選択した動作の [データ項目マッピング] ビューが表示されます。
電子メール通知の表示	選択した動作の [電子メール通知] ビューが表示されます。

キャンバスには次の 2 つのコントロールがあり、ワークフローを簡単に表示できます。

ズーム：ワークフローの表示倍率を増減する場合に、[ズーム] コントロールを使用します。ワークフロー表示の一部を拡大して、より詳しく表示したり、ワークフロー表示を縮小して、ワークフローの表示範囲を広げたりできます。[ズーム] コントロールの右側にある四角形をクリックすると、倍率は 100% に戻ります。

スケール：ワークフロー表示内の項目間のスペースを増減する場合に、[スケール] コントロールを使用します。たとえば、ワークフロー内の項目に多数のフローパスが付いている場合、スケールを増やせば、各フローパスが見やすくなります。[スケール] コントロールの右側にある四角形をクリックすると、スケールは 100% に戻ります。

図 7-2 ズームコントロールとスケールコントロール



7.1.2 パレット

パレットには、キャンバスにドラッグしてワークフローを作成できる動作を表すアイコンがあります。アイコンを操作するツールと動作をリンクするツールも用意されています。

図 7-3 ワークフローパレット



パレットには、次のツールが含まれます。

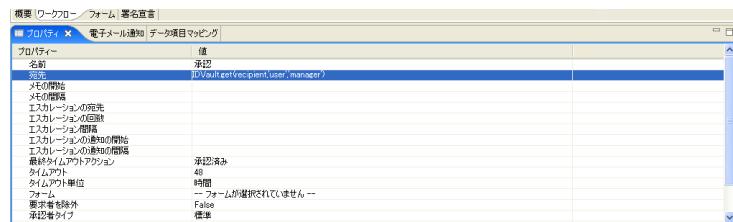
表 7-2 ワークフローパレット

ツール	説明
選択	個々のノードまたはフローパスを選択します。ノードを選択するには、[選択] ツールをクリックし、ノードをクリックします。
マーキー	複数のノードまたはフローパスを選択します。複数の項目をグループとして移動する場合は、このツールを使用します。複数の項目を選択するには、[マーキー] ツールをクリックし、選択する項目の外側の領域をクリックします。マウスボタンを押したまま選択する項目までドラッグし、マウスボタンを放します。
フローパス	複数の項目を選択した場合は、最初に選択した項目のプロパティのみが [プロパティ] ビューに表示されます（ビューの詳細については、「 167 ページのセクション 7.1.3 「ビュー」 」を参照）。
動作（開始、承認、ログなど）	ノード間にフローパスを作成します。フローパスは、ノードを接続するための接続ロジックを提供します。ノードの接続の詳細については、「 171 ページのセクション 7.3 「フローパスの追加」 」を参照してください。
	選択した動作をワークフローに挿入します。動作の追加の詳細については、「 167 ページのセクション 7.2 「ワークフローへの動作の追加」 」を参照してください。動作の詳細については、「 187 ページの第 8 章「ワークフローアクティビティの参照」 」を参照してください。

7.1.3 ビュー

[ワークフロー] ページには、[プロパティ]、[データ項目マッピング]、および [電子メール通知] ビューも含まれています。

図7-4 ワークフロービュー



動作を表すアイコンを右クリックして、コンテキストメニューからビューを選択できます。すべての動作ですべてのビューを使用できるわけではありません。次の表では、ビューとそのビューを使用する動作を示します。

表7-3 動作に対応するビュー

アクティビティ	プロパティ	電子メール通知	データ項目マッピング
開始	X		X
承認	X	X	X
ログ	X		
プランチ	X		
マージ	X		
条件	X		
マッピング	X		X
ワークフローステータス	X		X
電子メール			
終了	X	X	
統合	X		X
エンタイトルメント	X		X
エンティティ	X		X

7.2 ワークフローへの動作の追加

- 1 [ワークフロー] タブをクリックします。プロビジョニング要求定義のワークフローがグラフィカルに表示されます。



どのワークフローにも開始動作と完了動作が必要なので、これらの動作は自動的にキャンバスに追加されます。開始動作は、転送リンクで完了動作に接続されます。

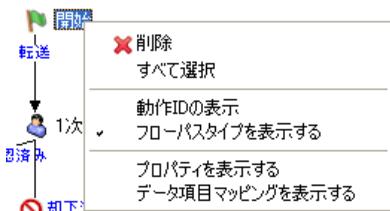
- 2 ワークフローに動作を追加するには、パレットで目的の動作を表すアイコンをクリックし、ワークスペースまでドラッグします。

フローパスによりリンクしている動作間に動作を挿入するには、フローパス上に動作をドロップします。動作間でフローパスを定義する方法の詳細については、「171 ページのセクション 7.3 「フローパスの追加」」を参照してください。ワークフローに動作を追加し終えたら、動作のプロパティを設定する必要があります（「168 ページのセクション 7.2.1 「動作の一般プロパティの設定」」を参照）。さまざまなタイプの動作を設定する方法の詳細については、「187 ページの第 8 章「ワークフローアクティビティの参照」」および「217 ページの第 9 章「統合アクティビティの操作」」を参照してください。

7.2.1 動作の一般プロパティの設定

- 1 プロパティを設定する動作を表すアイコンを右クリックし、メニューから [プロパティを表示する] を選択します。

ヒント: [PRD] メニューから [プロパティを表示する] を選択して、[プロパティ] タブを表示することもできます。



[プロパティ] ビューが表示されます。

概要		ワークフロー	フォーム
<input checked="" type="checkbox"/>	プロパティ	<input checked="" type="checkbox"/>	データ項目マッピング
<input checked="" type="checkbox"/>		<input checked="" type="checkbox"/>	電子メール通知
プロパティ	値		
名前	開始		

- 2** プロパティの列をクリックして、プロパティを設定します。各動作のプロパティの詳細については、「187 ページの第 8 章「ワークフローアクティビティの参照」」を参照してください。

各動作にはデフォルト名が付いています。このデフォルト名は、ワークフローでの動作の具体的な目的を説明する名前に置き換えることをお勧めします。このようにしておけば、ワークフローのグラフィカル表示を見たときに、ワークフローの意味が簡単にわかります。ユーザアプリケーションで表示されるコメントもわかりやすくなります。たとえば、次の図では、ユーザアプリケーションでのデフォルト ID と記述 ID を使用したコメントが示されています。

図 7-5 デフォルト名を使用したユーザコメントでの動作

タスクコメント			
日付	アクティビティ	ユーザ	コメント
13/11/2007 18:35:54	1次承認	IDMProv	レビュー・ア IDMadmin Testに割り当てられたユーザタスク
13/11/2007 18:35:54	2次承認	IDMProv	レビュー・ア IDMadmin Testに割り当てられたユーザタスク
13/11/2007 18:36:41	1次承認	IDMProv	レビュー・ア IDMadmin Testによって要求されたユーザタスク
13/11/2007 18:36:45	1次承認	IDMProv	レビュー・ア IDMadmin Testによって承認されたユーザタスク

1 - 4 / 4

ユーザコメントの表示
 システムコメントの表示

[更新](#) [閉じる](#)

図 7-6 記述名を使用したユーザコメントでの動作

タスクコメント			
日付	アクティビティ	ユーザ	コメント
13/11/2007 18:35:54	マネージャ承認	IDMProv	レビュー・ア IDMadmin Testに割り当てられたユーザタスク
13/11/2007 18:35:54	人事承認	IDMProv	レビュー・ア IDMadmin Testに割り当てられたユーザタスク
13/11/2007 18:36:41	マネージャ承認	IDMProv	レビュー・ア IDMadmin Testによって要求されたユーザタスク
13/11/2007 18:36:45	マネージャ承認	IDMProv	レビュー・ア IDMadmin Testによって承認されたユーザタスク

1 - 4 / 4

ユーザコメントの表示
 システムコメントの表示

[更新](#) [閉じる](#)

7.2.2 データ項目マッピングの定義

[データ項目マッピング] ビューを使用して、データフローからフォームのフィールドにデータをマップし（前動作マッピング）、データをフォームからデータフローに戻します（後動作マッピング）。

- 1** プロパティを設定する動作アイコンを右クリックし、メニューから [データ項目マッピングを表示する] を選択します。

ヒント : [PRD] メニューから [データ項目マッピングを表示する] を選択して、[データ項目マッピング] タブを表示することもできます。

[データ項目マッピング] ビューが表示されます。



- 2** 前動作マッピングの場合は、マップする項目の [ソース式] フィールドをクリックし、式を指定します。後動作マッピングの場合は、マップする項目の [ターゲット式] フィールドをクリックし、式を指定します。

前動作マップは次の操作に使用できます。

- ◆ フォームコントロール値を初期化する。
- ◆ フォームコントロールのデフォルト値を設定する。
- ◆ LDAP クエリから派生したデータリストを使用して複雑なフォームコントロールに入力する。
- ◆ 前の動作のフォームコントロールから、現在の動作のフォームコントロールにデータを渡す。
- ◆ 外部 Java クラスを呼び出してデータを処理する。

後動作マップは次の操作に使用できます。

- ◆ フローデータ内に新しいデータ項目を作成する。
- ◆ 動作からフローデータにフォームコントロールデータを移動する。
- ◆ 外部 Java クラスを呼び出してデータを処理する。

さまざまなタイプの動作に対するデータ項目マッピングの詳細については、「[187 ページの第 8 章「ワークフローアクティビティの参照」](#)」を参照してください。

開始動作の前動作マップには、ハードコーディングされた文字列、プロセスロケールや受信者などのシステム変数、および識別ポールト式 (ECMA 式ビルダ VDX Expr パネルを使用して作成) を含められます。

前動作マップでは、ユーザが入力するフォームフィールドの [ソース式] を空白にしておきます。または、ユーザが入力するフォームフィールドのデフォルト値を求めるソース式を作成します。どちらの場合も、フォームフィールドは編集可能なフィールドとして定義する必要があります。フォームフィールドのプロパティを設定する方法の詳細については、「[127 ページのセクション 6.5.2 「一般的なフォームコントロールのプロパティ」](#)」を参照してください。

7.2.3 電子メール通知設定の定義

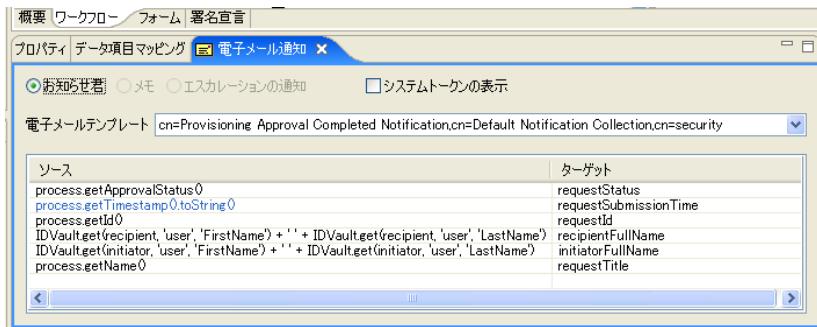
電子メールテンプレートを選択し、電子メールテンプレートに含まれる名前付きパラメータの値を設定するために式を指定するには、[電子メール通知] ビューを使用します。新しい承認動作が開始されたとき（作業があることを承認者に通知するため）、および完了

動作が終了したとき（ワークフローが終了したことをイニシエータに知らせるため）、電子メールが送信されます。

- 1 プロパティを設定する動作のアイコンを右クリックし、メニューから [電子メール通知の表示] を選択します。

ヒント：[PRD] メニューから [電子メール通知を表示] を選択して、[電子メール通知] タブを表示することもできます。

[電子メール通知] ビューが表示されます。



- 2 [電子メールテンプレート] フィールドをクリックし、定義済みテンプレートのリストから電子メールテンプレートを選択します。

電子メールテンプレートの編集：Designer で電子メールテンプレートを編集できます。この場合、[モデラー] で識別ポールトを選択し、[アウトライン] ビューで [デフォルト通知コレクション] までスクロールします。テンプレートを右クリックし、[テンプレートの編集] を選択します。

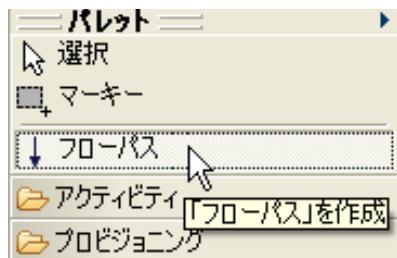
ローカライズされた電子メールテンプレート デフォルトでは、Designer はデフォルトの電子メール通知テンプレートを表示します。デフォルトのテンプレートを選択すると、電子メールはユーザのデフォルト言語で表示されます（デフォルト言語がサポートされている場合）。[すべてのローカライズされた電子メールテンプレートを表示] 初期設定を「True」に設定すれば、ユーザは Designer で、ローカライズされた電子メールテンプレートのリストから選択することができます。ローカライズされたテンプレートには、デフォルトと同じ名前が付いていますが、電子メールテンプレートの名前に Java 言語コードが付け加えられています。たとえば、cn=Provisioning Notification Activity_es, cn=Default Notification Collection, cn=security は、スペイン語版のテンプレートであることを示します。ローカライズされたテンプレートを選択した場合、電子メールは、ユーザのデフォルト言語ではなくテンプレートの言語で表示されます。

- 3 [ターゲット] のトークンに対応する [ソース] フィールドをクリックし、トークンに値を割り当てる ECMAScript 式を指定します。

電子メール通知設定の詳細については、「[187 ページの第 8 章「ワークフローアクティビティの参照」](#)」をしてください。

7.3 フローパスの追加

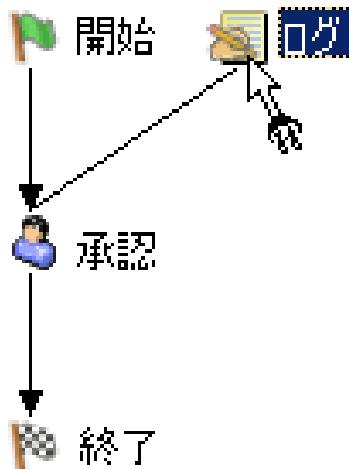
- 1 パレットでフローパスツールをクリックします。



マウスポインタがフロー・パス・ポインタに変わります。



- 2** フロー・パスを開始する動作をクリックし、次にフロー・パスを終了する動作をクリックします。



動作が接続されます。

- 3** フロー・パスを設定するには、パレットで「選択」ツールをクリックし、フロー・パスを右クリックして、「プロパティを表示する」を選択します。

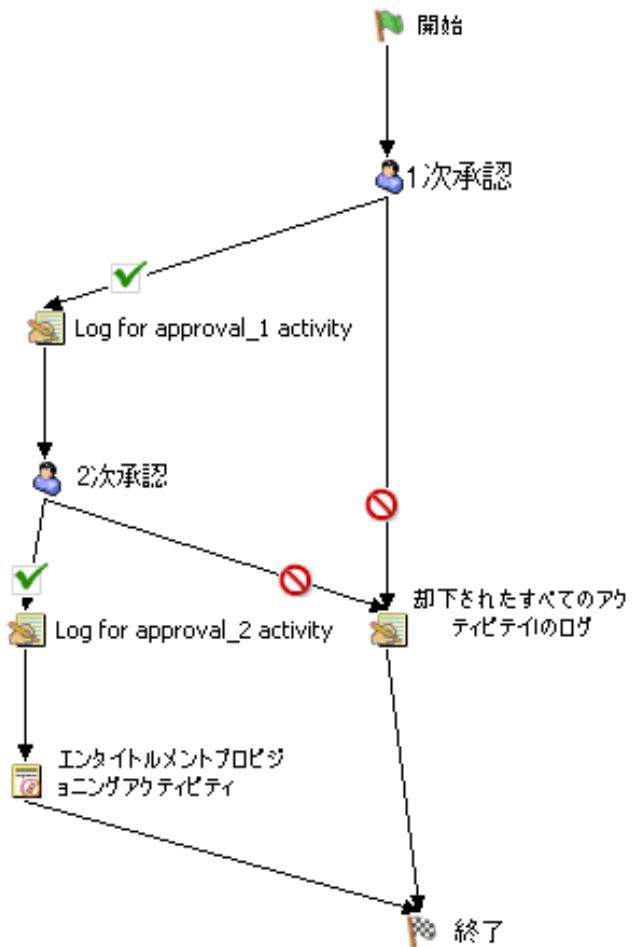
フロー・パスの設定の詳細については、「[172 ページのセクション 7.4 「フロー・パスの設定」](#)」を参照してください。

7.4 フロー・パスの設定

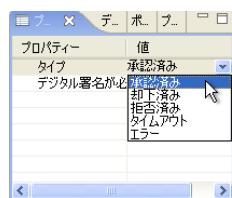
ワークフローダイアグラムにフロー・パスを追加し終えたら、パスタイプを指定できます。ワークフローにフロー・パスを追加する方法の詳細については、「[171 ページのセクション 7.3 「フロー・パスの追加」](#)」を参照してください。

フロー・パスを設定する

- 1** ワークフローダイアグラムでフロー・パスをクリックします。



- 2 [タイプ] ドロップダウンリストからいずれかのオプションを選択して、[プロパティ] タブでフロータイプを設定します。



次の表で、フローパスタイプについて説明します。

フロータイプ	説明
転送	<p>コントロールをワークフロー内の次の動作に転送します。</p> <p>転送フローパスは、次の動作を除くあらゆる動作の後で使用できます。</p> <ul style="list-style-type: none"> ◆ 承認 ◆ 条件 ◆ 終了
承認済み	<p>ユーザが要求を承認したときに行われる処理を指定します。</p> <p>承認済みフローパスは、承認動作の後でのみ有効です。</p>
却下済み	<p>ユーザが要求を却下したときに行われる処理を指定します。</p> <p>却下済みフローパスは、承認動作の後でのみ有効です。</p>
拒否済み	<p>ユーザが要求を拒否したときに行われる処理を指定します。</p> <p>拒否済みフローパスは、承認動作の後でのみ有効です。</p>
タイムアウト	<p>ユーザが応答しなかったために承認動作がタイムアウトしたときに行われる処理を指定します。</p> <p>タイムアウトフローパスは、承認動作の後でのみ有効です。</p>
エラー	<p>承認または条件動作がエラーで終了したときに行われる処理を指定します。</p> <p>エラーフローパスは、承認および条件動作の後でのみ有効です。</p>
true	<p>条件式が True になったときに行われる処理を指定します。</p> <p>True フローパスは、条件動作の後でのみ有効です。</p>
false	<p>条件式が False になったときに行われる処理を指定します。</p> <p>False フローパスは、条件動作の後でのみ有効です。</p>

[プロパティ] タブが表示されていない場合は、ワークフローダイアグラム内でフローパスを右クリックし、[プロパティを表示する] を選択します。

7.5 承認動作の宛先の指定

承認動作の宛先を指定するには、[宛先] プロパティに有効な式を入力する必要があります。動作の承認に必要な最終承認数は、[宛先] プロパティと [承認者タイプ] プロパティとの関係で決まります。

注: 承認動作の [宛先] プロパティで指定した式の評価が、存在しない DN になった場合(式が間違ってハードコードされた場合、間違って計算された場合、ユーザの選択で間違って送信された場合など)、ワークフローが実際には滞っていても、ワークフローが正常に処理していないことを示すメッセージは表示されません。アプリケーションサーバコンソールには標準の転送メッセージが表示され、コメントとフロー履歴には標準の「割り当てられた」メッセージが表示されます。この問題を回避するには、次のベストプラクティスに従うことをお勧めします。

1. 承認動作の前で条件動作を使用し、条件動作で宛先を検証します。

2. 宛先は条件動作で検証した後でも削除できるので、タイムアウト間隔と、ワークフローがタイムアウトした場合に必要なアクションを実行するリンクとを、承認動作に対して指定する必要があります。
-

7.5.1 有効な宛先式

宛先式（データ抽象化層エンティティを返す式を含む）は、ランタイム時に次のいずれかを解決される必要があります。

- ◆ 有効な個々の宛先。ユーザ DN の場合もグループ DN の場合もあります。
- ◆ 宛先の有効なリスト（たとえば、Java ベクトルオブジェクトを使用して作成）。複数のユーザ DN、複数のグループ DN、またはその両方を含めることができます。

最大承認数は、宛先数（ユーザ DN 数 + グループ DN 数）に等しく、グループの各メンバーは含まれず、カウントされません。

注：グループ DN は常に、[承認者タイプ] に関係なく、単記投票（グループのあるメンバーが動作を受けた場合、残りのメンバーはその動作を表示も要求もできなくなる）を与えるように処理されます。

次の表に、ECMA 式ビルダを使用して作成できる有効な宛先式の例を示します。

表 7-4 宛先式の例

式のタイプ	例
個々のユーザ DN	'cn=jdoe,ou=users,ou=mysample,o=myorg'
個々のグループ DN	'cn=Accounting,ou=groups,ou=mysample,o=myorg'
DN のベクトル（ユーザ DN またはグループ DN を含められる）	function DNVector() { v=new java.util.Vector(); v.add('CN=jdoe,' + USER_CONTAINER); v.add('CN=Accounting,' + GROUP_CONTAINER); v.add('CN=jsmith,' + USER_CONTAINER); v.add('CN=bsmith,' + USER_CONTAINER); return v; };

この例では、宛先の総数は 4 です（個人 3 名と会計グループのユーザ 1 名）。

7.5.2 宛先と承認者タイプとの関係

ワークフローの動作と必要な肯定承認の総数は、宛先式で指定した宛先のタイプと選択した承認者タイプに応じて異なります。

標準承認者タイプ

次の表では、さまざまなタイプの宛先が標準承認者タイプで使用されるときのワークフロー動作について説明します。

表 7-5 標準承認者タイプのワークフロー動作

宛先の値	説明
個々のユーザ DN またはエンティティ	<ul style="list-style-type: none">◆ そのユーザだけが、タスクリストにある承認動作を参照できます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
個々のグループ DN またはグループエンティティ	<ul style="list-style-type: none">◆ グループの各メンバーが、タスクリストにある動作を参照できます。◆ あるメンバーが動作を要求すると、この動作は他のメンバーのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
複数のユーザ DN またはユーザエンティティ (ユーザの仮想グループ)	使用不可。
複数のグループ DN またはグループエンティティ (グループの仮想グループ)	使用不可。
ユーザとグループの混在 (仮想グループの混在)	使用不可。

グループ DN と代理処理

ワークフローがグループに割り当てられ、電子メール通知が承認に使用される場合、グループのメンバー全員に電子メールが送信されます。代理ユーザがグループのメンバーに割り当てられる場合、次のように処理されます。

- ◆ 承認者が単一のユーザである場合、電子メール通知は両ユーザ（元のユーザと代理ユーザ）に送信されます。
- ◆ 承認者がグループ DN であり、グループ内のいずれかのユーザに代理ユーザが割り当てられる場合、代理であるユーザには、新しい要求がタスクリストに入っても電子メール通知は送信されません。

代理ユーザに電子メールで通知する場合は、承認タスクをグループのメンバーに割り当て、承認者タイプを [グループ承認者] に設定します。たとえば、以下に対して承認動作を割り当てるとしています。

```
IDVault.get('cn=Marketing,ou=groups,ou=idmsample,o=novell' ,  
'group','Member')
```

承認タイプをグループに設定すると、メンバーに代理が割り当てられている場合、各メンバーの代理に通知が送信されます。グループのメンバー 1 名が承認タスクを要求し、実行できます。これはグループ DN に直接割り当てた場合と同じ動作です。

グループ承認者タイプ

次の表では、さまざまなタイプの宛先がグループ承認者タイプで使用されるときのワークフロー動作について説明します。

表 7-6 グループ承認者タイプのワークフロー動作

宛先の値	説明
個々のユーザ DN またはエンティティ	<ul style="list-style-type: none">◆ そのユーザだけが、タスクリストにある承認動作を参照できます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
個々のグループ DN またはグループエンティティ	<ul style="list-style-type: none">◆ グループの各メンバーが、タスクリストにある動作を参照できます。◆ あるメンバーが動作を要求すると、この動作は他のメンバーのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
複数のユーザ DN またはユーザエンティティ (ユーザの仮想グループ)	<ul style="list-style-type: none">◆ 仮想グループの各ユーザが、タスクリストにある動作を参照できます。◆ 仮想グループのあるユーザが動作を要求すると、この動作は他のユーザのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
複数のグループ DN またはグループエンティティ (グループの仮想グループ)	<ul style="list-style-type: none">◆ 各グループの各メンバーが、タスクリストにある動作を参照できます。◆ 仮想グループのあるユーザが動作を要求すると、この動作はすべてのグループの他のユーザのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
ユーザとグループの混在 (仮想グループの混在)	<ul style="list-style-type: none">◆ 混在した仮想グループの各ユーザおよび各グループのメンバーが、タスクリストにある動作を参照できます。◆ 仮想グループのあるメンバーが動作を要求すると、この動作は他のメンバーのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。

複数承認者タイプ

次の表では、さまざまなタイプの宛先が複数承認者タイプで使用されるときのワークフロー動作について説明します。

表 7-7 複数承認者タイプのワークフロー動作

宛先の値	説明
個々のユーザ DN またはエンティティ	<ul style="list-style-type: none">◆ そのユーザだけが、タスクリストにある動作を参照できます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。
個々のグループ DN またはグループエンティティ	<ul style="list-style-type: none">◆ グループの各メンバーが、タスクリストにある動作を参照できます。◆ あるメンバーが動作を要求すると、この動作は他のメンバーのタスクリストから削除されます。◆ 動作を承認済みとして完了するために必要な承認は 1 つだけです。

宛先の値	説明
複数のユーザ DN または ユーザエンティティ (ユーザーの仮想グループ)	<ul style="list-style-type: none"> 仮想グループの各ユーザが、タスクリストにある動作を参照できます。 各ユーザが動作を要求できます。 動作を承認済みとして完了するには、各ユーザの承認が必要です。 1つでも却下があれば、動作は却下済みとして完了します。
複数のグループ DN または グループエンティティ (グループの仮想グループ)	<ul style="list-style-type: none"> 各グループの各メンバーが、タスクリストにある動作を参照できます。 グループのあるメンバーが動作を要求すると、この動作はそのグループに所属する他のメンバーのタスクリストから削除されます。 動作を承認済みとして完了するには、各グループが1つの承認を与える必要があります。 1つでも却下があれば、動作は却下済みとして完了します。
ユーザとグループの混在 (仮想グループの混在)	<ul style="list-style-type: none"> 混在した仮想グループの各ユーザおよび各グループの各メンバーが、タスクリストにある動作を参照できます。 各ユーザが動作を要求でき、各グループのメンバー1名が動作を要求できます (グループの他のメンバーにはそのタスクが表示されなくなります)。 動作を承認済みとして完了するには、各ユーザおよび各グループのメンバー1名が承認する必要があります。 1つでも却下があれば、動作は却下済みとして完了します。

定数承認者タイプ

次の表では、さまざまなタイプの宛先が定数承認者タイプで使用されるときのワークフロー動作について説明します。

表 7-8 定数承認者タイプのワークフロー動作

宛先の値	説明
個々のユーザ DN またはエ ンティティ	<ul style="list-style-type: none"> そのユーザだけが、タスクリストにある動作を参照できます。 動作を承認済みとして完了するために必要な承認は1つだけです。
個々のグループ DN または グループエンティティ	<ul style="list-style-type: none"> グループの各メンバーが、タスクリストにある動作を参照できます。 あるメンバーが動作を要求すると、この動作は他のメンバーのタスクリストから削除されます。 動作を承認済みとして完了するために必要な承認は1つだけです。
複数のユーザ DN または ユーザエンティティ (ユーザーの仮想グループ)	<ul style="list-style-type: none"> 仮想グループの各ユーザが、タスクリストにある動作を参照できます。 仮想グループのユーザ全員が同時に動作を要求できます。 動作を承認済みとして完了するには、絶対数または指定した割合の宛先が承認する必要があります。

宛先の値	説明
複数のグループ DN または グループエンティティ（グループの仮想グループ）	<ul style="list-style-type: none"> 各グループの各メンバーが、タスクリストにある動作を参照できます。 各グループのメンバー 1 名がタスクを要求できます（グループの他のメンバーにはそのタスクが表示されなくなります）。 動作を承認済みとして完了するには、絶対数または指定した割合の宛先が承認する必要があります。
ユーザとグループの混在 (仮想グループの混在)	<ul style="list-style-type: none"> 混在した仮想グループの各ユーザおよび各グループの各メンバーが、タスクリストにある動作を参照できます。 各ユーザが動作を要求でき、各グループのメンバー 1 名が動作を要求できます（グループの他のメンバーにはそのタスクが表示されなくなります）。 動作を承認済みとして完了するには、絶対数または指定した割合の宛先が承認する必要があります。

7.6 1 つのワークフローインスタンスによる複数の個人のプロビジョニング

1人の個人（チームマネージャなど）が、複数の個人（チームメンバー やグループなど）を1つのワークフローでプロビジョニングできるように、プロビジョニング要求定義を設定できます。プロビジョニング要求定義では、次のいずれかをプロビジョニングするよう設定できます。

- デフォルトユーザコンテナの複数の個人ユーザ
- デフォルトグループコンテナのグループの全メンバー（営業、マーケティング、人事、ITなど）
- 任意の識別ボールトコンテナの全メンバー

このタイプのワークフローを作成するには、プロビジョニング要求定義を通常どおりに作成します。[概要] パネルで、[フロー戦略] リストから [单一のフロープロビジョニングメンバ] を選択します。

7.6.1 ワークフローを使用するための基本手順

この節では、[单一のフロープロビジョニングメンバ] フロー戦略を利用するワークフローを使用するための基本手順について説明します。

- 1 ユーザアプリケーション管理者としてユーザアプリケーションにログインします。
- 2 [要求および承認] をクリックします。
- 3 [チームリソースの要求] をクリックします。



- 4 プロビジョニング要求が属するプロビジョニングカテゴリを選択し、[OK] をクリックします。

チームリソースの要求

ステップ1/4: 要求するリソースのカテゴリを選択します。

リソース検索条件:	すべて
すべて	
アカウント	
エンティティルメント	
グループ	

[続行]

人々の集団のアイコンでマークされたワークフローが表示されます。

チームリソースの要求

ステップ2/4: リストからリソースを選択します。

リソース	リソース検索条件	説明
Strategy_SingleFlowProvisionMembers	アカウント	Strategy_SingleFlowProvisionMembers

1 - 1 / 1

[戻る]

- 5 ワークフローの名前をクリックします。

フォームが表示され、プロビジョニングする複数のユーザを次の 3 つの方法で選択できます。

- ◆ 1人以上の受信者を指定する
- ◆ グループを指定する
- ◆ コンテナを指定する



6 受信者を指定して、[続行] をクリックします。

7.6.2 チームマネージャが使用するワークフローの設定

[单一のフロープロビジョニングメンバ] フロー戦略を使用したワークフローをチームマネージャが使用できるようにするには、次の追加設定手順を実行する必要があります。

- 1 管理者として iManager にログインします。
- 2 [役割とタスク] で、[プロビジョニング環境設定] を選択します。
- 3 [プロビジョニングチーム] を選択します。
- 4 まだチームを設定していない場合は設定します。
- 5 [プロビジョニング環境設定] の役割およびタスクを使用して [プロビジョニングチーム要求] を定義することにより、ワークフローをチームにバインドします。

7.7 エンティティ動作を使用した作業

エンティティ動作を使用して、識別ポールトのエンティティを更新します。エンティティ動作を使用した作業手順は、他の動作タイプを使用した作業手順とは少し異なります。

この節では、次のトピックについて説明します。

- ◆ 181 ページのセクション 7.7.1 「エンティティの追加または変更」
- ◆ 182 ページのセクション 7.7.2 「エンティティ動作を使用したエンティティの削除」
- ◆ 182 ページのセクション 7.7.3 「エンティティ動作を使用した属性または値の削除」

7.7.1 エンティティの追加または変更

- 1 [ワークフロー] ページから、パレット内のエンティティ動作アイコンをクリックし、キャンバスをクリックしてエンティティ動作をワークフローに挿入します。
- 2 [プロパティ] タブをクリックします。
- 3 [エンティティタイプ] フィールドの [値] 列をクリックし、作成または変更する [エンティティタイプ] (ユーザ、グループなど) を選択します **ステップ 6** で指定する ターゲットオブジェクトがすでに存在している場合はターゲットオブジェクトが変更され、存在していない場合は作成されます。
- 4 [操作] フィールドの [値] 列をクリックし、[作成/変更] を選択します。

- 5 [データ項目マッピング] タブをクリックします。
- 6 [エンティティDN] フィールドの隣にあるボタンをクリックして、ECMA 式ビルダを表示して、操作のターゲット（「受信者」など）を特定する式を指定します。
- 7 [OK] をクリックして、[データ項目マッピング] ビューに戻ります。
- 8 エンティティを作成するために必要な他の属性について式を指定します。
エンティティの追加の詳細については、「[43 ページのセクション 3.2 「エンティティおよび属性の操作」](#)」を参照してください。エンティティを追加する場合、必要な属性すべてについて式を入力する必要があります。

7.7.2 エンティティ動作を使用したエンティティの削除

- 1 [ワークフロー] ページから、パレット内のエンティティ動作アイコンをクリックし、キャンバスをクリックしてエンティティ動作をワークフローに挿入します。
- 2 [プロパティ] タブをクリックします。
- 3 [エンティティタイプ] フィールドの [値] 列をクリックして、削除するエンティティが属するエンティティタイプ（ユーザ、グループなど）を選択します。
- 4 [操作] フィールドの [値] 列をクリックして、[エンティティを削除します] を選択します。
- 5 [データ項目マッピング] タブをクリックします。
- 6 [エンティティDN] フィールドの隣にあるボタンをクリックして ECMA 式ビルダを表示し、削除するエンティティを特定する式を指定します。
- 7 [OK] をクリックして、[データ項目マッピング] ビューに戻ります。

7.7.3 エンティティ動作を使用した属性または値の削除

- 1 [ワークフロー] ページから、パレット内のエンティティ動作アイコンをクリックし、キャンバスをクリックしてエンティティ動作をワークフローに挿入します。
- 2 [プロパティ] タブをクリックします。
- 3 [エンティティタイプ] フィールドの [値] 列をクリックして、削除する属性または値が属するエンティティのエンティティタイプ（ユーザ、グループなど）を選択します。
- 4 [操作] フィールドの [値] 列をクリックして、[属性/ 値を削除します] を選択します。
- 5 [データ項目マッピング] タブをクリックします。
- 6 [エンティティDN] フィールドの隣にあるボタンをクリックして ECMA 式ビルダを表示し、削除する属性または値を含むエンティティを特定する式を指定します。
- 7 [OK] をクリックして、[データ項目マッピング] ビューに戻ります。
- 8 操作を適用する属性の [タイプの削除] フィールドをクリックして、リストから操作を選択します。
 - ◆ 単一値属性の場合は、[属性の削除] を選択します。
 - ◆ 複数値属性の場合は、[属性の削除] か [値の削除] のどちらかを選択します。
複数値属性について [値の削除] を選択した場合は、削除する値を特定する式を入力することも必要になります。

- 9 値を削除するには、操作を適用する属性の [値式の削除] フィールドをクリックし、削除する属性の値を求める式を指定します。

7.8 識別名参照の移植可能化

プロビジョニング要求定義の式で DN を使用する場合、プロビジョニング要求定義を構造の異なる識別ボルトに展開すると、式が失敗することがあります。通常、DN は次の場所で指定します。

- ◆ [概要] パネル: トラスティ仕様。
- ◆ ユーザ動作: 宛先およびエスカレーションの宛先。
- ◆ エンティティ動作: エンタイトルメント参照およびエンティティ DN。
- ◆ IDVault.get(dn, class, attribute) など他の多くの式。

受信者など一部の式は移植できます。ユーザ動作でデフォルトで使用されている次の式も移植できます。

```
IDVault.get(recipient, 'user', 'manager')  
IDVault.get(approval_A.getAddressee(), 'user', 'manager')
```

DN 式が識別ボルト間で移植できることを確認するために、次の変数のいずれかを使用できます。

- ◆ ROOT_CONTAINER: ou=idm-prov,o=novell など
- ◆ PROVISIONING_DRIVER: cn=UserApplication,cn=TestDrivers,o=novell など
- ◆ USER_CONTAINER: ou=users,ou=idm-prov,o=novell など
- ◆ GROUP_CONTAINER: ou=groups,ou=idm-prov,o=novell など

これらの変数は、ユーザアプリケーションのインストール中に定義され、ランタイム時に ECMAScript エンジンによって解決されます。ECMA 式ビルダのプロセスノードの下で確認できます。次の DN で、エンタイトルメントを参照する必要があるとします。

```
'cn=myEntitlement,cn=UserApplication,cn=TestDrivers,o=novell'
```

この DN を任意の識別ボルトに移植できるようにするには、次の式を使用できます。

```
'cn=MyEntitlement,' + PROVISIONING_DRIVER
```

ユーザおよびグループにもこの手法を使用できます。

注: トラスティは式でないので、この手法は使用できません。

7.9 デジタル署名サポートの設定

この節では、Designer を使用して、デジタル署名をサポートするようにプロビジョニング要求定義を設定する方法について説明します。デジタル署名をサポートするようにプロビジョニング要求定義を設定するには、次の表に示す手順に従ってください。

表 7-9 ワークフローでのデジタル署名サポートを指定する手順

手順	タスク	説明
1	1つ以上のデジタル署名宣言を作成します。	詳細については、「 185 ページのセクション 7.9.2 「署名宣言の作成」 」を参照してください。
2	プロビジョニング要求の初期化にデジタル署名が必要かどうかを指定します。	[ワークフロー] パネルで、開始動作をクリックし、次のプロパティを設定します。 <ul style="list-style-type: none">◆ [デジタル署名が必要]：「184 ページのセクション 7.9.1 「デジタル署名のワークフロープロパティ」」を参照してください。◆ [署名宣言]：ドロップダウンリストから署名宣言を選択します。手順 1（前の手順）を完了した場合にのみ、リストに項目が表示されます。
3	ワークフロー内の各承認手順でデジタル署名が必要かどうかを指定します。	承認手順ごとに複数の送信リンクを設定できます。各承認手順と各送信フローパスについて、[デジタル署名が必要] プロパティと [書名宣言] プロパティを指定する必要があります。プロパティ設定の説明については、「 184 ページのセクション 7.9.1 「デジタル署名のワークフロープロパティ」 」を参照してください。
4	要求と各承認フォームについて、タイトルコントロールを含めるかどうかを指定します。	タイトルコントロールには、[署名されたフォームドキュメントにタイトルを表示しません] というプロパティがあります。アプリケーションとデジタル署名の使用について、このプロパティを <code>true</code> に設定するか <code>false</code> に設定するかを指定します。このプロパティの詳細については、「 150 ページのセクション 6.5.19 「役職」 」を参照してください。

7.9.1 デジタル署名のワークフロープロパティ

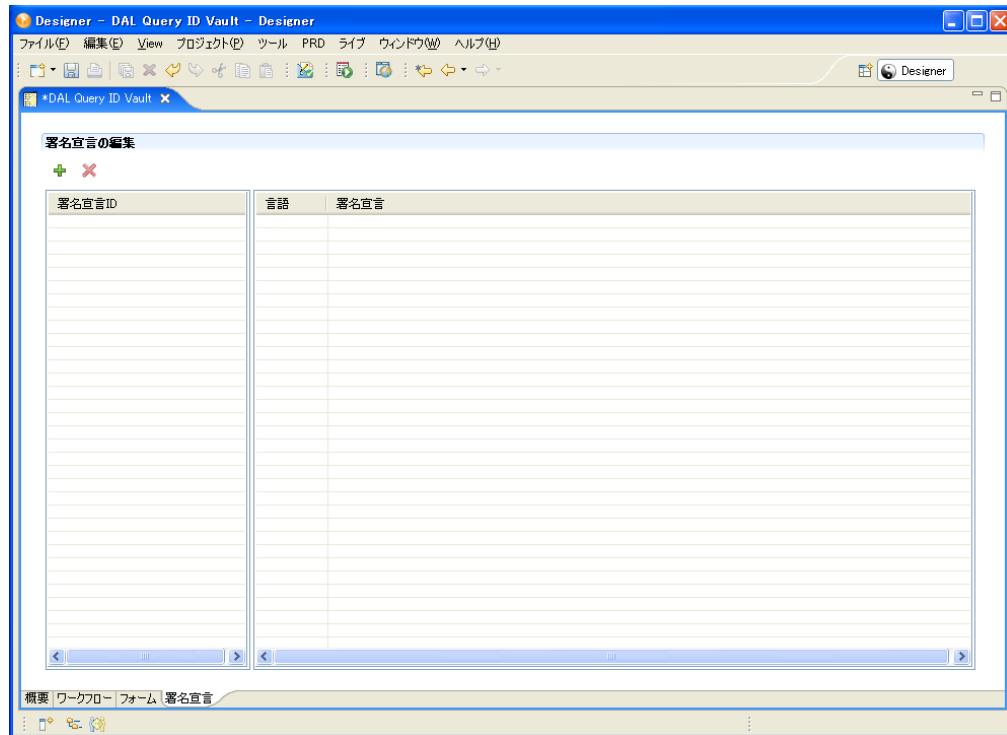
表 7-10 デジタル署名の設定

設定	説明
デジタル署名タイプ	デジタル署名のタイプとしてデータとフォームのどちらを使用するかを指定します。 <ul style="list-style-type: none">◆ データ：XML 署名がユーザの合意として機能することを指定します。[データ] を選択すると、XML データが監査ログに書き込まれます。◆ フォーム：デジタル署名宣言を含む PDF ドキュメントを生成するように指定します。このドキュメントがユーザの合意として機能します。ユーザは、要求または承認を送信する前に、生成された PDF ドキュメントをプレビューできます。[フォーム] を選択すると、PDF ドキュメント (XML にカプセル化) が監査ログに書き込まれます。

設定	説明
デジタル署名宣言	ユーザの署名を証明するデジタル署名確認文字列を指定します。詳細については、「 185 ページのセクション 7.9.2 「署名宣言の作成」 」を参照してください。

7.9.2 署名宣言の作成

- 1 [署名宣言] タブを開きます。



- 2 クリックして行を追加し、次のようにフィールドに入力します。

フィールド	説明
署名宣言 ID	署名宣言の固有識別子です。この ID は 185 ページの「デジタル署名宣言」 のドロップダウンに表示されます。
言語	言語を選択し、この言語での署名宣言の訳を指定します。署名宣言文字列は、プロビジョニングビューで [エクスポート] > [ローカリゼーションデータのエクスポート] の順に選択すると、その一部としてエクスポートされます。したがってこの宣言を送信して、残りのユーザアプリケーション表示ラベルおよび文字列の一部としてローカライズできます。
署名宣言	署名宣言としてフォームに表示する文字列です。

- 3 [保存] をクリックします。

ワークフローアクティビティの参照

8

この項では、さまざまなタイプのワークフローアクティビティを設定する方法について詳しく説明します。この項のトピックは次のとおりです。

- ◆ 187 ページのセクション 8.1 「開始アクティビティ」
- ◆ 190 ページのセクション 8.2 「承認アクティビティ」
- ◆ 199 ページのセクション 8.3 「ログアクティビティ」
- ◆ 201 ページのセクション 8.4 「プランチアクティビティ」
- ◆ 201 ページのセクション 8.5 「マージアクティビティ」
- ◆ 202 ページのセクション 8.6 「条件アクティビティ」
- ◆ 203 ページのセクション 8.7 「マッピングアクティビティ」
- ◆ 204 ページのセクション 8.8 「ワークフローステータス」
- ◆ 205 ページのセクション 8.9 「電子メールアクティビティ」
- ◆ 207 ページのセクション 8.10 「完了アクティビティ」
- ◆ 209 ページのセクション 8.11 「統合アクティビティ」
- ◆ 211 ページのセクション 8.12 「エンタイトルメントアクティビティ」
- ◆ 213 ページのセクション 8.13 「エンティティアクティビティ」

すべてのアクティビティの表示名は、アクティビティ名プロパティの「[文字列のローカライズ]」ボタン（「[33 ページのセクション 2.10 「表示ラベルのローカライズ」](#)」を参照）をクリックするとローカライズできます。アクティビティ表示名は、「[プロジェクトショーニングビュー]」で「[エクスポート]」、「[ローカリゼーションデータのエクスポート]」の順にクリックすると（「[34 ページのセクション 2.10.3 「ローカライズされたラベルのエクスポートとインポート」](#)」を参照）、その一部としてエクスポートされます。このアクティビティ名を送信して、残りのユーザアプリケーション表示ラベルおよび文字列の一部としてローカライズできます。

8.1 開始アクティビティ

開始アクティビティはワークフローで実行する最初のアクティビティです。リソースのプロジェクトショーニング要求をユーザが作成すると、実行を開始します。ユーザが要求を作成した後、開始アクティビティによって、初期要求フォームがユーザに表示されます。初期要求フォームでは、ユーザは要求の理由を示すコメントを指定するよう要求されることがあります。

アプリケーション要件に適合するように初期要求フォームをカスタマイズできます。フォームのカスタマイズの詳細については、「[107 ページの第 6 章 「プロジェクトショーニング要求定義のフォームの作成」](#)」を参照してください。

フォームをユーザに表示する前に、開始アクティビティに指定されたすべての前動作データマッピングが実行されます。

ユーザがフォームを送信した後、開始アクティビティに指定されたすべての後動作データマッピングが実行されます。通常、これらのマッピングには、フォームフィールドからフローデータオブジェクトへのデータのコピーが含まれます。

- ◆ 188 ページのセクション 8.1.1 「プロパティ」
- ◆ 189 ページのセクション 8.1.2 「データ項目マッピング」
- ◆ 190 ページのセクション 8.1.3 「電子メール通知」

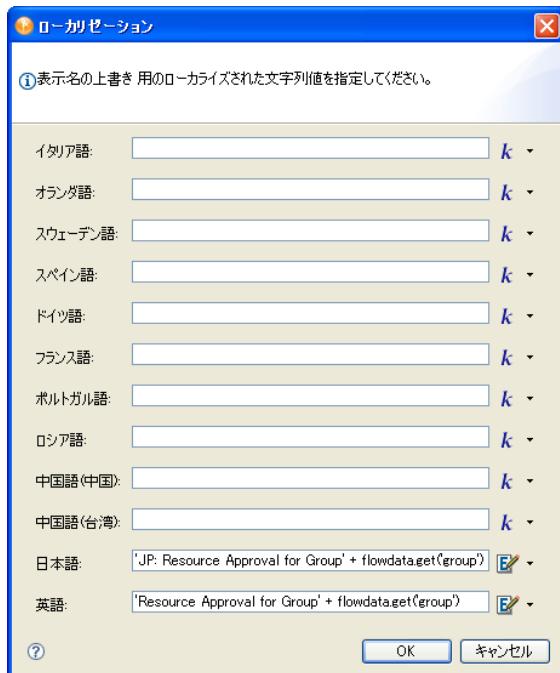
8.1.1 プロパティ

開始アクティビティのプロパティは次のとおりです。

表 8-1 開始アクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。

プロパティ名	説明
表示名の上書き	プロビジョニング要求定義の表示名を上書きできます。これは、ユーザが [マイ要求] または [チーム要求] を選択したときに [リソース] 列で表示される名前です。名前は定数にも ECMA 式の結果にもすることができ、サポートされるロケルごとにローカライズできます。定数または ECMA 式を指定するダイアログボックスにアクセスするには、[値] 列の [文字列のローカライズ] ボタンをクリックします。



定数を指定するには、**k** をクリックして、フィールドに値を入力します。値は、このフィールドに入力したとおりに表示されます。

ECMA 式を指定するには、**E** をクリックして、式を指定します。

定数と ECMA 式の両方を指定する場合、ダイアログボックスで示すように（前図）、ランタイム時には、最後に入力された値が表示されます。

デジタル署名タイプ

184 ページの 表 7-10 の [デジタル署名タイプ] を参照してください。

8.1.2 データ項目マッピング

開始アクティビティに関連付けられたデータ項目をバインドするには、前動作および後動作マッピングを定義します。前動作マッピングは、フローデータオブジェクトから取得した定数または値で要求フォームのデータを初期化します。後動作マッピングは、フォームデータをフローデータオブジェクトに戻します。

表 8-2 開始アクティビティのデータ項目マッピング

設定	説明
前動作	1つ以上の前動作マッピングを指定できます。このオプションを選択した場合、[ソース式] 列のセルをダブルクリックして、特定のターゲットフォームフィールドに使用するデータを初期要求フォームがどこから取得するかを指定できます。
	注： [前動作] オプションを選択した場合、[ターゲットフォームフィールド] 列のセルは編集できません。
後動作	1つ以上の後動作マッピングを指定できます。このラジオボタンを選択した場合、[ターゲット式] 列のセルをダブルクリックして、フォームが処理された後でデータをフォームフィールドからどこにコピーするかを指定できます。
	注： [後動作] オプションを選択した場合、[ソースフォームフィールド] 列のセルは編集できません。
ソース式	前動作マッピングのソース式を指定します。[ソース式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。
ターゲット式	後動作マッピングのターゲット式を指定します。[ターゲット式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

8.1.3 電子メール通知

このアクティビティではサポートされていません。

8.2 承認アクティビティ

承認アクティビティは、ユーザに承認フォームを表示するユーザ向きアクティビティです。承認フォームで、ユーザはプロビジョニング要求を承認、却下、または拒否できます。承認アクティビティには複数の発信フローパスを設定できますが、ランタイム時にはパスは 1 つしか実行されません。

アプリケーション要件に適合するように承認フォームをカスタマイズできます。フォームのカスタマイズの詳細については、「[107 ページの第 6 章「プロビジョニング要求定義のフォームの作成」](#)」を参照してください。

ユーザにフォームを表示する前に、承認アクティビティに指定されたすべての前動作データマッピングが実行されます。

ユーザがフォームを送信した後、承認アクティビティに指定されたすべての後動作データマッピングが実行されます。通常、これらのマッピングでは、フォームフィールドからフローデータオブジェクトへデータがコピーされます。

- ◆ [191 ページのセクション 8.2.1 「プロパティ」](#)

- ◆ 197 ページのセクション 8.2.2 「データ項目マッピング」
- ◆ 198 ページのセクション 8.2.3 「電子メール通知」

8.2.1 プロパティ

承認アクティビティには次のプロパティがあります。

表 8-3 承認アクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
受信者	アクティビティの宛先を指定するダイナミック式を指定します。名宛人は、式の値が求められる方法に基づいて、ランタイム時に特定されます。
	有効な宛先式を作成する方法および宛先が承認者タイプのプロパティを操作する方法の詳細については、「 174 ページのセクション 7.5 「承認動作の宛先の指定」 」を参照してください。
ヒント: 新しいワークフローをテストするプロセスを簡略化するために、宛先を受信者に設定できます。これにより、フォームをテストするたびにユーザーアプリケーションからログアウトして再びマネージャとしてログインする必要がなくなります。この手法は特に、ワークフローに複数レベルの承認が含まれている場合に役立ちます。テスト段階が完了した後、宛先を正しい値に変更できます。	
ECMA 式の作成の詳細については、「 269 ページの第 10 章 「ECMA 式の使用」 」を参照してください。ワークフローで使用できるシステム変数の詳細については、「 89 ページのセクション 4.3.3 「ワークフローデータについての理解」 」を参照してください。	

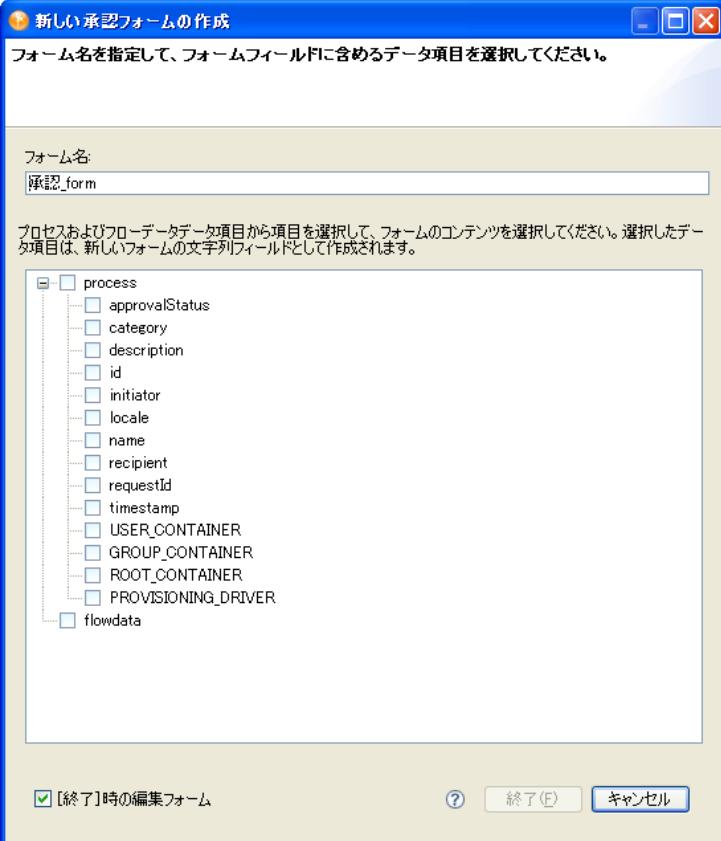
プロパティ名	説明
メモの開始	<p>最初のメモ電子メールを送信する時間をミリ秒単位で定義するダイナミック式を指定します。開始値は、アクティビティに関連付けられた最初の割り当てがかった時間からのオフセットです。</p> <p>ECMA Expression Builder の [ECMAScript 変数] ペインで、一般的な間隔（時間、日、週など）を表す事前定義された式を選択できます。</p> <p>これはメモ電子メール機能の一部です。このアクティビティが重要だと考えられ、至急実行する必要がある場合は、メモ電子メールをアクティビティ宛先に送信するようにアクティビティを設定できます。たとえば、アクティビティがタイムアウトする 5 日前から、アクティビティがタイムアウトするまで毎日、メモ電子メールを送信するようにメモ設定を行えます。このように設定するには、[メモの開始] 時間、[メモの間隔]、および送信する電子メールを指定します（「198 ページのセクション 8.2.3 「電子メール通知」」を参照）。</p> <p>ECMA 式の作成の詳細については、「269 ページの第 10 章 「ECMA 式の使用」」を参照してください。ワークフローで使用できるシステム変数の詳細については、「89 ページのセクション 4.3.3 「ワークフローデータについての理解」」を参照してください。</p>
メモの間隔	メモ電子メールを送信する間隔を定義するダイナミック式を指定します。ECMA Expression Builder の [ECMAScript 変数] ペインで、一般的な間隔（時間、日、週など）を表す事前定義された式を選択できます。
エスカレーションの宛先	承認者タイプが「複数」または「定数」である場合は使用できません。
	タイムアウト制限に達した場合に、このタスクを取得するユーザを特定するダイナミック式を指定します。
	エスカレーションの宛先は、式の値が求められる方法に基づいて、オンライン時に特定されます。
	ECMA 式の作成の詳細については、「 269 ページの第 10 章 「ECMA 式の使用」 」を参照してください。ワークフローで使用できるシステム変数の詳細については、「 89 ページのセクション 4.3.3 「ワークフローデータについての理解」 」を参照してください。

プロパティ名	説明
エスカレーションの回数	<p>承認者タイプが「複数」または「定数」である場合は使用できません。</p> <p>タイムアウト時にアクティビティが再試行する回数を指定します。</p> <p>アクティビティのタイムアウトが発生した場合、アクティビティに指定されたエスカレーションの回数に応じて、ワークフロープロセスがアクティビティを再試行できます。再試行のたびに、ワークフロープロセスはアクティビティを別のユーザにエスカレートすることができます。この場合、アクティビティは、別のユーザ（ユーザのマネージャなど）に再割り当てされた、アクティビティの作業を完了する機会をこのユーザに与えます。最後の再試行がタイムアウトした場合、アクティビティに指定された最終タイムアウトアクションに応じて、承認済み、却下済み、拒否済み、タイムアウト、またはエラーとしてアクティビティをマークできます。</p> <p>〔タイムアウト〕間隔（この表の〔タイムアウト〕を参照）は、〔エスカレーション間隔〕に優先します。たとえば、タイムアウトを10分に設定し、〔エスカレーションの回数〕を3に、〔エスカレーション間隔〕を5に指定した場合、アクティビティは一部の再試行を試みずに10分後に終了します。この例では、2番目の再試行がキャンセルされ、ワークフローはアクティビティの処理を終了します。アクティビティが終了すると、ワークフローエンジンは、最終タイムアウトアクションで定義されているリンクに従います。</p>
エスカレーション間隔	<p>承認者タイプが「複数」または「定数」である場合は使用できません。</p> <p>宛先がタスクを完了するまでの割り当て時間を定義するダイナミック式を指定します。エスカレーション間隔は、アクティビティが宛先により実行されるたびに適用されます。</p> <p>〔タイムアウト〕間隔（この表の〔タイムアウト〕を参照）は、〔エスカレーション間隔〕に優先します。たとえば、タイムアウトを10分に設定し、〔エスカレーションの回数〕を3に、〔エスカレーション間隔〕を5に指定した場合、アクティビティは一部の再試行を試みずに10分後に終了します。この例では、2番目の再試行がキャンセルされ、ワークフローはアクティビティの処理を終了します。アクティビティが終了すると、ワークフローエンジンは、最終タイムアウトアクションで定義されているリンクに従います。</p>
ECMA式	<p>ECMA式の作成の詳細については、「269ページの第10章」「ECMA式の使用」を参照してください。ワークフローで使用できるシステム変数の詳細については、「89ページのセクション4.3.3「ワークフローデータについての理解」」を参照してください。</p>

プロパティ名	説明
エスカレーションの通知の開始	承認者タイプが「複数」または「定数」である場合は使用できません。 最初のメモ電子メール（この表の「メモの開始」を参照）を「エスカレーションの宛先」に送信する時間を定義するダイナミック式を指定します。開始値は、エスカレーション割り当てがあつた時間からのオフセットです。ECMA Expression Builder の「[ECMAScript 変数]」ペインで、一般的な間隔（時間、日、週など）を表す事前定義された式を選択できます。
エスカレーションの通知の間隔	承認者タイプが「複数」または「定数」である場合は使用できません。 最初のエスカレーションの通知を送信してから、メッセージを「エスカレーションの宛先」に送信する頻度を定義するダイナミック式を指定します。ECMA Expression Builder の「[ECMAScript 変数]」ペインで、一般的な間隔（時間、日、週など）を表す事前定義された式を選択できます。
最終タイムアウトアクション	ワークフローがタイムアウトした場合の要求の最終状態を特定します。次の選択肢があります。 <ul style="list-style-type: none"> ◆ 承認済み ◆ 却下済み ◆ 拒否済み ◆ タイムアウト ◆ エラー
Timeout	宛先がタスクを完了するまでの割り当て時間を定義するダイナミック式を指定します。タイムアウト間隔は、アクティビティが名宛人により実行されるたびに適用されます。 タイムアウト設定は、「エスカレーションの回数」および「エスカレーション間隔」の値に優先します。1つ以上のエスカレーション試行が行われる前に、アクティビティの「タイムアウト」の設定値に達した場合、アクティビティは、それらのエスカレーション試行を実行することなく処理を終了します。たとえば、タイムアウトを10分に設定し、「エスカレーションの回数」を3に、「エスカレーション間隔」を5に指定した場合、アクティビティは一部のエスカレーション試行を試みずに10分後に終了します。この例では、2番目のエスカレーション試行がキャンセルされ、ワークフローはアクティビティの処理を終了します。アクティビティが終了すると、ワークフローエンジンは、最終タイムアウトアクションで定義されているリンクに従います。

ECMA式の作成の詳細については、「[269ページの第10章「ECMA式の使用」](#)」を参照してください。ワークフローで使用できるシステム変数の詳細については、「[89ページのセクション4.3.3「ワークフローデータについての理解」](#)」を参照してください。

プロパティ名	説明
時間単位	タイムアウト間隔で使用される測定単位を特定します。次の選択肢があります。
フォーム	ランタイム時に表示する承認フォームの名前を指定するか、新しいフォームを定義します。使用するフォームの名前を指定するか、[新しいフォームの作成] を選択します。新しいフォームの作成を選択した場合は、[新しいフォームの作成] ウィザードが起動し、次のように表示されます。



データ項目リストからフォームに含めるデータ項目を選択し、[完了] をクリックします。[承認フォーム] ウィザードは、選択したデータ項目それぞれを、新しいフォームの文字列タイプフィールドとして生成します。

承認アクティビティはフォームが関連付けられている必要があります。フォームが指定されていない場合、ランタイム時にエラーメッセージが表示されます。

プロパティ名	説明
要求者を除外	要求者が自分のプロビジョニング要求を承認できるかどうかを指定します。要求者が自分のプロビジョニング要求を承認できるように許可する場合は、[True] を選択します。許可しない場合は、[False] を選択します。
承認者タイプ	<p>許可する宛先数と、このアクティビティで強制される承認パターンを指定します。次の選択肢があります。</p> <ul style="list-style-type: none"> ◆ 標準: 承認を完了するには、宛先によるアクションが必要です。 ◆ グループ: 承認を完了するには、グループ内の 1 つの宛先によるアクションが必要です。 ◆ 複数: 承認を完了するには、宛先すべてによるアクションが必要です。 <p>複数承認者タイプでは、後動作データ項目マッピングを使用できません。</p> <ul style="list-style-type: none"> ◆ 定数: 承認を完了するには、一定割合の宛先または絶対数の宛先（この表の [定数] プロパティを参照）によるアクションが必要です。 <p>定数承認者タイプでは、後動作データ項目マッピングを使用できません。</p>
電子メールで通知	<p>[承認者タイプ] プロパティが [宛先] プロパティを操作する方法の詳細については、「174 ページのセクション 7.5 「承認動作の宛先の指定」」を参照してください。</p> <p>このアクティビティで電子メール通知を送信するかどうかを指定します。電子メールで通知する場合は [True] に設定します。通知しない場合は [False] に設定します。</p> <p>[電子メール通知] タブを使用して送信する電子メールを指定します（「198 ページのセクション 8.2.3 「電子メール通知」」を参照）。</p> <p>この機能を使用するには、プロビジョニング要求定義の [電子メールで参加者に通知] パラメータを [True] に設定する必要があります。103 ページの 15-§ 「概要プロパティ」</p>
定数	<p>承認者タイプが「標準」、「グループ」、または「複数」である場合は使用できません。</p> <p>定数值を指定できます。または、定数に達するまでに必要になる一定割合（「75%」など）の承認、または定数に達するまでに必要になる絶対数（「3」など）の承認を指定する ECMA 式を作成できます。</p>
デジタル署名タイプ	<p>184 ページの 表 7-10 の [デジタル署名タイプ] を参照してください。</p>
優先度	<p>承認アクティビティの優先度を定義するダイナミック式を指定します。有効な優先度値は、1、2、または 3 です。ワークフローデータから優先度を決定する式を定義することもできます。たとえば、<code>flowdata.get ("Priority")</code> と定義します。</p> <p>ユーザアプリケーションでは、ユーザは、タスクの優先度の値でタスクリストをソートできます。</p>

注: グループ DN への委任を有効にするには、承認者タイプは「グループ」または「標準」にできますが、宛先値は、そのグループに所属する各メンバーのユーザ DN を返す式にする必要があります。次に例を示します。IDVault.get(groupdn, 'sales', 'members')

8.2.2 データ項目マッピング

承認アクティビティに関連付けられたデータ項目をバインドするには、前動作および後動作マッピングを定義します。前動作マッピングは、定数、フローデータオブジェクトから取得した値、システムプロセス変数、システムアクティビティ変数、およびディレクトリ抽象化レイヤに式をコールして取得したデータで、承認フォームのデータを初期化します。後動作マッピングは、フォームデータをフローデータオブジェクトに戻します。

表 8-4 承認アクティビティのデータ項目マッピング

設定	説明
前動作	1つ以上の、前動作マッピングを指定できます。このオプションを選択した場合、[ソース式] 列のセルをダブルクリックして、特定のターゲットフォームフィールドに使用するデータを承認フォームがどこから取得するかを指定できます。 注: [前動作] 選択肢を選択した場合、[ターゲットフォームフィールド] 列のセルは編集できません。
後動作	1つ以上の後動作マッピングを指定できます。このオプションを選択した場合、[ターゲット式] 列のセルをダブルクリックして、フォームが処理された後でデータをフォームフィールドからどこにコピーするかを指定できます。 「複数」および「定数」の承認者タイプでは、後動作マッピングを使用できません（ 191 ページのセクション 8.2.1 「プロパティ」 を参照）。 承認アクティビティのフォームには、apwaComment という特別な内部コントロールが含まれています。このコントロールにより、ユーザメントがワークフローデータベースに書き込まれます。このコントロールには後動作マッピングを設定しないでください。このコントロールの詳細については、 137 ページのセクション 6.5.9 「DNMaker」 を参照してください。
ソース式	前動作マッピングのソース式を指定します。[ソース式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。
ターゲット式	後動作マッピングのターゲット式を指定します。[ターゲット式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。

ECMA 式の作成の詳細については、「[269 ページの第 10 章 「ECMA 式の使用」](#)」を参照してください。

8.2.3 電子メール通知

承認アクティビティの電子メール通知を有効にするには、使用する電子メールテンプレートと、電子メール本文のターゲットトークンを求めるソース式を指定する必要があります。

表 8-5 承認アクティビティの電子メール通知設定

設定	説明
お知らせ君	この電子メール通知が通知電子メールであることを指定します。
メモ	この電子メール通知がメモ電子メールであることを指定します。
再試行メモ	この電子メール通知が再試行メモ電子メールであることを指定します。
システムトークンの表示	[ターゲット] 列にシステムトークン (TO、CC、BCC、REPLYTO など) を表示します。
電子メールテンプレート	使用する電子メールテンプレートの名前を指定します。デフォルトで、承認アクティビティはプロビジョニング通知テンプレートを使用します。
	Designer で電子メールテンプレートを編集できます。詳細については、 171 ページの「電子メールテンプレートの編集」 を参照してください。

設定	説明
ソース/ターゲット	<p>電子メール本文でのターゲットトークンのソース式を指定します。</p> <p>ターゲットトークンのリストは、選択した電子メールテンプレートによって決まります。新しいトークンを追加することはできませんが、自分のソース式を作成することによってトークンに値を割り当てることができます。ランタイム時に、ソース式が評価され各トークンの値が決定します。</p> <p>次に、使用できるターゲットトークンを示します。</p> <ul style="list-style-type: none">◆ TO◆ CC◆ BCC◆ REPLYTO◆ recipientFullName◆ initiatorFullName◆ requestTitle◆ userFirstName

プロビジョニング要求定義テンプレートを使用してワークフローを作成する場合、各トークンにはデフォルトのソース式が与えられます。デフォルトの式は、ワークフロープロセス（プロセスオブジェクト）から、またはデータ抽象化レイヤ（IDVault オブジェクト）から値を取得します。アプリケーション要件に適合するようにこれらの式を変更できます。

注: リソース要求ポートレットで使用するワークフローを作成し、TO トークンの式として「_default_」を使用する場合、宛先式は IDVault 式にする必要があります。

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

注: 電子メール通知は、[概要] タブで [電子メールで参加者に通知] チェックボックスが選択され、承認アクティビティの [電子メールで通知] プロパティが [True] に設定されているときにのみサポートされます。

8.3 ログアクティビティ

ログアクティビティはメッセージをログに書き込むシステムアクティビティです。ワークフロープロセスの状態に関する情報をログとして記録する場合、ワークフローシステムは Novell® Audit とやり取りします。

注: Novell Audit は、ログおよびレポート機能を追加するために Novell Sentinel へ情報を送信するように設定できます。

処理中、ワークフローは発生したさまざまなイベントに関する情報をログとして記録できます。ログデータは、Novell レポートツールを使用して確認できます。

ログを使用する前に、ユーザアプリケーションでログを有効にする必要があります。

注: ワークフローの実行中に、ログアクティビティでは制御されない多くのシステムイベントがログに記録されます。たとえば、ワークフローシステムは、ワークフローが開始または停止したときや、承認、却下、拒否されたときにメッセージをログに書き込みます。

- ◆ 200 ページのセクション 8.3.1 「プロパティ」
- ◆ 200 ページのセクション 8.3.2 「データ項目マッピング」
- ◆ 201 ページのセクション 8.3.3 「電子メール通知」

8.3.1 プロパティ

ログアクティビティには次のプロパティがあります。

表 8-6 ログアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
Audit	ログメッセージを送信するかどうかを指定します。このプロパティを [True] に設定した場合、メッセージは Novell Audit を含むすべての log4j チャネルに送信されます。このプロパティを [False] に設定した場合、ログメッセージは送信されません。
著者	メッセージの著者を定義します。デフォルトでは、著者はプロビジョニング要求のイニシエータになっています。
メッセージ	ログメッセージのテキストを定義する ECMA 式を指定します。通常、このテキストは、このログアクティビティがプロセス内で実行されている箇所を示し、ログをわかりやすくするその他の情報を記しています。 ECMA 式の作成の詳細については、「 269 ページの第 10 章「ECMA 式の使用」 」を参照してください。ワークフローで使用できるシステム変数の詳細については、「 89 ページのセクション 4.3.3 「ワークフローデータについての理解」 」を参照してください。
コメント	ユーザコメントで表示できるテキストを定義する ECMA 式を指定します。コメントは、要求の理由や要求の完了した承認ステータスを記録するために使用できます。次に例を示します。 "Reason for request: "+ flowdata.get(ÅfreasonÅf) または "Process has been " + flowdata.get(IDM_COMPLETED_APPROVAL_STATUSÅf)

8.3.2 データ項目マッピング

このアクティビティではサポートされていません。

8.3.3 電子メール通知

このアクティビティではサポートされていません。

8.4 ブランチアクティビティ

並行処理をサポートするワークフローでは、ブランチアクティビティにより、複数のユーザが作業項目の異なる領域で同時に作業できます。ユーザが作業を完了すると、マージアクティビティにより、フローに合流するブランチが同期化されます。

ワークフローには複数のブランチアクティビティを設定できますが、ブランチアクティビティごとにマージアクティビティが関連付けられている必要があります。ブランチアクティビティから分かれ出たすべてのフローパスが実行されます。

ブランチアクティビティは、実行している間、ブランチ間の同期をサポートしません。各ブランチは、別のブランチで更新されているデータから独立している必要があります。データ同期はマージアクティビティによって強制されます。マージアクティビティが完了すると、ブランチで設定されているデータはすべて使用できます。

- ◆ 201 ページのセクション 8.4.1 「プロパティ」
- ◆ 201 ページのセクション 8.4.2 「データ項目マッピング」
- ◆ 201 ページのセクション 8.4.3 「電子メール通知」

8.4.1 プロパティ

ブランチアクティビティのプロパティは次のとおりです。

表 8-7 ブランチアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。

8.4.2 データ項目マッピング

このアクティビティではサポートされていません。

8.4.3 電子メール通知

このアクティビティではサポートされていません。

8.5 マージアクティビティ

並行処理をサポートするワークフローでは、マージアクティビティにより、フローに合流するブランチが同期化されます。マージアクティビティはブランチアクティビティとともに使用されます。このブランチアクティビティによって、2人のユーザが作業項目の異なる領域で同時に作業できます。ユーザが作業を完了すると、マージアクティビティにより、合流するブランチが同期化されます。

ワークフローには複数のブランチアクティビティを設定できますが、ブランチアクティビティごとにマージアクティビティが関連付けられている必要があります。

- ◆ 202 ページのセクション 8.5.1 「プロパティ」
- ◆ 202 ページのセクション 8.5.2 「データ項目マッピング」
- ◆ 202 ページのセクション 8.5.3 「電子メール通知」

8.5.1 プロパティ

マージアクティビティのプロパティは次のとおりです。

表 8-8 マージアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。

8.5.2 データ項目マッピング

このアクティビティではサポートされていません。

8.5.3 電子メール通知

このアクティビティではサポートされていません。

8.6 条件アクティビティ

条件アクティビティでは、ワークフローに条件付きロジックを追加できます。このロジックは、ワークフローが実行するときに行われる処理を制御するために使用できます。条件アクティビティでは、ブール値を評価する ECMA 式としてロジックを定義します。

各条件アクティビティには 2 つの発信フローパスが必要です。1 つは True と評価する条件を扱い、もう 1 つは False と評価する条件を扱います。オプションで、ECMA 式評価が失敗した場合に起きたるエラー条件を扱うために、3 番目のフローパスを追加できます。

- ◆ 202 ページのセクション 8.6.1 「プロパティ」
- ◆ 203 ページのセクション 8.6.2 「データ項目マッピング」
- ◆ 203 ページのセクション 8.6.3 「電子メール通知」

8.6.1 プロパティ

条件アクティビティには次のプロパティがあります。

表 8-9 条件アクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。

プロパティ名	説明
条件式	<p><code>True</code> または <code>False</code> を返す ECMA 式を指定します。返される値によって、アクティビティ実行終了後、どのフローパスに進むか決定されます。</p> <p>ヒント: 条件式で 2 つのオブジェクトが等しいかどうかをテストする必要がある場合、比較しているオブジェクトが同じタイプの Java オブジェクトであることが確かな場合を除き、<code>equals()</code> メソッドではなく <code>==</code> 演算子を使用する必要があります。たとえば、次の式を使用します：</p> <pre>(approval_A.getAction() == "DENIED")</pre> <p>次の式の代わりです：</p> <pre>(approval_A.getAction()).equals("DENIED")</pre>

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。ワークフローで使用できるシステム変数の詳細については、「[89 ページのセクション 4.3.3 「ワークフローデータについての理解」](#)」を参照してください。

8.6.2 データ項目マッピング

このアクティビティではサポートされていません。

8.6.3 電子メール通知

このアクティビティではサポートされていません。

8.7 マッピングアクティビティ

マッピングアクティビティでは、ワークフロー内のデータを追加または操作できます。マッピングアクティビティはソース式を評価して、結果を関連するデータ項目のターゲット式に保存します。このアクティビティは、並行処理される承認フォームのデータを、データがフローデータに移動された後に結合する方法として使用できます。

たとえば、並行承認コンテキストにおいて、相互に依存していたり、相互に計算したりする必要がある複数の承認フォームからデータを収集する必要がある場合があります。これを行うには、マッピングアクティビティをマージアクティビティの後、結果を使用するすべてのアクティビティ（条件、エンティティ、プロビジョニング、または別の承認アクティビティなど）の前に置きます。

また、マッピングアクティビティを使用して、データを操作しリソース集約的である外部 Java ルーチンへのコールを独立させることもできます。このため、前動作マッピング段階でも後動作マッピング段階でも、ユーザベースの承認アクティビティの処理は低下しません。

8.7.1 プロパティ

マッピングアクティビティのプロパティは次のとおりです。

表 8-10 マッピングアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。

8.7.2 データ項目マッピング

マッピングアクティビティに関連付けられたデータ項目をバインドするには、前動作および後動作マッピングを定義します。前動作マッピングは、定数、フローデータオブジェクトから取得した値、システムプロセス変数、システムアクティビティ変数、またはディレクトリ抽象化レイヤに式をコールして取得したデータで、フローデータ内のデータを初期化します。後動作マッピングは、データをフローデータオブジェクトに移動させます。

表 8-11 マッピングアクティビティのデータ項目マッピング

設定	説明
ソース式	ソース式を指定します。[ソース式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。例： <pre>function list () { s=new java.lang.String () ; if (wi.XPath ('count (flow-data/groups) ') > 0) s="There was a group selected"; return s;} list () ;</pre>
ターゲット式	ターゲット式を指定します。[ターゲット式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。または「すべてマップ」ボタンをクリックできます。ターゲット式の例： <code>flowdata.testexpression</code>

8.7.3 電子メール通知

このアクティビティではサポートされていません。

8.8 ワークフローステータス

ワークフローステータスアクティビティでは、プロビジョニングアクティビティを含まないワークフロー（エンタイトルメントまたはエンティティ）に対して承認ステータス（承認済みまたは却下済み）を指定できます。

- ◆ [204 ページのセクション 8.8.1 「プロパティ」](#)
- ◆ [205 ページのセクション 8.8.2 「データ項目マッピング」](#)
- ◆ [205 ページのセクション 8.8.3 「電子メール通知」](#)

8.8.1 プロパティ

ワークフローステータスアクティビティには次のプロパティがあります。

表 8-12 ワークフローステータスアクティビティのプロパティ

プロパティ	説明
名前	アクティビティの名前を指定します。
ワークフローステータス	承認済みまたは却下済みのどちらかの式として承認ステータスを指定します。

8.8.2 データ項目マッピング

このアクティビティではサポートされていません。

8.8.3 電子メール通知

このアクティビティではサポートされていません。

8.9 電子メールアクティビティ

電子メールアクティビティにより、承認プロセス外の関連当事者に電子メールを送信できます。

- ◆ 205 ページのセクション 8.9.1 「プロパティ」
- ◆ 206 ページのセクション 8.9.2 「データ項目マッピング」
- ◆ 206 ページのセクション 8.9.3 「電子メール通知」

8.9.1 プロパティ

電子メールアクティビティには次のプロパティがあります。

表 8-13 電子メールアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
電子メールで通知	このアクティビティで電子メール通知を送信するかどうかを指定します。電子メールで通知する場合は [True] に設定します。通知しない場合は [False] に設定します。 [電子メール通知] タブを使用して送信する電子メールを指定します（「206 ページのセクション 8.9.3 「電子メール通知」」を参照）。 この機能を使用するには、プロジェクト定義の [電子メールで参加者に通知] パラメータを [True] に設定する必要があります（「103 ページの 15-1§ 「概要プロパティ」」を参照）。

8.9.2 データ項目マッピング

このアクティビティではサポートされていません。

8.9.3 電子メール通知

このアクティビティの電子メール通知を有効にするには、使用する電子メールテンプレートと、電子メール本文のターゲットトークンを求めるソース式を指定する必要があります。

表 8-14 電子メールアクティビティの電子メール通知設定

設定	説明
電子メールテンプレート	使用する電子メールテンプレートの名前を指定します。デフォルトで、承認アクティビティはプロビジョニング通知テンプレートを使用します。 Designer で電子メールテンプレートを編集できます。詳細については、 171 ページの「電子メールテンプレートの編集」 を参照してください。

設定	説明
ソース/ターゲット	<p>電子メール本文でのターゲットトークンのソース式を指定します。</p> <p>ターゲットトークンのリストは、選択した電子メールテンプレートによって決まります。新しいトークンを追加することはできませんが、自分のソース式を作成することによってトークンに値を割り当てられます。ランタイム時に、ソース式が評価され各トークンの値が決定します。</p> <p>次に、使用できるターゲットトークンを示します。</p> <ul style="list-style-type: none"> ◆ TO ◆ CC ◆ BCC ◆ REPLYTO ◆ recipientFullName ◆ initiatorFullName ◆ requestTitle ◆ userFirstName <p>プロビジョニング要求定義テンプレートを使用してワークフローを作成する場合、各トークンにはデフォルトのソース式が与えられます。デフォルトの式は、ワークフロープロセス（プロセスオブジェクト）から、またはデータ抽象化レイヤ（IDVault オブジェクト）から値を取得します。アプリケーション要件に適合するようにこれらの式を変更できます。</p>

注: リソース要求ポートレットで使用するワークフローを作成し、TO トークンの式として「_default_」を使用する場合、宛先式は IDVault 式にする必要があります。

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

注: 電子メール通知は、[概要] タブで [電子メールで参加者に追加] チェックボックスが選択され、アクティビティの [電子メールで通知] プロパティが [True] に設定されているときにのみサポートされます。

8.10 完了アクティビティ

完了アクティビティはワークフローの完了をマークします。完了アクティビティが実行すると、ワークフローが完了したことを参加者に通知する電子メールメッセージが送信されます。

- ◆ [208 ページのセクション 8.10.1 「プロパティ」](#)
- ◆ [208 ページのセクション 8.10.2 「データ項目マッピング」](#)
- ◆ [208 ページのセクション 8.10.3 「電子メール通知」](#)

8.10.1 プロパティ

完了アクティビティには次のプロパティがあります。

表 8-15 完了アクティビティのプロパティ

プロパティ	説明
名前	アクティビティの名前を指定します。
電子メールで通知	完了アクティビティを実行するときに電子メール通知を開始する方法を指定します。このプロパティを [True] に設定した場合、電子メール通知が送信されます。このプロパティを [False] に設定した場合、電子メール通知は送信されません。 電子メール通知の設定の詳細については、「 208 ページのセクション 8.10.3 「電子メール通知」 」を参照してください。

8.10.2 データ項目マッピング

このアクティビティではサポートされていません。

8.10.3 電子メール通知

完了アクティビティの電子メール通知を有効にするには、使用する電子メールテンプレートと、電子メール本文のターゲットトークンを求めるソース式を指定する必要があります。

表 8-16 完了アクティビティの電子メール通知設定

設定	説明
電子メールテンプレート	使用する電子メールテンプレートの名前を指定します。デフォルトで、完了アクティビティはプロビジョニング承認完了通知テンプレートを使用します。 Designer で電子メールテンプレートを編集できます。詳細については、「 171 ページの「電子メールテンプレートの編集」 」を参照してください。

設定	説明
ソース	電子メール本文でのターゲットトークンのソース式を指定します。
ターゲット	<p>ターゲットトークンのリストは、選択した電子メールテンプレートによって決まります。新しいトークンを追加することはできませんが、自分のソース式を作成することによって事前定義されたトークンに値を割り当てられます。ランタイム時に、ソース式が評価され各トークンの値が決定します。</p> <p>プロビジョニング承認完了通知電子メールテンプレートの使用可能なターゲットトークンは、次のとおりです。</p> <ul style="list-style-type: none"> ◆ TO ◆ CC ◆ BCC ◆ REPLYTO ◆ requestStatus ◆ requestSubmissionTime ◆ requestID ◆ recipientFullName ◆ initiatorFullName ◆ requestTitle <p>プロビジョニング要求定義テンプレートを使用してワークフローを作成する場合、各トークンにはデフォルトのソース式が与えられます。デフォルトの式は、ワークフロープロセス(プロセスオブジェクト)から、またはデータ抽象化レイヤ(IDVault オブジェクト)から値を取得します。アプリケーション要件に適合するようにこれらの式を変更できます。</p> <p>注: リソース要求ポートレットで使用するワークフローを作成し、TO トークンの式として「_default_」を使用する場合、宛先式は IDVault 式にする必要があります。</p>

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

注: 電子メール通知は、[概要] タブで [電子メールで参加者に通知] チェックボックスが選択されている場合にのみサポートされます。

8.11 統合アクティビティ

統合アクティビティでは、Web サービスを使用してワークフローデータを処理できます。統合アクティビティの使用の詳細については、「[217 ページの第 9 章「統合アクティビティの操作」](#)」を参照してください。

- ◆ [210 ページのセクション 8.11.1 「プロパティ」](#)
- ◆ [210 ページのセクション 8.11.2 「データ項目マッピング」](#)
- ◆ [211 ページのセクション 8.11.3 「電子メール通知」](#)

8.11.1 プロパティ

統合アクティビティには次のプロパティがあります。

表 8-17 統合アクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
WSDL リソース	統合アクティビティで使用する Web サービスの WSDL ファイルを指定します。このファイルを指定した後に、WSDL がプロビジョニング要求定義ファイルに組み込まれます。
Timeout	WSDL ファイルを選択すると、統合アクティビティで使用する Web サービスポートタイプと操作を選択するダイアログボックスが表示されます。
再試行回数	統合アクティビティが完了するまでの割り当て時間を定義するダイナミック式を指定します。タイムアウト間隔は、アクティビティが名宛人により実行されるたびに適用されます。
最終タイムアウトアクション	ECMA 式の作成の詳細については、「 269 ページの第 10 章「ECMA 式の使用」 」を参照してください。ワークフローで使用できるシステム変数の詳細については、「 89 ページのセクション 4.3.3 「ワークフローデータについての理解」 」を参照してください。
	タイムアウト時にアクティビティが再試行する回数を指定します。
	アクティビティのタイムアウトが発生した場合、アクティビティに指定された再試行回数に応じて、ワークフロープロセスがアクティビティを再試行できます。最後の再試行がタイムアウトした場合、アクティビティに指定された最終タイムアウトアクションに応じて、成功、失敗、エラー、またはタイムアウトとしてアクティビティをマークできます。
	統合アクティビティがタイムアウトした場合の要求の最終状態を特定します。次の選択肢があります。
	<ul style="list-style-type: none">◆ 成功◆ 失敗◆ エラー◆ タイムアウト

8.11.2 データ項目マッピング

統合アクティビティに関連付けられたデータ項目をバインドするには、前動作および後動作マッピングを定義します。前動作マッピングは、フローデータオブジェクトから取得した値を、統合アクティビティがアクセスする Web サービスの入力メッセージの属性にマップします。後動作マッピングは、Web サービスからの応答をフローデータオブジェクトにマップします。統合アクティビティのデータ項目マッピングの詳細については、「[219 ページのセクション 9.3 「統合アクティビティとの間のデータの移動」](#)」を参照してください。

表 8-18 統合アクティビティのデータ項目マッピング

設定	説明
前動作	<p>1つ以上の前動作マッピングを指定できます。このオプションを選択した場合、[ソース式] 列のセルをダブルクリックして、特定の Web サービス入力フィールドのデータを統合アクティビティがどこから取得するかを指定できます。</p> <p>注: [前動作] オプションを選択した場合、[Web サービス入力フィールド] 列のセルは編集できません。</p>
後動作	<p>1つ以上の後動作マッピングを指定できます。このラジオボタンを選択した場合、[ターゲット式] 列のセルをダブルクリックして、フォームが処理された後でデータを Web サービス出力フィールドからどこにコピーするかを指定できます。</p> <p>注: [後動作] オプションを選択した場合、[Web サービス出力フィールド] 列のセルは編集できません。</p>
ソース式	<p>ソース式を指定します。[ソース式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。たとえば、Web サービス入力フィールドの場合は <code>flowdata.get ('Start/RequestRate/Country1')</code>、Web サービス出力フィールドの場合は <code>flowdata.Start/RequestRate/Country1</code> です。</p>
Web サービス入力フィールド	<p>この列には、WSDL ファイルを選択したときに指定したポートタイプと操作に関するすべての入力フィールドが表示されます。この列のフィールドは自動的に入力されます。入力フィールドを削除する場合は、[マッピング] をクリックし、サンプルドキュメントのノードを開き、削除するすべての入力フィールドを選択解除します。</p>
Web サービス出力フィールド	<p>この列には、WSDL ファイルを選択したときに指定したポートタイプと操作に関するすべての出力フィールドが表示されます。この列のフィールドは自動的に入力されます。出力フィールドを削除する場合は、[マッピング] をクリックし、サンプルドキュメントのノードを開き、削除するすべての出力フィールドを選択解除します。</p>
マッピング	<p>Web サービスに対する入力または出力についてのサンプルドキュメントの階層ビューが表示されます。この機能を使用して、入力または出力フィールドを選択解除できます（デフォルトではすべての Web サービス入力および出力フィールドが選択されています）。</p>

8.11.3 電子メール通知

このアクティビティではサポートされていません。

8.12 エンタイトルメントアクティビティ

エンタイトルメントアクティビティは、ユーザまたは他のエンティティタイプのエンタイトルメントを付与または取り消します。

ワークフローには、少なくとも 1 つのエンタイトルメントかエンティティアクティビティが必要です。

- ◆ 212 ページのセクション 8.12.1 「プロパティ」
- ◆ 212 ページのセクション 8.12.2 「データ項目マッピング」
- ◆ 213 ページのセクション 8.12.3 「電子メール通知」

8.12.1 プロパティ

エンタイトルメントアクティビティには次のプロパティがあります。

表 8-19 エンタイトルメントアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
ワークフローステータスの設定	プロビジョニング要求の承認ステータスを指定します。承認済みの場合は [True] に設定します。それ以外の場合は [False] に設定します。ワークフローステータスのこの設定方法は、他の方法（[デフォルトの完了ステータスを「承認」に設定] パラメータ（「103 ページの 15-1§ 「概要プロパティ」」を参照）や承認ステータスアクティビティ（「204 ページのセクション 8.8 「ワークフローステータス」」を参照）など）に優先します。

8.12.2 データ項目マッピング

エンタイトルメントアクティビティに関連付けられたデータ項目をバインドするには、いくつかの DirXML® 属性のマッピングを定義します。

表 8-20 エンタイトルメントアクティビティのデータ項目マッピング

設定	説明
ソース式	<p>DirXML マッピングのソース式を指定します。[ソース式] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。</p> <p>次に、エンタイトルメントの DirXML マッピングについて説明します。</p> <ul style="list-style-type: none">◆ dn はエンタイトルメントの受信者の識別名です。◆ DirXML-Entitlement-DN は、実行するエンタイトルメントの識別名です。たとえば、エンタイトルメントは次のように識別されます。 <code>'CN=Groups,CN=GroupEntitlementLoopback,CN=TestDrivers,O=novell'</code>ECMA Expression Builder の [ECMAScript 変数] パネルで、ドライバ内にある全エンタイトルメントのリストを確認できます。エンタイトルメントを選択するには、エンタイトルメントの完全識別名をダブルクリックします。◆ DirXML-Entitlement-Action は、エンタイトルメントが付与されているか、取り消されているかを示します。操作によりエンタイトルメントが付与されている場合、値は 1 になります。取り消されている場合は 0 になります。◆ DirXML-Entitlement-Parameter は、エンタイトルメントドライバで必要なパラメータを指定します。たとえば、エンタイトルメント操作で販売グループへのアクセスを認める場合、次のようにパラメータでグループを指定できます。 <code>'\\MYTREE\\novell\\idmsample-doc\\groups\\Sales'</code>◆ DirXML-Entitlement-MultiValueAllowed は、エンタイトルメントで複数の値がサポートされているかどうかを示します。複数の値をサポートしている場合、値は [True] になります。サポートしていない場合は、[False] になります。

ECMA 式の作成の詳細については、「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照してください。

8.12.3 電子メール通知

このアクティビティではサポートされていません。

8.13 エンティティアクティビティ

エンティティアクティビティは、識別ボールト内のエンティティを更新します。このアクティビティを使用して、エンティティの属性を作成、変更、または削除できます。このアクティビティを使用して、エンティティを作成または削除することもできます（「[181 ページのセクション 7.7 「エンティティ動作を使用した作業」](#)」を参照）。

ワークフローには、少なくとも 1 つのエンタイトルメントかエンティティアクティビティが必要です。

- ◆ [214 ページのセクション 8.13.1 「プロパティ」](#)
- ◆ [214 ページのセクション 8.13.2 「データ項目マッピング」](#)
- ◆ [216 ページのセクション 8.13.3 「電子メール通知」](#)

8.13.1 プロパティ

エンティティアクティビティには次のプロパティがあります。

表 8-21 エンティティアクティビティのプロパティ

プロパティ名	説明
名前	アクティビティの名前を指定します。
エンティティタイプ	ターゲットエンティティタイプを「ユーザ」または「グループ」に指定します。
説明	ターゲットエンティティで実行される操作の種類を次の中から指定します。 <ul style="list-style-type: none">◆ 作成 / 変更◆ 属性 / 値を削除します◆ エンティティを削除します
ワークフローステータスの設定	エンティティの属性を作成または変更するか、新しいエンティティを作成するには、[作成 / 変更] を選択します。エンティティの属性を削除するには、[削除] を選択します。 エンティティを削除するには、[オブジェクトの削除] を選択します。 プロジェクトの承認ステータスを指定します。承認済みの場合は [True] に設定します。それ以外の場合は [False] に設定します。ワークフローステータスのこの設定方法は他の方法に優先します。たとえば、[デフォルトの完了ステータスを「承認」に設定] パラメータ (103 ページの「概要プロパティ」 を参照) や承認ステータスアクティビティ (204 ページのセクション 8.8 「ワークフローステータス」 を参照) よりも優先します。

8.13.2 データ項目マッピング

エンティティアクティビティに関連付けられたデータ項目をバインドするには、ターゲットエンティティタイプに関連付けられた属性のマッピングを定義します。

表 8-22 エンティティアクティビティのデータ項目マッピング

設定	説明
エンティティ DN	<p>操作のターゲットであるエンティティを識別します。デフォルト値は受信者です。</p> <p>新しいオブジェクトを作成するには、まだ存在しない識別名を指定します。</p> <p>ヒント: DNMaker コントロールの出力は、エンティティ DN 値の入力として使用できます。DNMaker コントロールは、ユーザがテキストフィールドにネーミング属性を入力できるようにし、コンテナ選択用のインターフェースを表示することによって、DN を構築します。このデータが要求フォームで取得されると、出力をフローデータオブジェクトの変数にマップできます。エンティティアクティビティの定義では、式を使用して、エンティティ DN 設定内でこのフローデータ変数にアクセスできます。</p> <pre>flowdata.get(ÃfgroupdnÃf);</pre> <p>DNMaker コントロールの使用の詳細については、「137 ページのセクション 6.5.9 「DNMaker」」を参照してください。</p>
タイプの変更	<p>属性に対してマッピングを実行する方法を示します。次の選択肢があります。</p> <ul style="list-style-type: none">◆ 値を追加する◆ 値を置き換える◆ すべての値を置き換える <p>多くの属性にとって「[値を置き換える]」が唯一の妥当なオプションです。したがって、このオプションが自動的に選択され、変更できません。</p> <p>「[値式の変更]」設定を指定する前に、「[タイプの変更]」設定を指定する必要があります。</p>

設定	説明
値式の変更	<p>属性のソース式を指定します。[値式の変更] 列のセルをクリックすると、式の定義に役立つ ECMA Expression Builder が表示されます。使用できる属性のリストは、[プロパティ] タブで選択したエンティティタイプに応じて異なります。</p> <p>Designer は、サンプルの ECMAScript 式をこのフィールドに自動的に挿入します。与えられるコードは、[プロパティ] で指定した [操作] プロパティと [データ項目マッピング] で選択した [タイプの変更] に応じて異なります。たとえば、[操作] に対して [作成/変更] を、[タイプの変更] に対して [すべての値を置き換える] を指定した場合、Designer は、ベクトルの作成に役立つ式を挿入します。</p> <pre>function list() { v=new java.util.Vector(); v.add('{Enter Item 1}'); v.add('{Enter Item 2}'); return v; } list();</pre> <p>場合によっては、サンプルの式と同様またはそれ以上に機能する式を作成することもできます。たとえば、複数の属性値を示すベクトルを作成する代わりに、複数の属性値を格納するフローデータ変数（「89 ページのセクション 4.3.3 「ワークフローデータについての理解」」を参照）を作成し、getObject 関数を使用してフローデータ変数値を取得できます（「272 ページの「ECMAScript 変数」」を参照）。</p> <p>注：[ターゲット属性] 列のセルは編集できません。</p>

8.13.3 電子メール通知

このアクティビティではサポートされていません。

統合アクティビティの操作

9

この項では、統合アクティビティの操作について説明します。主なトピックは次のとおりです。

- ◆ 217 ページのセクション 9.1 「統合アクティビティについて」。
- ◆ 217 ページのセクション 9.2 「統合アクティビティの追加」
- ◆ 219 ページのセクション 9.3 「統合アクティビティとの間のデータの移動」
- ◆ 222 ページのセクション 9.4 「統合アクティビティエディタインターフェースの使用」
- ◆ 236 ページのセクション 9.5 「アクション」

9.1 統合アクティビティについて

統合アクティビティは、ワークフローが任意の Web サービスとデータを交換できるようするアクティビティです。Web サービスに送信されたデータは、個別のワークフローを組織内外の他のシステムと統合できます。Web サービスから受信したデータは承認フォームの意志決定サポート情報を提供できます。

ワークフローから Web サービスにデータを移動して処理するには、フローデータ変数を作成します。統合アクティビティは、指定する WSDL ドキュメントに基づいて、Web サービスとの連携のためのアクションモデルを自動的に作成します。

注：アクションモデルは、Novell® Integration Manager（旧称：Novell exteNd Composer）製品で利用可能な機能のサブセットです。

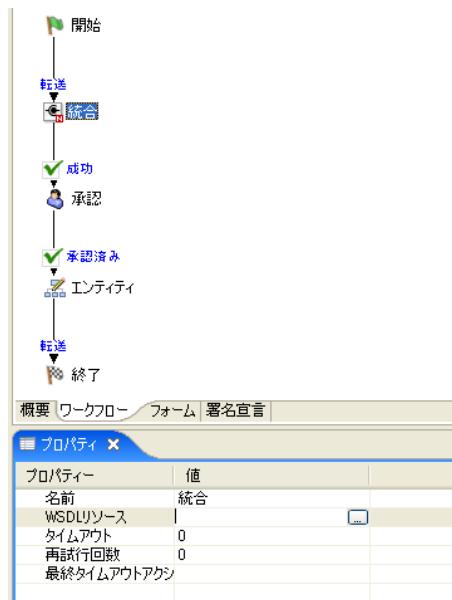
アクションモデルは、XML ドキュメントの処理と XML データソースとの通信に関する一連の指示を視覚的に表したもので、アクションモデルは、統合アクティビティ内のすべてのデータマッピング、データ変換、およびデータ転送を実行します。データを Web サービスに送信する前後にアクションモデルを編集して、データを操作することができます。そして、そのデータを統合アクティビティからフローデータ変数にマップして戻し、ワークフローで使用します。

9.2 統合アクティビティの追加

- 1 プロビジョニング要求定義を作成します（「[97 ページの第 5 章「プロビジョニング要求定義の作成」](#)」を参照）。
- 2 プロビジョニング要求定義のワークフローを作成します（「[163 ページの第 7 章「プロビジョニング要求定義のワークフローの作成」](#)」を参照）。
- 3 [ワークフロー] タブをクリックします。
- 4 統合アクティビティをパレットからドラッグして、ワークフロー内の希望する位置に配置します。



5 [プロパティ] タブをクリックします。

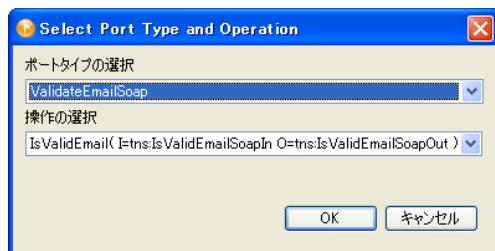


6 [名前] フィールドにアクティビティの名前を入力します。

7 [WSDL リソース] プロパティの [値] フィールドをクリックしてから、参照ボタンをクリックし、参照ボタンをクリックして、統合アクティビティでアクセスする Web サービスの WSDL ファイルを選択するためのダイアログボックスを表示します。

8 このダイアログボックスを使用してファイルシステムを参照し、使用する Web サービスの WSDL ファイルを選択します。WSDL ファイルの名前をクリックしてから、[開く] をクリックして [プロパティ] タブに戻ります。

Web サービスのポートタイプと操作を選択するためのダイアログボックスが表示されます。



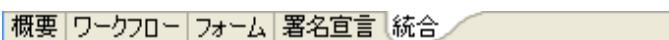
[ポートタイプの選択] リストには、Web サービスでサポートされているポートのタイプが示されます。各ポートタイプでサポートされる操作には、操作の入出力メッセージも含まれます。

- 9 リストからポートタイプを選択します。
- 10 [操作の選択] リストから操作を選択します。
- 11 [OK] をクリックします。

統合アクティビティは WSDL ドキュメントに基づいてアクションモデルを作成します。このアクションモデルを使用して、設計時に必要に応じて、データをワークフローに戻す前に、Web サービスへの入力のテスト、Web サービスからの応答のテスト、データのマップと変換を行います。

多くの Web サービスでは、設計者がアクションモデルについて気にかける必要はありません。統合アクティビティのデータ項目マッピングを作成するだけで十分です。アクションモデルが作成されると、新しいタブ [統合] がプロビジョニング要求定義エディタに追加されます。アクションモデルにアクセスするには、このタブを使用します。

- 12 [タイムアウト間隔]、[再試行]、および [最終タイムアウトアクション] プロパティを指定します（[209 ページのセクション 8.11 「統合アクティビティ」] を参照）。
- 13 アクションモデルを表示または編集するには、[統合] タブをクリックします。



9.3 統合アクティビティとの間のデータの移動

1 フォームフィールドを作成して、統合アクティビティがアクセスする Web サービスに入力情報を提供できるようにします（[107 ページの第 6 章「プロビジョニング要求定義のフォームの作成」] を参照）。たとえば、株価を提供する Web サービスを操作する場合、ユーザが銘柄記号を指定するためのフィールドが必要です。

2 ユーザ入力をフォームからワークフローに移動するには、ワークフローで統合アクティビティより前にあるアクティビティにフローデータ変数を作成します。

フローデータ変数の作成については、「[89 ページのセクション 4.3.3 「ワークフローデータについての理解」] を参照してください。

たとえば、「symbol」という名前のフォームフィールドを作成して Web サービスの入力に使用する銘柄記号を設定する場合は、[symbol] フィールドを含むフォームに関連付けられているアクティビティの後動作データ項目マッピングに移動し、[symbol] フィールドをフローデータ変数（たとえば flowdata.symbol）にマップすることになります。

3 [ワークフロー] タブで、統合アクティビティのアイコンを右クリックしてから、[データ項目マッピングを表示する] を選択します。

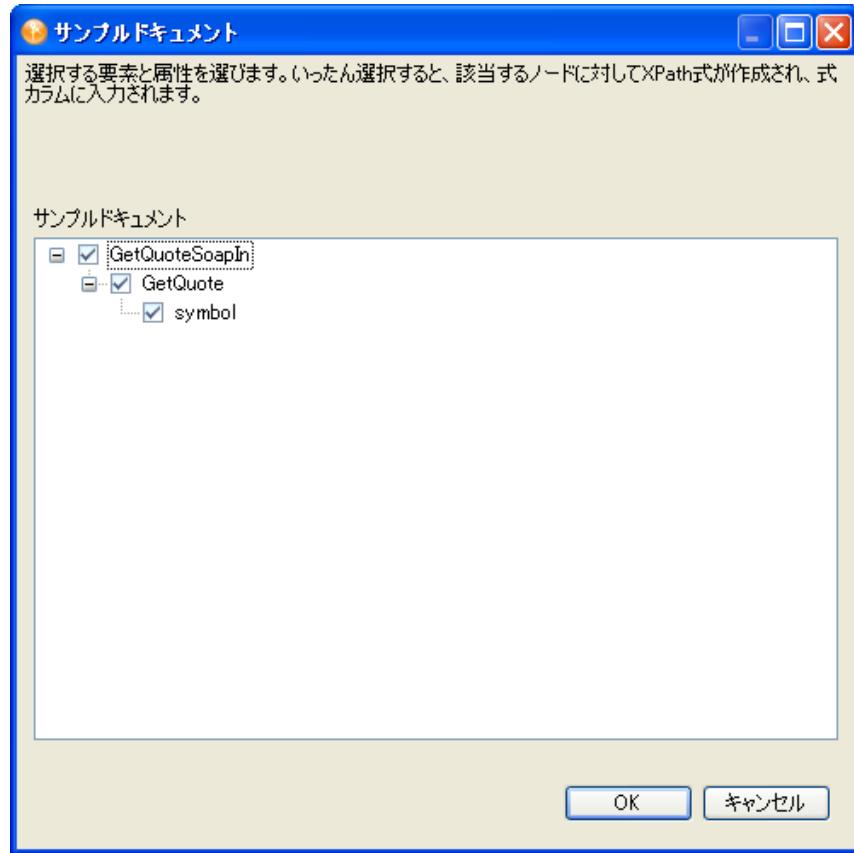
[データ項目マッピング] タブが表示されます。

4 [データ項目マッピング] ビューで、[前動作] をクリックします。

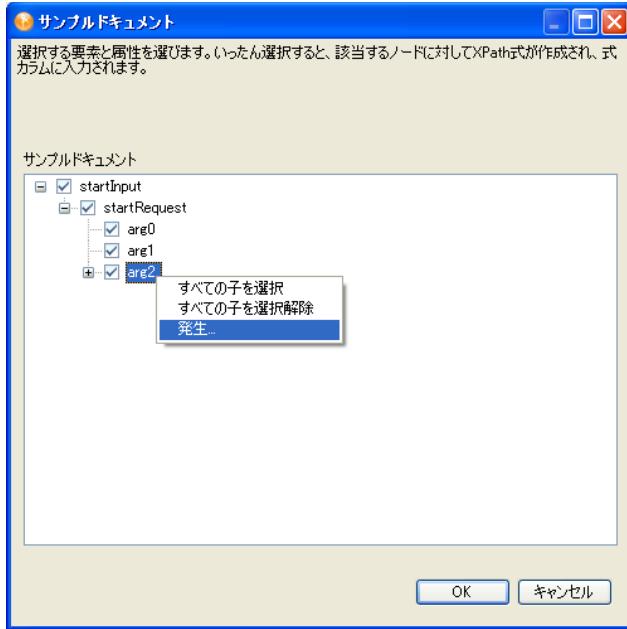
[Web サービス入力フィールド] グリッドに、[ステップ 9] と [219 ページのステップ 10] で指定されたポートタイプと操作に関連付けられているすべての入力フィールドと一致するフィールドが表示されます。

統合アクティビティはこのポートタイプと操作に関連付けられているすべての入力フィールドを自動的に選択します。入力フィールドの削除や入力フィールドのプロパティの変更は、次の手順に従います。

- 4a** [マッピング] をクリックします。
[サンプルドキュメント] ダイアログボックスが表示されます。



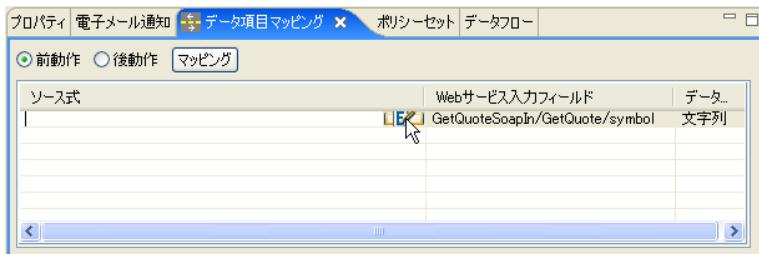
- 4b** サンプルドキュメントのノードを展開して、削除する入力フィールドの選択を解除します。
4c 入力フィールドがバインドされていない要素である場合は、右クリックして [発生] を選択します。



4d [繰り返し回数の入力] ダイアログに最大数を入力して、[OK] をクリックします。

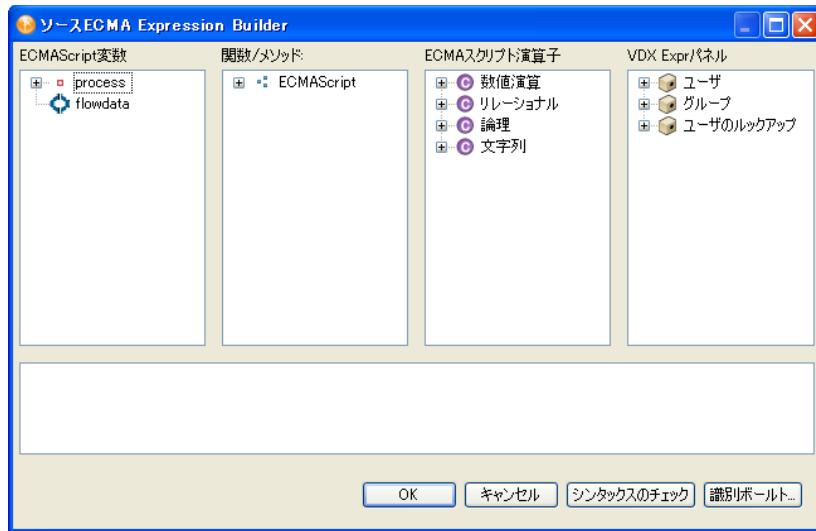
4e [OK] をクリックして、[データ項目マッピング] ビューに戻ります。

- 5** 各 [Web サービス入力フィールド] で、[ソース式] フィールドをクリックしてから、ECMA 式ビルダボタンをクリックします。



ECMA 式ビルダが表示されます。

- 6** ECMA 式ビルダの [ECMAScript 変数] ペインでフローデータノードを展開し、Web サービスへのユーザ入力のフローデータ変数をダブルクリックします。



7 [OK] をクリックして、[データ項目マッピング] ビューに戻ります。

8 [後動作] をクリックします。

[Web サービス出力フィールド] グリッドに、[ステップ 9](#) と [219 ページのステップ 10](#) で指定されたポートタイプと操作に関連付けられているすべての出力フィールドと一致するフィールドが表示されます。

9 統合アクティビティはこのポートタイプと操作に関連付けられているすべての出力フィールド自動的に選択します。一部の出力フィールドを削除する場合は、次の手順に従います。

9a [マッピング] をクリックします。

[サンプルドキュメント] ダイアログボックスが表示されます。

9b サンプルドキュメントのノードを展開して、削除する属性の選択を解除します。

9c [OK] をクリックして、[データ項目マッピング] ビューに戻ります。

10 [すべてマップ] をクリックして、各 [Web サービス出力フィールド] のフローデータ変数を自動作成します。

または、各 [Web サービス出力フィールド] で、[ソース式] フィールドをクリックしてから、ECMA 式ビルダボタンをクリックします。

11 ECMA 式ビルダの [ECMAScript 変数] ペインでフローデータノードを開き、Web サービスからデータを受信するフローデータ変数をダブルクリックします。

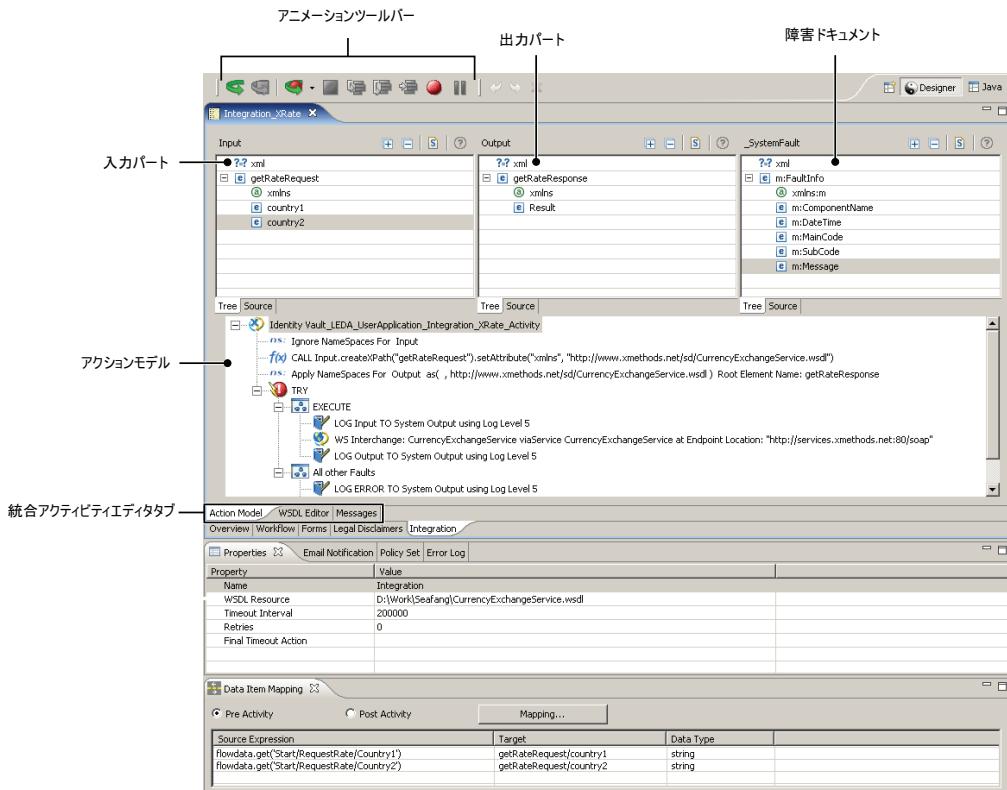
12 [OK] をクリックして、ECMA 式ビルダを閉じます。

これで、統合ビューで Web サービスとのやりとりをテストしたり改良したりできます。

9.4 統合アクティビティエディタインタフェースの使用

統合アクティビティエディタには、統合アクティビティの入力、出力、アクションのための作業環境が備わっています。統合アクティビティエディタは、アクションモデル、WSDL エディタ、メッセージの 3 つのビューで構成されています。

図9-1 統合アクティビティインターフェース



- ◆ 223 ページのセクション 9.4.1 「XML ビュー」
- ◆ 227 ページのセクション 9.4.2 「アクションモデル」
- ◆ 234 ページのセクション 9.4.3 「WSDL エディタ」
- ◆ 235 ページのセクション 9.4.4 「メッセージ」
- ◆ 235 ページのセクション 9.4.5 「アクションモデルのコードの再生成」
- ◆ 235 ページのセクション 9.4.6 「アクションモデルへのアクションの追加」

9.4.1 XML ビュー

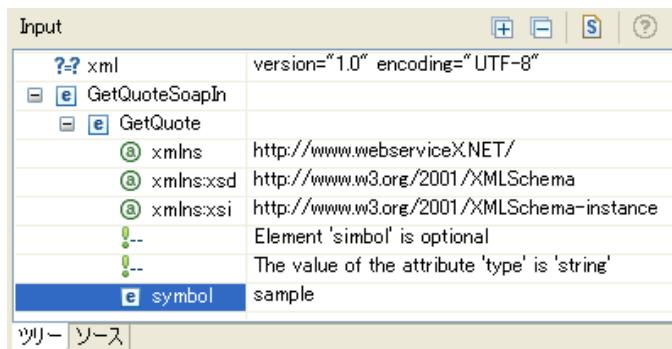
統合アクティビティには、WSDL ドキュメントから派生したいいくつかの XML ビュー（たとえば、入出力メッセージ、WSDL エディタ、メッセージ）が用意されています。これらのビューは共通のインターフェースを使用します。

- ◆ 223 ページの「ツリービュー」
- ◆ 226 ページの「ソースビュー」

ツリービュー

XML ドキュメントを階層的に表示して作業する場合は、ツリービューを使用します。[ツリー] タブをクリックすると、ツリービューを表示できます。

図9-2 ツリービュー



- ◆ 224 ページの「ツリービューの編集機能」
- ◆ 224 ページの「ツリービューのメニュー」
- ◆ 225 ページの「ツリービューのツールバー」
- ◆ 225 ページの「スキーマまたは DTD の添付」

ツリービューの編集機能

ツリービューには次の編集機能が備わっています。

- ◆ 属性値、属性名、ネームスペース名とその値、テキスト、コメントを編集できます。
- ◆ ツリービュー内部で右クリックして表示されるメニューを使って、新しいノードを挿入できます。メニューを使用して、選択したノードの前または後に子供としてノードを挿入できます。ノードが要素である場合は、属性を挿入できます。[子供の追加]、[後に追加]、[前に追加] のサブメニューには、正当に追加できるノードが示されます。文書にスキーマや DTD が何も関連付けられていない場合は、サブメニューに [新規属性] または [新規要素] が示されます。
- ◆ ノードを右クリックして [削除] を選択すると、ノードを削除できます。
- ◆ 複数のツリービューの間で（たとえば、入出力メッセージのビュードラッグアンドドロップして [マップ] アクションを作成できます（[マップ] アクションの詳細については、「[260 ページのセクション 9.5.8 「マップ」](#)」を参照）。
- ◆ 元に戻す、やり直し、切り取り、コピー、貼り付けを行えます。

ツリービューのメニュー

ツリービューで項目を右クリックすると、メニューが表示され、このメニューを使用して XML ドキュメントに対して操作を行えます。メニューはコンテキスト依存で、クリックした項目に適したコマンドのみが表示されます。

表9-1 ツリービューのメニュー

項目	説明
削除	選択した項目を削除します。
DTD 情報の追加	DTD 情報の追加に使用するダイアログボックスを表示します。ルート要素名、パブリック ID、システム ID を編集できます。

項目	説明
ネームスペースの編集	ネームスペース宣言の追加に使用するダイアログボックスを表示します。
属性の追加	新規属性の定義に使用するダイアログボックスを表示します。
子供の追加	サブメニューに次のオプションが表示されます。
前に追加	[コメント] [処理命令の追加] [#PCDATA] [CDATA セクション] [新規要素]
後に追加	
置換文字列	選択した項目でメニューから選択した項目を置き換えます。

ツリービューのツールバー

ツリービューのツールバーには次の機能が備わっています。

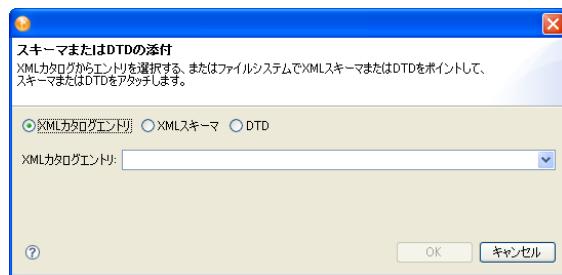
表9-2 ツリービューのツールバー

ボタン	説明
	ドキュメント内のすべてのノードを展開します。
	ドキュメント内のすべてのノードを縮小します。
	スキーマまたは DTD を添付します（「 225 ページの「スキーマまたは DTD の添付」 」を参照）。
	オンラインヘルプを表示します。

スキーマまたは DTD の添付

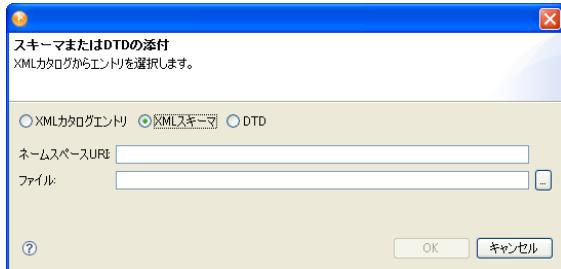
ツリービューの使用時に、現在の XML ドキュメントにスキーマまたは DTD を添付することができます。

- 1 ツリービューツールバーの をクリックします。[スキーマまたは DTD の添付] ダイアログボックスが表示されます。

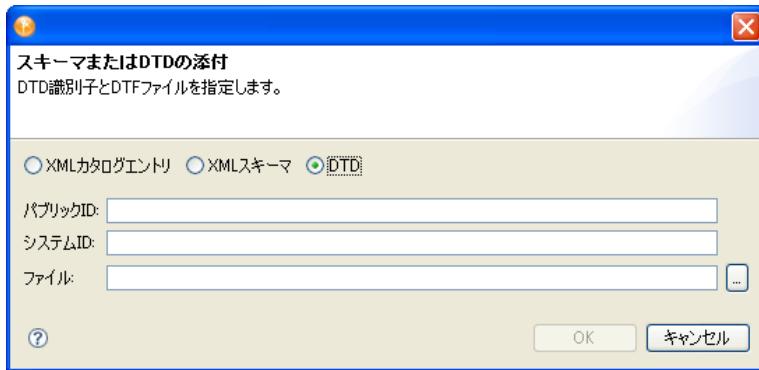


- 2 XML カタログのエントリのリストから選択するには、[XML カタログエントリ] リストからエントリを選択します。

- 3 ディスク上にある XML スキーマを指定するには、[XML スキーマ] をクリックします。



- 4 [ネームスペース URI] に入力し、[ファイル] フィールドの参照ボタンを使用してディスク上にある XML スキーマを選択します。
5 ディスク上にある DTD を指定するには、[DTD] をクリックします。



- 6 [パブリック ID] と [システム ID] に入力してから、[ファイル] フィールドの参照ボタンを使用してディスク上にある DTD ファイルを選択します。

ソースビュー

ドキュメントの XML ソースを表示するには、ソースビューを使用します。ソースビューは [ソース] タブをクリックすると表示されます。

図9-3 ソースビュー

The window has tabs at the bottom labeled 'ソース' (Source) and 'ツリー' (Tree)."/>

- ◆ 227ページの「ソースビューの機能」
- ◆ 227ページの「ソースビューのメニュー」

ソースビューの機能

ソースビューでは次の機能をサポートします。

- ◆ 構文の強調表示。
- ◆ DTD または XML スキーマに基づく、コンテキスト依存のコード補完。
- ◆ 入力時の検証。XML が無効である（たとえば、タグの閉じ角かっこが省略されている）場合、エディタはエラーを示します。
- ◆ 元に戻す、やり直し、切り取り、コピー、貼り付け、すべて選択などの一般的なテキスト編集操作。

ソースビューのメニュー

ソースビューで項目を右クリックすると、メニューが表示され、このメニューを使用して XML ドキュメントに対して操作を行えます。

表 9-3 ソースビューのメニュー

項目	説明
元に戻す	最後のアクションを取り消します。
やり直し	[元に戻す] 操作を取り消します。
切り取り	選択したテキストをクリップボードに切り取ります。
コピー	選択したテキストをクリップボードにコピーします。
貼り付け	クリップボードの内容を挿入位置に貼り付けます。
削除	選択したテキストを削除します。
すべて選択	ドキュメント内のすべてのテキストを選択します。
検索	ドキュメント内のテキストの検索置換に使用するダイアログボックスを表示します。

9.4.2 アクションモデル

アクションモデルには、アクションモデルビューとメッセージパートを表示するビューが組み込まれています。アクションモデルビューには、メッセージパートの内容を操作するアクションが表示されます。メッセージパートには、Web サービス入出力メッセージの XML が表示されます。

- ◆ 228 ページの 「アクションモデルビューについて」
- ◆ 228 ページの 「入力ビューについて」
- ◆ 228 ページの 「出力ビューについて」
- ◆ 229 ページの 「_SystemFault ビューについて」
- ◆ 230 ページの 「アクションモデルペインについて」
- ◆ 231 ページの 「アニメーション」

アクションモデルビューについて

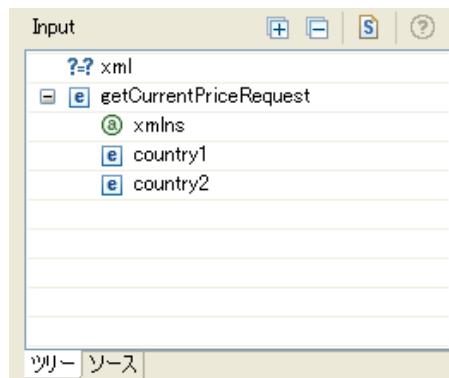
アクションモデルビューは設計時に Web サービスとのやりとりのテストに使用されます。アクションモデルビューでアクションを編集します。Web サービスに入力するテストデータを入力ビューに入力し、Web サービスからの応答を出力ビューで調べ、Web サービスから戻されるエラーメッセージがあれば _SystemFault ビューで確認できます。統合アクティビティには次のメッセージペインがあります。

- ◆ 入力ビュー
- ◆ 出力ビュー
- ◆ _SystemFault ビュー
- ◆ アクションモデルビュー

入力ビューについて

入力ビューには、Web サービスの WSDL ドキュメントから派生した入力メッセージが表示されます。右側の境界線をドラッグすると、ビューのサイズを変更できます。列の境界線をドラッグすると、ビュー内の列のサイズを変更できます。アクションモデルのテストで使用する値を入力パートに直接指定できます。この場合、指定した値はアクションモデルの実行後に破棄されます。[メッセージ] タブを使用して値を指定することもできます（「[235 ページのセクション 9.4.4 「メッセージ」](#)」を参照）。この場合、指定した値は値を削除するか、アクションモデルを再生成するまで保持されます（「[235 ページのセクション 9.4.5 「アクションモデルのコードの再生成」](#)」を参照）。

図 9-4 入力ビュー

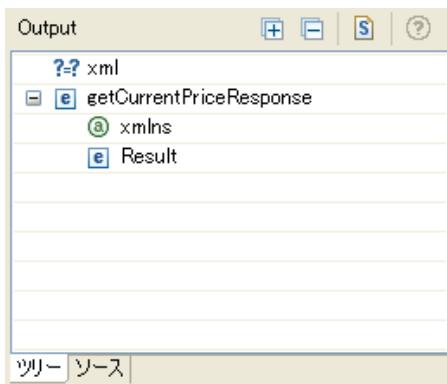


出力ビューについて

出力ビューには、Web サービスの WSDL ドキュメントから派生した出力メッセージが表示されます。アクションモデルを実行する際には、出力ビューを使用して Web サービスから戻される値を表示します。

左側の境界線をドラッグすると、ビューのサイズを変更できます。列の境界線をドラッグすると、ビュー内の列のサイズを変更できます。モデリングに使用する値を出力パートに直接指定できます。この場合、指定した値はアクションモデルの実行後に破棄されます。[メッセージ] タブを使用して値を指定することもできます（「[235 ページのセクション 9.4.4 「メッセージ」](#)」を参照）。この場合、指定した値は値を削除するまで保持されます。

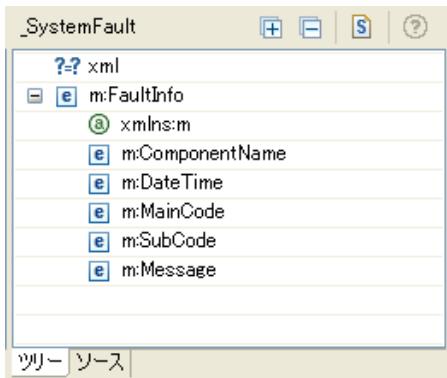
図9-5 出力ビュー



_SystemFault ビューについて

_SystemFault ビューには、アクションモデルの実行時に生成されたエラーメッセージがすべて表示されます。_SystemFault に含まれる XML 情報は、ERROR というグローバルオブジェクトにも書き込まれます。

図9-6 _SystemFault ビュー



FaultInfo ルートの下に、次の要素があります。

- ◆ *DateTime* にはフォルトが発生した日付と時刻が入ります。
- ◆ *ComponentName* にはフォルトをスローしたコンポーネントの名前が入ります。
- ◆ *MainCode* にはエラーの主なコード番号が入ります。
- ◆ *SubCode* にはエラーのサブコード番号が入ります。
- ◆ メッセージには「[フォルトのスロー] アクション設定時に定義したエラーメッセージが入ります（「[238 ページの「フォルトのスロー」](#)」を参照）。[フォルトのスロー] アクションにエラーメッセージを指定していない場合は、「ユーザ定義のフォルトが発生しました！」というメッセージが表示されます。」「試行 / フォルト時」アクション内でエラーが発生したが、フォルトを指定していなかった場合、この要素には例外メッセージが入ります。

アクションモデルペインについて

統合アクティビティには1つのアクションモデルがあります。アクションモデルは、Webサービス入出力メッセージに対して実行されるマッピング、変換、その他のアクションを表します。アクションモデルビューはサイズを変更できます。アクションモデルビューで行われるほとんどのアクティビティには、アクションの追加と編集が関係します。

- ◆ [230ページの「アクションモデルのコンテキストメニュー」](#)
- ◆ [230ページの「アクションモデル内のテキストの検索置換」](#)

アクションモデルのコンテキストメニュー

アクションモデルを右クリックすると、メニューが表示されます。

図9-7 アクションモデルメニュー

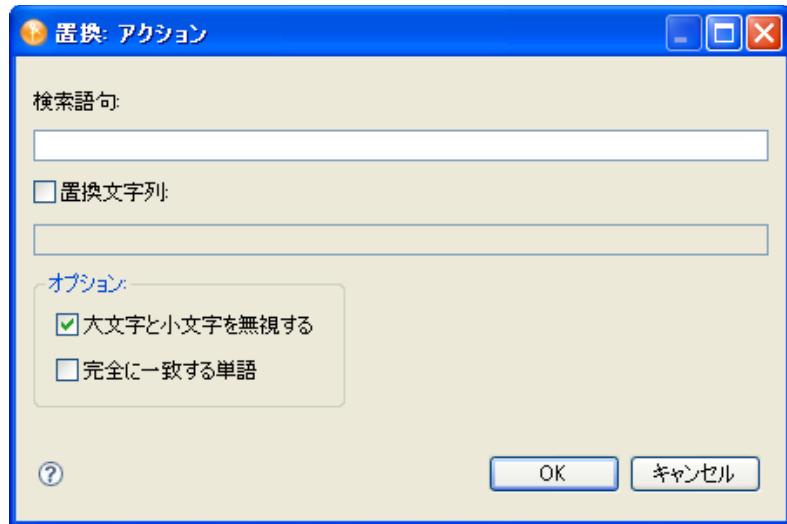


このメニューから、アクションの追加や編集（「[236ページのセクション9.5「アクション」](#)」を参照）、アクションモデル内のブレークポイントの切り替え（「[231ページの「アニメーション」](#)」を参照）、その他のタスクを実行できます。

アクションモデル内のテキストの検索置換

アクションモデルメニューの [置換] コマンドを使用して、単語または文字列を置換できます。

- 1 アクションモデルを右クリックしてから、[置換] を選択します。



- 2 検索テキストを指定します。
- 3 検索テキストを置換する場合は、[置換文字列] をクリックして、検索文字列を置換する文字列を入力します。
- 4 テキストの大文字小文字に関係なく検索テキストを検索する場合は、[大文字と小文字を無視する] をクリックします。
- 5 検索テキストに完全に一致する単語のみを検索する場合は、[完全に一致する単語] をクリックします。
- 6 [OK] をクリックします。

統合アクティビティは検索テキストの最初の出現箇所を検索します。操作が検索と置換の操作である場合、統合アクティビティは置換するか確認します。そのようにして、検索テキストの次の出現箇所またはすべての出現箇所を置換できます。

アニメーション

アクションモデルには、統合アクティビティ内でアクションを対話式にテストおよびトラブルシュートするために使用できるアニメーションツールが備わっています。アクションモデルを手順に従って実行し、各アクションの結果を確認できます。何らかのエラーが起きたときに確認できるだけでなく、接続とデータが計画どおりに動作したかをリアルタイムで検証できます。

アニメーションツールを使用すると、1つ以上のブレークポイントを切り替えることができます。アクションモデルの特定のセクションに注目するために、この機能を使用できます。[ブレークポイントまで実行] ツールと連動させて使用する場合、ブレークポイントを設定することで、正しく動作するアクションモデルセクションを速やかに実行し、特定のアクションで停止させます。そこから、各アクションを順番にステップ実行します。1ステップずつ実行すると時間がかかるループやその他のコードブロックをステップオーバーすることもできます。

基本アニメーションツール

アニメーションツールは Designer ツールバーから使用できます。

図9-8 アニメーションツールバー



表9-4 アニメーションツール

アニメーションツールバーボタン	名前	説明
	実行	アクションモデルを実行します。
	現在のアクションを実行	現在選択しているアクションを実行します。
	アニメーションの開始	アニメーション処理を開始します。[ステップイン]、[ステップオーバー]、[ブレークポイント/終了まで実行] を有効にします。
	アニメーションの終了	アニメーション処理を停止します。
	ステップイン	現在選択されているアクションを実行し、次の順次アクションを強調表示します。 [ループの繰り返し] アクションの場合、[ステップイン] をクリックすると、ループ内の各アクションが実行され、各ループが繰り返されます。 [決定] アクションの場合、[ステップイン] は [True] または [False] ブランチで次のアクションを処理します。 [試行 / フォルト時] アクションの場合、[ステップイン] は実行ブランチ内、および場合によっては [フォルト時] ブランチ内の次のアクションを処理します。
	ステップオーバー	現在選択されているアクションを実行し、次の順次アクションを強調表示します。[ステップイン] ボタンとは異なり、このボタンをクリックしても、[繰り返し]、[決定]、[試行 / フォルト時] アクションの詳細が強調表示および実行されることはありません。
	ブレークポイント/終了まで実行	次のブレークポイントまで、またはブレークポイントがない場合はアクションモデルの最後まで、アニメーションを実行します。
	ブレークポイントの切り替え	アクションモデル内の選択されたアクションをブレークポイントとして設定します。2つ以上のブレークポイントを設定することもできます。希望するアクションを右クリックし、メニューから [ブレークポイントの切り替え] を選択してブレークポイントを切り替えることもできます。
	アニメーションの一時停止	アニメーションを一時停止します。

アニメーションの開始

統合アクティビティをはじめて開いたときに有効なボタンは、[アニメーションの開始] と [ブレークポイントの切り替え] だけです。[アニメーションの開始] をクリックすると、残りのアニメーションボタンが有効になります。一時的にアニメーションを休止する

場合は、[アニメーションの一時停止] ボタンを使用できます。アニメーションを中止する場合は、いつでも [アニメーションの終了] をクリックして中止できます。

アニメーション時にはコピー、貼り付け、アクション編集操作（新規アクションの追加を含む）がすべて利用できますが、アニメーション中はアクションモデルを編集しないことをお勧めします。編集を行うと、例外や予期しない動作が起きる可能性があります。アクションモデルを編集する必要がある場合は、まず [アニメーションの終了] を使用してアニメーションを停止します。次いで、編集内容を適用して、再びアニメーションを開始します。

- 1 統合アクティビティを開きます。
- 2 Designer ツールバーの [アニメーションの開始] ボタンをクリックします。[アニメーションの開始] ボタンは灰色で表示され、それ以外のすべてのアニメーションボタンがアクティブになります。
- 3 希望するアニメーションアクティビティを実行するには、次の項の指示に従ってください。

ブレークポイントの切り替え

アクションモデル内のアニメーション処理を停止したい場所にブレークポイントを設定するには、[ブレークポイントの切り替え] ツールを使用します。正しく動作する複数の長いセクションで構成される非常に長いアクションモデルがある場合には、このツールが役立ちます。問題の原因になっているアクションごとにブレークポイントを設定し、アクションをステップ実行して問題をトラブルシュートできます。

- 1 [アクション] ペインで、ブレークポイントを設定するアクションを選択します。これがアニメーションの停止する場所です。
- 2 Designer ツールバーの [ブレークポイントの切り替え] をクリックするか、またはアクションを右クリックして [ブレークポイントの切り替え] を選択します。アクションモデルの左の境界線に、ブレークポイントを示すドットが表示されます。
- 3 必要に応じて、前のステップを繰り返して、追加のブレークポイントを選択します。

アクションのステップイン

[ステップイン] を選択すると、アクションモデル内の選択したアクションが実行され、次の順番のアクションに移動します。[ステップイン] ツールは、アクションモデル全体の中で各アクションをステップ実行するために使用したり、[ブレークポイントまで実行] ツールと組み合わせて使用したりできます。実行は次のブレークポイントに達したとき、またはアクションモデルが終了したとき（いずれか早い方）に停止します。

ブレークポイントを使用する可能性のあるシナリオは、10 個のアクションは正しく動作することがわかっているが、11 個目のアクションが疑わしい場合です。11 個目のアクションをブレークポイントとして設定し、[ブレークポイントまで実行] ツールを実行してから、[ステップイン] ツールを実行して 11 番目（およびそれ以降）のアクションをステップ実行します。

- 1 アニメーションを開始します（「[232 ページの 「アニメーションの開始」](#)」を参照）。
- 2 [ステップイン] をクリックします。アクションモデルの最初のアクションが選択されます。
- 3 もう一度 [ステップイン] をクリックします。選択されたアクションが実行され、次のアクションが選択されます。

- 4** アクションが実行されて次のアクションが選択されるたびに [ステップイン] をクリックして、アクションモデルの処理を続行します。

アクションのステップオーバー

[繰り返し]、[決定]、[試行 / フォルト時] アクションの詳細にステップインしない場合には、[ステップオーバー] を使用します。各アクションを個別にステップ実行せずに、コードのブロック全体を実行することができます。

[ブレークポイントまで実行] と組み合わせて [ステップオーバー] 使用できます。たとえば、ブレークポイントを切り替えて、[ブレークポイントまで実行] を実行してから、[ステップオーバー] を使用してブレークポイントに設定されているアクションを実行できます。[ステップオーバー] を使用することで、アクションをひとつひとつステップ実行するという単調で時間のかかる作業を行わずに済むので、非常に長いアクションモデルをテストする場合に時間をかなり節約できます。

- 1** アニメーションを開始します（[「232 ページの「アニメーションの開始」」](#) を参照）。
- 2** ループに達するか、インデントされたコードブロックに先行する他のコード行に達するまで、[ステップイン] でアクションモデルをステップ実行します。
- 3** [ステップオーバー] をクリックします。インデントされたコードブロックの後の最初のアクションが選択されます。インデントされたアクションを 1 つずつステップ実行しなくとも、インデントされたコード全体が正しく実行され、そのまま次のアクションのブロックに進みます。
- 4** 必要に応じて [ステップオーバー] をクリックして、アクションモデルの処理を続行します。
- 5** 引き続き [ステップイン] と [ステップオーバー] をクリックして、アクションモデル内のすべてのアクションを実行します。

アニメーションの一時停止

アクションモデル内のアクションの実行を一時停止するには、[アニメーションの一時停止] を使用します。アクションモデルに非常に長いループが含まれている場合に、特に役立ちます。

- 1** アクションの実行中に、[アニメーションの一時停止] をクリックします。
- 2** アニメーションの処理を再開するには、[ステップイン]、[ステップオーバー]、[ブレークポイントまで実行]（ブレークポイントが設定されている場合）をクリックします。

アニメーションの中止

アニメーションの処理を停止するには、[アニメーションの停止] を使用します。アニメーションを停止した場合、停止した場所から再開することはできません。アクションモデルの最初から再開する必要があります。

9.4.3 WSDL エディタ

WSDL エディタには、Web サービスの WSDL ドキュメントが表示されます。WSDL の編集は、ツリービューとソースビューの編集機能を使用して行えます（[「223 ページの「ツリービュー」」](#) および [「226 ページの「ソースビュー」」](#) を参照）。

9.4.4 メッセージ

メッセージビューには、Web サービスの WSDL ドキュメントから派生したメッセージが表示されます。メッセージの編集は、ツリービューとソースビューの編集機能を使用して行えます（「[223 ページの「ツリービュー」](#)」および「[226 ページの「ソースビュー」](#)」を参照）。設計時にアクションモデルを実行するときに使用するテストデータを入力するために、この機能を使用できます。メッセージビューで入力したデータは、アクションモデルの実行も保持されます。

9.4.5 アクションモデルのコードの再生成

WSDL エディタビューで作業している場合、[再生成] ボタンをクリックすることにより、アクションモデルのすべてのコードを再生成し、メッセージを再生成できます。メッセージビューで作業している場合は、アクションモデル内のすべてのアクションを再生成できます。[再生成] ボタンは Designer ツールバーにあります。

図 9-9 [再生成] ボタン



9.4.6 アクションモデルへのアクションの追加

アクションとは、統合アクティビティ内で行われる処理ステップのことです。アクションの集まりをアクションモデルと呼びます。アクションモデルの各アクションがそれぞれ1行に表示され、各行にはアクションの定義の要約とアクションタイプのアイコンが表示されます。一部のアクションは他のアクションに従属します。たとえば、ループ処理を制御する [繰り返し] アクションを作成してから、ループ内部にアクションを追加できます。ループ内部のアクションは [繰り返し] アクションに従属し、[繰り返し] アクションの下にインデントされて表示されます。従属アクションは、[繰り返し] アクションが True である限り処理されます。

アクションモデルにアクションを追加するには、アクションモデル内でアクションを挿入する位置の1つ上の行をクリックします。次のいずれかの方法を使用して、アクションを追加します。新規アクションは選択した行の下に挿入されます。

- ◆ ドラッグアンドドロップ。入力ビューから出力ビューに要素をドラッグアンドドロップして、[マップ] アクションを追加できます。
- ◆ コピーと貼り付け。アクションモデルビューのアクションをコピーして、ビュー内のどこか別の場所に貼り付けることができます。
- ◆ アクションモデル内でアクションを追加する位置の1つ上の行を右クリックしてから、希望するアクションをメニューから選択します。

注：アクションモデル内でアクションをドラッグして新しい位置に配置することで、アクションモデルのアクションを並び替えることができます。

アクションモデルを作成した後、展開する前に、作成したアクションモデルをテストします。テストはアニメーションツールを使用して行います。アニメーションツールを使って、ブレークポイントの設定、アニメーションの開始、アクションのステップインとス

テップオーバー、アニメーションの一時停止を行うことができます。詳細については、「[231 ページの「アニメーション」](#)」を参照してください。

9.5 アクション

この項では、アクションモデル内で利用可能なアクションについて説明します。アクションは、入力をパラメータの形で特定のタスクを実行する点で、プログラミングステートメントに似ています。アクションモデルは、XML ドキュメントの処理と XML データソースとの通信を行うための命令セットです。アクションモデルは、統合アクティビティ内のすべてのデータマッピング、データ変換、およびデータ転送を実行します。アクションモデル内のすべてのアクションは連動します。

ランタイム時に、すべてのアクションは ECMAScript の実行可能フォームに変換されて処理されます。設計時に、多くのアクションは ECMAScript 式をパラメータとして受け入れるので、アクションモデルの柔軟性と制御が大いに増します。[関数] アクションを使用すると、ECMAScript 言語の機能をすべて利用できるようになるため、優れた柔軟性と制御が得られます。

この項では、次のトピックについて説明します。

- ◆ [236 ページのセクション 9.5.1 「詳細」](#)
- ◆ [243 ページのセクション 9.5.2 「データ交換」](#)
- ◆ [248 ページのセクション 9.5.3 「繰り返し」](#)
- ◆ [255 ページのセクション 9.5.4 「コメント」](#)
- ◆ [256 ページのセクション 9.5.5 「決定」](#)
- ◆ [257 ページのセクション 9.5.6 「関数」](#)
- ◆ [258 ページのセクション 9.5.7 「ログ」](#)
- ◆ [260 ページのセクション 9.5.8 「マップ」](#)

9.5.1 詳細

この項では、次のアクションについて説明します。

- ◆ [236 ページの「ネームスペースの適用」](#)
- ◆ [238 ページの「フォルトのスロー」](#)
- ◆ [240 ページの「試行 / フォルト時」](#)

ネームスペースの適用

この項では、次のトピックについて説明します。

- ◆ [236 ページの「\[ネームスペースの適用\] アクションについて」](#)
- ◆ [237 ページの「\[ネームスペースの適用\] アクションの作成」](#)

[ネームスペースの適用] アクションについて

統合アクティビティが常に有効な XML ドキュメントを受信するのが理想的です（つまり、ドキュメントはデータをスキーマに照らして検証し、マップし、適切に変換して、有効な XML ドキュメントを送信します）。しかし、必ずしもそうなるとは限りません。

ネームスペースをすべて無視したいという場合もあります。何らかの方法で XML ドキュメントを検証できることが重要です。このようなさまざまな XML 处理の事例では、統合アクティビティに備わっているプレフィックスとネームスペースの処理を変更または上書きする方法が必要です。

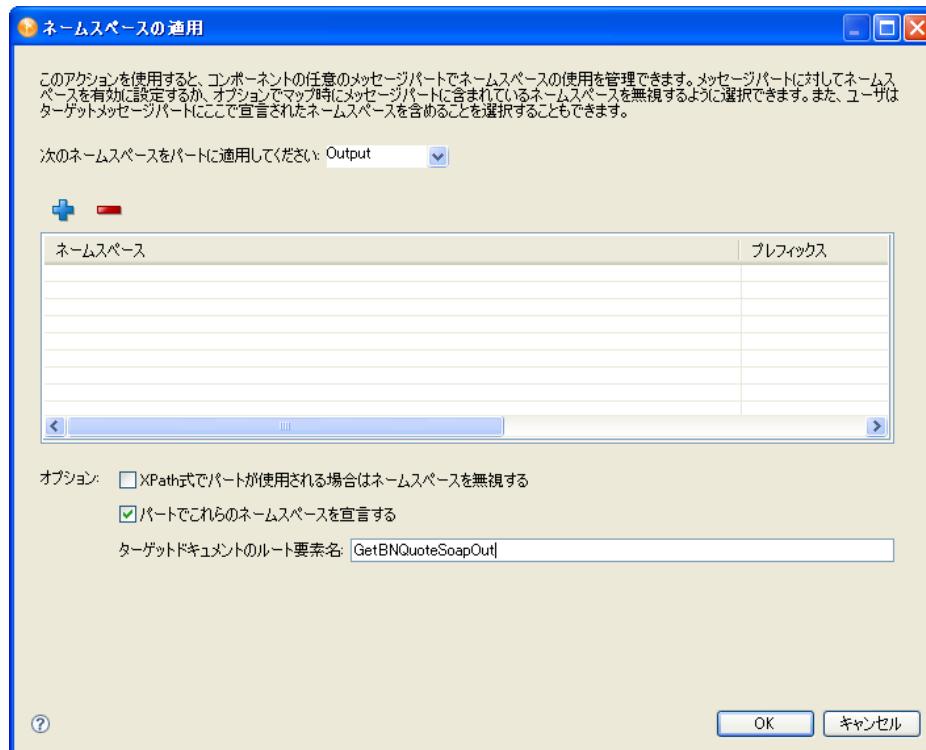
[ネームスペースの適用] アクションには、アクションモデル内で入出力メッセージのためのネームスペースとネームスペースプレフィックスを有効に管理するメカニズムが用意されています。このアクションを使用することにより、Web サービスのネームスペース宣言とプレフィックス宣言を 1ヶ所に集約したり、入出力メッセージに宣言されたものを上書きしたり、ネームスペースをすべて無視したりできます。

[ネームスペースの適用] アクションは入出力メッセージに適用できます。1つのメッセージパートに複数の [ネームスペースの適用] アクションを用意して、アクションモデルに指定されている条件に基づいてネームスペースを効果的に変更することができます。宣言されたネームスペースは、アクションモデルの終わりに達するか、または別の [ネームスペースの適用] アクションが実行されるまで有効です。つまり、最新の [ネームスペースの適用] アクションだけが有効です。

新しい統合アクティビティを作成すると、WSDL がネームスペースを宣言している場合には、[ネームスペースの適用] アクションが出力メッセージ用に自動的に作成されます。コンポーネントを作成した後は、追加の [ネームスペースの適用] アクションを手動で作成できます。

[ネームスペースの適用] アクションの作成

- 1 アクションモデル内で [ネームスペースの適用] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [詳細] > [ネームスペースの適用] の順に選択します。



3 ネームスペースを適用するメッセージ（入力、出力、_SystemFault、プロジェクト）を [次のネームスペースを適用するパート] リストから選択します。

4 プラス (+) アイコンをクリックして新しい行を追加してから、[ネームスペース] 列をクリックしてネームスペース URI を入力します。

表には選択したメッセージパートのすべてのネームスペース宣言が表示されます。新しい [ネームスペースの適用] アクションを作成すると、表に選択したパートの宣言のリストが含まれている場合とそうでない場合があります。宣言のリストは、最初は WSDL に定義されている宣言から作成されます。

メッセージパートの宣言リスト内で、プレフィックスは固有でなければなりません。しかし、URI のプレフィックスが固有であれば、重複するネームスペース URI があってもかまいません。それで、同一のネームスペース URI に関連付けられた複数のプレフィックスを宣言できます。

5 必要に応じて、[ドキュメントを XPath 式で使用する場合は、ネームスペースを無視する] をクリックします。

[マップ] アクションのソース XPath が要素を XML ローカル名のみで検索するようになるには、このオプションを使用します。

これは [マップ] アクションを指定するより制限の少ない方法（「[260 ページのセクション 9.5.8 「マップ」](#)」を参照）を提供し、[マップ] アクションのソース指定に誤ったプレフィックスが含まれているか、またはプレフィックスが含まれていない場合に役立ちます。これにより、入力メッセージにプレフィックスが含まれているかどうかをテストする [決定] アクション（「[256 ページのセクション 9.5.5 「決定」](#)」を参照）の内部に [ネームスペースの適用] アクションを配置することができますが、さらに [マップ] アクションを 1 セット配置して入力を別のパートにマップすることができます。言い換えると、統合アクティビティは通常プレフィックスを含む入力を予期するので、すべての [マップ] アクションをプレフィックス名で設計します。プレフィックスを含まない入力があった場合に、[決定] アクションは定義された [ネームスペースの適用] アクションをアクティブにして入力のネームスペースを無視し、どちらの場合でも [マップ] アクションが動作するようにします。

6 自分のアクションモデルによってビルトされた出力メッセージのルート要素にネームスペースのセットを宣言する場合は、[これらのネームスペースをパートに宣言する] をクリックします。

出力の場合は、マップアクションの結果として出力内に作成されるプレフィックス付きの要素が確実に適切なネームスペースに解決されるように、通常はこのオプションを選択します。

これにより、出力の受信者はドキュメントを適切に検証することができます。このオプションを選択した [ネームスペースの適用] アクションを使用して、すでに宣言を含む既存のドキュメントに新しい宣言を追加することもできます。

[ターゲットドキュメントルート要素名] は、ネームスペース宣言属性を含むルート要素名を指定するために使用します。統合アクティビティは、WSDL ドキュメントの情報と [次のネームスペースを適用するパート] で指定したメッセージパートに基づいて、自動的にこの値を入力します。

7 [OK] をクリックします。新しいアクションがアクションモデルに追加されます。

フォルトのスロー

この項では、次のトピックについて説明します。

- ◆ [239 ページの 「\[フォルトのスロー\] アクションについて」](#)

- ◆ 240 ページの 「[フォルトのスロー] アクションの追加」

[フォルトのスロー] アクションについて

[フォルトのスロー] アクションを使用して、次のことを行います。

- ◆ アクション失敗時に XML メッセージに情報を書き込む
- ◆ フォルトをスローする前に複数のアクションを実行する
- ◆ コンポーネントの実行を休止する

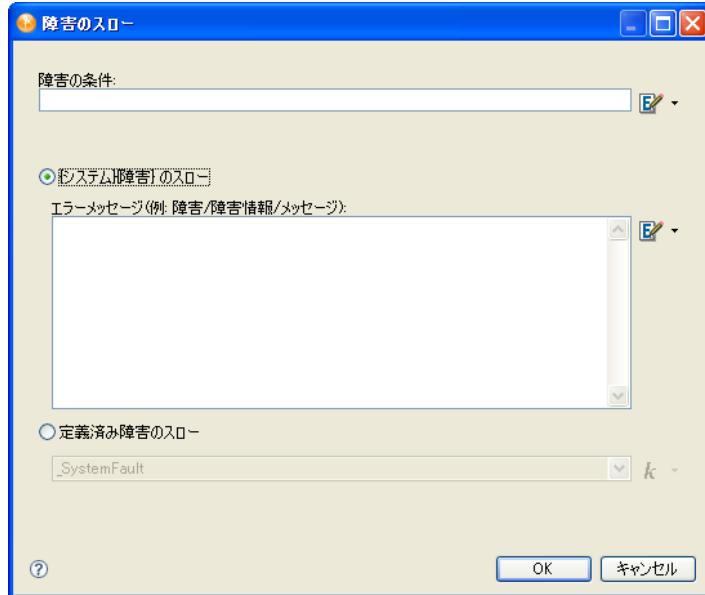
[フォルトのスロー] は、指定した条件が真である場合にのみ実行されます。[フォルトのスロー] アクションが実行されたときに書き込まれるメッセージパートはフォルトドキュメントと呼ばれ、このメッセージ内の XML は ERROR というグローバルオブジェクトに含まれます。

[フォルトのスロー] アクションは、次のようにいくつかの方法で使用できます。

- ◆ [フォルトのスロー] アクションをそれ自体で使用する。[フォルトのスロー] アクションダイアログ内にフォルト条件とエラーメッセージを指定できます。アクションを実行すると、フォルト条件が評価されます。条件が True として評価されると、次のようにになります。
 - ◆ 指定した [スロー前] アクションがあれば実行されます。実行を休止する前にアプリケーションを特定の状態のままにしておくのにとても役立ちます。たとえば、実行が完了しなかったことを伝えるメールメッセージを送信することができます。
 - ◆ エラーメッセージの内容は、指定したノードのフォルトドキュメントとグローバルオブジェクト ERROR に書き込まれます。
 - ◆ アクションモデルの実行が休止します。
- ◆ [決定] アクションの決定式内に [フォルトのスロー] アクションを使用する。[決定] アクションの決定式に、フォルト条件を入力して指定できます。次に、[決定] アクションの [True] ブランチに [フォルトのスロー] ステートメントを配置します。ここで、[フォルトのスロー] フォルト条件に追加条件を指定するか、またはフォルト条件を空白のままにしてフォルト情報を書き込むフォルトドキュメントを指定します。アクションが実行されて、すべての条件が True であると、[フォルトのスロー] アクションが実行されます。[決定] アクションまたは [フォルトのスロー] アクションのフォルト条件が False である場合は、アクションモデル内の次のアクションが実行されます。
- ◆ [試行 / フォルト時] アクション内部で [フォルトのスロー] を使用する（「240 ページの 「試行 / フォルト時」」を参照）上記の方法のいずれかを [試行 / フォルト時] アクションの実行ブランチ内に配置することにより、統合アクティビティの実行が休止するのを防ぎ、フォルトに対処したりフォルトから回復したりできるようになります。出力をテストしたい他のアクションが正しく動作した後に、上の 2 つの方法のいずれかを使用して、[試行 / フォルト時] アクションの実行ブランチ内部でフォルト条件を作成します。固有のフォルトをいくつでも指定でき、それにより発生したフォルトに応じて統合アクティビティをいくつかの異なる方向に分岐させることができます。特定のフォルトに対して [フォルトのスロー] アクションがトリガされると、コンポーネントの実行を休止する代わりに、制御が [試行 / フォルト時] アクションの適切なブランチに渡されます。ここで、エラーを修復するまたはエラーに応答する他のアクションを指定できます。

[フォルトのスロー] アクションの追加

- 1 アクションモデル内で [フォルトのスロー] アクションを追加する位置の 1 つ上の行を右クリックします (選択した行の下に新しいアクションが挿入されます)。
- 2 [新規アクション] > [詳細] > [フォルトのスロー] の順に選択します。



- 3 [フォルト条件] フィールドに、True であればアクションがフォルトをスローするような有効な ECMA Script 式を入力します。

ECMA 式ビルダボタンをクリックして、式を作成することもできます (「[269 ページの第 10 章「ECMA 式の使用」](#)」を参照)。

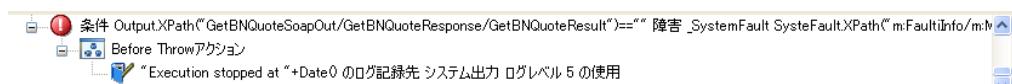
- 4 [{ システム}{ フォルト} のスロー] を選択して、エラーメッセージを _SystemFault ドキュメントに書き込みます。

デフォルトでは、[エラーメッセージ] フィールドに入力するメッセージがドキュメントの Fault/FaultInfo/Message ノードに配置されます。必要に応じて異なるノードを指定します。ECMA 式ビルダボタンをクリックして、式を作成することもできます。

- 5 [定義済みフォルトのスロー] を選択して、統合アクティビティに関連付けられているメッセージパートの 1 つであるフォルトドキュメントを選択します。

- 6 [OK] をクリックします。

新しいアクションがアクションモデルに追加されます。アプリケーションが停止する前に実行させるすべてのアクションを [スローアクションの前] ブランチに配置します。



試行 / フォルト時

この項では、次のトピックについて説明します。

- ◆ [241 ページの「\[試行 / フォルト時\] アクションについて」](#)

- ◆ 241 ページの 「[試行 / フォルト時] アクションの追加」

[試行 / フォルト時] アクションについて

[試行 / フォルト時] アクションは、[試行 / フォルト時] アクションの [実行] ブランチ内でフォルトが発生したときに一連のアクションを実行します。定義済みのフォルトを [実行] ブランチ内にいくつでも指定できます。[試行 / フォルト時] アクションを使用して、予期されるエラーを捕捉し、修復やフォルトの報告を実行する他のアクションを実行します。たとえば、[試行 / フォルト時] を使用してファイルの検出に失敗する [XML 交換] アクションに応答することができます。

[試行 / フォルト時] アクションを追加すると、いくつかの行がアクションモデルに追加されます。

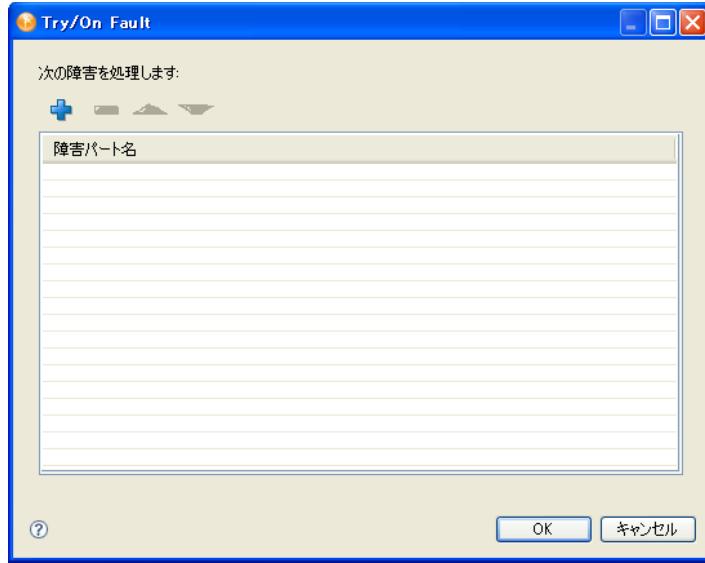
- ◆ [試行] アクションの始まり
- ◆ [実行] ブランチ
- ◆ 指定したフォルトごとにブランチ
- ◆ [その他すべてのフォルト] ブランチ

アクションがフォルトを生成する可能性があることに気付いたら、それらのアクションを [実行] ブランチに配置します。次いで、固有の状況を処理する各 [フォルト時] ブランチの下にエラー処理アクションを配置します。フォルトが発生すると、[フォルト時] ブランチのアクションが実行されます。

前述の例で、[XML 交換] アクションでフォルトが予期される場合には、[実行] ブランチの下にアクションを配置します。ある [フォルト時] ブランチには、別の場所からのファイル読み取りを試行する別の [XML 交換] アクションを追加し、別の [フォルト時] ブランチには、拡張子の異なるファイルを検索する別の [XML 交換] アクションを追加することができます。

[試行 / フォルト時] アクションの追加

- 1 アクションモデル内で [試行 / フォルト時] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [詳細] > [試行 / フォルト時] の順に選択します。



プラス (+) アイコンを使用して、[フォルトパート名] リストに新しいフォルトパートを追加します。リストからフォルトパートを削除するには、赤い - アイコンを使用します。フォルトの順番を変更するには、上矢印と下矢印のアイコンを使用します。

フォルトパートを指定しない場合は、[試行 / フォルト時] アクションのデフォルトの [その他すべてのフォルト] ブランチに訂正のアクションを配置できます。

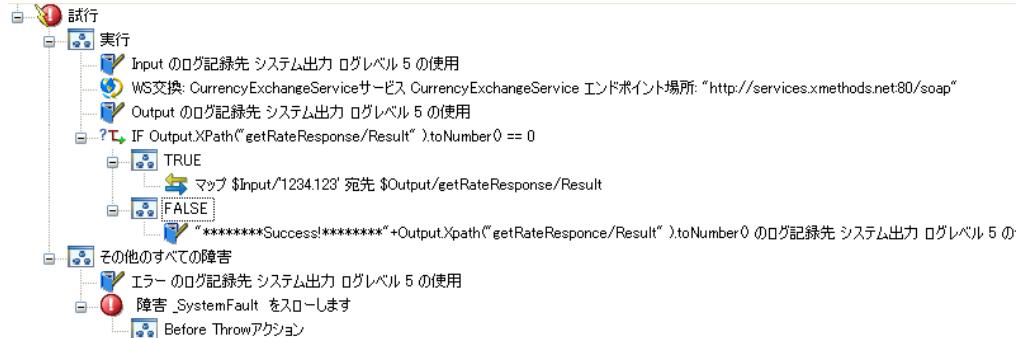
3 [OK] をクリックします。

アクションモデルビューアには、[試行 / フォルト時] アクションアイコンと、1つの [実行] ブランチ、1つ以上の [フォルト時] ブランチ、および1つの [その他すべてのフォルト] ブランチが表示されます。

4 エラーになりそうなすべてのアクションを [実行] ブランチに追加します。

5 [フォルト時] ブランチに、[実行] ブランチで指定したエラーを解決するアクションを追加します。

次の図は、アクションモデル内の完成した [試行 / フォルト時] アクションを示しています。



9.5.2 データ交換

この項では、次のアクションについて説明します。

- ◆ 243 ページの 「WS 交換」
- ◆ 245 ページの 「XML 交換」

WS 交換

この項では、次のトピックについて説明します。

- ◆ 243 ページの 「[WS 交換] アクションについて」
- ◆ 243 ページの 「[WS 交換] アクションの追加」

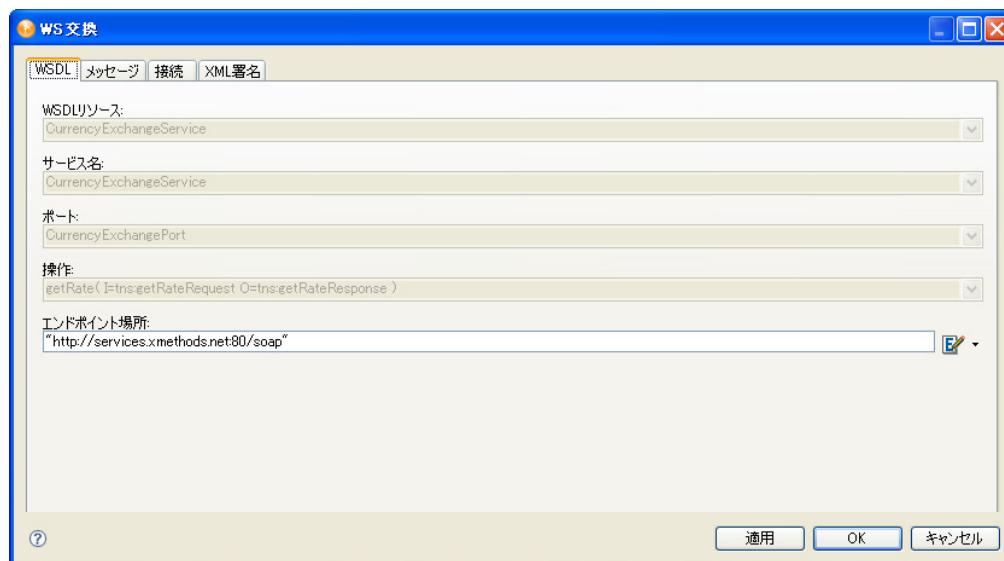
[WS 交換] アクションについて

[WS (Web サービス) 交換] アクションは統合アクティビティで最も重要なアクションであり、WSDL ファイルに指定された呼び出し規則に従って、統合アクティビティが Web サービスを呼び出せるようにします。統合アクティビティはアクションモデル作成時に [WS 交換] アクションを自動的に作成します。

ほとんどの場合、アクションモデルに別の [WS 交換] アクションを追加する必要はありません。しかし、そうする必要のある状況もあります。次の手順では、[WS 交換] アクションを追加する方法を説明します。

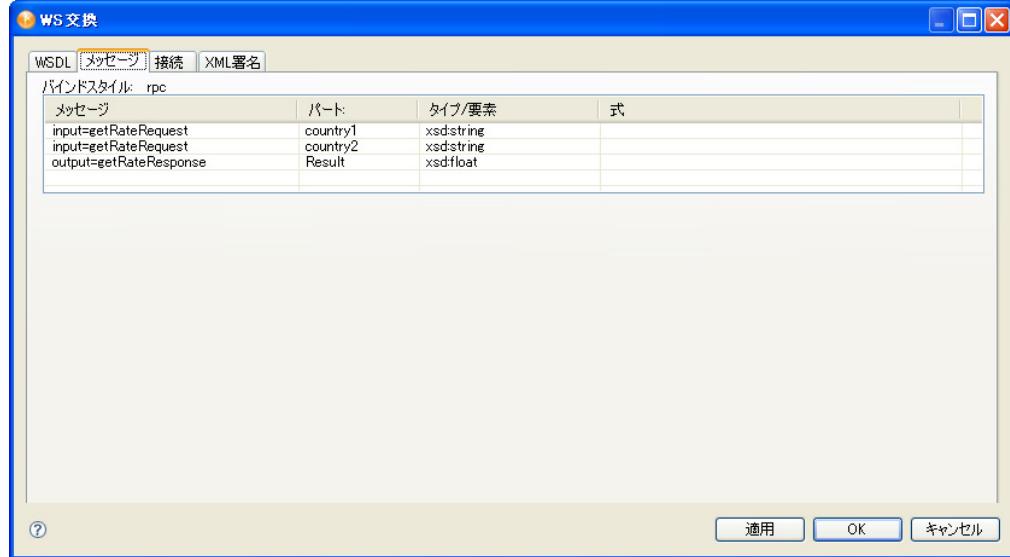
[WS 交換] アクションの追加

- 1 アクションモデル内で [WS 交換] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [データ交換] > [WS 交換] の順に選択します。



[WSDL リソース]、[サービス名]、[ポート]、および [操作] の各フィールドは、統合アクティビティ用に指定された WSDL の情報に基づいて自動的に入力されます。

- 3 必要に応じて、使用する Web サービスの [エンドポイントの場所] フィールド（通常はサーブレットを示す URL）の情報（引用符で囲む）を変更します。または、ランタイムに [エンドポイントの場所] に評価される ECMAScript 式を入力します。
- 4 [メッセージ] タブをクリックします。



[メッセージ]、[パート]、[種類/要素] の各フィールドは、統合アクティビティ用に指定された WSDL の情報に基づいて自動的に入力されます。

- 5 必要に応じて、メッセージの [式] 列をクリックし、ECMA 式ビルダを使用して、メッセージのソースとターゲットを記述する ECMAScript 式を作成します。通常、これは入力パートまたは出力パートに XPath の場所を指定する式です。
 - 6 [接続] タブをクリックします。
- このタブを使用して、認証が必要な HTTP サーバの接続パラメータを指定します。
- 7 接続に使用するユーザ ID を [ユーザID] フィールドに、ユーザのパスワードを [パスワード] フィールドに入力します。
- 実際には、接続確立中にユーザ ID とパスワードは送信されません。ここでは定義するだけです。パスワードは暗号化されます。ECMAScript の UserID および Password 変数にアクセスし、ユーザ ID とパスワードを画面の値としてマップできるようになります。このようにして、パスワードはだれからも見えなくなります。
- 8 接続にクライアント証明書が必要である場合、[クライアント証明書] フィールドの参照ボタンをクリックし、この接続に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。
 - 9 接続にクライアント秘密鍵が必要な場合は、[クライアント秘密鍵] フィールドの参照ボタンをクリックし、クライアント秘密鍵ファイルを選択して、クライアント秘密鍵を選択します。
 - 10 クライアント秘密鍵のパスワードを [秘密鍵パスワード] フィールドに入力します。
 - 11 接続タイムアウト値を [接続タイムアウト] フィールドに秒数で指定します。
 - 12 リアルタイムで [WS 交換] アクションをテストするには [適用] をクリックし、閉じるには [OK] をクリックします。

XML 交換

この項では、次のトピックについて説明します。

- ◆ 245 ページの 「[XML 交換] アクションについて」
- ◆ 245 ページの 「[XML 交換] アクションの追加」

[XML 交換] アクションについて

[XML 交換] アクションは、外部 XML ドキュメントを DOM に読み込み、DOM のデータを XML ファイルとして書き出します。[XML 交換] アクションには次の 4 種類があります。

- ◆ GET
- ◆ PUT
- ◆ POST
- ◆ POST with Response

GET 交換を使用する場合は、統合アクティビティに取り込む XML ドキュメントを示す URL を [交換 URL 式] フィールドに入力します。[応答パート] フィールドでは、XML を受信する DOM（入力、出力、_SystemFault、またはプロジェクト）を選択します。

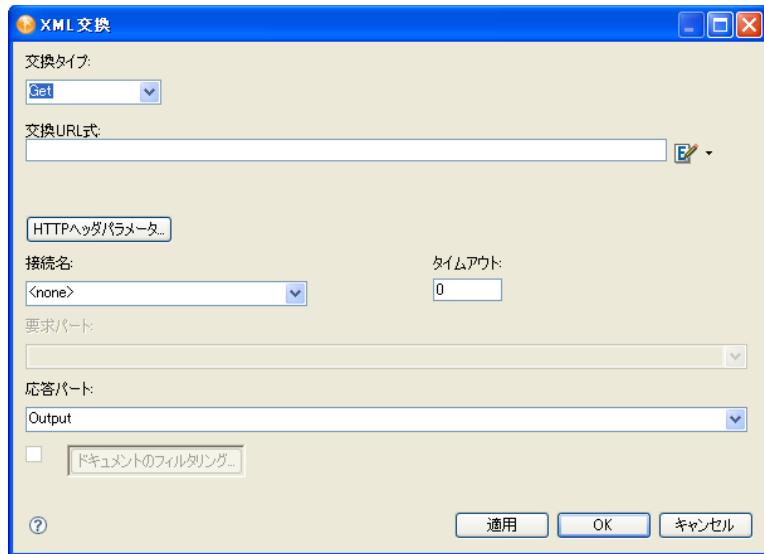
PUT 交換を使用する場合は、XML ドキュメントを書き込む場所を示す URL を [交換 URL 式] に入力します。[要求パート] フィールドでは、XML としてデータを送信する元の DOM の名前を選択します。

POST 交換を使用する場合は、XML ドキュメントを書き込む場所を示す URL を [交換 URL 式] に入力します。[要求パート] では、XML としてデータを送信する元の DOM の名前を選択します。

POST with Response 交換を使用する場合は、Post の場合と同じパラメータと、追加パラメータを 1 つ指定します。つまり、Post with Response 交換から応答 XML ドキュメントを受信するために、[応答パート] の DOM も指定する必要があります。Post 交換との違いは、Post with Response が起点サーバから応答 XML オブジェクトを予期するところにあります。

[XML 交換] アクションの追加

- 1 アクションモデル内で [XML 交換] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [データ交換] > [XML 交換] の順に選択します。



- 3 [交換タイプ] (Get、Put、Post、または Post with Response) を選択します。
- 4 [交換 URL 式] フィールドに、次のサポートされているプロトコルのいずれかを使用して、XML ドキュメントの完全修飾 URL を定義する式を入力します。

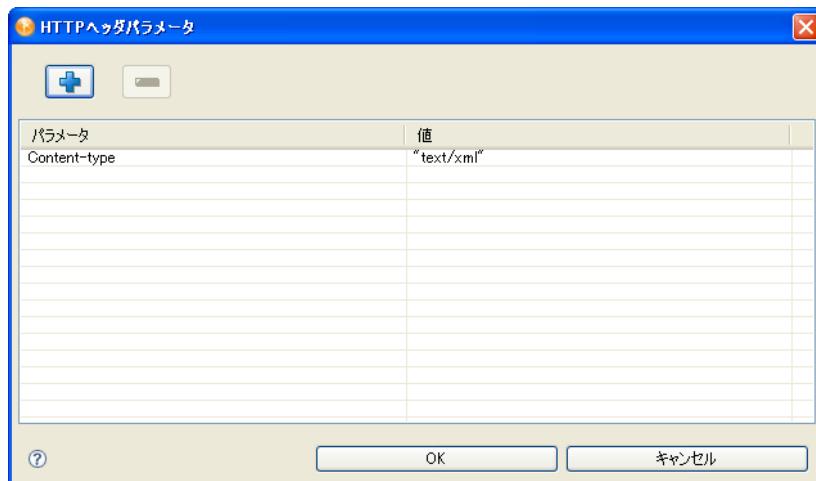
- ◆ file
- ◆ FTP
- ◆ HTTP
- ◆ HTTPS

選択した [交換タイプ] に応じて、この URL が XML 交換アクションに使用する XML ファイルのソースまたは宛先になります。例：

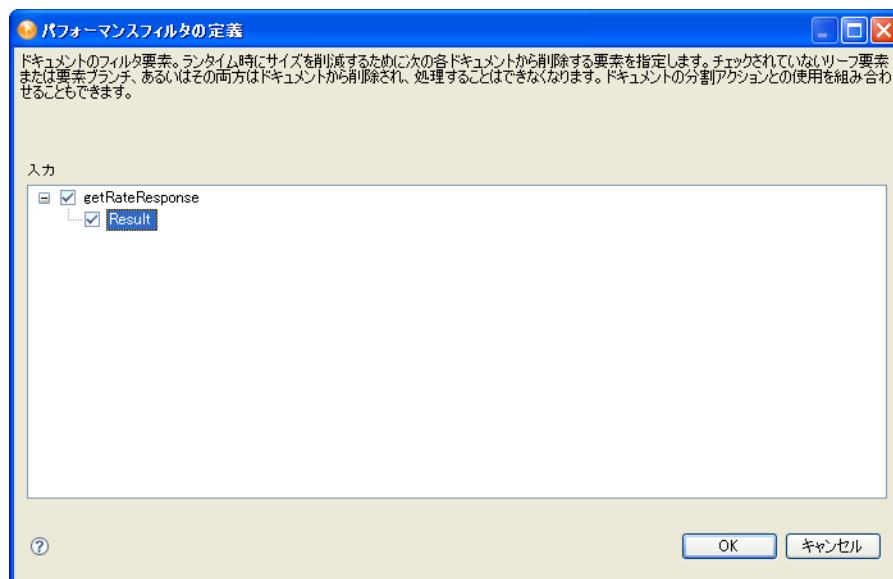
```
file:///g:/xmldata/invoicebatch1.xml  
ftp://accounting:password@123.456.789.987:21/invoices/inv1.xml
```

これは ECMAScript 式なので、URL 文字列は引用符で囲む必要があります。

- 5 HTTP ヘッダパラメータを指定する必要がある場合は、[HTTP ヘッダパラメータ] をクリックします。



- 6 プラス (+) アイコンをクリックして新しいヘッダパラメータを追加してから、パラメータ名とそれに対応する値を入力します。共通の HTTP ヘッダパラメータは「Content-Type」、「Content-Length」、および「Keep-Alive」です。パラメータと値のペアはいくつでも追加できます。
- 7 [OK] をクリックして、[XML 交換] ダイアログボックスに戻ります。
- 8 [要求パート] フィールドで ([交換タイプ] が Put、Post、または Post with Response の場合に有効になる)、XML としてデータを送信する送信元 DOM の名前を選択します。
- 9 [応答パート] フィールドで ([交換タイプ] が Put、Post、または Post with Response の場合に有効になる)、XML を受信する DOMツリーの名前を選択します。
- 10 着信する XML ドキュメントをフィルタしてパフォーマンスを改善する場合は、[ドキュメントのフィルタ] ボタンの隣にあるチェックボックスを選択してから、[ドキュメントのフィルタ] ボタンを選択します。



表示されるドキュメントは、[XML 交換] ダイアログボックスの [応答パート] で選択したドキュメントです。

このダイアログを使用して、着信する XML ドキュメントからリアルタイムに保持するノード（破棄するノードではない）を個別に指定して、パフォーマンスを改善し RAM オーバーヘッドを減らします。

- 11 ドキュメント内に保持するノードを選択します。
選択されなかったノードは DOM を解析する前に破棄されます。
- 12 保持するノードを選択したら、[OK] をクリックして [XML 交換] ダイアログボックスに戻ります。
- 13 [OK] をクリックします。
または、[適用] をクリックして、ダイアログボックスを閉じずに [XML 交換] アクションの影響を確認することができます。この方法で、[XML 交換] アクションを繰り返し編集して、すぐに結果を確認することができます。

9.5.3 繰り返し

この項では、次のアクションについて説明します。

- ◆ [248 ページの「中断」](#)
- ◆ [248 ページの「続行」](#)
- ◆ [249 ページの「グループの宣言」](#)
- ◆ [250 ページの「要素の繰り返し」](#)
- ◆ [252 ページの「グループの繰り返し」](#)
- ◆ [254 ページの「次の間の繰り返し」](#)

中断

この項では、次のトピックについて説明します。

- ◆ [248 ページの「\[中断\] アクションについて」](#)
- ◆ [248 ページの「\[中断\] アクションの追加」](#)

[中断] アクションについて

[中断] アクションは、「要素の繰り返し」、「グループの繰り返し」、または「次の間の繰り返し」ループの実行を停止します。アクションモデルは実行を継続し、ループの外にある次のアクションに進みます。

[中断] の使用が適しているのは、ループを使用して特定の1つの項目のノードリストを検索している場合などです。ターゲット項目が検出された場合は、繰り返しを継続する必要はありません。[中断] を使用して、ループを直ちに終了することができます。

注: [中断] アクションは、通常（ループ内の）[決定] アクションの1つのブランチで使用されます。[決定] アクションの [True] または [False] ブランチのいずれかに [中断] アクションを配置します。

[中断] アクションの追加

- 1 [中断] アクションを組み込むための変更を行う繰り返しアクション内で、[中断] アクションを配置するループ内部の位置を選択します。

一般に、これは [決定] アクションのどちらか一方の枝です。

- 2 [新規アクション] > [繰り返し] > [中断] の順に選択します。

[中断] アクションはアクションモデルに挿入されます。

続行

この項では、次のトピックについて説明します。

- ◆ [249 ページの「\[続行\] アクションについて」](#)
- ◆ [249 ページの「\[続行\] アクションの追加」](#)

[続行] アクションについて

[続行] アクションは、現在実行している [要素の繰り返し]、[グループの繰り返し]、または [次の間の繰り返し] ループなどの繰り返し部分を停止し、次の繰り返し部分をループの最初から実行します。[続行] アクションによって、ループ内の下流アクションをスキップしつつ、ループを続行して次の繰り返し部分に進むことができます。

[続行] アクションは、リスト内のある項目を何らかの理由でスキップしなければならないが、ループの実行は続行する必要がある場合に適しています。

注: [続行] アクションは、通常、ループ内の [決定] アクションの 1 つのブランチで使用します。[決定] アクションの [True] または [False] ブランチのいずれか適切なほうのブランチに、[続行] アクションを配置します。

[続行] アクションの追加

- 1 [続行] アクションを組み込むために変更する繰り返しアクション内で、[続行] アクションを配置する [ループ] アクション内部の位置を選択します。

通常、これは [決定] アクションのどちらか一方の枝の内部です。

- 2 [アクション] メニューから、[新規アクション] > [繰り返し] > [続行] の順に選択します。

[続行] アクションがアクションモデルに表示されます。

グループの宣言

この項では、次のトピックについて説明します。

- ◆ 249 ページの 「[グループの宣言] アクションについて」
- ◆ 249 ページの 「[グループの宣言] アクションの追加」

[グループの宣言] アクションについて

[グループの宣言] アクションは、DOM に関する 2 つの特別なリストを作成するときに使用します。作成したこれらのグループリストは、[グループの繰り返し] アクションでループの基礎として使用できます。リストを作成するには、グループ名と XPath を指定します。次いで、統合アクティビティは次のようにリストを作成します。

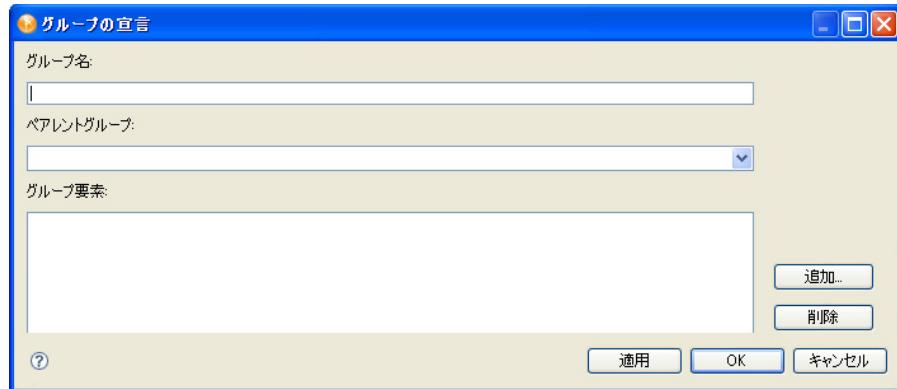
- ◆ グループリストを作成します。このリストには、XPath と一致するすべての要素で検出された固有値ごとに 1 つのエントリが含まれます。グループリストの名前は、指定したグループ名で示されます。
- ◆ グループリストの各固有エントリに関する詳細リストを作成します。このリストには、グループに属するメンバーの数のエントリが含まれます。詳細リストの名前は、指定したグループ名の後ろにラベル (Detail) を付けた形で示されます。

グループを使用すると、入力 DOM の繰り返し要素を選択し、その繰り返し要素のすべての兄弟の間で固有な値を使用することで、作成する要素の数を減らすことができます。出力 DOM では、要素が重複せず、固有な要素値ごとに要素が 1 つ存在します。

[グループの宣言] アクションの追加

- 1 アクションモデル内で [グループの宣言] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。

2 [新規アクション] > [繰り返し] > [グループの宣言] の順に選択します。



3 [グループ名] フィールドにグループの名前を入力します。

4 複数のグループレベルを作成する場合は、これまでに定義したグループが含まれている [親グループ] リストからグループを選択します。

5 [追加] をクリックします。[要素の追加] ダイアログボックスが表示されます。



6 パート名と要素を選択します。

7 [OK] をクリックします。

8 ステップ 5 からステップ 7 を繰り返して、グループに要素をさらに追加します。

9 グループに必要なすべての要素が揃ったら、[OK] をクリックします。

要素の繰り返し

この項では、次のトピックについて説明します。

- ◆ 250 ページの 「[要素の繰り返し] アクションについて」
- ◆ 251 ページの 「[要素の繰り返し] アクションの追加」

[要素の繰り返し] アクションについて

[繰り返し] アクションはアクションモデル内にループ構造を作成します。ループを使用すると、1つ以上の一連のアクションを繰り返すことができます。ループには、[要素の繰り返し]、[グループの繰り返し]、[次の間の繰り返し] の3種類があります。

XML では、ドキュメント内に、1 つの要素のインスタンスを複数含めることができます（データベーステーブルに複数のレコードがあることに似ています）。インスタンスの数はドキュメントごとに異なる可能性があり、ドキュメントスキーマ (DTD または XML スキーマ) に定義されます。たとえば、毎日の請求書の項目を含む XML ドキュメントを受信するとします。XML ドキュメントの項目の数は毎日異なります。XML ドキュメントに「項目」のインスタンスがいくつあるのかがわからないと、この項目の数を入力 XML ドキュメントから出力 XML ドキュメントにプログラムによって転送する場合、問題になります。[要素の繰り返し] アクションはこの問題を解決します。

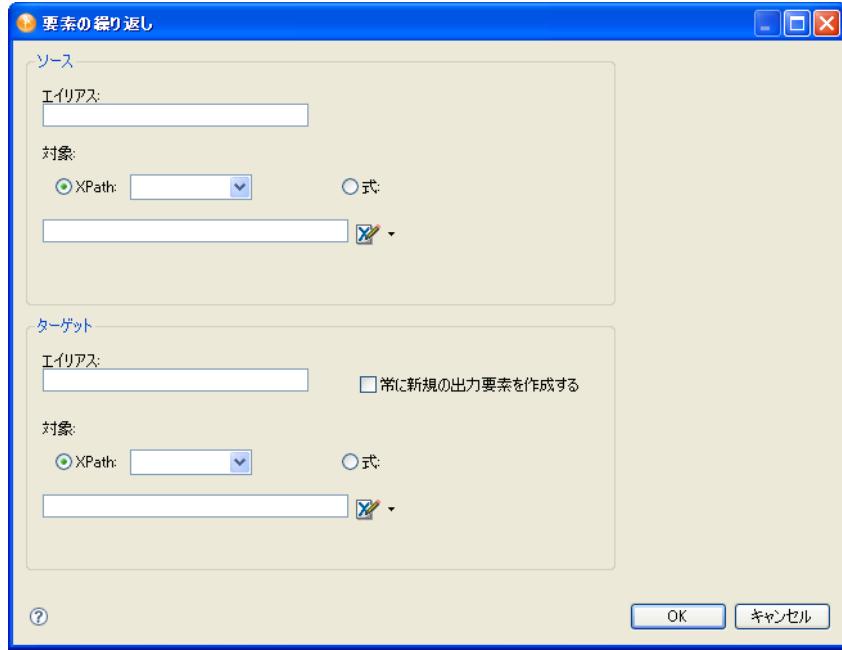
[要素の繰り返し] アクションを使用すると、複数回出現する要素にマークを付けることができます。このアクションは、マークが付いた要素の各インスタンスに対して、次のインスタンスが存在しなくなるまで 1 つ以上のアクションを実行する処理ループを設定します。前述の例では、処理ループに項目数を転送する単一の [マップ] アクションが含まれ、このアクションはすべての項目がマップされるまで繰り返されます。

[要素の繰り返し] アクションは、別名の概念も使用します。別名は 2 つの役割を果たします。別名はマークが付いた繰り返し要素の代替名または省略名なので、長い XPath 式を何回も指定せずに済みます。場合によっては、繰り返し要素がドキュメント階層の数レベル下にあることもあります。マークが付いた子要素を転送する [マップ] アクションを [繰り返し] ループ内に作成する場合に、別名を使用すれば長い XPath 式を入力し直すよりも短時間で済みます。別名は、[繰り返し] ループ内の [マップ] アクションにとって、ループが処理されるたびに繰り返し要素の次のインスタンスを使用するための標識にもなります。別名を使用しない [要素の繰り返し] ループ内の [マップ] アクションは、常にソースメッセージ内の要素の最初のインスタンスを参照します。

[要素の繰り返し] アクションを使用すると、ループ内で複数のアクションを処理できます。最も単純な例として、現在の要素インスタンスの値を入力パートから出力パートに転送する [マップ] アクションが繰り返しループに 1 つだけ含まれている場合があります。この処理ループに複数のアクションを定義することもできます。たとえば、現在の値を転送する [マップ] アクションや、各転送の監査を作成してファイルに書き込む [ログ] アクションがあります。

[要素の繰り返し] アクションの追加

- 1 アクションモデル内で [要素の繰り返し] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [繰り返し] > [要素の繰り返し] の順に選択します。



3 [ソース] 情報を指定します。

3a 別名をソースの [別名] フィールドに入力します。

別名の命名規則として良いのは、要素名にソースまたはターゲットを示すプレフィックスと繰り返しアクションの種類を付けて使用するというものです (S1Lineitem など)。

3b XPath 式を入力するか、または ECMA 式ビルダボタンをクリックして繰り返し要素の XPath 式を作成します。

4 [ターゲット] 情報を指定します。

4a ターゲットの [別名] フィールドに別名を入力します。

4b 既存の要素を更新するのではなく新しい要素を追加する繰り返しアクションがある場合は、[常に新しい出力要素を作成する] を選択します。

4c XPath 式を指定するか、または ECMA 式ビルダボタンをクリックして繰り返し要素の XPath 式を作成します。

5 [OK] をクリックします。[要素の繰り返し] ループがアクションモデルに追加されます。

6 アクションモデルで [ループアクション] をクリックして、ループ内で実行するアクションの追加を開始します。

グループの繰り返し

この項では、次のトピックについて説明します。

- ◆ 253 ページの 「[グループの繰り返し] アクションについて」
- ◆ 253 ページの 「[グループの繰り返し] アクションの追加」

[グループの繰り返し] アクションについて

受信する XML ドキュメントの形式が、常に自分のビジネスプロセスの要件に合う形式であるとは限りません。[グループの繰り返し] アクションを使用すると、データを再構築し、データの集計値を計算するフレームワークを確立できます。グループ化により、入力データの繰り返し要素を選択し、その繰り返し要素のすべての兄弟の間で固有な値を使用することで、作成する要素の数を減らすことができます。

[グループの繰り返し] アクションは、[グループの宣言] アクションによって作成される 2 つのリストのいずれかに基づいて、処理ループを設定します。ループは、使用するリスト（グループリストと詳細リストのどちらか）内にあるエントリ数と同じ回数だけ実行されます。[グループの繰り返し] と [マップ] コマンドを組み合わせることにより、元のものとは異なる構造とデータを持つ新しい XML ドキュメントを作成できます。

[要素の繰り返し] アクションと同じく、[グループの繰り返し] アクションも別名の概念を使用します。[グループの繰り返し] ダイアログボックスで使用するソースグループの値は、[グループの宣言] アクションで作成されたリスト名です。リスト名は 2 つの役割を果たします。リスト名は、ループ内の [マップ] アクションの XPath ソースの代替名または省略名です。リスト名を使用することで、長い XPath 式を何回も指定せずに済みます。グループリスト名を [マップ] アクションソースの DOM 名の代わりに使用すると、グループリスト名は [繰り返し] ループ内の [マップ] アクションにとって、ループが処理されるたびにグループリストの次のインスタンスを使用するための標識にもなります。グループ名を使用しない [グループの繰り返し] ループ内の [マップ] アクションは、常にソースパート内の要素の最初のインスタンスを参照します。

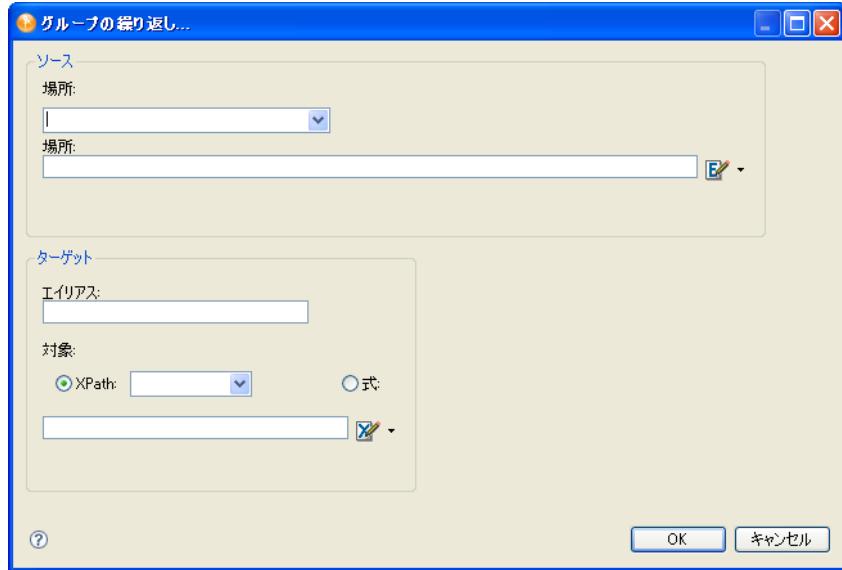
[グループの繰り返し] アクションで作成されたターゲット別名も、2 つの役割を果たします。ターゲット別名はループ内の [マップ] アクションの XPath ターゲットの代替名または省略名です。ターゲット別名を使用することで、長い XPath 式を何回も指定せずに済みます。ターゲット別名をパート名の代わりに使用すると、ターゲット別名は繰り返しループ内の [マップ] アクションにとって、ターゲットメッセージパートでソースの新しいインスタンスを作成するための標識にもなります。ターゲット別名を使用しない [グループの繰り返し] ループ内の [マップ] アクションは、常に、ターゲットメッセージパート内に作成された最初のインスタンスを、ソースグループリストの後続のインスタンスで上書きします。

[グループの繰り返し] アクションの作成は次の 3 つの作業からなります。

- ◆ [グループの宣言] アクションを作成して、グループリストを作成する。
- ◆ [グループの繰り返し] アクションを作成して、どのグループリストを使用するかを指定する。
- ◆ ループ内に [マップ] アクションを作成する。

[グループの繰り返し] アクションの追加

- 1 アクションモデル内で [グループの繰り返し] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [アクション] > [新規アクション] > [繰り返し] > [グループの繰り返し] の順に選択します。



- 3 [ソース] セクションで、[グループの繰り返し] アクションループの基になるグループ名を [場所] リストから選択します。
- 4 必要に応じて、[場所] フィールドに Where 文節を入力してグループリストをフィルタするか、または ECMA 式ビルダアイコンをクリックして Where 式を作成します。
- 5 ターゲット要素の別名がわかっている場合は、[別名] フィールドに名前を入力します。
- 6 別名がわからない場合は、[XPath] ボタンを選択してリストから要素を選択するか、または [式] ボタンを選択して式を入力します（または ECMA 式ビルダボタンをクリックして式を作成します）。
- 7 [OK] をクリックします。

次の間の繰り返し

この項では、次のトピックについて説明します。

- ◆ 254 ページの 「[次の間の繰り返し] アクションについて」
- ◆ 255 ページの 「[次の間の繰り返し] アクションの追加」

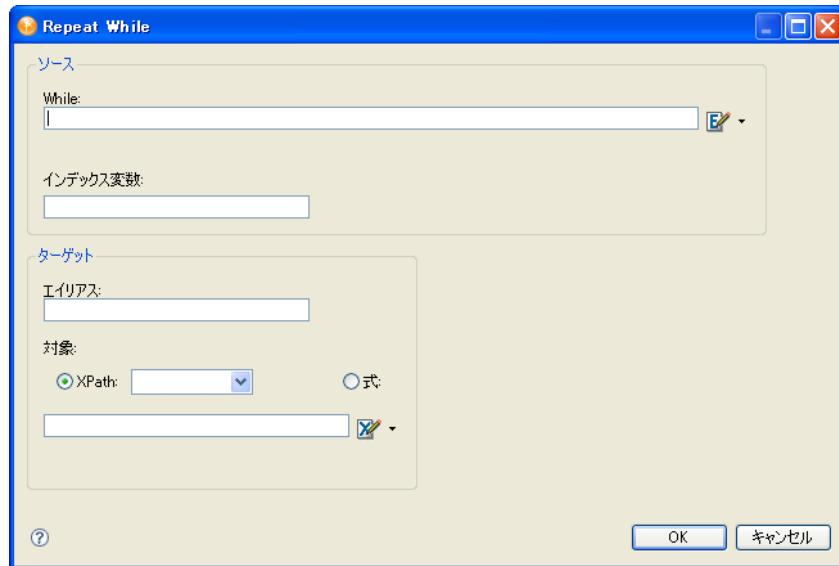
[次の間の繰り返し] アクションについて

[次の間の繰り返し] アクションは、指定した条件が True である限り 1 つ以上のアクションを繰り返します。[次の間の繰り返し] アクションに作成するターゲット別名は、2 つの役割を果たします。ターゲット別名はループ内の [マップ] アクションの XPath ターゲットの代替名または省略名です。ターゲット別名を使用することで、長い XPath 式を何回も指定せずに済みます。ターゲット別名を [マップ] アクションの DOM 名の代わりに使用すると、ターゲット別名は繰り返しループ内の [マップ] アクションにとって、ターゲット DOM でソースの新しいインスタンスを作成するための標識にもなります。ターゲット別名を使用しない [グループの繰り返し] ループ内の [マップ] アクションは、常に、ターゲット DOM 内に作成された最初のインスタンスを、ソースの後続のインスタンスで上書きします。

注: [要素の繰り返し] や [グループの繰り返し] とは異なり、[次の間の繰り返し] は DOMツリーのデータに基づく必要がありません。DOMツリーのデータとは独立してループを操作できます。

[次の間の繰り返し] アクションの追加

- 1 アクションモデル内で [次の間の繰り返し] アクションを追加する位置の 1 つ上の行を右クリックします (選択した行の下に新しいアクションが挿入されます)。
- 2 [新規アクション] > [繰り返し] > [次の間の繰り返し] の順に選択します。



- 3 [次の間] フィールドに、[次の間] ループをテストする式を入力するか、または ECMA 式ビルダボタンをクリックして式をビルドします。
- 4 [インデックス変数] フィールドに、ループの状況を監視する変数の名前を入力します。
- 5 ターゲット要素の別名がわかっている場合は、[別名] フィールドに名前を入力します。
- 6 別名がわからない場合は、[XPath] ボタンを選択してリストから要素を選択するか、または [式] ボタンを選択して式を入力します (または ECMA 式ビルダボタンをクリックして式を作成します)。
- 7 [OK] をクリックします。

9.5.4 コメント

この項では、次のトピックについて説明します。

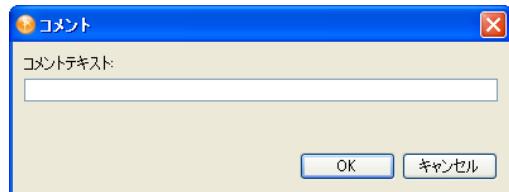
- ◆ 256 ページの 「[コメント] アクションについて」
- ◆ 256 ページの 「[コメント] アクションの追加」

[コメント] アクションについて

[コメント] アクションを使用すると、アクションモデルをドキュメント化して、実行する処理を明確にすることができます。アクションモデル内のどこでもコメントを追加できます。コメントはそれだけでは何も処理しません。

[コメント] アクションの追加

- 1 アクションモデル内でコメントを追加する位置の 1 つ上の行を右クリックします (選択した行の下に新しいアクションが挿入されます)。
- 2 [新規アクション] > [コメント] の順に選択します。



- 3 コメントを入力します。
- 4 [OK] をクリックします。

9.5.5 決定

この項では、次のトピックについて説明します。

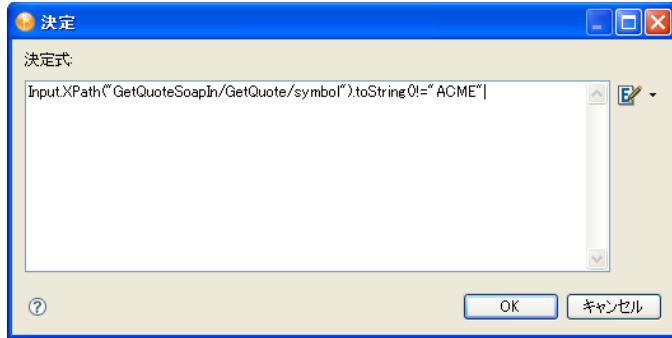
- ◆ 256 ページの 「[決定] アクションについて」
- ◆ 256 ページの 「[決定] アクションの追加」

[決定] アクションについて

[決定] アクションは、アクションどうしまたはアクションのグループの間に *if...then* 構成体を作成します。指定した条件に基づいてブランチを選択するときに、[決定] アクションを使用します。条件には ==、<、>、!、>=、<=、(&)、OR (||)、<> などの ECMAScript 比較演算子を使用する必要があります。式はブール値の True または False ステートメントに解決されなければなりません。

[決定] アクションの追加

- 1 アクションモデル内で [決定] アクションを追加する位置の 1 つ上の行を右クリックします (選択した行の下に新しいアクションが挿入されます)。
- 2 [新規アクション] > [決定] の順に選択します。



- 3 ECMAScript 式を入力するか、または ECMA 式ビルダボタンをクリックしてランタイムで *true* または *false* と評価される決定スクリプトを作成します。
- 4 [OK] をクリックします。

次のような [決定] アクションが表示されます。



- 5 アクションモデルペインで、[TRUE] ブランチを選択します。
 - 6 式が True の場合に実行するアクションを 1 つ以上追加します。
 - 7 [FALSE] ブランチを選択します。
 - 8 式が False の場合に実行するアクションを 1 つ以上追加します。
- [決定] アクションの TRUE または FALSE ブランチ内部に他の [決定] アクションをネストすることができます。

9.5.6 関数

- ◆ 257 ページの 「[関数] アクションについて」
- ◆ 257 ページの 「[関数] アクションの追加」

[関数] アクションについて

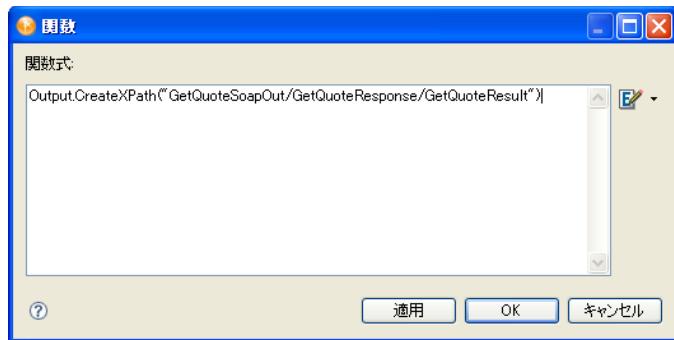
[関数] アクションは ECMAScript 関数を実行します。DOM 要素を操作するために、[関数] アクションで呼び出すスクリプトは現在の統合アクティビティ内の完全修飾 DOM 要素名を参照する必要があります。

作成してアクションモデルに追加したカスタムスクリプト関数は、どの XML ツリー要素に対しても実行できます。たとえば、データ要素の形式を変更する関数を作成できます。要素の内容に対して数値演算関数を実行する関数も作成できます。また、メッセージパートとのやり取りがないファイルシステム関数、データベース関数、URL 関数を実行することもできます。

[関数] アクションの追加

- 1 アクションモデル内で [関数] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。

2 [新規アクション] > [関数] の順に選択します。



3 [関数式] フィールドに関数を入力するか、または ECMA 式ビルダボタンをクリックして ECMAScript 式を作成します。

4 [OK] をクリックします。

9.5.7 ログ

- ◆ 258 ページの「[ログ] アクションについて」
- ◆ 258 ページの「[ログ] アクションの追加」

[ログ] アクションについて

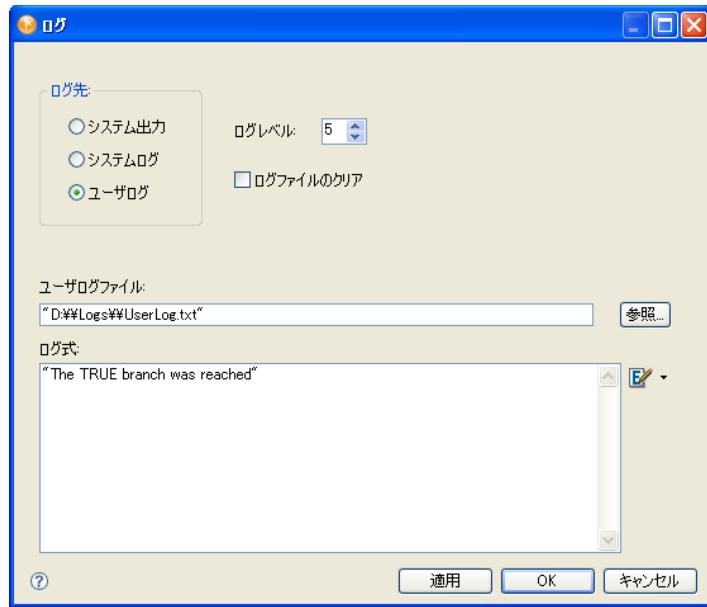
[ログ] アクションは、統合アクティビティにカスタマイズ可能な報告機能を提供します。ログレベル設定値を指定して、報告の程度を制御できます。

[ログ] の使用例は次のとおりです。

- ◆ [試行 / フォルト時] 条件に達したときに、特定のエラー情報をオペレータコンソールに書き込みます。
- ◆ ECMAScript 式を使用して、ランタイムにしかわからない値を持つ変数や DOM の内容に関する情報を記録することにより、デバッグを支援します。
- ◆ [要素の繰り返し] ループの各サイクルから特定の情報をキャプチャします。

[ログ] アクションの追加

- 1 アクションモデル内で [ログ] アクションを追加する位置の 1 つ上の行を右クリックします (選択した行の下に新しいアクションが挿入されます)。
- 2 [新規アクション] > [ログ] の順に選択します。



3 作成するログのタイプを [ログの出力先] グループから選択します。

[ログ] アクションはアクションで指定した場所に情報を書き込みます。ログ出力先には [システム出力]、[システムログ]、[ユーザログ] の 3 つの場所があります。

ログの場所	説明
システム出力	[ログ式] フィールドに指定するメッセージを、設計時にはオペレーティングシステムコンソールに、ランタイムにはアプリケーションサーバコンソールに書き込むには、[システム出力] を選択します。
	注: オペレーティングシステムコンソール上にメッセージを表示するには、Eclipse -debug および -consoleLog 起動パラメータを使用して Designer を起動します。
システムログ	[ログ式] フィールドに指定するメッセージをアプリケーションサーバログファイルに書き込むには、[システムログ] を選択します。
ユーザログ	[ログ式] フィールドに指定するメッセージを [ユーザログファイル] フィールドに指定するファイルに書き込むには、[システムログ] を選択します。

4 [ログレベル] を使用して、この [ログ] アクションの優先度レベル (1 から 10) を選択します。

デフォルトの優先度レベルは 5 です。ログファイルに表示するメッセージには、5 から 10 までの数値を割り当てる必要があります。ここで割り当てる優先度はしきい値の数値（内部的に 5 に設定されており変更不可）と比較されます。優先度がしきい値以上である場合、メッセージは記録されます。それ以外の場合、ログは記録されません。

- 5 コンポーネントを実行するたびにログファイルのデータをクリアする場合は、[ログファイルをクリアする] をオンにします。
- 6 [ログの出力先] グループで [ユーザログ] を選択した場合は、[ユーザログファイル] フィールドにログファイルのパスを入力するか、または [参照] を使用してログファイルを指定します。
存在しないファイルを指定すると、そのファイルが作成されます。Windows* システムの場合、パスを入力するときには、パスに円記号が出現する箇所には必ず 1 つ余分に円記号を追加します（たとえば、C:\Windows は C:\\Windows とします）。
- 7 ログに記録するメッセージを [ログ式] フィールドで作成します。
フィールドにメッセージを入力するか、または ECMA 式ビルダを使用して式を作成します。

9.5.8 マップ

この項では、次のトピックについて説明します。

- ◆ [260 ページの「\[マップ\] アクションについて」](#)
- ◆ [261 ページの「\[マップ\] アクションの追加」](#)
- ◆ [262 ページの「\[詳細\] マッピングオプション」](#)
- ◆ [265 ページの「コンテンツエディタを使用した要素の変換」](#)

[マップ] アクションについて

[マップ] アクションは、DOM ノードの入出力マッピングを実行します。このアクションは、あるドキュメントコンテキストのデータを別のドキュメントコンテキストに転送および変換します。コンテキストには 2 つのパートがあります。通常、最初のパートは DOM を識別し、2 番目のパートは DOM 内の場所を識別します。統合アクティビティの基本ドキュメントコンテキストは、DOM 名に XPath 式で表した要素の場所を組み合わせて示されます。DOM 名は通常 Input、Output、System Fault、または Project です。DOM 内の場所を識別する XPath 式には、パス要素が「/」で区切って指定されています。

注: 統合アクティビティのコンテキストは、それ自体は単に XPath 式の別名または省略名であるグループ名にすることもできます。

- ◆ [260 ページの「デフォルトのマッピング動作」](#)
- ◆ [261 ページの「マークアップを含むリーフ要素」](#)

デフォルトのマッピング動作

[マップ] アクションを使用して XML ドキュメント内の要素と属性をマップすると、特定のデフォルト動作が生じます。次の表に、これらのデフォルト動作を示します。

表 9-5 デフォルトのマッピング動作

マップタイプ	デフォルトの動作
リーフ要素からリーフ要素へ	要素データのみを転送します。
リーフ要素からブランチ要素へ	要素データのみを転送します。

マップタイプ	デフォルトの動作
ブランチ要素からリーフ要素へ	ブランチ全体（ブランチの下のすべての子要素と属性データを含む）を転送します。
ブランチ要素からブランチ要素へ	ターゲットの現在のブランチを削除した後、ブランチ全体（ブランチの下のすべての子要素と属性データを含む）を転送します。
リーフ要素のリストに含まれる特定のリーフ要素から要素へ	選択したリーフの要素データ（または要素インスタンス）をターゲット要素へ転送します。
属性から属性へ	属性データのみを転送します。
要素から属性へ	要素データを属性データに転送します。
属性から要素へ	属性データのみを転送します。

これらの動作の多くは、「高度なマッピング」ダイアログボックスを使用してアクションごとに変更できます（[「262 ページの「\[詳細\] マッピングオプション」](#)を参照）。

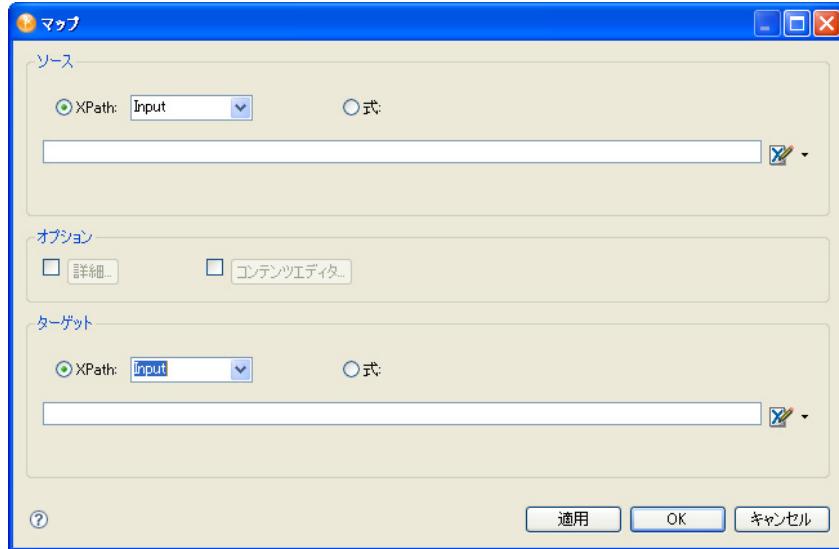
マークアップを含むリーフ要素

ランタイムに Java または ECMAScript 操作によって要素が入力されると、問題が発生する場合があります。要素はマークアップ（<や>などの不正な文字を含む文字列）が含まれるデータを受信する場合があります。統合アクティビティがこのような要素の内容を出力 DOM のノードにマップしようとすると、出力ドキュメントの形式が不正になります。統合アクティビティでは、マークアップを含むすべてのデータをターゲットドキュメントの新しい CDATA セクションにマッピングすることによって、この問題を解決します。

注：設計時にマークアップが入力されている場合は、動作が異なります。マークアップをノードに入力して [ソース] ビューで生 XML を調べてみると、ノードに入力したマークアップ文字がエンティティに変換されていることに気づきます。たとえば、「<」文字は < に変換されています。

[マップ] アクションの追加

- 1 アクションモデル内で [マップ] アクションを追加する位置の 1 つ上の行を右クリックします（選択した行の下に新しいアクションが挿入されます）。
- 2 [新規アクション] > [マップ] の順に選択します。



- 3 [ソース] セクションで、[XPath] を選択します。
- 4 リストからパート（入力、出力、_SystemFault、プロジェクト）を選択してから、適切な XPath 式を入力するか、または ECMA 式ビルダを使用して必要な要素を選択します。
それと同時に、パート名と XPath に [マップ] アクションのソースコンテキストを指定します。
- 5 ステップ 3 とステップ 4 を [ターゲット] セクションについて繰り返します。
- 6 マッピングをさらに調整するには、[詳細]（「[262 ページの「\[詳細\] マッピングオプション」](#)」を参照）または [コンテンツエディタ]（「[265 ページの「コンテンツエディタを使用した要素の変換」](#)」を参照）オプションを選択します。
[適用] をクリックすると、ダイアログボックスを閉じないで [マップ] アクションの影響を確認できます。この方法で、[マップ] アクションを繰り返し編集して、すぐに結果を確認することができます。
- 7 [OK] をクリックします。

【詳細】マッピングオプション

[マップ] ダイアログボックスの [詳細] オプションを選択すると、[詳細] ダイアログボックスが表示されます。[詳細] ダイアログボックスで設定するオプションは、現在の [マップ] アクションにのみ影響します。

図9-10 詳細



このダイアログボックスのオプションでは、入力パートノードを出力パートにマップする方法を細かく調整できます。

この項では、次のトピックについて説明します。

- ◆ 263 ページの 「属性のコピー」
- ◆ 264 ページの 「ディープコピー」
- ◆ 264 ページの 「ターゲットの作成」
- ◆ 264 ページの 「ターゲットを CDATA セクションとして作成」

属性のコピー

【属性のコピー】を使用して、属性をマップする方法を指定します。【属性のコピー】には次のオプションがあります。

表9-6 【属性のコピー】オプション

オプション	説明
リーフ以外のルートノードと従属要素の場合	端点ではない（リーフ以外の）要素を出力にマップする場合に、要素（その属性を除く）とその子を出力にマップすることを指定します。子の属性データは含まれますが、元の（親）要素の属性データは含まれません。
Never	マッピング時に属性データを（親ノードからもリーフノードからも）継承しないことを指定します。

オプション	説明
常時	すべてのノードのすべての属性データを出力にマップすることを指定します。

ディープコピー

デフォルトの統合アクティビティの動作では、一度にすべてのブランチをマップします（ターゲットノードとそのすべての子）。この操作をディープコピーを呼びます。場合によっては、この動作をオフにして、子を除いた親要素だけをコピーすることもできます。ディープコピーを無効にするには、[従属要素のマップ] の選択を解除します。

ターゲットの作成

[ターゲットの作成] は、ソースノードがソース DOM にあるかどうかに基づいて、[マップ] アクションの [ターゲット] グループに指定した宛先ノードを作成する場合に使用します（[261 ページの「[マップ] アクションの追加」] を参照）。デフォルトでは、ランタイムソース DOM にマッピングのソース XPath で指定したノードが含まれているかどうかにかかわりなく、統合アクティビティは常にターゲットを作成します。

たとえば、[マップ] アクションで、ソース XPath を次のように指定したとします。

```
$Input/Root/MySourceElement
```

ターゲット XPath で、次のように指定したとします。

```
$Output/Root/MyParentNode/SomeOtherElement
```

着信する入力ドキュメントに Root/MySourceElement に対応するノードがない場合、統合アクティビティはデフォルトで空の Root/MyParentNode/SomeOtherElement ノードを出力 DOM に作成します。場合によっては、これが希望する処理ではないこともあります。[ターゲットの作成] マッピングを使用することにより、デフォルト動作を変更できます。

表 9-7 ターゲットの作成

オプション	説明
ソースが存在する場合のみ	ソース XPath に指定されたノードが入力メッセージに存在していない場合は、[マップ] アクションをスキップすることを指定します（ターゲットノードは出力 DOM に作成されません）。
エラーの生成	ソース XPath に指定されたノードが入力ドキュメントに含まれていない場合、ランタイムにエラーと見なされることを指定します。したがって、[マップ] アクションを [Try/OnError] ブロックでラップして、エラーを処理できるようにするよう計画します。
常時	ターゲットノードを必ず作成することを指定します（デフォルトの動作）。[常時] を選択すると、[デフォルト値] フィールドを使用してターゲット要素のデフォルトデータ値を指定することができます。

ターゲットを CDATA セクションとして作成

[ターゲットを CDATA セクションとして作成] は、要素データを CDATA セクションにマップする方法を制御するときに使用します。

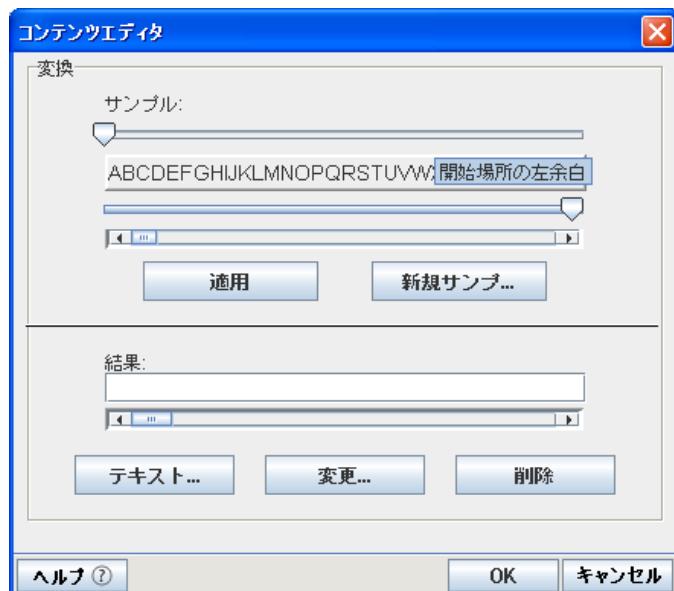
表9-8 ターゲットをCDATAセクションとして作成

オプション	説明
ソースにマークアップが含まれる場合のみ	ソースデータに（このコンテキストで）不正な文字が使用されている XML、HTML、または他のタイプのマークアップが含まれている場合に、データをターゲット DOM の CDATA セクションに変更せずに配置することを指定します。これはデフォルトの動作です。
Never	ソースデータを出力用に CDATA セクションでラップしないことを指定します。ソースデータに出現する不正な文字は、出力側でエスケープされたエンティティに変換されます（たとえば、> は > に変換されます）。
常時	どのような形式のソースデータであっても、出力時に CDATA セクションでラップすることを指定します。

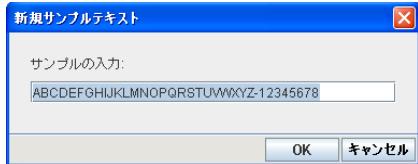
コンテンツエディタを使用した要素の変換

コンテンツエディタは、入力要素のフォーマットおよびコンテンツを出力要素の要件に合わせて変更するときに使用します。コンテンツエディタを使用することにより、入力データを小さいパートにスライスし、パート相互の相対的な位置を移動させたり、新しいパートを追加したり、一部のパートを省略したり、個々のパートに関数を適用したりできます。

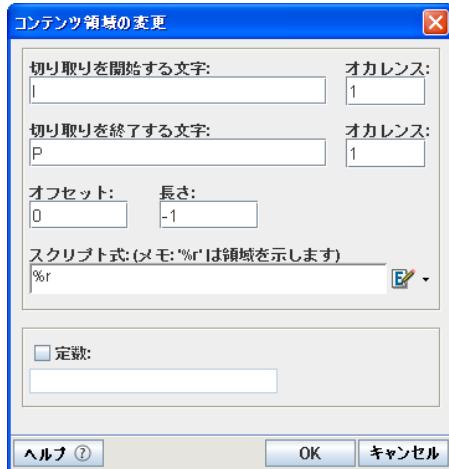
- 1 アクションモデルで、異なるパートから（たとえば、入力パートと出力パートから）マップする 2 つの要素を選択します。
- 2 [新規アクション] > [マップ] の順に選択します。
- 3 [マップ] アクションダイアログボックスで（「[261 ページの「\[マップ\] アクションの追加」を参照](#)」）、[コンテンツエディタ] チェックボックスを選択し、[コンテンツエディタ] ボタンをクリックします。



- 4 必要に応じて、[新規サンプル] をクリックしてサンプル文字列を入力します。



- 5 [OK] をクリックして [コンテンツエディタ] ダイアログボックスに戻ります。
- 6 [サンプル] セクションで、サンプルの上にあるスライダを切り取り開始位置に移動してから、サンプルの下のスライダを切り取り終了位置に移動します。
これらのスライダにより、入力データから部分文字列をどのように取り出すかが決まります。
- 7 [適用] をクリックします。
部分文字列は別個のオブジェクトとして [結果] フィールドにコピーされます。
- 8 サンプルの各部分について任意の順序で、**ステップ 6** と **ステップ 7** を繰り返します。
この方法を使用することにより、元の入力の部分文字列から新しい文字列を取り出して作成することができます。
- 9 [結果] フィールドのオブジェクトの形式を変更するには、次のようにします。
 - 9a オブジェクトを選択します。
 - 9b [変更] をクリックします。



[切り取りを開始する文字] フィールドには、最初の切り取りを行う文字列内の文字が表示されます。最初の [出現箇所] フィールドには、いつ切り取りを行うかが示されます。上の図では、最初の切り取りが文字 1 の最初の出現箇所で行われます。[切り取りを終了する文字] フィールドには、最後の切り取りを行う文字列内の文字を表示します。2 番目の [出現箇所] フィールドには、いつ切り取りを行うかが示されます。[オフセット] フィールドには、オブジェクトの開始位置にする、元の文字列の始めからの文字数が表示されます。[長さ] フィールドにはオブジェクトの長さが表示されます。

- 9c 必要に応じて、[スクリプト式] フィールドに ECMAScript 式を入力して、コンテンツ領域を変更できます。
[スクリプト式] フィールドに示されている '%r' は、関数を適用するコンテンツ領域を表すローカル変数です。たとえば、.toUpperCase() 関数をコンテンツ領

域に適用するには、`var test='%r'; test.toUpperCase()` というスクリプト式を入力します。

9d オブジェクトに定数を割り当てるには、[定数] を選択してから、定数文字列を入力します。

9e コンテンツエディタで文字列形式のマッピングが完了したなら、[OK] をクリックして変更内容を保存し、コンテンツエディタを閉じます。

10 [OK] をクリックして、アクションモデルに戻ります。

ECMA 式の使用

10

この項では、ECMA 式ビルダの使用方法の詳細について説明します。主なトピックは次のとおりです。

- ◆ [269 ページのセクション 10.1 「ECMA 式ビルダについて」](#)
- ◆ [279 ページのセクション 10.2 「ECMAScript の例」](#)
- ◆ [282 ページのセクション 10.3 「ECMAScript API」](#)

10.1 ECMA 式ビルダについて

Designer には、ECMAScript インタープリタと式エディタが組み込まれています。式エディタでは、ワークフローデータを参照したり変更したりするスクリプト式を作成できます。たとえば、スクリプティングを使って次のことができます。

- ◆ フローデータ要素の下にワークフローに必要なデータ項目を作成する。
- ◆ データに対して基本的な文字列操作、日付操作、数値操作、リレーションナル操作、連結操作、および論理操作を実行する。
- ◆ より高度なデータ操作を実行するため、標準またはカスタム Java クラスを呼び出す。
- ◆ 式を使ってランタイム制御を行い、次の操作を実行する。
 - ◆ フォームフィールドラベルを変更または上書きする。
 - ◆ フォームフィールドデータを初期化する。
 - ◆ 電子メールアドレスとコンテンツをカスタマイズする。
 - ◆ エンタイトルメントの付与または取り消しの権限およびパラメータを設定する。
 - ◆ 終了したアクティビティのデータを評価し、条件アクティビティを使って条件付きでワークフローパスをたどる。
 - ◆ 条件付きでトリガされる異なったログメッセージを单一のログアクティビティで書き出す。

この項では、スクリプトティングの使用時に利用できる技術や機能について説明します。

- ◆ [270 ページのセクション 10.1.1 「ECMAScript について」](#)
- ◆ [270 ページのセクション 10.1.2 「ECMAScript の機能」](#)
- ◆ [270 ページのセクション 10.1.3 「ECMA 式ビルダの使用」](#)
- ◆ [276 ページのセクション 10.1.4 「Java の統合について」](#)
- ◆ [277 ページのセクション 10.1.5 「XPath の統合について」](#)
- ◆ [279 ページのセクション 10.1.6 「パフォーマンスについて」](#)

注：ワークフローの式を定義するには、ワークフローアクティビティの設定方法について理解しておく必要があります。また、ワークフロー内で使用できるさまざまなデータタイプについても知っておく必要があります。ワークフローアクティビティの詳細については、[187 ページの第 8 章「ワークフローアクティビティの参照」](#) を参照してください。

ワークフロー内で使用可能なシステム変数については、89 ページのセクション 4.3.3 「ワークフローデータについての理解」を参照してください。

10.1.1 ECMAScript について

ECMAScript は、ホスト環境（この場合は Designer）でオブジェクトの操作に使用するオブジェクト指向のスクリプト言語です。ECMAScript (ECMA-262 と ISO/IEC 16262) は、JavaScript (Netscape*) と JScript* (Microsoft*) の基盤になる標準ベースのスクリプト言語です。ECMAScript は、Designer のグラフィカルユーザインターフェースのような、ホスト環境の既存の機能を補ったり拡張したりするために使用します。ホスト環境である Designer は、ECMAScript に処理のため、さまざまなオブジェクトへのアクセスを提供します。ECMAScript は、それらのオブジェクトを操作する Java によく似た言語を提供します。

ECMAScript は、拡張性に富み、正規表現などの強力な文字列処理ツールを備えています。このため、Designer の機能を拡張する理想的な言語として利用できます。

注 : ECMAScript の詳細については、[欧州コンピュータ製造工業会 \(ECMA\) の Web サイト](http://www.ecma-international.org) (<http://www.ecma-international.org>) を参照してください。

10.1.2 ECMAScript の機能

スクリプティングを利用すると、細かく調整されたカスタムロジックをワークフローに組み込むことができるだけでなく、データ操作の柔軟性も大幅に向上します。これは、ECMAScript 拡張機能の一部として、DOM および XPath 関連のオブジェクトやメソッドが式ビルダに組み込まれているからです。

ECMAScript は、特にメモリ内の DOM を操作するときに役に立ちます。ECMA 式ビルダは、W3C DOM Level 2 の仕様に従って、XML ドキュメントをメモリ内オブジェクトとして作成します。DOM 2 仕様では、ECMAScript バインディング (W3C 推奨仕様 「★ [ECMAScript Language Binding ★](http://www.w3.org/TR/DOM-Level-2-Core/ecma-script-binding.html)」 (<http://www.w3.org/TR/DOM-Level-2-Core/ecma-script-binding.html>) を参照) が定義され、DOM ツリーコンテンツにすぐにアクセスできる多数のメソッドおよびプロパティが用意されています。ECMA 式ビルダは、フローデータの DOM を認識します。ECMA 式ビルダでは、W3C で定義されている DOM 用のあらゆる ECMAScript 拡張機能を使用できます。

ECMAScript は、XPath などの他の式言語と DOM を結び付けるブリッジとしても機能します。このため、XPath 構文を DOM に使用し、ドキュメント構造内のさまざまな要素を処理できます。

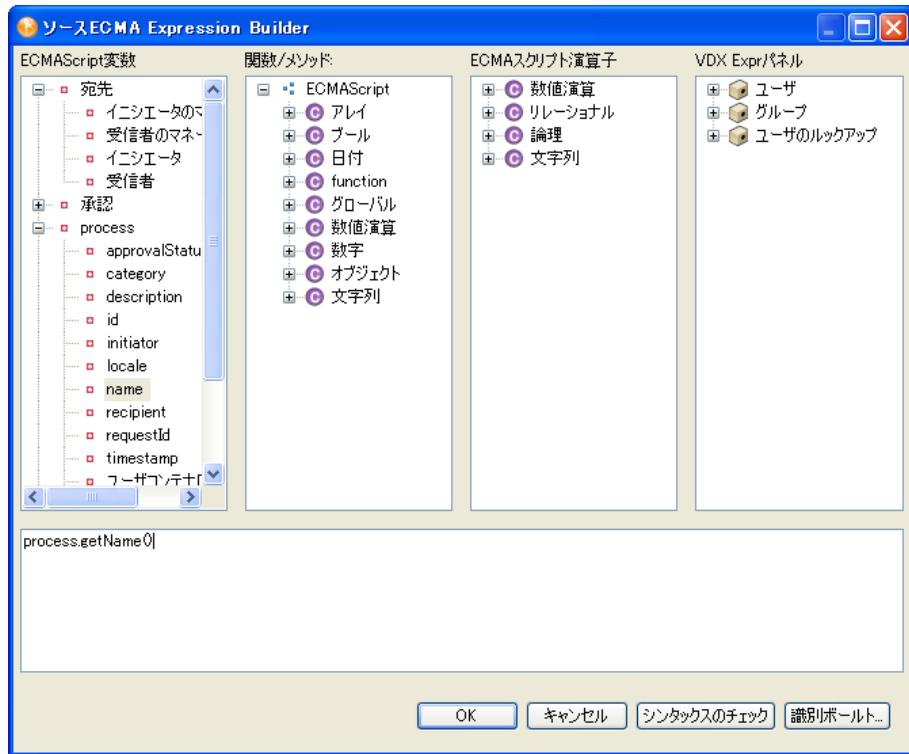
10.1.3 ECMA 式ビルダの使用

Designer では、ユーザアプリケーションデザインツールのさまざまな場所から ECMAScript にアクセスできます。最も一般的なアクセスフォームは、式ビルダからのアクセスです。次のボタンが表示されれば、いつでも式ビルダを表示できます。



このボタンは、条件アクティビティの「プロパティ」画面や、エンタイトルメントプロビジョニングアクティビティの「データ項目マッピング」ビューなどの Designer のディスプレイにあります。このボタンをクリックすると、ECMA 式ビルダが表示されます。

図 10-1 ECMA 式ビルダ



ECMA 式ビルダのトップペインには、使用可能なオブジェクト、メソッド、およびプロパティの選択リストがあります（すべてサイズ変更可能）。項目をロールオーバーすると、ECMAScript ステートメントの作成に役立つツールヒントが表示されます。選択リスト内の任意の項目をダブルクリックすると、ウィンドウ下部の編集ペインに、対応する ECMAScript ステートメントが表示されます。図では、「ECMAScript 変数」ペインで「プロセス」選択リストが選択され、「名前」変数がダブルクリックされています。このワークフロー変数のコンテンツにアクセスできる ECMAScript 式が、編集ペインに自動的に挿入されます。

この項では、次のトピックについて説明します。

- ◆ 272 ページの 「構文のチェック」
- ◆ 272 ページの 「DN の選択」
- ◆ 272 ページの 「ECMAScript 変数」
- ◆ 273 ページの 「関数 / メソッド」
- ◆ 273 ページの 「ECMAScript 演算子」
- ◆ 275 ページの 「VDX Expr」
- ◆ 275 ページの 「特殊文字の使用」

構文のチェック

ECMA 式ビルダには、[シンタックスのチェック] ボタンがあります。このボタンをクリックすると、ECMAScript インタープリタにより、式の構文がチェックされます。ECMAScript の構文に問題が見つかった場合は、エラーメッセージが表示されます。その場合は式を編集し、必要に応じて再検証することができます。検証はオプションです。

注：構文のチェック時には、式は実行されません。構文がチェックされるだけです。ECMAScript はインターフリタ言語なので、構文のチェック時には、ランタイムに依存する式はチェックされません。これらの式が有効な ECMAScript 構文に準拠しているかどうかが確認されるだけです。

DN の選択

ECMA 式ビルダには、[識別ポールト] ボタンもあります。このボタンは、識別ポールトから DN を選択する必要があるアクティビティ（開始アクティビティ、承認アクティビティ、エンタイトルメントアクティビティなど）の操作中に表示されます。

図 10-2 [識別ポールト] ボタン



[識別ポールト] ボタンをクリックすると、識別ポールトをナビゲートして DN を選択するためのダイアログボックスが表示されます。識別ポールトに接続していない場合は、識別ポールトを使用できないことを示すため、[識別ポールト] ボタンがかすんだ表示になります。

ECMAScript 変数

このペインには、現在のコンテキストに関連する変数の名前が表示されます。たとえば、プロビジョニング要求定義エディタを使用している場合、現在のワークフロープロセスのシステム変数、現在のアクティビティのシステム変数、および現在のワークフローで作成されたフローデータ変数が表示されます。変数名をダブルクリックすると、その変数がスクリプトに挿入されます。ワークフローで使用できるシステム変数については、[89 ページのセクション 4.3.3 「ワークフローデータについての理解」](#)を参照してください。

ECMA 式ビルダでは、次の 2 つのメソッドを使ってフローデータ変数を読み取ることができます。

表 10-1 フローデータ変数を読み取るメソッド

メソッド	説明
<code>flowdata.get(variable-name)</code>	ワークフロードキュメント内の変数 (XPath 式表示) のノード値として文字列を返します。
<code>flowdata.getObject(variable-name)</code>	ワークフロードキュメント内の変数 (XPath 式表示) のノード値としてオブジェクトを返します。このメソッドを使って複数値コントロールの値を取得できます。

関数 / メソッド

ECMA 式ビルダで使用できる関数とメソッドの説明は、[282 ページのセクション 10.3 「ECMAScript API」](#) を参照してください。

ECMAScript 演算子

次の表では、ECMA 式ビルダで使用できる演算子について説明します。

表 10-2 Math

演算子	説明
+ 加算	2つの数値（リテラルまたは変数）の和を返します。
- 減算	一方の数からもう一方の数を差し引きます。
* 乗算	2つの数値（リテラルまたは変数）の積を返します。
/ 除算	一方の数をもう一方の数で割ります。

表 10-3 割り当て

演算子	説明
= 割り当て	右辺のオペランドの値を左辺のオペランドに割り当てます。
+= これに足す	左辺のオペランドと右辺のオペランドを足し合わせた結果を左辺のオペランドに割り当てます。たとえば「 $a += b$ 」は「 $a = a + b$ 」と同じことです。
-= これから引く	左辺のオペランドから右辺のオペランドを差し引いた結果を左辺のオペランドに割り当てます。たとえば「 $a -= b$ 」は「 $a = a - b$ 」と同じことです。
*= これに掛ける	2つのオペランドを掛け合わせた結果を左辺のオペランドに割り当てます。たとえば「 $a *= b$ 」は「 $a = a * b$ 」と同じことです。
/= これを割る	左辺のオペランドを右辺のオペランドで割った結果を左辺のオペランドに割り当てます。たとえば「 $a /= b$ 」は「 $a = a / b$ 」と同じことです。
%= モジュラス	左辺のオペランドを右辺のオペランドで割った余りを左辺のオペランドに割り当てます。たとえば「 $a %= b$ 」は「 $a = a \% b$ 」と同じことです。
&= Bitwise AND をこれに適用する	オペランドの Bitwise AND を実行し、結果を左辺のオペランドに割り当てます。たとえば「 $a &= b$ 」は「 $a = a \& b$ 」と同じことです。
= Bitwise OR をこれに適用する	オペランドの Bitwise OR を実行し、結果を左辺のオペランドに割り当てます。たとえば「 $a = b$ 」は「 $a = a b$ 」と同じことです。

演算子	説明
<code><<= Bitwise 左シフトをこれに適用する</code>	オペランドの Bitwise 左シフトを実行し、結果を左辺のオペラントに割り当てます。たとえば「 <code>a <<= b</code> 」は「 <code>a = a << b</code> 」と同じことです。
<code>>>= Bitwise 記号付き右シフトをこれに適用する</code>	オペランドの Bitwise 右シフトを実行し、結果を左辺のオペラントに割り当てます。たとえば「 <code>a >>= b</code> 」は「 <code>a = a >> b</code> 」と同じことです。
<code>>>>= Bitwise 記号なし右シフトをこれに適用する</code>	オペランドの Bitwise 記号なし右シフトを実行し、結果を左辺のオペラントに割り当てます。たとえば「 <code>a >>>= b</code> 」は「 <code>a = a >> b</code> 」と同じことです。

表 10-4 その他

演算子	説明
% モジュラス	左辺のオペランドを右辺のオペランドで割った余りの整数を返します。
<code>++ 自動増分</code>	オペランドを 1 増分します（オペランドの前後で使用可能）。
<code>-- 自動減分</code>	オペランドを 1 減分します（オペランドの前後で使用可能）。
<code>~ Bitwise NOT</code>	オペランドのビットを反転させます。
<code>& Bitwise AND</code>	両方のオペランドの対応ビットが 1 であるとき、各ビット位置に 1 を返します。
<code> Bitwise OR</code>	少なくとも一方のオペランドの対応ビットが 1 であるとき、各ビット位置に 1 を返します。
<code>^ Bitwise XOR</code>	どちらか一方だけのオペランドの対応ビットが 1 であるとき、各ビット位置に 1 を返します。
<code><< Bitwise 左シフト</code>	第 1 オペランドのバイナリ表現の数字を、第 2 オペランドで指定された桁数だけ左にシフトします。このとき、空になった右端の下位ビット位置には 0 が入り、左端の上位ビットは破棄されます。
<code>>> 記号付き Bitwise 右シフト</code>	第 1 オペランドのバイナリ表現の数字を、第 2 オペランドで指定された桁数だけ右にシフトし、右端の下位ビットは破棄されます。空になった左端には、最上位（左端）ビットがコピーされ、元の符号を保持します。
<code>>>> 記号なし Bitwise 右シフト</code>	第 1 オペランドのバイナリ表現の数字を、第 2 オペランドで指定された桁数だけ右にシフトします。右端の下位ビットは破棄され、空になった左端の上位ビットには 0 が入ります。

表 10-5 リレーションナル

演算子	説明
<code>== 等しい</code>	右辺のオペランドの値を左辺のオペランドに割り当てます。
<code>!= 等しくない</code>	両辺のオペランドが等しくない場合にブール値の <code>true</code> を返します。

演算子	説明
< より小さい	左辺のオペランドが右辺のオペランドより小さい場合に true を返します。
> より大きい	左辺のオペランドが右辺のオペランドより大きい場合に true を返します。
<= より小さいか等しい	左辺のオペランドが右辺のオペランドより小さいか右辺のオペランドと等しい場合に true を返します。
>= より大きいか等しい	左辺のオペランドが右辺のオペランドより大きいか右辺のオペランドと等しい場合に true を返します。

表 10-6 論理演算

演算子	説明
&& AND	両辺のオペランドが true の場合はブール値の true 、そうでない場合は false を返します。
OR	どちらかのオペランドが true の場合は true を返します。両辺のオペランドが false の場合は false を返します。
! NOT	一方のオペランドを true に変換できる場合（またはこれがブール値でない場合）は false を返します。オペランドを false に変換できる場合は true を返します。

表 10-7 文字列

演算子	説明
+ 連結	2つの文字列オペランドを連結し、その和集合になる文字列を返します。

VDX Expr

このペインでは、識別ボルトに定義されているエンティティ定義（43 ページのセクション 3.2 「エンティティおよび属性の操作」を参照）を、スクリプトに挿入することができます。システムエンティティ、ユーザ定義エンティティの両方を使用できます。識別ボルトからデータを取得する式のフォーマットは、次のとおりです。

```
IDVault.get(dn, object-type, attribute)
```

たとえば、データ項目の受信者のマネージャ情報を取りたい場合、VDX Expr パネルの「[ユーザ]」ノードを開き、「マネージャ」項目をダブルクリックします。すると、`IDVault.get ({ ここに DN 式を入力 }, 'user', 'manager')` という式が挿入されます。この式は、マネージャの DN (LDAP 識別名) の文字列を評価します。

特殊文字の使用

ECMA 式ビルダでは、エスケープシーケンスを使って、リテラル文字列で特殊文字を使用できます。エスケープシーケンスはバックスラッシュ文字 (\) で開始します。次の表に、よく使用するエスケープシーケンスを示します。

表 10-8 エスケープシーケンス

エスケープシーケンス	文字
\b	バックスペース
\f	フォームフィード
\n	復帰改行
\r	キャリッジリターン
\t	水平タブ
\"	二重引用符
\	円記号 (\)
\'	アポストロフィ

\u とそのすぐ後に続く 4 桁の 16 進数を使って、ユニコード文字を指定することもできます。次に例を示します。

表 10-9 ユニコード文字のエスケープシーケンス

エスケープシーケンス	文字
\u00A3	ポンド記号 (£)
\u20AC	ユーロ記号 (€)

10.1.4 Java の統合について

Java は、外部 Java オブジェクトへのプリッジを提供する ECMA 式ビルダを通して、ワークフロープロセスに統合されています。ECMA 式ビルダから Java クラスにアクセスするには、ワークフローエンジンのクラスパスにクラスを配置する必要があります。このためには、ユーザアプリケーション WAR ファイル (IDM.war) 内の WEB-INF\lib ディレクトリに Java クラスを追加します。

注: フォームアクションスクリプトは、プロビジョニング要求定義エディタのその他の部分で使用できる ECMAScript とは異なり、サーバ上ではなくブラウザ上で実行されます。フォームアクションスクリプトからのディレクトリアクセスはすべて、ブラウザからサーバへの AJAX 呼び出しを通して処理されます。詳細については、[282 ページのセクション 10.3.1 「フォームアクションスクリプトメソッド」](#) を参照してください。

ユーザアプリケーション WAR への Java クラスの追加

- 1 WAR ファイルユーティリティを使って IDM.war ファイルを開きます。IDM.war ファイルは、アプリケーションサーバの \server>IDM\deploy ディレクトリ内にあります。
- 2 WEB-INF\lib ディレクトリに Java クラスをコピーします。

ECMAScript からの Java へのアクセス

Java クラスにアクセスするには、ECMA 式ビルダでインライン関数を作成します。関数のインスタンスを作成し、この関数内で ECMAScript 構文を使って Java メソッドを呼び出します。次に、ベクタの作成例を示します。

```
function list() { v=new java.util.Vector(); v.add('{Enter Item 1}'); v.add('{Enter Item 2}'); return v; } list();
```

カスタム Java クラスにアクセスするには、クラス名の前に「Packages」というプレフィックスを付けます。次に例を示します。

```
v = new Packages.com.novell.myClass("value");
```

ECMA 式ビルダは Mozilla* Rhino に基づいています。Rhino は、完全に Java で記述された JavaScript のオープンソース実装です。ECMAScript から Java にアクセスする方法の詳細については、「★ Scripting Java ★」(<http://www.mozilla.org/rhino/ScriptingJava.html>) を参照してください。

10.1.5 XPath の統合について

この項では、次のトピックについて説明します。

- ◆ 277 ページの 「ワークフロー内の XPath」
- ◆ 278 ページの 「統合アクティビティの XPath」

ワークフロー内の XPath

プロビジョニング要求定義ワークフローは、フローデータという特殊なオブジェクトをサポートします (89 ページのセクション 4.3.3 「ワークフローデータについての理解」を参照)。フローデータオブジェクトは、DOM の一種です。つまり、メモリ内のオブジェクトとして作成された XML ドキュメントです。XPath 構文を使って、フローデータ DOM の構造をナビゲートしたり、要素やコンテンツを追加、変更、または削除したりできます。

フローデータへのオブジェクトの追加

構文	例
 flowdata.  arguments	flowdata.parent/child[1] flowdata.reason

フローデータからのオブジェクトの取得

構文	例
 flowdata.get  arguments	flowdata.getObject('parent/child[1]') flowdata.get('reason')

flowdata.get () メソッドと flowdata.getObject () メソッドについては、272 ページの表 10-1 を参照してください。

統合アクティビティの XPath

ECMAScript インタープリタは、統合アクティビティ内のカスタムメソッド XPath () を認識します。このメソッドでは、次のような式を作成できます。

```
Input.XPath("GetBNQuoteSoapIn/GetBNQuote/sISBN")
```

ECMA 式ビルダで、選択リストからノードを選択すると、このような式が自動的に作成されます。

統合アクティビティでは、W3C で採用されている XPath アドレス指定構文を使用します。XPath 構文は、URI アドレス指定構文と似ていますが、XML のアドレス指定や操作に役立つ高度な機能を多数備えています。次の表に、よく使用する構文規則をいくつか紹介します。

表 10-10 XPath 構文

XPath 構文	説明
/	スラッシュ 1 個は、要素の絶対パスを表します。たとえば、/ABC と指定すると、ルート要素 ABC が選択されます。
//	2 重のスラッシュは、条件を満たすパス内のすべての要素を表します。たとえば、//ABC と指定すると、すべての ABC のオカレンスが選択されます。//ABC//DEF と指定すると、ABC の子であるすべての DEF 要素が選択されます。
*	アスタリスクを指定すると、先行するパス内のすべての要素が選択されます。たとえば、*ABC/DEF と指定すると、要素 ABC/DEF で囲まれたすべての要素が選択されます。/* と指定すると、すべての要素が選択されます。
[]	角括弧は、特定の要素を指定するときに使用します。たとえば、/ABC[3] と指定すると、ABC 内の 3 番目の要素が選択されます。つまり、角括弧は、SQL ステートメントの Where 句に類似したフィルタとして使用できます。//ABC["Table"] と指定すると、「Table」というコンテンツを持つすべての要素が選択されます。
@	アットマークは、指定された属性を持つ要素を選択するときに使用します。たとえば、/ABC@name と指定すると、ABC 内の「name」という属性を持つすべての要素が選択されます。
	縦棒は、複数のパスを指定するときに使用します。たとえば、//ACB//DEF と指定すると、ACB 内のすべての要素と DEF 内のすべての要素が選択されます。
\$	ドル記号は、現在のドキュメントの他に参照するドキュメントがある場合に使用します。INVOICEBATCH/INVOICE[SELLER/NAME= \$PROJECT/USERCONFIG/COMPANYNAME]
関数 ()	XPath には、XPath アドレスに追加できる多数の関数が用意されています。たとえば、//*[count (*) =2] と指定すると、子を 2 つ持つすべての要素が選択されます。
数値演算子 ()	XPath には、XPath アドレスに追加できる多数の数値演算子が用意されています。たとえば、/ABC[position () mod 2 = 0] と指定すると、ABC 内のすべての偶数要素が選択されます。

使用できるすべての演算子については、W3 推奨の [XML Path Language \(XPath\) \(http://www.w3.org/TR/xpath.html\)](http://www.w3.org/TR/xpath.html) を参照してください。

10.1.6 パフォーマンスについて

ECMAScript はインタープリタ言語なので、式に含まれるすべてのスクリプト行が解析され、同等の Java 構文に変換されてから実行されます。このため、コードのオーバーヘッドが大きくなり、純粋な Java と比べてスクリプトの実行速度が遅くなります。

ECMAScript を使用する前に、パフォーマンスの問題について考慮する必要があります。

現在のコンポーネントおよびサービスで最適なパフォーマンスを得るには、次のガイドラインが役に立ちます。

- ◆ ECMAScript から呼び出すことができるカスタム Java クラスを使ってタスクを実行できるかどうかを考える。
- ◆ スクリプティングによる細かい調整が必要な場合は、ECMAScript を使用する。

パフォーマンス向上の鍵は、実装方法にあります。たとえば、正しいアルゴリズムを選択し、変数ができるだけ再使用するということです。処理の速い言語で書かれた不適切なコードよりも、処理の遅い言語で書かれた良いコードのほうが、高いパフォーマンスを発揮することがあります。Java で書かれたコードのほうが、ECMAScript で書かれた同等のロジックよりパフォーマンスが高いとは限りません。なぜなら、Java には Java のオーバーヘッド制約があるからです。たとえば、コンストラクタの呼び出しチェーンがその好例です（他のオブジェクトを継承する Java オブジェクトのコンストラクタを呼び出すと、その上位（祖先）のすべてのオブジェクトが呼び出される）。

ECMAScript のコアオブジェクト（String、Array、Date など）には、データの操作、書式設定、解析、ソート、文字列変換、配列変換などに使用する便利なメソッドが多数組み込まれています。これらのメソッドは、高度に最適化された Java コードによって、インタープリタ内に実装されています。データ解析や書式設定のための関数をカスタマイズして作成してもかまいませんが、利用できるときはこれらのメソッドを利用するほうが賢明です。たとえば、長い文字列を区切り記号の位置で複数の部分文字列に分割したい場合、`indexOf()` と `substring()` の 2 つの String メソッドを含むループを作成する方法があります。ループを使って文字列の解析を行い、結果として生成された部分文字列を配列内のスロットに割り当てます。しかし、この方法は効率がよくありません。そこで、次のようなシンプルな方法を使用することもできます。

```
var myArrayOfSubstrings = bigString.split( delimiter );
```

ECMAScript の String メソッド `split()` は、指定の区切り記号の位置で文字列を分割し、部分文字列の配列を生成します。このメソッドはネイティブ Java で実行されるので、1 行のスクリプトを変換するだけでもインタープリタが必要です。一方、ループを使って `indexOf()` と `substring()` を繰り返し呼び出す方法では、インタープリタは不要な上、関数呼び出しのオーバーヘッドもかかるないので、パフォーマンスに悪影響がありません。

組み込みの ECMAScript メソッドを上手に利用することで、パフォーマンスを向上させることができます。大量のスクリプトを使用する場合は、ECMAScript 言語について詳しく学ぶことをお勧めします。そのことが、パフォーマンスのボトルネックの解決に役立ちます。

10.2 ECMAScript の例

この項では、ECMAScript を使って実行できる一般的な操作の例を紹介します。

- ◆ 280 ページのセクション 10.2.1 「一般的な例」

- ◆ 280 ページのセクション 10.2.2 「フローデータの例」
- ◆ 280 ページのセクション 10.2.3 「フォームコントロールの例」
- ◆ 281 ページのセクション 10.2.4 「エラー処理」

10.2.1 一般的な例

ECMA 式ビルダで関数を作成する場合、次のようにインライン関数を作成します。

```
function abc() { var v1 = "" ; for ( i = 0; i < 9 ; i++) v1 += "$";  
return v1; } ; abc();
```

10.2.2 フローデータの例

この項では、フローデータオブジェクトの使用方法を示すスクリプティングの例を紹介します。

- ◆ 280 ページの 「フローデータ変数の値の取得」
- ◆ 280 ページの 「子要素を持つ XML 要素を作成し、フローデータに追加する」

フローデータ変数の値の取得

先ほどの例では、ApprovalStatus という属性を持つ start_reason という XML 要素とその子要素 approval_reason を作成して、フローデータに承認ステータスの情報を入力しました。ApprovalStatus 属性の値を取得するには、前動作マップ内で次の式を使用します。

```
flowdata.get('start_reason/approval_reason/@ApprovalStatus')
```

この式を入力するには、ECMA 式ビルダの [ECMAScript 変数] ペインで、[フローデータ] ノードを展開して、[ApprovalStatus] 属性をダブルクリックします。

図 10-3 属性の選択



子要素を持つ XML 要素を作成し、フローデータに追加する

先ほどの例では、ApprovalStatus というフォームフィールドへのユーザ入力を取得しました。ここでは、この情報をフローデータに追加し、ダウンストリームのアクティビティで使用できるようにします。後動作マップ内で次の式を使用します。

```
flowdata.start_reason/approval_reason/@ApprovalStatus
```

10.2.3 フォームコントロールの例

この項では、フォームコントロールを使ったスクリプティングの例を紹介します。

- ◆ 281 ページの 「フォームフィールドの値の取得」
- ◆ 281 ページの 「複数値コントロールからの個々の値の取得」
- ◆ 281 ページの 「リストやチェックボックス項目にデータを取り込む」

- ◆ 281 ページの 「DN の比較」

フォームフィールドの値の取得

ApprovalStatus という名前のフォームフィールドがあるとします。このフィールドの値を取得するには、前動作マップで次の式を使用します。

```
process.get('ApprovalStatus')
```

この式を入力するには、ECMA 式ビルダの [ECMAScript 変数] ペインで、[プロセス] ノードを開き、[ApprovalStatus] をダブルクリックします。

複数値コントロールからの個々の値の取得

複数値コントロール（たとえば [colors] というチェックボックス）からの個々の値を取得するには、まずそのコントロールをフローデータに追加する必要があります。アップストリームアクティビティの後動作マッピングで次の式を使用します。

```
flowdata.colors
```

[colors] から 1 つの値（たとえば最初の値）を取得するには、ダウンロードストリームアクティビティで次の式を使用します。

```
flowdata.getObject('colors[1]')
```

リストやチェックボックス項目にデータを取り込む

スクリプトを使ってリストコントロール（PickList、MVEditor など）や [MVCheckbox] コントロールにデータを取り込むには、前動作マッピングで次のような式を使用します。

```
function list() {var l=new java.util.Vector();l.add('Blue');l.add('Red'); l.add('Green'); return l;} list();
```

DN の比較

複数の DN を比較してそれらが等しいかどうかを確認するには、次のような式を使用します。

```
if ( IDVault.DNcompare(flowdata.get('Activity3/CardRequest/Candidate'),recipient) ) true; else false ;
```

この比較では、大文字と小文字は区別されません。たとえば、DNCompare で次の DN を比較した場合、True が返されます。

```
CN=jdoe,ou=users,ou=idmsample,o=acmecn=JDOE,ou=users,ou=idmsample,o=acme
```

10.2.4 エラー処理

エラー処理の方法は、前動作マップと後動作マップで異なります。後動作マップの場合、承認アクティビティまたは条件アクティビティのエラーフローパスを使って、後動作マッピング中に発生したエラーをキャッチできます。この方法は、前動作マップでは使用できません。なぜなら、データ取得の処理中のエラーは、フォームがユーザに表示される前に発生するからです。この場合、フォーム下部のフォームコントロールの代わりに、次のようなエラーメッセージがユーザに表示されます。

```
XXXX FAILED to generate form due to: No data items are available!
```

このシナリオでは、前動作マップ内のフィールドのソース式で try-catch ステートメントを使用できます。

```
function getTheData() {    var theData;    try {        theData = IDVault.get( 'cn=jsmith,ou=users,ou=idmsample1,o=acme' , 'user', 'FirstName') + ' ' + IDVault.get( 'cn=jsmith,ou=users,ou=idmsample1,o=acme' , 'user', 'LastName');    }    catch (error) { theData = 'Error retrieving data.'; }    return theData;};getTheData();
```

10.3 ECMAScript API

この項では、次のトピックについて説明します。

- ◆ [282 ページのセクション 10.3.1 「フォームアクションスクリプトメソッド」](#)
- ◆ [292 ページのセクション 10.3.2 「DOM メソッド」](#)
- ◆ [318 ページのセクション 10.3.3 「ECMAScript Core」](#)
- ◆ [337 ページのセクション 10.3.4 「Global 関数」](#)
- ◆ [337 ページのセクション 10.3.5 「IDVault 関数」](#)

10.3.1 フォームアクションスクリプトメソッド

ECMAScript はワークフローのその他のコンポーネントでも実行できますが、フォームスクリプトはサーバでは実行できません。フォームスクリプトを実行できるのは、Web ブラウザだけです。フォームスクリプトからのすべてのディレクトリアクセスは、ブラウザからサーバへの AJAX 呼び出しによって処理されます。

この項では、ECMA 式ビルダで使用できるすべてのフォームアクションメソッドとプロパティを一覧表示します。この項では、次のトピックについて説明します。

- ◆ [282 ページの「フォーム」](#)
- ◆ [287 ページの「フィールド」](#)
- ◆ [290 ページの「イベント」](#)
- ◆ [291 ページの「リスト」](#)
- ◆ [291 ページの「クエリ」](#)
- ◆ [292 ページの「コンテナ」](#)

フォーム

フォームメソッドの操作に使用します。この項では、次のメソッドについて説明します。

- ◆ [283 ページの「focus \(fieldname\)」](#)
- ◆ [283 ページの「select \(fieldname\)」](#)
- ◆ [283 ページの「activate \(fieldname\)」](#)
- ◆ [283 ページの「setRequired \(fieldname, is-required\)」](#)
- ◆ [284 ページの「InterceptAction \(actionname, order, function\)」](#)
- ◆ [284 ページの「getLocale \(\)」](#)
- ◆ [284 ページの「getRBMMessage \(\)」](#)

- ◆ 284 ページの 「stringToDate ()」
- ◆ 284 ページの 「dateToString ()」
- ◆ 285 ページの 「isValidDate (date)」
- ◆ 285 ページの 「isValidDate (date,include-time)」
- ◆ 285 ページの 「alert (string)」
- ◆ 285 ページの 「showMsg (string)」
- ◆ 285 ページの 「showWarning (string)」
- ◆ 285 ページの 「showError (string)」
- ◆ 286 ページの 「showFatal (string)」
- ◆ 286 ページの 「enable (fieldname)」
- ◆ 287 ページの 「disable (fieldname)」
- ◆ 287 ページの 「getValue (fieldname)」
- ◆ 287 ページの 「getValues (fieldname)」
- ◆ 287 ページの 「setValues (fieldname)」

focus (fieldname)

```
form.focus(fieldname)
```

指定のフィールドにフォーカスを設定します。リストベースまたは選択ベースのコントロールでは、選択された選択肢または選択がされなかった場合は最初の選択肢にフォーカスを設定します。fieldname パラメータによって渡されるフィールドがリスト形式または選択形式の場合、このパラメータに対応する項目にフォーカスを設定します。値が配列の場合は、最初の値のみが、フォーカスを設定するチェックボックスまたはラジオボタンの決定に使用されます。指定のフィールドが非表示または無効になっている場合、このメソッドは無効です。

select (fieldname)

```
form.select(fieldname)
```

fieldname パラメータが渡されない場合は、その下のテキストフィールドにフォーカスを設定します。リストベースまたは選択ベースのコントロールでは、選択された選択肢または選択がされなかった場合は最初の選択肢にフォーカスを設定します。値パラメータが渡され、フィールドがリストベースまたは選択ベースの場合、fieldname パラメータに対応する選択肢にフォーカスを設定します。フィールドが無効または非表示になっている場合、このメソッドは無効です。

activate (fieldname)

```
form.activate(fieldname)
```

setFocus () と select () の組み合わせです。

setRequired (fieldname, is-required)

```
form.setRequired("fieldname", is-required)
```

フィールドを、is-required が True の場合は必須、それ以外の場合はオプションに設定します。必須フィールドが空の場合、フォームの送信がブロックされます。

InterceptAction (actionname, order, function)

```
form.interceptAction("actionname", "order", "function")
```

アクションボタンに添付されているスクリプトを遮断します。渡された関数は、order パラメータに基づいて実行されます。

有効な actionname 値は、SubmitAction と CancelAction です。

承認フォームで使用できる actionname は、ApprovalAction、RefusalAction、DenyAction、UpdateAction、CancelAction、および CommentAction です。order パラメータには、次の値を指定できます。before: ボタンに添付されているスクリプトが実行される前に関数を呼び出します。after: ボタンのスクリプトが実行された後で関数を呼び出します。around: ボタンのスクリプトを実行するかどうかを決定するパラメータを渡すことができます。次は、送信アクションを遮断する例です。フォームは、ユーザが [はい] と応答した場合にのみ送信されます。

```
window.inv=function (invocation) { if (confirm( "Are you sure you want to submit ?")) { var result = invocation.proceed(); return result; };  
};  
form.interceptAction("SubmitAction", "around", window.inv);
```

getLocale ()

```
form.getLocale()
```

現在のロケールを返します。ロケールパラメータをサポートするすべてのメソッドの入力として使用できます。

getRBMessage ()

```
form.getRBMessage(key)  
form.getRBMessage(key, value[s])  
form.getRBMessage(key, value[s], bundle)
```

このメソッドは ID (bId) のリソースバンドル内の、キー「key」を持つエントリを検索します。リソースバンドルの Java クラスは、java.util.ListResourceBundle.Parameter を拡張します。このパラメータは、たとえば「msg」というメッセージ内で、({0}, {1}, など) の代わりに次のようなパラメータを渡すことができます。

```
var msg = frm.getRBMessage ("mykey", ["value0", "value1"],  
"mybundle");
```

stringToDate ()

```
form.stringToDate(date)  
form.stringToDate(date, include-time)
```

日付文字列を Date に変換します。フォーマットは DatePicker で使用されているように、現在のロケールの日付フォームを使用してください。DatePicker コントロールの値は、このメソッドで変換できます。例：

```
form.showMsg("Date="+form.stringToDate(d,true));
```

dateToString ()

```
form.dateToString(date)  
form.dateToString(date, include-time)
```

次のように、日付を DatePicker に格納できる文字列に変換します。

```
var d = form.dateToString(new Date(), true);
form.setValues("hireDate", d);
```

isValidDate (date)

```
form.isValidDate(date)
```

日付文字列の書式が正しいかどうかを検証するために使用します。

isValidDate (date,include-time)

```
form.isValidDate(date, include-time)
```

日付文字列の書式が正しいかどうかを検証するために使用します。

alert (string)

```
form.alert("msg")
```

アラートボックスにメッセージを表示します。

showMsg (string)

```
form.showMsg(msg, param, bId)
```

フォームのステータス部分にメッセージを追加します。msg 文字列パラメータには、メッセージテキスト自体か、リソースバンドル bId 内のエントリをポイントするキーを含めることができます。このメソッドは、常に、bId という ID を持つリソースバンドル内で、msg というキーを持つエントリの検索を試みます。param パラメータは、msg 内のステークホルダ ({0}, {1}, など) の置換として渡すために使用できます。

注: スクリプトにデバッグメッセージを追加したい場合は、form.showDebugMsg () を使用することをお勧めします。

例：

```
form.showMsg("my message" {0},{1}, ["value0","value1"]);
```

showWarning (string)

```
form.showWarning(msg, param, bId)
```

フォームのステータス部分に警告を追加します。

msg 文字列パラメータには、警告テキスト自体か、リソースバンドル bId 内のエントリをポイントするキーを含めることができます。このメソッドは、常に、bId という ID を持つリソースバンドル内で、msg というキーを持つエントリの検索を試みます。param パラメータは、msg 内のステークホルダ ({0}, {1}, など) の置換を渡すために使用できます。

注: スクリプトにデバッグメッセージを追加したい場合は、form.showDebugMsg () を使用することをお勧めします。

例：

```
form.showWarning("my warning" {0},{1}, ["value0","value1"]);
```

showError (string)

```
showError (msg, param, bId) ;
```

フォームのステータス部分にエラーメッセージを追加します。

msg 文字列パラメータには、エラーテキスト自体か、リソースバンドル bId 内のエントリをポイントするキーを含めることができます。このメソッドは、常に、bId という ID を持つリソースバンドル内で、msg というキーを持つエントリの検索を試みます。param パラメータは、msg 内のステークホルダ ({0}, {1}, など) の置換を渡すために使用できます。

注: スクリプトにデバッグメッセージを追加したい場合は、form.showDebugMsg () を使用することをお勧めします。

通常のエラーと致命的なエラーのどちらも、フォーム送信をブロックします。通常のエラーは、フォーム検証の直前に（フォーム送信のため）リセットされるという点で、致命的なエラーと異なります。致命的なエラーは記録されるので、再起動しない限りフォーム送信をブロックします。通常のエラーは、検証フェーズで生成された場合にのみ、フォーム送信をブロックします。 onload イベントまたはカスタムイベントの実行中に通常のエラーが発生しても、フォーム送信時までにエラーは削除されます。

注: スクリプトにデバッグメッセージを追加したい場合は、form.showDebugMsg () を使用することをお勧めします。

例：

```
form.showError("my error" {0},{1}, ["value0","value1"]);
```

showFatal (string)

```
form.showFatal ("my fatal" {0},{1}, ["value0","value1"]);
```

フォームのステータス部分に致命的なエラーメッセージを追加します。

msg 文字列パラメータには、致命的なエラーテキスト自体か、リソースバンドル bId 内のエントリをポイントするキーを含めることができます。このメソッドは、常に、bId という ID を持つリソースバンドル内で、msg というキーを持つエントリの検索を試みます。param パラメータは、msg 内のステークホルダ ({0}, {1}, など) の置換を渡すために使用できます。

通常のエラーと致命的なエラーのどちらも、フォーム送信をブロックします。通常のエラーは、フォーム検証の直前に（フォーム送信のため）リセットされるという点で、致命的なエラーと異なります。致命的なエラーは記録されるので、再起動しない限りフォーム送信をブロックします。通常のエラーは、検証フェーズで生成された場合にのみ、フォーム送信をブロックします。 onload イベントまたはカスタムイベントの実行中に通常のエラーが発生しても、フォーム送信時までにエラーは削除されます。

注: スクリプトにデバッグメッセージを追加したい場合は、form.showDebugMsg () を使用することをお勧めします。

例：

```
form.showFatal("my fatal" {0},{1}, ["value0","value1"]);
```

enable (fieldname)

```
form.enable("fieldname")
```

フォームのフィールドを有効にします。

```
disable (fieldname)  
form.disable("fieldname")
```

フォームのフィールドを無効にします。

注: 無効になったフィールドも、ワークフローエンジンにデータを返し続けます。無効になったフィールドのコンテンツは、フォーム送信時または `field.validate ()` メソッドの呼び出し時に検証されます。

```
getValue (fieldname)  
form.getValue("fieldname")
```

フィールドの最初の値を返します。フィールドのデータタイプに関係なく、常に文字列タイプを返します。フィールドに値が入っていない場合、テキスト入力フィールド (Text、TextArea、DatePicker、DNLookup など) の場合、空の文字列を返します。選択ベースのコントロール (StaticList、ラジオボタン、チェックボックスなど) の場合、「未定義」というメッセージを返します。DN タイプコントロールの場合、常に DN を返し、表示式は返しません。

```
getValues (fieldname)  
form.getValues("fieldname")
```

値を含む文字列配列を返します。値が見つからない場合、配列は空（サイズ = 0）になります。DN タイプコントロールの場合、常に DN を返し、表示式は返しません。

```
setValues (fieldname)  
form.setValues("fieldname", data-values, display values,  
KeepOldValues)
```

値を設定します。複数値をサポートします。このメソッドでは、リストベースのコントロール (StaticList、MVCheckbox、PickList など) の使用可能なエントリを変更できます。デフォルトでは、`KeepOldValues` パラメータが `True` でない限り、既存の値は削除されます。リストベースでないコントロールの場合、`display values` パラメータは無視されます。

フィールドの初期値を設定または変更したい場合は、「`onload`」イベントで実行します。

注: このメソッドは、フィールドの `onchange` イベントをトリガします。

例 :

```
field.setValues("cn=jdoe,ou=users,ou=mysample,o=novell"); // for a  
DNLookupfield.setValues(["jdoe@novell.com", "test@novell.com"]) // for  
an MVEditorfield.setValues(["W","B"], ["White","Black"],true); // for a  
StaticList
```

フィールド

フィールドメソッドの操作に使用します。この項では、次のメソッドについて説明します。

- ◆ 288 ページの 「`activate ()`」

- ◆ 288 ページの 「無効 ()」
- ◆ 288 ページの 「有効 ()」
- ◆ 288 ページの 「fireEvent ()」
- ◆ 288 ページの 「focus ()」
- ◆ 289 ページの 「getLabel ()」
- ◆ 289 ページの 「getName ()」
- ◆ 289 ページの 「getValue ()」
- ◆ 289 ページの 「hide ()」
- ◆ 289 ページの 「getValues ()」
- ◆ 289 ページの 「show ()」
- ◆ 289 ページの 「select ()」
- ◆ 290 ページの 「setRequired ()」
- ◆ 290 ページの 「setValues (fieldname)」
- ◆ 290 ページの 「validate ()」

```
activate ()
field.activate(value[s])
```

このメソッドは、`field.focus ()` と `field.select ()` の組み合わせです。

```
無効 ()
field.disable()
```

フィールドを無効にします。

注: 無効になったフィールドも、ワークフローエンジンにデータを返し続けます。無効になったフィールドのコンテンツは、フォーム送信時またはフィールドの呼び出し時に検証されます。

```
有効 ()
field.enable()
```

フィールドを有効にします。

```
fireEvent ()
field.fireEvent("eventname")
```

カスタムイベントを起動します。起動されたカスタムイベントの名前を渡します。起動されたイベントの値を取得するには、`form.getValues (event.getOrigin ())` を使用します。

```
focus ()
field.focus(value[s])
```

値パラメータが渡されない場合は、その下のテキストフィールドにフォーカスを設定します。リストベースまたは選択ベースのコントロールでは、選択された選択肢または選択がされなかった場合は最初の選択肢にフォーカスを設定します。値パラメータにが渡され、フィールドがリストベースまたは選択ベースの場合、このメソッドは値パラメータに対応

する選択肢にフォーカスを設定します。値が配列の場合は、最初の値のみが、フォーカスが設定されるチェックボックスまたはラジオボタンの決定に使用されます。フィールドが無効または非表示になっている場合、このメソッドは無効です。

getLabel ()

```
field.getLabel()
```

フィールドのラベルを取得します。ラベルが見つからない場合、フィールドの名前を返します。

getName ()

```
field.getName()
```

フィールドの名前を取得します。

getValue ()

```
field.getValue()
```

フィールドの最初の値を返します。フィールドのデータタイプに関係なく、常に文字列タイプを返します。フィールドに値が入っていない場合、テキスト入力フィールド (Text、TextArea、DatePicker、DNLookup など) の場合、空の文字列を返します。選択ベースのコントロール (StaticList、ラジオボタン、チェックボックスなど) の場合、「未定義」というメッセージを返します。DN タイプコントロールの場合、常に DN を返し、表示式は返しません。

hide ()

```
field.hide()
```

このフィールドを非表示にします。

getValues ()

```
form.getValues()
```

要求された値を含む文字列配列を返します。値が見つからない場合、配列は空（サイズ = 0）になります。DN タイプコントロールの場合、常に DN を返し、表示式は返しません。

show ()

```
field.show()
```

このフィールドを表示します。

select ()

```
field.select(value[s])
```

値パラメータが渡されない場合は、その下のテキストフィールドにフォーカスを設定します。リストベースまたは選択ベースのコントロールでは、選択された選択肢または選択がされなかった場合は最初の選択肢にフォーカスを設定します。値パラメータによって渡されるフィールドがリストベースまたは選択ベースの場合、この値パラメータに対応する選択肢にフォーカスを設定します。値パラメータが配列の場合は、最初の値のみが、フォーカスが設定されるチェックボックスまたはラジオボタンの決定に使用されます。フィールドが無効または非表示になっている場合、このメソッドは無効です。

```
setRequired ()  
field.setRequired(is-required)
```

フィールドを、`isRequired` が `True` の場合は必須、それ以外の場合はオプションに設定します。必須フィールドが空の場合、フォームの送信がブロックされます。

```
setValues (fieldname)  
field.setValues(data-values, display-values, KeepOldValues)
```

値を設定します。複数値をサポートします。このメソッドでは、リストベースのコントロール (`StaticList`、`MVCheckbox`、`PickList` など) のエントリを変更できます。デフォルトでは、`KeepOldValues` パラメータが `True` でない限り、既存の値は削除されます。リストベースでないコントロールの場合、`display-values` パラメータは無視されます。

フィールドの初期値を設定または変更したい場合は、`onload` イベントで実行します。

注: このメソッドは、フィールドの `onchange` イベントをトリガします。

例:

```
field.setValues("cn=jdoe,ou=users,ou=mysample,o=novell"); // for a  
DNLookupfield.setValues(["jdoe@novell.com", "test@novell.com"]); // for  
an MVEditorfield.setValues(["W", "B"], ["White", "Black"], true); // for a  
StaticList
```

```
validate ()  
field.validate()
```

フィールドのブラウザ検証をトリガします。ユーザが別のフィールドに移動した直後に、このフィールドに入力されたデータを検証するには、このメソッドを `onchange` イベントで呼び出します。検証エラーが検出された場合は `True`、それ以外の場合は `False` を返します。

イベント

イベントの操作に使用します。この項では、次のメソッドについて説明します。

- ◆ [290 ページの 「getEventName \(\)」](#)
- ◆ [290 ページの 「getOrigin \(\)」](#)
- ◆ [291 ページの 「getValue \(\)」](#)
- ◆ [291 ページの 「getValues \(\)」](#)

```
getEventName ()  
event.getEventName()
```

イベントの名前を返します。

```
getOrigin ()  
event.getOrigin()
```

イベントがトリガされたフィールドの名前を返します。

```
getValue ()  
event.getValue()
```

イベントの最初の値を含む文字列を返します。

このメソッドの戻り値は使用しないでください。イベントがトリガされた後、ユーザがフィールド内のデータを変更している可能性があるからです。代わりに、form.getValue メソッドの戻り値を使用してください。たとえば、form.getValue (event.GetOrigin ()) があります。これにより、フィールドの現在の値を取得できます。ECMA 式ビルダの選択リストから event.getValue () を選択した場合、form.getValue (event.GetOrigin ()) が挿入されます。

```
getValues ()  
event.getValues()
```

イベント内のすべての値を含む文字列配列を返します。

このメソッドの戻り値は使用しないでください。イベントがトリガされた後、ユーザがフィールド内のデータを変更している可能性があるからです。代わりに、form.getValues メソッドの戻り値を使用してください。たとえば、form.getValue (event.GetOrigin ()) があります。これにより、フィールドの現在の値を取得できます。ECMA 式ビルダの選択リストから event.getValue () を選択した場合、form.getValue (event.GetOrigin ()) が挿入されます。

リスト

リストの操作に使用します。

```
globalList (fieldname, key, locale)  
IDVault.globalList("fieldname", "key", "locale")
```

グローバルリストのキーを識別して、ディレクトリ抽象化レイヤからグローバルリストを取得します。フィールド名を指定した場合、クエリの結果によってフィールドのコンテンツが更新されます。結果をフィールドに格納しないでリストを取得するには、fieldname パラメータに null 値を指定します。ロケールはオプションです。ロケールを指定しなかった場合、HTTP 要求のロケールが使用されます。

例：

```
IDVault.globalList("dallist", "departments", "en");
```

クエリ

クエリの操作に使用します。

```
globalQuery (fieldname, key, param)  
globalQuery(fieldname, key, param)
```

事前定義されたディレクトリ抽象化レイヤクエリキーを実行します ([75 ページの「クエリの一般プロパティ」](#) を参照)。フィールド名を指定した場合、クエリの結果によってフィールドのコンテンツが更新されます。結果をフィールドに格納しないでリストを取得するには、fieldname パラメータに null 値を指定します。param パラメータはクエリの入力に使用します。パラメータのフォームは {parname1:value,parname2:value} です。value は

単一の値または配列です。結果リストの最初のカラム（常に DN）はデータ値に使用され、2番目のカラムは表示ラベルに使用されます。

例：

```
IDVault.globalQuery("canchangepwd", "getsites"); // query without a parameter
IDVault.globalQuery("building", "getbuildings",
{site:form.getValue("site")}); // query with one parameter
IDVault.globalQuery("room", "getrooms",
{site:form.getValue("site"), building:form.getValue("building")}); // query with two parameters
```

コンテナ

コンテナの操作に使用します。

containers (fieldname, rootdn, Search scope, Show DN)

```
IDVault.containers("test", rootdn, SearchScope, ShowDN)
```

「サブツリー」と等しい、または同じレベルのスコープを持つコンテナのリストを取得します。このメソッドは、2つのエントリを持つ配列を返します。最初の配列は結果の DN、2番目のエントリは表示ラベル付きの配列です。

表 10-11 コンテナパラメータ

パラメータ	説明
fieldname	フィールド名を指定した場合、クエリの結果によってフィールドのコンテンツが更新されます。結果をフィールドに格納しないでリストを取得するには、 fieldname パラメータに null 値を指定します。
rootdn	rootdn パラメータが空の場合、デフォルトエンティティのルートコンテナが使用されます。
scope	scope パラメータが空の場合、 one-level が使用されます。有効なスコープの指定は、「o」(one-level) と 's' (subtree) です。
showdn	showDN パラメータの値が true の場合、表示ラベルに完全な DN が表示されます。それ以外の場合は、ネーミング部 (ou、dc など) が表示されます。

例：

```
IDVault.containers("assetProp2", null, "o", true); // set the entries in a StaticList to all containers directly under the root DN of the default entity
```

10.3.2 DOM メソッド

この項では、ECMA 式ビルダで使用できるすべての DOM 関連メソッドとプロパティを一覧表示します。これには、W3C 標準に定義されている DOM-1 や DOM-2 拡張だけでなく、Designer 固有の ECMA Script 拡張も含まれています。拡張メソッドであることは、テキスト内に明記されています。DOM メソッドは、統合アクティビティで式を操作しているとき、ECMA 式ビルダに表示されます。

この項では、次のトピックについて説明します。

- ◆ 293 ページの 「ノード」
- ◆ 298 ページの 「ドキュメント」
- ◆ 303 ページの 「要素」
- ◆ 309 ページの 「属性」
- ◆ 310 ページの 「CharacterData」
- ◆ 311 ページの 「ノードリスト」
- ◆ 313 ページの 「NamedNodeMap」
- ◆ 315 ページの 「テキスト」
- ◆ 315 ページの 「DocumentType」
- ◆ 316 ページの 「DOMImplementation」
- ◆ 317 ページの 「表記」
- ◆ 317 ページの 「エンティティ」
- ◆ 318 ページの 「ProcessingInstruction」

ノード

ノードの操作に使用します。この項では、次のトピックについて説明します。

- ◆ 294 ページの 「属性」
- ◆ 294 ページの 「childNodes」
- ◆ 294 ページの 「firstChild」
- ◆ 294 ページの 「lastChild」
- ◆ 294 ページの 「nextSibling」
- ◆ 294 ページの 「nodeName」
- ◆ 294 ページの 「nodeType」
- ◆ 295 ページの 「nodeValue」
- ◆ 295 ページの 「ownerDocument」
- ◆ 295 ページの 「parentNode」
- ◆ 295 ページの 「previousSibling」
- ◆ 295 ページの 「XML」
- ◆ 295 ページの 「appendChild (newChild)」
- ◆ 295 ページの 「cloneNode (deep)」
- ◆ 295 ページの 「createXPath (XPathType asPattern)」
- ◆ 295 ページの 「hasChildNodes ()」
- ◆ 296 ページの 「insertBefore (newChild, refChild)」
- ◆ 296 ページの 「removeChild (oldChild)」
- ◆ 296 ページの 「replaceChild (newChild, oldChild)」
- ◆ 296 ページの 「getXML ()」
- ◆ 296 ページの 「ownerDocument」

- ◆ 296 ページの 「namespaceURI」
- ◆ 296 ページの 「prefix」
- ◆ 296 ページの 「localName」
- ◆ 297 ページの 「normalize ()」
- ◆ 297 ページの 「hasAttributes ()」
- ◆ 297 ページの 「isSupported (feature, version)」

属性

W3C DOM Level 1 Node プロパティ。このプロパティは、Node の属性の NamedNodeMap オブジェクトを返します。

childNodes

W3C DOM Level 1 Node プロパティ。このプロパティは、Node の直下の子で構成される NodeList オブジェクトを返します。

firstChild

W3C DOM Level 1 Node プロパティ。このプロパティは、Node オブジェクトの最初の子ノードを返します。

lastChild

W3C DOM Level 1 Node プロパティ。このプロパティは、Node オブジェクトの最後の子ノードを返します。

nextSibling

W3C DOM Level 1 Node プロパティ。このプロパティは、Node オブジェクトの次の兄弟ノードを返します。

nodeName

W3C DOM Level 1 Node プロパティ。このプロパティは、ノード名を String オブジェクトとして返します。

nodeType

W3C DOM Level 1 Node プロパティ。このプロパティは、ノードタイプを短い形式で返します。次のいずれかの値になります。

- 1 = Element
- 2 = Attribute
- 3 = Text
- 4 = CDATASection
- 5 = EntityReference
- 6 = Entity
- 7 = ProcessingInstruction
- 8 = Comment
- 9 = Document

10 = DocumentType
11 = DocumentFragment
12 = Notation

nodeValue

W3C DOM Level 1 Node プロパティ。このプロパティは、ノードテキストデータを String として返します。

ownerDocument

W3C DOM Level 1 Node プロパティ。このプロパティは、Document オブジェクトを返します。

parentNode

W3C DOM Level 1 Node プロパティ。このプロパティは、Node オブジェクトの親ノードオブジェクトを返します。

previousSibling

W3C DOM Level 1 Node プロパティ。このプロパティは、Node オブジェクトの以前の兄弟ノードを返します。

XML

Designer 拡張プロパティ。このプロパティは、DOM を表す文字列を返します。コンポーネントをデバッグするためのログアクションで役立ちます (Input.XML など)。

appendChild (newChild)

Node appendChild(newChild)

W3C DOM Level 1 Node メソッド。このメソッドは、Node の最後の子としてノードを追加します。newChild パラメータは Node タイプです。

cloneNode (deep)

Node cloneNode(deep)

W3C DOM Level 1 Node メソッド。このメソッドは、接続されていない Node オブジェクトを作成します。deep パラメータはブール型です。

createXPath (XPathType asPattern)

Object createXPath(XPathType asPattern)

ECMAScript 拡張メソッド。XPath パターンを作成します。XPath Type asPattern パラメータは、省略形式の XPath 表記と明示的な序数だけをサポートします。XPath 関数はサポートしません。

hasChildNodes ()

boolean hasChildNodes()

W3C DOM Level 1 Node メソッド。このメソッドは、ノードが子を持っているかどうかを示すブール値を返します。

insertBefore (newChild, refChild)

```
Node insertBefore(newChild, refChild)
```

W3C DOM Level 1 Node メソッド。このメソッドは、親ノードの refChild ノードの前に Node オブジェクトを挿入します。newChild パラメータは Node タイプです。refChild パラメータは Node タイプです。

removeChild (oldChild)

```
Node removeChild(oldChild)
```

W3C DOM Level 1 Node メソッド。このメソッドは、親ノードからノードを削除し、接続されていないノードを返します。oldChild パラメータは Node タイプです。

replaceChild (newChild, oldChild)

```
Node replaceChild(newChild, oldChild)
```

W3C DOM Level 1 Node メソッド。このメソッドは、一方のノードをもう一方のノードで置き換えます。newChild パラメータは Node タイプです。oldChild パラメータは Node タイプです。

getXML ()

```
String getXML()
```

ECMAScript 拡張メソッド。このプロパティは、DOM を表す文字列を返します。コンポーネントをデバッグするためのログアクションで役立ちます。例：

```
Input.XPath("root/child").getXML()
```

ownerDocument

W3C DOM Level 2 の変更 Node プロパティこのノードに関連付けられた Document オブジェクトを返します。この Document オブジェクトは、新しいノードを作成するときに使用されます。例：

```
someNodeObject.ownerDocument
```

namespaceURI

W3C DOM Level 2 Node プロパティ。このノードのネームスペース URI を返します。ネームスペース URI が指定されていない場合は null を返します。例：

```
someNodeObject.namespaceURI
```

prefix

W3C DOM Level 2 Node プロパティ。このノードのネームスペースプレフィックスを返します。ネームスペースプレフィックスが指定されていない場合は null を返します。例：

```
someNodeObject.prefix
```

localName

W3C DOM Level 2 Node プロパティ。このノードの修飾名のローカル部分を返します。例

```
:
```

```
someNodeObject.localName
```

```
normalize ()  
void normalize()
```

W3C DOM Level 2 の変更 Node メソッド。属性ノードを含む、この Node の下にあるサブツリーの全階層に含まれるすべての Text ノードを「標準」の形式にします。標準の形式では、Text ノードは構造（要素、コメント、処理命令、CDATA セクション、エンティティ参照など）によってのみ分離されます。つまり、隣接する Text ノードや空の Text ノードはありません。

```
hasAttributes ()  
boolean hasAttributes()
```

W3C DOM Level 2 Node メソッド。ノードに属性がある場合は True、それ以外の場合は False を返します。例：

```
Temp.XPath("A/B/C").item(0).hasAttributes()
```

```
isSupported (feature, version)  
boolean isSupported(feature, version)
```

W3C DOM Level 2 Node メソッド。指定された機能がこのノード上でサポートされている場合は、true を返します。それ以外の場合は False を返します。

表 10-12 *IsSupported* メソッドのパラメータ

パラメータ	機能
feature	Core XML HTML Views Stylesheets CSS CSS2 Events UIEvents MouseEvents MutationEvents HTMLEvents Range Transversal
version	テストする機能のバージョン番号を指定します。Level 2、バージョン 1 では、文字列「2.0」です。バージョンが指定されていない場合、どのバージョンの機能がサポートされていても、メソッドは True を返します。

例：

```
aNodeObject.isSupported("Core", "2.0")
```

ドキュメント

ドキュメントの操作に使用します。この項では、次のトピックについて説明します。

- ◆ 298 ページの 「doctype」
- ◆ 298 ページの 「documentElement」
- ◆ 299 ページの 「implementation」
- ◆ 299 ページの 「text」
- ◆ 299 ページの 「createAttribute (name)」
- ◆ 299 ページの 「createCDATASection (data)」
- ◆ 299 ページの 「createComment (data)」
- ◆ 299 ページの 「createDocumentFragment ()」
- ◆ 299 ページの 「createElement (tagName)」
- ◆ 299 ページの 「createEntityReference (name)」
- ◆ 300 ページの 「createProcessingInstruction (target,data)」
- ◆ 300 ページの 「createTextNode (data)」
- ◆ 300 ページの 「getElementsByTagName (tagName)」
- ◆ 300 ページの 「reset ()」
- ◆ 300 ページの 「setDTD (Node RootElementName, Object PublicName, Object URL)」
- ◆ 300 ページの 「setValue (Object aValue)」
- ◆ 300 ページの 「toString ()」
- ◆ 300 ページの 「XPath (String asPattern)」
- ◆ 301 ページの 「importNode (sourceNode, deep)」
- ◆ 301 ページの 「createElementNS (namespaceURI, qualifiedName)」
- ◆ 301 ページの 「createAttributeNS (namespaceURI, qualifiedName)」
- ◆ 302 ページの 「getElementsByTagNameNS (namespaceURI, localName)」
- ◆ 302 ページの 「getElementById (elementId)」
- ◆ 302 ページの 「setSkipNameSpaces (abFlag)」
- ◆ 302 ページの 「setEncoding (encoding)」

doctype

W3C DOM Level 1 Document プロパティ。このプロパティは、ドキュメントの DTD を反映する DocumentType オブジェクトを返します。Document には、Node のすべてのプロパティとメソッドがあります。

documentElement

W3C DOM Level 1 Document プロパティ。このプロパティは、Element オブジェクト（ルート要素）を返します。Document には、Node のすべてのプロパティとメソッドがあります。

implementation

W3C DOM Level 1 Document プロパティ。このプロパティは、DOMImplementation オブジェクトを返します。Document には、Node のすべてのプロパティとメソッドがあります。

text

Designer 拡張プロパティ。このプロパティは、その下にあるすべてのテキストノード（コンテンツ）の連結文字列を返します。

createAttribute (name)

```
Attr createAttribute (name)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない Attr オブジェクトを返します。name パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createCDATASection (data)

```
CDataSection createCDATASection (data)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない CDataSection オブジェクトを返します。data パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createComment (data)

```
Comment createComment (data)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない Comment オブジェクトを返します。data パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createDocumentFragment ()

```
DocumentFragment createDocumentFragment ()
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない DocumentFragment を返します。Document には、Node のすべてのプロパティとメソッドがあります。

createElement (tagName)

```
Element createElement (tagName)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない Element を作成します。tagName パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createEntityReference (name)

```
EntityReference createEntityReference (name)
```

W3C DOM Level 1 Document メソッド。接続されていない EntityReference を作成します。name パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createProcessingInstruction (target,data)

```
ProcessingInstruction createProcessingInstruction(target,data)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない ProcessingInstruction オブジェクトを返します。target パラメータと data パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

createTextNode (data)

```
Text createTextNode(data)
```

W3C DOM Level 1 Document メソッド。このメソッドは、接続されていない Text オブジェクトを作成します。data パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

getElementsByTagName (tagName)

```
NodeList getElementsByTagName(tagName)
```

W3C DOM Level 1 Document メソッド。このメソッドは、タグ変数要素ノードから構成される NodeList オブジェクトを返します。tagName パラメータは String タイプです。Document には、Node のすべてのプロパティとメソッドがあります。

reset ()

```
void reset()
```

W3C DOM Level 1 Document メソッド。ドキュメントをクリアします。

setDTD (Node RootElementName, Object PublicName, Object URL)

```
setDTD(Node RootElementName, Object PublicName, Object URL)
```

ECMAScript 拡張メソッド。ドキュメントの DTD ファイルを設定します。

setValue (Object aValue)

```
setValue(Object aValue)
```

ECMAScript 拡張メソッド。渡されたオブジェクトからドキュメントの値を設定します。渡されたオブジェクトが別のドキュメントである場合、子ノード（要素と属性）がコピーされます。渡されたオブジェクトがテキストである場合、そのテキストが DOM を作成するためには解析されます。

toString ()

```
String toString()
```

ECMAScript 拡張メソッド。DOM ドキュメントを XML 形式の文字列に変換します。

例：

```
Input.XPath("root/child").item(0).toString()
```

XPath (String asPattern)

```
NodeList XPath(XPathType asPattern)
```

ECMAScript 拡張メソッド。XPathTypes のタイプには、NodeList、String、Number、または Boolean があります。通常、XPath パターンと一致する Nodelist を返すために使用されます。角括弧を使用して、リストから特定のノードを選択します。す。たとえば、

`Input.XPath ("INVOICE/LINEITEM[1]")`、`Input.XPath ("INVOICE/LINEITEM[last ()]")` のようにします。属性でノードを選択する場合は、`@` 記号を使用します。たとえば、`Input.XPath ("INVOICE/LINEITEM[@myattr]")` のようにします。属性値で選択する場合は、`Input.XPath ("INVOICE/LINEITEM[@myattr='abc']")` のようにします。

```
importNode (sourceNode, deep)
Node importNode (sourceNode, deep)
```

W3C DOM Level 2 Document メソッド。任意のドキュメントから現在のドキュメントにノードをインポートします。このメソッドは、`sourceNode` の新しいコピーを作成します。`sourceNode` は変更されません。Document には、Node のすべてのプロパティとメソッドがあります。

表 10-13 `ImportNode` メソッドのパラメータ

パラメータ	説明
<code>sourceNode</code>	インポートするノード。
<code>deep</code>	ブール値。 <code>True</code> の場合、指定されたノードの下のサブツリーを再帰的にインポートします。 <code>False</code> の場合、指定されたノードだけをインポートします。

例：

```
Temp.importNode (Input.XPath ("A/B[2]"), false)
```

```
createElementNS (namespaceURI, qualifiedName)
Element createElementNS (namespaceURI, qualifiedName)
```

W3C DOM Level 2 Document メソッド。指定された `qualifiedName` と `namespaceURI` を持つ Element を作成します。Document には、Node のすべてのプロパティとメソッドがあります。

表 10-14 `createElementNS` メソッドのパラメータ

パラメータ	説明
<code>namespaceURI</code>	要素用として作成するネームスペース URI を表す文字列。
<code>qualifiedName</code>	要素用として作成する名前を表す文字列。 <code>qualifiedName = namespaceprefix + : + localName</code>

例：

```
Temp.createElementNS ("someURI", "nsprefix:PRICE")
```

```
createAttributeNS (namespaceURI, qualifiedName)
Attr createAttributeNS (namespaceURI, qualifiedName)
```

W3C DOM Level 2 Document メソッド。指定された `qualifiedName` と `namespaceURI` を持つ Attribute を作成します。Document には、Node のすべてのプロパティとメソッドがあります。

表 10-15 *createAttributeNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	属性用として作成するネームスペース URI を表す文字列。
qualifiedName	属性用として作成する名前を表す文字列。qualifiedName = namespaceprefix + : + localName

例：

```
Temp.createAttributeNS ("someURI", "nsprefix:PRICE")
```

getElementsByTagNameNS (namespaceURI, localName)

```
NodeList getElementsByTagNameNS (namespaceURI, localName)
```

W3C DOM Level 2 Document メソッド。指定された localName と namespaceURI を持つすべての Element の NodeList を、Document ツリーの先行順走査の順番に返します。Document には、Node のすべてのプロパティとメソッドがあります。

表 10-16 *getElementsByTagNameNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	一致する要素の文字列。特殊な値「*」は、すべてのネームスペースに一致します。
qualifiedName	一致する要素の文字列。特殊な値「*」は、すべてのローカル名に一致します。

例：

```
Temp.getElementsByTagNameNS ("someURI", "someName")
```

getElementById (elementId)

```
Element getElementById (elementId)
```

W3C DOM Level 2 Document メソッド。elementId によって指定された ID の Element を返します。該当する要素が存在しない場合は null を返します。指定された ID を持つ要素が複数存在する場合の動作は未定義です。Document には、Node のすべてのプロパティとメソッドがあります。

例：

```
Temp.getElementById ("someId")
```

setSkipNameSpaces (abFlag)

```
void setSkipNameSpaces (boolean flag)
```

このメソッドを使用すると、ネームスペースの使用をオフにし、プレフィックスを持たないノードに一致させ、つまりワイルドカード一致と同様の動作を行うことができます。

setEncoding (encoding)

```
void setEncoding (String encoding)
```

このメソッドは、ドキュメントの文字セットのエンコードを設定します。

要素

要素の操作に使用します。この項では、次のトピックについて説明します。

- ◆ 303 ページの 「tagName」
- ◆ 304 ページの 「text」
- ◆ 304 ページの 「booleanValue ()」
- ◆ 304 ページの 「countOfElement (String propertyName)」
- ◆ 304 ページの 「doubleValue ()」
- ◆ 304 ページの 「exists (String propertyName)」
- ◆ 304 ページの 「getAttribute (name)」
- ◆ 304 ページの 「getAttributeNode (name)」
- ◆ 304 ページの 「getElementsByTagName (name)」
- ◆ 304 ページの 「getIndex ()」
- ◆ 304 ページの 「getParent ()」
- ◆ 305 ページの 「normalize ()」
- ◆ 305 ページの 「removeAttribute (name)」
- ◆ 305 ページの 「removeAttributeNode (oldAttr)」
- ◆ 305 ページの 「setAttribute (name,value)」
- ◆ 305 ページの 「setAttributeNode (newAttr)」
- ◆ 305 ページの 「setIndex (int aiIndex)」
- ◆ 305 ページの 「setText (String asText)」
- ◆ 305 ページの 「setValue (Object aValue)」
- ◆ 306 ページの 「toNumber ()」
- ◆ 306 ページの 「toString ()」
- ◆ 306 ページの 「XPath (XPathType asPattern)」
- ◆ 306 ページの 「getAttributeNS (namespaceURI, localName)」
- ◆ 306 ページの 「setAttributeNS (namespaceURI, qualifiedName, value)」
- ◆ 307 ページの 「removeAttributeNS (namespaceURI, localName)」
- ◆ 307 ページの 「getAttributeNodeNS (namespaceURI, localName)」
- ◆ 308 ページの 「setAttributeNodeNS (newAttr)」
- ◆ 308 ページの 「getElementsByTagNameNS (namespaceURI, localName)」
- ◆ 308 ページの 「hasAttribute (name)」
- ◆ 308 ページの 「hasAttributeNS (namespaceURI, localName)」

tagName

W3C DOM Level 1 Element プロパティ。このプロパティは、要素名を含む String オブジェクトを返します。Element には、Node のすべてのプロパティとメソッドがあります。

text

Designer 拡張プロパティ。このプロパティは、その下にあるすべてのテキストノードの連結テキストを返します。

booleanValue ()

```
boolean booleanValue()
```

ECMAScript 拡張メソッド。可能な場合、このオブジェクトのブール値 (True または False) を返します。

countOfElement (String propertyName)

```
Number countOfElement(String propertyName)
```

ECMAScript 拡張メソッド。名前付きの子の数を返します。

doubleValue ()

```
double doubleValue()
```

ECMAScript 拡張メソッド。可能な場合、このオブジェクトの double 値を返します。

exists (String propertyName)

```
Boolean exists(String propertyName)
```

ECMAScript 拡張メソッド。名前付きの子の存在をチェックします。

getAttribute (name)

```
String getAttribute(name)
```

W3C DOM Level 1 Element メソッド。このメソッドは、属性値から構成される String を返します。name パラメータは String タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

getAttributeNode (name)

```
Attr getAttributeNode(name)
```

W3C DOM Level 1 Element メソッド。このメソッドは、Attr を返します。name パラメータは String タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

getElementsByTagName (name)

```
NodeList getElementsByTagName(name)
```

W3C DOM Level 1 Element メソッド。指定された名前を持つすべての要素の NodeList を返します。name パラメータは String タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

getIndex ()

```
int getIndex()
```

ECMAScript 拡張メソッド。現在のインデックスを返します。

getParent ()

```
Node getParent()
```

ECMAScript 拡張メソッド。親要素を返します。

```
normalize ()  
void normalize()
```

W3C DOM Level 1 Element メソッド。このメソッドは void を返します。Element には、Node のすべてのプロパティとメソッドがあります。

```
removeAttribute (name)  
void removeAttribute (name)
```

W3C DOM Level 1 Element メソッド。このメソッドは要素から属性を削除します。name パラメータは String タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

```
removeAttributeNode (oldAttr)  
Attr removeAttributeNode (oldAttr)
```

W3C DOM Level 1 Element メソッド。このメソッドは、要素から属性を削除し、接続されていない Attr を返します。oldAttr パラメータは Attr タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

```
setAttribute (name,value)  
void setAttribute (name, value)
```

W3C DOM Level 1 Element メソッド。このメソッドは、要素の属性ノードの値を設定します。name パラメータは String タイプです。value パラメータは String タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

```
setAttributeNode (newAttr)  
Attr setAttributeNode (newAttr)
```

W3C DOM Level 1 Element メソッド。このメソッドは要素に属性ノードを接続します。newAttr パラメータは Attr タイプです。Element には、Node のすべてのプロパティとメソッドがあります。

```
setIndex (int aiIndex)  
setIndex (int aiIndex)
```

ECMAScript 拡張メソッド。この要素の iterator インデックス値を設定します。

```
setText (String asText)  
setText (String asText)
```

ECMAScript 拡張メソッド。この要素に関連付けられているテキストノードを設定します。

```
setValue (Object aValue)  
setValue (Object aValue)
```

ECMAScript 拡張メソッド。渡されたオブジェクトから要素の値を設定します。渡されたオブジェクトが別の要素である場合、子ノード（要素と属性）もコピーされます。

```
toNumber ()  
Number toNumber()
```

ECMAScript 拡張メソッド。テキストノードを取得し、数値に変換します。

```
toString ()  
String toString()
```

ECMAScript 拡張メソッド。この要素に関連付けられているテキストノードを取得します。

XPath (XPathType asPattern)

```
NodeList XPath(XPathType asPattern)
```

ECMAScript 拡張メソッド。XPathTypes のタイプには、NodeList、String、Number、または Boolean があります。通常、XPath パターンと一致する Nodelist を返すために使用されます。角括弧を使用して、リストから特定のノードを選択します。たとえば、Input.XPath ("INVOICE/LINEITEM[1]")、Input.XPath ("INVOICE/LINEITEM[last ()]") のようにします。属性でノードを選択する場合は、@ 記号を使用します。たとえば、Input.XPath ("INVOICE/LINEITEM[@myattr]") のようにします。属性値で選択する場合は、Input.XPath ("INVOICE/LINEITEM[@myattr='abc']") のようにします。

```
getAttributeNS (namespaceURI, localName)  
string getAttributeNS (namespaceURI, localName)
```

W3C DOM Level 2 Element メソッド。Attr 値を文字列として返します。Element には、Node のすべてのプロパティとメソッドがあります。

表 10-17 *getAttributeNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	ターゲット Attr のネームスペース URI を表す文字列を指定します。
localName	ターゲット Attr のローカル名の文字列を指定します。

例：

```
Temp.XPath ("A/B[0]").getAttributeNS ("someURI", "someAttr")
```

setAttributeNS (namespaceURI, qualifiedName, value)

```
void setAttributeNS (namespaceURI, qualifiedName, value)
```

W3C DOM Level 2 Element メソッド。新しい属性を追加します。要素内に同じネームスペース URI とローカル名を持つ属性が存在している場合、プレフィックスは qualifiedName パラメータのプレフィックス部に変更され、値は値パラメータに変更されます。Element には、Node のすべてのプロパティとメソッドがあります。

表 10-18 *setAttributeNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	作成または変更する属性のネームスペース URI。
qualifiedName	作成または変更する属性の修飾名を指定します。
ヒント : qualifiedName = namespaceprefix + : + localName	
value	設定する値を文字列形式で指定します。

例 :

```
Temp.XPath("A/B[0]").setAttributeNS("someURI", "someAttrName",  
"someAttrvalue")
```

removeAttributeNS (namespaceURI, localName)

```
void removeAttributeNS(namespaceURI, localName)
```

W3C DOM Level 2 Element メソッド。ローカル名とネームスペース URI で属性を削除します。削除された属性にデフォルト値がある場合は、ただちに置き換えられます。置き換える属性は、削除された属性と同じネームスペース URI、ローカル名、および元のプレフィックスを持ちます。Element には、Node のすべてのプロパティとメソッドがあります。

表 10-19 *removeAttributeNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	削除する属性のネームスペース URI を指定します。
localName	削除する属性の名前を指定します。

例 :

```
Temp.XPath("A/B[0]").removeAttributeNS("someURI", "someAttrName")
```

getAttributeNodeNS (namespaceURI, localName)

```
Attr getAttributeNodeNS(namespaceURI, localName)
```

W3C DOM Level 2 Element メソッド。ローカル名とネームスペース URI で属性ノードを取得します。Element には、Node のすべてのプロパティとメソッドがあります。

表 10-20 *getAttributeNodeNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	取得する属性のネームスペース URI を指定します。
localName	取得する属性の名前を指定します。

例 :

```
Temp.XPath("A/B[0]").getAttributeNodeNS("someURI", "someAttr")
```

setAttributeNodeNS (newAttr)

```
Attr setAttributeNodeNS(newAttr)
```

W3C DOM Level 2 Element メソッド。新しい属性を追加します。要素内に同じネームスペース URI とローカル名を持つ属性がすでに存在する場合、新しい属性によって置き換えられます。newAttr 属性によって同じローカル名とネームスペース URI を持つ既存の属性が置き換えられる場合、置き換えられた Attr ノードが返されます。他の場合は null が返されます。newAttr パラメータは新しい属性オブジェクトです。Element には、Node のすべてのプロパティとメソッドがあります。

例：

```
Temp.XPath("A/B[0]").setAttributeNodeNS(newAttr)
```

getElementsByTagNameNS (namespaceURI, localName)

```
NodeList getElementsByTagNameNS(namespaceURI, localName)
```

W3C DOM Level 2 Element メソッド。指定されたローカル名とネームスペース URI を持つすべての子孫 Element の NodeList を、この Element ツリーの先行順走査の順番に返します。Element には、Node のすべてのプロパティとメソッドがあります。

表 10-21 *getElementsByTagNameNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	一致する要素のネームスペース URI を指定します。特殊な値「*」は、すべてのネームスペースに一致します。
localName	一致する要素のローカル名を指定します。特殊な値「*」は、すべてのローカル名に一致します。

例：

```
Temp.XPath("A/B[0]").getElementsByTagNameNS("someURI", "someName")
```

hasAttribute (name)

```
boolean hasAttribute()
```

W3C DOM Level 2 Element メソッド。指定された名前の属性がこの要素に対して指定されているか、デフォルト値を持つ場合は、True を返します。それ以外の場合は False を返します。name パラメータは、検索する属性名を指定する文字列です。Element には、Node のすべてのプロパティとメソッドがあります。

例：

```
Temp.XPath("A/B[0]").hasAttribute("someName")
```

hasAttributeNS (namespaceURI, localName)

```
boolean hasAttributeNS(namespaceURI, localName)
```

W3C DOM Level 2 Element メソッド。指定されたローカル名とネームスペース URI の属性がこの要素に対して指定されているか、デフォルト値を持つ場合は、True を返します。

それ以外の場合は `False` を返します。`Element` には、`Node` のすべてのプロパティとメソッドがあります。

表 10-22 `hasAttributeNS` メソッドのパラメータ

パラメータ	説明
<code>namespaceURI</code>	検索する属性のネームスペース URI を指定します。
<code>localName</code>	検索する属性のローカル名を指定します。

例：

```
Temp.XPath("A/B[0]").hasAttributeNS("someURI", "someName")
```

属性

属性の操作に使用します。この項では、次のトピックについて説明します。

- ◆ 309 ページの 「name」
- ◆ 309 ページの 「specified」
- ◆ 309 ページの 「text」
- ◆ 309 ページの 「value」
- ◆ 309 ページの 「setValue (Object aValue)」
- ◆ 310 ページの 「toString ()」
- ◆ 310 ページの 「ownerElement」

name

W3C DOM Level 1 Attr プロパティ。このプロパティは、属性のタグ名を示す String オブジェクトを返します。属性には、`Node` のすべてのプロパティおよびメソッドがあります。

specified

W3C DOM Level 1 Attr プロパティ。このプロパティはブール値を返します。属性には、`Node` のすべてのプロパティおよびメソッドがあります。

text

Designer 拡張プロパティ。このプロパティは、属性のテキスト値を返します。

value

W3C DOM Level 1 Attr プロパティ。このプロパティは、属性のテキスト値を現す String オブジェクトを返します。属性には、`Node` のすべてのプロパティおよびメソッドがあります。

setValue (Object aValue)

```
setValue(Object aValue)
```

Designer 拡張メソッド。渡されたオブジェクトから属性の値を設定します。

```
toString ()  
String toString ()
```

ECMAScript 拡張メソッド。この属性に関連付けられたテキストノードを取得します。

ownerElement

W3C DOM Level 2 Attr プロパティこの属性の接続されている Element ノードを返します。この属性が使用されていない場合は null を返します。Attr には、Node のすべてのプロパティおよびメソッドがあります。

例：

```
attributeObject.ownerElement
```

CharacterData

文字データの操作に使用します。この項では、次のトピックについて説明します。

- ◆ [310 ページの 「data」](#)
- ◆ [310 ページの 「length」](#)
- ◆ [310 ページの 「appendData \(arg\)」](#)
- ◆ [310 ページの 「insertData \(offset, arg\)」](#)
- ◆ [311 ページの 「deleteData \(offset, count\)」](#)
- ◆ [311 ページの 「replaceData \(offset, count, arg\)」](#)
- ◆ [311 ページの 「substringData \(offset, count\)」](#)

data

W3C DOM Level 1 CharacterData プロパティ。このプロパティは String タイプで、CharacterData オブジェクトのコンテンツを表します。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

length

W3C DOM Level 1 CharacterData プロパティ。このプロパティは、CharacterData オブジェクトの長さを表します。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

appendData (arg)

```
void appendData(arg)
```

W3C DOM Level 1 CharacterData メソッド。このメソッドは、CharacterData オブジェクトにテキストを追加します。arg パラメータは String タイプです。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

insertData (offset, arg)

```
void insertData(offset, arg)
```

W3C DOM Level 1 CharacterData メソッド。このメソッドは、CharacterData オブジェクトにテキストを挿入します。offset パラメータは unsigned long タイプです。arg パラメータは String タイプです。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

```
deleteData (offset, count)
void deleteData(offset, count)
```

W3C DOM Level 1 CharacterData メソッド。このメソッドは、CharacterData オブジェクト内のテキストを削除します。offset パラメータと count パラメータは unsigned long タイプです。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

```
replaceData (offset, count, arg)
void replaceData(offset, count, arg)
```

W3C DOM Level 1 CharacterData メソッド。このメソッドは、CharacterData オブジェクト内のテキストを置き換えます。offset パラメータと count パラメータは unsigned long タイプです。arg パラメータは String タイプです。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

```
substringData (offset, count)
String substringData(offset, count)
```

W3C DOM Level 1 CharacterData メソッド。このメソッドは、CharacterData オブジェクトの部分文字列を返します。offset パラメータと count パラメータは unsigned long タイプです。CharacterData には、Node のすべてのプロパティおよびメソッドがあります。

ノードリスト

ノードリストの操作に使用します。この項では、次のトピックについて説明します。

- ◆ 311 ページの 「length」
- ◆ 311 ページの 「avg ('[NodeList]')」
- ◆ 312 ページの 「count ('[NodeList]')」
- ◆ 312 ページの 「item (index)」
- ◆ 312 ページの 「min ('[NodeList]')」
- ◆ 312 ページの 「max ('[NodeList]')」
- ◆ 312 ページの 「sum ('[NodeList]')」
- ◆ 313 ページの 「where (XPathType asPattern)」
- ◆ 313 ページの 「toNumber ()」

length

W3C DOM Level 1 NodeList プロパティ。このプロパティは、NodeList オブジェクト内のノード数を返します。

```
avg ('[NodeList]')
Number avg(' [NodeList] ')
```

ECMAScript 総拡張メソッド。NodeList の平均値に等しい数を返します。NodeList パラメータは XPath タイプです。パラメータを指定しない場合、現在の NodeList/GroupName が使用されます。関数の引数はシングルクォーテーションで囲み、ネスト化された呼び出しの場合はエスケープする必要があります。

例：

```
Input.XPath("rootElem/childElem").avg()  
  
count ('[NodeList]')  
Number count(' [NodeList] ')
```

ECMAScript 総拡張メソッド。NodeList 内にある、データを持つノード数に等しい数を返します。データがないノードや子要素しか持たないノードはカウントされません。すべてのノードをカウントするには、nodeList オブジェクトで `.length` プロパティを使用します。オプションの NodeList パラメータは XPath タイプです。パラメータを指定しない場合（通常は指定しない）、現在の NodeList/GroupName が使用されます。関数の引数はシングルクォーテーションで囲み、ネスト化された呼び出しの場合はエスケープする必要があります。

例：

```
Input.XPath("rootElem/childElem").count()
```

```
item (index)  
Node item(index)
```

W3C DOM Level 1 NodeList メソッド。このメソッドは、指定された Node を NodeList より返します。index パラメータは `unsigned long` タイプです。Index は 0 から始まります。

```
min ('[NodeList]')  
Number min(' [NodeList] ')
```

ECMAScript 総拡張メソッド。NodeList の最小値に等しい数を返します。NodeList パラメータは XPath タイプです。パラメータを指定しない場合、現在の NodeList/GroupName が使用されます。関数の引数はシングルクォーテーションで囲み、ネスト化された呼び出しの場合はエスケープする必要があります。

例：

```
Input.XPath("rootElem/childElem").min()
```

```
max ('[NodeList]')  
Number max(' [NodeList] ')
```

ECMAScript 総拡張メソッド。NodeList の最大値に等しい数を返します。NodeList パラメータは XPath タイプです。パラメータを指定しない場合、現在の NodeList/GroupName が使用されます。関数の引数はシングルクォーテーションで囲み、ネスト化された呼び出しの場合はエスケープする必要があります。

例：

```
Input.XPath("rootElem/childElem").max()
```

```
sum ('[NodeList]')  
Number sum(' [NodeList] ')
```

ECMAScript 総拡張メソッド。NodeList の値の合計に等しい数を返します。NodeList パラメータは XPath タイプです。パラメータを指定しない場合、現在の NodeList/GroupName が使用されます。関数の引数はシングルクォーテーションで囲み、ネスト化された呼び出しの場合はエスケープする必要があります。

例：

```
Input.XPath("rootElem/childElem").sum()
```

where (XPathType asPattern)

```
NodeList where(String asPattern)
```

ECMAScript 拡張メソッド。XPath パターンに一致するノードの NodeList を取得します。

toNumber ()

```
toNumber()
```

NodeList 内の最初のインスタンスのデータを ECMAScript Number オブジェクトに変換します。データのアルファベット文字または埋め込みスペースは NaN を返します。先頭と末尾のスペースは問題ありません。

例：

```
var myNum = Input.XPath("Invoice/Amount").toNumber()
```

NamedNodeMap

名前付きノードマップの操作に使用します。この項では、次のトピックについて説明します。

- ◆ 313 ページの 「length」
- ◆ 313 ページの 「getNamedItem (name)」
- ◆ 313 ページの 「getNamedItemNS (namespaceURI, localName)」
- ◆ 314 ページの 「item (index)」
- ◆ 314 ページの 「removeNamedItem (name)」
- ◆ 314 ページの 「removeNamedItemNS (namespaceURI, localName)」
- ◆ 314 ページの 「setNamedItem (arg)」
- ◆ 315 ページの 「setNamedItemNS (Node arg)」

length

W3C DOM Level 1 NamedNodeMap プロパティ。このプロパティは、NamedNodeMap 内のノード数を返します。

getNamedItem (name)

```
Node getNamedItem(name)
```

W3C DOM Level 1 NamedNodeMap メソッド。このメソッドは、指定された名前の選択された Node をすべて返します。name パラメータは String タイプです。

getNamedItemNS (namespaceURI, localName)

```
Node getNamedItemNS(namespaceURI, localName)
```

W3C DOM Level 2 NamedNodeMap メソッド。ローカル名とネームスペース URI で指定されたノードを返します。

表 10-23 *NamedNodeMap* メソッドのパラメータ

パラメータ	説明
namespaceURI	取得するノードのネームスペース URI を指定します。
localName	取得するノードのローカル名を指定します。

例：

```
Temp.XPath("A/B").item(0).getAttributes().getNamedItemNS("someURI", "anAttrName")
```

item (index)

```
Node item(index)
```

W3C DOM Level 1 *NamedNodeMap* メソッド。このメソッドは、指定された *Node* を *NamedNodeMap* から返します。index パラメータは *unsigned long* タイプです。インデックスは 0 から始まります。

removeNamedItem (name)

```
Node removeNamedItem(name)
```

W3C DOM Level 1 *NamedNodeMap* メソッド。このメソッドは、*NamedNodeMap* から指定されたノードを削除し、接続されていないノードを返します。name パラメータは *String* タイプです。

removeNamedItemNS (namespaceURI, localName)

```
Node removeNamedItemNS(namespaceURI, localName)
```

W3C DOM Level 2 *NamedNodeMap* メソッド。ネームスペース URI とローカル名で指定されたノードを削除し、返します。

表 10-24 *removeNamedItemNS* メソッドのパラメータ

パラメータ	説明
namespaceURI	削除するノードのネームスペース URI を指定します。
localName	削除するノードのローカル名を指定します。

例：

```
Temp.XPath("A/B").item(0).getAttributes().removeNamedItemNS("someURI", "anAttrName")
```

setNamedItem (arg)

```
Node setNamedItem(arg)
```

W3C DOM Level 1 *NamedNodeMap* メソッド。このメソッドは *Node* を返します。arg パラメータは *Node* タイプです。

setNamedItemNS (Node arg)

```
Node setNamedItemNS (arg)
```

W3C DOM Level 2 NamedNodeMap メソッド。新しい Node によって既存のノードを置き換える場合、置き換えられた Node が返されます。それ以外の場合は null が返されます。

例：

```
var item = Temp.XPath("A/B").item(0);
item.getAttributes().setNamedItemNS(aNodeObject)
```

テキスト

テキストの操作に使用します。

splitText (offset)

```
Text splitText (offset)
```

W3C DOM Level 1 Element メソッド。このメソッドは、オフセットまでのテキストを削除し、削除されたテキストで接続されていないテキストノードを作成します。offset パラメータは unsigned long タイプです。Text には、CharacterData のすべてのプロパティとメソッドがあります。

DocumentType

ドキュメントタイプの操作に使用します。この項では、次のトピックについて説明します。

- ◆ [315 ページの 「name」](#)
- ◆ [315 ページの 「entities」](#)
- ◆ [315 ページの 「internalSubset」](#)
- ◆ [316 ページの 「notations」](#)
- ◆ [316 ページの 「publicId」](#)
- ◆ [316 ページの 「systemId」](#)

name

W3C DOM Level 1 DocumentType プロパティ。このプロパティは、ドキュメントタイプ名を表す String を返します。

entities

W3C DOM Level 1 DocumentType プロパティ。このプロパティは、ドキュメント内に定義されたエンティティの NamedNodeMap を返します。

internalSubset

W3C DOM Level 2 DocumentType プロパティ。このプロパティは、内部サブセットを文字列として表す String を返します。

notations

W3C DOM Level 1 DocumentType プロパティ。このプロパティは、ドキュメント内に定義された表記の NamedNodeMap を返します。

publicId

W3C DOM Level 2 DocumentType プロパティ。このプロパティは、外部サブセットのパブリック識別子を表す String を返します。

systemId

W3C DOM Level 2 の DocumentType プロパティ。このプロパティは、外部サブセットのシステム識別子を表す String を返します。

DOMImplementation

DOM 実装の操作に使用します。この項では、次のトピックについて説明します。

- ◆ 316 ページの 「createDocument (namespaceURI, qualifiedName, doctype)」
- ◆ 316 ページの 「createDocumentType (qualifiedName, publicID, systemID)」
- ◆ 317 ページの 「hasFeature (feature, version)」

createDocument (namespaceURI, qualifiedName, doctype)

```
Document createDocument(namespaceURI, qualifiedName, doctype)
```

W3C DOM Level 2 DOMImplementation メソッド。指定されたタイプの XML Document オブジェクトをそのドキュメント要素で作成します。

表 10-25 DOMImplementation メソッドのパラメータ

パラメータ	説明
namespaceURI	作成するドキュメント要素のネームスペース URI を指定します。
qualifiedName	作成するドキュメント要素の修飾名を指定します。qualifiedName = namespaceprefix + : + localName
doctype	作成するドキュメントのタイプか、null を指定します。

createDocumentType (qualifiedName, publicID, systemID)

```
DocumentType createDocumentType(qualifiedName, publicID, systemID)
```

W3C DOM Level 2 DOMImplementation メソッド。空の DocumentType ノードを作成します。
パラメータ : qualifiedName は、作成するドキュメントタイプの名前の文字列です。
publicID パラメータは、外部サブセットのパブリック識別子です。systemID は、外部サブセットのシステム識別子です。メモ : qualifiedName = namespaceprefix + : + localName

表10-26 `createDocumentType` メソッドのパラメータ

パラメータ	説明
<code>qualifiedName</code>	作成するドキュメント要素の修飾名を指定します。 <code>qualifiedName</code> = <code>namespaceprefix</code> + : + <code>localName</code>
<code>publicID</code>	外部サブセットのパブリック識別子を指定します。
<code>systemID</code>	外部サブセットのシステム識別子を指定します。

`hasFeature (feature, version)`

```
boolean hasFeature (feature, version)
```

W3C DOM Level 1 DOMImplementation メソッド。このメソッドはブール値を返します。`feature` パラメータは String タイプです。`version` パラメータは String タイプです。

表記

表記の操作に使用します。この項では、次のトピックについて説明します。

- ◆ [317 ページの「publicId」](#)
- ◆ [317 ページの「systemId」](#)

`publicId`

W3C DOM Level 2 のプロパティ。このプロパティは、外部サブセットのパブリック識別子を表す String を返します。

`systemId`

W3C DOM Level 2 プロパティ。このプロパティは、外部サブセットのシステム識別子を表す String を返します。

エンティティ

エンティティの操作に使用します。この項では、次のトピックについて説明します。

- ◆ [317 ページの「publicId」](#)
- ◆ [317 ページの「systemId」](#)
- ◆ [318 ページの「notationName」](#)

`publicId`

W3C DOM Level 2 プロパティ。このプロパティは、外部サブセットのパブリック識別子を表す String を返します。

`systemId`

W3C DOM Level 2 プロパティ。このプロパティは、外部サブセットのシステム識別子を表す String を返します。

notationName

W3C DOM Level 1 Entity プロパティ。このプロパティは String タイプです。Entity には、Node のすべてのプロパティおよびメソッドがあります。

ProcessingInstruction

処理命令の操作に使用します。この項では、次のトピックについて説明します。

- ◆ [318 ページの 「target」](#)
- ◆ [318 ページの 「data」](#)

target

W3C DOM Level 1 ProcessingInstruction プロパティ。このプロパティは、処理命令のターゲット部の String 表現です。

data

W3C DOM Level 1 ProcessingInstruction プロパティ。このプロパティは、処理命令のデータ部の String 表現です。

10.3.3 ECMAScript Core

この項では、ECMA 式ビルダで使用できるすべての ECMAScript コアメソッドとプロパティを一覧します。この項では、次のトピックについて説明します。

- ◆ [318 ページの 「Array オブジェクト」](#)
- ◆ [319 ページの 「Boolean オブジェクト」](#)
- ◆ [320 ページの 「Date オブジェクト」](#)
- ◆ [326 ページの 「Function オブジェクト」](#)
- ◆ [326 ページの 「Global」](#)
- ◆ [328 ページの 「Math オブジェクト」](#)
- ◆ [332 ページの 「Number オブジェクト」](#)
- ◆ [334 ページの 「Object」](#)
- ◆ [334 ページの 「String オブジェクト」](#)

Array オブジェクト

配列の操作に使用します。この項では、次のトピックについて説明します。

- ◆ [319 ページの 「Array \(item0, item1, ...\)」](#)
- ◆ [319 ページの 「join \(separator\)」](#)
- ◆ [319 ページの 「length」](#)
- ◆ [319 ページの 「reverse \(\)」](#)
- ◆ [319 ページの 「sort \(comparefn\)」](#)
- ◆ [319 ページの 「toString \(\)」](#)

Array (item0, item1, ...)

Array()

コンストラクタ

join (separator)

Array join(separator)

配列要素は文字列に変換されます。その後、これらの文字列は連結され、セパレータで区切られます。セパレータが指定されていない場合、单一のカンマがセパレータとして使用されます。

length

配列の長さ。この Array オブジェクトの length プロパティ。

reverse ()

reverse()

配列要素が再配置され、順序が逆になります。操作は適切に行われ、元の配列は変更されます。

sort (comparefn)

Array sort()

この配列の要素はソートされます。ソートは必ずしも安定していません。comparefn が指定されている場合、x と y の 2 つの引数を受け入れる関数となり、 $x < y$ の場合は負の値、 $x = y$ の場合は 0、または $x > y$ の場合は正の値を返します。

toString ()

Array toString()

このオブジェクトの要素は文字列に変換されます。その後、これらの文字列は連結され、カンマ文字で区切られます。結果は組み込みの join メソッドが引数なしでこのオブジェクトに対して呼び出された場合と同じとなります。

Boolean オブジェクト

通常、True/False のリテラル値の代わりに Boolean オブジェクトバージョンを使用することはありません。このオブジェクトは補完の目的で提供されています。この仕様については、「ECMA-262」を参照してください。

この項では、次のトピックについて説明します。

- ◆ [319 ページの 「Boolean \(\)」](#)
- ◆ [320 ページの 「toString \(\)」](#)
- ◆ [320 ページの 「valueOf \(\)」](#)

Boolean ()

Boolean([true/false])

コンストラクタ。オプションとして、True または False を引数に取ります。

```
toString ()  
Boolean toString()
```

このブール値が True の場合、文字列「true」が返されます。それ以外の場合、このブール値は False になります。この場合、文字列「false」が返されます。

```
valueOf ()  
Boolean valueOf()
```

このブール値を返します。

Date オブジェクト

日付と時刻の操作に使用します。この項では、次のトピックについて説明します。

- ◆ 321 ページの 「Date ()」
- ◆ 321 ページの 「getDate ()」
- ◆ 321 ページの 「getDay ()」
- ◆ 321 ページの 「getFullYear ()」
- ◆ 321 ページの 「getHours ()」
- ◆ 321 ページの 「getMilliseconds ()」
- ◆ 322 ページの 「getMinutes ()」
- ◆ 322 ページの 「getMonth ()」
- ◆ 322 ページの 「getSeconds ()」
- ◆ 322 ページの 「getTime ()」
- ◆ 322 ページの 「getTimezoneOffset ()」
- ◆ 322 ページの 「getUTCDate ()」
- ◆ 322 ページの 「getUTCDay ()」
- ◆ 322 ページの 「getUTCFullYear ()」
- ◆ 322 ページの 「getUTCHours ()」
- ◆ 323 ページの 「getUTCMilliseconds ()」
- ◆ 323 ページの 「getUTCMinutes ()」
- ◆ 323 ページの 「getUTCSeconds ()」
- ◆ 323 ページの 「getYear ()」
- ◆ 323 ページの 「parse (string)」
- ◆ 323 ページの 「 setDate (date)」
- ◆ 323 ページの 「setFullYear (year[,mon[,date]])」
- ◆ 323 ページの 「setHours (hour[,min[,sec[,ms]]])」
- ◆ 323 ページの 「setMilliseconds (ms)」
- ◆ 324 ページの 「setMinutes (min[,sec[,ms]])」
- ◆ 324 ページの 「setMonth (mon[,date])」
- ◆ 324 ページの 「setSeconds (sec [, ms])」

- ◆ 324 ページの 「`setTime (time)`」
- ◆ 324 ページの 「`setUTCDate (date)`」
- ◆ 324 ページの 「`setUTCFullYear (year[,mon[,date]])`」
- ◆ 324 ページの 「`setUTCHours (min[,sec[,ms]])`」
- ◆ 324 ページの 「`setUTCMilliseconds (ms)`」
- ◆ 325 ページの 「`setUTCMinutes (min[,sec[,ms]])`」
- ◆ 325 ページの 「`setUTCMonth (mon[,date])`」
- ◆ 325 ページの 「`setUTCSeconds (sec [, ms])`」
- ◆ 325 ページの 「`setYear (year)`」
- ◆ 325 ページの 「`toLocaleString ()`」
- ◆ 325 ページの 「`toString ()`」
- ◆ 325 ページの 「`toUTCString ()`」
- ◆ 325 ページの 「`UTC ()`」
- ◆ 325 ページの 「`valueOf ()`」

`Date ()`

`Date()`

`Date` のコンストラクタには、さまざまな署名があります。`date` コンストラクタの形式は、最大 7 個のパラメータを受け入れることができます。`new Date (year,month,date,hrs,mins,secs,ms)` の形式を使用します。Identity Manager ユーザアプリケーションのワークフローシステムで使用する場合、`date` には ECMAScript Date オブジェクトではなく、`java.util.Date` オブジェクトを指定する必要があります。

`getDate ()`

`getDate()`

`DateFromTime (LocalTime (t))` を返します。

`getDay ()`

`getDay()`

`WeekDay (LocalTime (t))` を返します。週の曜日は 0 から 6 までの数値で表されます。0 は日曜日、6 は土曜日です。

`getFullYear ()`

`getFullYear()`

`YearFromTime (LocalTime (t))` を返します。

`getHours ()`

`getHours()`

`HourFromTime (LocalTime (t))` を返します。

`getMilliseconds ()`

`getMilliseconds()`

`msFromTime (LocalTime (t))` を返します。

`getMinutes ()`
`getMinutes ()`

`MinFromTime (LocalTime (t))` を返します。

`getMonth ()`
`getMonth ()`

`MonthFromTime (LocalTime (t))` を返します。月は 0 から 11 までの整数値で表されます。0 は 1 月、11 は 12 月です。

`getSeconds ()`
`getSeconds ()`

`SecFromTime (LocalTime (t))` を返します。

`getTime ()`
`getTime ()`

時間値を表す数値を返します。この数値は、指定された Date オブジェクトをミリ秒単位で表しています。

`getTimezoneOffset ()`
`getTimezoneOffset ()`

`(t * LocalTime (t)) / msPerMinute` を返します。GMT とローカルタイムの時差を分単位で表しています。

`getUTCDate ()`
`getUTCDate ()`

`DateFromTime (t)` を返します。

`getUTCDay ()`
`getUTCDay ()`

`WeekDay (t)` を返します。週の曜日は 0 から 6 までの数値で表されます。0 は日曜日、6 は土曜日です。

`getUTCFullYear ()`
`getUTCFullYear ()`

`YearFromTime (t)` を返します。`getYearUTC` メソッドは存在しません。このため、このメソッドを使用して UTC Date オブジェクトから年を取得する必要があります。

`getUTCHours ()`
`getUTCHours ()`

`HourFromTime (t)` を返します。

getUTCMilliseconds ()
getUTCMilliseconds()

msFromTime (t) を返します。

getUTCMinutes ()
getUTCMinutes()

MinFromTime (t) を返します。

getUTCSeconds ()
getUTCSeconds()

SecFromTime (t) を返します。

getYear ()
getYear()

YearFromTime (LocalTime (t)) —1900 を返します。getFullYear () 関数が 2000 年問題を回避しているため、通常はこちらが推奨されます。

parse (string)
parse(string)

ToString 演算子を引数に適用して、結果の文字列を日付として解釈した後、この日付に対する UTC 時間値を示す数値を返します。文字列は、そのコンテンツによって、ローカルタイム、UTI 時間、またはその他のタイムゾーンとして解釈されます。

setDate (date)
setDate(date)

1 から 31 までの整数値を使って、指定された date の日付をローカルタイムで設定します。

setFullYear (year[,mon[,date]])
setFullYear(year[,mon[,date]])

この値の [Value] プロパティを UTC ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。

setHours (hour[,min[,sec[,ms]]])
setHours(hour[,min[,sec[,ms]]])

この値の [Value] プロパティを UTC 時間に設定します。この値の [Value] プロパティの値を返します。時間の値を入力する場合、23 より大きい時間値は既存の値に追加され、設定は行われません。

setMilliseconds (ms)
setMilliseconds(ms)

引数から UTC を計算し、この値の [Value] プロパティを TimeClip (calculatedUTCtime) に設定します。この値の [Value] プロパティの値を返します。

```
setMinutes (min[,sec[,ms]])
```

```
setMinutes(min[, sec[, ms]])
```

この値の [Value] プロパティを UTC 時間に設定します。この値の [Value] プロパティの値を返します。

```
setMonth (mon[,date])
```

```
setMonth(mon[, date])
```

この値の [Value] プロパティを UTC ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。この [Value] プロパティが 11 より大きい場合、この [Value] プロパティは既存の月に追加され、設定は行われません。

```
setSeconds (sec [, ms ] )
```

```
setSeconds(sec [, ms ] )
```

この値の [Value] プロパティを UTC 時間に設定します。この値の [Value] プロパティの値を返します。

```
setTime (time)
```

```
setTime(time)
```

この値の [Value] プロパティを TimeClip (time) に設定します。この値の [Value] プロパティの値を返します。この [Value] プロパティは、TimeClip (time) メソッドによって変換されるミリ秒値で示されます。

```
setUTCDate (date)
```

```
setUTCDate(date)
```

この値の [Value] プロパティを ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。この [Value] プロパティが 30 または 31 より大きい場合、この [Value] は既存の日付に追加され、設定は行われません。

```
setUTCFullYear (year[,mon[,date]])
```

```
setUTCFullYear(year[, mon[, date]])
```

この値の [Value] プロパティを ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。

```
setUTCHours (min[,sec[,ms]])
```

```
setUTCHours(min[, sec[, ms]])
```

この値の [Value] プロパティを時間に設定します。この値の [Value] プロパティの値を返します。時間の値を入力する場合、23 より大きい時間値は既存の時間値に追加され、設定は行われません。

```
setUTCMilliseconds (ms)
```

```
setUTCMilliseconds(ms)
```

この値の [Value] プロパティを時間に設定し、この値の [Value] プロパティの値を返します。

setUTCMinutes (min[,sec[,ms]])
setUTCMinutes(min[,sec[,ms]])

この値の [Value] プロパティを時間に設定します。この値の [Value] プロパティの値を返します。

setUTCMonth (mon[date])
setUTCMonth(mon[,date])

この値の [Value] プロパティを ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。この [Value] プロパティが 11 より大きい場合、[Value] プロパティは既存の月に追加され、設定は行われません。

setUTCSeconds (sec [, ms])
setUTCSeconds(sec [, ms])

この値の [Value] プロパティを時間に設定します。この値の [Value] プロパティの値を返します。

setYear (year)
setYear(year)

この値の [Value] プロパティを UTC ECMAScript.Date に設定します。この値の [Value] プロパティの値を返します。

toLocaleString ()
toLocaleString()

文字列値を返します。文字列のコンテンツは実装に依存しますが、地域ロケールまたは文化ロケールに適切な、使いやすく人間による読み込みが可能な形式で日付が示されます。

toString ()
toString()

この文字列値を返します。文字列のコンテンツは実装依存ですが、人間が読める使いやすい形式で現在のタイムゾーンの Date を表します。

toUTCString ()
toUTCString()

文字列値を返します。文字列のコンテンツは実装に依存しますが、人間が読める使いやすい形式で UTC の Date を表します。

UTC ()
UTC()

このメソッドは多数のさまざまな引数を受け入れます。UTC 関数は 2 つの点で Date コンストラクタと異なっています。1 つは、Date オブジェクトを作成せず、時間値を数字で返す点です。もう 1 つは、引数をローカルタイムではなく UTC で解釈する点です。

valueOf ()
valueOf()

時間値を表す数値を返します。`valueOf()` 関数は汎用ではありません。このため、オブジェクトが Date オブジェクトでない場合、ランタイムエラーが生成されます。

Function オブジェクト

Function オブジェクトを操作できます。この項では、次のトピックについて説明します。

- ◆ [326 ページの 「Function \(p1, p2, ..., pn, body\)」](#)
- ◆ [326 ページの 「length」](#)
- ◆ [326 ページの 「toString \(\)」](#)

`Function (p1, p2, ..., pn, body)`

Function コンストラクタ。最後の引数で、関数の本文（実行可能コード）が指定されます。先行するその他の引数では正式なパラメータが指定されます。

`length`

`length` プロパティの値は、通常、その関数に設定される「典型的な」引数の数を示す整数です。ただし、この言語では、それ以外の数の引数で関数を呼び出すことができます。`length` プロパティに指定された数とは別の数の引数が指定されたときの関数の動作は、関数によって異なります。

`toString ()`

```
String toString()
```

関数の実装依存の表現が返されます。この表現の構文は、`FunctionDeclaration` と同じです。表現文字列での空白文字、行末文字、およびセミコロンの使用と配置は、実装に依存します。

Global

ECMAScript は、親オブジェクトを持たず、あらゆるコンテキストで使用できる「トップレベル」のメソッドおよびプロパティを提供します。

この項では、次のトピックについて説明します。

- ◆ [326 ページの 「escape \(string\)」](#)
- ◆ [327 ページの 「eval \(x\)」](#)
- ◆ [327 ページの 「Infinity」](#)
- ◆ [327 ページの 「isFinite \(number\)」](#)
- ◆ [327 ページの 「isNaN \(value\)」](#)
- ◆ [327 ページの 「NaN」](#)
- ◆ [327 ページの 「parseFloat \(string\)」](#)
- ◆ [327 ページの 「parseInt \(string, radix\)」](#)
- ◆ [328 ページの 「unescape \(string\)」](#)

`escape (string)`

```
String escape()
```

`escape` 関数は、URL に適していない文字列を 16 進のエスケープシーケンスに置換することにより、新しく URL に適した文字列を計算します。

```
eval (x)
eval()
```

1 つの引数 `x` を指定して `eval` 関数を呼び出すと、次の処理が行われます。

1. `x` が文字列値でない場合、`x` を返します。
2. `x` を ECMAScript プログラムとして解析します。解析に失敗した場合、ランタイムエラーが生成されます。
3. 手順 2 からプログラムを評価します。
4. 値「V」の後で `Result (3)` が「標準」に終了した場合、値 `V` を返します。
5. `undefined` を返します。

Infinity

正の数の無限値を表す特別なプリミティブ値です。

```
isFinite (number)
isFinite()
```

引数に `Number ()` を適用して、結果が `Nan` (数字以外)、`+`、または `**` の場合は `False` を返します。それ以外の場合は `True` を返します。

```
isNaN (value)
isNaN()
```

引数が `Nan` (数字意外) 評価の場合は `True`、それ以外の場合は `False` を返します。

注: `Nan` と任意の値 (`Nan` 自体も含む) の論理比較の形式は、常に `False` を返します。変数または戻り値などが `Nan` に等しいかどうかを調べるには、`isNaN ()` を使用します。is equal to `Nan`.

NaN

プリミティブ値 `NaN` は、IEEE 標準の「数字以外」の値セットを表します。

```
parseFloat (string)
number parseFloat()
```

文字列引数のコンテンツを解釈して、浮動小数点数を生成します。文字列を数値に変換できない場合は、特別な値 `NaN` (327 ページの「`NaN`」を参照) が返されます。

```
parseInt (string, radix)
number parseInt()
```

文字列引数のコンテンツを指定の基底に従って解釈することによって決定される整数値を生成します。

```
unescape (string)
String unescape()
```

escape 関数で指定されたエスケープシーケンスをそれらが表す文字に置き換えることにより、新しい文字列値を計算します。

Math オブジェクト

Math オブジェクトのプロパティとメソッドはすべてスタティックです。つまり、コード内のプロパティ名やメソッド名の先頭に「Math」を付加する必要があります。たとえば「PI」ではなく「Math.PI」を使用します。

この項では、次のトピックについて説明します。

- ◆ [328 ページの 「E」](#)
- ◆ [329 ページの 「LN10」](#)
- ◆ [329 ページの 「LN2」](#)
- ◆ [329 ページの 「LOG2E」](#)
- ◆ [329 ページの 「LOG10E」](#)
- ◆ [329 ページの 「PI」](#)
- ◆ [329 ページの 「SQRT1.2」](#)
- ◆ [329 ページの 「SQRT2」](#)
- ◆ [329 ページの 「abs \(x\)」](#)
- ◆ [329 ページの 「acos \(x\)」](#)
- ◆ [329 ページの 「asin \(x\)」](#)
- ◆ [330 ページの 「atan \(x\)」](#)
- ◆ [330 ページの 「atan2 \(x,y\)」](#)
- ◆ [330 ページの 「ceil \(x\)」](#)
- ◆ [330 ページの 「cos \(x\)」](#)
- ◆ [330 ページの 「exp \(x\)」](#)
- ◆ [331 ページの 「floor \(x\)」](#)
- ◆ [331 ページの 「log \(x\)」](#)
- ◆ [331 ページの 「max \(x,y\)」](#)
- ◆ [331 ページの 「min \(x,y\)」](#)
- ◆ [331 ページの 「pow \(x,y\)」](#)
- ◆ [331 ページの 「random \(\)」](#)
- ◆ [332 ページの 「round \(x\)」](#)
- ◆ [332 ページの 「sin \(x\)」](#)
- ◆ [332 ページの 「sqrt \(x\)」](#)
- ◆ [332 ページの 「tan \(x\)」](#)

E

自然対数の底である e の数値。約 2.7182818284590452354 です。

LN10

10 の自然対数の数値。約 2.302585092994046 です。

LN2

2 の自然対数の数値。約 0.6931471805599453 です。

LOG2E

自然対数の底である e の 2 を底とした対数の数値。この値は約 1.4426950408889634 です。
Math.LOG2E の値は、Math.LN2 の値の逆数に近い値になります。

LOG10E

自然対数の底である e の 10 を底とした対数の数値。この値は約 0.4342944819032518 です。
Math.LOG10E の値は、Math.LN10 の値の逆数に近い値になります。

PI

円の円周に対する直径の比である * の数値。約 3.14159265358979323846 です。

SQRT1.2

1/2 の平方根の数値。約 0.7071067811865476 です。Math.SQRT1_2 の値は、Math.SQRT2 の
値の逆数に近い値になります。

SQRT2

2 の平方根の数値。約 1.4142135623730951 です。

abs (x)

Number abs(x)

引数 x の絶対値を返します。通常、結果の大きさは引数と同じですが、正符号が付きます。
入力値 x には任意の数値を使用できます。

例：

Math.abs(-123.23940) = 123.23940

acos (x)

Number acos(x)

この関数は、引数のアークコサインに対する実装依存の近似値を返します。結果はラジアンで表され、+0 から +PI (3.14159...) ラジアンの範囲です。入力値 x は -1.0 から 1.0 の範囲の数字にする必要があります。

例：

PI/4 = 0.785 Math.acos(0.785) = 0.6681001997570769

asin (x)

Number asin(x)

この関数は、引数のアークサインに対する実装依存の近似値を返します。結果はラジアンで表され、PI/2 から +PI/2 の範囲です。入力値 x は -1.0 から 1.0 の範囲の数字にする必要があります。

例：

```
PI/4 = 0.785 Math.asin(0.785) = 0.9026961270378197
```

atan (x)

```
Number atan(x)
```

この関数は、引数のアークタンジェントに対する実装依存の近似値を返します。結果はラジアンで表され、PI/2 から +PI/2 の範囲です。入力値 x には任意の数値を使用できます。

例：

```
3PI/4 = 2.355 Math.atan(2.355) = 1.169240427545485
```

atan2 (x,y)

```
Number atan2(x,y)
```

この関数は、引数 y と x の商 y/x のアークタンジェントに対する実装依存の近似値を返します。引数の符号で、結果の象限を判断します。2つの引数を使用するアークタンジェント関数では、従来から意図的に、y が第 1 引数、x が第 2 引数になります。結果はラジアンで表され、-PI から +PI の範囲です。入力値 x は点の x 座標です。入力値 y は点の y 座標です。

例：

```
PI/2 = 1.57 Math.atan2(1.57,-1.57) = 2.356194490192345
```

ceil (x)

```
Number ceil(x)
```

この関数は、引数以上で数学的な整数と等しい、最小（負の数の無限値に最も近い）の数値を返します。引数がすでに整数の場合、結果は引数自体になります。入力値 x には、任意の数値または式を使用できます。Math.ceil (x) 関数のプロパティは、-Math.floor (-x) と同じです。例：

例：

```
Math.ceil(123.78457) = 123
```

cos (x)

```
Number cos(x)
```

この関数は、引数のコサインに対する実装依存の近似値を返します。引数はラジアンで表されます。

exp (x)

```
Number exp(x)
```

この関数は、引数の指数関数（引数の乗数で累乗された e。e は自然対数の底）に対する実装依存の近似値を返します。入力値 x には、0 より大きい任意の数値または式を使用できます。

例：

```
Math.exp(10) = 22026.465794806718
```

floor (x)

```
Number floor(x)
```

この関数は、引数以下で数学的な整数に等しい最大（正の数の無限値に最も近い）の数値を返します。引数がすでに整数の場合、結果は引数自体になります。入力値 x には、任意の数値または式を使用できます。

例：

```
Math.floor(654.895869)=654
```

log (x)

```
Number log(x)
```

この関数は、引数の自然対数に対する実装依存の近似値を返します。入力値 x には、0 より大きい任意の数値または式を使用できます。

例：

```
Math.log(2) = 0.6931471805599453
```

max (x,y)

```
Number max(x, y)
```

この関数は、2 つの引数の大きいほうの値を返します。入力値 x および y には、任意の数値または式を使用できます。

例：

```
Math.max(12.345,12.3456)= 12.3456
```

min (x,y)

```
Number min(x, y)
```

この関数は、2 つの引数の小さいほうの値を返します。入力値 x および y には、任意の数値または式を使用できます。

例：

```
Math.min(-12.457,-12.567)= -12.567
```

pow (x,y)

```
Number pow(x, y)
```

この関数は、x を y の乗数で累乗した結果に対する実装依存の近似値を返します。入力値 x は、乗数で累乗される数値にする必要があります。入力値 y は、累乗する乗数にする必要があります。

例：

```
Math.pow(2,4) = 16
```

random ()

```
Number random()
```

このメソッドは引数を取りません。0から1の間の擬似ランダム数を返します。この数値は、実装依存のアルゴリズムや方法を使用して、該当する範囲内にほぼ均等に分布しています。この関数は引数を取りません。

例：

```
Math.random()=0.9545176397178535
```

round (x)

```
Number round(x)
```

この関数は、引数に最も近い数学的な整数である数値を返します。2つの整数の数値が引数に同等に近い場合、正の数の無限値に近い数値が選択されます。引数がすでに整数の場合、結果は引数自体になります。入力値 x には任意の数値を使用できます。

例：

```
Math.round(13.53) = 14
```

sin (x)

```
Number sin(x)
```

この関数は、引数のサインに対する実装依存の近似値を返します。引数はラジアンで表されます。入力値 x はラジアン単位の角度にする必要があります。

sqrt (x)

```
Number sqrt(x)
```

この関数は、引数の平方根に対する実装依存の近似値を返します。入力値 x には、0以上の任意の数値または式を使用できます。入力値 x が 0 未満の場合、文字列「NaN」が返されます。NaN は「数値以外」を表します。

例：

```
Math.sqrt(25) = 5
```

tan (x)

```
Number tan(x)
```

この関数は、引数のタンジェントに対する実装依存の近似値を返します。引数はラジアンで表されます。入力値 x はラジアン単位の角度にする必要があります。

Number オブジェクト

数値の操作に使用します。Number オブジェクトは、プリミティブな数値のオブジェクトラッパーです。

この項では、次のトピックについて説明します。

- ◆ [333 ページの 「MAX_VALUE」](#)
- ◆ [333 ページの 「MIN_VALUE」](#)
- ◆ [333 ページの 「NaN」](#)
- ◆ [333 ページの 「NEGATIVE_INFINITY」](#)
- ◆ [333 ページの 「Number \(\)」](#)
- ◆ [333 ページの 「POSITIVE_INFINITY」](#)

- ◆ 333 ページの 「`toString (radix)`」
- ◆ 334 ページの 「`valueOf ()`」

MAX_VALUE

数値タイプの最大の正の有限値で、約 1.7976931348623157e308 です。

例：

```
Number.MAX_VALUE
```

MIN_VALUE

数値タイプの最小のゼロでない正の値で、約 5e-324 です。

例：

```
Number.MIN_VALUE
```

NaN

プリミティブ値 NaN は、IEEE 標準の「数字以外」の値セットを表します。

例：

```
Number.NaN
```

NEGATIVE_INFINITY

負の無限値です。

例：

```
Number.NEGATIVE_INFINITY
```

Number ()

```
Number()
```

Number のコンストラクタには、`Number (value)` と `Number ()` の 2 つの形式があります。

POSITIVE_INFINITY

正の無限値です。

例：

```
Number.POSITIVE_INFINITY
```

toString (radix)

```
toString()
```

基数が数字 10 または指定されていない場合、この数値が `ToString` 演算子の引数になります。結果の文字列値が返されます。10 を除く 2 から 36 までの整数である基数が指定された場合、結果は文字列になります。この選択は実装依存です。`toString` 関数は汎用ではありません。この値が Number オブジェクトではない場合はランタイムエラーが生成されます。このため、この関数を他の種類のオブジェクトに転送して、メソッドとして使用することはできません。

```
valueOf ()  
valueOf()
```

この数値を返します。valueOf 関数は汎用ではありません。値が Number オブジェクトではない場合はランタイムエラーが生成されます。このため、この関数を他の種類のオブジェクトに転送して、メソッドとして使用することはできません。

Object

オブジェクトを操作できます。Object は、プリミティブ JavaScript オブジェクトタイプです。すべての ECMAScript オブジェクトは、オブジェクトの子孫です。このため、すべての ECMAScript オブジェクトは、オブジェクト用に定義されたメソッドを持っています。

この項では、次のトピックについて説明します。

- ◆ [334 ページの 「Object \(\)」](#)
- ◆ [334 ページの 「toString \(\)」](#)
- ◆ [334 ページの 「valueOf \(\)」](#)

Object ()

オブジェクトのコンストラクタ。

```
toString ()  
Object toString()
```

任意のオブジェクトで toString メソッドを呼び出した場合、次の処理が行われます。

1. このオブジェクトの [[Class]] プロパティを取得します。
2. 「[object]」、「Result (1)」、および「]」の 3 つの文字列を連結して、文字列値を計算します。
3. Result (2) を返します。

```
valueOf ()  
Object valueOf()
```

オブジェクトの valueOf メソッドは、通常、オブジェクトを返します。しかし、オブジェクトがホストオブジェクトのラッパーである場合は、Object コンストラクタによって生成されるように、含まれるホストオブジェクトが返されます。

String オブジェクト

String オブジェクトを操作できます。この項では、次のトピックについて説明します。

- ◆ [335 ページの 「String \(x\)」](#)
- ◆ [335 ページの 「charAt \(pos\)」](#)
- ◆ [335 ページの 「charCodeAt \(pos\)」](#)
- ◆ [335 ページの 「fromCharCode \(char0, char1, ...\)」](#)
- ◆ [335 ページの 「indexOf \(searchString, pos\)」](#)
- ◆ [336 ページの 「lastIndexOf \(searchString, pos\)」](#)

- ◆ 336 ページの 「length」
- ◆ 336 ページの 「match (RegExp)」
- ◆ 336 ページの 「replace (RegExp, String)」
- ◆ 336 ページの 「search (RegExp)」
- ◆ 336 ページの 「split (separator)」
- ◆ 336 ページの 「substring (start, end)」
- ◆ 336 ページの 「toLowerCase ()」
- ◆ 337 ページの 「toString ()」
- ◆ 337 ページの 「toUpperCase ()」
- ◆ 337 ページの 「valueOf ()」

String (x)

String(x)

文字列のコンストラクタ。

charAt (pos)

charAt(pos)

このオブジェクトを文字列に変換した結果生成された文字列の位置 pos の文字を含む文字列を返します。該当する位置に文字がない場合、結果は空の文字列になります。結果は文字列値であり、String オブジェクトではありません。

charCodeAt (pos)

charCodeAt(pos)

このオブジェクトを文字列に変換した結果の文字列の位置 pos の文字の Unicode コードポイントエンコードを表す数値 (2¹⁶ より小さい正の整数) を返します。該当する位置に文字がない場合、結果は NaN になります。

fromCharCode (char0, char1, ...)

fromCharCode(char0, char1, . . .)

引数と同じ数の文字を含む文字列値を返します。各引数は結果文字列の 1 つの文字を指定します。第 1 引数で最初の文字を指定して、左から右へと順番に進みます。引数は ToUint16 を適用することによって文字に変換され、結果の 16 ビットの整数は文字の Unicode コードポイントのエンコードとして認識されます。引数が指定されていない場合、結果は空の文字列になります。

indexOf (searchString, pos)

indexOf(searchString, pos)

指定された searchString がこのオブジェクトを文字列に変換した結果の部分文字列として、指定された位置またはその右側の 1 つまたは複数の位置に表示される場合、一番左側の位置のインデックスが返されます。それ以外の場合は -1 が返されます。位置が定義されていない場合や指定されていない場合は、すべての文字列を検索するため、0 が使用されます。

```
lastIndexOf (searchString, pos)  
lastIndexOf(searchString, pos)
```

指定された `searchString` がこのオブジェクトを文字列に変換した結果の部分文字列として指定された位置またはその左側の 1 つまたは複数の位置に表示される場合、一番右側の位置のインデックスが返されます。それ以外の場合は -1 が返されます。位置が定義されていない場合や指定されていない場合は、すべての文字列を検索するため、文字列値の長さが使用されます。

length

`String` の長さを返します。

```
match (RegExp)  
String match(RegExp)
```

正規表現オブジェクトを引数として使用します。一致項目の配列を返します。それ以外の場合は `null` を返します。

```
replace (RegExp, String)  
String replace(RegExp, String)
```

正規表現と置換文字列を使用します。置換を実行し、元の文字列を返します。

```
search (RegExp)  
String search(RegExp)
```

正規表現を単独の引数として使用し、一致する最初の部分文字列のオフセットを返します。一致しない場合は -1 を返します。

```
split (separator)  
split(separator)
```

`Array` オブジェクトを返します。この `Array` オブジェクトには、このオブジェクトを文字列に変換した結果の部分文字列が格納されています。文字列は、指定のセパレータを左から右へ検索することによって決定されます。セパレータは文字列値を分割するために使用するもので、返される配列の部分文字列には含まれていません。セパレータは任意の長さの文字列になります。

```
substring (start, end)  
substring(start, end)
```

このオブジェクトを文字列に変換した結果の部分文字列を、文字列の開始位置 `start` から終了位置 `end` まで返します。2 番目のパラメータが存在しない場合、終了位置が文字列の末尾であると認識されます。結果は文字列値であり、`String` オブジェクトではありません。

```
toLowerCase ()  
toLowerCase()
```

このオブジェクトを文字列に変換した結果の長さと等しい長さの文字列を返します。結果は文字列値であり、`String` オブジェクトではありません。結果の各文字は、対応する Unicode 2.0 の小文字がない場合、対応する文字列の文字と等しくなります。対応する

Unicode 2.0 の小文字がある場合は、その小文字が使用されます。実装やロケールに関係なく、正準的な Unicode 2.0 ケースマッピングが使用されます。

toString ()
toString()

この文字列値を返します。文字列値での空白文字、行末文字、およびセミコロンの使用と配置は、実装に依存します。

toUpperCase ()
toUpperCase()

このオブジェクトを文字列に変換した結果の長さと等しい長さの文字列を返します。結果は文字列値であり、String オブジェクトではありません。結果の各文字は、対応する Unicode 2.0 の大文字がない場合、対応する文字列の文字と等しくなります。対応する Unicode 2.0 の大文字がある場合は、その大文字が使用されます。実装やロケールに関係なく、正準的な Unicode 2.0 ケースマッピングが使用されます。

valueOf ()
valueOf()

この文字列値を返します。valueOf () 関数は汎用関数ではありません。このため、オブジェクトが String オブジェクトでない場合、ランタイムエラーが生成されます。

10.3.4 Global 関数

ECMAScript の Global 関数は、特定のオブジェクトに依存しない関数です。

式ビルダは、次の Global 関数をサポートします。

- ◆ [337 ページの 「getEnvironmentCountry」](#)
- ◆ [337 ページの 「getEnvironmentLanguage」](#)

getEnvironmentCountry

getEnvironmentCountry()

ユーザのコンピュータに現在設定されているロケーションを表す、2 文字の文字列 (US など) を返します。

getEnvironmentLanguage

getEnvironmentLanguage()

ユーザのコンピュータの現在の入力言語を表す、2 文字の文字列 (EN など) を返します。

10.3.5 IDVault 関数

この項では、IDVault データとともに使用する関数を紹介します。

- ◆ [338 ページの 「DNCompare」](#)
- ◆ [338 ページの 「get \(\)」](#)

- ◆ 338 ページの 「execService ()」

DNCompare

```
DNcompare(String dn1, String dn2)
```

識別ポールトから、DN の比較を実行します。大文字と小文字は区別しません。DN が同じである場合は True を返します。

DN は識別名（LDAP 名とコンテキスト）をカプセル化したものです。DN の構文は、DN の String 表現に関する規定 RFC 2253 に準拠している必要があります。DNCompare では次の DN を使用できます。比較の結果は True です。

```
cn=jdoe,ou=users,ou=idmsample,o=acmeCN=jdoe,ou=users,ou=idmsample,o=acmeCN=JDOE,ou=users,ou=idmsample,o=acme
```

RFC 2253 の詳細については、「RFC 2353」(<http://www.ietf.org/rfc/rfc2253.txt>) を参照してください。

例：

```
if ( IDVault.DNcompare(flowdata.get('Activity3/CardRequest/Candidate'),recipient) ) true; else false ;
```

get ()

```
get()  
get()  
get(fieldname)  
get(fieldname,dn)  
get(fieldname,dn,entity-type)  
get(fieldname,dn,entity-type,attribute)
```

ワークフロースクリプトエンジンの IDVault.get () 関数に対応しています。与えられたエンティティの属性値を取得します。結果は配列になります。fieldname パラメータを指定した場合、クエリの結果によってフィールドのコンテンツが更新されます。fieldname パラメータが指定されていない場合、クエリの結果を使用するかどうかはフォーム開発者が決定します。例：

```
IDVault.get("assetProp",dn,"user","LastName");
```

execService ()

```
IDVault.execService(service)  
IDVault.execService(service, param)  
IDVault.execService(service, param, locale)
```

AJAX サービスを実行し、その結果を使ってフィールドのコンテンツを更新します。サービスは UI コントロールレジストリに登録する必要があります。結果リストの最初のカラムには表示値、2 番目のカラムにはデータ値が入ります。例：

```
var r=IDVault.execService("dnlookup2",params);var  
res=r?r["_data"]["raw"][dn]["value"]:"error encountered";  
field.setValues("IDVault.execService(\"dnlookup2\") :" + res);
```

プロジェクトチームの設定

11

Identity Manager ユーザインターフェースの [要求と承認] タブには、[マイチームの作業] というアクションカテゴリがあります。[マイチームの作業] アクションカテゴリにあるアクションを使うことにより、ワークフロー内のチームメンバーのタスクと要求を処理することができます。この節では、チームを作成し、その特性（メンバー、マネージャ、要求権など）を定義する方法について説明します。主なトピックは次のとおりです。

- ◆ 339 ページのセクション 11.1 「チームの概要」
- ◆ 341 ページのセクション 11.2 「プロジェクトチームの管理」

11.1 チームの概要

チームはユーザのグループを表し、このチームに関連付けられたプロジェクト要求と承認タスクを管理できるユーザが定義されています。チーム定義は、以下で説明するように、チームマネージャ、チームメンバー、およびチームオプションのリストで構成されています。

- ◆ チームマネージャとは、チームメンバーのための要求やタスクを管理できるユーザです。チームマネージャには、チームメンバーのために代理や委任を設定する権限を与えることもできます。チームマネージャにはユーザまたはグループがなることができます。
- ◆ チームメンバーとは、チームに参加することを許可されたユーザのことです。チームメンバーにはディレクトリ内のユーザ、グループ、またはコンテナがなることができます。または、ディレクトリ関係から生成することもできます。たとえば、メンバーのリストは組織内の「マネージャ - 従業員」の関係から生成できます。この場合のチームメンバーは、チームマネージャに報告義務のあるすべてのユーザを指します。

注: プロジェクトアプリケーション管理者は、階層式関係をサポートするためのディレクトリ抽象化レイヤを設定できます。これにより、組織内のいくつかのレベルを1チームに含めることができます。管理者は含めるレベルの数を設定できます。

-
- ◆ チームオプションは、プロジェクト要求のスコープを決定するもので、チームマネージャが個別のプロジェクト要求を実行できるのか、1つ以上の要求カテゴリを実行できるのか、またはすべての要求を実行できるのかを指定します。チームオプションは、チームマネージャがチームメンバーのために代理を設定できるか、または委任の目的でチームメンバーの要請を設定できるかを決定します。

注: ユーザアプリケーションは、1 レベルの代理人割り当てのみをサポートしています。代理人割り当てが複数レベルに反映されることはありません。

プロジェクトアプリケーション管理者は、すべてのチーム管理機能を実行できます。

チームとグループの違い チームがアイデンティティボールト中のグループを表す場合もありますが、チームとグループは別物です。アイデンティティボールトでグループを定義する場合、共通の特徴を持つ複数ユーザを指定します。ただし、ユーザアプリケーション内でグループにチームの機能が自動的に与えられることはできません。ユーザアプリケー

ションでチーム機能を活用するには、そのグループを示すチームを定義する必要があります。

- ◆ 340 ページのセクション 11.1.1 「プロビジョニングチーム要求について」
- ◆ 341 ページのセクション 11.1.2 「チームを使ったダイレクトレポートの管理」

11.1.1 プロビジョニングチーム要求について

チーム要求オブジェクトは、プロビジョニングチームが実行できる要求を指定します。要求権は、チームマネージャがプロビジョニング要求およびタスクで実行できるアクションを指定します。

チーム定義は、チーム要求オブジェクトと 1 対多の関係です。つまり、各チームには少なくとも 1 つのチーム要求オブジェクトが定義されている必要がありますが、複数定義することもできます。各チーム要求オブジェクトは 1 つのチーム定義だけに関連付けられます。チーム要求オブジェクトを設定するときに、チームマネージャのタスクスコープと許可を設定します。

タスクスコープオプションは、次のタスクを実行するマネージャの能力を定義します。

- ◆ チームメンバーが宛先であるタスク
- ◆ チームメンバーが受信者であるタスク

警告: セキュリティ上の理由から、デフォルトでは受信者のタスクスコープオプションは無効になっています。チームマネージャに、要求の受信者がチームメンバーである場合のタスクの処理権限を与えると、いくつかのセキュリティ上の問題が発生することがあります。まず、マネージャは与えられているトラスティ権利に関係なく、ワークフローの実行中に表示される任意のフォームに含まれているデータを参照できてしまいます。また、許可オプション(後述)によっては、チームマネージャはタスクの引き受けや承認によって、またはタスクの再割り当てによって、承認プロセスを迂回できてしまいます。

許可オプションは、次のようなチームマネージャの能力を定義します。

- ◆ チームメンバーに代わってプロビジョニング要求を開始する。
- ◆ チームメンバーに代わってプロビジョニング要求を撤回する。
- ◆ チームメンバーを他のチームメンバーのプロビジョニング要求の委任先にする。
- ◆ 受信者または宛先であるチームメンバーのタスクを要求する(タスクスコープに基づく)。
- ◆ 受信者または宛先であるチームメンバーのタスクを再割り当てる(タスクスコープに基づく)。

両方のタスクスコープオプションが無効になっている場合、チームマネージャはアクティブな要求を表示および実行できません。したがって、チームマネージャに対して許可オプションを少なくとも 1 つ有効にする必要があります。

注: ユーザアプリケーションは、1 レベルの委任割り当てのみをサポートしています。委任割り当てが複数レベルに反映されることはありません。

プロビジョニング要求に対して定義されたトラスティ権は、チームメンバーに代わって要求を開始するチームマネージャに適用されます。チームマネージャがプロビジョニング要

求定義に対するトラスティ権を持っていない場合、ユーザアプリケーションがプロビジョニング要求を表示しないため、チームマネージャは要求を作成できません。

11.1.2 チームを使ったダイレクトレポートの管理

組織内のマネージャがダイレクトレポートのプロビジョニング環境を管理できるようにするチームを定義できます。適切に定義すれば、単一のチーム定義で、すべてのマネージャがダイレクトレポートの動作を管理できます。つまり、レポートинг関係ごとに別々のチームを定義する必要はありません。

組織内でダイレクトレポートをサポートするチームの基本的な要件を次に示します。

- ◆ チームのメンバーは、マネージャ - 従業員リレーションシップで定義されます。
- ◆ チームのマネージャは、マネージャのみを取得する検索フィルタを使って、サブコンテナを検索するダイナミックグループにより定義されます。

チームを定義したら、すべてのマネージャはナビゲーションメニューのチーム管理アクションを使用できるようになります。これにより、マネージャはダイレクトレポートが実行できるプロビジョニングアクティビティを制御することができます。

ダイレクトレポートを管理するためのチームの定義方法については、[348 ページのセクション 11.2.3 「ダイレクトレポートを管理するチームの作成」](#) を参照してください。

11.2 プロビジョニングチームの管理

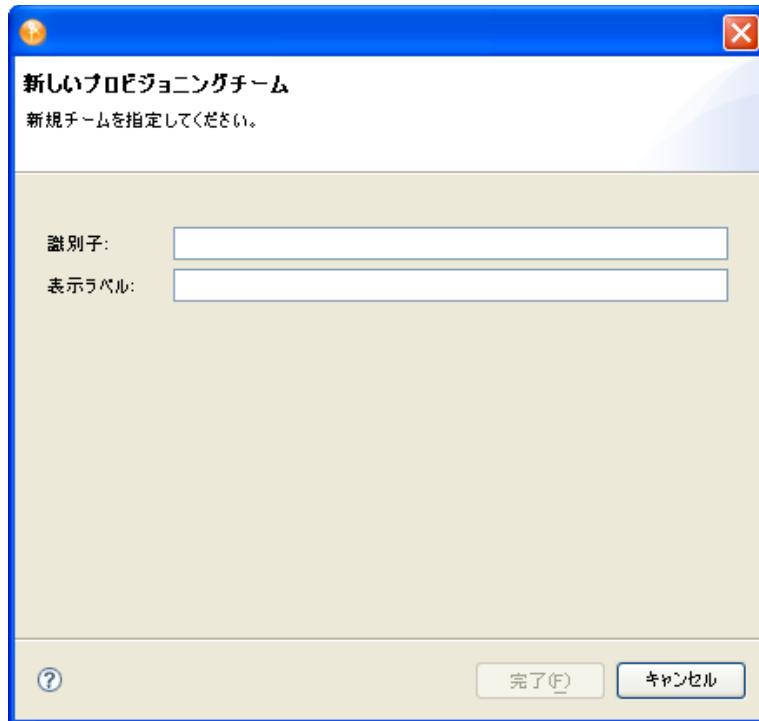
この節では、次のトピックについて説明します。

- ◆ [341 ページのセクション 11.2.1 「プロビジョニングチームの作成」](#)
- ◆ [348 ページのセクション 11.2.2 「プロビジョニングチームの削除」](#)
- ◆ [348 ページのセクション 11.2.3 「ダイレクトレポートを管理するチームの作成」](#)

11.2.1 プロビジョニングチームの作成

新しいプロビジョニングチームを作成する

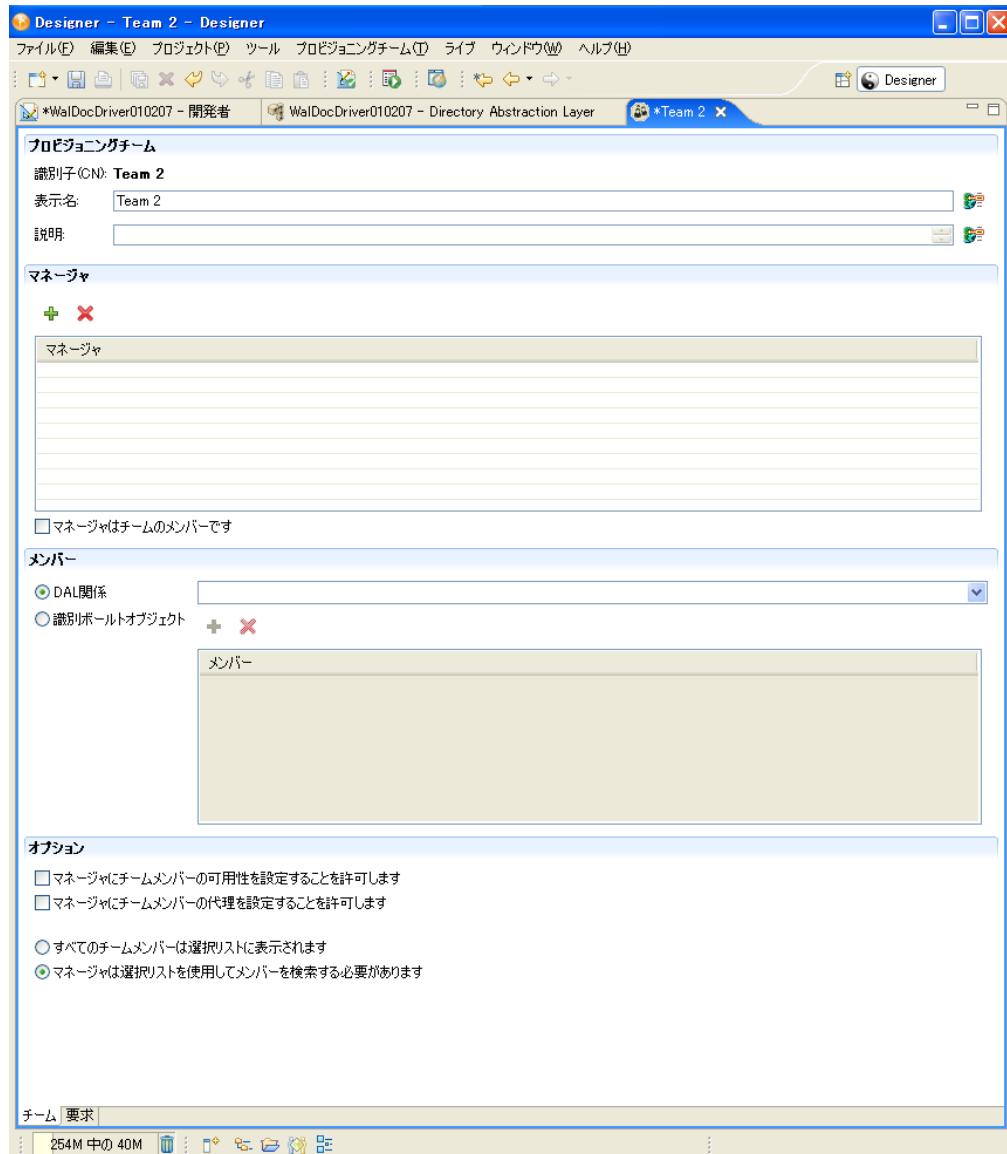
- 1 [プロビジョニング] ビューから [チーム] ノードを右クリックし、[新規] を選択して、● New Teams ● ウィザードを起動します。[● New Provisioning Teams ●] ダイアログボックスが表示されます。



2 次のようにフィールドに値を入力します。

フィールド	説明
<i>Identifier</i>	チームの一般名 (CN) を入力します。
<i>Display Label</i>	プロジェクトチームの名前を入力します。 これが Designer で表示される名前であり、ユーザーアプリケーションのランタイムでも表示されます。ラベルはプロジェクトチームエディタでローカライズできます。

3 [完了] をクリックします。プロジェクトチームエディタの [チーム] パネルが表示されます。



4 説明を入力します。

5 チームマネージャを追加するには、をクリックし、識別ポールトからオブジェクトを選択します。

識別ポールトにアクセスするときに資格情報を求められることがあります。マネージャはユーザでもグループでもかまいません。

6 チームで送信する要求にマネージャをチームメンバーとして含める場合は、「マネージャはチームのメンバーです」を選択します。

7 チームのメンバーを定義するには、次のいずれかを行います。

- ◆ 「DAL 関係」をクリックし、チームのメンバーシップを表す関係を選択します。
- ◆ 「識別ポールトオブジェクト」をクリックします。をクリックし、識別ポールトからメンバーを選択します。

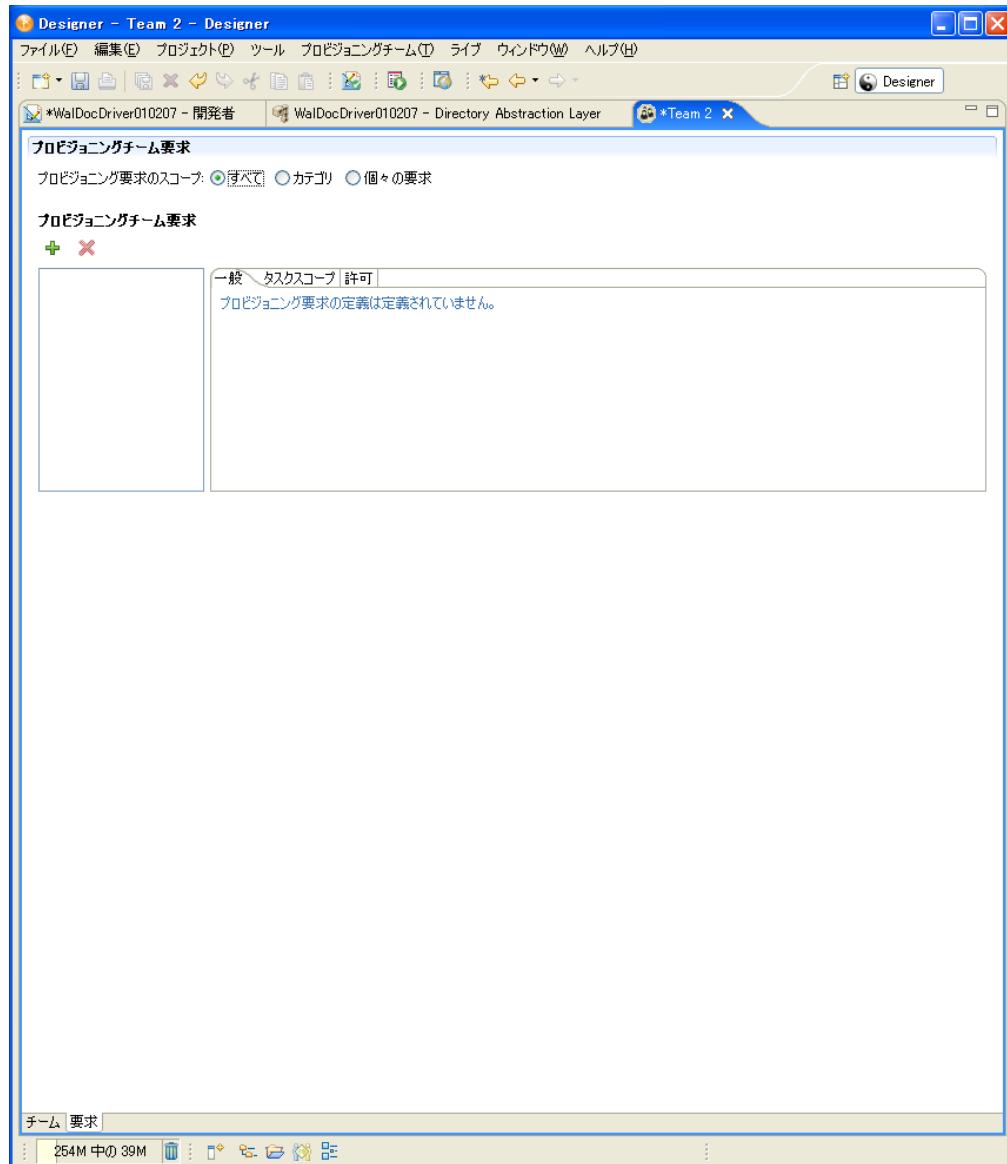
8 パネルの「オプション」セクションで次のように指定します。

フィールド	説明
マネージャにチームメンバーの可用性を設定することを許可します	この設定を有効にした場合、チームマネージャは、ユーザアプリケーションのナビゲーションメニューで [チームの可用性] アクションにアクセスできます。
マネージャにチームメンバーの代理を設定することを許可します	この設定を有効にした場合、チームマネージャは、ユーザアプリケーションのナビゲーションメニューで [チームの代理人割り当て] アクションにアクセスできます。
すべてのチームメンバーは選択リストに表示されます	このオプションを選択した場合、マネージャはドロップダウンリストからチームメンバーを選択できます。チームのメンバーが少ない場合に、このオプションを使用します。
マネージャは選択リストを使用してメンバーを検索する必要があります	このオプションを選択した場合、マネージャはチームメンバーを選択する前に検索を実行する必要があります。チームのメンバー数が多い場合に、このオプションを使用します。

特定のチーム定義によりマネージャが代理やチーム可用性を設定できないようになっている場合でも、マネージャは、管理者または他のチーム（自チームのメンバーが他に属している）のマネージャによって自チームのメンバーに定義された設定を表示できます。ただし、チームマネージャはこれらの設定の編集、詳細表示、新規代理割り当て／チーム可用性設定の作成はできません。

これで、このチームが作業できる要求のタイプを指定できます。

- 9 [要求] をクリックします。[プロビジョニングチーム要求] ページが表示されます。



- 10 このチームの [プロビジョニング要求のスコープ] を選択します。の値は次のとおりです。

選択	説明
すべて	このチーム定義がすべての要求タイプに適用されることを指定します。
カテゴリ	このチーム定義が特定のカテゴリに関連付けられたすべての要求タイプに適用されることを指定します。
個々の要求	このチーム定義が単一の要求タイプに適用されることを指定します。

- 11 をクリックし、新しいプロビジョニングチーム要求オブジェクトの識別子を指定します。

12 [一般] パネルの値を指定します。選択したスコープに応じて異なるオプションが表示されます。

- ◆ カテゴリの場合、[使用可能なカテゴリ] 列から 1 つ以上のカテゴリを選択し、[選択されたカテゴリ] 列に移動します。
- ◆ 個々の要求の場合、ドロップダウンリストから要求を選択します。

各カテゴリまたは個々の要求は、1 つの要求オブジェクトにのみ関連付けることができます。選択すると、作成する次の要求オブジェクトで選択できる値のリストから削除されます。

13 [タスクスコープ] タブで次のように指定します。

フィールド	説明
マネージャに、チームメンバーが宛先である場合にタスクを実行することを許可します	この設定を有効にすると、チームマネージャは [チームタスク] を使って、チームメンバーが宛先になっているタスクに対して作業を行うことができます。この作業には、要求の承認や拒否も含まれます。
マネージャに、チームメンバーが受信者である場合にタスクを実行することを許可します	チームマネージャに対してチームメンバーが宛先になっているタスクに対する作業を許可しない場合、タスクは表示できますが、その詳細を参照したり、何らかの作業を行うことはできません。
マネージャに、チームメンバーが受信者である場合にタスクを実行することを許可します	この設定を有効にすると、チームマネージャは [チームタスク] を使って、チームメンバーが受信者になっているタスクに対して作業を行うことができます。この作業には、要求の承認や拒否も含まれます。
チームマネージャに対してチームメンバーが受信者になっているタスクに対する作業を許可しない場合、タスクは表示できますが、その詳細を参照したり、何らかの作業を行うことはできません。	チームマネージャに対してチームメンバーが受信者になっているタスクに対する作業を許可しない場合、タスクは表示できますが、その詳細を参照したり、何らかの作業を行うことはできません。
注: セキュリティ上の理由から、デフォルトでは受信者のタスクスコープオプションは無効になっています。チームマネージャに、要求の受信者がチームメンバーである場合のタスクの処理権限を与えると、いくつかのセキュリティ上の問題が発生することがあります。まず、マネージャは与えられているトラスティ権利に関係なく、ワークフローの実行中に表示される任意のフォームに含まれているデータを参照できてしまいます。第 2 に、チームマネージャは、許可オプションに応じて、タスクを要求または承認するか、別の誰かに再割り当てすることによって、承認プロセスを回避できます。	

14 [許可] ページで次のように指定します。

フィールド	説明
チームメンバーの代わりに、マネージャがプロジェクトリソース要求を開始できるようにします	この設定を有効にすると、ユーザアプリケーションの「チームリソースの要求」ページに表示されるリソースのリストに、このチームのスコープ内のリソースが含まれます。この設定を無効にすると、これらのリソースは含まれません。
チームメンバーの代わりに、マネージャがプロジェクトリソース要求を取り消せるようにします	この設定を有効にした場合、「チーム要求」ページで、このチームのスコープ内にある要求に対して「[撤回]」ボタンが表示されます。この設定を無効にした場合、「[撤回]」ボタンは表示されません。
マネージャに、チームメンバーを、他のチームメンバーのプロジェクトリソース要求の委任者とすることを許可します	このオプションを有効にした場合、マネージャは「チームの委任割り当て」アクションを使用してあるチームメンバーを他のチームメンバーのプロジェクトリソース要求の委託先(委任)に指名することができます。
マネージャに、受信者または宛先(あるいはその両方)であるチームメンバーのためにタスクを要求することを許可します	このオプションを無効にした場合、マネージャは管理者またはユーザが所属する他のチームのマネージャが定義した委任設定を参照することができます。ただし、チームマネージャはこれらの設定の編集または削除、詳細表示、新規委任の割り当てはできません。
マネージャに、受信者または宛先(あるいはその両方)であるチームメンバーのためにタスクを再割り当てすることを許可します	この設定を有効にした場合、「チームのタスク」ページで、このチームのスコープ内にある要求に対して「[要求]」ボタンが有効になります。この設定を無効にした場合、「[要求]」ボタンは淡色表示になります。
	この設定を有効にした場合、「チームのタスク」ページで、このチームのスコープ内にある要求に対して「[再割り当て]」ボタンが有効になります。この設定を無効にした場合、「[再割り当て]」ボタンは無効になります。

15 [保存] をクリックします。

ユーザアプリケーションで利用できるようにプロジェクトリソースチームを展開する必要があります。参考先 [26 ページのセクション 2.7 「プロジェクトリソースのオブジェクトの展開」](#)。プロジェクトリソースチームを展開すると、ユーザアプリケーションドライバ AppConfig チームノードに次の 2 つのオブジェクトが作成されます。

- ◆ `srvprvTeam`: プロジェクトリソースチームオブジェクトが含まれています。
- ◆ `srvprvTeamRequest`: 要求オブジェクトが含まれています。

プロジェクトリソースチームのユーザアプリケーションドライバにはオブジェクトが 2 つ格納されているので、比較およびインポート操作は、タイプが異なるプロジェクトリソースオブジェクトの操作とは違います。両方のオブジェクトを比較またはインポートするには、個々の要素ではなく、チームノードで比較またはインポートを行う必要があります。個々の要素で比較またはインポートを行った場合、`srvprvTeam` オブジェクトだけがインポートまたは比較されます。

11.2.2 プロビジョニングチームの削除

プロビジョニングチームオブジェクトを [プロビジョニング] ビューから削除するには、チームを選択し、右クリックして、[削除] を選択します。削除を確認するダイアログボックスで、オブジェクトをローカルでのみ削除するか、親オブジェクトを次回展開するときに識別ポールトから削除するかを指定できます。Designer でチーム要求オブジェクトを削除した場合、チーム要求オブジェクトは、チームを展開するときに識別ポールトから削除されます。

11.2.3 ダイレクトレポートを管理するチームの作成

- 1 iManager で、「Managers」というダイナミックグループを作成します。
 - 1a [検索スコープ] に [サブコンテナを検索] を設定します。
 - 1b [検索フィルタ] に [(&(isManager=TRUE))] を設定します。
ダイナミックグループの作成の詳細については、『Novell Identity Manager: 管理ガイド』を参照してください。
- 2 Designer で、新しいプロビジョニングチームを作成し、「DirectReports」という名前を付けます。
 - 2a チームマネージャを指定するには、前に作成した Managers ダイナミックグループを選択します。
 - 2b チームメンバーを指定するには、[マネージャ - 従業員] 関係を選択します。
 - 2c チームオプションを定義する
 - ◆ [チームメンバーのチーム可用性をマネージャが設定できるようにします] を選択します。
 - ◆ [チームメンバーのプロキシを設定することをマネージャに許可します] を選択します。
 - ◆ [すべてのチームメンバーが選択リストに表示されます] を選択します。
- 3 [要求] タブを選択して、プロビジョニングチーム要求オブジェクトを定義します。
- 4 [プロビジョニング要求のスコープ] を [すべてのプロビジョニング要求] に設定し、新しいプロビジョニングチーム要求の名前を「DirectReportsTeamRequestRights」にします。
- 5 [タスクスコープ] タブをクリックし、タスクスコープオプションを次のように指定します。
 - ◆ [マネージャに、そのチームメンバーが宛先となっているタスクの作業を許可します] を選択します。
 - ◆ [マネージャに、そのチームメンバーが受信者となっているタスクの作業を許可します] の選択を解除します。
- 6 [許可] タブをクリックし、許可オプションを次のように指定します。
 - ◆ [チームメンバーの代わりに、マネージャがプロビジョニング要求を開始できるようにします] を選択します。
 - ◆ [チームメンバーの代わりに、マネージャがプロビジョニング要求を取り消せるようにします] を選択します。
 - ◆ [マネージャに、チームメンバーを、他のチームメンバーのプロビジョニング要求の委任者とすることを許可します] を選択します。

- ◆ [マネージャが、タスクスコープに基づいた受信者、宛先、またはその両方となっているチームメンバーのタスクを要求できるようにします] を選択します。
- ◆ [マネージャが、タスクスコープに基づいた受信者、宛先、またはその両方となっているチームメンバーにタスクを再割り当てできるようにします] を選択します。

7 チームを保存し、展開します。

詳細については、[26 ページのセクション 2.7 「プロビジョニングオブジェクトの展開」](#) を参照してください。