



## Script Editor のステップの参照情報

---

この章では、Auto Attendant (AA; 自動受付) スクリプトおよび IVR (interactive voice response; 自動音声応答) スクリプトの作成で利用できるすべてのステップの一覧を示します。付記されているとおり、一部のステップは IVR スクリプト専用です。これらのステップはすべて、[Palette] ペインからアクセスできます ([Palette] ペイン) (P.12) を参照してください。この章は、次の各項で構成されています。

- 「Call Contact ステップ」 (P.67)
- 「Contact ステップ (IVR のみ)」 (P.73)
- 「Database ステップ (IVR のみ)」 (P.77)
- 「Document ステップ (IVR のみ)」 (P.90)
- 「eMail Contact ステップ (IVR のみ)」 (P.96)
- 「Fax Contact ステップ (IVR のみ)」 (P.100)
- 「General ステップ」 (P.104)
- 「HTTP Contact ステップ (IVR のみ)」 (P.121)
- 「Media ステップ」 (P.131)
- 「Prompt ステップ」 (P.169)
- 「Session ステップ」 (P.180)
- 「User ステップ」 (P.182)
- 「VoiceMail ステップ」 (P.185)

### Call Contact ステップ

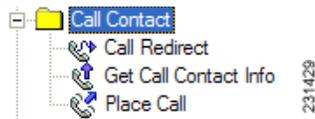
**Call Contact** パレットのステップは、通話を管理する手段を提供します。

**Call Contact** パレットには、次のステップがあります。

- 「Call Redirect」 (P.68)
- 「Get Call Contact Info」 (P.69)
- 「Place Call (IVR のみ)」 (P.71)

図 49 に、**Call Contact** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 49 Call Contact パレットのステップ



## Call Redirect

**Call Redirect** ステップは、通話を他の内線に転送する場合に使用します。

**Call Redirect** ステップは、目的の内線が指定された後で通話を転送するアプリケーションでよく使用されます。

**Call Redirect** ステップは、4 つの出力ブランチを生成します。

- **Successful** : 通話は、指定された内線呼び出しています。
- **Busy** : 指定された内線が使用中で、通話を転送できません。
- **Invalid** : 指定された内線は存在しません。
- **Unsuccessful** : 転送ステップが内部で失敗しました。

4 つの各ブランチの後で、転送された通話の結果を処理するスクリプト ステップを設定します。

図 50 に、**Call Redirect** ステップのカスタマイザ ウィンドウを示します。

図 50 [Call Redirect] カスタマイザ ウィンドウ



表 7 で、[Call Redirect] カスタマイザ ウィンドウのフィールドについて説明します。

表 7 [Call Redirect] カスタマイザ ウィンドウのフィールド

フィールド	説明
Call Contact	転送するコンタクト。他のコンタクトが定義されていない場合、デフォルトは Triggering Contact です。選択した <b>Call Contact</b> の名前と extension 変数が、[Design] ペインの <b>Call Redirect</b> ステップアイコンの横に表示されます。

表 7 [Call Redirect] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
Extension	通話を転送する先の内線を保持する変数。サポートされる内線については、表 8 を参照してください。
Reset CTI Called Address	[Yes] の場合、通話の元の宛先が、転送先の宛先にリセットされます。 [No] の場合、 <b>Call Redirect</b> ステップが実行された後も、通話の元の宛先が維持されます。通話に関連付けられた情報は、このルート ポイントまたは CTI ポートが通話に関係することを示しません。 デフォルトは [Yes] です。このフィールドは、転送される宛先の要件に応じて設定します。

表 8 で、**Call Redirect** ステップがサポートする内線について説明します。

表 8 Call Redirect : サポートする内線

内線	説明
「#」または「*」で始まる内線	ネットワークのテークバックをトリガーする内線で、指定された文字列をそのとおりにアウトパルスした先に転送します。5 秒以内に切断イベントが発生した場合に、転送は成功します。 <b>(注)</b> 文字列でカンマ (,) を使用すると、次の数字をアウトパルスする前に 1 秒間の休止を挿入できます。
「.wav」で終了する内線	転送のネットワーク通知タイプをトリガーする内線。呼び出しトーンをシミュレーションし、指定された .wav ファイルを 4 回再生し、最後にファーストビジー トーンをシミュレートします。 発信者が電話を切るか、ファーストビジー トーンの末尾まで到達して通話が接続解除された場合に、転送は成功します。
「PROBLEMS」と等しい内線	転送のネットワーク通知タイプをトリガーし、システムの問題を通知する内線。 発信者が電話を切るか、音声の末尾まで到達した場合に、転送は成功します。この通話は、転送ではなく接続解除された通話としてレポートされます。
「BUSY」、「RNA <sup>1</sup> 」、「FASTBUSY」、または「DIALTONE」と等しい内線	通話が接続解除される前に、指定された音声処理が生成されます。 発信者が電話を切るか、音声の末尾まで到達した場合に、転送は成功します。この通話は、転送ではなく接続解除された通話としてレポートされます。

1. RNA = Ring No Answer (応答なし)

## Get Call Contact Info

**Get Call Contact Info** ステップは、通話固有の情報にアクセスし、指定した変数に値を格納する場合に使用します。

このステップを使用して、発信元、およびセッションに関連付けられたその他のプロパティに応じて、通話をさまざまな方法で処理できます。たとえば、このステップを **Call Redirect** ステップと共に使用して、通話を別の内線に転送できます。また、このステップを **Play Prompt** ステップと共に使用して、音声プロンプトを再生できます。

図 51 に、**Get Call Contact Info** ステップのカスタマイザ ウィンドウを示します。

図 51 [Get Call Contact Info] カスタマイザ ウィンドウ

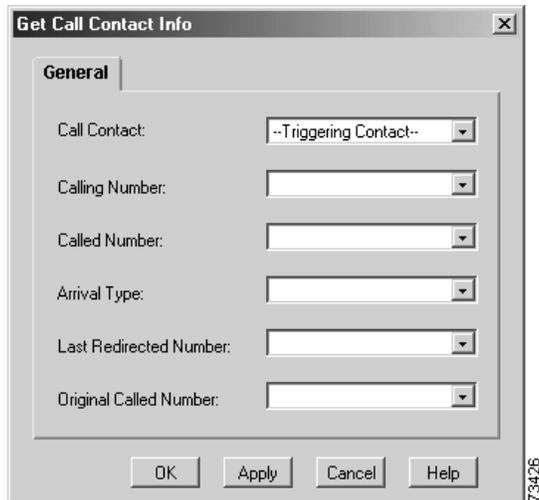


表 9 で、[Get Call Contact Info] カスタマイザ ウィンドウのフィールドについて説明します。

表 9 [Get Call Contact Info] カスタマイザ ウィンドウのフィールド

フィールド	説明
<b>Call Contact</b>	情報を取得するコンタクト。他のコンタクトが定義されていない場合、デフォルトは <b>Triggering Contact</b> です。選択した <b>Call Contact</b> 変数の名前が、[Design] ペインの <b>Get Call Contact Info</b> ステップアイコンの横に表示されます。
<b>Calling Number</b>	通話の発信者の番号を格納する変数。
<b>Called Number</b>	発信者が呼び出した番号を格納する変数。
<b>Arrival Type</b>	通話の到達タイプを保持する変数。サポートされる到達タイプについては、表 10 を参照してください。
<b>Last Redirected Number</b>	最後に通話のリダイレクトまたは転送が行われた元の番号。この番号は、現在の番号の直前に通話が発生した番号です。
<b>Original Called Number</b>	発信者の視点から見た、呼び出された番号。

表 10 で、**Get Call Contact Info** ステップの到達タイプについて説明します。

表 10 Get Call Contact Info : 到達タイプ

(イベント) 到達タイプ	説明
UNKNOWN	通話の到達方法をシステムが判別できません。
DIRECT	発信者から直接到達した着信通話。
REDIRECT	このアプリケーションに転送された着信通話。
FORWARD_ALL	元の宛先から転送された着信通話。
FORWARD_BUSY	元の内線が使用中だったために、現在のアプリケーションに転送された通話。

表 10 Get Call Contact Info : 到達タイプ (続き)

(イベント) 到達タイプ	説明
FORWARD_NO_ANSWER	元の内線の呼び出し回数が最大数を越えたために、現在のアプリケーションに転送された通話。
TRANSFER	Transfer 機能の一部としてローカルに発信された着信通話。
OUTBOUND	アプリケーションが作成した発信通話の結果である通話。
TIME_OF_DAY	time-of-day 転送の結果である通話。
DO_NOT_DISTURB	do-not-disturb 転送の結果である通話。
FOLLOW_ME	follow-me 転送の結果である通話。
OUT_OF_SERVICE	元の着信側が使用不能だったために受信した通話。
AWAY	元の着信側が不在だったために受信した通話。

## Place Call (IVR のみ)

**Place Call** ステップは、通話を発信する場合に使用します。

たとえば、このステップを使用すると、事前に設定された時間帯に患者に通話を発信して、相互のやり取りの現状を通知できます。ユーザ、ボイスメール システム、留守番電話、または FAX が応答した場合、通知呼び出しは成功したと見なされます。

**Place Call** ステップは、6 つの出力ブランチを生成します。

- **Successful** : 通話は成功しました。
- **NoAnswer** : 通話を試みましたが、応答なし (RNA) タイムアウト制限に到達しました。
- **Busy** : 通話を試みましたが、回線が使用中でした。
- **Invalid** : 通話を試みましたが、内線が無効でした。
- **NoResource** : 通話を発信するためのリソースが使用できなかったため、通話を試みませんでした。
- **Unsuccessful** : 通話を試みましたが、内部システム エラーが原因で失敗しました。

 [52](#) に、**Place Call** ステップのカスタマイザ ウィンドウを示します。

図 52 [Place Call] カスタマイザ ウィンドウ



表 11 で、[Place Call] カスタマイザ ウィンドウのフィールドについて説明します。



(注)

Primary CallControlGroupId および Secondary CallControlGroupId の値が正しく設定されていない場合、表 11 で説明するとおり、発信通話の確立に失敗することがあります。

表 11 [Accept] カスタマイザ ウィンドウのフィールド

フィールド	説明
Destination Telephone No	発信通話の宛先番号を格納する変数。
RNA Timeout (sec)	応答なし (RNA) 条件によってスクリプトがリモート側の応答を待機するのを停止し、出力ブランチを通じて「NoAnswer」を返すまでの時間 (秒)。
Primary CallControlGroupId	発信通話に使用するプライマリ コール制御グループの ID。この値は、Cisco Unified Communications Manager Express (旧 Cisco Unified CallManager Express) と連動する場合は 0、Cisco Unified Communications Manager (旧 Cisco Unified CallManager) と連動する場合は 1 に設定する必要があります。
Secondary CallControlGroupId	発信通話に使用するセカンダリ コール制御グループの ID。このフィールドは、Cisco Unified Communications Manager Express と連動する場合は使用されません。この値は、Cisco Unified Communications Manager と連動する場合に 0 に設定する必要があります。
Source Telephone Number	(将来的に使用) 着信通話の発信元番号を格納する変数。

表 11 [Accept] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
Use Media for this Call	[Yes] の場合は、通話の一部としてメディア チャネルも確立されます。[No] の場合、この通話に対してプロンプトは再生されません。
Call Contact	ステップが成功したときに作成される通話コンタクトを格納する変数。

## Contact ステップ (IVR のみ)

**Contact** パレットのステップは、コンタクトを制御する手段を提供します。**Contact** ステップの詳細については、『*Cisco Unity Express 7.0 IVR CLI Administrator Guide*』を参照してください。

コンタクトは、通話呼など、リモート ユーザとの接続の 1 つの形態を表します。スクリプトはコンタクトを使用して、システム全体を通じて接続を追跡します。コンタクトは、接続が行われたときに確立します。コンタクトは、スクリプトが通話呼を転送または接続解除するなど、接続が終了するまで継続します。

コンタクトで動作する各ステップを、暗黙のコンタクト(「-- Triggering Contact --」のデフォルトを選択)を受け入れるか、このコンタクトの識別子を保持する変数を使用するように設定します。**Contact** パレットの **Set Contact Info** ステップは、レポートのために必要な **Handled** のマークをコンタクトに付ける場合に使用します。

**Contact** パレットには、次のステップがあります。

- 「Accept」 (P.73)
- 「Get Contact Info」 (P.74)
- 「Set Contact Info」 (P.76)
- 「Terminate」 (P.77)

図 53 に、**Contact** パレットのステップを Cisco Unity Express ScriptEditor の [Palette] ペインに表示されるとおりに示します。

図 53 Contact パレットのステップ



## Accept

**Accept** ステップは、特定のコンタクトを受け付ける場合に使用します。

着信コンタクトでトリガーされる Cisco Unity Express スクリプトでは、**Start** ステップの後、通常、**Accept** ステップが最初のステップになります。

スクリプトがこのステップに到達するまで、発信者は呼び出し音を聞きます。

図 54 に、**Accept** ステップのカスタマイザ ウィンドウを示します。

図 54 [Accept] カスタマイザ ウィンドウ



表 12 で、[Accept] カスタマイザ ウィンドウのフィールドについて説明します。

表 12 [Accept] カスタマイザ ウィンドウのフィールド

フィールド	説明
Contact	Contact 変数。他のコンタクトが定義されていない場合、デフォルトは Triggering Contact です。 <b>Contact</b> 変数の名前が、[Design] ペインの <b>Accept</b> ステップ アイコンの横に表示されます。

## Get Contact Info

**Get Contact Info** ステップは、コンタクトから情報を抽出し、スクリプトの後続のステップでこのコンタクトの情報を利用できるように、スクリプト変数に格納する場合に使用します。

図 55 に、**Get Contact Info** ステップのカスタマイザ ウィンドウを示します。

図 55 [Get Contact Info] カスタマイザ ウィンドウ



表 13 で、[Get Contact Info] カスタマイザ ウィンドウのフィールドについて説明します。

表 13 [Get Contact Info] カスタマイザ ウィンドウのフィールド

フィールド	説明
Contact	情報を取得する Contact 変数。他のコンタクトが定義されていない場合、デフォルトは Triggering Contact です。選択した Contact 変数の名前が、[Design] ペインの <b>Get Contact Info</b> ステップ アイコンの横に表示されます。
Attribute/Variable	コンタクト情報の種類に関する属性と変数。
Set...	[Get Contact Type] ダイアログボックスを表示します。選択した変数の名前が、[Get Contact Info] カスタマイザ ウィンドウで選択した属性の横の [Variable] カラムに表示されます。
Clear	選択した属性から変数をクリアします。

表 14 では、**Get Contact Info** ステップによって、スクリプトの他のステップで利用できるようになる情報について説明します。

表 14 Get Contact Info の属性

属性	説明
Type	コンタクトの種類を表す String。Cisco Unity Express では、種類は call です。
Language	このオプションは、将来使用します。
ASR Supported	このオプションは、将来使用します。
Is Active	通話がアクティブなままかどうかを示す Boolean 値。
Is Aborting	通話が打ち切られたかどうかを示す Boolean 値。
Session Handled	通話が以前に処理済みとしてマークされたかどうかを示す Boolean 値。
Identifer	すべてのコンタクトの中で一意であることが保証される、システムが割り当てたコンタクトの識別情報を含む Integer 値。
Implementation ID	実装固有のコンタクトの識別情報を含む String 値。この値は、コンタクトの種類ごとに一意です。Cisco JTAPI 通話コンタクトでは、この値は、Cisco Unified Communications Manager ソフトウェアで取得されるグローバル通話 ID と同じです。
Sequence Number	コンタクトがセッションと関連付けられている場合にシステムが割り当てるコンタクトの順序番号を含む Integer 値。コンタクトがセッションと関連付けられていない場合、この値は -1 です。新しいコンタクトがセッションに関連付けられると、システムがこの値を 1 つずつ増やします。
Session	コンタクトに関連付けられたセッション オブジェクト。見つからない場合は Null です。

[Get Contact Type] ダイアログボックスを [図 56](#) に示します。

図 56 [Get Contact Type] ダイアログボックス



## Set Contact Info

**Set Contact Info** ステップは、コンタクトに関連付けられているコンテキスト情報を変更する場合に使用します。

**Set Contact Info** ステップは、スクリプトの **Redirect** ステップの後で、コンタクトに **Handled** のマークを付けるためによく使用します。

コンタクトは、アクティブな場合にだけ **Handled** のマークを付けることができます。コンタクトが非アクティブになった場合（正常に転送された後など）、スクリプトはコンタクトに **Handled** のマークを付けるために最大 5 秒待機します。そのようにしないと、マークがレポートで有効にならないためです。



(注) コンタクトに **未処理** のマークを付けることはできません。コンタクトが **Handled** としてレポートされると、それ以降、常にこの状態でレポートされます。

図 57 に、**Set Contact Info** ステップのカスタマイザ ウィンドウを示します。

図 57 [Set Contact Info] カスタマイザ ウィンドウ



表 15 で、[Set Contact Info] カスタマイザ ウィンドウのフィールドについて説明します。

表 15 [Set Contact Info] カスタマイザ ウィンドウのフィールド

フィールド	説明
Contact	情報を設定する Contact 変数。他のコンタクトが定義されていない場合、デフォルトは <b>Triggering Contact</b> です。選択した <b>Contact</b> 変数の名前が、[Design] ペインの <b>Set Contact Info</b> ステップ アイコンの横に表示されます。
Attribute/Value	コンタクト情報の種類に関する属性と値。

表 15 [Set Contact Info] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
Set...	Contact 変数に Handled 属性を設定します。値として X が表示されます。
Clear	Contact 変数の Handled 属性をクリアします。

表 16 で、**Set Contact Info** ステップのカスタマイザ ウィンドウで提供される属性情報について説明します。

表 16 Set Contact Info の属性

属性	説明
Language	このオプションは、将来使用します。
Handled	コンタクトの最終結果。レポート目的で重要です。
Session	このオプションは、将来使用します。

## Terminate

**Terminate** ステップは、通話を接続解除する場合に使用します。

図 58 に、**Terminate** ステップのカスタマイザ ウィンドウを示します。

図 58 [Terminate] カスタマイザ ウィンドウ



表 17 で、[Terminate] カスタマイザ ウィンドウのフィールドについて説明します。

表 17 [Terminate] カスタマイザ ウィンドウのフィールド

フィールド	説明
Contact	終了する Contact 変数。他のコンタクトが定義されていない場合、デフォルトは Triggering Contact です。選択した <b>Contact</b> 変数の名前が、[Design] ペインの <b>Terminate</b> ステップ アイコンの横に表示されます。

## Database ステップ (IVR のみ)

**Database** パレットのステップは、データベースのテーブルとビューからデータを読み取ったり、そこにデータを書き込んだりするための手段を提供します。データベース設定の詳細については、『[Cisco Unity Express 7.0 IVR CLI Administrator Guide](#)』を参照してください。

**Database** ステップを使用してスクリプトを作成する前に、次の前提条件を満たしている必要があります。

- Script Editor ソフトウェアが Cisco Unity Express ルータに接続できない場合でも、Database ステップのフィールドにスキーマ情報を手動で入力することによって、スクリプトを作成できる。  
たとえば、テーブル名またはフィールド名が更新されるのは、Script Editor ソフトウェアと Cisco Unity Express ルータの間に接続があり、Cisco Unity Express ルータが、定義されているデータベース プロファイル設定を使用して外部データベースに接続できる場合だけです。
- Cisco Unity Express Script Editor は、Java RMI プロトコルを使用し、ネットワークを介して Cisco Unity Express ルータに到達可能である必要がある。
- Cisco Unity Express ルータの IP アドレスまたはドメイン ネーム サーバ (DNS) ホスト名を定義する必要がある。モジュールのネットワーク情報は、Cisco Unity Express Script Editor から [Tools]、[Options] の順にクリックして指定できます。
- Cisco Unity Express のグラフィカルユーザ インターフェイス (GUI) またはコマンドライン インターフェイス (CLI) を使用して、データベース プロファイルを作成する必要がある。Database ステップは、データベース プロファイルなしで記述されるスクリプトには使用できません。データベース プロファイルが作成されていない場合は、エラー メッセージが表示されます。

**Database** ステップには関連付けられたリソース名があり、そのリソース名は複数ステップのリンクするために使用されます。たとえば、クエリーを実行するために、**DB Read** ステップを *myquery* というリソース名とともに発行すると、*myquery* は、そのクエリーの結果を取得するために **DB Get** ステップに指定されます。リソース名 *myquery* は、クエリーによって要求されたときに **DB Release** ステップがデータベース リソースを解放する前にも必要となります。

クエリーを実行するためにリソース名が使用され、別の **DB Read** ステップが同じリソース名を使用するように設定されている場合、2 番目の **DB Read** ステップによって前のクエリーの結果は上書きされます。

リソース名は、**DB Read** ステップおよび **DB Write** ステップに関連付けられたテキスト フィールドに入力します。リソース名ごとに 1 つのデータベース接続を使用します。各プロファイルの許容接続数は制限されています。ステップの実行中に許容接続数を超えた場合、Structured Query Language (SQL; 構造化照会言語) 例外が発行されます。接続数がプロファイルの最大値を超えないようにするには、**DB Read** ステップまたは **DB Write** ステップの実行後に接続リソースが開放されるように、**DB Release** ステップを使用する必要があります。スクリプトの実行を終了した場合も、データベース リソースが解放されます。

**DB Read** ステップおよび **DB Get** ステップは、**Database** スクリプトによって、順次実行されます。**DB Read** ステップは、データベース クエリーを実行し、返されたデータをエンタープライズ データベース サブシステム (EDBS) の内部に格納します。**DB Get** ステップは、**DB Read** ステップによって取得されたデータを使用します。**DB Read** ステップの実行後に実際のデータベースの内容が変更された場合、**DB Get** ステップはそれらの変更を反映しないため、別の **DB Read** ステップを実行して、更新されたデータをデータベースから取得する必要があります。

**Database** パレットには、次のステップがあります。

- 「DB Read」 (P.79)
- 「DB Get」 (P.82)
- 「DB Write」 (P.85)
- 「DB Release」 (P.89)

図 59 に、**Database** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 59 Database パレットのステップ



## DB Read

**DB Read** ステップは、データベース選択クエリーを実行する場合に使用します。**DB Read** ステップを使用すると、選択クエリーがどのカラムに値を返すかを指定したり、**DB Get** ステップを使用して結果のデータを格納する変数を指定したりできます（「**DB Get**」(P.82)を参照）。**DB Read** ステップでは、クエリー元として複数のデータベースのテーブルまたはビューを指定できます。

**DB Read** ステップには、次の制限があります。

- **DB Read** ステップは、**select count(\*)** および **select min(\*)** など、**count(\*)** 形式または **min(\*)** 形式の SQL 文のバリエーションをサポートしません。**DB Read** ステップはデータベース ビューをサポートしているため、これらの SQL クエリーを使用するデータベース内にビューを作成できます。
- **DB Read** ステップは、SQL 文の *where* 句のセマンティック チェックをサポートしていません。
- **DB Read** ステップは、異なるテーブルから同じ名前のカラムを取得する *join* クエリーをサポートしていません。
- **DB Read** ステップは、カラム名のエイリアスの使用をサポートしていません。たとえば、**select a.x, b.x from a,b** と **select a.x as y from a** を作成することはできません。このようなクエリーが入力されていないことを確認するためのチェックは行われません。**DB Read** ステップで入力できるのは、**select** SQL 文のみです。
- **update** または **insert** SQL 文はサポートしていません。

**DB Read** ステップが返された結果が不要になったら、**DB Release** ステップを使用して結果をクリアし、データベース リソースを解放します。

**DB Read** ステップは、3 つの出力ブランチを生成します。

- **Successful** : エラーが発生しなかったことを示します。データが見つからなかったか、1 行以上のデータが見つかりました。
- **Connection Not Available** : クエリーの実行に使用できるデータベース接続がなかったことを示します。
- **SQL Error** : SQL、タイムアウト、または発生したその他の例外に関連するエラーが発生したことを示します。

[DB Read] カスタマイザ ウィンドウには 3 つのタブがあります。

- 「[General] タブ」(P.79)
- 「[SQL] タブ」(P.80)
- 「[Comments] タブ」(P.82)

## [General] タブ

[DB Read] カスタマイザ ウィンドウの [General] タブ (図 60) は、データベース リソース名、データベース プロファイル名、タイムアウト値 (秒) を設定する場合に使用します。

図 60 [DB Read] カスタマイザ ウィンドウ : [General] タブ

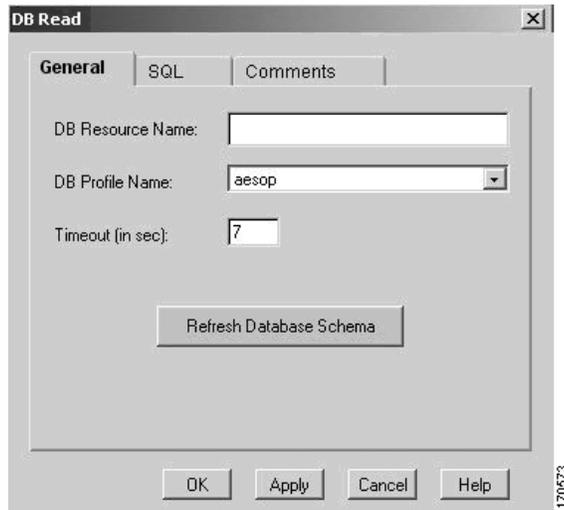


表 18 で、[General] タブのフィールドについて説明します。

表 18 [DB Read] カスタマイザ ウィンドウのフィールド : [General] タブ

プロパティ	説明
<b>DB Resource Name</b>	このデータベース クエリーを識別する名前。DB Read ステップごとに一意の DB Resource Name が必要です。DB Read ステップの Resource Name を DB Write ステップで使用しないでください。一連の <b>DB Get</b> ステップと <b>DB Release</b> ステップでは、同じ DB Resource Name を使用します。
<b>DB Profile Name</b>	アクセスするデータベース プロファイルの名前。このリストには、Cisco Unity Express モジュールに設定されているデータベース プロファイルが表示されます。
<b>Timeout (in sec)</b>	データベースが使用できない場合に、スクリプトが無期限に待機するのを防ぐための間隔。タイムアウト間隔の値が 0 の場合は、無期限の待機状態に陥ります。  無期限の待機状態になると、スクリプトはリモート切断などのイベントに応答できなくなります。
<b>Refresh Database Schema</b>	選択された Cisco Unity Express データベースで定義されているフィールドで、Editor を更新します。

## [SQL] タブ

[DB Read] カスタマイザ ウィンドウの [SQL] タブ (図 61) は、SQL コマンドを設定する場合に使用します。

図 61 [DB Read] カスタマイザ ウィンドウ : [SQL] タブ

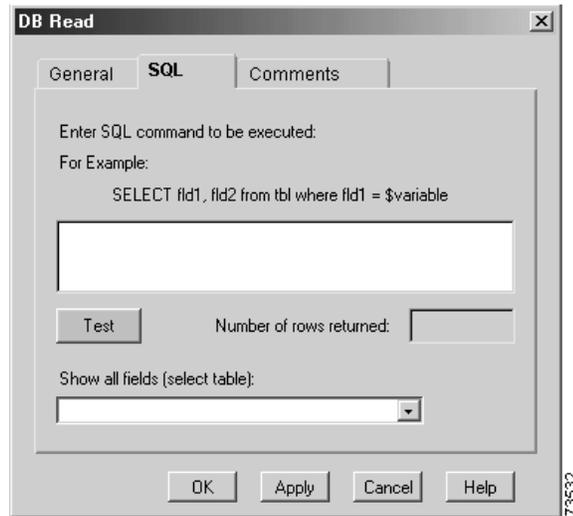


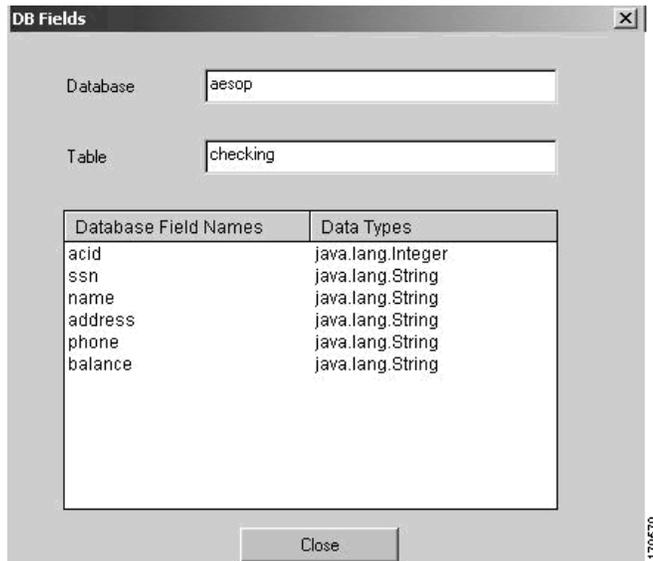
表 19 で、[SQL] タブのフィールドについて説明します。

表 19 [DB Read] カスタマイザ ウィンドウのフィールド : [SQL] タブ

プロパティ	説明
Enter SQL Command to be Executed	実行する SQL コマンド。
Number of Rows Returned	[Test] ボタンをクリックしたときに返される行数。
Show All Fields (Select Table)	このデータベースの特定のテーブルで定義されているフィールド。指定のプロファイルに対応するすべてのテーブルが表示されます。テーブル名を選択すると、[DB Field] ダイアログボックス (図 62) が開き、このテーブルのフィールドとそのデータ型が表示されます。

[Show All Fields] ドロップダウン リストで特定のテーブルを選択すると、[DB Field] ダイアログボックス (図 62) が開きます。

図 62 [DB Field] ダイアログボックス



## [Comments] タブ

[DB Read] カスタマイザ ウィンドウの [Comments] タブ (図 63) は、テキスト コメントを入力する場合に使用します。

図 63 [DB Read] カスタマイザ ウィンドウ : [Comments] タブ



## DB Get

**DB Get** ステップは、**DB Read** ステップから結果を取得する場合に使用します。

**DB Get** ステップの内部（または設定中に使用されるスクリプト変数）では、選択したテーブルまたはビューのフィールドに変数がマッピングされます。**DB Get** ステップの外部（またはスクリプト内の任意の場所で使用可能なスクリプト変数）では、スクリプト変数名を使用してデータにアクセスできます。結果を使用が終了したら、**DB Release** ステップを使用してクエリー リソースを解放します。

スクリプトが **DB Get** ステップを実行するたびに、**DB Read** ステップが返す結果を 1 行ずつ取得し、それらに変数を割り当てます。結果セット内の次の行に移動するには、**DB Get** ステップを再度実行する必要があります。結果のすべての行を読み取ると、**DB Get** は **No Data** ブランチに進みます。



(注) **DB Get** ステップは、対応する **DB Read** ステップの **Successful** ブランチの下に配置することを強く推奨します。対応する **DB Get** ステップを試行したときに **DB Read** ステップが失敗した場合、スクリプトは中断されます。

**DB Get** ステップは、3 つの出カブランチを生成します。

- **Successful** : エラーは発生しませんでした。データが見つかりました。
- **No Data** : クエリーは **DB Read** ステップで何も返しませんでした。結果セットのすべての行を読み取った場合にも、このブランチに進みます。
- **SQL Error** : SQL、タイムアウト、または検出された例外が含まれます。

[DB Get] カスタマイザ ウィンドウには、2 つのタブがあります。

- 「[General] タブ」 (P.79)
- 「[Field Selection] タブ」 (P.84)

## [General] タブ

[DB Get] カスタマイザ ウィンドウの [General] タブ (図 64) は、**DB Read** ステップの結果からデータベース リソースおよびデータベース プロファイルを選択する場合に使用します。

図 64 [DB Get] カスタマイザ ウィンドウ : [General] タブ

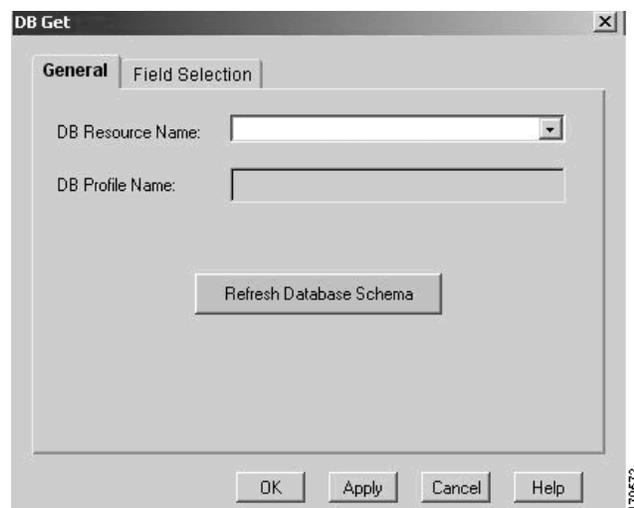


表 18 で、[General] タブのフィールドについて説明します。

表 20 [DB Get] カスタマイザ ウィンドウのフィールド : [General] タブ

プロパティ	説明
DB Resource Name	<b>DB Read</b> ステップによって定義されるクエリー。スクリプトが同時に複数の <b>DB Read</b> ステップをオープンした場合は、特定のクエリーを選択できます。
DB Profile Name	DB Resource Name と関連付けられたデータベース プロファイルの名前。DB Resource Name を選択すると、このフィールドには自動的にプロファイル名が入力されます。
Refresh Database Schema	選択された Cisco Unity Express データベースで定義されているフィールドで、Editor を更新します。

## [Field Selection] タブ

[DB Get] カスタマイザ ウィンドウの [Field Selection] タブ (図 65) は、**DB Read** ステップの結果からテーブル、フィールド、データ型、およびローカル変数を選択する場合に使用します。

図 65 [DB Get] カスタマイザ ウィンドウ : [Field Selection] タブ

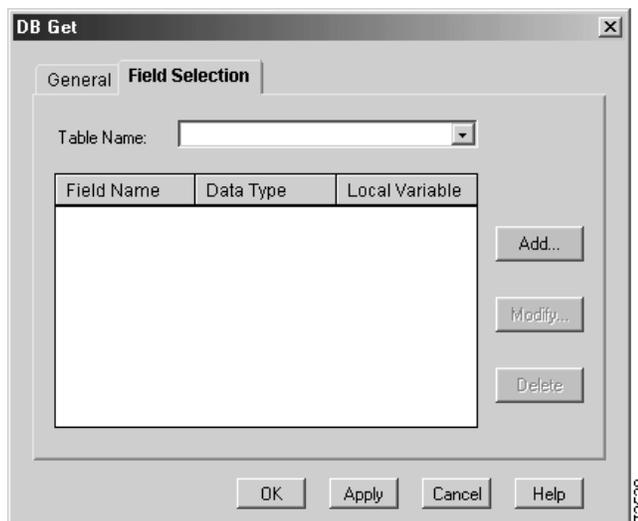


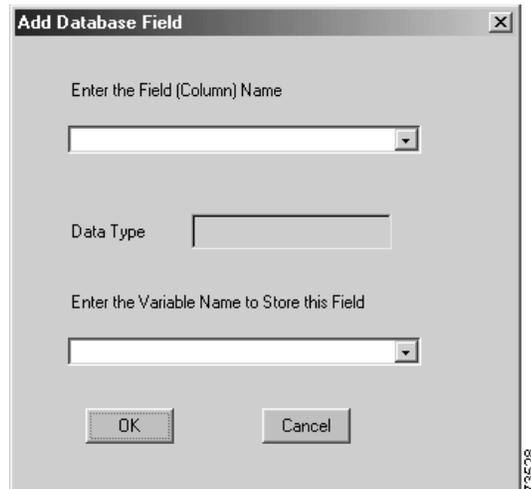
表 21 で、[Field Selection] タブのフィールドについて説明します。

表 21 [DB Get] カスタマイザ ウィンドウのフィールド : [Field Selection] タブ

プロパティ	説明
Table Name	選択したデータベース内のテーブルの名前。
Field Name	選択したデータベース テーブル内のフィールド (カラム) の名前。
Data Type	変数のデータ型。
Local Variable	関連付けられたフィールドの値を格納する変数。

[Add] をクリックすると (図 65)、[Add Database Field] ダイアログボックス (図 66) が開きます。

図 66 [Add Database Field] ダイアログボックス



## DB Write

**DB Write** ステップは、SQL 文の *update*、*insert*、および *delete* を入力して、データベース レコードを更新する場合に使用します。



(注) **DB Write** ステップがサポートしている SQL 文は、*update*、*insert*、*delete* の 3 つだけです。

**DB Write** ステップが完了したら、**DB Release** ステップ（「[DB Release](#)」(P.89) を参照）を使用して、データベース リソースを解放します。

**DB Write** ステップは、3 つの出力ブランチを生成します。

- **Successful** : エラーは発生しませんでした。ステップに入力された SQL 文の実行は成功しました。
- **Connection Not Available** : このクエリーの実行に使用できるデータベース接続がありませんでした。
- **SQL Error** : エラーが発生しました。SQL、タイムアウト、または検出された例外が含まれます。

[DB Write] カスタマイザ ウィンドウには 4 つのタブがあります。

- 「[\[General\] タブ](#)」(P.85)
- 「[\[SQL\] タブ](#)」(P.86)
- 「[\[Test\] タブ](#)」(P.88)
- 「[\[Comments\] タブ](#)」(P.88)

## [General] タブ

[DB Write] カスタマイザ ウィンドウの [General] タブ (図 67) は、SQL 文の **update**、**insert**、**delete** を適用するデータベース リソースおよびデータベース プロファイルを選択する場合に使用します。

図 67 [DB Write] カスタマイザ ウィンドウ : [General] タブ

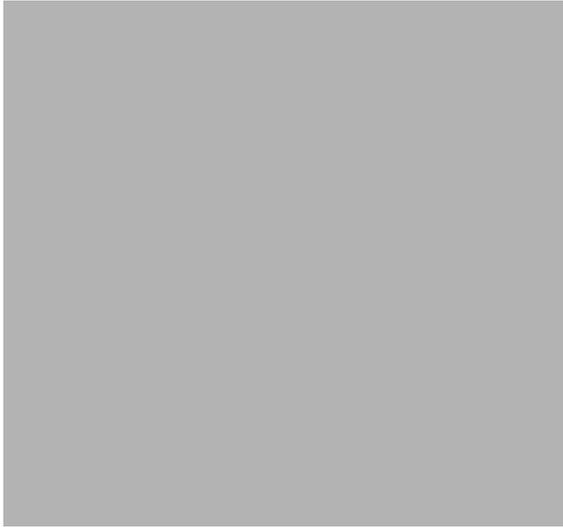


表 22 で、[General] タブのフィールドについて説明します。

表 22 [DB Write] カスタマイザ ウィンドウのフィールド : [General] タブ

プロパティ	説明
DB Resource Name	このデータベース クエリーを識別するために割り当てられている名前。
DB Profile Name	update クエリーの実行対象となるデータベース プロファイルの名前。
Refresh Database Schema	選択された Cisco Unity Express データベースで定義されているフィールドで、Editor を更新します。

## [SQL] タブ

[DB Write] カスタマイザ ウィンドウの [SQL] タブ (図 68) は、SQL コマンドを入力する場合に使用します。

図 68 [DB Write] カスタマイザ ウィンドウ : [SQL] タブ

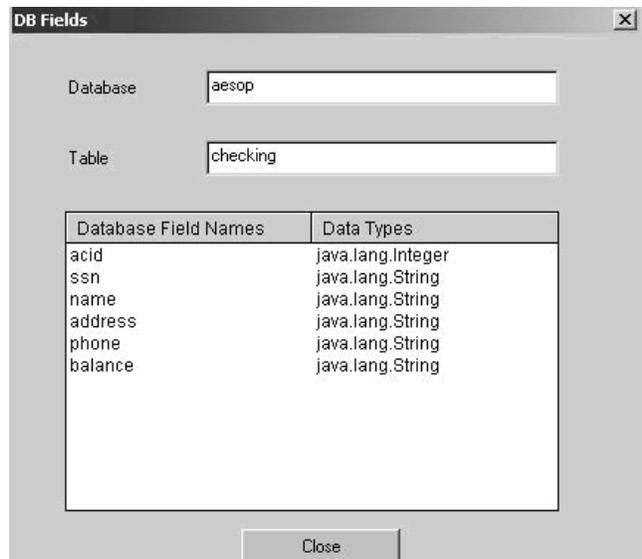


表 23 で、[SQL] タブのフィールドについて説明します。

表 23 [DB Write] カスタマイザ ウィンドウのフィールド : [SQL] タブ

プロパティ	説明
Enter SQL Command to be Executed	実行する SQL コマンド。
Show All Fields (Select Table)	このデータベース内の特定のテーブルで定義されているフィールド。ドロップダウンリストからテーブルを選択すると、[DB Fields] ダイアログボックス (図 69) が表示されます。

図 69 [DB Fields] ダイアログボックス



## [Test] タブ

[DB Write] カスタマイザ ウィンドウの [Test] タブ (図 70) は、SQL 文の *update*、*insert*、または *delete* を実行し、データベースを変更することなく結果を表示する場合に使用します。

図 70 [DB Write] カスタマイザ ウィンドウ : [Test] タブ

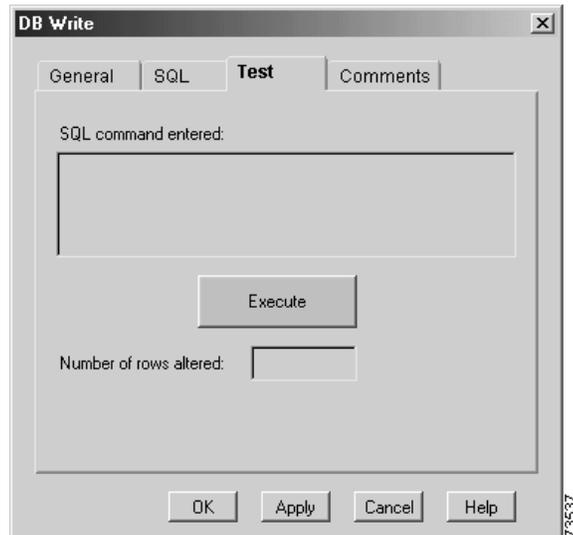


表 24 で、[Test] タブのフィールドについて説明します。

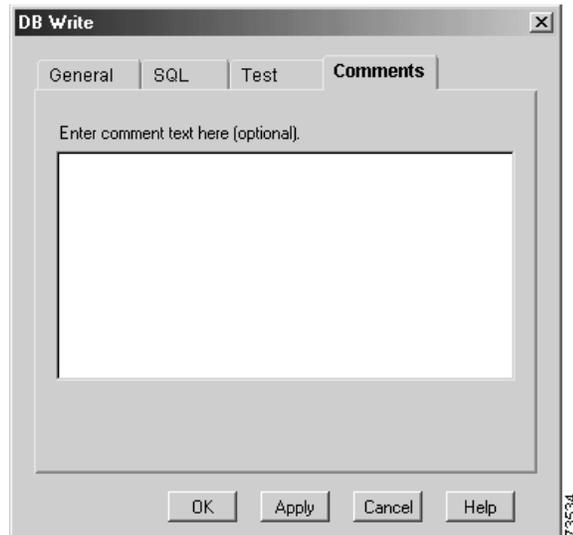
表 24 [DB Write] カスタマイザ ウィンドウのフィールド : [Test] タブ

プロパティ	説明
SQL Command Entered	SQL 文の <i>update</i> 、 <i>insert</i> 、または <i>delete</i> を入力して、試験的に実行します。
Number of Rows Altered	テストによって変更される行数を表示します。

## [Comments] タブ

[DB Write] カスタマイザ ウィンドウの [Comments] タブ (図 71) は、テスト結果に関するテキスト コメントを入力する場合に使用します (「[Test] タブ」(P.88) を参照)。

図 71 [DB Write] カスタマイザ ウィンドウ : [Comments] タブ



## DB Release

**DB Release** ステップは、リソース名に関連付けられたデータベース リソースを解放する場合に使用します。

**DB Read** ステップと **DB Write** ステップ（「[DB Read](#)」(P.79) および 「[DB Write](#)」(P.85) を参照）では、返された結果が不要になったときに、**DB Release** ステップを使用してリソースを解放する必要があります。

図 72 に、**DB Release** ステップのカスタマイザ ウィンドウを示します。

図 72 [DB Release] カスタマイザ ウィンドウ



表 25 で、[DB Release] カスタマイザ ウィンドウのフィールドについて説明します。

表 25 [DB Release] カスタマイザ ウィンドウのフィールド

プロパティ	説明
DB Resource Name	リソースを解放するデータベース リソースの名前。

## Document ステップ (IVR のみ)

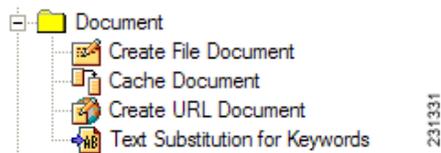
**Document** パレットのステップは、各種ドキュメントを取り扱う手段を提供します。ドキュメントステップの詳細については、『*Cisco Unity Express 7.0 IVR CLI Administrator Guide*』を参照してください。

**Document** パレットには、次のステップがあります。

- 「Cache Document」 (P.90)
- 「Create File Document (IVR のみ)」 (P.91)
- 「Create URL Document」 (P.92)
- 「Text Substitution for Keywords」 (P.94)

図 73 に、**Document** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 73 Document パレットのステップ



## Cache Document

**Cache Document** ステップは、Hypertext Transfer Protocol (HTTP) 要求の作成などの入出力 (I/O) 操作を実行したり、結果として得られたドキュメントをメモリ バッファにキャッシュしたりする場合に使用します。



(注)

**Cache Document** ステップは、最大 512 KB のメモリを使用できます。**Cache Document** のサイズが 512 KB のメモリより大きい場合、512 KB を超えるデータはすべて切り捨てられます。

I/O 操作は、このステップに先行する **Create File Document** ステップ (「[Create File Document \(IVR のみ\)](#)」 (P.91) を参照) や **Create URL Document** ステップ (「[Create URL Document](#)」 (P.92) を参照) などのステップで定義されたドキュメントや、ハードコーディングされたドキュメントを含むドキュメント式によって指定されます。**Create File Document** ステップまたは **Create URL Document** ステップを実行すると、**Document** 変数が作成されます。URL 要求を送信したり、ファイル システムにアクセスしたりはしません。

**Send Response** ステップ (「[Send Response](#)」 (P.128) を参照) などの他のステップがドキュメントを参照し、後続のステップを実行する前に I/O 操作の完了を許可すると、I/O 操作が実行されます。次の例では、**Cache Document** ステップが mybank.money.com への HTTP 要求を作成します。**Cache Document** ステップを使用しない場合、**Send eMail** ステップ (「[Send eMail](#)」 (P.99) を参照) を実行するまで I/O 操作は実行されません。

```
doc=Create URL Document("http://mybank.money.com/debit?amount=500")

doc=Cache Document(doc)

. . .

SendEmail(From, To, doc)
```

図 74 に、**Cache Document** ステップのカスタマイザ ウィンドウを示します。

図 74 [Cache Document] カスタマイザ ウィンドウ

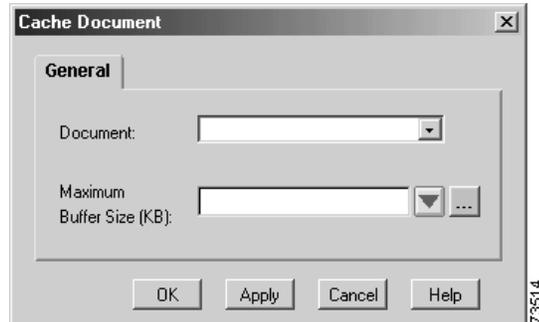


表 26 で、[Cache Document] カスタマイザ ウィンドウのフィールドについて説明します。

表 26 [Cache Document] カスタマイザ ウィンドウのフィールド

プロパティ	説明
Document	キャッシュするドキュメントをドロップダウン リストから選択できます。
Maximum Buffer Size (KB)	ドキュメントの最大バッファ サイズ。 <ul style="list-style-type: none"> <li>バッファ サイズが大きいとシステムのパフォーマンスに影響を与えることがあります。</li> <li>設定された最大バッファ サイズより大きいドキュメントは切り捨てられます。</li> <li>有効な範囲は 1 ~ 512 KB です。</li> </ul>

## Create File Document (IVR のみ)

**Create File Document** ステップは、テンプレート、Tagged Image File Format (TIFF) 添付ファイル、および他の汎用ドキュメントをスクリプトに組み込む場合に使用します。

このステップで作成されるドキュメントは、電子メールの本文に含めたり、添付ファイルとして付加したり、FAX 送信と同じ方法で使用したりすることができます。

**Create File Document** ステップを使用するときは、Cisco Unity Express モジュールにアップロードされているファイルの名前を選択する必要があります。ファイルは、CLI または GUI を使用してモジュールにロードできます。ファイルをモジュールにアップロードするときには、そのファイルに割り当てられるファイル タイプも選択する必要があります。

図 75 に、**Create File Document** ステップのカスタマイザ ウィンドウを示します。

図 75 [Create File Document] カスタマイザ ウィンドウ

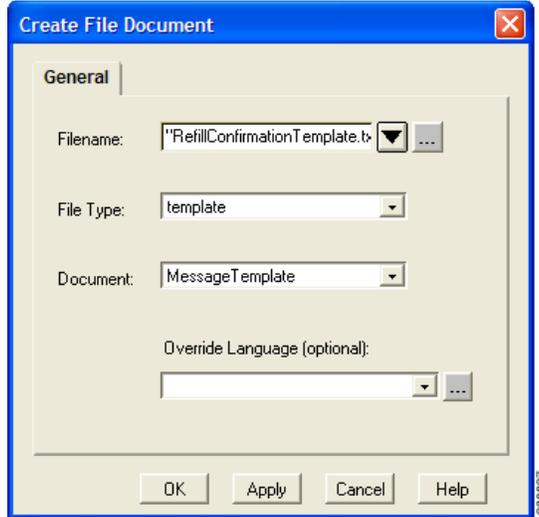


表 27 で、[Create File Document] カスタマイザ ウィンドウのフィールドについて説明します。

表 27 [Create File Document] カスタマイザ ウィンドウのフィールド

プロパティ	説明
Filetype	ファイル タイプをドロップダウン リストから選択します。リストには、Template、TIFF、Document が含まれます。
Filename	Document 変数と関連付けられるドキュメントのファイル名。ファイル名には、引用符で囲んだファイルの名前や String 変数の参照を含めることができます。
Document	指定されたドキュメントを表す変数。ドロップダウン リストから選択します。

## Create URL Document

**Create URL Document** ステップは、HTTP URL を入力して Document 変数を定義する場合に使用します。**Create URL Document** には、FTP または HTTP を使用できます。

**Create URL Document** ステップは、HTTP 要求を発行しません。HTTP 要求が発生するのは、[Send Response](#)、[Send eMail](#)、[Send Fax](#)、または [Cache Document](#) など他のステップがドキュメントを使用するときです。図 78 の例では、**Send Email** ステップを開始したときに HTTP 要求が作成され、結果として得られた HTML ドキュメントが電子メールとして user1@company1.com に送信されます。

図 76 ドキュメントが添付された電子メールの送信

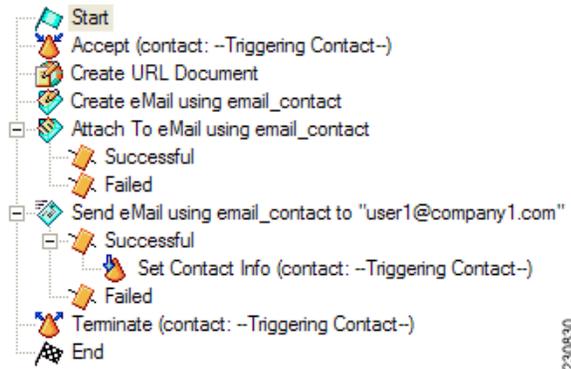


図 77 の例では、**Create URL Document** ステップを使用して HTTP 要求のみを作成し、電子メール、FAX、または HTTP 要求を送信しない場合に、**Cache Document** ステップを使用して HTTP 要求を作成します。

図 77 キャッシュした URL 応答の送信



表 28 に示すように、HTTP 要求には GET と POST の 2 つのメソッドがあります。表 28 で、[Create URL Document] カスタマイザ ウィンドウのフィールドについて説明します。

表 28 [Create URL Document] カスタマイザ ウィンドウのフィールド

プロパティ	説明
URL	ドキュメントの URL。URL は、引用符で囲まれた文字列または String 変数です。適切な形式の HTTP URL である必要があります。URL 文字列が適切な形式の HTTP URL であるかどうかは検証されません。
Method	URL が HTTP 要求を表す場合に使用するメソッド。 <ul style="list-style-type: none"> <li>• <b>GET</b> : URL にパラメータを付加します。このステップは、URL [url?name=value,name=value] 形式のドキュメント式を使用するのと同じです。</li> <li>• <b>POST</b> : HTML 形式で入力された場合と同様にパラメータを含めます。</li> </ul>
Parameters	Web サーバに送信するパラメータ文字列を形成する名前/値ペア。名前/値ペアは、URL エンコードされたパラメータ文字列に変換されます。
Document	結果として得られたドキュメント オブジェクトを格納する変数名。

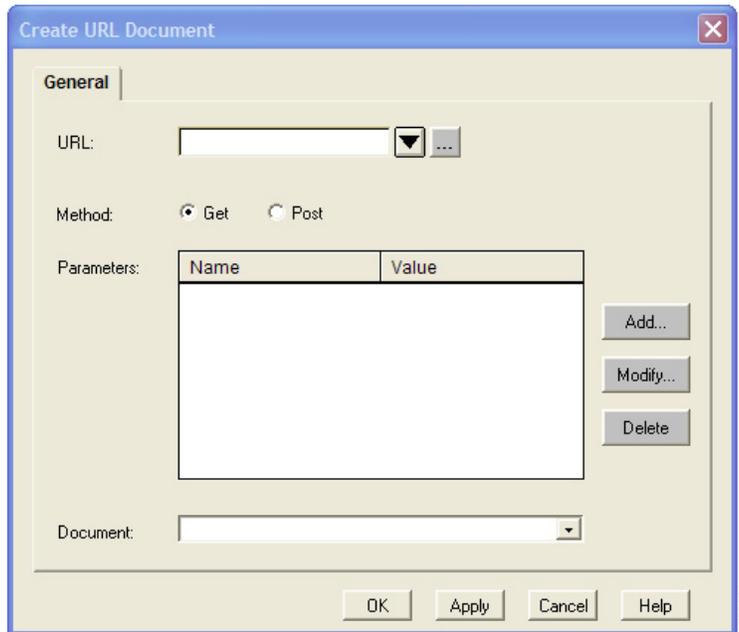
GET HTTP の場合、通常、パラメータは HTTP 要求内の URL に付加され、要求の実行に必要なデータを提供します。たとえば、合計値を 500 に設定すると、次の HTTP 要求が作成されます。

`http://mybank.money.com/debit?amount=500`

POST の場合、パラメータは、HTML 形式で入力された場合と同様に、HTTP 要求の本体に渡されます。

図 78 に、**Create URL Document** ステップのカスタマイザ ウィンドウを示します。

図 78 [Create URL Document] カスタマイザ ウィンドウ



## Text Substitution for Keywords

**Text Substitution for Keywords** ステップは、ドキュメント内のキーワードをローカル変数に保持されている値で置き換えることによって、HTTP、電子メール、および FAX の動的コンテンツを作成する場合に使用します。

キーワードは変数にマッピングされます。**Text Substitution for Keywords** ステップは、キーワードをローカル変数の現在の値で置き換えます。キーワードは、ドキュメント内では %NAME% などのようにパーセント (%) 文字で示されます。%NAME% は、String 変数で動的に人の名前に置き換わります。

図 79 に、**Text Substitution for Keywords** ステップのカスタマイザ ウィンドウを示します。

図 79 [Text Substitution for Keywords] カスタマイザ ウィンドウ

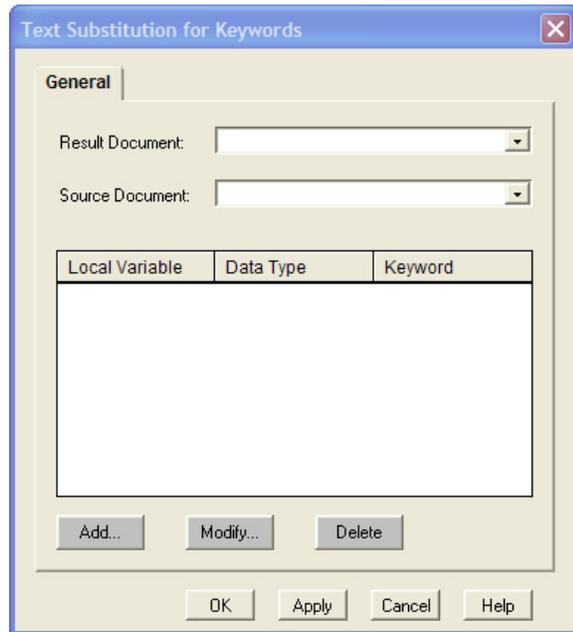


表 29 で、[Text Substitution for Keywords] カスタマイザ ウィンドウのフィールドについて説明します。

表 29 [Text Substitution for Keywords] カスタマイザ ウィンドウのフィールド

プロパティ	説明
<b>Result Document</b>	テキスト置換を含むドキュメントを格納する変数。
<b>Source Document</b>	ソース テンプレート ドキュメントを識別する変数。ウィンドウの下半分で、変更を加え、ドキュメント テンプレート内の特定のキーワードに置き換わるローカル変数を設定できます。ローカル変数は、必要に応じて追加、変更、削除できます。
<b>Local Variable</b>	キーワードの置換に使用する変数。
<b>Data Type</b>	string 型や integer 型などの、ローカル変数のデータ型。このフィールドには、選択したローカル変数のデータ型が自動的に入力されます。
<b>Keyword</b>	パーセント (%) 文字で示され、置換されるソース ドキュメント内のキーワード。このフィールドにキーワードを入力するときは、% 文字を使用しないでください。キーワードに使用できるのは英数字のみです。ダッシュ (-) 文字と下線 (_) 文字は使用できます。
<b>Add</b>	新しい置換エントリを追加します。[Add] をクリックすると、[Keyword Mapping] ダイアログボックス (図 80) が表示されます。
<b>Modify</b>	既存のエントリを変更します。[Modify] をクリックすると、[Keyword Mapping] ダイアログボックス (図 80) が表示されます。
<b>Delete</b>	エントリを削除します。

[Text Substitution for Keywords] カスタマイザ ウィンドウ (図 79) で、[Add] または [Modify] をクリックすると、[Keyword Mapping] ダイアログボックス (図 80) が表示されます。

図 80 [Keyword Mapping] ダイアログボックス



## eMail Contact ステップ (IVR のみ)

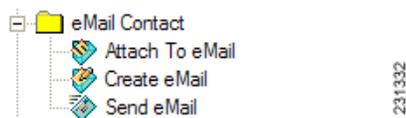
eMail Contact パレットのステップは、電子メールの作成、ファイルの添付、送信を行う機能を提供します。電子メール関連のステップは、Cisco Unity Express Script Editor の eMail Contact パレットで使用できます。eMail Contact ステップの詳細については、『[Cisco Unity Express 7.0 IVR CLI Administrator Guide](#)』を参照してください。

eMail Contact パレットには、次のステップがあります。

- 「Attach To eMail」 (P.96)
- 「Create eMail」 (P.98)
- 「Send eMail」 (P.99)

図 81 に、eMail Contact パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 81 eMail Contact パレットのステップ



## Attach To eMail

**Attach to eMail** ステップは、既存のドキュメントを電子メールに添付し、後で送信する場合に使用します。

**Create eMail** ステップ（「[Create eMail](#)」 (P.98) を参照）を使用して、事前に電子メールを作成し、対応する電子メール コンタクトに一致させておく必要があります。添付ファイルのコンテンツは、Document 変数によって指定します。

**Attach To eMail** ステップは、2 つの出カブランチを生成します。

- **Successful** : ドキュメントは電子メール オブジェクトに添付されました。
- **Failed** : ドキュメントを電子メール オブジェクトに添付できませんでした。

電子メールを添付するには、[Attach to eMail] カスタマイザ ウィンドウ (図 82) で [Add] をクリックし、[Add Attachment] カスタマイザ ウィンドウ (図 83) を完了します。

図 82 [Attach to eMail] カスタマイザ ウィンドウ

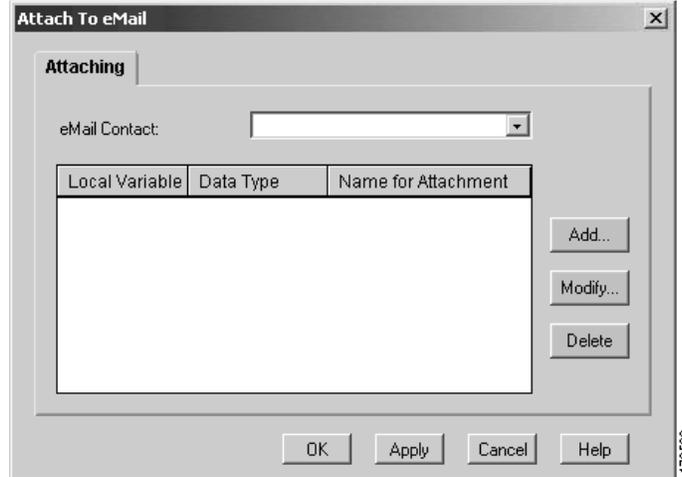


図 83 [Add Attachment] カスタマイザ ウィンドウ



表 30 で、[Attach to eMail] カスタマイザ ウィンドウと [Add Attachment] カスタマイザ ウィンドウのフィールドについて説明します。

表 30 [Attach to eMail] カスタマイザ ウィンドウおよび [Add Attachment] カスタマイザ ウィンドウのフィールド

フィールド	説明
eMail Contact	<b>Create eMail</b> ステップ（「 <a href="#">Create eMail</a> 」 (P.98) を参照）で設定された電子メール メッセージに関連付けられたコンタクト。名前は、引用符で囲んだテキストとして直接入力するか、または String 変数を使用して入力できます。
Local Variable	電子メールの添付ファイルとして送信するドキュメントを指定します。
Data Type	ローカル変数のデータ型。デフォルトのデータ型は Document です。ローカル変数を選択すると、[Data Type] フィールドに自動的に Document と表示されます。
Name for attachment	電子メール添付ファイルのテキスト名。
Add	添付ファイルを追加します。最大 5 つの添付ファイルを追加できます。
Modify	添付ファイルを変更します。
Delete	添付ファイルを削除します。

## Create eMail

**Create eMail** ステップは、電子メールを構成する場合に使用します。電子メール コンタクトと関連付ける電子メールの件名とテキスト本文を含めます。電子メールの本文を指定するには、次の手順を実行します。

1. 電子メール メッセージの本文用の **[Body]** フィールドで使用する変数を選択します。
2. **[Body]** フィールドの右側の [...] をクリックし、**[Expression Editor]** ポップアップ ウィンドウに文字列式を入力します。
3. メッセージの本文に別個のドキュメントを使用する場合は、**[Use Document for Body]** チェックボックスをオンにして、メッセージ フィールドの右側のドロップダウン リストから **Document** 変数を選択します。

図 84 に、**Create eMail** ステップのカスタマイザ ウィンドウを示します。

図 84 [Create eMail] カスタマイザ ウィンドウ



表 31 で、[Create eMail] カスタマイザ ウィンドウのフィールドについて説明します。

表 31 [Create eMail] カスタマイザ ウィンドウのフィールド

フィールド	説明
<b>eMail Contact</b>	電子メールを説明する変数。
<b>Subject</b>	電子メールの件名の説明。String 変数への参照を入力するか、または引用符で囲んだテキストを直接入力できます。件名の文字列は、最大 256 文字です。256 文字を超える文字列を入力した場合、257 文字目以降は切り捨てられます。
<b>Body</b>	電子メール メッセージのテキスト本文。String 変数または Document 変数を参照するか、または引用符で囲んだテキストを直接入力できます。Document 変数を選択した場合、参照ドキュメントにはテキストしか含めることができません。参照されるドキュメントが .JPEG ファイルや .GIF ファイルなどのバイナリ データを含む場合、システムはドキュメントのコンテンツ タイプの判別を試みます。また、電子メールは、受信時に正しく表示されない場合があります。  電子メールの本文に別個のドキュメントを使用するには、 <b>[Use Document for Body]</b> をオンにします。このチェックボックスをオンにした場合は、ドロップダウン リストから <b>Document</b> 変数を選択する必要があります。

## Send eMail

**Send eMail** ステップは、電子メールを送信する場合に使用します。

電子メールを送信する前に、**Create eMail** ステップ（「**Create eMail**」(P.98) を参照）を使用して電子メールを作成する必要があります。

**Send eMail** ステップは、宛先アドレスと送信元アドレスを設定し、配信用に設定された **Simple Mail Transfer Protocol (SMTP)**（シンプル メール転送プロトコル）サーバに電子メールを送信します。SMTP サーバは、**Cisco Unity Express** の GUI または CLI を使用して設定できます。送信元アドレスには、返信の送信先として希望するアドレスを設定できます。このアドレスは、SMTP サーバから受信した配信不能メッセージの通知先にすることもできます。

電子メールは、すぐに送信することもできれば、バックグラウンドプロセスとして後で送信するためにキューに入れることもできます。

- **Immediate** : [immediate] オプションを選択した場合、電子メールは SMTP サーバに送信され、クライアントは SMTP サーバがメッセージを受け取るまで待機します。サーバまたはネットワークの問題でサーバを使用できない場合、クライアントはトランザクションタイムアウト（30 秒）まで待機する必要があります。プロンプトを発行して、電子メールを送信中であることをクライアントに警告し、電子メール メッセージの伝送が完了するまで回線を保持するように要求する必要があります。
- **Queued** : [queued] オプションを選択した場合、ステップはすぐに戻りますが、スクリプトは、電子メールが正常に送信されたかどうかを示す通知を受信しません。

**Send eMail** ステップは、次の 3 つのブランチを生成します。

- **Successful** : 電子メールは送信され、伝送は完了しました。
- **Failed** : 電子メールを送信できませんでした。失敗の理由としては、SMTP サーバが不正に設定されていた、SMTP 認証に失敗した、SMTP サーバが到達不能でトランザクションがタイムアウトになった、などが考えられます。

図 85 に、**Send eMail** ステップのカスタマイザ ウィンドウを示します。

図 85 [Send eMail] カスタマイザ ウィンドウ



表 32 で、[Send eMail] カスタマイザ ウィンドウのフィールドについて説明します。

表 32 [Send eMail] カスタマイザ ウィンドウのフィールド

フィールド	説明
eMail Contact	<b>Create eMail</b> ステップ（「 <a href="#">Create eMail</a> 」 (P.98) を参照）で設定された電子メール メッセージに関連付けられたコンタクト。引用符で囲んだテキストとして直接入力するか、または String 変数を使用して入力できます。
To	電子メールの送信相手。変数または文字列式を使用できます。
[Use Default From] チェックボックス	<ul style="list-style-type: none"> <li>オンにした場合：Cisco Unity Express は、Cisco Unity Express の GUI または CLI を通じて設定された電子メール サブシステムのデフォルトのアカウントを使用します。</li> <li>オフにした場合：入力するか、または From 電子メール アドレスから選択する必要があります。</li> </ul>
From	電子メールの送信元のアカウント。引用符で囲んだテキストとして直接入力するか、または String 変数を使用して入力できます。
Send	電子メール転送の方式。 <ul style="list-style-type: none"> <li><b>Immediate</b>：すぐに電子メールを SMTP サーバに送信し、SMTP サーバがメッセージを受け取るまで待機します。</li> <li><b>Queued</b>：電子メールを個別のバックグラウンド プロセスで送信します。</li> </ul>

## Fax Contact ステップ (IVR のみ)

Cisco Unity Express Script Editor の **Fax Contact** パレットのステップは、スクリプトの FAX プログラミング機能を提供します。Fax Contact ステップの詳細については、『[Cisco Unity Express 7.0 IVR CLI Administrator Guide](#)』を参照してください。

**Fax Contact** パレットには、次のステップがあります。

- 「[Create Fax](#)」 (P.100)
- 「[Attach to Fax](#)」 (P.101)
- 「[Send Fax](#)」 (P.102)

図 86 に、**Fax** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 86 Fax Contact パレットのステップ



## Create Fax

**Create Fax** ステップは、FAX メッセージを作成する場合に使用します。

FAX の作成では、FAX の件名と本文を入力し、電子メール コンタクトと FAX を関連付けます。

図 87 に、**Create Fax** ステップのカスタマイザ ウィンドウを示します。

図 87 [Create Fax] カスタマイザ ウィンドウ



表 33 で、[Create Fax] カスタマイザ ウィンドウのフィールドについて説明します。

表 33 [Create Fax] カスタマイザ ウィンドウのフィールド

プロパティ	説明
Subject	FAX の件名。String 変数または引用符で囲んだテキストを入力できます。
Body	FAX メッセージの本文。String 変数または引用符で囲んだテキストを入力できます。FAX の本文は、1 KB に制限されています。Document 変数を選択した場合、参照されるドキュメントにはテキストしか含めることができません。それ以外が含まれる場合、FAX 送信は失敗します。Create Fax ステップは、FAX の本文がバイナリ データかどうかをチェックしません。
Fax Contact	FAX を識別する変数。

## Attach to Fax

**Attach to Fax** ステップは、テキスト ドキュメントまたは .TIFF イメージを FAX の添付ファイルとして送信する場合に使用します。

FAX に添付する前に、**Create Fax** ステップを使用して事前に FAX を作成し（「**Create Fax**」(P.100) を参照）、同じ FAX コントクトを使用する必要があります。添付ファイルのコンテンツは、[Local Variable] フィールドに入力された Document 変数によって識別されます。[Local Variable] フィールドで参照されるファイルには、テキストまたは .TIFF イメージを含めることができます。.TIFF イメージは、RFC2306 で定義されている TIFF-F Profile に準拠している必要があります。添付ファイルがテキストまたは .TIFF イメージでない場合、FAX は送信されません。.TIFF イメージが TIFF-F Profile 要件に準拠していない場合、FAX は送信されますが、イメージは正しく伝送されない場合があります。



(注) **Send Fax** ステップは、添付されているドキュメントのフォーマットを検証しません。

**Attach to Fax** ステップは、次の 2 つのブランチを生成します。

- **Successful** : ドキュメントは FAX オブジェクトに添付されました。
- **Failed** : ドキュメントは FAX オブジェクトに添付されませんでした。

図 88 に、**Attach to Fax** ステップのカスタマイザ ウィンドウを示します。

図 88 [Attach to Fax] カスタマイザ ウィンドウ



表 34 で、[Attach to Fax] カスタマイザ ウィンドウのフィールドについて説明します。

表 34 [Attach to Fax] カスタマイザ ウィンドウのフィールド

プロパティ	説明
Fax Contact	Create Fax ステップ（「Create Fax」 (P.100) を参照）で作成された FAX メッセージに関連付けられたコンタクト。
Document to Attach	FAX の添付ファイルとして送信するドキュメントを参照するローカル変数。ASCII テキストまたは .TIFF イメージを含む Document データ型に限定されています。

## Send Fax

**Send Fax** ステップは、FAX を送信する場合に使用します。

Send Fax ステップは、FAX 要求をキューに入れた直後にステップに戻ります。FAX に添付する前に、Create Fax ステップ（「Create Fax」 (P.100) を参照）を使用して事前に FAX を作成し、同じ FAX コンタクトを使用する必要があります。FAX は、添付ファイルを保持している場合があります。「Attach to Fax」 (P.101) を参照してください。

**Send Fax** ステップは、2 つの出力ブランチを生成します。

- **Successful** : FAX は、FAX SMTP サーバに送信するために、正常にキューに入りました。
- **Failed** : FAX は、送信のためのキューに入りませんでした。

図 89 に、**Send Fax** ステップのカスタマイザ ウィンドウを示します。

図 89 [Send Fax] カスタマイザ ウィンドウ



表 34 で、[Send Fax] カスタマイザ ウィンドウのフィールドについて説明します。

表 35 [Send Fax] カスタマイザ ウィンドウのフィールド

プロパティ	説明
Fax Contact	Create Fax ステップ（「Create Fax」 (P.100) を参照）で作成された FAX メッセージに関連付けられたコンタクト。
Fax Number	目的の FAX 機の電話番号に使用する変数または文字列式。引用符で囲んだテキストとして直接入力するか、または String 変数を使用して入力できます。この番号には、通話の発信に必要なディジットをすべて含める必要があります。
[Default From] チェックボックス	<p>Cisco Unity Express の GUI または CLI を通じて FAX サブシステム用に設定されたアカウントの使用を有効または無効にします。</p> <ul style="list-style-type: none"> <li>オンにした場合：Cisco Unity Express の GUI または CLI によって FAX サブシステム用に設定されたアカウントの使用を有効にします。</li> </ul> <p>[Default From] の値が設定されていない場合、送信される FAX 電子メールの [From] フィールドは localhost@&lt;localhostname&gt; に設定されます。</p> <p>FAX が正しく送信された場合、localhost@&lt;localhostname&gt; のアドレスを送信される FAX の送信側の電子メールアドレスとして使用します。</p> <p>FAX が正常に送信されなかった場合、デフォルトの From アドレスに失敗通知が送信されます。</p> <ul style="list-style-type: none"> <li>オフにした場合：Cisco Unity Express の GUI または CLI を通じて FAX サブシステム用に設定されたアカウントの使用を無効にします。</li> </ul>
From	電子メールの送信元のアカウント。引用符で囲んだテキスト、または String 変数を使用できます。このフィールドに入力されたアドレスは、電子メールを受信可能なアドレスである必要があります。
[Send Successful Fax Delivery Notification] オプション ボタン	<p>FAX が正常に伝送されたときに通知電子メールを送信するかどうかを指定します。</p> <ul style="list-style-type: none"> <li>Yes：通知電子メールを送信します。</li> <li>No：通知電子メールを送信しません。</li> </ul>

# General ステップ

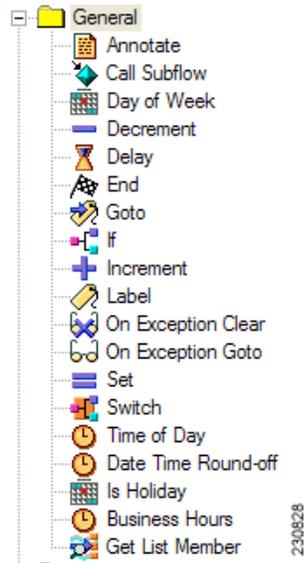
Cisco Unity Express Script Editor の **General** パレットのステップは、スクリプトの基本的なプログラミング機能を提供します。

**General** パレットには、次のステップがあります。

- 「Annotate」 (P.105)
- 「Business Hours」 (P.106)
- 「Call Subflow」 (P.106)
- 「Date Time Round-off」 (P.109)
- 「Day of Week」 (P.109)
- 「Decrement」 (P.111)
- 「Delay」 (P.111)
- 「End」 (P.112)
- 「Get List Member (IVR のみ)」 (P.112)
- 「Goto」 (P.113)
- 「If」 (P.114)
- 「Increment」 (P.114)
- 「Is Holiday」 (P.115)
- 「Label」 (P.115)
- 「On Exception Clear」 (P.116)
- 「On Exception Goto」 (P.116)
- 「Set」 (P.117)
- 「Start」 (P.118)
- 「Switch」 (P.118)
- 「Time of Day」 (P.120)

図 90 に、**General** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

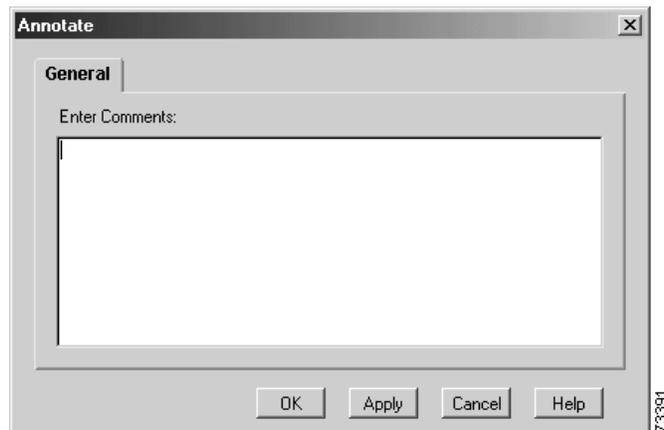
図 90 General パレットのステップ



## Annotate

**Annotate** ステップは、スクリプトのいずれかのポイントにコメントを入力する場合に使用します。このステップは、スクリプトのロジックには影響しません。図 91 に、**Annotate** ステップのカスタマイザ ウィンドウを示します。

図 91 [Annotate] カスタマイザ ウィンドウ



スクリプトに注釈を付けるには、[Enter Comments] フィールドにコメントを入力し、[OK] をクリックします。

[Annotate] カスタマイザ ウィンドウが閉じ、注釈の先頭の単語が Cisco Unity Express Script Editor の [Design] ペインの **Annotate** ステップ アイコンの横に表示されます。

## Business Hours

**Business Hours** ステップは、自動受付が通話を受けた時刻が営業時間内かどうかを判別する場合に使用します。このステップは、営業時間外に通話を受けたときに、「Business is closed」プロンプトを再生できます。設定すると、**Schedule** 変数の名前が [Design] ペインの **Business Hours** ステップの横に表示されます。

**Business Hours** ステップは、2 つの出力ブランチを自動的に追加します。

- **Open** : 指定された日時が営業時間内の場合、このブランチに続くステップが実行されます。
- **Closed** : 指定された日時が営業時間外の場合、このブランチに続くステップが実行されます。

ビジネス スケジュール設定の詳細については、『*Cisco Unity Express 7.0 GUI Administrator Guide*』または『*Cisco Unity Express Voice-Mail and Auto-Attendant CLI Administrator Guide for 3.0 and Later Versions*』を参照してください。

図 92 に、**Business Hours** ステップのカスタマイザ ウィンドウを示します。

図 92 [Business Hours] カスタマイザ ウィンドウ

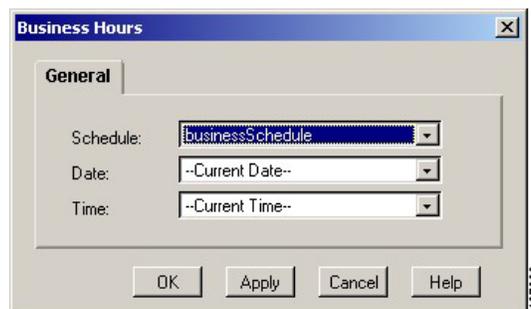


表 36 で、[Business Hours] カスタマイザ ウィンドウのフィールドについて説明します。

表 36 [Business Hours] カスタマイザ ウィンドウのフィールド

フィールド	説明
<b>Schedule</b>	Business Hours Schedule の名前。Cisco Unity Express の GUI オプションまたは CLI コマンドを使用して作成したスケジュールまたはカスタマイズしたスケジュールのいずれか 1 つを選択します。
<b>Date</b>	現在の日付。設定の必要はありません。営業時間内かどうかを確認する Date 変数を選択することもできます。
<b>Time</b>	現在の時刻。設定の必要はありません。営業時間かどうかを確認する Time 変数を選択することもできます。

## Call Subflow

**Call Subflow** ステップは、サブフローを実行する場合に使用します。サブフローは、構造化プログラミングのサブルーチンまたはモジュールとも呼ばれます。

Cisco Unity Express Script Editor を使用して、他のスクリプトで再利用できる独立したスクリプトとしてサブフローを作成します。サブフローはネストできます。つまり、サブフローとして使用されるスクリプトからサブフローを呼び出すことができます。

実行時にサブフロー内で例外が発生し、このサブフロー内で例外を処理しない場合は、親スクリプトで例外を処理できます。例外の詳細については、「[On Exception Goto](#)」(P.116) を参照してください。

[Call Subflow] カスタマイザ ウィンドウには、2 つのタブがあります。

- 「[General] タブ」(P.107)
- 「[Parameter Mapping] タブ」(P.107)

## [General] タブ

[Call Subflow] カスタマイザ ウィンドウの [General] タブ (図 93) は、呼び出すサブフローのファイル名を指定する場合に使用します。

図 93 [Call Subflow] カスタマイザ ウィンドウ : [General] タブ



表 37 で、[General] タブのフィールドについて説明します。

表 37 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Subflow Name	呼び出すサブフローのファイル名。この名前は、[Design] ペインの <b>Call Subflow</b> ステップ アイコンの横に表示されます。
Disable Interruptions	このチェックボックスがオンの場合は、外部イベントがステップの実行に割り込むことができません。

## [Parameter Mapping] タブ

[Call Subflow] カスタマイザ ウィンドウの [Parameter Mapping] タブ (図 94) は、メイン スクリプトの変数または式を [Call Subflow] カスタマイザ ウィンドウの [General] タブで指定したサブフローの変数にマッピングする場合に使用します。



(注) 変数をマッピングする前に、マップ スクリプトの変数を定義する必要があります。

変数は、同じ型の変数にだけマッピングできます。たとえば、メイン スクリプトの String 変数は、サブフローの String 変数にだけマッピングできます。

任意の有効な式を渡すことができます。たとえば、「4」または「counter + 3」などの式を渡すことができます。

スクリプトがサブフローを呼び出すとき、サブフローは [Parameter Mapping] タブで指定されたメインスクリプトの変数にアクセスできます。サブフローがマッピングされた変数の値を変更した場合、サブフローが制御をメインスクリプトに戻したときに、この変更がメインスクリプトに引き継がれます。

図 94 [Call Subflow] カスタマイザ ウィンドウ : [Parameter Mapping] タブ



表 38 で、[Parameter Mapping] タブのフィールドについて説明します。

表 38 [Call Subflow] カスタマイザ ウィンドウ : [Parameter Mapping] タブ

フィールド	説明
<b>From</b>	サブフロー スクリプトの変数の値を受け取るメイン スクリプトの変数名を表示します。
<b>To</b>	メイン スクリプトの変数に割り当てるサブフロー スクリプトの変数名を表示します。
<b>Type</b>	変数の型を表示します。
<b>Add</b>	クリックすると、メイン スクリプトの変数を追加し、サブフローにマッピングできます。別のダイアログボックスが表示されます。
<b>Modify</b>	クリックすると、選択した変数を変更できます。
<b>Delete</b>	クリックすると、選択した変数を削除できます。

[Parameter Mapping] ダイアログボックスを図 95 に示します。

図 95 [Parameter Mapping] ダイアログボックス

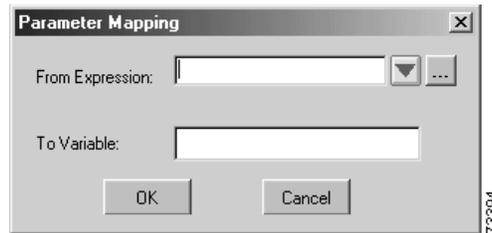


表 39 [Parameter Mapping] ダイアログボックス

フィールド	説明
From Expression	メイン スクリプトの変数名または式を入力します。
To Variable	サブフローの変数名を入力します。この名前は、[Call Subflow] ウィンドウのリスト ボックスに表示されます。
...	Expression Editor が表示されます。

## Date Time Round-off

**Date Time Round-off** ステップ ユーティリティは、現在の日時を切り上げまたは切り捨てる場合に使用します。Cisco Unity Express システム クロックが、丸めた値セットを持つ近似時刻（丸めた値を含む接続に関連付けられている時刻）と一致する場合、スクリプトは、その出力ブランチに設定されている任意のステップを実行します。

次に例を示します。

- このステップの丸め設定が 15 分に設定されている場合、時刻 6:43 は 6:45 に切り上げられます。
- このステップの丸め設定が 25 分に設定されている場合、時刻 6:43 は 7:25 に切り上げられます。
- このステップの丸め設定が 2 時間に設定されている場合、時刻 6:43 は 8:00 に切り上げられます。

## Day of Week

**Day of Week** ステップは、現在の曜日に応じて異なる出力ブランチに接続するようにスクリプトに指示する場合に使用します。

Cisco Unity Express のシステム クロックが、接続に関連付けられた曜日のいずれかと一致する場合、スクリプトは、その曜日の接続ブランチに設定されている任意のステップを実行します。

すべての曜日に出力ブランチを設定し、各曜日をそれぞれの接続に割り当てます。曜日が出力ブランチに割り当てられていない場合、[Day of Week] カスタマイザ ウィンドウを閉じるときに、Cisco Unity Express Script Editor は警告ダイアログボックスを表示します。

図 96 に、**Day of Week** ステップのカスタマイザ ウィンドウを示します。

図 96 [Day of Week] カスタマイザ ウィンドウ

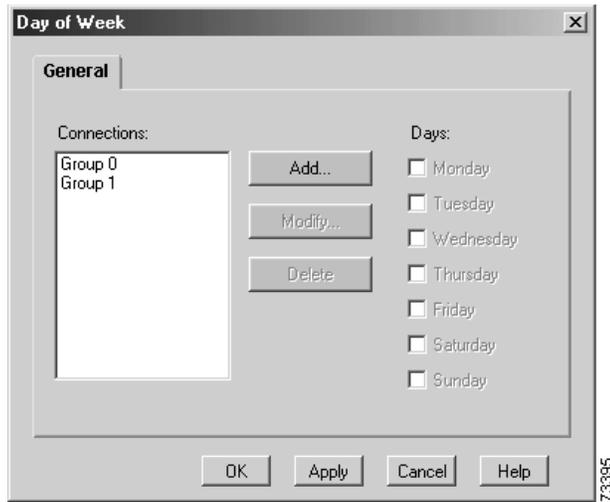


表 40 で、[Day of Week] カスタマイザ ウィンドウのフィールドについて説明します。

表 40 [Day of Week] カスタマイザ ウィンドウのフィールド

フィールド	説明
Connections	指定された曜日に応じて実行する出力ブランチ。[Connections] リストボックスで接続を選択し、このブランチに関連付ける曜日のチェックボックスをオンにします。
Days	各接続ブランチの曜日。
Add	クリックすると、接続名を追加できます。
Modify	クリックすると、選択した接続名を変更できます。スクリプトをわかりやすくするなどの目的で、接続する既存の出力ブランチの名前を変更するには、[Connections] リストボックスで接続を選択し、[Modify] をクリックします。[Modify Connection Name] 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には、[Add Connection Name] ダイアログボックスと同じフィールドがあり、同じ方法で設定します。

[Add Connection Name] ダイアログボックスを図 97 に示します。また、表 41 でフィールドについて説明します。

図 97 [Add Connection Name] ダイアログボックス

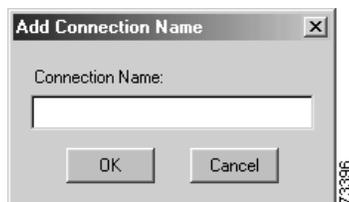


表 41 [Add Connection Name] ダイアログボックス

フィールド	説明
Connections Name	接続のブランチの名前を入力します。この名前は、[Day of Week] カスタマイザ ウィンドウの [Connections] リスト ボックスに表示されます。

## Decrement

**Decrement** ステップは、選択した Integer 変数の値を 1 つずつ減らす場合に使用します。このステップは、変数に値を代入するときに使用する **General** パレットの **Set** ステップの特殊なバージョンです。

図 98 に、**Decrement** ステップのカスタマイザ ウィンドウを示します。

図 98 [Decrement] カスタマイザ ウィンドウ



[Variable] ドロップダウン メニューから目的の変数を選択します。この変数は、[Design] ペインの **Decrement** ステップ アイコンの横に表示されます。

## Delay

**Delay** ステップは、指定した秒数で、スクリプトの処理を一時停止する場合に使用します。

図 99 に、**Delay** ステップのカスタマイザ ウィンドウを示します。

図 99 [Delay] カスタマイザ ウィンドウ



表 42 で、[Delay] カスタマイザ ウィンドウのフィールドについて説明します。

表 42 [Delay] カスタマイザ ウィンドウのフィールド

フィールド	説明
Enter delay time in seconds or enter an expression	遅延の秒数、または遅延時間を指定する式を入力します。
...	Expression Editor を表示します。遅延時間を指定する式を入力します。
Interruptible	[Yes] をクリックすると、外部イベントによって遅延に割り込むことができます。

## End

**End** ステップは、スクリプトの最後で、処理を完了し、割り当てられたすべてのリソースを解放する場合に使用します。

スクリプト ロジックのブランチの最後でも **End** ステップを使用できます。このステップが実行された時点でアクティブな呼び出しはすべて、自動的にシステムのデフォルト ロジックで処理されます。

このステップにはプロパティがなく、カスタマイザの必要はありません。

## Get List Member (IVR のみ)

**Get List Member** ステップは、デリミタで区切られた文字列リストの特定の位置（インデックス）からメンバー（トークン）を取得します。

このステップを使用して、データベースから取得した文字列データを解析し、その一部を抽出できます。抽出されたメンバー（トークン）は、他のステップで使用できます。たとえば、データベース クエリーによって、ある患者が現在使用しているすべての処方箋番号をカンマ区切りのリストとして取得するものとします。このステップを使用すると、そのリストから個々の処方箋番号を抽出し、**Play Prompt** ステップを使用して各処方箋番号を発信者に対して再生し、補充する処方箋番号を選択するように要求することができます。

**Get List Member** ステップは、2 つの出力ブランチを生成します。

- **Success** : データは正常に取得され、適切な出力文字列に配置されました。
- **Invalid Position** : データの抽出元となる指定された位置（インデックス）が無効でした。ループを使用してメンバーを取得している場合は、ループの終了を示していることもあります。

 100 に、**Get List Member** ステップのカスタマイザ ウィンドウを示します。

図 100 [Get List Member] ウィンドウ



表 43 で、[Get List Member] カスタマイザ ウィンドウのフィールドについて説明します。

表 43 [Get List Member] カスタマイザ ウィンドウのフィールド

フィールド	説明
List	文字列リストを保持する変数。
Position	文字列メンバーの抽出元となる位置（インデックス）を保持する変数。
Delimiter	文字列リスト内のメンバーを区切るデリミタを指定する変数。
Output String	ステップが成功した場合に、文字列リスト内の位置（インデックス）から抽出された出力文字列を格納する変数。

## Goto

**Goto** ステップは、スクリプト内で指定した **Label** ステップにスクリプト ロジックを分岐させる場合に使用します。



(注) **Label** ステップは、**Goto** ステップの分岐先を示す場合に使用します。

図 101 に、**Goto** ステップのカスタマイザ ウィンドウを示します。

図 101 [Goto] カスタマイザ ウィンドウ



[Select a Label] ドロップダウン メニューから **Label** ステップを選択します。この名前は、[Design] ペインの **Goto** ステップ アイコンの横に表示されます。

## If

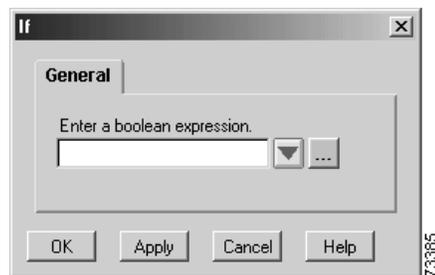
**If** ステップは、指定された **Boolean** 式の評価に基づき、2 つのブランチのいずれかをスクリプトに選択させる場合に使用します。

**If** ステップは、**True** と **False** の 2 つの出力ブランチを自動的に追加します。

- **True** : 式が **True** の場合、この出力ブランチに続くステップが実行されます。
- **False** : 式が **False** の場合、この出力ブランチに続くステップが実行されます。

☒ 102 に、**If** ステップのカスタマイザ ウィンドウを示します。

図 102 [If] カスタマイザ ウィンドウ



テキスト フィールドに式を入力するか、**Expression Editor (...)** ボタンをクリックして式を入力します。この式は、[Design] ペインの **If** ステップ アイコンの横に表示されます。

## Increment

**Increment** ステップは、選択した **Integer** 変数の値を 1 つずつ増やす場合に使用します。このステップは、変数に値を代入するときに使用する **General** パレットの **Set** ステップの特殊なバージョンです。

☒ 103 に、**Increment** ステップのカスタマイザ ウィンドウを示します。

図 103 [Increment] カスタマイザ ウィンドウ



[Variable] ドロップダウン メニューから **Integer** 型の変数を選択します。この変数は、[Design] ペインの **Increment** ステップ アイコンの横に表示されます。

## Is Holiday

**Is Holiday** ステップは、事前に定義した休日に従ってフローを分岐させる条件付きステップです。このステップは、「Business closed for the holiday」プロンプトを再生できます。

**Is Holiday** ステップは、2つの出力ブランチを自動的に追加します。

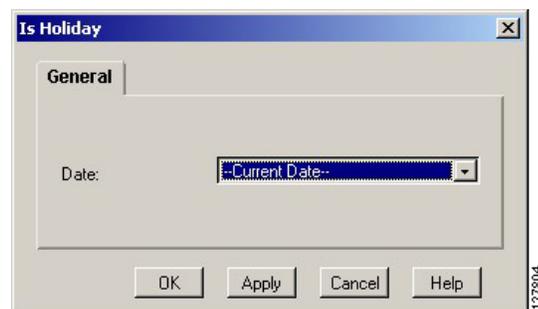
- **True** : 指定された日付が休日の場合、このブランチに続くステップが実行されます。
- **False** : 指定された日付が休日でない場合、このブランチに続くステップが実行されます。

ステップを実行するときに、システムは、指定された日付を休日のリストと比較します。指定された日付が休日の場合、**True** ブランチが実行されます。指定された日付が休日でない場合、**False** ブランチが実行されます。

休日スケジュール設定の詳細については、『[Cisco Unity Express 7.0 GUI Administrator Guide](#)』または『[Cisco Unity Express Voice-Mail and Auto-Attendant CLI Administrator Guide for 3.0 and Later Versions](#)』を参照してください。

☒ 104 に、Is ステップのカスタマイザ ウィンドウを示します。

☒ 104 [Is Holiday] カスタマイザ ウィンドウ



[Date] フィールドには、システムが休日のチェックに使用する Date 変数が含まれています。**Current Date** 変数を変更しないでおくことも、[Date] ドロップダウンメニューをクリックして Date 変数を選択することもできます。Date 変数の名前は、[Design] ペインの **Is Holiday** ステップ アイコンの横に表示されます。

## Label

**Label** ステップは、**Goto** ステップのターゲットの役割を果たします。Label ステップは、1つ以上の Goto ステップが実行できるスクリプトの一部を示します。Label ステップと Goto ステップは、同じスクリプト内に存在する必要があります。

☒ 105 に、Label ステップのカスタマイザ ウィンドウを示します。

図 105 [Label] カスタマイザ ウィンドウ



[Enter Label Name] テキスト フィールドに名前を入力します。このラベル名は、[Design] ペインの **Label** ステップ アイコンの横に表示されます。

## On Exception Clear

**On Exception Clear** ステップは、以前の **On Exception Goto** ステップで設定された例外を削除する場合に使用します。

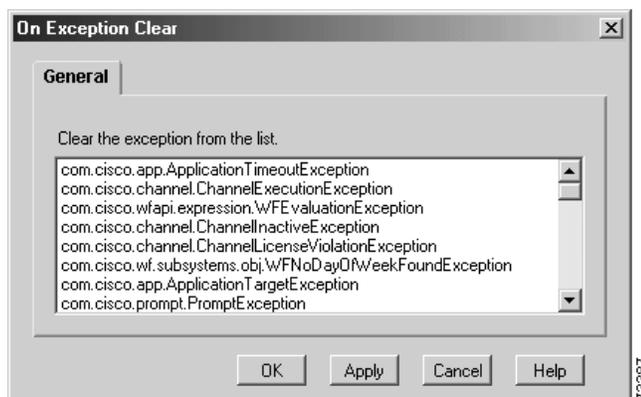
通常、このステップは、次の順序で使用されます。

1. **On Exception Goto** ステップが、スクリプトを **Label** ステップに誘導する。
2. **Label** ステップに、例外を処理するスクリプトが設定されている。
3. **On Exception Clear** ステップが、例外をクリアする。

選択した例外をスクリプト内で処理する必要がなくなったときにも、このステップを使用します。

図 106 に、**On Exception Clear** ステップのカスタマイザ ウィンドウを示します。

図 106 [On Exception Clear] カスタマイザ ウィンドウ



リスト ボックスから特定の例外を選択します。クリアされる例外は、[Design] ペインの **On Exception Clear** ステップ アイコンの横に表示されます。

## On Exception Goto

**On Exception Goto** ステップは、スクリプトの実行中に発生する問題を検出し、この状態を正しい手順で終了させることができるようにする場合に使用します。

例外に応答するために使用する Exception Flow ブランチには、任意のスクリプト ステップを含めることができます。

サブフローを使用し、このサブフローで例外を処理しない場合、例外はスクリプトに返され、スクリプトで例外に応答できます。

図 107 に、**On Exception Goto** ステップのカスタマイザ ウィンドウを示します。

図 107 [On Exception Goto] カスタマイザ ウィンドウ



表 44 で、[On Exception Goto] カスタマイザ ウィンドウのフィールドについて説明します。

表 44 On Exception Goto のフィールド

フィールド	説明
Choose the exception from the list	ステップの実行をトリガーする例外。この例外は、[Design] ペインの <b>On Exception Goto</b> ステップ アイコンの横に表示されます。
Choose the label from the list	スクリプトが分岐する先のラベル。
Save root cause (optional)	例外の原因。[Save root cause] フィールドを使用して、例外オブジェクトに保存されます。  オブジェクトの型は、検出する例外の型、またはその例外の基本クラスと対応している必要があります。対応していない場合、設計時に警告は生成されず、実行時にエラーが発生します。

## Set

**Set** ステップは、変数の値を変更する場合に使用します。

**Set** ステップは、任意の Number データ型 (Integer、Float、Long、Double、BigInteger、BigDecimal) から他の Number データ型への型変換をサポートします (精度が失われる可能性があります)。

**Set** ステップを使用して、**String** 変数を任意の **Number** データ型に変換することもできます。**String** 変換を行う場合、システムは変換を実行する前に、すべての「\*」文字を小数点（「.」）に置換します。

図 108 に、**Set** ステップのカスタマイザ ウィンドウを示します。

図 108 [Set] カスタマイザ ウィンドウ

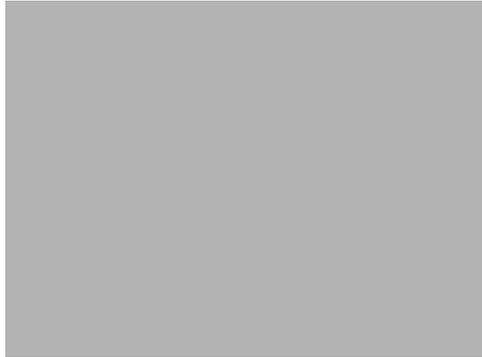


表 45 で、[Set] カスタマイザ ウィンドウのフィールドについて説明します。

表 45 [Set] カスタマイザ ウィンドウのフィールド

フィールド	説明
Variable	値を設定する変数。この変数は、[Design] ペインの <b>Set</b> ステップ アイコンの横に表示されます。
Type	変数の型。アプリケーション ソフトウェアがこの値を割り当てます。
Assign	指定した変数の値。[Assign] ドロップダウン メニューから値を選択するか、[Expression Editor (...)] ボタンをクリックして有効な式を入力します。

## Start

**Start** ステップは、新しいスクリプトを作成したときに、Cisco Unity Express Script Editor によって自動的に追加されます。このステップにはプロパティがなく、カスタマイザの必要はありません。どのパレットにも表示されません。

## Switch

**Switch** ステップは、指定された式の評価に基づき、複数の **case** のいずれかにプログラム ロジックを分岐させる場合に使用します。

**case** が、変数の値に基づいてスクリプト ロジックを提供します。1 つの変数は、複数の値のうち 1 つを持つことができます。各値に 1 つの **case** を割り当てることができます。**Switch** ステップを使用して、任意の数の **case** 出力ブランチを定義できます。次に、各 (**case**) ブランチごとに別のスクリプト ロジックを作成できます。

**Switch** ステップは、次の変数に基づいた切り替えをサポートします。

- Integer : 整数の比較
- String : String 変数の比較 (大文字と小文字を区別しない)

切り替えの種類は、指定された式の型によって、自動的に決定されます。

case に指定された整数または文字列の式が [Switch Expression] フィールドで定義されたグローバル式と等しい場合、スクリプトは case 出力ブランチに設定されたステップを実行します。

ステップの **Default** ブランチを使用すると、式と一致するブランチがない case を処理できます。

図 109 に、**Switch** ステップのカスタマイザ ウィンドウを示します。

図 109 [Switch] カスタマイザ ウィンドウ

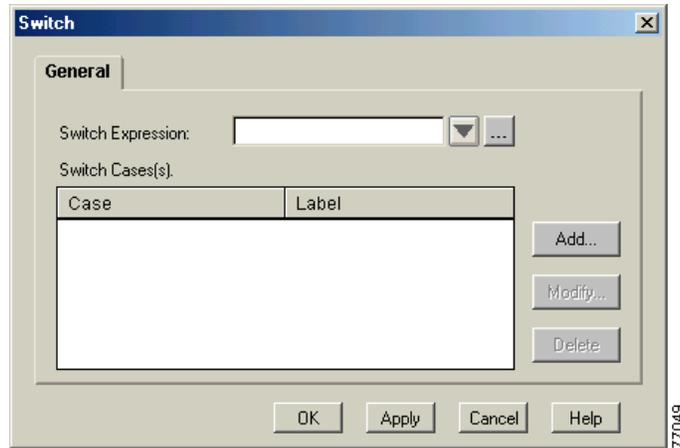


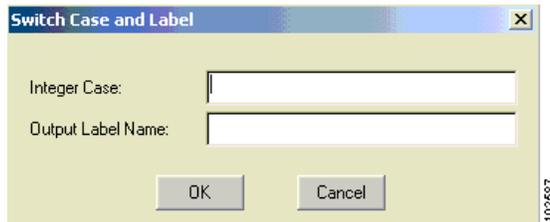
表 46 で、[Switch] カスタマイザ ウィンドウのフィールドについて説明します。

表 46 [Switch] カスタマイザ ウィンドウのフィールド

フィールド	説明
Switch Expression	実行する式。変数または式を選択するか、 <b>Expression Editor (...)</b> ボタンをクリックして有効な式を入力します。
Switch Case(s)	<ul style="list-style-type: none"> <li>• <b>Case</b> : ある変数値に固有のスクリプト ロジックを含む出力ブランチ。</li> <li>• <b>Label</b> : 変数が特定の値と等しい場合にスクリプトが分岐する先のターゲット。</li> </ul>
Add	case を追加します。
Modify	case を変更します。
Case	Integer または Case。Switch Expression のタイプによって自動的に判別されます。
Output Label Name	出力ラベル名を入力します。変数が [Case] フィールドで指定された値と等しい場合、スクリプトはこのラベルに分岐します。

[Switch Case and Label] ダイアログボックスを図 110 に示します。

図 110 [Switch Case and Label] ダイアログボックス



## Time of Day

**Time of Day** ステップは、現在の時刻に応じて異なる接続ブランチにスクリプトを分岐させる場合に使用します。

Cisco Unity Express のシステム クロックの時刻が、接続に関連付けられた時刻と一致する場合、スクリプトは、その出力ブランチに設定された任意のステップを実行します。

各出力ブランチを特定の時間範囲に関連付けます。

実行時に、現在時刻が設定された時間範囲に当てはまらない場合、スクリプトは **Time of Day** ステップの **Rest** 出力ブランチを実行します。

図 111 に、**Time of Day** ステップのカスタマイザ ウィンドウを示します。

図 111 [Time of Day] カスタマイザ ウィンドウ



表 47 で、[Time of Day] カスタマイザ ウィンドウのフィールドについて説明します。

表 47 [Time of Day] カスタマイザ ウィンドウのフィールド

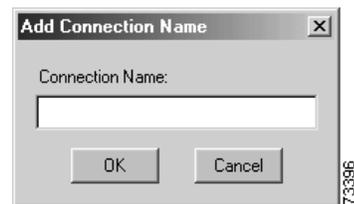
フィールド	説明
Connections	指定された時刻に応じて実行する出力ブランチ。
Time Ranges	各接続ブランチの時間範囲。
Add	クリックすると、接続を追加できます。ダイアログボックスに接続名を入力します。
Modify	クリックすると、接続を変更できます。
Delete	クリックすると、選択した接続を削除できます。

表 47 [Time of Day] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
Add Time	クリックすると、接続に時刻を追加できます。接続の時間範囲を指定するには、[Start Time] と [End Time] を選択します。
Modify Time	クリックすると、接続の時刻を変更できます。
Delete Time	クリックすると、選択した時刻を削除できます。

[Add Connection Name] ダイアログボックスを [図 112](#) に示します。

図 112 [Add Connection Name] ダイアログボックス



[Add Range of Hours] ダイアログボックスを [図 113](#) に示します。

図 113 [Add Range of Hours] ダイアログボックス



## HTTP Contact ステップ (IVR のみ)

Cisco Unity Express Script Editor の **HTTP Contact** パレットのステップを使用すると、Web 対応のサーバアプリケーションで HTTP 要求を受信し、HTTP 応答を送信できます。HTTP Contact ステップの詳細については『*Cisco Unity Express 7.0 IVR CLI Administrator Guide*』を参照してください。

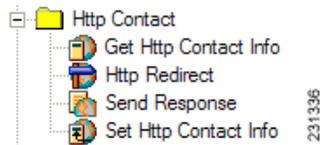
**HTTP Contact** パレットには、次のステップがあります。

- 「[Get Http Contact Info](#)」 (P.122)
- 「[Http Redirect](#)」 (P.128)

- 「Send Response」 (P.128)
- 「Set Http Contact Info」 (P.129)

図 114 に、**HTTP Contact** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 114 HTTP Contact パレットのステップ



## Get Http Contact Info

**Get Http Contact Info** ステップは、HTTP 要求のパラメータをローカルに定義された変数にマッピングする場合に使用します。

**Get Http Contact Info** ステップは、URL パラメータ、HTTP ヘッダー、クッキー、または Common Gateway Interface (CGI; コモン ゲートウェイ インターフェイス) 環境変数を取得します。URL パラメータ、ヘッダー情報、クッキー、CGI 変数は、ローカルで定義された変数にマッピングされます。

[GET Http Contact Info] カスタマイザ ウィンドウには、5 つのタブがあります。

- 「[General] タブ」 (P.122)
- 「[Headers] タブ」 (P.123)
- 「[Parameters] タブ」 (P.124)
- 「[Cookies] タブ」 (P.125)
- 「[Environment] タブ」 (P.126)

### [General] タブ

[Get Http Contact Info] カスタマイザ ウィンドウの [General] タブ (図 115) は、**Get Http Contact Info** ステップの実行をトリガーする変数の型を選択する場合に使用します。

図 115 [Get Http Contact] カスタマイザ ウィンドウ : [General] タブ



表 48 で、[General] タブのフィールドについて説明します。

表 48 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Http Contact	ステップの実行をトリガーする Contact 変数。他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。

## [Headers] タブ

[Get Http Contact Info] カスタマイザ ウィンドウの [Headers] タブは、ローカル変数にマッピングされている HTTP ヘッダーを表示する場合に使用します。

HTTP ヘッダーには、ブラウザのタイプや HTTP のバージョンなどの一般的な情報が含まれます。各ヘッダーは、ヘッダー名によって識別される 1 つの値を提供します。たとえば、HTTP ヘッダーの情報をより複雑なスクリプトで使用して、異なる HTTP バージョンまたはブラウザタイプに合わせてスクリプトの動作をカスタマイズできます。

HTTP には、4 種類のヘッダーがあります。

- **General** : サーバとクライアント (ブラウザ) の両方が使用します。
- **Server** : サーバのみが使用します。
- **Request** : クライアント (ブラウザ) のみが使用します。
- **Entity** : サーバと、POST メソッドまたは PUT メソッドを使用するクライアントが使用します。

共通の HTTP 要求ヘッダーには次のものがあります。

- **Accept** : 推奨されるメディア タイプ
- **Authorization** : クライアントのユーザ名とパスワード
- **From** : クライアントの電子メール アドレス
- **Host** : 元の要求を受信するサーバのホスト名とポート番号
- **Referrer** : ソース ドキュメントの URL
- **User-Agent** : ブラウザのタイプ



(注)

これらのヘッダーおよび他のヘッダーの詳細については、HTTP リファレンス ガイドを参照してください。

図 116 に、[Headers] タブのカスタマイザ ウィンドウを示します。

図 116 [Get Http Contact Info] カスタマイザ ウィンドウ : [Headers] タブ



表 49 で、[Headers] タブのフィールドについて説明します。

表 49 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [Headers] タブ

フィールド	説明
Header	ヘッダーの名前。
Variable	ヘッダーにマッピングされる変数。

## [Parameters] タブ

[Get Http Contact Info] カスタマイザ ウィンドウの [Parameters] タブ (図 117) は、Cisco Unity Express Editor でスクリプト パラメータを変数にマッピングする場合に使用します。

HTML フォームを完了すると、通常、値はパラメータとして Web サーバに渡されます。**Get Http Contact Info** ステップは、GET メソッドおよび POST メソッド (表 28 (P.93) を参照) を使用した HTTP 要求からこれらのパラメータの値を読み取り、スクリプト内のローカル変数を更新します。

図 117 [Get Http Contact Info] カスタマイザ ウィンドウ : [Parameters] タブ



表 50 で、[Parameters] タブのフィールドについて説明します。

表 50 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [Parameters] タブ

フィールド	説明
Parameter	パラメータの名前。
Variable	パラメータにマッピングされている変数の名前。

## [Cookies] タブ

[Get Http Contact Info] カスタマイザ ウィンドウの [Cookies] タブは、ローカル変数の情報をクッキーにマッピングする場合に使用します。

クッキーはブラウザが保持している情報で、通常は HTTP サーバによって送信されます。クッキーの情報に基づいて、Web ページへのアクセスを改善することができます。ほとんどのクッキーは、認証情報または識別情報を格納しています。サーバは、いったんブラウザを認証すると、認証クレデンシャルまたは他のユーザ ID をブラウザのクッキーに送信できます。それ以降は、認証または識別なしで Web ページにアクセスできます。

クッキーを使用して、セッション オブジェクトにマッピング識別子を格納できます。これによって、後続の要求では、以前の HTTP 要求と関連付けられた元のセッション オブジェクトを取得して、新しい HTTP コンタクトと関連付けし直すことができます。



(注)

認証にクッキーを使用すると、認証されたブラウザが無人のまま放置された場合に、セキュリティリスクが生じる可能性があります。

図 118 に、[Cookies] タブのカスタマイザ ウィンドウを示します。表 51 で、そのフィールドについて説明します。また、図 119 は、クッキー名と変数名を追加する [Cookies] ウィンドウを示しています。表 52 で、そのフィールドについて説明します。変数名には、変数が保持するデータを示す名前を使用します。

図 118 [Get Http Contact Info] カスタマイザ ウィンドウ : [Cookies] タブ

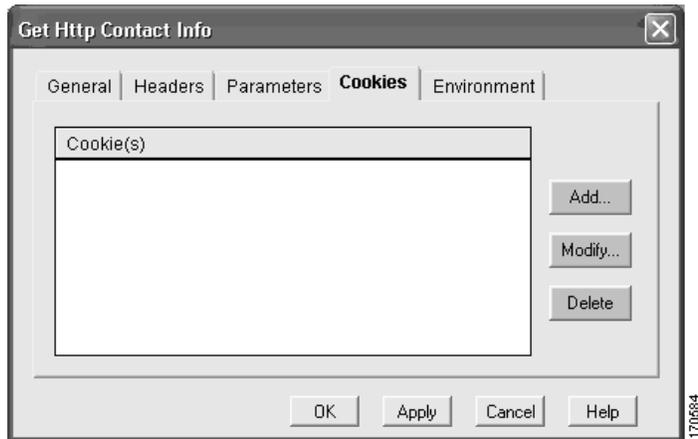


表 51 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [Cookies] タブ

フィールド	説明
Cookie	クッキーの名前。

図 119 [Cookies] ウィンドウ

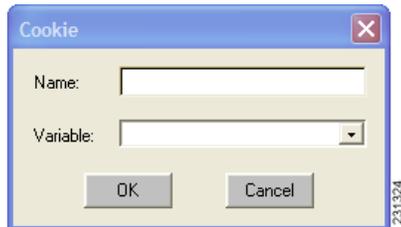


表 52 [Cookies] ウィンドウのフィールド

フィールド	説明
Name	クッキーの名前。
Variable	変数の名前。

## [Environment] タブ

[Get Http Contact Info] カスタマイザ ウィンドウの [Environment] タブは、CGI 環境変数の情報をローカル変数にマッピングする場合に使用します。

次に、環境変数をアルファベット順に一覧します。

- **AUTH\_TYPE** : サーバがユーザ認証をサポートし、スクリプトが認証を必要とする場合に、ユーザの検証に使用するプロトコル固有の認証メソッド。
- **CONTENT\_LENGTH** : クライアントが指定したデータのコンテンツ長。
- **CONTENT\_TYPE** : 添付された情報を持つ、HTTP GET や HTTP POST などのクエリー用データのコンテンツ タイプ。

- **PATH\_INFO** : クライアントが提供する追加のパス情報。仮想パス名の末尾に追加情報を付加したもので、スクリプトにアクセスできます。追加情報は、**PATH\_INFO** として送信されます。URL から取得された場合、サーバはこの情報をスクリプトに渡す前にデコードします。
- **PATH\_TRANSLATED** : **PATH\_INFO** を変換したバージョン。関連付けられた仮想パスと物理パス間のマッピングを持ちます。
- **QUERY\_STRING** : このスクリプトを参照する URL 内で疑問符 (?) に続く情報。この情報は、クエリー情報です。デコードしないでください。クエリー情報がある場合は、コマンドラインデコードかどうかにかかわらず、必ずこの変数を設定します。
- **REMOTE\_ADDR** : 要求を作成するリモートホストの IP アドレス。
- **REMOTE\_HOST** : 要求を作成しているホスト名。サーバにこの情報がない場合、**REMOTE\_ADDR** が設定されます。
- **REMOTE\_USER** : サーバがユーザ認証をサポートし、スクリプトが認証を必要とする場合に、認証されたユーザ名。
- **REQUEST\_METHOD** : GET または POST (表 28 (P.93) を参照) など、作成された要求のメソッド。
- **SCRIPT\_NAME** : 実行中のスクリプトへの仮想パスで、自己参照 URL に使用されます。
- **SERVER\_NAME** : サーバのホスト名、DNS エイリアス、または IP アドレスで、自己参照 URL に表示されます。
- **SERVER\_PORT** : 要求の TCP ポート番号。
- **SERVER\_PROTOCOL** : 要求の情報プロトコルの名前とリビジョンで、プロトコル/リビジョンの形式を使用します。

図 120 に、[Environment] タブのカスタマイザ ウィンドウを示します。

図 120 [Get Http Contact Info] カスタマイザ ウィンドウ : [Environment] タブ

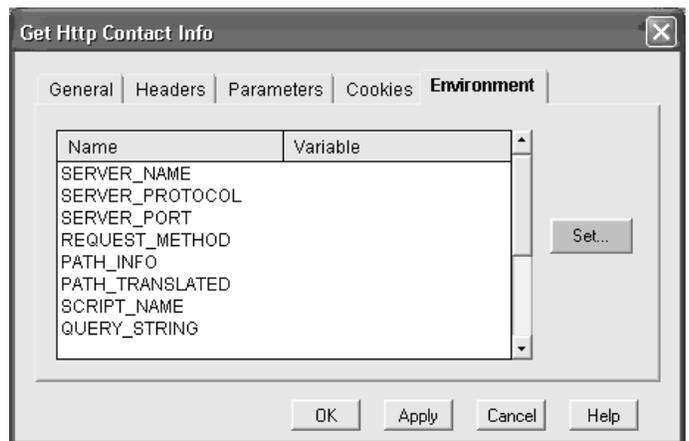


表 53 で、[Environment] タブのフィールドについて説明します。

表 53 [Get Http Contact Info] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Name	環境変数の名前。
Variable	環境変数にマッピングされている変数の名前。

## Http Redirect

**Http Redirect** ステップは、HTTP 要求に回答するのではなく、指定された URL にブラウザをリダイレクトする場合に使用します。

**Http Direct** ステップまたは **Send Response** ステップ（「[Send Response](#)」(P.128) を参照）を使用して、HTTP 要求に回答できます。両方のステップを使用しないでください。両方のステップを使用した場合、応答をブラウザに返した後で HTTP Contact スクリプトは最終状態に移行し、すでに 1 つの応答を送信したため、もう 1 つの応答を送信できなくなります。

スクリプトが条件付きロジックを使用している場合、**Http Direct** ステップと **Send Response** ステップは異なる条件で使用できます。

**Contact** パレットの **Terminate** ステップ（「[Terminate](#)」(P.77) を参照）も、HTTP 要求に回答するために使用できます。

図 121 に、**Http Redirect** ステップのカスタマイザ ウィンドウを示します。

図 121 [Http Redirect] カスタマイザ ウィンドウ



表 54 で、[Http Redirect] カスタマイザ ウィンドウのフィールドについて説明します。

表 54 [Http Redirect] カスタマイザ ウィンドウのフィールド

フィールド	説明
Http Contact	<b>Redirect</b> ステップの実行をトリガーします。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。
URL	ブラウザのリダイレクト先の URL。

## Send Response

**Send Response** ステップは、ドキュメントをブラウザに送信する場合に使用します。

ドキュメントは、送信する前に **Document** 変数に格納する必要があります（「[Document ステップ \(IVR のみ\)](#)」(P.90) を参照）。ブラウザに送信する前に、ドキュメントを動的情報で更新するには、スクリプト内で、**Send Response** ステップの前に **Text Substitution for Keywords** ステップ（「[Text Substitution for Keywords](#)」(P.94) を参照）を配置します。

**Send Response** ステップまたは **Http Direct** ステップ（「[Http Redirect](#)」(P.128) を参照）を使用して、HTTP 要求に回答できます。両方のステップは使用しないでください。両方のステップを使用した場合、応答をブラウザに返した後で HTTP Contact スクリプトは最終状態に移行し、すでに一度応答を送信しているため、次の応答を送信できなくなります。

スクリプトが条件付きロジックを使用している場合、**Http Direct** ステップと **Send Response** ステップは異なる条件で使用できます。

**Contact** パレットの **Terminate** ステップ (「[Terminate](#)」 (P.77) を参照) も、HTTP 要求に応答するために使用できます。

図 122 に、**Send Response** ステップのカスタマイザ ウィンドウを示します。

図 122 [Send Response] カスタマイザ ウィンドウ



表 55 で、[Send Response] カスタマイザ ウィンドウのフィールドについて説明します。

表 55 [Send Response] カスタマイザ ウィンドウのフィールド

フィールド	説明
Http Contact	ステップの実行をトリガーする <b>Contact</b> 変数。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。
Document	送信するドキュメントを格納する変数。

## Set Http Contact Info

**Set Http Contact Info** ステップは、HTTP 応答の HTTP ヘッダーとクッキーの値を設定する場合に使用します。



(注) 拡張スクリプトでこのステップを使用するには、HTTP に関する専門的な知識が必要です。

[Set Http Contact Info] カスタマイザ ウィンドウには、3 つのタブがあります。

- 「[\[General\] タブ](#)」 (P.129)
- 「[\[Headers\] タブ](#)」 (P.130)
- 「[\[Cookies\] タブ](#)」 (P.131)

### [General] タブ

[Set Http Contact Info] カスタマイザ ウィンドウの **[General]** タブ (図 123) は、ステップの実行をトリガーする HTTP Contact 変数を指定する場合に使用します。

図 123 [Set Http Contact Info] カスタマイザ ウィンドウ : [General] タブ



表 56 で、[General] タブのフィールドについて説明します。

表 56 [Set Http Contact Info] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Http Contact	ステップの実行をトリガーする Contact 変数。他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。

## [Headers] タブ

[Set Http Contact Info] カスタマイザ ウィンドウの [Headers] タブ (図 124) は、ローカル変数や、ステップ実行時にヘッダーの値を取得する有効な式に、各 HTTP ヘッダーをマッピングする場合に使用します。

図 124 [Set Http Contact Info] カスタマイザ ウィンドウ : [Headers] タブ

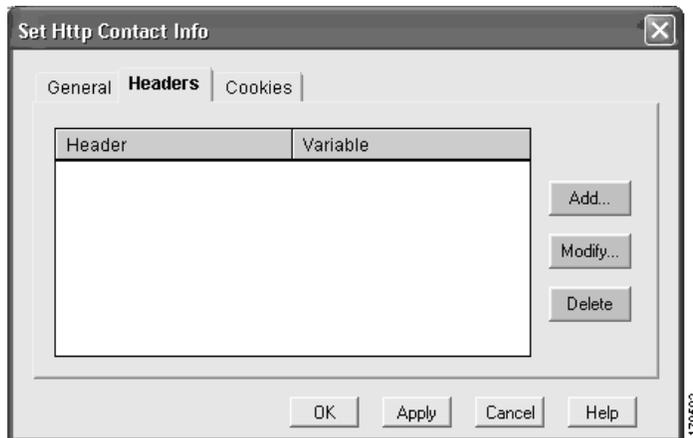


表 57 で、[Headers] タブのフィールドについて説明します。

表 57 [Set Http Contact Info] カスタマイザ ウィンドウのフィールド : [Headers] タブ

フィールド	説明
Header	ヘッダーの名前。
Variable	ヘッダーにマッピングされる変数。

## [Cookies] タブ

[Set Http Contact Info] カスタマイザ ウィンドウの [Cookies] タブ (図 125) は、ステップの実行時にクッキーの値を取得するローカル変数に各クッキーをマッピングする場合に使用します。

クッキーはブラウザが保持している情報で、HTTP 要求への応答時に HTTP サーバによって送信されます。クッキーの詳細については、「[Cookies] タブ」(P.125) を参照してください。

図 125 [Set Http Contact Info] カスタマイザ ウィンドウ : [Cookies] タブ

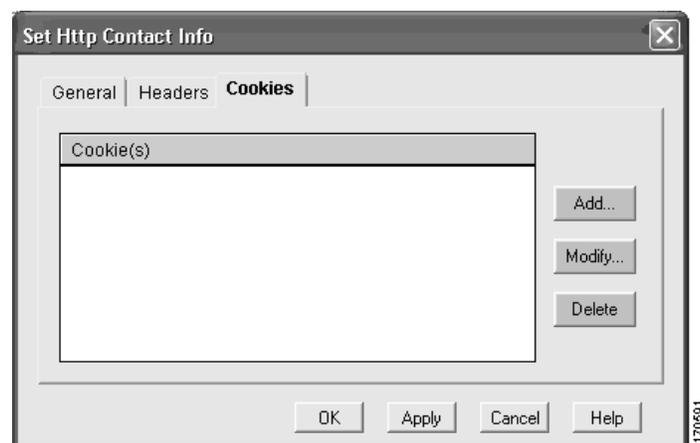


表 58 で、[Cookies] タブのフィールドについて説明します。

表 58 [Set Http Contact Info] カスタマイザ ウィンドウのフィールド : [Cookies] タブ

フィールド	説明
Cookie	変数にマッピングされているクッキーの名前。

## Media ステップ

Cisco Unity Express Script Editor の **Media** パレットのステップは、発信者とのメディア対話を処理する手段を提供します。

メディア対話には、プロンプトの再生、DTMF (Dual Tone MultiFrequency) 入力の取得などがあります。

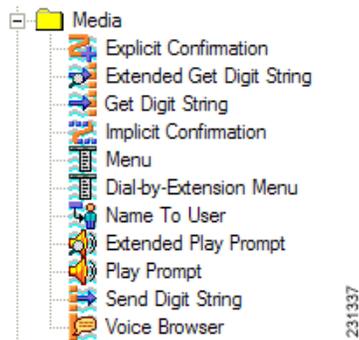
**Media** パレットには、次のステップがあります。

- 「Explicit Confirmation」(P.132)
- 「Implicit Confirmation」(P.144)
- 「Get Digit String」(P.140)
- 「Menu」(P.146)

- 「Name To User」 (P.150)
- 「Play Prompt」 (P.156)
- 「Extended Play Prompt」 (P.158)
- 「Send Digit String (IVR のみ)」 (P.160)
- 「Dial-by-Extension Menu」 (P.161)
- 「Voice Browser (IVR のみ)」 (P.166)

図 126 に、Media パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 126 Media パレットのステップ



(注)

これらの Media ステップのいずれかをメディアチャンネル (ダイアログチャンネル) に関連付けられていないコンタクトに適用すると、ChannelUnsupportedException エラーが発生します。

## Explicit Confirmation

**Explicit Confirmation** ステップは、プロンプトへの明示的な応答を確認する場合に使用します。

**Explicit Confirmation** ステップは、[Yes] として 1、[No] として 2 を受け入れます。

**Explicit Confirmation** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.132)
- 「[Prompts] タブ」 (P.133)
- 「[Input] タブ」 (P.134)

### [General] タブ

[Explicit Confirmation] カスタマイザ ウィンドウの [General] タブ (図 127) は、確認を実行するコンタクトを選択し、[Interruptible] オプションを設定する場合に使用します。

図 127 [Explicit Confirmation] カスタマイザ ウィンドウ : [General] タブ

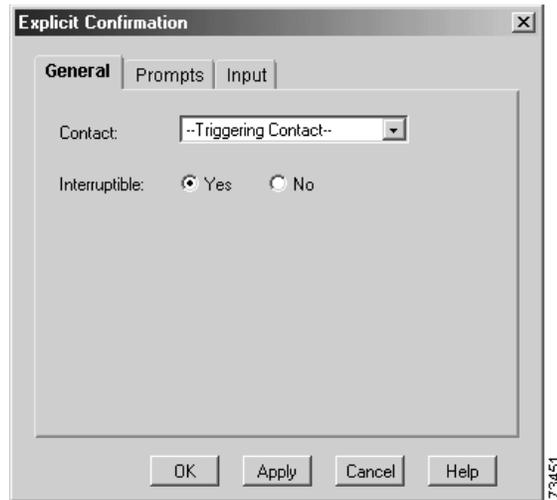


表 59 で、[General] タブのフィールドについて説明します。

表 59 [Explicit Confirmation] カスタマイザ ウィンドウのフィールド : [General] タブ

プロパティ	説明
<b>Contact</b>	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。
<b>Interruptible</b>	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompts] タブ

[Explicit Confirmation] カスタマイザ ウィンドウの [Prompts] タブ (図 128) は、開始プロンプト、エラー プロンプト、およびタイムアウト プロンプトを指定し、[Barge In] オプションおよび [Continue On Prompt Errors] オプションを設定する場合に使用します。

図 128 [Explicit Confirmation] カスタマイザ ウィンドウ : [Prompts] タブ



表 60 で、[Prompts] タブのフィールドについて説明します。

表 60 [Explicit Confirmation] カスタマイザ ウィンドウのフィールド : [Prompts] タブ

フィールド	説明
<b>Initial Prompt</b>	最初に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Error Prompt</b>	入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Timeout Prompt</b>	タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Barge In</b>	[Yes] の場合は、発信者がプロンプトに割り込むことができます。 [No] の場合は、プロンプトの再生が完了するまで、発信者は応答できません。
<b>Continue on Prompt Errors</b>	[Yes] の場合は、プロンプト エラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できません。

## [Input] タブ

[Explicit Confirmation] カスタマイザ ウィンドウの [Input] タブ (図 129) は、タイムアウトの期間、最大再試行回数、および [Flush Input Buffer] オプションを設定する場合に使用します。

図 129 [Explicit Confirmation] カスタマイザ ウィンドウ : [Input] タブ



表 61 で、[Input] タブのフィールドについて説明します。

表 61 [Explicit Confirmation] カスタマイザ ウィンドウのフィールド : [Input] タブ

フィールド	説明
Timeout (in sec)	タイムアウトを発生させるまで、ステップが応答を待つ秒数。
Maximum Retries	タイムアウトまたは無効なキーの後で入力できる新しいエントリの回数。
Flush input Buffer	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。 [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。
Grammar	[Yes] または [No] の認識に使用するオプションの文法式。指定した場合、システムのデフォルトの文法が上書きされます。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用して文法式を入力します。

## Extended Get Digit String ステップ

**Extended Get Digit String** ステップは、スクリプト内の DTMF エントリをキャプチャする場合に使用します。このステップは、タイムアウトに達するまで、または発信者が次のいずれかを実行するまで入力を待機します。

- 終了キーを押す。
- 最大再試行回数に達する。
- 入力したキーの数が最大数に達する。

**Extended Get Digit String** ステップは、次の場合を除き、既存の **Get Digit String** ステップとまったく同様に動作します。

- **Get Digit String** ステップが、[Interruptible] フィールドと [Clear DTMF Buffer on Retry] フィールドの Boolean 式を提供します。

- どちらのステップにも同じ制限が適用されますが、スクリプトの実行中に変更可能な変数を使用して、Extended Get Digit String ステップのほとんどのプロパティを定義できます。

このステップのカスタマイザの [Terminating Key] フィールドと [Cancel Key] フィールドは、Character オブジェクトと String オブジェクトをサポートしています。終了キーまたはキャンセル キーとして文字列が渡された場合、その文字列の先頭の文字のみが使用されます。特殊な文字列 *none* を使用すると、終了キーもキャンセル キーも必要ないことを指定できます。

Extended Get Digit String ステップには、次の出力ブランチがあります。

- **Successful** : 入力が有効だった。
- **Timeout** : 再試行の制限に達した後、最終試行がタイムアウトした。
- **Unsuccessful** : 再試行の制限に達した後、無効なキーが押された。

このステップがエラー (Timeout または Unsuccessful) を返した場合は、収集されたディジットがすべて返され、指定した入力変数に格納されます。

Extended Get Digit String ステップのカスタマイザ ウィンドウには、4 つのタブがあります。

- 「[General] タブ」 (P.136)
- 「[Prompt] タブ」 (P.137)
- 「[Input] タブ」 (P.138)
- 「[DTMF Control] タブ」 (P.139)

## [General] タブ

Extended Get Digit String ステップの [General] タブ (図 134) は、コンタクトを選択し、ディジット文字列を格納する変数を指定し、外部のイベントがステップに割り込むことができるかどうかを指定する場合に使用します。

図 130 [Extended Get Digit String] カスタマイザ ウィンドウ : [General] タブ



表 62 で、[General] タブのフィールドについて説明します。

表 62 [Extended Get Digit String] カスタマイザ ウィンドウ : [General] タブ

プロパティ	説明
Contact	他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。
Result Digit String	結果として得られるディジット文字列を保持する変数。
Interruptible	true の場合は、発信者によって通話がリモート切断されるなどの外部のイベントが、ステップに割り込むことができます。false の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompt] タブ

[Get Digit String] カスタマイザ ウィンドウの [Prompt] タブ (図 135) は、プロンプトを指定し、[Barge In] オプションおよび [Continue on Prompt Errors] オプションを設定する場合に使用します。

図 131 [Extended Play Prompt] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

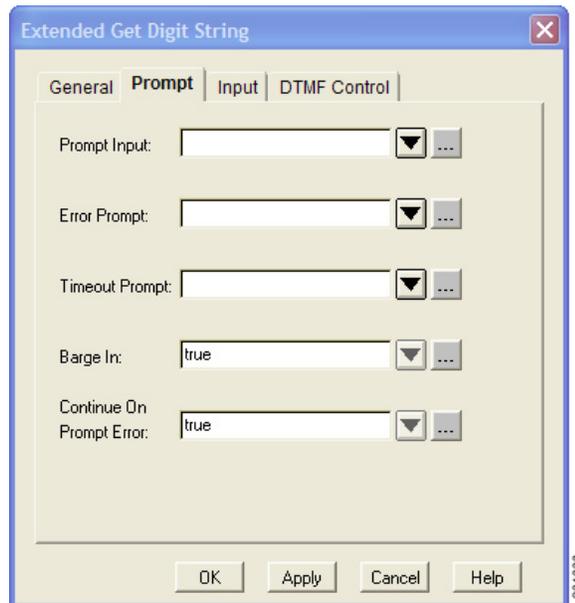


表 63 で、[Prompt] タブのフィールドについて説明します。

表 63 [Extended Get Digit String] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

プロパティ	説明
Prompt Input	発信者に対して再生され、ディジット文字列の入力を要求するプロンプト。
Error Prompt	入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
Timeout Prompt	タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。

表 63 [Extended Get Digit String] カスタマイザ ウィンドウのフィールド : [Prompt] タブ (続き)

プロパティ	説明
<b>Barge In</b>	true の場合、発信者はプロンプトに割り込むことができます。false の場合は、発信者の入力を受け入れる前にプロンプトの再生を完了する必要があります。
<b>Continue On Prompt Error</b>	[Yes] の場合、ステップはリストの次のプロンプトに進みます。これが最終プロンプトの場合は、発信者からの入力を待機します。[No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Exnteded Get Digit String] カスタマイザ ウィンドウの [Input] タブ (図 132) は、発信者の入力を受信する条件を設定する場合に使用します。

図 132 [Extended Get Digit String] カスタマイザ ウィンドウ : [Input] タブ

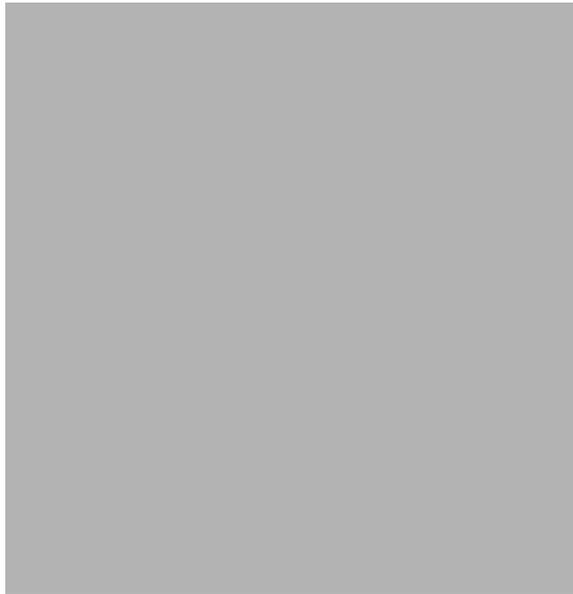


表 64 で、[Filter] タブのフィールドについて説明します。

表 64 [Extended Get Digit String] カスタマイザ ウィンドウのフィールド : [Input] タブ

プロパティ	説明
<b>Input Length</b>	このステップが受け入れるディジットの最大数。この制限に達すると、ステップは Successful を返します。
<b>Maximum Retries</b>	タイムアウトまたは無効なキーの後で再開できるエントリの回数。値 0 は再試行できず、スクリプトが再試行シナリオを処理する必要があることを意味します。
<b>Initial Timeout</b>	[Yes] の場合、ステップはリストの次のプロンプトに進みます。これが最終プロンプトの場合は、発信者からの入力を待機します。[No] の場合は、例外が発生します。例外はスクリプトで処理できます。

表 64 [Extended Get Digit String] カスタマイザ ウィンドウのフィールド : [Input] タブ (続き)

プロパティ	説明
Interdigit Timeout	発信者からの最初のディジットを受信した後、次のディジットの入力を待つ時間 (秒)。
Flush Input Buffer	式の評価が true の場合、それ以前に入力されたディジットを破棄してから、このステップを実行します。
Clear DTMF Buffer on Retry	式の評価が true の場合、各再試行の前にスクリプトが DTMF バッファをクリアします。

## [DTMF Control] タブ

[Extended Get Digit String] カスタマイザ ウィンドウの [DTMF Control] タブ (図 133) は、終了キー、キャンセル キー、入力フィルタを設定する場合に使用します。

図 133 [Extended Get Digit String] カスタマイザ ウィンドウ : [DTMF Control] タブ

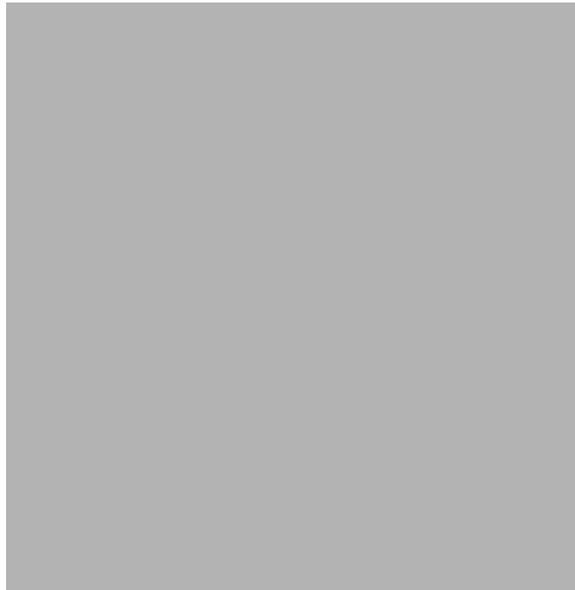


表 65 で、[DTMF Control] タブのフィールドについて説明します。

表 65 [Extended Get Digit String] カスタマイザ ウィンドウ : [DTMF Control] タブ

プロパティ	説明
Terminating Key	発信者の入力の末尾を示すキー。入力を終了するときに、終了キーは Maximum Input Length よりも優先されます。このフィールドを空のままにしておくか、none または n に設定すると、終了キーが存在しないことを意味します。
Cancel Key	再開するために発信者が押すことのできるキー。このフィールドを空のままにしておくか、none または n に設定すると、キャンセル キーが存在しないことを意味します。
Input Filter	入力できる有効な DTMF キー (終了キーとキャンセル キーを除く) を定義する式。

## Get Digit String

**Get Digit String** ステップは、発信者がプロンプトに回答した DTMF デジット文字列をキャプチャする場合に使用します。

**Get Digit String** ステップは、発信者が次のいずれかを実行するまで入力を待ちます。

- 終了キーを押す (DTMF のみ)。
- 最大再試行回数に達する。
- 入力したキーの数が最大に達する (DTMF のみ)。
- タイムアウト長に達するまで応答しない。



(注)

スクリプトの以前のエスカレーティング プロンプトが **Get Digit String** ステップに入ると、以前のエスカレーティング プロンプトはリスト中の最初のプロンプトにリセットされます。

**Get Digit String** ステップには、3 つの出力ブランチがあります。

- **Successful** : 入力が有効だった。
- **Timeout** : 再試行の制限に達した後、最終試行がタイムアウトした。
- **Unsuccessful** : 再試行の制限に達した後、無効なキーが押された。



(注)

エラーが発生した場合、累積されたデジットが返され、**Unsuccessful** または **Timeout** 出力ブランチを経由してスクリプトが終了する前に、指定された変数に保存されます。

**Get Digit String** ステップのカスタマイザ ウィンドウには、4 つのタブがあります。

- 「[General] タブ」 (P.140)
- 「[Prompt] タブ」 (P.141)
- 「[Input] タブ」 (P.142)
- 「[Filter] タブ」 (P.144)

### [General] タブ

**Get Digit String** ステップの [General] タブ (図 134) は、コンタクトを選択し、デジット文字列を格納する変数を指定し、外部のイベントがステップに割り込むことができるかどうかを指定する場合に使用します。

図 134 [Get Digit String] カスタマイザ ウィンドウ : [General] タブ

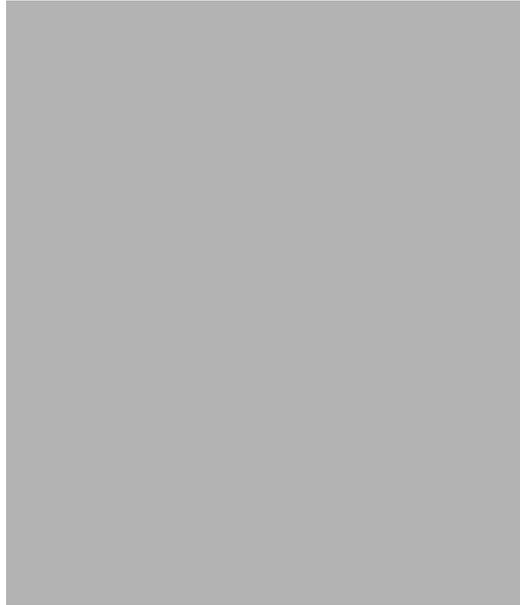


表 66 で、[General] タブのフィールドについて説明します。

表 66 [Get Digit String] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Contact	ステップの実行をトリガーするコンタクト。 他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。
Result Digit String	発信者が入力したディジットを格納する変数の名前。
Interruptible	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompt] タブ

[Get Digit String] カスタマイザ ウィンドウの [Prompt] タブ (図 135) は、プロンプトを指定し、[Barge In] オプションおよび [Continue on Prompt Errors] オプションを設定する場合に使用します。

図 135 [Get Digit String] カスタマイザ ウィンドウ : [Prompt] タブ

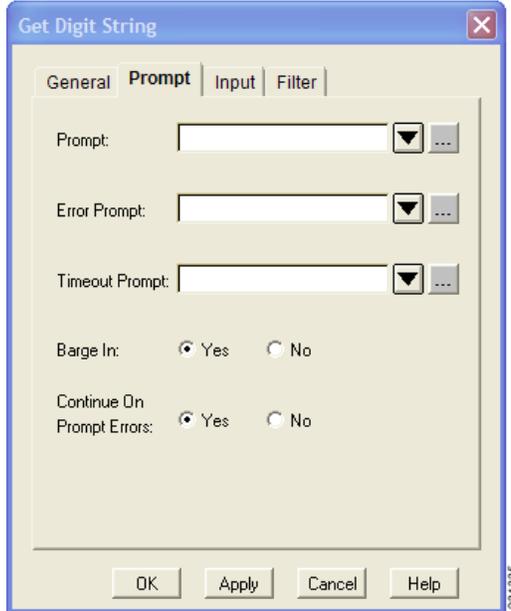


表 67 で、[Prompt] タブのフィールドについて説明します。

表 67 [Get Digit String] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

フィールド	説明
<b>Prompt</b>	再生されるプロンプト。
<b>Error Prompt</b>	入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Timeout Prompt</b>	タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Barge In</b>	[Yes] の場合は、発信者がプロンプトに割り込むことができます。 [No] の場合は、プロンプトの再生が完了するまで、発信者は応答できません。
<b>Continue on Prompt Errors</b>	[Yes] の場合は、プロンプトエラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Get Digit String] カスタマイザ ウィンドウの [Input] タブ (図 136) は、発信者の入力を受信する条件を設定する場合に使用します。

図 136 [Get Digit String] カスタマイザ ウィンドウ : [Input] タブ

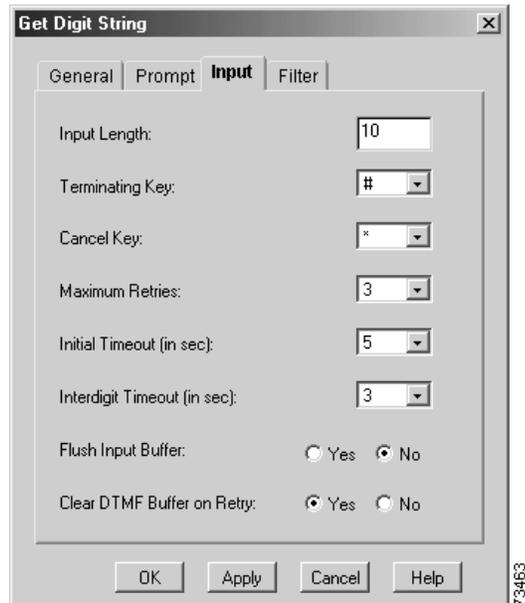


表 68 で、[Input] タブのフィールドについて説明します。

表 68 [Get Digit String] カスタマイザ ウィンドウのフィールド : [Input] タブ

フィールド	説明
<b>Input Length</b>	ディジットまたは文字の最大数。この制限に達すると、ステップは停止し、ディジットを累積して返します。
<b>Terminating Key</b>	発信者の入力の末尾を示すキー (DTMF のみ)。入力を終了するときに、終了キーは <b>Input Length</b> よりも優先されます。
<b>Cancel Key</b>	発信者が最初からやり直すときに押すキー。
<b>Maximum Retries</b>	タイムアウトまたは無効なキーの後で入力できる新しいエントリの回数。 最大再試行回数に達した後、最後の試行がタイムアウトだったか、無効なキーが入力されたかによって、ステップは <b>Timeout</b> 出力ブランチまたは <b>Unsuccessful</b> 出力ブランチに進みます。無効なキーによる再試行時には、システムプロンプトが再生されます。 値「0」は、再試行を許可しないことを意味します。この場合、スクリプトが再試行シナリオを処理する必要があります。
<b>Initial timeout (in sec)</b>	発信者からの最初の入力を待つ秒数。
<b>Interdigit timeout (in sec)</b>	発信者からの最初の入力を受信した後、次のディジットの入力を待つ秒数 (DTMF)。
<b>Flush Input Buffer</b>	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。 [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。
<b>Clear DTMF Buffer on Retry</b>	[Yes] の場合は、各再試行の前に DTMF バッファをクリアします。 [No] の場合は、各再試行の前に DTMF バッファをクリアしません。

## [Filter] タブ

[Get Digit String] カスタマイザ ウィンドウの [Filter] タブ (図 137) は、発信者から受け付けるディジットを指定する場合に使用します。

発信者から受け付けるディジットを指定するには、目的のディジットのチェックボックスをオンにして、[OK] をクリックします。

[Get Digit String] カスタマイザ ウィンドウが閉じます。トリガーするコンタクトの名前と結果のディジット文字列変数が、[Design] ペインの **Get Digit String** ステップ アイコンの横に表示されます。

図 137 [Get Digit String] カスタマイザ ウィンドウ : [Filter] タブ

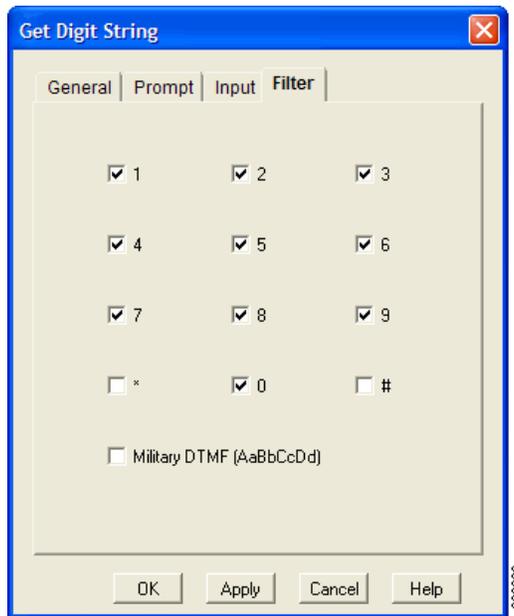


表 69 で、[Filter] タブのフィールドについて説明します。

表 69 [Get Digit String] カスタマイザ ウィンドウのフィールド : [Filter] タブ

フィールド	説明
ディジット選択チェックボックス	[Filter] タブを使用して、発信者から受け付けるディジットを指定します (終了キーおよびキャンセルキー以外)。選択しなかったディジットを発信者が入力した場合、システムは発信者に対してエラー プロンプトを再生し、最大再試行回数に達するまで、 <b>Input</b> ステップを再試行します。最大再試行回数に達した場合は、 <b>Unsuccessful</b> 出力ブランチが実行されます。

## Implicit Confirmation

**Implicit Confirmation** ステップは、質問をせずにアクションを確認する場合に使用します。

アクションを説明するプロンプトが再生され、設定された秒数、発信者からの入力を待ちます。設定されたタイムアウトより前に発信者が任意の DTMF デイジットを押した場合、確認は失敗したと見なされ、**Explicit Confirmation** ステップが使用されます。



(注)

スクリプトの以前のエスカレーティングプロンプトが **Implicit Confirmation** ステップに入ると、以前のエスカレーティングプロンプトはリスト中の最初のプロンプトにリセットされます。

たとえば、有効なディジット文字列を受信したときに、発信者の入力に基づいて、ダイヤルする内線をプロンプトで再生します。この例の **Implicit Confirmation** ステップは、発信者がプロンプトを聞いた後、タイムアウトの前に確認を拒否する時間として 2 秒間待つように設定されます。

**Implicit Confirmation** ステップの **No** 出力ブランチの下で、**If** ステップによって、スクリプトが後続のステップに進むまでの確認回数を追跡します。

発信者に対して再生された内線が正確で、発信者が操作を停止しなかった場合は、**Yes** 出力ブランチが実行され、**Call Redirect** ステップによって目的の内線への接続が試行されます。

図 138 に、**Implicit Confirmation** ステップのカスタマイザ ウィンドウを示します。

図 138 [Implicit Confirmation] カスタマイザ ウィンドウ

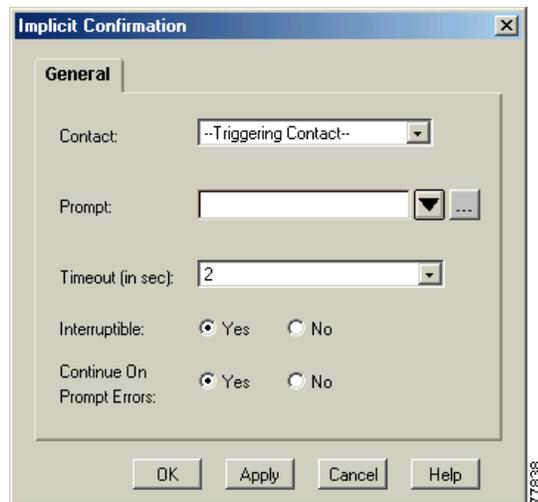


表 70 で、[Implicit Confirmation] カスタマイザ ウィンドウのフィールドについて説明します。

表 70 [Implicit Confirmation] カスタマイザ ウィンドウのフィールド

フィールド	説明
<b>Contact</b>	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。トリガーするコンタクトの名前が、[Design] ペインの <b>Implicit Confirmation</b> ステップアイコンの横に表示されます。
<b>Prompt</b>	発信者に対して再生されるプロンプト。
<b>Timeout (in secs)</b>	確認が成功したと見なされるまでの、発信者が応答しない秒数。通常、この値は 2 秒です。

表 70 [Implicit Confirmation] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
<b>Interruptible</b>	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。
<b>Continue on Prompt Errors</b>	[Yes] の場合は、プロンプト エラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できません。

## Menu

**Menu** ステップは、発信者が一連のオプションを選択できるメニューを提供する場合に使用します。  
**Menu** ステップは、発信者が入力した単一ディジットを受信し、この入力を一連のオプション出力ブランチにマッピングします。システムは、これらの各オプション出力ブランチの後に追加されたステップを実行します。



(注)

スクリプトの以前のエスカレーティング プロンプトが **Menu** ステップに入ると、以前のエスカレーティング プロンプトはリスト中の最初のプロンプトにリセットされます。

**Menu** ステップは、**Get Digit String** ステップと **Switch** ステップの機能を組み合わせたものですが、発信者が入力できるディジットは 1 つだけです。

デフォルトでは、**Menu** ステップには次の出力ブランチがあります。

- **Output 1**
- **Output 2**
- **Output 3**
- **Timeout**
- **Unsuccessful**

[Menu] カスタマイザ ウィンドウの [General] タブで、さらに出力ブランチを追加できます。

**Menu** ステップは、タイムアウトまたは無効なディジット入力（どの接続にも関連付けられていないディジット）が発生した場合に、再試行します。最大再試行回数に達すると、最後の失敗の理由に応じて、**Menu** ステップは **Timeout** 接続または **Unsuccessful** 接続に進みます。

**Menu** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.146)
- 「[Prompt] タブ」 (P.148)
- 「[Input] タブ」 (P.149)

## [General] タブ

[Menu] カスタマイザ ウィンドウの [General] タブ (図 139) は、ディジット（通常、発信者が電話のキーボードから入力）を出力ブランチ ラベルに関連付ける場合に使用します。

複数の入力を 1 つの出力ブランチ ラベルに関連付けることができますが、1 つの入力に関連付けることができる出力ブランチ ラベルは 1 つだけです。

図 139 [Menu] カスタマイザ ウィンドウ : [General] タブ

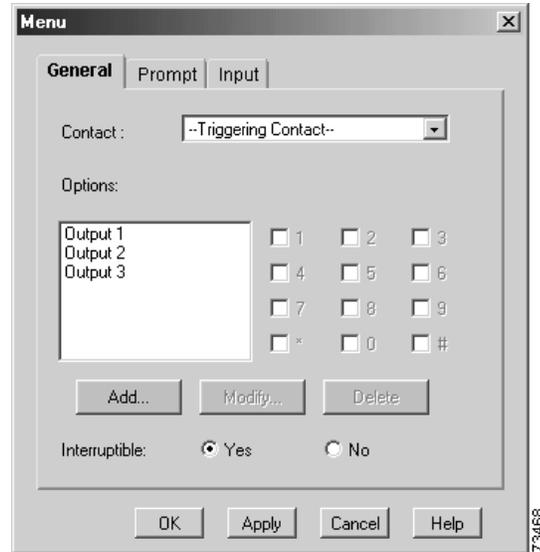


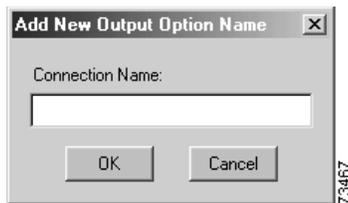
表 71 で、[General] タブのフィールドについて説明します。

表 71 [Menu] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
<b>Contact</b>	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。
<b>Options</b>	出力値ごとのラベル。
<b>Add</b>	新しいオプションを追加します。新しいオプションが [Options] リストボックスに表示されます。
<b>Modify</b>	[Rename Output Option] ダイアログボックスを使用して、選択されたオプションを変更します。このダイアログボックスには、[Add New Output Option Name] ダイアログボックスと同じフィールドがあり、同じ方法で設定します。
<b>Interruptible</b>	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

[Add New Output Option Name] ダイアログボックスを [図 140](#) に示します。

図 140 [Add New Output Option Name] ダイアログボックス



## [Prompt] タブ

[Menu] カスタマイザ ウィンドウの [Prompt] タブ (図 141) は、再生するプロンプトを選択し、[Barge In] オプションおよび [Continue on Prompt Errors] オプションを設定する場合に使用します。

図 141 [Menu] カスタマイザ ウィンドウ : [Prompt] タブ

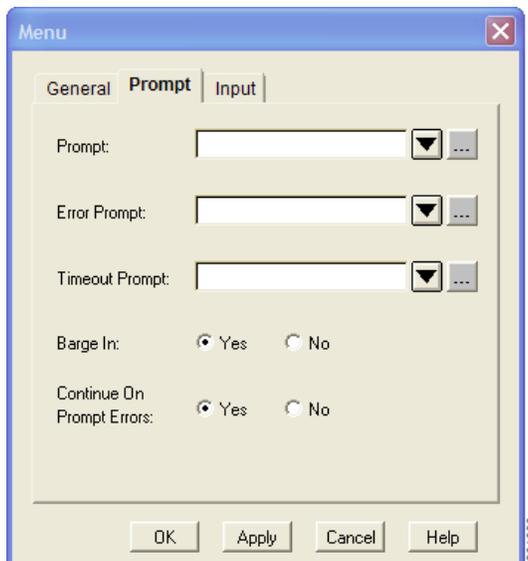


表 72 で、[Prompt] タブのフィールドについて説明します。

表 72 [Menu] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

フィールド	説明
<b>Prompt</b>	発信者に対して再生されるプロンプトを指定します。
<b>Error Prompt</b>	入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
<b>Timeout Prompt</b>	タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。

表 72 [Menu] カスタマイザ ウィンドウのフィールド : [Prompt] タブ (続き)

フィールド	説明
Barge In	[Yes] の場合は、発信者がプロンプトに割り込むことができます。 [No] の場合は、プロンプトの再生が完了するまで、発信者は応答できません。
Continue on Prompt Errors	[Yes] の場合は、プロンプト エラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Menu] カスタマイザ ウィンドウの [Input] タブ (図 142) は、タイムアウトの設定、最大再試行回数、および [Flush Input Buffer] オプションを設定する場合に使用します。

図 142 [Menu] カスタマイザ ウィンドウ : [Input] タブ

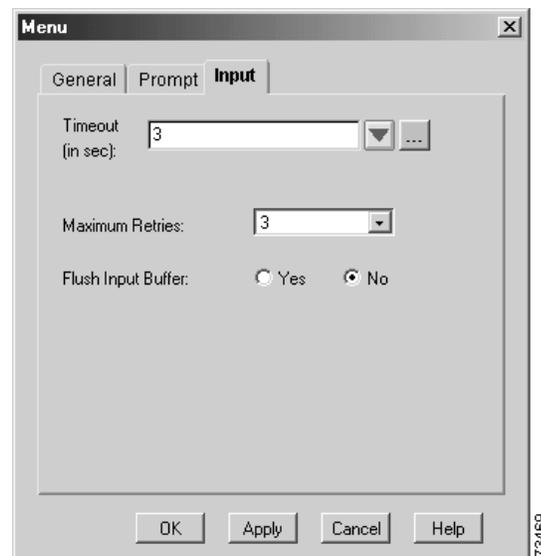


表 73 で、[Input] タブのフィールドについて説明します。

表 73 [Menu] カスタマイザ ウィンドウのフィールド : [Input] タブ

プロパティ	説明
Timeout	発信者からの入力を待つ時間。このタイマーが経過すると、システムはプロンプトをもう一度再生するか、発信者がいるかどうかを確認するシステム プロンプトを再生します。値を入力するか、タイムアウト値を格納する変数を [Timeout] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして数字を入力します。
Maximum Retries	タイムアウトまたは無効な入力応答の後で再開できるエントリの回数。最大再試行回数に達した後、最後の試行がタイムアウトだったか、無効な入力応答が入力されたかによって、Menu ステップは Timeout 出力ブランチまたは Unsuccessful 出力ブランチに進みます。  値「0」は、再試行を許可しないことを意味します。この場合、スクリプトが再試行シナリオを処理する必要があります。
Flush Input Buffer	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。  [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。

## Name To User

**Name To User** ステップは、通常、着信側の人の名前を (DTMF で) 発信者に要求し、ディレクトリに格納されている名前と比較するために使用します。**Name To User** ステップは、自動的に発信者を着信側の内線に転送するスクリプトでよく使われます。

**Name To User** ステップのもう 1 つの便利な機能として、後で **Get User Info** ステップを使用して、発信者が選択したユーザの内線、電子メールアドレス、音声名などの情報を取得できるように、値を変数に割り当てる機能があります。



(注)

スクリプトの以前のエスカレーティング プロンプトが **Name To User** ステップに入ると、以前のエスカレーティング プロンプトはリスト中の最初のプロンプトにリセットされます。

**Name To User** ステップは、次の数値キーパッド マッピングを使用して、発信者からの DTMF 入力を受信します。

- 2 = ABC
- 3 = DEF
- 4 = GHI
- 5 = JKL
- 6 = MNO
- 7 = PQRS
- 8 = TUV
- 9 = WXYZ

このステップの情報を使用して、発信者が選択したユーザが存在する場合、スクリプトはそのユーザのあらかじめ録音された名前を再生する後続のプロンプトを作成します。録音がない場合、スクリプトはユーザの名前のスペルを読み上げます。



(注) **Name To User** ステップがスペルを読み上げる名前は、ASCII 文字だけの名前です。そのため、国によっては、この機能が制限される場合があります。

**Name To User** ステップは、次の出力ブランチを生成します。



(注) **Name To User** ステップで、発信者の入力 **Lightweight Access Directory Protocol (LDAP)** ディレクトリの単一ユーザと一致した場合、発信者が選択を確認する必要なしに、その結果が即座に返されます。

- **Successful** : 発信者からの入力とディレクトリの名前が一致した。
- **Timeout** : 発信者からの入力を受信せずに、ステップの最大再試行回数 (カスタマイザ ウィンドウで設定) に達した。
- **Unsuccessful** : 発信者からの入力 **ディレクトリ** の名前と一致しなかった。
- **Operator** : オペレータの内線が入力された。



(注) **Operator** 出力ブランチは、[Name To User] カスタマイザ ウィンドウの [General] タブの [Operator] オプションを [Yes] に設定した場合にだけ、スクリプトの **Name To User** ステップの下に表示されます (図 143 を参照)。

**Name To User** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.151)
- 「[Prompt] タブ」 (P.152)
- 「[Input] タブ」 (P.154)

## [General] タブ

[General] タブ (図 143) は、**Result User** 変数を指定し、**Name To User** ステップのその他のプロパティを設定する場合に使用します。

図 143 [Name To User] カスタマイザ ウィンドウ : [General] タブ



表 74 で、[General] タブのフィールドについて説明します。

**Name To User** ステップの [General] タブのフィールドを設定するには、次の手順を実行します。

表 74 [Name To User] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
<b>Contact</b>	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。トリガーするコンタクトの名前と、 <b>Result User</b> 変数の名前が、[Design] ペインに表示されます。
<b>Result User</b>	発信者が選択したユーザを表すユーザ オブジェクトを格納する変数。
<b>Announce When Number of Matches Less Than</b>	一致した数がこの値より少ない場合、一致リストから正しいエントリを選択するように要求するプロンプトを再生します。一致した数がこの値以上の場合、一致数を減らすために追加の文字を入力するように要求するプロンプトを再生します。
<b>Name Search Order</b>	検索で発信者の姓または名のどちらを使用するかを選択します。
<b>Operator</b>	[Yes] の場合、発信者は「0」を押してオペレータに接続できます。 [No] の場合、発信者はオペレータに接続できません。
<b>Interruptible</b>	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompt] タブ

[Prompt] タブ (図 144) は、**Name To User** ステップで再生するプロンプトを指定し、[Barge In] オプションおよび [Continue on Prompt Errors] オプションを設定する場合に使用します。

図 144 [Name To User] カスタマイザ ウィンドウ : [Prompt] タブ

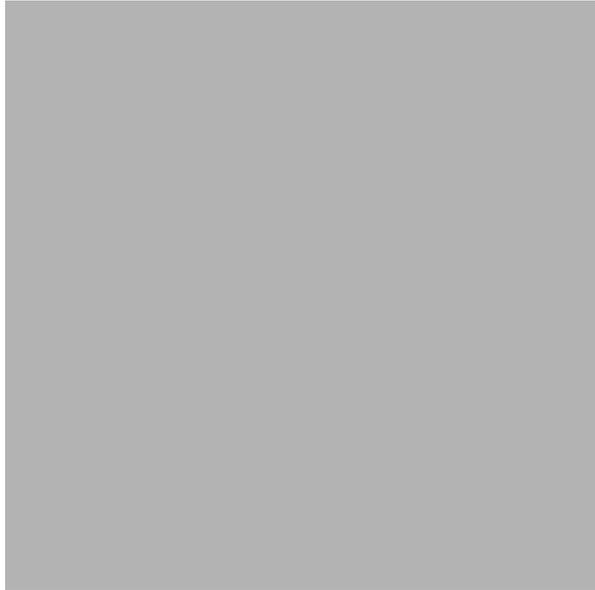


表 75 で、[Prompt] タブのフィールドについて説明します。

表 75 [Name To User] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

フィールド	説明
Prompt	<p>発信者に対して再生されるプロンプトを指定します。</p> <p>ドロップダウン メニューから、次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> <li>• <b>Default prompt</b> : Cisco Unity Express ソフトウェアに組み込まれている「Spell the last name followed by the first name」というシステム プロンプト。</li> <li>• <b>Customized prompt</b> : スクリプト設計者が作成したプロンプト。</li> <li>• <b>No prompt</b> : プロンプトを再生しない。</li> </ul> <p>値を入力するか、[List of Prompts] ドロップダウン メニューから再生するプロンプトを選択するか、または [Expression Editor (...)] ボタンをクリックして再生するプロンプトを指定する式を入力します。</p>
Error Prompt	<p>入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウン メニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。</p>
Timeout Prompt	<p>タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウン メニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。</p>

表 75 [Name To User] カスタマイザ ウィンドウのフィールド : [Prompt] タブ (続き)

フィールド	説明
<b>Barge In</b>	[Yes] の場合は、発信者がプロンプトに割り込むことができます。 [No] の場合は、プロンプトの再生が完了するまで、発信者は応答できません。
<b>Continue on Prompt Errors</b>	[Yes] の場合は、プロンプトエラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Input] タブ (図 145) は、Name To User ステップのさまざまな入力プロパティを設定する場合に使用します。

図 145 [Name To User] カスタマイザ ウィンドウ : [Input] タブ



表 76 で、[Input] タブのフィールドについて説明します。

表 76 [Name To User] カスタマイザ ウィンドウのフィールド : [Input] タブ

プロパティ	説明
<b>Input Length</b>	発信者の入力のチェックを自動的に開始するまでに必要な最小ディジット数。
<b>Terminating Key</b>	発信者の入力の末尾を示すキー。
<b>Cancel Key</b>	再開するために発信者が押すキー。 <b>Cancel</b> キーが機能するのは、最大試行回数に達するまでです。

表 76 [Name To User] カスタマイザ ウィンドウのフィールド : [Input] タブ (続き)

プロパティ	説明
Maximum Retries	ステップが有効な入力の受信を試行する回数。値「0」は、再試行を許可しないことを意味します。この場合、スクリプトが再試行シナリオを処理する必要があります。
Initial Timeout (in sec)	発信者からの最初の入力を待つ秒数。
Interdigit Timeout (in sec)	発信者からの最初の入力を受信した後、次のディジットの入力を待つ秒数。
Flush Input Buffer	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。 [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。

## [Dialog] タブ

[Dialog] タブ (図 146) は、Name To User ステップのさまざまな入力プロパティを設定する場合に使用します。

図 146 [Name To User] カスタマイザ ウィンドウ : [Dialog] タブ



表 76 で、[Dialog] タブのフィールドについて説明します。

表 77 [Name To User] カスタマイザ ウィンドウのフィールド : [Input] タブ

プロパティ	説明
No match prompt	発信者の入力チェック後、一致するものが見つからなかった場合に再生されるプロンプトを指定します。
Too many matches prompt	発信者の入力チェック後、見つかった一致が多すぎる場合に再生されるプロンプトを指定します。

表 77 [Name To User] カスタマイザ ウィンドウのフィールド : [Input] タブ (続き)

プロパティ	説明
Selection prompt	発信者が別の選択を行う必要がある場合に再生されるプロンプトを指定します。
Invalid selection prompt	発信者が認識されない選択を行った場合に再生されるプロンプトを指定します。
Cancel key help prompt	発信者がキャンセル ヘルプ キーを選択したときに再生されるプロンプトを指定します。
Operator key help prompt	発信者がオペレータ ヘルプ キーを選択したときに再生されるプロンプトを指定します。

## Play Prompt

**Play Prompt** ステップは、発信者に対して指定したプロンプトを再生する場合に使用します。



(注)

スクリプトの以前のエスカレーティング プロンプトが **Play Prompt** ステップに入ると、以前のエスカレーティング プロンプトはリスト中の最初のプロンプトにリセットされます。

**Play Prompt** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.156)
- 「[Prompt] タブ」 (P.157)
- 「[Input] タブ」 (P.157)

## [General] タブ

[General] タブ (図 147) は、コンタクトを識別し、[Interruptible] オプションを設定する場合に使用します。

図 147 [Play Prompt] カスタマイザ ウィンドウ : [General] タブ



表 78 で、[General] タブのフィールドについて説明します。

表 78 [Play Prompt] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Contact	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。トリガーするコンタクトの名前と Prompt 変数の名前が、[Design] ペインに表示されます。
Interruptible	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。 [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompt] タブ

[Play Prompt] カスタマイザ ウィンドウの [Prompt] タブ (図 148) は、再生するプロンプトを指定し、[Barge In] オプションおよび [Continue on Prompt Errors] オプションを設定する場合に使用します。

図 148 [Play Prompt] カスタマイザ ウィンドウ : [Prompt] タブ



表 79 で、[Prompt] タブのフィールドについて説明します。

表 79 [Play Prompt] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

フィールド	説明
Prompt	再生するプロンプトを指定します。
Barge In	[Yes] の場合は、発信者がプロンプトに割り込むことができます。 [No] の場合は、プロンプトの再生が完了するまで、発信者は応答できません。
Continue on Prompt Errors	[Yes] の場合は、プロンプト エラーが発生した場合に、リストにある次のプロンプトでステップを継続します。このプロンプトがリストの最後の場合は、発信者の入力を待ちます。 [No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

Play Prompt ステップの [Input] タブ (図 149) は、発信者の入力をキャプチャする前に、前に入力された入力を消去するかどうかを指定する場合に使用します。

図 149 [Play Prompt] カスタマイザ ウィンドウ : [Input] タブ

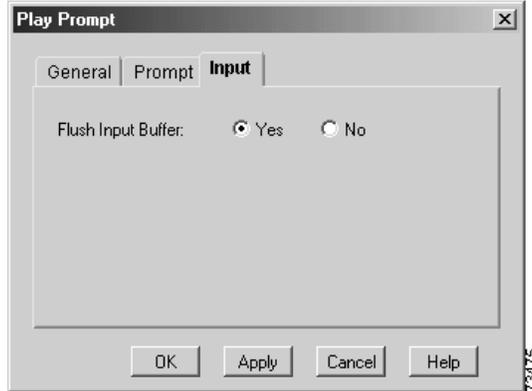


表 80 で、[Input] タブのフィールドについて説明します。

表 80 [Play Prompt] カスタマイザ ウィンドウのフィールド : [Input] タブ

フィールド	説明
Flush Input Buffer	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。 [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。

## Extended Play Prompt

**Extended Play Prompt** ステップは、発信者に対してプロンプトを再生する場合に使用します。このステップを Play Prompt ステップの代わりに使用すると、Expression Editor を使用して条件付きテストを含めることができます。

Extended Play Prompt ステップには、出力ブランチはありません。

**Extended Play Prompt** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.158)
- 「[Prompt] タブ」 (P.159)
- 「[Input] タブ」 (P.160)

### [General] タブ

[Extended Play Prompt] カスタマイザ ウィンドウの [General] タブ (図 150) は、確認を実行するコンタクトを選択し、[Interruptible] オプションを設定する場合に使用します。

図 150 [Extended Play Prompt] カスタマイザ ウィンドウ : [General] タブ



表 81 で、[General] タブのフィールドについて説明します。

表 81 [Extended Play Prompt] カスタマイザ ウィンドウ : [General] タブ

プロパティ	説明
Contact	他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。
Interruptible	true の場合は、発信者によって通話がリモート切断されるなどの外部のイベントが、ステップに割り込むことができます。false の場合は、このステップが完了するまで、他のプロセスを実行できません。

## [Prompt] タブ

[Extended Play Prompt] カスタマイザ ウィンドウの [Prompt] タブ (図 151) は、Prompt を設定し、Barge In 条件および Continue On Prompt Error 条件を設定する場合に使用します。

図 151 [Extended Play Prompt] カスタマイザ ウィンドウ : [Prompt] タブ

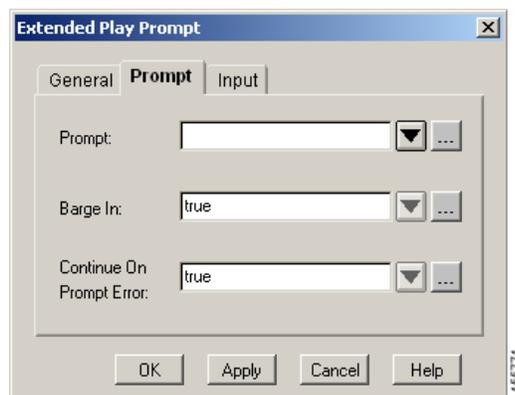


表 82 で、[Prompt] タブのフィールドについて説明します。

表 82 [Extended Play Prompt] カスタマイザ ウィンドウ : [Prompt] タブ

プロパティ	説明
Prompt	再生するプロンプトを指定します。
Barge In	true の場合は、発信者がプロンプトに割り込むことができます。false の場合は、発信者の入力を受け入れる前にプロンプトの再生を完了する必要があります。
Continue On Prompt Error	[Yes] の場合は、リストの次のプロンプトに進みます。[No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Extended Play Prompt] カスタマイザ ウィンドウの [Input] タブ (図 152) は、Flush Input Buffer 条件を設定する場合に使用します。

図 152 [Extended Play Prompt] カスタマイザ ウィンドウ : [Input] タブ



表 83 で、[Prompt] タブのフィールドについて説明します。

表 83 [Extended Play Prompt] カスタマイザ ウィンドウのフィールド : [Prompt] タブ

プロパティ	説明
Flush Input Buffer	式の評価が true の場合は、それ以前に入力されたディジットを破棄してから、このステップを実行します。

## Send Digit String (IVR のみ)

**Send Digit String** ステップは、指定された DTMF デジットのセットを発信者にアウトパルスまたは返信する場合に使用します。

DTMF デジットはゲートウェイによって、アウトオブバンドで送信され、インバンドではアウトパルスされます。



(注)

一連の DTMF デジットにカンマ (,) を挿入して、残りの DTMF デジットのアウトパルスを続行する前に、1 秒間のポーズを挿入するようにスクリプトに指示できます。複数のカンマを使用して、ポーズ時間を長くすることができます。

図 153 に、Send Digit String のカスタマイザ ウィンドウを示します。

図 153 [Send Digit String] カスタマイザ ウィンドウ

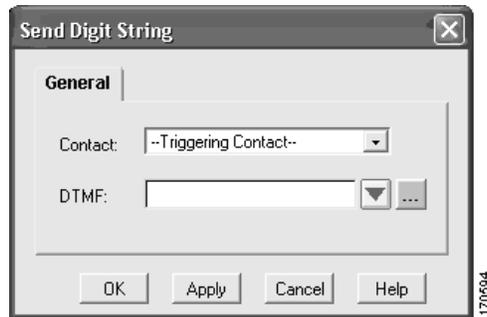


表 84 で、[Send Digit String] カスタマイザ ウィンドウのフィールドについて説明します。

表 84 [Send Digit String] カスタマイザ ウィンドウのフィールド

フィールド	説明
Contact	ステップの実行をトリガーするコンタクトを示す変数。他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。
DTMF	送信する DTMF デジットのシーケンスを示す変数または式。

## Dial-by-Extension Menu

**Dial-by-Extension Menu** ステップは、発信者が入力するディジットと出力ブランチ ラベルを関連付ける場合に使用します。

[Dial-by-Extension Menu] カスタマイザ ウィンドウの [General] タブ (図 154) は、ディジット (通常、発信者が電話のキーパッドから入力) を出力ブランチ ラベルに関連付ける場合に使用します。

複数の入力を 1 つの出力ブランチ ラベルに関連付けることができますが、1 つの入力に関連付けることができる出力ブランチ ラベルは 1 つだけです。

**Dial-by-Extension Menu** ステップは、発信者が入力した単一ディジットを受信し、この入力を一連のオプション出力ブランチにマッピングします。システムは、これらの各オプション出力ブランチの後に追加されたステップを実行します。

**Dial-by-Extension Menu** ステップのカスタマイザ ウィンドウには、4 つのタブがあります。

- 「[General] タブ」 (P.161)
- 「[Prompt] タブ」 (P.163)
- 「[Input] タブ」 (P.164)
- 「[Dial-by-Extension] タブ」 (P.165)

### [General] タブ

デフォルトでは、**Dial-by-Extension Menu** ステップには次の出力ブランチがあります。

- Output 1
- Output 2

- **Output 3**
- **Timeout**
- **Unsuccessful**
- **Digits Collected**

[Dial-by-Extension Menu] カスタマイザ ウィンドウの [General] タブで、さらに出カブランチを追加できます。

**Dial-by-Extension Menu** ステップは、タイムアウトまたは無効なディジット入力（どの接続にも関連付けられていないディジット）が発生した場合に、再試行します。最大再試行回数に達すると、最後の失敗の理由に応じて、**Dial-by-Extension Menu** ステップは **Timeout** 接続、**Unsuccessful** 接続、または **Digits Collected** 接続に進みます。

図 154 [Dial-by-Extension Menu] カスタマイザ ウィンドウ : [General] タブ

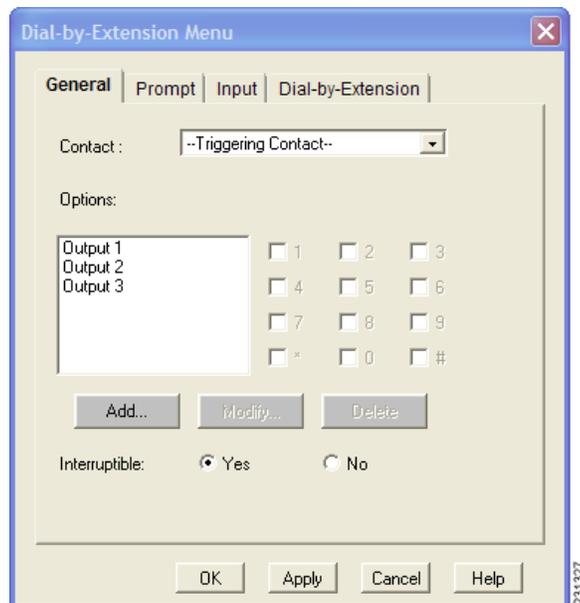


表 85 で、[General] タブのフィールドについて説明します。

表 85 [Dial-by-Extension Menu] カスタマイザ ウィンドウのフィールド : [General] タブ

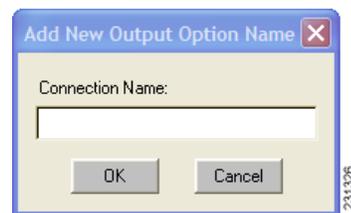
フィールド	説明
<b>Contact</b>	ステップの実行をトリガーするコンタクト。他のコンタクトが指定されていない場合、デフォルトは <b>Triggering Contact</b> です。
<b>Options</b>	出力値ごとのラベル。
<b>Add</b>	新しいオプションを追加します。新しいオプションが [Options] リストボックスに表示されます。

表 85 [Dial-by-Extension Menu] カスタマイザ ウィンドウのフィールド : [General] タブ (続き)

フィールド	説明
Modify	[Rename Output Option] ダイアログボックスを使用して、選択されたオプションを変更します。このダイアログボックスには、[Add New Output Option Name] ダイアログボックスと同じフィールドがあり、同じ方法で設定します。
Interruptible	[Yes] の場合は、外部のイベント（発信者が電話を切るなど）がステップに割り込むことができます。  [No] の場合は、このステップが完了するまで、他のプロセスを実行できません。

[Add New Output Option Name] ウィンドウを図 155 に示します。

図 155 [Add New Output Option Name] ウィンドウ



## [Prompt] タブ

[Dial-by-Extension Menu] カスタマイザ ウィンドウの [Prompt] タブ (図 156) は、Prompt を設定し、Barge in 条件および Continue On Prompt Errors 条件を設定する場合に使用します。

図 156 [Dial-by-Extension Menu] カスタマイザ ウィンドウ : [Prompt] タブ

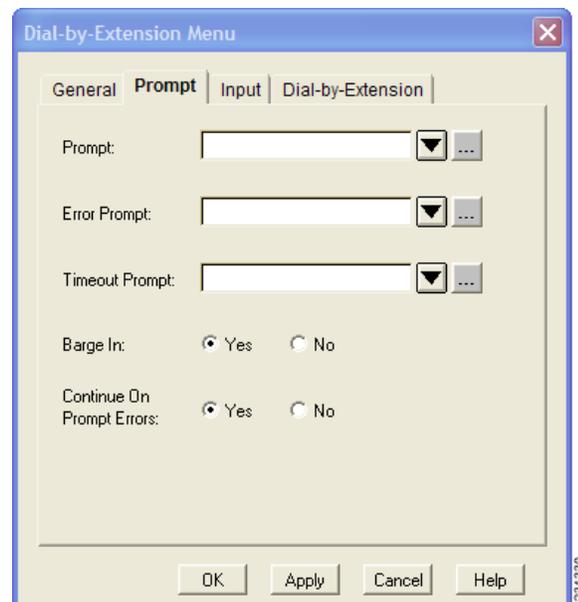


表 86 で、[Prompt] タブのフィールドについて説明します。

表 86 [Dial-by-Extension Menu] カスタマイザ ウィンドウ : [Prompt] タブ

プロパティ	説明
Prompt	再生するプロンプトを指定します。
Error Prompt	入力エラーが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
Timeout Prompt	タイムアウトが発生した場合に再生されるプロンプト。値を入力するか、ドロップダウンメニューから変数を選択するか、または Expression Editor を使用してプロンプト式を入力します。
Barge In	true の場合は、発信者がプロンプトに割り込むことができます。false の場合は、発信者の入力を受け入れる前にプロンプトの再生を完了する必要があります。
Continue On Prompt Errors	[Yes] の場合は、リストの次のプロンプトに進みます。[No] の場合は、例外が発生します。例外はスクリプトで処理できます。

## [Input] タブ

[Dial-by-Extension Menu] カスタマイザ ウィンドウの [Input] タブ (図 157) は、タイムアウトの設定、最大再試行回数、および [Flush Input Buffer] オプションを設定する場合に使用します。

図 157 [Dial-by-Extension Menu] カスタマイザ ウィンドウ : [Input] タブ

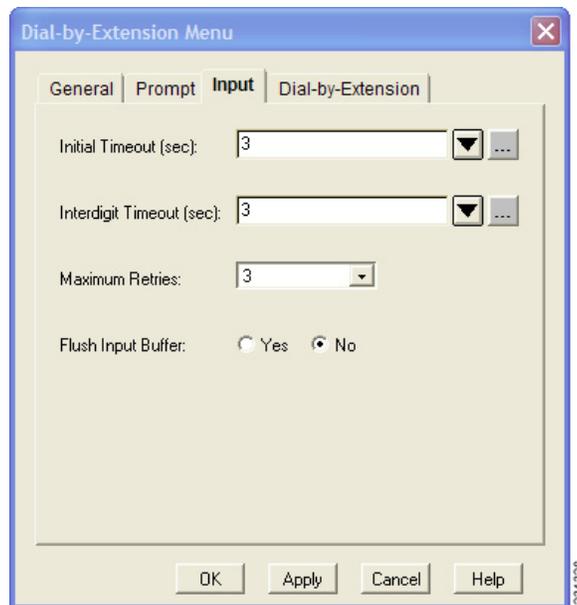


表 87 で、[Prompt] タブのフィールドについて説明します。

表 87 [Dial-by-Extension Menu] カスタマイザ ウィンドウのフィールド : [Input] タブ

プロパティ	説明
Initial Timeout (sec)	発信者からの入力を待つ時間。このタイマーが経過すると、システムはプロンプトをもう一度再生するか、発信者がいるかどうかを確認するシステム プロンプトを再生します。値を入力するか、タイムアウト値を格納する変数を [Timeout] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして数字を入力します。
Interdigit Timeout (sec)	発信者が押すディジットとディジットの間で入力を待つ時間。このタイマーが経過すると、システムはプロンプトをもう一度再生するか、発信者がいるかどうかを確認するシステム プロンプトを再生します。値を入力するか、タイムアウト値を格納する変数を [Timeout] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして数字を入力します。
Maximum Retries	タイムアウトまたは無効な入力応答の後で再開できるエントリの回数。最大再試行回数に達した後、最後の試行がタイムアウトだったか、無効な入力応答が入力されたかによって、 <b>Menu</b> ステップは <b>Timeout</b> 出力ブランチまたは <b>Unsuccessful</b> 出力ブランチに進みます。  値「0」は、再試行を許可しないことを意味します。この場合、スクリプトが再試行シナリオを処理する必要があります。
Flush Input Buffer	[Yes] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去します。  [No] の場合は、発信者の入力をキャプチャする前に、前に入力された入力を消去しません。

## [Dial-by-Extension] タブ

[Dial-by-Extension Menu] カスタマイザ ウィンドウの [Dial-by-Extension] タブ (図 158) は、内線番号のダイヤル オプションを設定する場合に使用します。

図 158 [Dial-by-Extension Menu] カスタマイザ ウィンドウ : [Dial-by-Extension] タブ

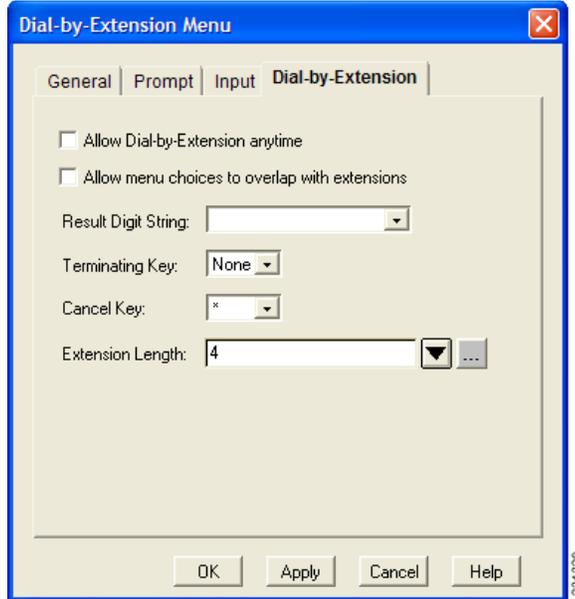


表 88 で、[Dial-by-Extension] タブのフィールドを一覧し、説明します。

表 88 [Dial-by-Extension Menu] カスタマイザ ウィンドウのフィールド : [Dial-by-Extension] タブ

プロパティ	説明
<b>Allow Dial-by-Extension anytime</b>	<ul style="list-style-type: none"> <li>オンにした場合 : Cisco Unity Express は、内線番号のダイヤル機能を常に使用可能にします。</li> <li>オフにした場合 : Cisco Unity Express は、内線番号のダイヤル機能を常に使用不可にします。</li> </ul>
<b>Allow menu choices to overlap with extensions</b>	<ul style="list-style-type: none"> <li>オンにした場合 : Cisco Unity Express は、内線番号のダイヤル機能における内線番号の重複を許可します。</li> <li>オフにした場合 : Cisco Unity Express は、内線番号のダイヤル機能における内線番号の重複を許可しません。</li> </ul>
<b>Result Digit String</b>	結果として得られるディジット文字列を保持する変数。
<b>Terminating Key</b>	内線番号の入力を終了するために発信者が押すキー。
<b>Cancel Key</b>	内線番号の入力をキャンセルするために発信者が押すキー。
<b>Extension Length</b>	内線に含まれるディジット数。

## Voice Browser (IVR のみ)

**Voice Browser** ステップは、VoiceXML アプリケーションを起動する場合に使用します。

**Voice Browser** ステップは、DTMF シグナリングのみをサポートしています。Cisco.com の Cisco Unity Express の「Download Software」サイトから、サンプル スクリプト (webapp.aef) を入手できます。

**Voice Browser** ステップのカスタマイザ ウィンドウには、3 つのタブがあります。

- 「[General] タブ」 (P.167)
- 「[Request Parameters] タブ」 (P.167)
- 「[Return Parameters] タブ」 (P.168)

## [General] タブ

[General] タブ (図 159) は、コンタクトを識別し、ブラウザに特定の URL をポイントさせる変数を選択する場合に使用します。

図 159 [Voice Browser] カスタマイザ ウィンドウ : [General] タブ



表 89 で、[General] タブのフィールドについて説明します。

表 89 [Voice Browser] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Contact	他のコンタクトが指定されていない場合、デフォルトは Triggering Contact です。
URL	URL を手動で入力するか、URL を保持している変数をドロップダウン リストから選択できます。
Interruptible	外部イベントによる割り込みを有効または無効にします。 <ul style="list-style-type: none"> <li>• Yes : 外部イベントによってステップに割り込むことができます。</li> <li>• No : ステップが終了してから、外部イベントを考慮します。</li> </ul>
Continue on Prompt Errors	<b>Voice Browser</b> ステップの場合は無効です。

## [Request Parameters] タブ

[Request Parameters] タブ (図 160) は、目的の VoiceXML コンテンツを取得するために、HTTP 要求と一緒に渡されるパラメータを定義する場合に使用します。

図 160 [Voice Browser] カスタマイザ ウィンドウ : [Request Parameters] タブ

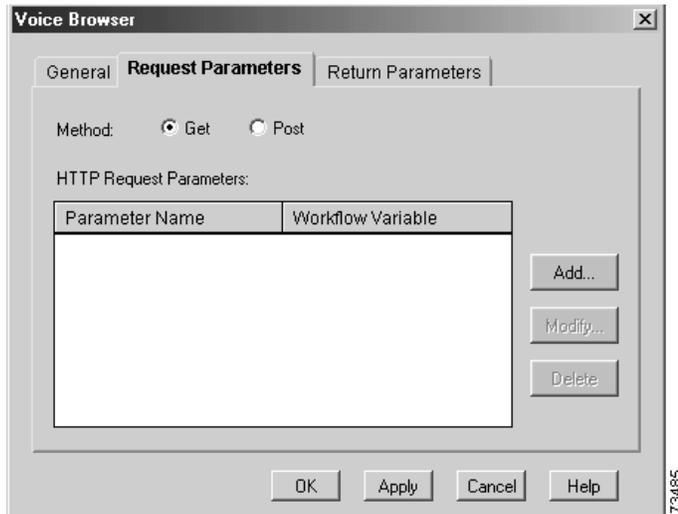


表 90 で、[Request Parameters] タブのフィールドについて説明します。

表 90 [Voice Browser] カスタマイザ ウィンドウのフィールド : [Request Parameters] タブ

フィールド	説明
<b>Method</b>	URL が HTTP 要求を表す場合に使用するメソッド。 <ul style="list-style-type: none"> <li>• <b>GET</b> : URL にパラメータを付加します。このステップは、URL [url?name=value,name=value] 形式のドキュメント式を使用するのと同じです。</li> <li>• <b>POST</b> : HTML 形式で入力された場合と同様にパラメータを含めます。</li> </ul>
<b>[HTTP Request Parameters] リストボックス</b>	HTTP 要求パラメータの名前とそれに対応するワークフロー変数名。

## [Return Parameters] タブ

[Return Parameters] タブ (図 161) は、パラメータ情報を VoiceXML アプリケーションの .aef スクリプト ファイルに戻す場合に使用します。

図 161 [Voice Browser] カスタマイザ ウィンドウのフィールド : [Return Parameters] タブ



表 91 で、[Return Parameters] タブのフィールドについて説明します。

表 91 [Voice Browser] カスタマイザ ウィンドウのフィールド : [Return Parameters] タブ

フィールド	説明
Expr Parameter	VoiceXML アプリケーションが次の形式の exit 文を使用している場合に、戻り値を格納する変数。 <pre>&lt;exit expr = "ECMAScript_Expression"/&gt;</pre>
Namelist Parameters	VoiceXML アプリケーションが次の形式の exit 文を使用している場合に、VoiceXML アプリケーションによって返される名前と、それに対応し、戻り値が格納されるスクリプト変数。 <pre>&lt;exit namelist = "string"/&gt;</pre>

## Prompt ステップ

Cisco Unity Express Script Editor の **Prompt** パレットのステップは、インテリジェントプロンプトを作成する手段を提供します。

**Prompt** パレットには、次のステップがあります。

- 「[Create Conditional Prompt](#)」 (P.170)
- 「[Create Container Prompt](#)」 (P.171)
- 「[Create Generated Prompt](#)」 (P.174)

図 162 に、**Prompt** パレットのステップを Cisco Unity Express Script Editor の [Palette] ペインに表示されるとおりに示します。

図 162 Prompt パレットのステップ



## Create Conditional Prompt

**Create Conditional Prompt** ステップは、指定された Boolean 式の評価結果に基づいてプロンプトを作成する場合に使用します。

渡されるプロンプトは、プロンプト オブジェクトとしてすぐに評価され、再生時まで解決されません。つまり、式の一部として入力された変数の値が、このプロンプトが作成されたときから再生されるときまでの間に変更された場合、条件式の評価には新しい変数の値が使用されます。

図 163 に、**Create Conditional Prompt** ステップのカスタマイザ ウィンドウを示します。

図 163 [Create Conditional Prompt] カスタマイザ ウィンドウ



表 92 で、[Create Conditional Prompt] カスタマイザ ウィンドウのフィールドについて説明します。

表 92 [Create Conditional Prompt] カスタマイザ ウィンドウのフィールド

フィールド	説明
Output Prompt	<b>Create Conditional Prompt</b> ステップの結果であるプロンプトを格納する変数。条件付きプロンプトの名前が、[Design] ペインの <b>Create Conditional Prompt</b> ステップ アイコンの横に表示されます。
Condition Expression	2 つのプロンプトのどちらを再生するかをスクリプトが判定するために使用する Boolean 式。値を入力するか、条件の評価に使用する式を格納する変数を [Condition Expression] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして条件を評価するための式を入力します。

表 92 [Create Conditional Prompt] カスタマイザ ウィンドウのフィールド (続き)

フィールド	説明
Prompt If True	式が True の場合に使用されるプロンプト。値を入力するか、式の評価が True の場合に使用するプロンプトを格納する変数を [Prompt If True] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして、式の評価が True の場合に使用するプロンプトを指定する式を入力します。
Prompt If False	式が False の場合に使用されるプロンプト。値を入力するか、式の評価が False の場合に使用するプロンプトを格納する変数を [Prompt If False] ドロップダウン メニューから選択するか、または [Expression Editor (...)] ボタンをクリックして、式の評価が False の場合に使用するプロンプトを指定する式を入力します。

## Create Container Prompt

**Create Container Prompt** ステップは、複数のプロンプトを 1 つの大きなプロンプトに組み合わせる場合に使用します。3 種類のコンテナ プロンプトを作成できます。

- **Concatenated Prompt** (連結プロンプト) : 指定された順序で再生することによって単一のプロンプトになるプロンプト フレーズのリストが含まれています。

たとえば、「Your checking account balance is one hundred and sixty-eight dollars」というプロンプトの場合、(1) ユーザ プロンプト「Your」から開始します。(2) 次に <accountType == "check"> のような条件を指定した条件付きプロンプトを続け、条件が True の場合は「checking account」を再生し、False の場合は「savings account」を再生します。(3) 最後に口座残高を再生します。

- **Escalating Prompt** (エスカレーティング プロンプト) : 最小限の情報を提供する質問プロンプトを再生し、応答がないたびに追加のプロンプト フレーズを追加します。

たとえば、必要に応じて発信者に情報を提供するプロンプトの場合、**Get Digit String** ステップなどのメディア ステップに渡されたときに、「What is your account number?」など、エスカレーティング プロンプト内の最初の短いプロンプトを再生します。

発信者が口座番号を入力しなかったためにステップが失敗した場合は、「Please provide your account number by entering the account number using your touch tone phone followed by the pound key」のような 2 番目のプロンプトを再生します。

- **Random Prompt** (ランダム プロンプト) : 指定されたリスト中の 1 つのフレーズを、ランダムな順序で再生するプロンプトを作成します。たとえば、発信者が利用可能なエージェントを待つ間、一連の宣伝または情報メッセージをランダムな順序で再生できます。

図 164 に、**Create Container Prompt** ステップのカスタマイザ ウィンドウを示します。

図 164 [Create Container Prompt] カスタマイザ ウィンドウ



表 93 で、[Create Container Prompt] カスタマイザ ウィンドウのプロパティについて説明します。

表 93 [Create Container Prompt] カスタマイザ ウィンドウのプロパティ

プロパティ	説明
<b>Output Prompt</b>	Create Container Prompt ステップで生成された、組み合わせたプロンプトを保持するスクリプト変数。連結プロンプトの名前が、[Design] ペインの <b>Create Container Prompt</b> ステップ アイコンの横に表示されます。
<b>Prompt Container Type</b>	Concatenated、Escalating、または Random。
<b>Prompts</b>	コンテナ プロンプトに組み合わせるプロンプトのリスト。
<b>Add</b>	プロンプトを追加します。値を入力するか、追加するプロンプトを格納する変数を [Prompt] ドロップダウン リストから選択するか、または [Expression Editor (...)] ボタンをクリックして追加するプロンプトを指定する式を入力します。ランダム プロンプトの場合、シーケンス内でプロンプトの優先順位を表す数字を [Weight] ドロップダウン メニューから選択します。Prompt 変数の名前が、[Create Container Prompt] カスタマイザ ウィンドウの [Prompts] リスト ボックスに表示されます。
<b>Modify</b>	[Modify Prompt] 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には、[Add Prompt] ダイアログボックスと同じフィールドがあり、同じ方法で設定します。

表 93 [Create Container Prompt] カスタマイザ ウィンドウのプロパティ (続き)

プロパティ	説明
UP、Down	プロンプトの再生順序を決定します。個々のプロンプトを選択し、そのプロンプトを 1 レベル上げるには <b>Up</b> を、1 レベル下げるには <b>[Down]</b> をクリックします。
Override Language (optional)	このオプションは、将来使用します。

[Add Prompt] ダイアログボックスを [図 165](#) に、ランダム プロンプトの場合のウィンドウを [図 166](#) に示します。

図 165 [Add Prompt] ダイアログボックス

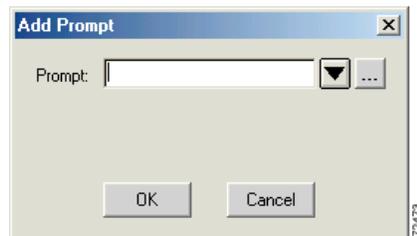
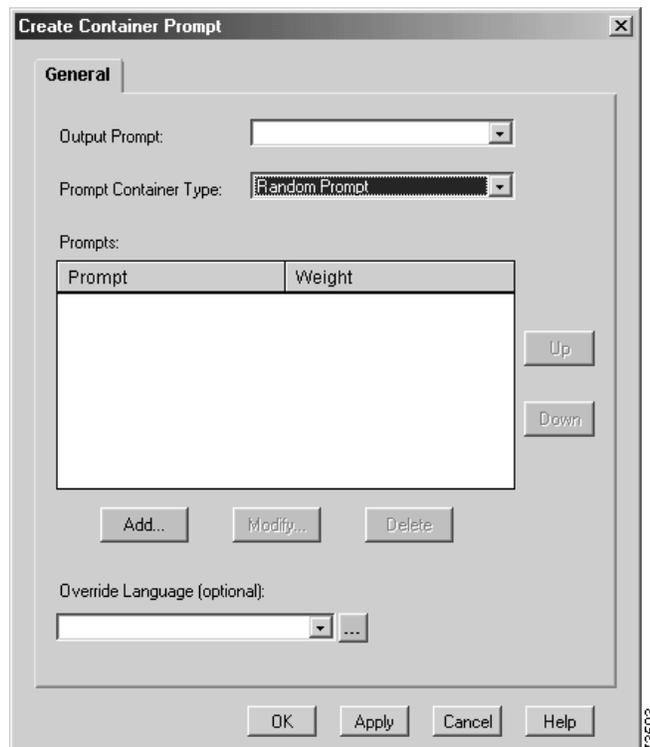


図 166 [Create Container Prompt] カスタマイザ ウィンドウ : Random Prompt



[Add Prompt] ダイアログボックスを [図 167](#) に示します。

図 167 [Add Prompt] ダイアログボックス



## Create Generated Prompt

**Create Generated Prompt** ステップは、実行時のスクリプト情報に基づいて動的に値が決定される中間変数から、プロンプトフレーズを作成する場合に使用します。

たとえば、スクリプトのある時点でデータベースに口座残高を問い合わせ、Currency ジェネレータを使用して数値を生成することによって、プロンプトフレーズ「account balance is one hundred and sixty-eight dollars」を作成できます。



(注) Script Editor を使用してデータベースに対してクエリを実行することはできません。



(注) **Create Generated Prompt** ステップは、4 桁の形式の年だけを受け付けます。3 桁の日付形式は受け付けません。



(注) **Create Generated Prompt** ステップは、無効な時刻が指定されると、4:00 P.M を出力します。0000 から 2400 までの有効な時刻を指定してください。

図 168 に、**Create Generated Prompt** ステップのカスタマイザ ウィンドウを示します。

図 168 [Create Generated Prompt] カスタマイザ ウィンドウ

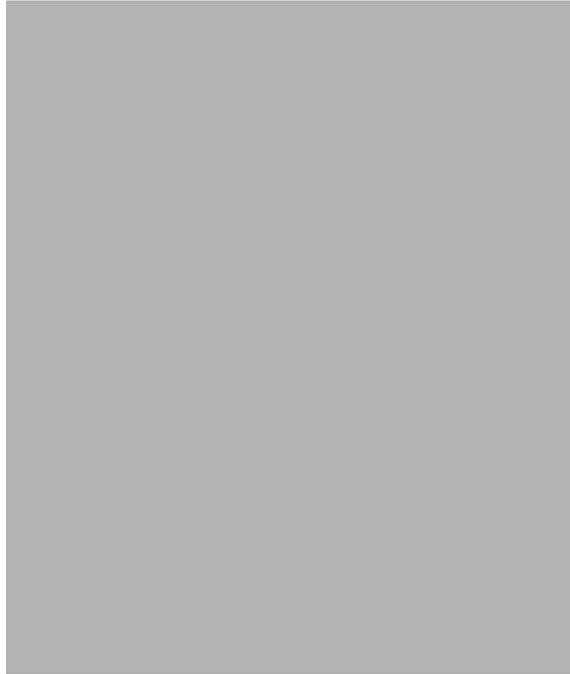


表 94 で、[Create Generated Prompt] カスタマイザ ウィンドウのフィールドについて説明します。

表 94 [Create Generated Prompt] カスタマイザ ウィンドウのフィールド

フィールド	説明
<b>Output Prompt</b>	このステップの結果として生成されるプロンプト オブジェクトが格納されるプロンプト スクリプト変数。生成されるプロンプトの名前が、[Design] ペインの <b>Create Generated Prompt</b> ステップ アイコンの横に表示されます。
<b>Generator Type</b>	生成される情報の型。 サポートされる 12 のジェネレータ型の説明については、次の項を参照してください。
<b>Constructor Type</b>	ジェネレータ型に対応するコンストラクタ型。
<b>Argument Information</b>	引数と、その値。ダブルクリックして、項目の値を定義します。
<b>Enter Variable or Expression</b>	引数の値を定義します。値を入力するか、引数の値を保持する変数を [Enter Variable Expression] ドロップダウンメニューから選択するか、または [Expression Editor (...)] ボタンをクリックして有効な式を入力します。引数の名前と値が、[Create Generated Prompt] カスタマイザ ウィンドウの [Argument Information] リスト ボックスに表示されます。
<b>Override Language (optional)</b>	このオプションは、将来使用します。

[Define Character] ダイアログボックスを図 169 に示します。

図 169 [Define Character] ダイアログボックス



## ジェネレータ型

**Create Generated Prompt** ステップは、次のジェネレータ型をサポートします。

- 「[Number ジェネレータ](#)」 (P.176)
- 「[Character ジェネレータ](#)」 (P.177)
- 「[Spelling ジェネレータ](#)」 (P.177)
- 「[Date ジェネレータ](#)」 (P.177)
- 「[Time ジェネレータ](#)」 (P.178)
- 「[Ordinal ジェネレータ](#)」 (P.178)
- 「[Currency ジェネレータ](#)」 (P.179)
- 「[Country ジェネレータ](#)」 (P.179)
- 「[Language ジェネレータ](#)」 (P.179)
- 「[Telephone Number ジェネレータ](#)」 (P.179)
- 「[Credit Card Number ジェネレータ](#)」 (P.180)
- 「[Credit Card Expiration Date ジェネレータ](#)」 (P.180)
- 「[Fullname ジェネレータ](#)」 (P.180)

## Number ジェネレータ

**Number** ジェネレータ型は、次のコンストラクタをサポートします。

- (Number number)
- (String number)
- (Number number, Number gender)
- (String number, Number gender)
- (Number number, Boolean play.full)
- (String number, Boolean play.full)
- (Number number, Boolean play.full, Number gender)
- (Number number, Boolean play.full, Number gender)

3つのパラメータの意味は、次のとおりです。

- **number** : 再生する番号を定義する任意の Number オブジェクト (Integer、Long、Float、Double、BigInteger、BigDecimal) または String オブジェクト。
- **gender** : 特定の性別コンテキストで番号を再生する必要がある場合に、このパラメータでコンテキストを指定します。有効な値は、0 (性別の区別なし)、1 (男性)、2 (女性) です。
- **play.full** : このオプションの Boolean 引数が True または省略されている場合は、番号を完全な形式で再生します。たとえば「709」の場合、「Seven Hundred and Nine」。それ以外の場合は、番号を短い形式で再生します。たとえば「709」の場合、「Seven Oh Nine」。



(注) 番号を完全な形式で再生する場合にサポートされる最大の番号は ±999,999,999,999 です。

## Character ジェネレータ

**Character** ジェネレータ型は、次のコンストラクタをサポートします。

- (Character character)
- (Character character, Boolean play\_all)

2つのパラメータの意味は、次のとおりです。

- **character** : 再生される Character オブジェクト。
- **play\_all** : スペース、句読点などの特殊文字を、無音 (250 ~ 500 ms) で表すのではなく、通常文字と同様に再生するかどうかを示すオプションの Boolean 型フラグ。

## Spelling ジェネレータ

**Spelling** ジェネレータ型は、次のコンストラクタをサポートします。

- (String string)
- (String string, Boolean punctuation)
- (Object object)
- (Object object, Boolean punctuation)

3つのパラメータの意味は、次のとおりです。

- **string** : 再生される String オブジェクト。
- **object** : `String.valueOf()` メソッドが返す、読み上げる対象の文字列表現のオブジェクト。
- **punctuation** : スペース、句読点などの特殊文字を通常文字と同様に再生するか、無音で表すかを示すオプションの Boolean 型フラグ。



(注) **Spelling** ジェネレータの **punctuation** のデフォルトの動作は、**Character** ジェネレータの **play\_all** のデフォルトの動作とは異なります。

## Date ジェネレータ

**Date** ジェネレータ型は、次のコンストラクタをサポートします。

- (Date date)
- (Date date, Boolean skip.current.year)
- (Number year)
- (Number year, Number month)

- (Number year, Number month, Boolean skip.current.year)
- (Number year, Number month, Number day)
- (Number year, Number month, Number day, Boolean skip.current.year)

5 つのパラメータの意味は、次のとおりです。

- **date** : 再生する日付を抽出する **Date** オブジェクト。
- **skip.current.year** : **True** に設定されている場合、現在の年と同じ年は再生されません。
- **year** : 再生する日付の年。この年は、完全な形式で指定する必要があります (2007 など)。



(注) システムは、指定された任意の数値を再生します。そのため、発信者は有効な年を正確に指定する必要があります。

- **month** : 再生する日付の月。有効な範囲は 1 ~ 12 で、1 が 1 月、12 が 12 月を表します。
- **day** : 再生する日付の日。有効な範囲は 1 ~ 31 で、指定された月と年に基づいて実行時に検証されます。

## Time ジェネレータ

**Time** ジェネレータ型は、次のコンストラクタをサポートします。

- (Time)
- (Hours, Minutes)

3 つのパラメータの意味は、次のとおりです。

- **time** : 再生する時刻を表す **Date** または **Time** オブジェクト。**time** は、再生する時刻を指定する 0 ~ 2359 の **Number** オブジェクト (**Integer**、**Float**、**Long** など) として定義することもできます (たとえば、12:34 は「12 34 PM」として再生されます)。指定された値が 2359 より大きい場合、**time** は、「the epoch」と呼ばれる標準基準時刻である January 1, 1970, 00:00:00 GMT からのミリ秒数と見なされます。
- **hours** : 再生する時間を指定する **Number** オブジェクト。
- **minutes** : 再生する分を指定する **Number** オブジェクト。

## Ordinal ジェネレータ

**Ordinal** ジェネレータ型は、次のコンストラクタをサポートします。

- (Number number)
- (String number)
- (Number number, Number gender)
- (String number, Number gender)

2 つのパラメータの意味は、次のとおりです。

- **number** : 再生する序数を定義する任意の **Number** または **String** オブジェクト。サポートされる範囲は、1 ~ 999999 です。
- **gender** : 特定の性別コンテキストで序数を再生する必要がある場合に、このパラメータでコンテキストを指定します。有効な値は、0 (性別の区別なし)、1 (男性)、2 (女性) です。



(注) 通話に関連付けられた言語が、性別によって異なる動作をしない場合、このパラメータは無視されます。

## Currency ジェネレータ

**Currency** ジェネレータ型は、次のコンストラクタをサポートします。

- (Currency designator)
- (Number amount)
- (Number amount, Currency currency)
- (Number dollar, Number cent)
- (Number dollar, Number cent, Currency currency)
- (Number amount, Boolean colloquial)
- (Number amount, Boolean colloquial, Currency currency)
- (Number dollar, Number cent, Boolean colloquial, Currency currency)

6つのパラメータの意味は、次のとおりです。

- **designator** : 再生する通貨の呼称。たとえば、「USD」は「U.S.Dollar」と再生されます。
- **amount** : システムで設定されたデフォルトの通貨、または指定された通貨で再生する通貨の額。
- **dollar** : 再生する通貨単位での金額を表す **Number** オブジェクト。数値の整数部分だけが再生されます。小数部分があった場合は、無視されます。
- **cent** : 再生する通貨の下位単位を表す **Number** オブジェクト。数値の整数部分だけが再生されます。小数部分があった場合は、無視されます。



(注) 下位単位で許可される最大値を超える数値が指定された場合、超過分が適切に通貨単位の数値に加算されます。たとえば、「5 dollars and 233」と指定すると、「7 dollars and 33 cents」になります。

- **colloquial** : 口語的な通貨表現を使用するかどうかを指定するオプションの **Boolean** 型フラグ（「US Dollars」の代わりに「Dollars」を使用するなど）。省略した場合、通貨金額は口語的な形式で再生されます。
- **currency** : 金額を再生する通貨。指定しなかった場合、システムのデフォルト設定通貨が再生されます。

## Country ジェネレータ

**Country** ジェネレータ型は、1つのコンストラクタ (**Language language**) だけをサポートします。

## Language ジェネレータ

**Language** ジェネレータ型は、次のコンストラクタをサポートします。パラメータ「**language**」は、再生する言語を取得する **Language** オブジェクトです。たとえば、**en\_US** の場合、米国英語として再生されます。

## Telephone Number ジェネレータ

**Telephone Number** ジェネレータ型は、1つのコンストラクタ (**String number**) だけをサポートします。パラメータ「**number**」は、ディジットのシーケンスとして再生される電話番号を指定する **String** オブジェクトです。

文字列に、文字 "-" ( ) が含まれている場合、文字は 250 ms の無音に置き換えられます。それ以外の場合、文字列は自動的にフォーマットされます。

文字列の自動フォーマットでは、ディジットのセクション間に 250 ms の無音が挿入されます。このセクションは、文字列が正確に 5 桁のディジットである場合を除いて、「XXX-XXX-XXX-XXXX」という規則に従います。5 桁のディジットの場合は、5 桁のディジットで構成される単一セクションであると見なされます。

DMTF (Dual-Tone MultiFrequency) ディジットには、標準タッチトーン電話キーボード上のディジット ABCD0123456789#\* が含まれます。タッチトーン キーボード上のボタンを押すと、DMTF が 2 つのトーンの組み合わせ (1 つは高周波、もう 1 つは低周波) を作成します。

文字「x」は、「Extension」と再生されます。DTMF ディジット (ABCD0123456789#\*) は、通常の文字として再生されます。

x が DTMF ディジット (0123456789) のとき、「\*xx」という形式の文字列は、「star xx」と再生されます (たとえば、「\*53」は「star fifty-three」と再生されます)。

### Credit Card Number ジェネレータ

**Credit Card Number** ジェネレータ型は、1 つのコンストラクタ (String number) だけをサポートします。パラメータ「number」は、ディジットのシーケンスとして再生されるクレジット カード番号を指定する String オブジェクトです。

指定されたクレジット カード番号に「-」が含まれている場合は、「-」文字が 250 ms の無音に置き換えられて再生されます。それ以外の場合、数字は自動的に 4 桁のディジットで構成されるセクションに分割され、セクション間に 250 ms の無音が挿入されて再生されます。

### Credit Card Expiration Date ジェネレータ

**Credit Card Expiration Date** ジェネレータ型は、次のコンストラクタをサポートします。

- (Number year, Number month, Number day)
- (Number year, Number month)

パラメータは、次の **Generated Date** コンストラクタと同じです。

- day が 0 または省略 : GeneratedDate (year, month, true)
- それ以外のすべての場合 : GeneratedDate (year, month, day, true)

### Fullname ジェネレータ

**Fullname** ジェネレータ型は、次のコンストラクタをサポートします。

- firstname : 再生する名を指定する String オブジェクト。
- lastname : 再生する姓を指定する String オブジェクト。

## Session ステップ

**Session** パレットのステップは、カスタム変数をワークフロー スクリプトに格納する手段を提供します。

カスタム変数は、スクリプトの実行時に割り当てることができ、通話終了時に履歴レポート データベースに保存されます。

Session パレットには 1 つのステップがあります。

- 「[Set Session Info](#)」 (P.181)

図 170 に、**Session** パレットのステップを示します。

図 170 Session パレットのステップ



## Set Session Info

**Set Session Info** ステップは、Call Contact Detail Record (CCDR) レポートに使用するカスタム データの保存をサポートします。

**Set Session Info** ステップには、次の 2 つのタブがあります。

- 「[General] タブ」 (P.181)
- 「[Context] タブ」 (P.181)

### [General] タブ

**Set Session Info** ステップの [General] タブ (図 171) は、コンテキスト情報を追加または変更するセッション変数を指定する場合に使用します。

セッション情報を設定するには、[General] タブを使用して、[Session] ドロップダウン メニューから変数を選択し、[Apply] をクリックして、[Context] タブ (図 172) を完了します。

図 171 [Set Session Info] カスタマイザ ウィンドウ : [General] タブ



表 95 で、[General] タブのフィールドについて説明します。

表 95 [Set Session Info] カスタマイザ ウィンドウのフィールド : [General] タブ

フィールド	説明
Session	コンテキスト情報を追加または変更するセッション。

### [Context] タブ

**Set Session Info** ステップの [Context] タブは、属性変数の値を追加または変更する場合に使用します。

一部のセッション コンテキスト プロパティは、このセッションと関連付けられている通話の履歴データベースに保存されている Call Contact Detail Record の一部として含まれています。これらのプロパティの属性名は、\_ccdrVar1 ~ \_ccdrVar10 です。

たとえば、**Set Session Info** ステップを使用して、\_ccdrVar1、\_ccdrVar2、\_ccdrVar3、\_ccdrVar4、\_ccdrVar5、\_ccdrVar6、\_ccdrVar7、\_ccdrVar8、\_ccdrVar9、\_ccdrVar10 という名前のコンテキスト変数の 1 つまたは複数に希望する値を割り当てることができます。

[\_ccdrVar10] フィールドを除き、各 [\_ccdrVar] フィールドは最大 40 文字までの英数字の値を保持できます。[\_ccdrVar10] フィールドは、最大 256 文字の英数字の値を保持できます。



(注) ccdr データベースに正しくデータを書き込むには、\_ccdrVar 名を 1 つの下線 ( \_ ) で始める必要があります。

図 172 に、[Context] タブのカスタマイザ ウィンドウを示します。

図 172 [Set Session Info] カスタマイザ ウィンドウ : [Context] タブ

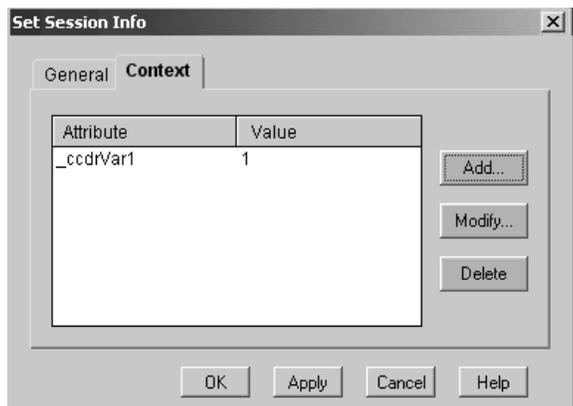


表 96 で、[Context] タブのフィールドについて説明します。

表 96 [Set Session Info] カスタマイザ ウィンドウのフィールド : [Context] タブ

フィールド	説明
Attribute/Value	[General] タブで識別されたセッションの属性と、関連付けられている値。値を 10 個の Call Contact Detail Record (CCDR) 属性 _ccdrVar1 ~ _ccdrVar10 に割り当て、履歴データベースに保存できます。属性 _ccdrVar1 ~ _ccdrVar9 の最大文字数は 40 文字です。属性 _ccdrVar10 の最大文字数は 255 文字です。

## User ステップ

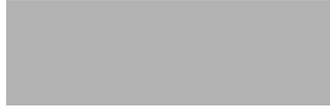
**User** パレットのステップは、ユーザ属性を取得する手段を提供します。

User パレットには、次のステップがあります。

- 「Get User Info」 (P.183)
- 「Extension To User」 (P.184)

図 173 に、**User** パレットのステップを示します。

図 173 User パレットのステップ



## Get User Info

**Get User Info** ステップ (図 174) は、スクリプトでユーザ属性を利用できるようにする場合に使用します。

図 174 [Get User Info] カスタマイザ ウィンドウ



表 97 で、[Get User Info] カスタマイザ ウィンドウのフィールドについて説明します。

表 97 [Get User Info] カスタマイザ ウィンドウのフィールド

フィールド	説明
User	スクリプト変数に格納されている、以前に取得したユーザ オブジェクト (たとえば、 <b>Name To User</b> ステップなどを使用)。User 変数の名前が、[Design] ペインの <b>Get User Info</b> ステップ アイコンの横に表示されます。
Attribute/Variable	ユーザの属性、および関連付けられた変数。
Set...	[Get User Full Name] ダイアログボックスが表示されます。ユーザのフルネーム用の変数を選択します。
Clear	[Get User Full Name] ダイアログボックスで設定された変数をクリアします。

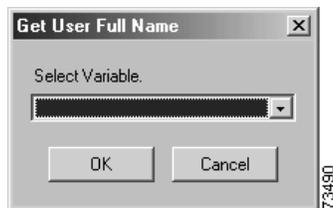
表 98 で、**Get User Info** ステップを使用して取得できる属性について説明します。

表 98 Get User Info の属性

属性	説明
Full Name	Cisco Unity Express GUI 管理インターフェイスで設定したユーザの氏名を表す文字列。
Last Name	ユーザの姓を表す文字列。
First Name	ユーザの名を表す文字列。
Extension	Cisco Unity Express GUI 管理 Web インターフェイスで選択した、プライマリ内線番号を表す文字列。
E-mail	このユーザの電子メール ID を表す文字列。 現在は、ユーザ ID フィールドが返されます。
User ID	このユーザに対して設定されたユーザ ID を表す文字列。
Spoken Name	ユーザの名前の録音を表すドキュメント オブジェクト。

[Get User] ダイアログボックスを図 175 に示します。

図 175 [Get User] ダイアログボックス



## Extension To User

**Extension To User** ステップ (図 176) は、発信者が入力した内線に基づいてユーザを検索する場合に使用します。**Extension To User** ステップは、発信者が入力した内線と、ディレクトリに格納されている内線を比較します。一致を検出すると、システムは、一致した内線のユーザを返します。この情報を **Get User Info** ステップで使用して、ユーザに関するより詳しい情報を取得できます。**Extension To User** ステップを使用することによって、スクリプトが外部の番号に通話を転送することを防止できます。

**Extension To User** ステップは、2 つの出力ブランチを自動的に追加します。

- **Successful** : 指定された内線と一致する内線のユーザが見つかった場合は、このブランチの後のステップが実行されます。
- **Unsuccessful** : 指定された内線と一致する内線のユーザが見つからなかった場合は、このブランチの後のステップが実行されます。

図 176 [Extension To User] カスタマイザ ウィンドウ



表 99 で、[Extension To User] カスタマイザ ウィンドウのフィールドについて説明します。

表 99 [Extension To User] カスタマイザ ウィンドウのフィールド

フィールド	説明
Extension	Cisco Unity Express Administration ページの User ページで選択した、プライマリ内線番号を表す文字列。 <b>Extension</b> 変数の名前が、[Design] ペインの <b>Extension To User</b> ステップ アイコンの横に表示されます。
Result User	指定された内線を持つユーザを表すユーザ オブジェクトが格納される変数。

## VoiceMail ステップ

VoiceMail パレットのステップは、ボイスメール属性を設定する手段を提供します。

VoiceMail パレットには、Send Voice Message ステップがあります。

図 177 に、VoiceMail パレットのステップを示します。

図 177 VoiceMail パレットのステップ



### [General] タブ

VoiceMail ステップの [General] タブ (図 178) は、コンテキスト情報を追加または変更する Sender ID 変数、Recipient ID 変数、および Prompt 変数を指定する場合に使用します。

Send Voice Message 情報を設定するには、[General] タブを使用して、ドロップダウン メニューから変数を選択し、[Apply] をクリックします。

図 178 [Send Voice Message] カスタマイザ ウィンドウ

