

Windows® Phone 7 向け

# UI デザイン/操作ガイド



このドキュメントは、最終的な製品版の発売時に実質的に変更される可能性があるソフトウェア製品のプレリリース版に適用されます。このドキュメントに記載された内容は情報提供のみを目的としており、明示または黙示に関わらず、これらの情報についてマイクロソフトはいかなる責任も負わないものとします。このドキュメントに記載されている情報 (URL 等のインターネット Web サイトに関する情報を含む) は、将来予告なしに変更することがあります。お客様がこのドキュメントを運用した結果の影響については、お客様が負うものとします。別途記載されていない場合、このソフトウェアおよび関連するドキュメントで使用している会社、組織、製品、ドメイン名、電子メール アドレス、ロゴ、人物、場所、出来事などの名称は架空のものです。実在する名称とは一切関係ありません。お客様ご自身の責任において、適用されるすべての著作権関連法規に従ったご使用をお願いします。このドキュメントのいかなる部分も、米国 **Microsoft Corporation** の書面による許諾を受けることなく、その目的を問わず、どのような形態であっても、複製または譲渡することは禁じられています。ここでいう形態とは、複写や記録など、電子的な、または物理的なすべての手段を含みます。ただしこれは、著作権法上のお客様の権利を制限するものではありません。

マイクロソフトは、このドキュメントに記載されている内容に関し、特許、特許申請、商標、著作権、またはその他の無体財産権を有する場合があります。別途マイクロソフトのライセンス契約上に明示の規定のない限り、このドキュメントはこれらの特許、商標、著作権、またはその他の無体財産権に関する権利をお客様に許諾するものではありません。

©2010 Microsoft Corporation. All rights reserved.

Microsoft、Internet Explorer、Outlook、Silverlight、Windows、および Windows Phone は、マイクロソフト グループの商標です。

その他、記載されている会社名、製品名には、各社の商標のものもあります。

## 目次

---

Windows Phone 7 向け UI デザイン/操作ガイド .....	5
入力の種類 .....	7
タッチのためのデザイン .....	7
使いやすくアクセスしやすいレイアウト .....	8
自然で直感的な操作 .....	10
認識と反応のパフォーマンス .....	11
ユニークでワクワクするようなエクスペリエンス .....	12
サポートされるタッチ ジェスチャ .....	13
タップ .....	14
ダブルタップ .....	15
パン .....	16
フリック .....	17
縮小と拡大 .....	18
タッチしたまま押さえる .....	19
タッチのデザインに関する考慮事項 .....	19
ハードウェア ボタン .....	20
キーボード .....	22
スクリーン キーボード .....	22
ハードウェア キーボード .....	24
センサー .....	25
ユーザー インターフェイスのフレームワーク .....	25
画面の向き .....	27
フレームとページの構造 .....	28
スクロール ビューアー .....	31
テーマ .....	31
Windows Phone アプリケーションのコントロール .....	33
Windows Phone 7 でサポートされているコントロール .....	33
Windows Phone 7 でサポートされていないコントロール .....	34
プッシュ ボタン .....	35
トグル スイッチ .....	36
チェック ボックス .....	36
オプション ボタン .....	38
ハイパーリンク コントロール .....	40
スライダー コントロール .....	40
テキスト ブロック .....	42
テキスト ボックス .....	43
リスト ボックス .....	44
リスト ビュー項目 .....	44
プログレスバー .....	46
ページ タイトル .....	47

パノラマ アプリケーション .....	48
デザインに関する推奨事項 .....	49
ピボット コントロール .....	56
<b>Windows Phone システム コントロール</b> .....	57
システム トレイとアプリケーション バー .....	58
コンテキスト メニュー .....	60
<b>Windows Phone の通知</b> .....	61
プッシュ通知の概要 .....	61
プッシュ通知の種類 .....	62
進行状況のインジケータ .....	64
<b>UI テキストのガイドライン</b> .....	65
<b>Windows Phone 7 の態とトーン</b> .....	65
大文字化 .....	66
句読点 .....	67
<b>UI テキストのデザインに関する考慮事項</b> .....	67



## Windows Phone 7 向け UI デザイン/操作ガイド

Windows® Phone 7 のユーザー インターフェイス (UI) は、"Metro" というコードネームで呼ばれている Windows Phone デザイン システムを基盤としています。Metro のデザイン原理は、空港システムや地下鉄システムの標識の視覚的な言語を模倣した活字を使用する見た目を中心としています。これは、エンド ユーザーを目的のコンテンツにわかりやすく導くことを目的としています。

Metro インターフェイスは、調和のとれた、機能的で魅力的な視覚要素の実現を目指して作られています。理想を言えば、優れた UI デザインとは、アプリケーションを操作する際に楽しみながらいろいろ試してみようという気を起こさせるものであり、ユーザーが驚きや楽しさを感じるものであるべきです。明確で単純明快なデザインは、アプリケーションを見やすくするだけでなく、使おうという気持ちにさせます。このガイドでは、このような UI 開発に必要な、デザインに関する知識や基本事項を紹介します。開発者は、できる限りこの Metro のデザイン スタイルを採用することを強くお勧めします。要件はアプリケーションに応じて変わる可能性があります、Metro のデザイン スタイルに合わせることで、デフォルトもしくはカスタマイズされたアプリケーション ビューを通じて、より一貫性のある滑らかな UI エクスペリエンスが実現されます。

また、このガイドでは、標準的な入力、UI フレームワーク内の機能、Metro のテーマに沿ってデザインされた Silverlight® コントロールとシステム ベースのコントロールなど、Windows Phone 7 アプリケーションで使用される可能性がある操作方法のいくつかについて、詳しく説明します。このような領域における主要なデザイン上の概念および考慮事項を理解し取り入れることで、より優れたエンド ユーザー エクスペリエンスを実現するアプリケーションを作成することができます。また、このガイドを読むと、開発者が Windows Phone 7 で使用できる、ハードウェアおよびソフトウェアの操作要素の数に対する理解も深まります。Windows Phone 7 の操作モデルを逸脱することは通常は許可されません。ただし、アプリケーションの要件はさまざまなので、UI の動作が異なる例外がいくつかあります。

開発者が利用できる、プラットフォームの機能には、次のようなものがあります。

- ✓ **ハードウェア: Windows Phone** ハードウェアには、**WVGA** マルチタッチ スクリーンが採用されており、**GPS** 機能などのコンポーネントや加速度計などのセンサーが組み込まれています。このハードウェアでは、ユーザーは "ジェスチャ" (静電式タッチパネル上での指を使った動作) によって移動や入力を行うことができます。開発者はこうしたコンポーネントをアプリケーションに使用して、エンド ユーザーのエクスペリエンスを強化することができます。
- ✓ **UI フレームワーク: Windows Phone 7** の UI フレームワークは、エンド ユーザーに最適な表示エクスペリエンスを提供するように作られています。たとえば、アプリケーションでは、より見やすいように、縦画面表示と横画面表示を切り替えることができます。このガイドでは、**Windows Phone 7** で使用されるフレーム ページ ナビゲーション モデルについて説明します。また、アプリケーションを特定のテーマに沿ってデザインして視覚要素をカスタマイズできることについても説明します。
- ✓ **Silverlight: Windows Phone 7** で提供される **Silverlight®** UI フレームワークは、**Silverlight** の進化における次の段階であり、新たなレベルのモバイル デザイン エクスペリエンスを実現可能にします。**Silverlight** では **.NET** の技術が活用され、さまざまなコントロール、充実したレイアウト、スタイルなどが用意されています。また、**Silverlight** では、ベクター ベースのグラフィックおよびアニメーション **API** もサポートされています。一部の **Silverlight** コントロールは、**Windows Phone 7** プラットフォームを明確に対象とした新しく魅力的な外観でまとめられています。開発者は、これまでの **Silverlight** および **.NET** での開発経験をこの一連のモバイル コントロールの使用に役立て、アプリケーションに応用することができます。

以降のセクションでは、次の内容を扱います。

- ✓ 入力の種類
- ✓ ユーザー インターフェイスのフレームワーク
- ✓ **Windows Phone** アプリケーションのコントロール
- ✓ **Windows Phone** システム コントロール

## 入力の種類

---

**Windows Phone 7** アプリケーションでは、タッチ ジェスチャ、スクリーン キーボード、ハードウェア ボタン、加速度計を始めとするセンサーなど、いくつかの操作方法を利用できます。このセクションでは、開発者がアプリケーションでこれらの入力の種類を利用する方法について説明し、開発プロセスにおけるデザイン上の考慮事項を示します。

以下のセクションでは、さまざまな入力の種類の詳細について説明します。

### タッチのためのデザイン

**Windows Phone 7** のユーザー インターフェイスは、タッチ操作用にデザインされており、指によるジェスチャを組み合わせるあらゆる操作を行えるようになっています。

### サポートされるタッチ ジェスチャ

タッチ ジェスチャの例とデザイン上の考慮事項を紹介します。

### ハードウェア ボタン

**Windows Phone 7** のハードウェア ボタン レイアウトの概要を紹介します。

### キーボード

タッチ ベースの画面上でテキストを入力する際に使用されるスクリーン キーボードについて紹介します。

### センサー

加速度センサーを含む **Windows Phone 7** のセンサーについて紹介します。

## タッチのためのデザイン

---

**Windows Phone 7** のユーザー インターフェイスは、タッチ操作用にデザインされており、指でのジェスチャを組み合わせるあらゆる操作を行えるようになっています。使いやすさと直感的なアプリケーション デザインをするために、適切なサイズのタッチ ベースのコントロールを適切に配置することが重要です。以降のセクションでは、**Windows Phone** アプリケーションでこのようなエクスペリエンスを実現する方法についてのガイダンスを紹介します。このセクションでは、**UI** の視覚要素をカスタマイズするためのタッチ ガイダンスを提供しますが、通常は、デフォルトで提供される **Metro** のテーマに沿ってデザインされたコントロールを使用することをお勧めします。これらのコントロールは、タッチ操作に適したサイズに設定されており、このセクションで提供されるガイドラインに基づいて構築されています。



注:

尚、上記に関しては、ゲーム開発などの例外があります。ゲーム開発では、視覚要素のタッチのためのサイズ設定がアプリケーションによって異なる可能性があります。

## 使いやすくアクセスしやすいレイアウト

タッチ操作可能な UI 要素はすべて、指で簡単にタッチできる必要があります。そのためには、対象を指で指定し易くするために、サイズ、間隔、場所、および見た目を調整する必要があります。

### タッチ対象の最小サイズ

ピクセルを単位として使用している場合、以降のセクションでは、タッチ対象のサイズを表す mm を解像度と画面のサイズに合わせてピクセルに変換する必要があります。

- ✓ すべての UI 要素について、タッチ対象のサイズ（高さと幅）は 9 mm 以上にするをお勧めします。
- ✓ すべての UI 要素について、見た目のサイズ（高さと幅）はタッチ対象のサイズの 60 ~ 100% にする必要があります。
- ✓ タッチ対象は目に見えないため、視覚要素より大きくてもかまいません。ただし、タッチ対象を視覚要素よりも小さくはしないようにしてください。



### 大きめのタッチ対象に関する考慮事項

以下の場合、すべてのタッチ対象のサイズを 9 mm より大きくする必要があります。

- ✓ その UI 要素が頻繁にタッチされる場合。
- ✓ タッチ ミスでユーザーが間違っただけの対象をタッチした場合の結果が深刻である場合（たとえば、作成途中の SMS が送信されてしまう、電子メール メッセージが削除されてしまうなど）。
- ✓ タッチ ミスの結果がいらだたしいものである場合（たとえば、新しいダイアログ画面やアプリケーションが起動される、望ましくない画面に移動するなど）。
- ✓ UI 要素が画面の端の方にあり、タッチするのが比較的難しい場合。
- ✓ UI 要素を一連の操作の中で使用する、もしくは画面上で近接する一連の対象を複数回タッチする場合（たとえば、テンキーを使用する場合など）。



注:

スクリーン キーボードや Windows Phone® Internet Explorer® 内のハイパーリンクは、異なるサイズに設定されたタッチ対象が含まれるため例外です。



### 適切

少なくとも、各要素の間隔は 2 mm より広くする必要があります。

### 不適切

間隔が不適切な UI 要素は、タッチ ミスにつながります。

## 小さめのタッチ対象に関する考慮事項

UI 要素に使用する高さと幅は、最小でも 7 mm 以上にする必要があります。

2 つの UI 要素 (タッチ対象) の間隔は、対象のサイズが 7 mm 未満の場合 (テキスト、オプション ボタンなど)、2 mm より広くする必要があります。

## サイズ設定の制限に対する対応

サイズ設定に関する特別な制限がある UI 要素では、次の対応策のいずれかまたは両方を使用します。

- ✓ タッチ対象のサイズを 9 mm にできるように、UI 要素の間隔を広げます。
- ✓ 垂直方向のサイズが限られている場合は、タッチしやすい長方形のコントロールを使用します。タッチ対象の幅が 20 mm を超えるのであれば、高さは 7 mm にまで縮めることができます。

## 自然で直感的な操作

**Metro** のデザイン原理は、**UI** 要素は本当にデジタルであるべきだという概念に基づいています。デジタルのメタファーを使用するのが自然で適切な場合は使用するべきであり、現実世界の操作を模倣するのが適切でない場合は必ずしも模倣する必要はありません。開発者は、「**UI** というのは、現実世界の動作を単に見ただけで模倣するのではなく、外観やその操作感が優れたものであるべきだ」という概念を理解する必要があります。また、必要であれば、直接操作を通じて現実世界の動作を模倣してみるといいでしょう。

### インターフェイスとしてのコンテンツ

- ✓ 写真や **Web** ページなどのアプリケーション コンテンツを **UI** として使用して、ユーザーがこうした **UI** をタッチによって操作できるようにします。
- ✓ 繰り返しになりますが、一貫性を維持するため、**Windows Phone 7** の一般的な **Silverlight** コントロールを使用するべきでしょう。

### 自然な反応

- ✓ 写真をパンして移動するなど、ジェスチャを使用して現実世界の動作を模倣します。
- ✓ ジェスチャはタスクへのショートカットとして使用せず、意図されたとおりの方法でのみ使用してください。タッチ ジェスチャと機能の詳細については、「[サポートされるタッチ ジェスチャ](#)」を参照してください。

### 単純で一貫性がある

- ✓ **Windows Phone 7** では、片手でより簡単に操作できるシングルタッチ ジェスチャを使用できます。また、姿勢を変える必要がありますが、より高度なジェスチャ機能を提供するマルチタッチ ジェスチャも利用できます。**Windows Phone 7** のタッチ ジェスチャの詳細については、「[サポートされるタッチ ジェスチャ](#)」を参照してください。
- ✓ 基本的なタスクや一般的なタスクはすべて **1** 本の指で完了する必要があります。
- ✓ ジェスチャへの反応は、デバイス全体で一貫している必要があります。提供される、**Metro** のテーマに沿ってデザインされたコントロールを使用すると、一貫性を維持するのに役立ちます。

## 認識と反応のパフォーマンス

現実世界の対象物は触れるととすぐに反応します。Windows Phone 7 の UI でも、この規則に従うことを強くお勧めします。反応が遅い UI や遷移に時間がかかる UI は、ユーザー エクスペリエンスに悪い影響を与えます。

### タッチに対するフィードバック

- ✓ UI 要素が操作されたことを示す視覚的なフィードバックを提供するようにしましょう。たとえば、Windows Phone 7 では、タッチすると、"傾く" アニメーションが使用されます。同様の理由で、写真や地図は傾きません。また、要素が操作されたことを動きで示すのもいいでしょう。
- ✓ 要素が操作されたことを音によるフィードバックで示すこと一つの方法です。

### 反応性が優れた UI

- ✓ UI 要素はタッチされるとすぐに反応する必要があります。ユーザーがその反応を待たなくて済むようにしなくてはなりません。なぜなら、待ち時間が発生すると、集中が途切れ、操作の流れが中断されるからです。すべての操作に対しては、明白な結果がすぐに返される必要があります。
- ✓ UI 要素は、ジェスチャが発生した後ではなく、発生している間に反応する必要があります。たとえば、ユーザーが写真をフリックした場合に、ジェスチャが完了した後で動作が発生するのは望ましくありません。
- ✓ パンがフリックに変わったり、タップがダブルタップに変わったりするなど、あるジェスチャから別のジェスチャへ移行したり連結することもできます。



### 右方向へのジェスチャ

アプリケーションは、タッチ ジェスチャにすぐに反応する必要があります。たとえば、あるページからタイム ラグなしでスムーズに遷移する必要があります。また、遷移は動作の完了後ではなく動作中に行われる必要があります。

## 時間がかかる処理

- ✓ 時間のかかる処理が発生している場合、プログレスバーを通じて、何かが起こっていることをユーザーがわかるようなフィードバックをスマートに提供する必要があります。コンテンツを使用して進行状況を示すのです。たとえば、コンテンツのダウンロードが進むにつれて表示するコンテンツを増やします。プログレスバー コントロールは、1つの処理または一連のイベントに関連する、アプリケーション内の処理状況を示します。このコントロールは開発者が独自に構築することもできますが、Windows Phone 7 で提供されています。プログレスバーは必要な場合にのみ使用し、メッセージ通知は最後の手段として使用するようにしてください。詳細については、「[プログレスバー](#)」を参照してください。

## ユニークでワクワクするようなエクスペリエンス

**Windows Phone** アプリケーションを使用するとき、ユーザーはデバイスを操作しながら驚きを感じるでしょう。ナビゲーションと操作性を強化し、ワクワクするような見た目にするすることで、ユーザーを引き付けます。

### 発見

- ✓ タッチ ジェスチャを使用した、コンテンツの発見を促します。ユーザーは、タスクの完了だけでなく、完了までの過程も楽しめるでしょう。

### ユーザーを楽しませる

- ✓ タッチ ジェスチャは、1つのジェスチャが実行される場合も、連続したジェスチャの組み合わせが実行される場合も、アプリケーションに楽しい使用感をもたらします。タッチ ジェスチャでは、ユーザーは操作とその結果として行われる動作とが対応していることを確認できるので、楽しむことができます。1つのジェスチャも、マルチタッチ ジェスチャも、また、連続したジェスチャも、より生き生きとした直感的なエクスペリエンスを提供します。

## サポートされるタッチ ジェスチャ

---

タッチ ジェスチャとは、タッチ スクリーン上で 1 本または複数の指を使用してなんらかの動作を実行することです。タッチ ジェスチャの例としては、プッシュ ボタンなどの UI 要素のタップが挙げられます。タッチ ジェスチャは、ユーザーが Windows Phone を操作するための主な方法です。Windows Phone 7 でサポートされるタッチ ジェスチャとその使用方法および動作に関する詳細な参考情報を以下に示します。提供される Metro のテーマに沿ってデザインされたコントロールをタッチ操作の要素として使用します。繰り返しになりますが、これらのコントロールは、タッチ操作に適したサイズに設定されています。Windows Phone 7 では、次のようなシングルタッチ ジェスチャとマルチタッチ ジェスチャを使用できます。

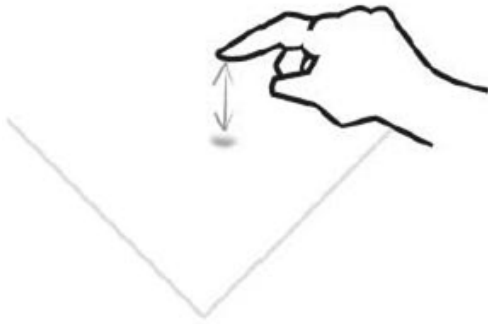
### シングルタッチ:

- ✓ タップ
- ✓ ダブルタップ
- ✓ パン
- ✓ フリック
- ✓ タッチ アンド ホールド

### マルチタッチ:

- ✓ 縮小と拡大

## タップ



画面をシングル タッチします。

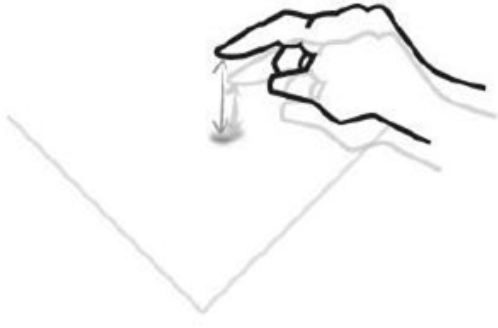
特定の領域内の 1 点に指を下ろし、すぐに指を画面から離します。

タップ ジェスチャに関連する動作は次の 2 つです。

- ✓ 指を画面に下ろすと、タッチであることが示されます。
- ✓ 指を画面から離すと、アクションが実行されます。

タップを行うと、画面上のあらゆる種類のコンテンツの動きが止まります。

## ダブルタップ



特定の領域内で 2 回すばやくタップします。

ダブルタップを行うと、コントロールやアプリケーションの拡大と縮小の状態が切り替わります。アプリケーションは現在の拡大/縮小の状態を特定し、それに応じて拡大または縮小を行います。拡大と縮小の状態は、アプリケーションによって定義されます。

## パン



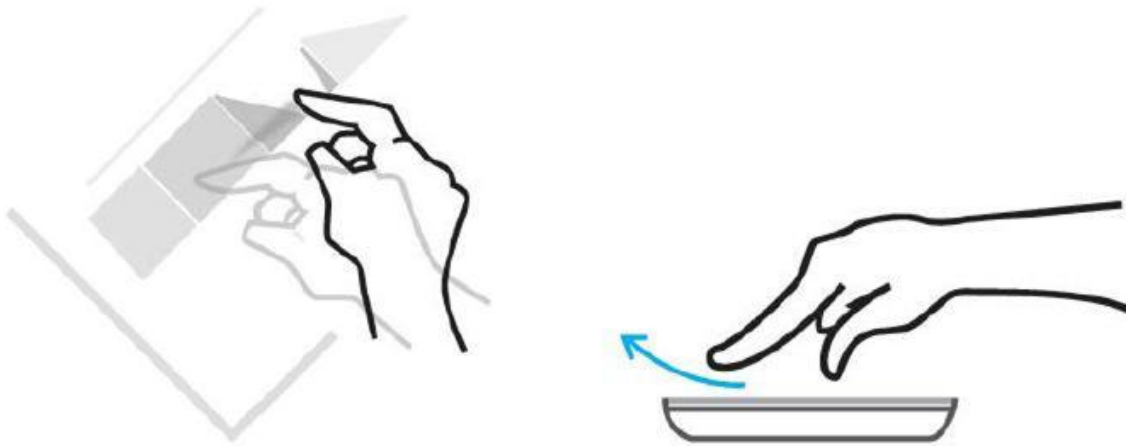
指を画面に下ろした後、一方向または複数方向に指を動かします。

パンは、指を画面から離すと（または別のジェスチャが開始されると）終了します。

パン ジェスチャに関連する動作は次の 2 つです。

- ✓ コンテンツを直接操作によって移動することができ、コンテンツは指から離れずについていきます。サポートされるパンの方向は、コントロールやアプリケーションが決定できます。パンの方向として、水平方向、垂直方向、または指定されたその他の方向をサポートできます。また、コンテンツが中間の状態まで移動された場合、コンテンツは最も近い状態に戻ります。
- ✓ パンを行うと、特定のアイテムを移動したり並べ替えたりできます。アイテムは指から離れずについていき、指が画面から離れたときに新しい場所にドロップされます。

## フリック

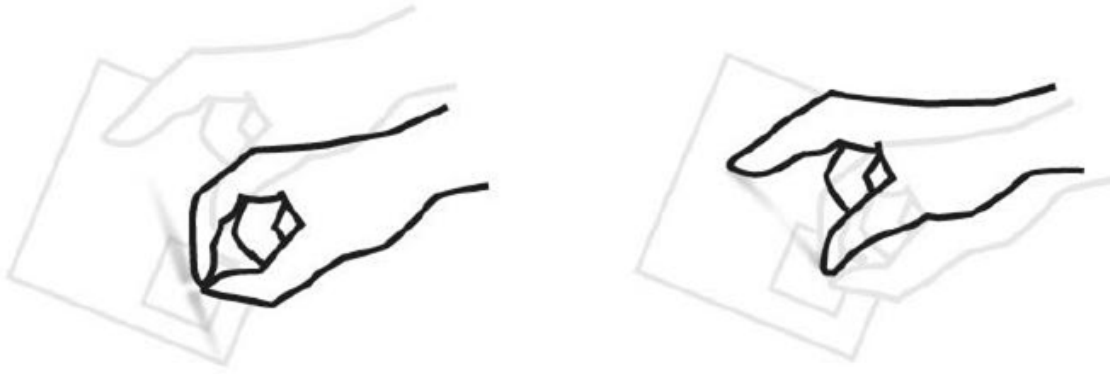


指を画面に下ろした後、一方向に指をすばやく動かし、指を画面から離します。

フリックは、パン ジェスチャの後に続けて行うこともできます。

フリック ジェスチャを行うと、コンテンツがある領域から別の領域に移動されます。サポートされるフリックの方向は、コントロールやアプリケーションが決定できます。フリックの方向として、水平方向、垂直方向、または別の指定された方向をサポートできます。水平または垂直の方向が選択された場合、その他の方向に指を動かすと、垂直方向または水平方向の移動に変換されます。

## 縮小と拡大



2本の指を画面に下ろした後、その2本の指を近づけるか、離します。

縮小と拡大のジェスチャを行うと、2本の指の中心点を拡大/縮小の中心点として、コンテンツがスムーズに拡大または縮小されます。

## タッチ アンド ホールド



特定の領域内の 1 点に指を下ろし、決められた時間そのままにします。

タッチしたまま押さえるジェスチャを行うと、アイテムのコンテキスト メニューやオプション ページが表示されます。

## タッチのデザインに関する考慮事項

アプリケーションのタッチのデザインに関する考慮事項は次のとおりです。

- ✓ **Windows Phone 7** の **Silverlight** コントロールはすべて、このセクションで説明したタッチ ジェスチャを標準サポートします。
- ✓ 一貫性のあるジェスチャ動作を維持する必要があります。カスタムのコントロールや **UI** を作成する場合、そのようなコントロールや **UI** はジェスチャに同じように反応する必要があります。
- ✓ **Windows Phone 7** のジェスチャは、**Windows** デスクトップのジェスチャと一貫性があります。ただし、画面サイズの差や **Windows** デスクトップではマウスがサポートされることを理由とした、意図的な違いもあります。このような違いのほとんどはショートカットの編集に関するものなので、スクリーン キーボードで対処できる可能性があります。**Windows Phone 7** のアプリケーションでは、ジェスチャ エクスペリエンスが、対応する **Windows** デスクトップ アプリケーションとの一貫性を持つようにしてください。
- ✓ **Windows Phone 7** では、ジェスチャの拡張はサポートされていません。このトピックで紹介したサポートされるジェスチャのみを使用して、ここで述べたとおりの動きを再現することができます。

## ハードウェア ボタン

Windows Phone デバイスには、いくつかのハードウェア ボタンが付いています。各ボタンには、実行中のアプリケーションを調整したり、実行中のアプリケーションに影響を与えたりする可能性がある別々の機能が備わっています。

- 1 電源/スリープ
- 2 音量
- 3 画面
- 4 カメラ
- 5 戻る
- 6 スタート
- 7 検索



注:

ハードウェア ボタンの機能を、アプリケーション内でプログラムを使用して拡張または変更することはできません。

### スタート ボタン

ユーザーがこのハードウェア ボタンを押すと、デバイスで何を表示していても、スタート メニューが表示されます。アプリケーションを実行している場合でも同様です。

### 検索 ボタン

検索ボタンを押すと、ユーザーがデバイスで何を表示していても、コンテンツを探すための検索機能が起動します。

## 戻るボタン

ハードウェアの戻るボタンを使用すると、アプリケーション内で前の操作に戻ったり、あるアプリケーションから前のアプリケーションに戻ったりすることができます。このボタンが押されると、アプリケーションではフレームワークに操作の実行を許可します。また、戻るボタンを使用すると、スクリーン キーボードや、メニュー、ダイアログを閉じたり、また前の画面に移動したり、さらに検索操作を終了したりすることができます。さらには、アプリケーションを切り替えることも可能です。ただし、現在の画面から前の画面に移動するというのが主な使い方です。



注:

戻るボタンは、入力したテキストを削除する **BackSpace** キーとしては使用できません。

## カメラ ボタン

機能面から見ると、**Windows Phone 7** のカメラ ボタンは、全押しモードと半押しモードをサポートするデュアル アクションです。ユーザーがこのボタンを全押しすると、カメラ アプリケーションが起動します。半押しすると、自動フォーカス機能が有効になります。プログラムの観点から言うと、アプリケーションでは、カメラを制御する **API** を呼び出すことで、カメラ機能を起動できます。また、アプリケーションでは **API** を呼び出すタイミングと場所を決めることもできます。



注:

繰り返しになりますが、アプリケーションでは、ハードウェアのカメラ ボタンの動作を上書きすることはできません。

## 音量ボタン

音量ボタンは、主に、アクティブなオーディオ ストリームの音量を調整するために使用されます。アクティブなオーディオ ストリームとしては、電話の呼び出し音、音楽のオーディオ、ラジオのオーディオ、再生中のムービーなどが考えられます。オーディオ ストリームがアクティブでない場合、音量ボタンを使用して、アクティブなサウンド プロファイルを変更できます。音量コントロールを使用すると、前へ、次へなどのオーディオ トランスポート コントロールも表示されます。

音量ボタンの詳細情報

- ✓ 電話の着信があった場合、音量ボタンを使用して着信音を消すことができます。
- ✓ アクティブなオーディオ ストリームを再生している場合、音量ボタンを使用して、アクティブなオーディオ ストリームの音量レベルを変更できます。システム サウンドのイベントや効果は例外となります。
- ✓ デバイスがロックされていても、音量ボタンはアクティブなままとなり、実行された操作に反応します。開発者はこの機能を変更できません。
- ✓ ユーザーが音量ボタンを押したままにすると、キーが繰り返し押されているのと同じことになります。音量を上げるキーを押し続けると、徐々に音量が上がります。

## 電源ボタン

デバイスを使用しているときに電源ボタンを短めに押すと、ディスプレイが消え、デバイスがロックされます。デバイスを使用していないときに電源ボタンを短めに押すと、スリープが解除されます。

## キーボード

---

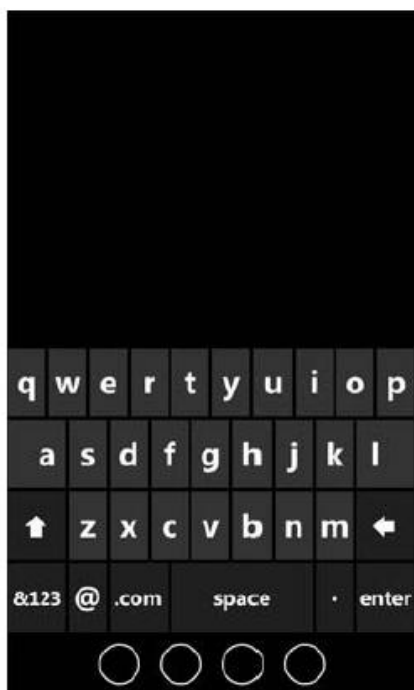
### スクリーン キーボード

Windows Phone でのテキスト入力には、スクリーン キーボードまたはハードウェア キーボードを使用できます。スクリーン キーボードは、編集コントロールがアクティブになると自動的に展開されます。ユーザーが編集コントロールの外をタップすると、スクリーン キーボードは閉じられます。デバイスにハードウェア キーボードが搭載されている場合、ハードウェア キーボードを展開するとスクリーン キーボードは自動的に閉じられます。Windows Phone 7 のデバイスには、テキスト候補やオートコレクトなど、いくつかの入力補助機能が用意されています。これらの機能は、スクリーン キーボードとハードウェア キーボードの両方に適用されます。



注:

デバイスにハードウェア キーボードが搭載されるかどうかは、機種によって異なります。



## スクリーン キーボードのデザインに関する考慮事項

- ✓ アプリケーションでキーボードが展開される場合、キーボードは画面の下部に表示されます。
- ✓ テキスト候補が有効になっている場合、候補はキーボードの上にドッキングされ、アプリケーションの表示領域がさらに狭くなります。
- ✓ キーボードが展開されている場合、アクティブな編集コントロールとキャレットが表示されるように、アプリケーションがスクロールする必要があります。複数行の編集コントロールを使用している場合、コントロールの一部がスクリーン キーボードの背後に隠れてしまうことがあります。キャレットが含まれている行が必ずスクリーン キーボードの上に表示されるようにする必要があります。
- ✓ アプリケーションによっては、ユーザーがテキスト ボックスや編集コントロールの外をタップして、スクリーン キーボードを閉じ、隠れていたコンテンツを表示するのが難しいことがあります。たとえば、アプリケーションが画面領域のすべてまたは大半を占めている場合などです。この場合、ユーザーが入力ではなくコンテンツの表示を行おうとしているときはアプリケーションで自動的にスクリーン キーボードを閉じるようにすることができます。たとえば、ユーザーがアプリケーション領域をスクロールしたときや、アクティブな編集コントロールが可視領域の外に出たときに、スクリーン キーボードを自動的に閉じることができます。読み取りビューと編集ビューを実装するという方法もあります。その場合は、ユーザーが編集ビューに切り替えたらアプリケーションでスクリーン キーボードを展開します。ユーザーが戻るボタンを押すなどして読み取りビューに戻ったら、アプリケーションでスクリーン キーボードを閉じます。
- ✓ キーボードでは、テキスト ボックスや編集フィールドのコンテキストに応じて複数のキー レイアウトがサポートされます。
- ✓ 開発者は、適切なレイアウトを選択して適切な入力補助機能を有効にするために、編集フィールドに入力スコープを設定する必要があります。たとえば、開発者が URL の入力スコープを選択すると、「.com」キーを含むキーボード レイアウトが表示されます。これを行うには、プロジェクト内で、テキスト ボックスや編集コントロールの入力スコープ プロパティを設定する必要があります。

次のように、目的に応じていくつかのキーボード レイアウトがあります。

キーボードの種類	レイアウト
既定	標準的な QWERTY レイアウト
テキスト	ASCII ベースの絵文字を含む標準的なレイアウト
電子メール アドレス	.com キーと @ キーを含む標準的なレイアウト
電話番号	一般的な 12 キーのレイアウト
Web アドレス	.com キーとカスタマイズされた Enter キーを含む標準的なレイアウト
マップ	カスタマイズされた Enter キーを含む標準的なレイアウト
検索	Search (検索) キーと .com キーを含む半透明のレイアウト
SMS アドレス	電話番号レイアウトに簡単にアクセスできる標準的なレイアウト

## ハードウェア キーボード

Windows Phone にハードウェア キーボードが搭載されるかどうかは、デバイスの製造元しだいです。Windows Phone 7 では、引き出し式バー、縦スライド構成、向きの反転や回転などのデザインを含むいくつかのハードウェア キーボードを使用できるようになります。



注:

Windows Phone 7 では、QWERTY、AZERTY、QWERTZ など、完全なアルファベットのレイアウトのみがサポートされます。12 キーまたは 20 キーのレイアウトはサポートされません。

ハードウェア キーボードを使用すると、文字、アクセント付き文字、数字、および記号を入力できます。

- ✓ **Shift** キーを使用すると、大文字を入力できます。**Shift** モードには、オン、オフ、およびロック (CapsLock) の 3 つがあります。
- ✓ **Emoticon** (絵文字) キーを使用すると、絵文字ピッカーが表示されます。
- ✓ **Accent** (アクセント) キーは、アクセント付き文字の入力に使用します。**Accent** (アクセント) キーを押すと、キュレットの左側にある文字にアクセント記号が追加されます。このキーを繰り返し押すと、使用できるアクセントが順番に表示されます。ファンクション キー (FN) を押しながら **Accent** (アクセント) キーを押すと、アクセントが逆順で表示されます。**Accent** (アクセント) キーを押したままにすると、アクセント ピッカーが表示されます。
- ✓ **Symbols** (記号) キー (SYM) を押すと表示される記号ピッカーを使用すると、キーボード上にない文字にアクセスできます。
- ✓ **SYM** キーを押したままにすると、言語ピッカーが表示されます。FN キーを押しながら **SYM** キーを押すと、次の言語に切り替わります。
- ✓ システム トレイの入力インジケータでは、**Shift** モード、**FN** モード、およびアクティブな言語が表示されます。
- ✓ キーボードのキーはオーバーロードできます。キーを押したままにするか、FN キーを使用すると、2 次機能にアクセスすることができます。

### ハードウェア キーボードのデザインに関する考慮事項

- ✓ ハードウェア キーボードは入力にのみ使用され、UI の操作には使用されません。
- ✓ ハードウェア キーボードには、編集コントロール内でキュレットを移動できる方向キーを含めることができます。ただし、このような方向キーを、フォーカスの移動、リストのスクロール、およびマップ上

や Web ページ間の移動に使用するべきではありません。

- ✓ 次のキーは、ハードウェア キーボードで必ず使用できます。
  - ✓ 文字 (A ~ Z)、Enter、Space、BackSpace、Shift、Emoticon (絵文字)、SYM、ピリオド (.), およびコンマ (,)
  - ✓ 1 次的または 2 次的な文字としての数字 (0 ~ 9)
  - ✓ ドイツ語、フランス語、イタリア語、およびスペイン語のキーボードにおける Accent (アクセント) キー
- ✓ 次のキーは、ハードウェア キーボードでサポートされなくなりました。
  - ✓ 方向パッドなど、ナビゲーション専用のハードウェア
  - ✓ "OK/ホーム" および "送信/終了" を表すハードウェア ソフト キー
  - ✓ Del、Ins、Ctrl、Alt、CapsLock、Tab、PageUp、PageDown、および Esc の各キー
  - ✓ キーボードの一部としての Start (スタート) キー、Search (検索) キー、および Back キー
- ✓ 記号ピッカー、アクセント ピッカー、および言語ピッカーは、起動されると画面の下部に表示されます。ユーザーが記号/アクセント/言語を選択するか、事前に定義されたタイムアウトに達すると、これらのピッカーは画面から消えます。
- ✓ アプリケーションは、API を使用して、ハードウェア キーボードが使用可能になったり展開されたりしたタイミングを認識し、それに応じて動作します。
- ✓ ハードウェア キーボードを展開すると、スクリーン キーボードは閉じられます。デバイスにハードウェア キーボードが備わっている場合、スクリーン キーボードは展開されません。
- ✓ 入力補助機能はハードウェア キーボードでも使用できます。

## センサー

---

### 加速度計

**Windows Phone 7** の加速度計は、重力や外的要因によって生じる加速を測定する電子機械デバイスであり、デバイスの X 面、Y 面、および Z 面に対して加えられる力に関する情報を継続的に提供する 3D モーション センサーです。**Windows Phone** では、優れたエンド ユーザー エクスペリエンスを実現するためにこの機能を活用することができます。画面を自動で回転する、本体を傾けることでスクロールする、ゲームをするなどのシナリオを柔軟に実現できる使いやすいマネージ API を提供することが、デザインの目標の一部です。

### アプリケーションのデザインに関する考慮事項

高いレベルの精度と感度が必要なアプリケーションやゲームを作成する開発者は、調整されたデータを使用する必要があります。



注:

**Windows Phone 7** アプリケーション プラットフォームの では、調整用の API は公開されていません。この機能は後のリリースで追加される予定です。

## ユーザー インターフェイスのフレームワーク

---

**Windows Phone 7** のユーザー インターフェイスは、エンド ユーザーに最適な表示エクスペリエンスを提供するように作られています。このセクションでは、開発者向けに、全画面表示モード、画面の向き、フ

フレームおよびページのナビゲーション、アプリケーションのスクロールなどのトピックについて説明します。

このセクションでは、以下のトピックについて説明します。

#### 画面の向き

縦長と横長の 2 種類の向きをサポートしています。

#### フレームとページの構造

**Windows Phone 7** アプリケーションは、ユーザーがさまざまなコンテンツ画面を移動できる **Silverlight** ページ モデルを基にしています。

#### スクロール ビューアー

コンテンツが画面のサイズよりも大きい場合、アプリケーションのスクロールが発生します。

#### テーマ

テーマは、**Windows Phone** 上の視覚要素をカスタマイズするために使用されるリソースです。

## 画面の向き

Windows Phone は、縦長、横長 (左)、および横長 (右) の 3 種類の画面の向きをサポートしています。横長 (左) と横長 (右) の違いは、アプリケーション バーとシステム トレイの向きです。



注:

アプリケーションでは、アプリケーション バーとシステム トレイの位置を左側のみまたは右側のみに指定することはできません。



## 縦長の向き

縦長モードでは、ページは垂直方向に配置され、ページの高さがページの幅より大きくなります。向きは、デバイスを回転する、スライド式ハードウェア キーボードを引き出すなどのイベントに応じて変化します。向きが切り替わると、向きを認識するように作成されているアプリケーションは、新しい向きに対応します。

## 横長の向き

横長モードにするは、ページの幅がページの高さより大きくなるようにデバイスを横向きに動かします。次のような場合には、横長の向きを使用します。

- ✓ 使用できる場合は横長のスライド式キーボードを使用して、電子メール、テキスト メッセージなど、タイピングによる入力を行う。
- ✓ 横長モードに適した **Web** ページを閲覧する。
- ✓ カメラで写真を撮る。
- ✓ 1 枚の写真を表示したり、スライドショーを見たりする。
- ✓ ムービーやビデオを見る。
- ✓ ゲームをする。

## アプリケーションのデザインに関する考慮事項

アプリケーションでの縦長モードと横長モードの使用についての、デザイン上の考慮事項を以下に示します。

- ✓ 複数の向きをサポートするように、アプリケーションを構成する必要があります。アプリケーションの特定のページで、横長の向きと縦長の向きの両方がサポートされている場合、ユーザーは、物理的にデバイスを回転したり縦長または横長のキーボードを展開したりすることで、向きを変えることができます。向きのプロパティは読み取り専用で設定されているため、プログラムを使用して向きを直接切り替える方法はありません。間接的に切り替える方法としては、サポートされている向きのプロパティを希望の向きに設定します。
- ✓ 横長モードに変更すると、システム トレイとアプリケーション バー、アプリケーション メニュー、音量に関するユーザー インターフェイス コンポーネント、通知、およびバッテリー低下の通知などのシステム ダイアログは、アプリケーション内で横向き対応になります。
- ✓ 縦長モードまたは横長モードでのテキスト入力は、ハードウェア キーボード (使用できる場合) またはスクリーン キーボードを通じて行います。

## フレームとページの構造

このセクションでは、アプリケーションのフレームとページの構造について詳しく説明します。また、**Windows Phone 7** 向けのカスタム UI フレームワーク デザインについても説明します。

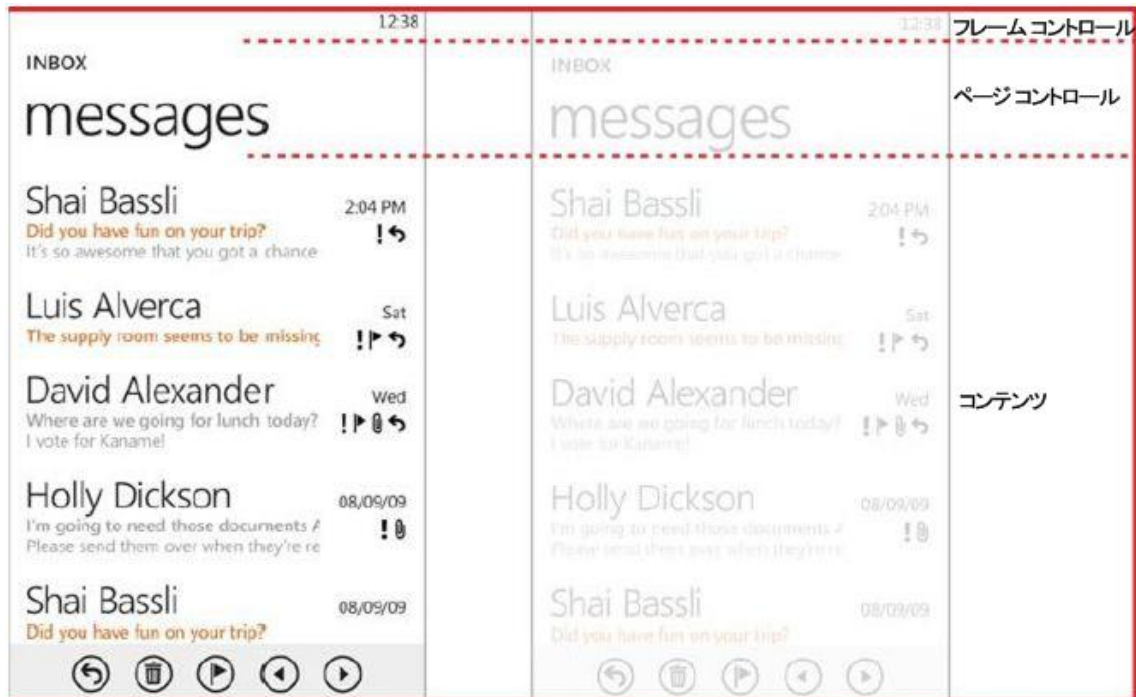


注:

このセクションは、このドキュメントの今後のリリースで、より詳しい内容になる予定です。

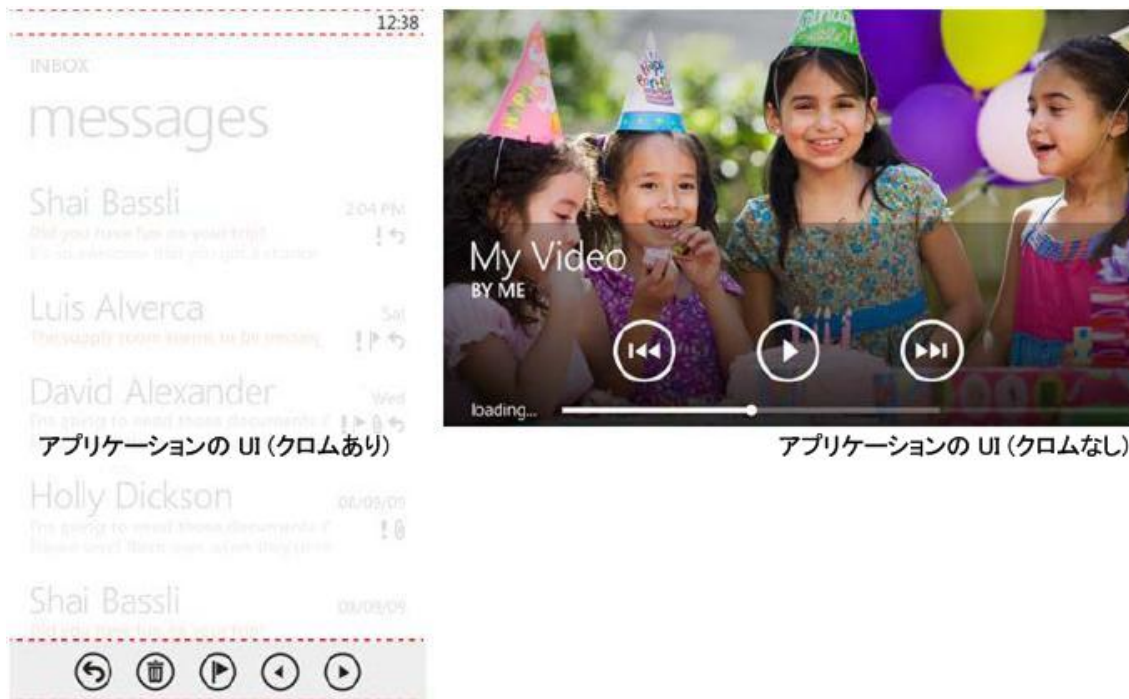
**Windows Phone 7** アプリケーションは、ユーザーがさまざまなコンテンツ画面を移動できる **Silverlight** ページ モデルを基にしています。また、**Windows Phone** の "戻る" ハードウェア ボタンを使用して前に戻ることもできます。このモデルの目標は、**Windows Phone 7** のナビゲーション モデルに自然に適合するビュー ベースのアプリケーションを簡単に作成できるようにすることです。

主要な要素には、**PhoneApplicationPage** (ページ) をホストできる **PhoneApplicationFrame** (フレーム) と呼ばれる最上位のコンテナ コントロールがあります。ページには、アプリケーションのコンテンツの個別のセクションが保持されます。**Windows Phone 7** には、別々のセクションのコンテンツへの移動を容易にするフレーム クラスとページ クラスが用意されています。アプリケーションのコンテンツを表示するのに必要な数のページを作成し、フレームからそれらのページに移動します。次の画像は、アプリケーションのフレームとページの階層の一例を示しています。



## カスタムの画面 **UI** インターフェイス

次の画像は、ページのコンテキストが通常表示から全画面表示に切り替わる状態を示しています。このとき、アプリケーションでは、システム トレイやアプリケーション バーを含めるかどうかを選択できます。ただし、これは自動的には行われないので、開発者がこの動作を実装する必要があります。たとえば、**Windows Phone** でゲームを開始し、ゲームの設定が開かれたとします。ユーザーがその設定メニューから新しいゲームを開始すると、ページが全画面表示モードで開かれるべきです。



注:

上記は、カスタムの全画面表示 **UI** モードを使用する場合のベスト プラクティスと考えられています。アプリケーションが全画面表示でのみ使用されている間、電話の着信を含む視覚的な通知はすべて、通常どおり **UI** に渡すことができます。

アプリケーションを全画面表示モードで起動すると、システム トレイとアプリケーション バーは表示されません。開発者は、アプリケーションのアプリケーション バーを非表示にしたり表示したりできます。アプリケーション バーとシステム トレイの表示/非表示を切り替えることができる可視性のプロパティ (**Page.FullScreen**) があります。

## スクロール ビューアー

スクロール表示コントロール内のコンテンツがコントロールの範囲内に収まりきらない場合、アプリケーションのスクロールが発生します。コンテンツは、ページのスクロールを処理するスクロール パネル内にラップされます。これにより、コンテンツの長さや幅がページの長さや幅よりも大きいかどうか、およびページ上の現在の位置をスクロール インジケータで示すことができるようになります。

### アプリケーションのデザインに関する考慮事項

- ✓ スクロール インジケータは、垂直方向のスクロールの場合は右側に、水平方向のスクロールの場合は下部に配置されます。スクロール インジケータは、ページを操作する際にのみ表示されます。
- ✓ 画面に対するジェスチャ（フリック、パンなど）が行われない場合、スクロール インジケータはしばらくするとタイムアウトします。このようなタッチ ジェスチャが行われると、スクロール インジケータは再び表示されます。
- ✓ スクロール インジケータは、コンテンツ内のユーザーの位置を示します。たとえば、ユーザーがページの上部にいる場合、スクロール インジケータは上部に表示されます。
- ✓ リスト ボックスにたくさんのコンテンツが含まれる場合、スクロール ビューアー内にリスト ボックスを配置するのは避けてください。スクロール ビューアー内に配置すると、パフォーマンスが低下する可能性があります。

## テーマ

テーマは、Windows Phone 上の視覚要素をカスタマイズするために使用されるリソースです。Windows Phone 7 のアプリケーション開発者は、スタイルの面でネイティブのデバイス ユーザー インターフェイス (UI) の外観や雰囲気維持されたアプリケーションを作成することができます。このスタイルのプロパティには、背景色や強調色が含まれます。テーマに対応することで、コントロールや UI 要素の外観がプラットフォーム全体で一貫性のあるものとなり、東夷鬱間のないユーザー エクスペリエンスになってしまうことを回避できます。

この機能の目標のいくつかを以下に示します。

- ✓ Windows Phone 7 のアプリケーション開発者は、アプリケーションを構築し、テーマ対応のプロパティにコード内で直接アクセスすることができます。
- ✓ 開発者は、テーマ対応の任意のプロパティの値を、独自のデザインの要件に合わせて明示的に変更することができます。

アプリケーション プラットフォームでテーマを使用するメリットは、Metro のデザイン原理との一貫性と互換性を確保できる点です。開発者は、色などの一般的なプロパティを調整することなく、既定のコントロール セットを使用することができます。このようなスタイルはシステム テーマ ファイルを通じて実行時に変更されることがわかっているからです。Windows Phone 上でアプリケーションを実行すると、このシステム テーマ ファイルが自動的に取得され、その内容に応じてアプリケーションの外観が変更されます。また、開発者は、アプリケーション レベルでテーマを変更できます。たとえば、ある企業が強力なブランド カラーを使用してアプリケーションを構築しており、この色を維持したいと考えているとします。開発者は独自のリソースを提供して、テーマに対応した任意のプロパティを変更することができます。ただし、テーマを無効にすることはできません。



注:

テーマの一部となっているのは色だけです。フォント、コントロールのサイズなど、その他の要素は変化しません。



## アプリケーションのデザインに関する考慮事項

- ✓ ユーザーは **light** または **dark** テーマを選択でき、開発者は **UI** を開発する際、これを考慮に入れる必要があります。たとえば、アプリケーション全体に白一色の背景を選択すると、有機 **LED** ディスプレイのバッテリー寿命に影響を与える可能性があります。開発中は常に、アプリケーションの背景がバッテリー寿命にどのような影響を与えるかに注意を払う必要があります。
- ✓ ユーザーは、アプリケーションの強調色を **5** 種類の中から選択できます。アプリケーションで強調色のテーマ プロパティを使用する場合、これらの色は、ユーザーの好みに応じて変化します。強調色には、オレンジ、青、緑、および赤があります。デバイスのメーカーは、最後の **1** つである **5** つ目の強調色を提供することができます。
- ✓ システム全体にわたるテーマは変更できません。変更できるのはアプリケーションのテーマのみです。
- ✓ コントロールの前景色または背景色を明示的に設定する場合、暗いテーマと明るいテーマのどちらでもコンテンツがはっきり見えることを確認する必要があります。選択した色がはっきり見えない場合は、より適切な色を選択するか、色の違いがはっきりするように背景または前景を明示的に設定します。

## Windows Phone アプリケーションのコントロール

Windows Phone 7 の UI は、Metro と呼ばれるデザイン システムを基盤としています。Windows Phone 7 には、アプリケーションで使用できる、そのような Metro のテーマに沿ってデザインされた Silverlight コントロールが用意されています。Metro インターフェイスは、統一感があり、機能的で魅力的な視覚要素の実現を目指して作られています。開発者は、できる限り Metro のデザイン スタイルを採用することを強くお勧めします。UI の要件はアプリケーションに応じて変わる可能性があります、Metro のデザイン スタイルに合わせることで、アプリケーションでは、より一貫性のあるスマートな UI エクスペリエンスが実現されます。このトピックでは、Metro のテーマに沿ってデザインされた Windows Phone 7 用の Silverlight コントロールを示します。また、このプラットフォームでサポートされていないコントロールの一覧も示します。このセクションに続くトピックでは、使用可能なコントロールの一部についての詳細な概要とガイダンスを示します。

注:

次のセクションのトピックでは、Windows Phone 7 のコントロールの一部のみを扱います。

## Windows Phone 7 でサポートされているコントロール

Border	Button	Canvas	Check Box	Content Control
Content Presenter	Control	Grid	Hyperlink Button	image
Ink Presenter	List Box	Media Element	Multi Scale Image	Panel
Password Box	Progress Bar	Option Button	Scroll Viewer	Slider
Stack Panel	Text Block	Text Box	Phone Application Page	List View Item
List View	Toggle Control Switch	Phone Application Frame	User Control	

## Windows Phone 7 でサポートされていないコントロール

Combo Box	Message Box	Open File Dialog	Save File Dialog	ToolTip
Calendar	Data Grid	Date Picker	Frame (Use PhoneApplication Frame Class)	Page (User PhoneApplication Page Class)
Grid Splitter	Label	Scroll Bar	Tab Control	Tree View

Windows Phone 7 の リリースでは、上記のサポートされていないコントロールに関して次の点に留意してください。

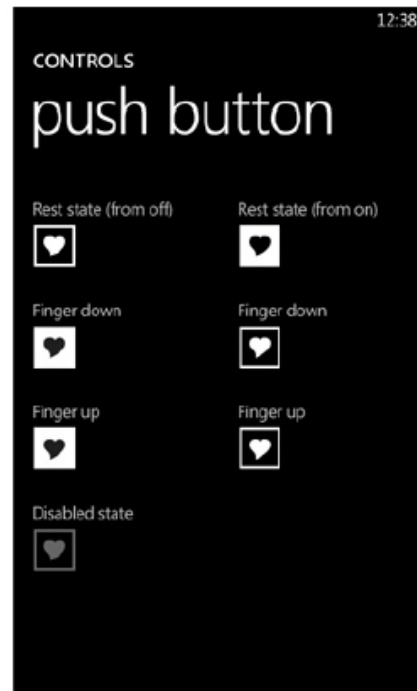
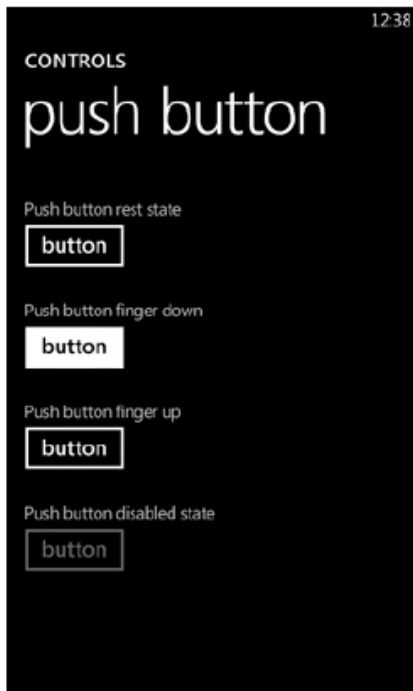
- ✓ **Combo Box**、**Scroll Bar**、および**Tool Tips** は、ツールボックスには表示されませんが、アプリケーションでは使用できます。コンボ ボックスは **Metro** のテーマに沿ったデザインになっていないので、UI の一貫性を維持するために、**Metro** のテーマに沿ったデザインにする必要があります。スクロールバーとヒントは、タッチ ベースのシステムで使用するのに適したコントロールではありません。
- ✓ **Open File Dialog** と **Save File Dialog** はサポートされていません。アプリケーションで分離ストレージを使用する必要があります。
- ✓ カレンダー、データ グリッド、日付の選択、グリッド スプリッター、タブ コントロール、およびツリー ビュー コントロールは、**Silverlight 3.0** プラットフォームで提供されるものであり、使用できません。これらのコントロールの独自のバージョンを作成し、**Metro** のテーマに沿ったデザインにする必要があります。

次のトピックでは、**Windows Phone 7** のコントロールの一部について説明し、アプリケーションでこうしたコントロールを使用する方法についてのデザイン ガイダンスを示します。



## プッシュ ボタン

---



ユーザーがプッシュ ボタンをタップすると、操作が開始されます。通常、形は長方形で、標準レイアウトでは、テキストか画像のいずれかを含めることができます。

プッシュ ボタンには以下の特徴があります。

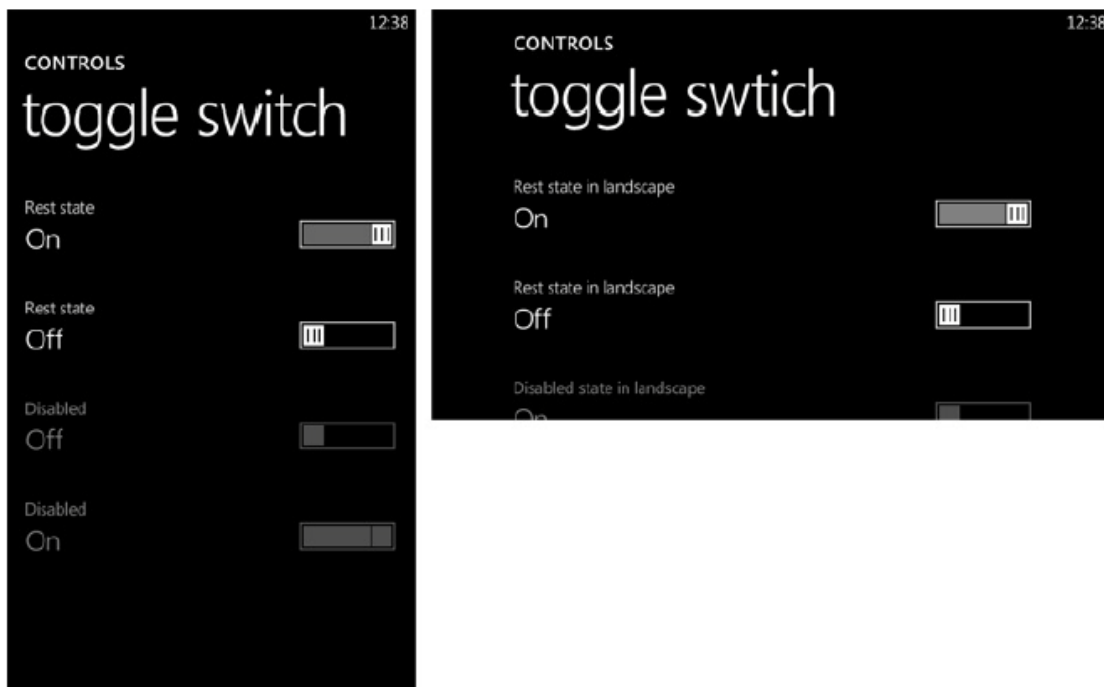
- ✓ プッシュ ボタンでは、通常状態、押下状態、および無効状態があります。
- ✓ フォーカス状態は表示上の変化はありません。

### アプリケーションのデザインに関する考慮事項

- ✓ ジェスチャのサポート: プッシュ ボタンは、タップ ジェスチャで操作することができます。
- ✓ プッシュ ボタンでは、カスタムのスタイル設定とテーマがサポートされます。
- ✓ プッシュ ボタンに含める単語数は 1 つが推奨です。
- ✓ テキストは簡潔にし、通常は動詞（日本語では動名詞）を使用します。

## トグル スイッチ

---



トグル コントロールでは、2 つの選択肢のどちらかが常に選択されており、現在の状態が視覚的にわかるようになっています。2 つの選択肢のうち選択する方を切り替えるには、コントロールをタップします。



### 重要:

このコントロールをアプリケーションに配置する場合、適切にスタイルを設定する必要があります。詳細については、「[Windows Phone Development](#)」(英語) の「Base Controls for Windows Phone」というトピックを参照してください。

次にいくつかの特徴を挙げます。

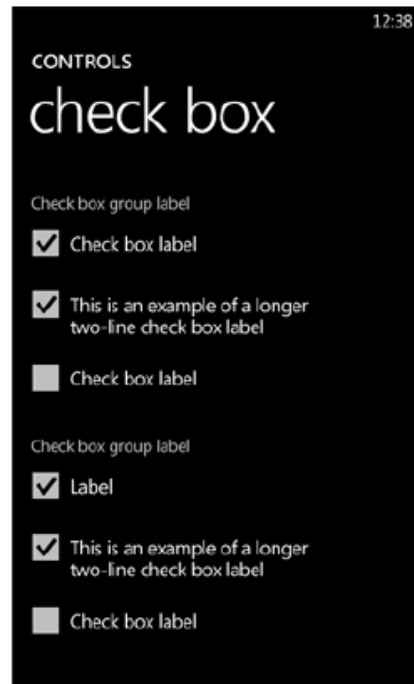
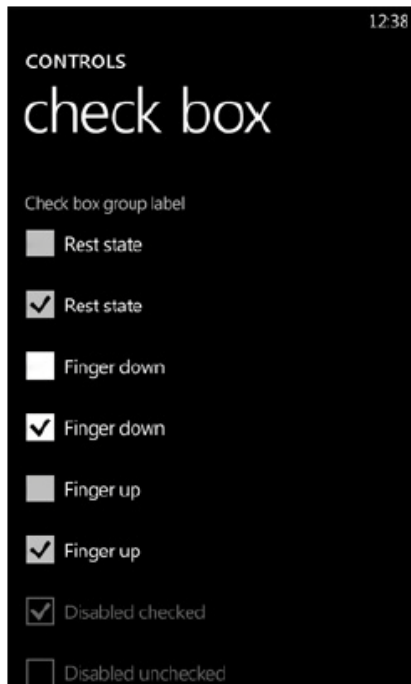
- ✓ トグル スイッチでは、オンとオフの設定の両方に対して、通常状態と無効状態がサポートされます。
- ✓ トグル スイッチでは、テーマがサポートされます。
- ✓ このコントロールでは、フォーカス状態は表示上の変化はありません。

### アプリケーションのデザインに関する考慮事項

- ✓ トグル スイッチでのジェスチャのサポート: トグル スイッチは、タップ ジェスチャによって切り替えることができます。
- ✓ トグル スイッチは、よく検討して使用したほうがよいでしょう。

## チェック ボックス

---



チェック ボックス コントロールは、一覧から項目を選択するのに使用します。チェック ボックスは、複数の選択肢を示し、そこから 1 つ以上の選択肢をユーザーに選んでもらうために、グループで使用することができます。選択肢を選ぶには、ボックスか、添えられているテキストをタップします。また、このコントロールは複数行をサポートすることも可能ですが、デザインの観点から、テキストは 1 ～ 2 行にすることをお勧めします。

次にいくつかの特徴を挙げます。

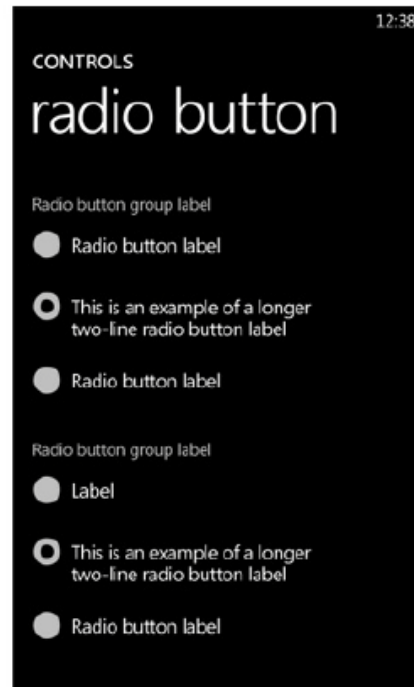
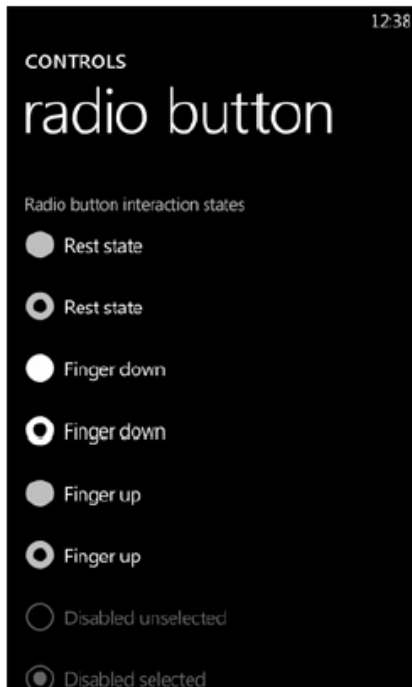
- ✓ チェック ボックス コントロールでは、選択された設定と選択されていない設定の両方に対して、通常状態、押下状態、および無効状態がサポートされます。
- ✓ フォーカス状態は表示上の変化はありません。
- ✓ チェック ボックス コントロールでは、テーマがサポートされます。
- ✓ このコントロールでは、中間状態もサポートされています。

### アプリケーションのデザインに関する考慮事項

- ✓ 選択肢がいくつかある場合は、スクロール ビューアーを使用し、スタック パネルを追加することをお勧めします。
- ✓ ジェスチャのサポート: チェック ボックスは、チェック記号とラベルを一つのタッチ対象として、タップによって操作できます。タップによって、選択された状態と選択されていない状態が切り替わります。
- ✓ このコントロールでは、中間状態もサポートされています。この状態は、チェック ボックスのオンとオフの状態が同時に存在していることを集約して通知するために使用できます。しかし、対象となる項目のどれがオンになっていてどれがオフになっているのかわからないため、ユーザーが混乱するおそれがあるので、この状態を使用することはお勧めしません。より適切な代替案としては、そのチェック ボックスのデータ ソースを別々の複数のチェック ボックスに割り当てるか、複数選択リストを使用します。動的なデータ セットを使用する場合は、後者の方が適切な可能性があります。
- ✓ ラベル付けなど、より高度なレイアウトが必要な場合は、チェック ボックス コントロールを使用しないことをお勧めします。

## オプション ボタン

---



オプション ボタンは、関連する複数の項目の中から 1 つだけを選択するためのコントロールです。オプション ボタンのテキストをタップすると、コントロールを選択できます。一度に選択できるオプションは 1 つだけです。

次にいくつかの特徴を挙げます。

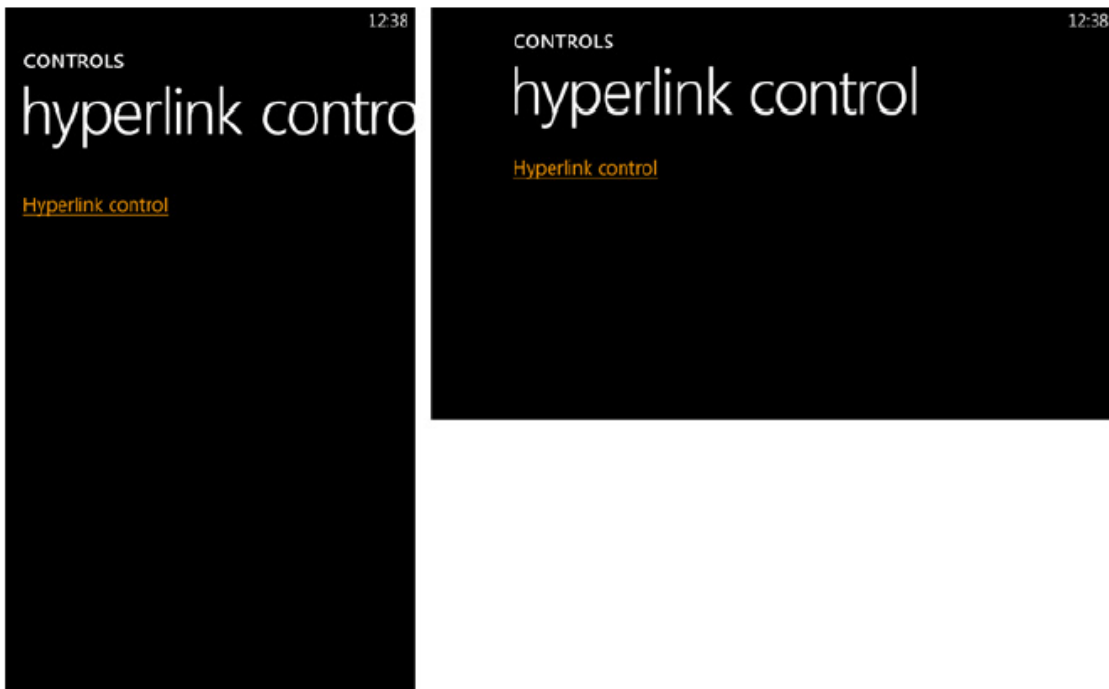
- ✓ オプション ボタン コントロールは、選択された設定と選択されていない設定の両方に対して、通常状態、押下状態、および無効状態を実装します。
- ✓ オプション ボタン コントロールでは、テーマがサポートされます。
- ✓ フォーカス状態は表示上の変化はありません。

#### アプリケーションのデザインに関する考慮事項

- ✓ 選択肢がいくつかある場合は、リスト ボックスとリスト ビュー項目を使用することをお勧めします。
- ✓ オプション ボタンでは複数行の形式がサポートされているので、オプション ボタンのテキストは、折り返されて 2 行目に続くことがあります。ですが、デザインの観点から、1 ～ 2 行の形式を使用することをお勧めします。また、アプリケーションでは、追加のテキストのために十分なスペースを提供する必要があります。
- ✓ このコントロールでは、中間状態もサポートされています。この状態は、チェック ボックスのオンとオフの状態を、階層下にあるいくつかの項目の状態をまとめて通知するために使用できます。しかし、階層下にあるチェック ボックスのどれがオンになっていてどれがオフになっているのかについてはユーザーはわからないため、あまりこの状態を使用することはお勧めしません。より適切な代替案としては、こういったチェック ボックスを別々の区分に変更するか、もしくは複数項目選択可能なリストを使用します。動的なデータ セットを使用する場合は、後者の方が適切な可能性があります。
- ✓ ジェスチャのサポート：オプション ボタンは、タップ ジェスチャによって操作でき、記号かラベルを一つのタッチ対象とします。タップによって、選択された状態と選択されていない状態が切り替わります。

## ハイパーリンク コントロール

---



ハイパーリンク コントロールを使用すると、ページにハイパーテキスト リンクを埋め込むことができます。このコントロールは、ナビゲーション目的のみに使用する必要があります。

次にいくつかの特徴を挙げます。

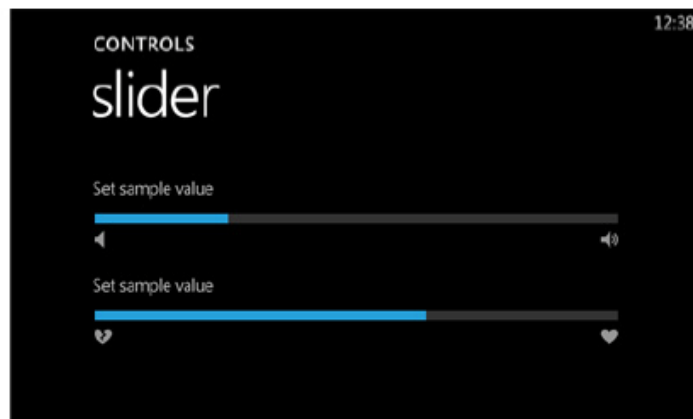
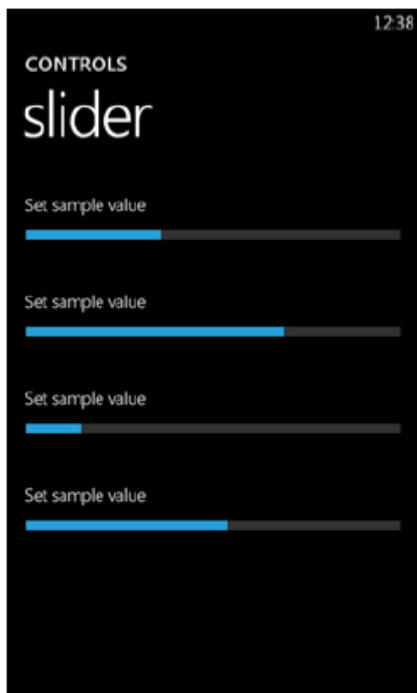
- ✓ このコントロールでは、通常状態と押下状態がサポートされます。
- ✓ 移動先を指定できます。
- ✓ フォーカス状態は表示上の変化はありません。
- ✓ テーマがサポートされます。

### アプリケーションのデザインに関する考慮事項

- ✓ ユーザーが個々のリンクを選択しにくくなるため、多数のリンク コントロールを非常に近くに配置するのは避けてください。
- ✓ ユーザーが追加のテキストを表示したり非表示にしたりできるようにする場合、アプリケーションでは、リンク コントロールの代わりにプッシュ ボタン コントロールを使用する必要があります。たとえば、"詳細情報..." や "詳細" などのテキストを使用する場合はこれに該当します。
- ✓ リンク コントロールでは、状態が一時的である場合（他のシステム プロセスが発生しているなど）、または、ユーザー操作によって状態を有効にできる場合にのみ、無効状態を使用する必要があります。使用できないリンクや、ユーザー操作によって有効にできないリンクは、表示しないようにします。

## スライダー コントロール

---



スライダー コントロールは、連続するデータ範囲の値を設定するのに使用します。たとえば、音量や明るさのレベルを設定する場合に使用します。スライダーには、最小インクリメント値と最大インクリメント値があります。必要に応じて、スライダーのトラックが押されたときに、進行状況のインジケータを表示することもできます。



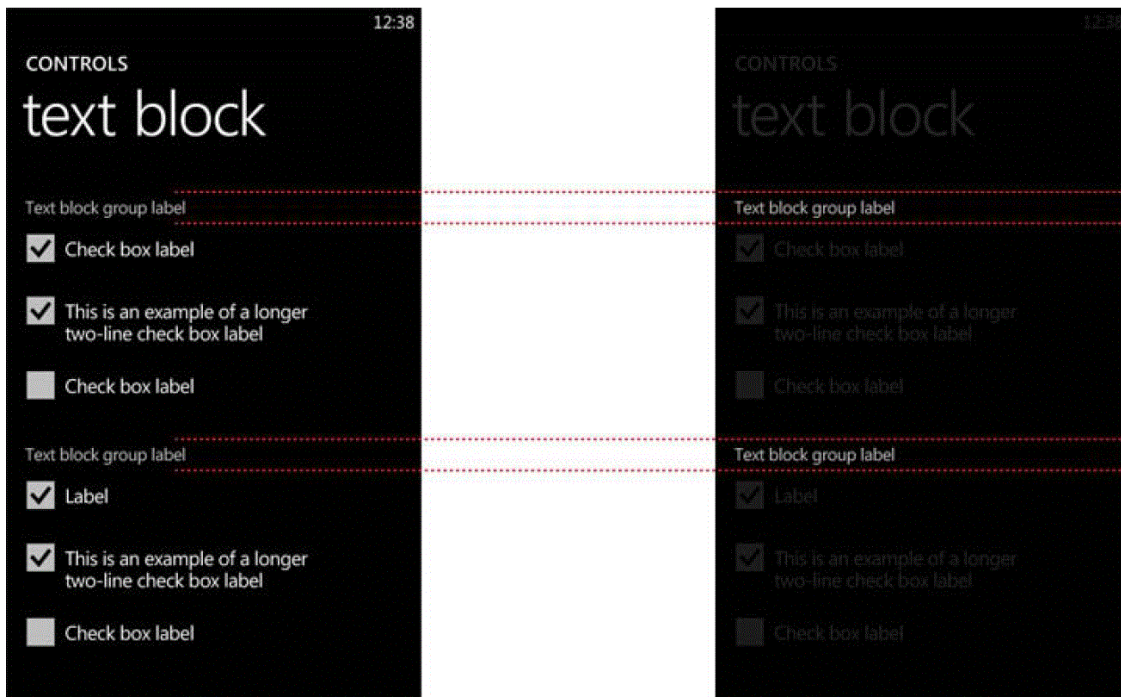
注:

#### アプリケーションのデザインに関する考慮事項

- ✓ アプリケーションでは、スライダー コントロールを水平方向に配置して使用することも垂直方向に配置して使用することもできます。ただし、推奨されるのは水平方向のレイアウトです。

## テキスト ブロック

---



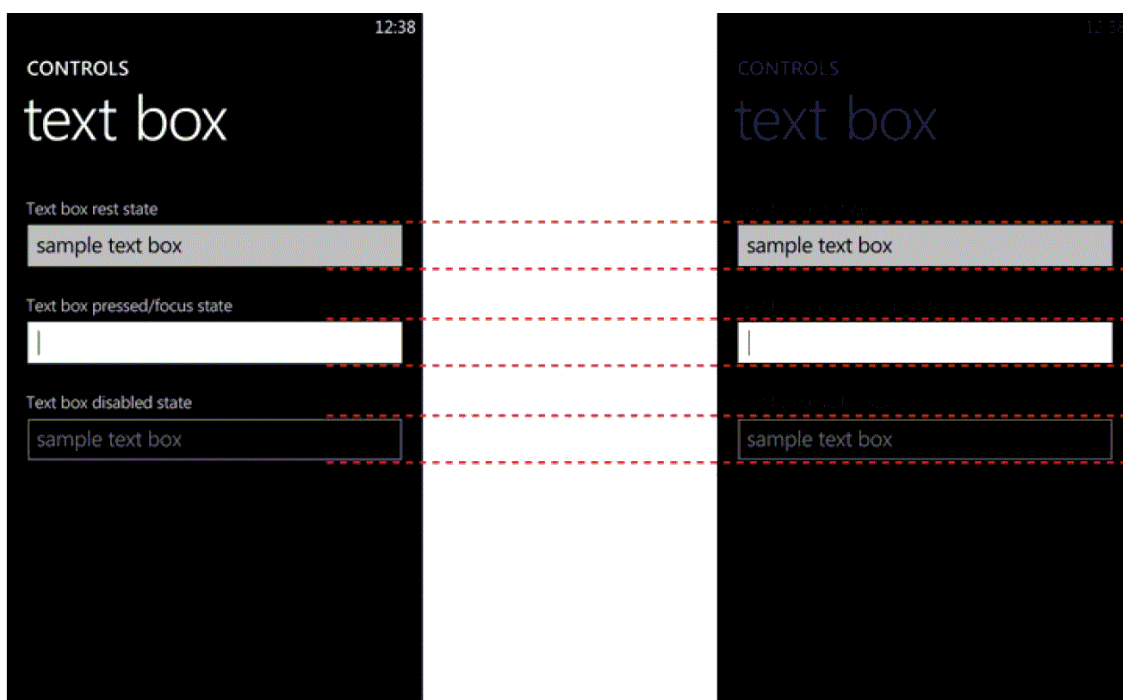
テキスト ブロックを使用すると、一定の量のテキストを表示することができます。テキスト ブロックは、コントロールやコントロール グループにラベルを付けるのに使用されます。テキスト ブロックでは、関連するコントロールの状態にかかわらず、同じ状態が保たれます。このコントロールでは、テーマと右端での折り返しがサポートされます。

## テキスト ボックス

テキスト ボックス コントロールを使用すると、コンテンツの表示、およびユーザーによるコンテンツの入力や編集が可能になります。読み取り専用に設定することもできますが、一般的には、編集可能なテキストに使用されます。テキスト ボックスでは、1 行または複数行を表示することができます。複数行の場合、コントロールのサイズに合わせてテキストが折り返されます。

次にテキスト ボックスのいくつかの特徴を挙げます。

- ✓ テキスト ボックス コントロールでは、編集可能状態と読み取り専用状態がサポートされます。
- ✓ テキスト ボックス コントロールでは、テーマがサポートされます。
- ✓ テキスト ボックス内にフォーカスが設定されると、スクリーン キーボードが自動的に表示されます。



### アプリケーションのデザインに関する考慮事項

- ✓ スクリーン キーボードの入力スコープを追加するには、テキスト ボックス コントロールの入力スコープ プロパティを構成する必要があります。
- ✓ ジェスチャのサポート: タップによって、フォーカスの設定と選択を行うことができます。
- ✓ パスワードが見えないようにするには、パスワード ボックス コントロールを使用します。

## リスト ボックス

---

リスト ボックス コントロールには、データの集合が格納されています。このコントロールにデータを設定するには、データ ソースにバインドするか、バインドされていない項目を表示します。リスト ボックスは、他のコントロールやテキストをデータとして設定することもできる項目コントロールです。たとえば、リスト ビュー アイテム コントロールは、見た目とタッチ対象の異なるコントロールを、その内部に配置することができます。

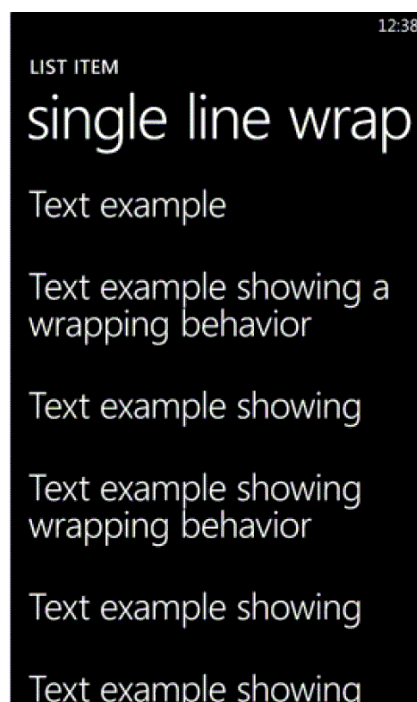
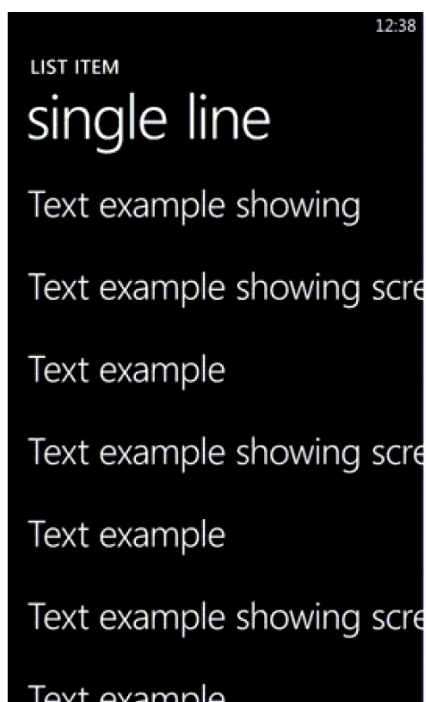
## リスト ビュー項目

---

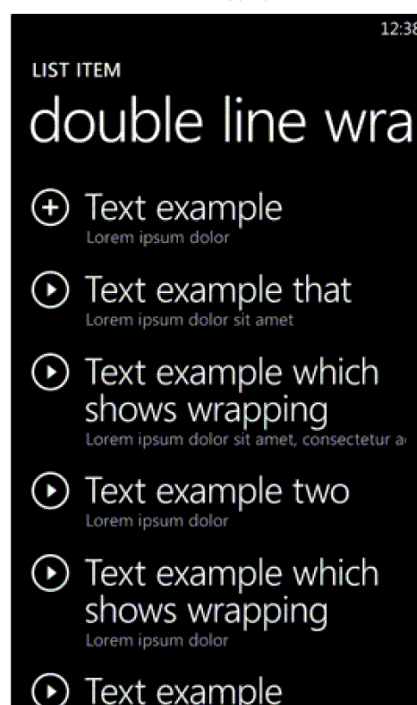
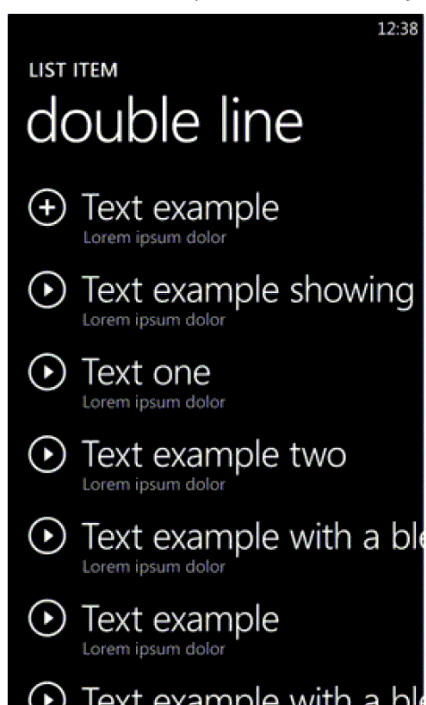
Windows Phone 7 のリスト コントロールを使用すると、開発者は、データ コンテンツを整理された形で表示することができます。リスト ビュー アイテムは、リスト内に表示される、長方形をベースとした視覚要素です。Windows Phone 7 では、いくつかのリスト ビュー アイテム形式を使用できます。

- ✓ 1 行
- ✓ 2 行
- ✓ アイコン付きの 1 行
- ✓ アイコン付きの 2 行
- ✓ 2 次的なタッチ対象を持つ 1 行または 2 行

1 行表示の例を示します。



2 つ目の例として、アイコン付きの 2 行形式を示します。このコントロールでは複数行のテキストがサポートされており、コントロールの境界の右か左に画像のプレースホルダーを生成することができます。

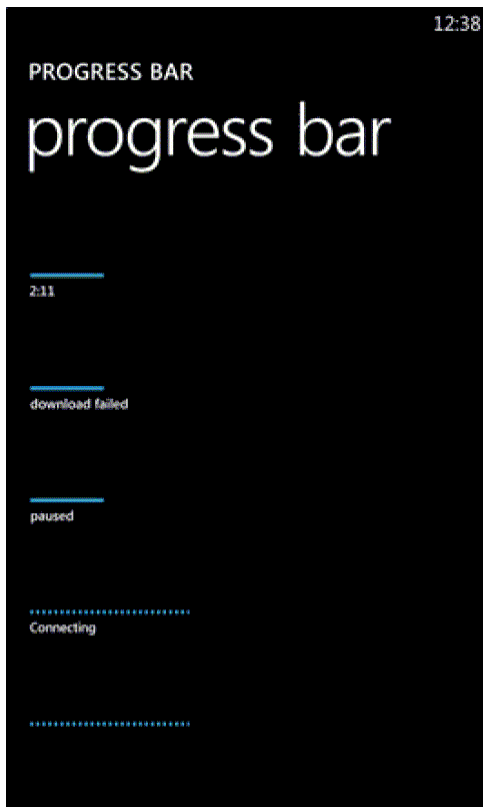


## リスト項目のデザインに関する考慮事項

- ✓ 複数のタッチ対象は、リスト ボックス コントロールを使用する場合のみ利用可能です。
- ✓ 複数のタッチ対象を使用する場合、はじめの操作対象は 2 番目の操作対象よりも大きくすべきです。
- ✓ 初めのタッチ対象はアプリケーションのナビゲーションとして使用され、2 番目のタッチ対象は特定の機能を実行するのに使用される必要があります。

## プログレスバー

---



プログレスバーは、操作の進行状況を示すコントロールです。このコントロールを使用すると、一般的な進行状況や、値に基づいて変化する進行状況を示すことができます。次にいくつかの特徴を挙げます。

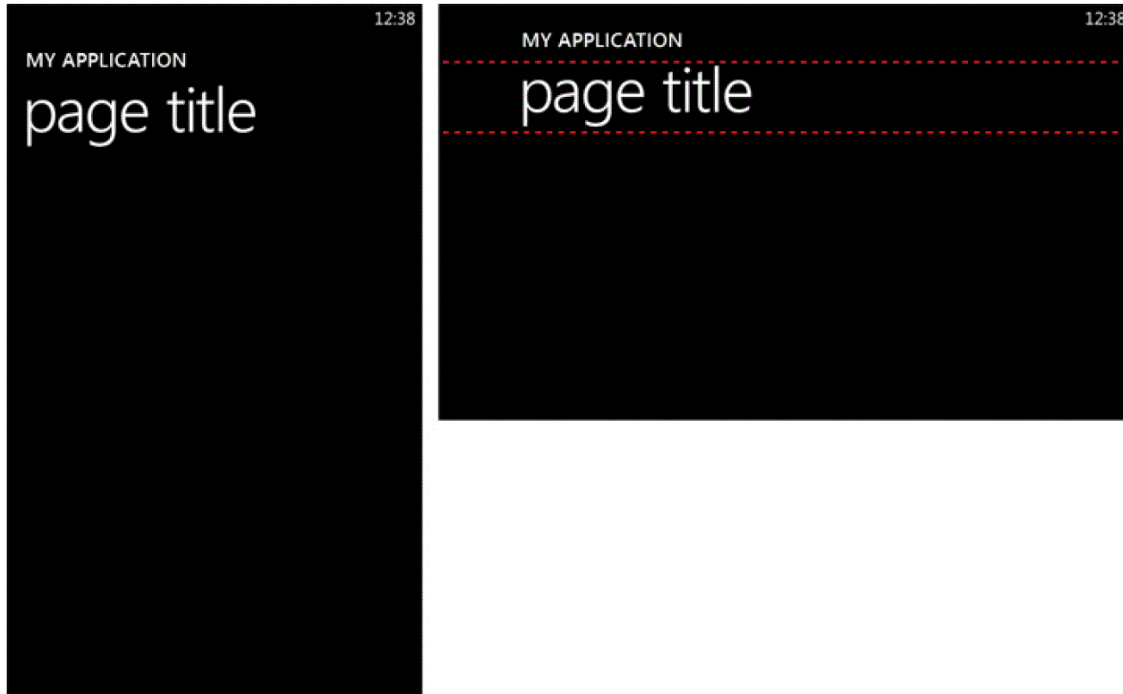
- ✓ プログレスバーでは、マーカー（中間）モードがサポートされます。
- ✓ プログレスバーでは、テーマがサポートされます。

## アプリケーションのデザインに関する考慮事項

- ✓ プログレスバーには、有効状態と無効状態があります。有効になっている場合は、進行を止めるなどの操作を行うことができます。

## ページ タイトル

これは使用可能なコントロールではありませんが、このトピックではページ タイトルのレイアウトを扱います。ページ タイトルは、ページ コンテンツの情報を明確に示すために使用します。

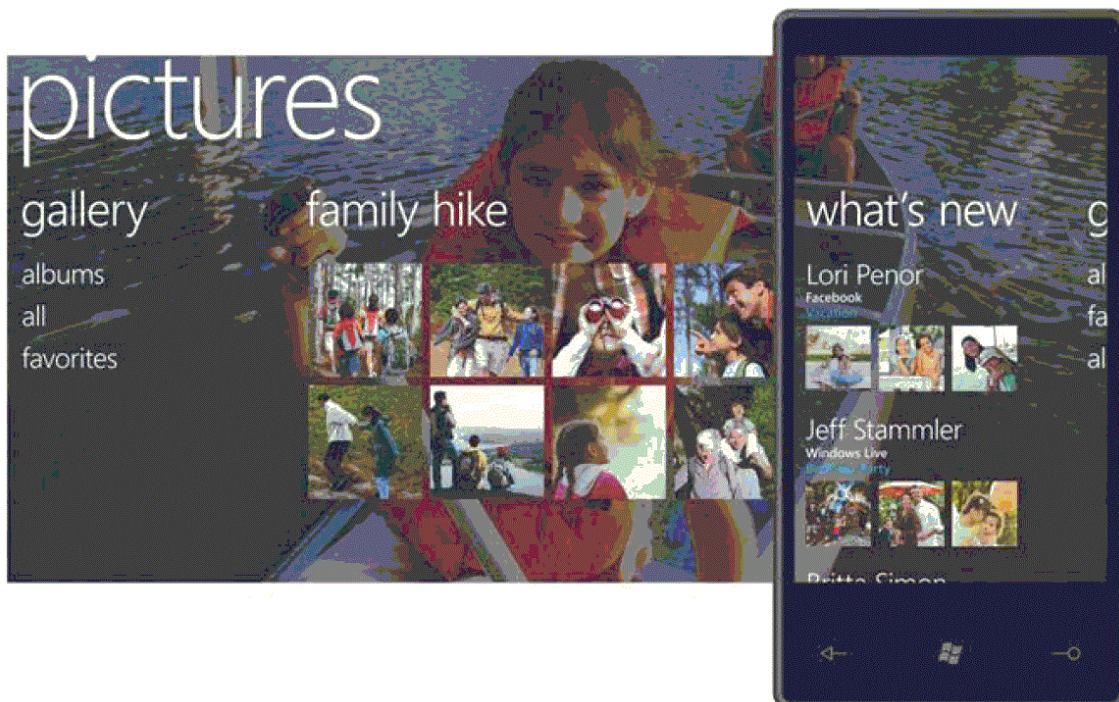


### アプリケーションのデザインに関する考慮事項

- ✓ ページ タイトルは、スクロールしません。
- ✓ アプリケーションでは、ページ タイトルを表示するかどうかを選択できます。アプリケーションで、ページ タイトルを表示することを選択した場合、一貫性を保ち、表示されるウィンドウのサイズを均一にするため、アプリケーションのすべてのページにおいてスペースが確保される必要があります。
- ✓ アプリケーションでページ タイトルを表示する場合、タイトルは、アプリケーションの名前か、表示されるデータに関連する明確で説明的なテキスト行にします。ページ タイトルは、現在使用できません。

## パノラマ アプリケーション

パノラマ アプリケーションは、Windows Phone OS 7.0 のコア エクスペリエンスの一部です。デバイスの画面領域内に収まるようにデザインされている標準的なアプリケーションとは異なり、パノラマ アプリケーションは、コントロール、データ、およびサービスの表示に、画面領域内に収まらない、横に長いキャンバスを使用します。この本質的に動的なアプリケーションでは、レイヤーが視差効果と同じようにさまざまな速度でスムーズにスクロールするように、レイヤー化されたコンテンツとアニメーションが使用されます。



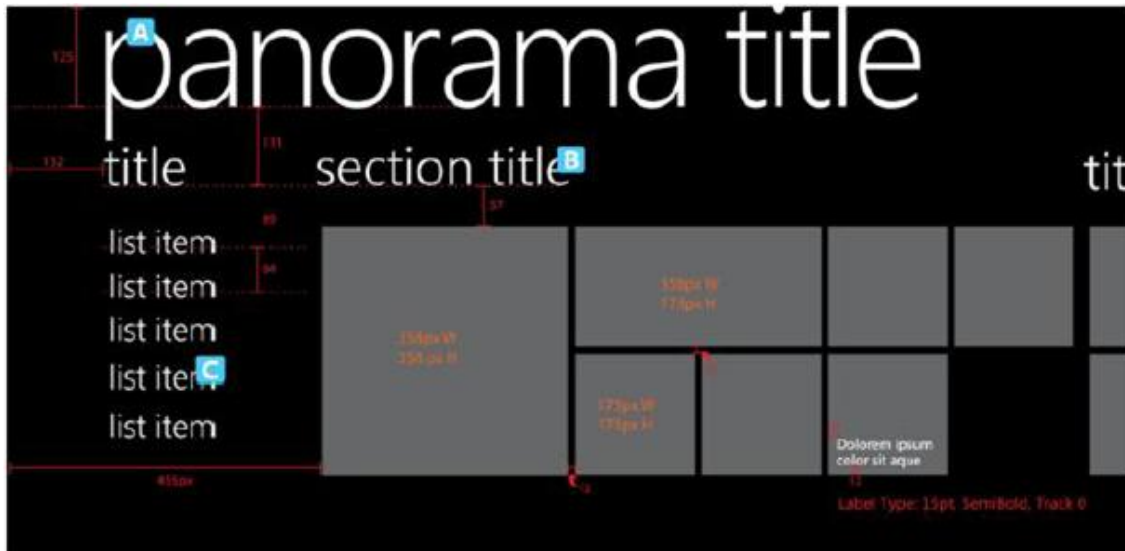
## デザインに関する推奨事項

カスタムの **Silverlight** パノラマ アプリケーションには、公式の制限はありませんが、マイクロソフトからいくつかの推奨事項が提供されています。これらの推奨事項の目的は、開発者が統合されたパノラマ アプリケーションを忠実に再現し、より一貫性のあるエクスペリエンスをデバイス上で提供するのを、サポートすることです。



## 共通の UI 要素

パノラマ アプリケーションのユーザー インターフェイスは、それぞれ独立したモーション ロジックで動作する 4 種類のレイヤー（背景画像、パノラマ タイトル、パノラマ セクション、パノラマ セクション タイトル）で構成されています。

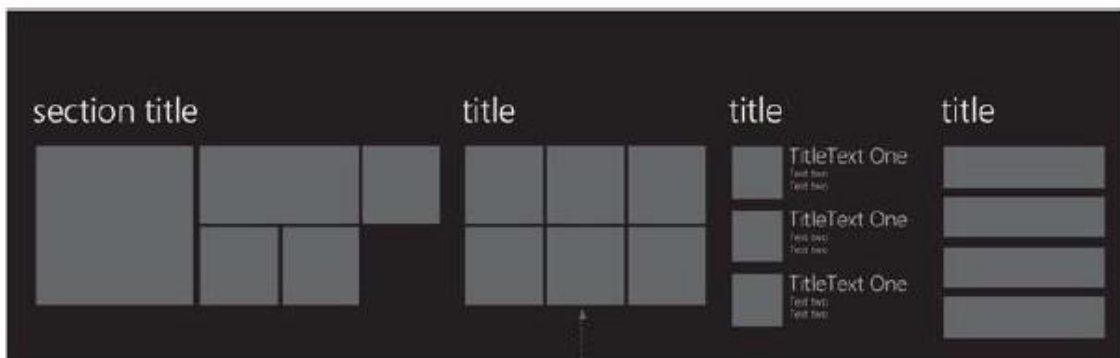


**A** Segoe UI Light  
140 ポイント  
トラッキング -10

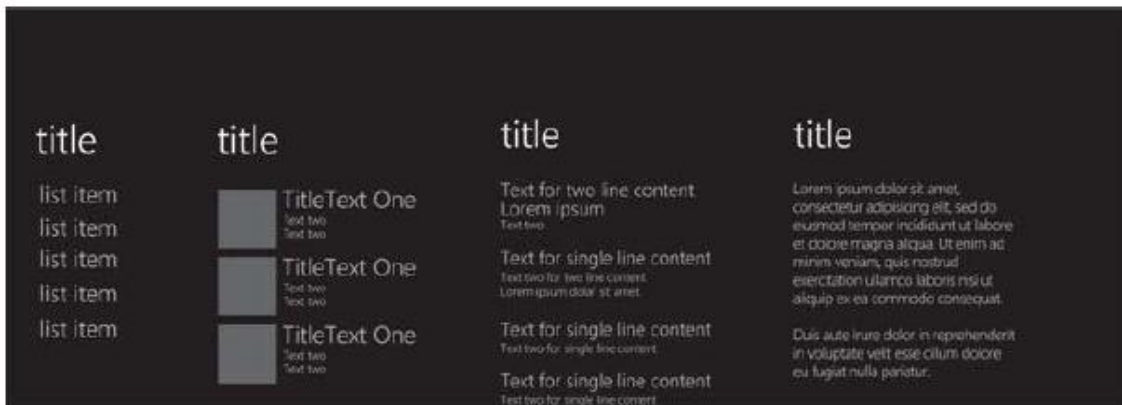
**B** Segoe UI Semilight  
54 ポイント  
トラッキング -10

**C** Segoe UI Semilight  
24 ポイント  
トラッキング -10

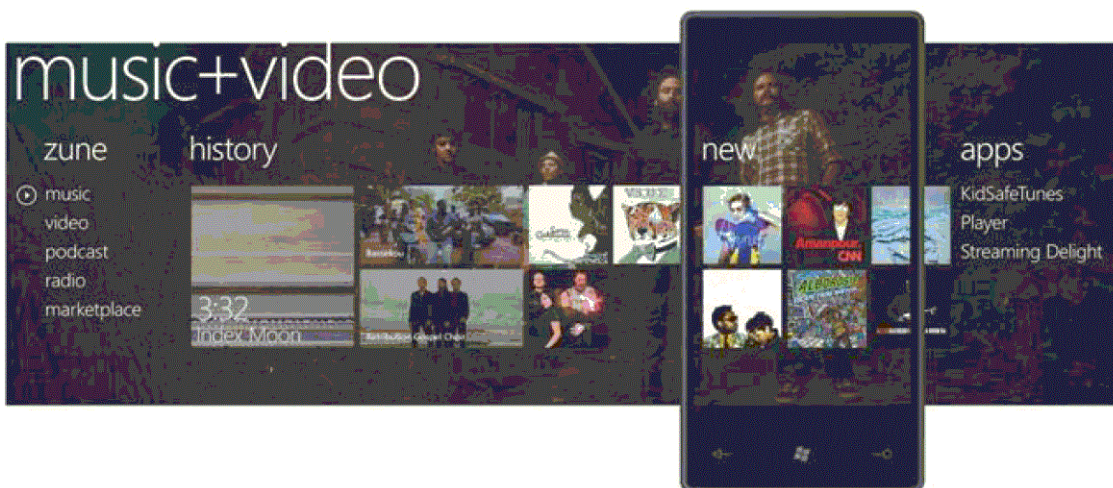
次の画像は、画像ベースの要素を示しています。



次の画像は、テキスト ベースの要素を示しています。



## 背景画像



背景画像はパノラマ アプリケーションの最も下のレイヤーで、パノラマに雑誌のような鮮やかな雰囲気を与えることを目的としています。背景は通常、画面の隅から隅まで表示され、アプリケーションにおいて最も視覚に訴える部分となる可能性があります。良好なエクスペリエンスを実現するためには、次のような多くの考慮事項を念頭に置く必要があります。

- ✓ 単色の背景、または、パノラマ全体に広がる画像を使用します。画像を使用することにした場合、**Silverlight** でサポートされている種類であればどのような種類の **UI** 画像でも使用できますが、サイズ上の配慮から、**JPG** 形式を使用することをお勧めします。
- ✓ 複数の画像を背景として使用してもかまいませんが、表示するのは常に **1** 枚だけにする必要があります。
- ✓ 優れたパフォーマンスを実現し、読み込み時間を最小限に抑え、サイズ変更を行わなくて済むようにするには、サイズが **800 x 480 ~ 800 x 1024** ピクセル (高さ x 幅) の画像を使用します。
- ✓ **4** つのセクションがあるパノラマ アプリケーションには、アスペクト比 **16 x 9** を使用します。
- ✓ テキストを読みやすくするためには、透明な黒か白のフィルターを使用します。
- ✓ 動的な **UI** 要素には、ドロップ シャドウ効果を使用しないようにします。
- ✓ モーションの速度 (背景画像の幅に対する、最上位にあるコンテンツ レイヤーの幅全体の割合によって決まります) は、パンの速度に比例するようにします。
- ✓ アニメーション化されたモーションは、アプリケーションに背景画像がある場合にのみ使用します。

- ✓ ユーザーが画像の幅を超えてパンしたときは、画像の端まで表示してから、表示可能な領域に戻ります。

## パノラマ タイトル



パノラマ タイトルは、パノラマ アプリケーション全体のタイトルです。これは、ユーザーにアプリケーションを識別してもらうためのもので、どのような方法でアプリケーションを起動しても表示される必要があります。パノラマ タイトルに関する推奨事項を以下に示します。

- ✓ パノラマ タイトルには、書式なしテキスト、またはロゴなどの画像を使用します。ロゴとテキスト（または、他の **UIElement**）など、複数の要素を使用してもかまいません。
- ✓ フォントや画像の色は背景全体に対して見やすいものにします。また、タイトルが背景画像に混ざらないようにします。
- ✓ 一貫性を保つため、スタート メニューの起動タイルには同じタイトルを使用します。
- ✓ タイトルをアニメーション化したり、タイトルのサイズを動的に変更したりするのは避けます。
- ✓ モーションの速度は、最上位のコンテンツ レイヤーよりは遅く、背景画像よりは速くします。
- ✓ ユーザーが画像の幅を超えてパンしたときは、画像の端まで表示してから、表示可能な領域に戻ります。

## パノラマ セクション



パノラマ セクションは、他のコントロールとコンテンツをカプセル化する、パノラマ アプリケーションのコンポーネントです。パノラマ セクションに関する推奨事項を以下に示します。

- ✓ 最大 4 つのセクションを使用して、パノラマ アプリケーションがスムーズに実行されるようにします。
- ✓ リストやグリッドの垂直スクロールは、セクションの範囲内であり、また、水平スクロールと同時に行われない限り、問題ありません。
- ✓ 垂直スクロールは、セクションの幅が画面の幅より狭い限り問題ありません。
- ✓ カスタム コントロールと標準的なコントロールをすべてサポートします。
- ✓ 指によるドラッグと同じ速度で動くようにします。
- ✓ ユーザーが新しいセクションに移動すると、画面の外に出ていくようにします。
- ✓ セクションのレイアウトは、次のセクションの兆しが見えるようにデザインします。わずかに重なりを持たせることで、アプリケーションをパンする必要があるということをユーザーが直感的に理解できます。
- ✓ 表示するコンテンツがない間はパノラマ セクションは隠しておくことを検討してください。

## パノラマ セクションのタイトル

パノラマ セクションでは、タイトルは省略可能です。タイトルを使用する場合は、次の推奨事項について考慮します。

- ✓ 書式なしテキストを使用するのが理想的ですが、画像も使用できます。画像とテキスト（または、他の UIElement）など、複数の要素を使用してもかまいません。
- ✓ セクション タイトルが、背景画像と混ざらないようにします。
- ✓ タイトルは動くため、アニメーション化されたタイトルは避けます。
- ✓ コントロールが複数存在しても、パノラマ セクションのタイトルはセクション全体に及ぶようにします。
- ✓ ユーザーが新しいセクションに移動すると、画面の外に出ていくようにします。



注:

パノラマ セクションのタイトルは、セクションの幅が画面の幅より広い狭いかに応じて、異なる動きをする必要があります。セクションの幅の方が広い場合は、水平方向のアニメーション化を行う必要があります。つまり、タイトルは、セクションの左上にとどまるのではなく、パノラマが動いている間、異なる速度で上部を移動する必要があるということです。このような状況では、垂直スクロールは行われるべきではありません。一方、セクションの幅が画面の幅よりも狭い場合は、タイトルは常にセクションの左上に固定されている必要があります。このような状況では、水平方向のアニメーション化を行うべきではなく、垂直スクロールが可能な場合は、タイトルはコンテンツと共に移動する必要があります。

## サムネイルの指定

サムネイルは、パノラマ ビューの主要な要素で、パノラマ エクスペリエンスの外で使用されるコンテンツやメディアにリンクします。次の画像に示すように、画像全体ではなく、特定の項目に焦点が当たっているトリミングされた画像を使用する必要があります。テキストなしでは画像が何を表すのかわかりにくい場合、内容を明確にするためにテキストを 2 行まで使用できます。



システム テキストのないサムネイル



フォント: Segoe Semibold、15 ポイント、トラッキング 0



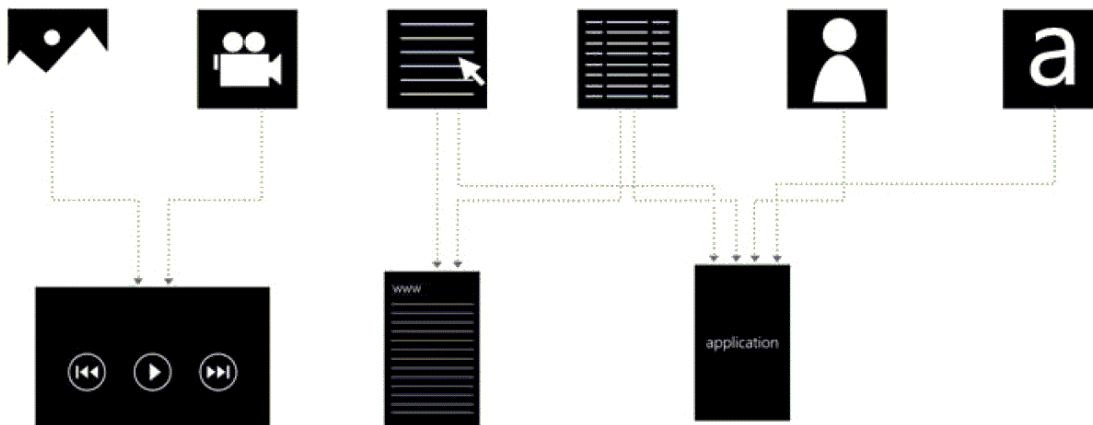
マスクINGの必要がない、テキスト入りのサムネイル



画像に 25% の黒いマスクINGが施された、テキスト入りのサムネイル

## 要素のフロー

パノラマ アプリケーションの要素は、より詳細なエクスペリエンスの出発点となります。次の画像は、主要なパノラマ アプリケーションで使用する標準的なフローを表しています。



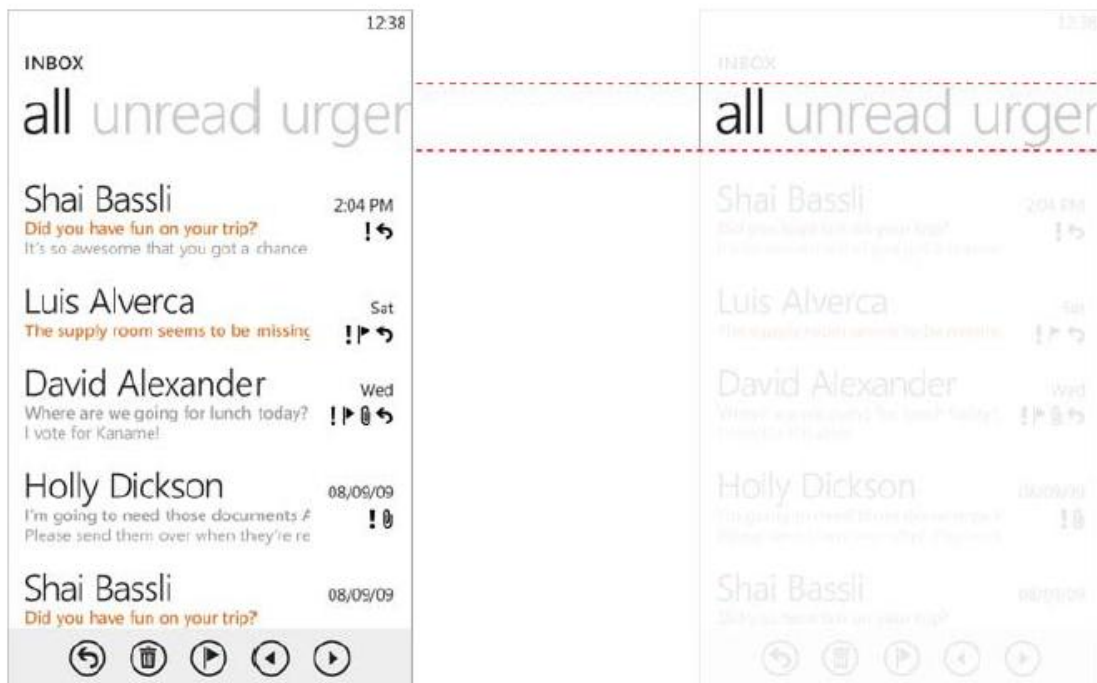
### 注:

上記の要素フローは、プラットフォームの機能ではなく、エンド ユーザー エクスペリエンスを示しています。たとえば、エンド ユーザーの目には、パノラマ アプリケーションから別のアプリケーションが起動されているように見えるかもしれませんが、起動されたアプリケーションは、実際には同じパノラマ アプリケーションが異なって見えているだけのものです。

## ピボット コントロール

---

ピボット コントロールを使用すると、アプリケーション内のビューやページをすばやく管理できます。ピボットは、大きなデータ セットのフィルター処理、複数のデータ セットの表示、およびアプリケーション ビューの切り替えに使用できます。ピボット コントロールでは、個々のビューが横に並べて配置され、左右のナビゲーションが管理されます。ページ上で左から右にフリックまたはパンすると、次のページのコンテンツに進みます。



## アプリケーションのデザインに関する考慮事項

- ✓ アプリケーションでは、ピボット ページの数を最小限にする必要があります。
- ✓ ピボット ページのコンテンツはアプリケーションによって定義されます。
- ✓ ピボット ページは循環式です。
- ✓ ピボット ページでは水平方向のパンと水平方向のフリックの機能を変更するべきではありません（ピボット コントロールの操作デザインと相反するため）。
- ✓ ピボット ヘッダーとテキストの長さに制限はありません。表示されるテキストの量は、ピボット コントロールの幅によって制限されます。
- ✓ ピボット ヘッダーの高さは固定されており、変更できません。
- ✓ ピボット コントロールは、同じような種類の項目やデータを表示するためにのみ使用する必要があります。
- ✓ このコントロールは、タスク フローに使用するべきではありません。異なるページに移っても、外観と雰囲気面でシームレスである必要があります。また、ページがユーザーのアクティビティを劇的に変化させることは避ける必要があります。
- ✓ ピボット コントロールは、慎重に使用し、エクスペリエンスに適したシナリオにのみ使用する必要があります。
- ✓ 空のピボット ページは、追加の情報がユーザーの操作によって追加される可能性がない場合にのみ削除するようにします。

## Windows Phone システム コントロール

このセクションでは、Windows Phone 7 のシステム コントロールを取り上げます。ここでは、こうしたコントロールの使い方や、こうしたコントロールがアプリケーションとどのように影響しあうかについて詳しく説明します。

- ✓ [システム トレイとアプリケーション バー](#)
- ✓ [コンテキスト メニュー](#)
- ✓ [Windows Phone の通知](#)
- ✓ [進行状況のインジケーター](#)

## システム トレイとアプリケーション バー

Windows Phone 7 のクロムには、システム トレイとアプリケーション バーという 2 つの主要なコンポーネントが含まれています。

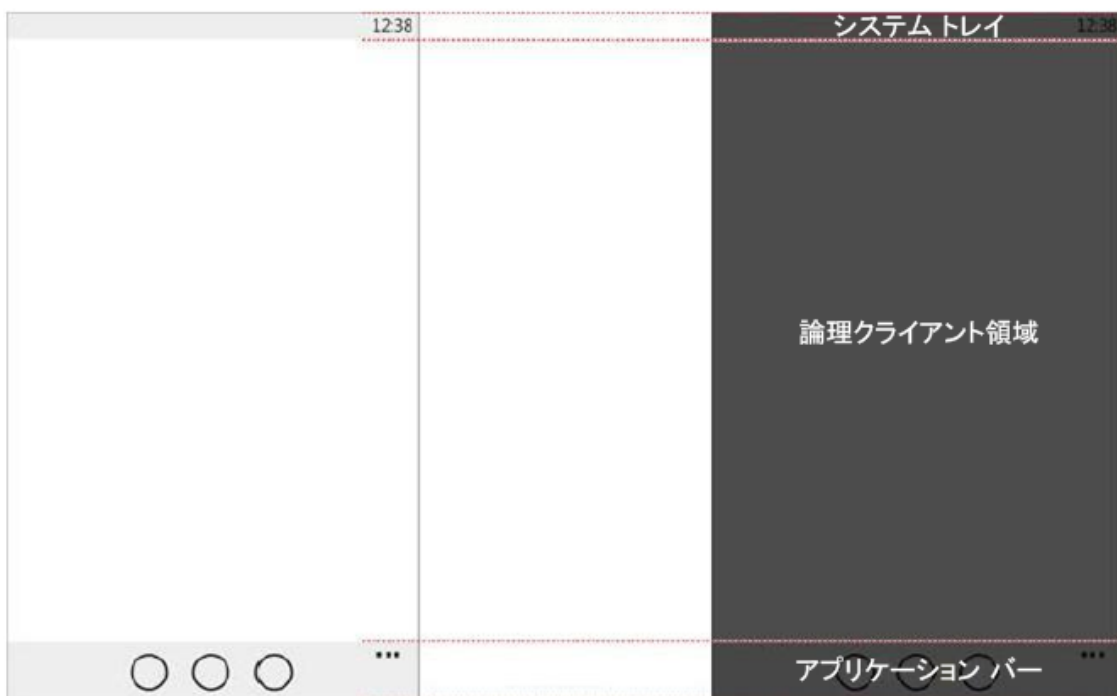
- ✓ **システム トレイ:** アプリケーション ワークスペース内に用意された、システム レベルの状態情報を簡単かつわかりやすく表示するインジケータ バーです。システム トレイが、さまざまな通知を提供しユーザーが常に重要な情報を認識しておくことができるようにするために更新されると、ユーザーは必要に応じてシステム トレイをタッチします。
- ✓ **アプリケーション バー:** アプリケーション バーは、最も一般的なタスクとビューを表示するための場所を提供します。また、アプリケーション バーには、アプリケーション メニューを 1 つ必要に応じて表示することができます。一般的なタスクとビューは、ツール バーにグラフィカルなボタンとして表示され、その他のタスクはメニューにまとめられます。次の図では、システム トレイとアプリケーション バーがアプリケーションの装飾となっています。



### 注:

システム トレイとアプリケーション バーの表示/非表示を切り替えるには、**FullScreen** プロパティを使用します。アプリケーション バーの表示/非表示を切り替え、不透明度を変更するには、アプリケーション バーの **Visible** プロパティと **Opacity** プロパティをそれぞれ設定します。

次の画像は、アプリケーション内でのこれらの要素の配置を示しています。



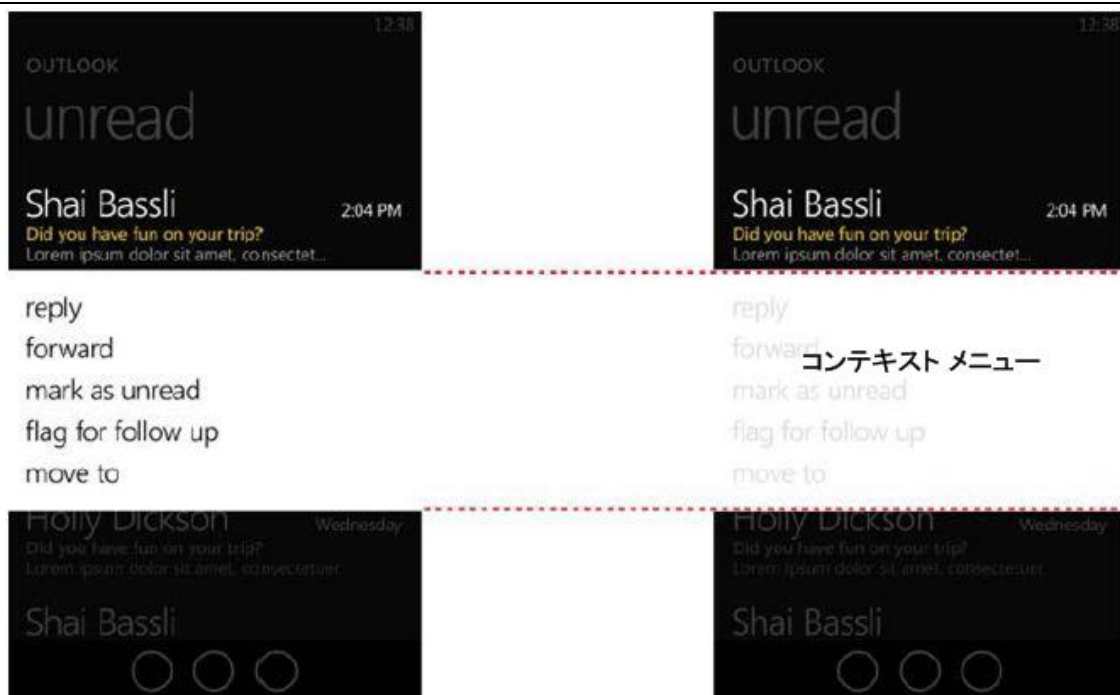
アプリケーション バーの特徴には、次のようなものがあります。

- ✓ アプリケーション バーはディスプレイの下端に沿って配置され、幅は画面の左端から右端まで及びます。
- ✓ アプリケーション バーの高さは固定されており、変更できません。
- ✓ アプリケーション バーにボタンを追加すると、中央揃えで左から右に配置されていきます。
- ✓ アプリケーション バーのメニューに項目が存在する場合、それを視覚的に示す **3** つのドットがアプリケーション バーに表示されます。これらのドットをタップすると、アプリケーション バーのメニューが表示されます。
- ✓ アプリケーション バーのボタンは無効にすることができます。たとえば、アプリケーション バーに削除ボタンを用意するとします。なんらかの項目を削除するのが適切ではないシナリオでは、このボタンを無効にすることができます。たとえば、項目が読み取り専用の要素である場合などです。
- ✓ ユーザーは、メニューの外をタップすると、アプリケーション メニューを閉じることができます。
- ✓ アプリケーション バーに配置できる項目は最大 **4** つです。頻繁に行わないタスクは、アプリケーション メニューに配置する必要があります。この処理は、自動では行われません。

### アプリケーションのデザインに関する考慮事項

- ✓ どうしても色をカスタマイズする必要がある場合以外は、アプリケーション バーには、システムの既定のテーマ色を使用します。アプリケーション バーにカスタムの色を使用すると、ボタン アイコンの画質に影響したり、メニュー アニメーションで意図しない視覚効果が発生したりします。また、ディスプレイの種類によっては、電力消費量に影響が出る場合もあります。
- ✓ アプリケーション バーの不透明度は細かく調整できますが、不透明度の値には、**0**、**0.5**、または **1** のみを使用することをお勧めします。アプリケーション バーの不透明度が **1** 未満の場合、アプリケーション バーは **UI** に重なって表示されます。不透明度が **1** に設定されている場合、表示されるページ サイズが変わります。
- ✓ ジェスチャのサポート：アプリケーション バーをフリックまたはタップすると、アプリケーション メニューが表示されます。
- ✓ アプリケーション バーのアイコン画像は、**48 x 48** ピクセルにする必要があります。それ以外のサイズの画像は適切なサイズに調整されますが、画質が悪くなります。
- ✓ アイコン画像では、アルファ チャネルを使用して、透明な背景に白い前景を使用する必要があります。アプリケーション バーでは、現在のスタイル設定に基づいてアイコンに色が付けられるので、色付きのアイコンを使用すると、アプリケーション バーによる色付け処理の結果が予期せぬ見た目になる可能性があります。
- ✓ 各アイコン ボタンに表示される円は、アプリケーション バーによって描画されるものなので、元の画像には含めないようにします。
- ✓ アイコン ボタンは、アプリケーション ビューで最もよく使用される主要な操作に使用します。操作の中には、アイコンでわかりやすく表すのが難しいものもあります。その場合は、アイコンの代わりにメニュー項目を使用することを検討します。必要でない場合は、**4** つのスロットすべては埋めないようにします。
- ✓ メニュー項目のテキストが長すぎると、画面からはみ出てしまいます。画面に収まる文字数は決まっていますが、文字数は可変のため、テキストは **14** ～ **20** 文字にすることをお勧めします。これが、メニュー項目のテキストについての推奨される最大範囲です。

## コンテキスト メニュー



コンテキスト メニューも、ユーザーが特定のタスクにアクセスする方法の 1 つです。スクロールをしなくても済むように、コンテキスト メニューに配置できる項目数は最大 5 個となっています。



注:

このセクションでは、システム コントロール版のコンテキスト メニューを示していますが、現在のリリースでは、開発者がアプリケーションでカスタムのコンテキスト メニューを独自に作成する必要があります。次のセクションでは、この種のメニューを作成する際に役立つ可能性のある特徴とガイダンスを紹介します。

次にいくつかの特徴を挙げます。

- ✓ コンテキスト メニューは、ユーザーがメニューを閉じるまで画面に表示され続けます。メニューを閉じるには、メニューの外をタップする、ハードウェアの戻るボタンを押す、メニュー項目を選択する方法もあります。システム イベントによってコンテキスト メニューが閉じられることもあります。
- ✓ ジェスチャのサポート: リスト項目をタップしたまま押さえると、コンテキスト メニューを表示できます。



注:

あるタスクを実行する方法がコンテキスト メニューのみにはならないようにしてください。コンテキスト メニューはあくまでも、目的とする操作にたどり着くためのプロセスをスピードアップする方法の 1 つです。

## Windows Phone の通知

Windows Phone 7 には、デバイスで発生したイベントについてユーザーに通知する方法がいくつかあります。通知の種類は次のとおりです。

- ✓ 認識通知では、ユーザー ワークフローを中断することなく、発生した可能性のある変更やイベントをユーザーに通知します。このような通知は、スタート メニューに "タイル" という視覚的な情報として表示されます。
- ✓ 操作要求通知は、システム全体にわたる通知で、ユーザー ワークフローを中断せず、解決するのに介入は必要ありません。この通知は、たとえば、エンド ユーザーがテキスト メッセージやインスタント メッセージを受信したときなどに行われます。
- ✓ アプリケーションで処理が必要なことを示す通知は、アプリケーションによって完全に制御され、そのアプリケーションのみに影響を及ぼします。

### プッシュ通知の概要

アプリケーション開発用に、プッシュ通知サービスは、モバイル デバイスに通知をプッシュするための、回復力と持続性を備えた専用のチャネルをクラウド サービスに提供するようにデザインされています。クラウド サービスがデバイスにプッシュ通知を送信する必要がある場合、クラウド サービスは、プッシュ通知サービスに通知要求を送信します。その後、プッシュ通知サービスによって通知はトースト通知またはタイル通知としてアプリケーションかデバイスに転送されます。トースト通知とタイル通知のどちらの形になるかは、送信される通知の種類によって決まります。デバイスのプッシュ クライアントは、通知チャネルを通じてプッシュ通知を受け取ります。チャネルが作成されると、サブスクリプションも作成され、クラウド サービスがそのチャネルに通知をプッシュできるようになります。チャネルは、サブスクリプションに関連するすべての情報を含む URL によって表されます。アプリケーションは、プッシュ通知を受け取ると、クラウド サービスのプロトコルを使用してクラウド サービスにアクセスし、必要な情報をすべて取得することができます。

## プッシュ通知の種類

タイル通知とトースト通知はどちらも、クラウド サービスが、関連する実用的なフィードバックをアプリケーション自体のユーザー インターフェイスの外でユーザーに提供できるようにするメカニズムです。さらに、クラウド サービスは、未処理の通知要求を送信することもできます。送信された通知は、種類に基づいてアプリケーションかシェルに転送されます。

### タイル通知

タイルは、アプリケーションやそのコンテンツが視覚的かつ動的に表現されたもので、デバイスのスタート画面のクイック起動領域内に表示されます。たとえば、天気アプリケーションでは、特定の地域の時間と気象条件をタイル内に表示することができます。クラウド サービスはいつでもタイルの外観を変更できるので、このメカニズムを使用して情報を継続的にユーザーに伝えることができます。ユーザーがデバイス上で起動できるアプリケーションそれぞれが 1 つのタイルと関連付けられていますが、どのタイルをクイック起動領域に固定するかを制御できるのはユーザーだけです。現時点では、あるアプリケーションのタイルがクイック起動領域に固定されているかどうかをそのアプリケーションが判断することはできません。

クラウド サービスは、タイルの背景画像、カウンター ("バッジ")、およびタイトルのプロパティを制御できます。これらのプロパティは、**Windows Phone Developer Tools** を使用して構成できます。アニメーションとサウンドのプロパティは、アプリケーションによってではなく、プラットフォームがどのように構成されているかによって制御されます。たとえば、プラットフォームが、どのタイルの更新時にもアニメーション化を行い、警告音を鳴らすように構成されている場合、どのタイルの更新時にもこのような動作が行われます。

タイルの背景画像は、アプリケーションの展開の一部であるローカル リソースを参照することも、クラウド リソースを参照することもできます。クラウド内のリソースを参照することで、アプリケーションは、タイルの背景画像を動的に更新できるようになります。これにより、表示前に背景画像を処理する必要があるシナリオが実現可能になります。ほとんどのシナリオでは、タイルに必要な背景画像がすべてアプリケーション パッケージに含まれるのが望ましい方法です。パフォーマンスとバッテリー寿命にとってはこれが最善の方法だからです。

次の画像は、タイルの構造とタイル画像のサイズを示しています。これらの情報は、タイルの UI を構築する際に必要となります。



注:

タイル画像には、JPG 形式も PNG 形式も使用できます。タイル通知カウンターは、タイルの構築において省略可能なコンポーネントです。タイルを作成したりタイルに変更を加えたりする方法の詳細については、<http://go.microsoft.com/fwlink/?LinkId=185182> (英語) を参照してください。

### トースト通知

クラウド サービスでは、ユーザーの現在の画面にオーバーレイ表示される、トースト通知と呼ばれる特別な種類のプッシュ通知を生成できます。トースト通知は、たとえば、インスタント メッセージング クライアントや、その他のピア ツー ピア指向のアプリケーションを通じて生成されます。ユーザーがトースト通知をクリックすると、アプリケーションが起動して他の操作を実行することができます。クラウド サービスでは、トースト通知のタイトルとサブタイトルを制御できます。また、トースト通知では、アプリケーションの展開パッケージに含まれているアプリケーション アイコンも表示されます。

トースト通知に関する注意点を以下に示します。

- ✓ トースト通知は、個人に関連したもので、かつ、タイム クリティカルにする必要があります。
- ✓ 主にピア ツー ピア通信に特化する必要があります。
- ✓ アプリケーションでは、既定ではトースト通知がオフになっている必要があります。トースト通知を有効にするためには、ユーザーがアプリケーション UI や設定メニューで、有効にすることを明示的に選択する必要があります。

## 未処理の通知

アプリケーションでは、プッシュ通知を使用せずにクラウド サービスからメッセージを受信する必要がある場合、(長い HTTP ポーリングを使用して) 絶えずクラウド サービスのポーリングを行う必要がありました。プッシュ通知サービスを使用すると、アプリケーションでは、フォアグラウンドで実行されているときにクラウド サービスから未処理の通知を受信することができます。これで、アプリケーションでは、バック エンドのクラウド サービスへの固定接続を開いたままにしておくためのカスタム コードを記述する必要がなくなりました。アプリケーションがフォアグラウンドで実行されていない場合は、未処理の通知はプッシュ通知サービスで破棄され、デバイスには送信されません。

## 進行状況のインジケータ

進行状況のインジケータでは、1 つのアクティビティまたは複数のイベントに関連する、アプリケーション内のアクティビティが示されます。このシステム コントロールは、システム トレイに統合され、複数のアプリケーション ページにわたって表示できます。

進行状況のインジケータには、進行状況がわかるものとわからないものがあります。

- ✓ 進行状況がわかるインジケータには、開始位置と終了位置があります。
- ✓ 進行状況がわからないインジケータは、タスクが終了するまで表示され続けます。



## UI テキストのガイドライン

このトピックでは、**Windows Phone 7** での、推奨される **UI** テキストのガイドラインを提供します。準拠は必須ではありませんが、以下のガイドラインは、アプリケーションのテキストを、十分な情報を含み、より簡潔で親しみやすいものにするのに役立つ可能性があります。ここで提供するアドバイスの多くは、アプリケーションの **UI** を **Windows Phone 7** のネイティブのテキスト形式に合わせるのに役立つので、必要に応じて参考にしてください。

## Windows Phone 7 の世界観とトーン

このセクションでは、**Windows Phone 7** アプリケーションにおける世界観とトーンについての洞察を提供します。世界観とトーンはアプリケーションによって異なるので、ここで紹介するのは例です。ただし、開発者は世界観とトーン、およびそれが **UI** テキストにどのように投影されるかを意識する必要があります。また、アプリケーション全体で、世界観とトーンにある程度の一貫性を維持する必要があります。

ここでいう世界観とは、テキストに含まれる性格のことを指します。たとえば、作家の世界観とは、作り手が書いたものに反映されているその人の全体的な性格を指します。理想を言えば、**Windows Phone 7** アプリケーションの世界観は、信頼でき、わかりやすく、対象ユーザーが使用する言語を反映している必要があります。トーンは、喜び、怒りなどのテキストの全体的な雰囲気です。アプリケーションには、親しみやすく、明るく、共感を示すようなトーンを使用することをお勧めします。怒ったトーンや機械的なトーンは使用しないでください。

たとえば、デバイスに関するなんらかの問題について友人を手助けするときのような世界観とトーンを使用します。アプリケーションに表示されるエラー メッセージを理解するのを手助けするなどのシナリオが考えられます。開発者は、問題について説明する際、堅苦しく情報が少ない返答をするべきではありません。たとえば、ほとんどのエンド ユーザーは、"エラー コード: **4560363**" というメッセージを理解できないでしょう。ですが、"情報が不足しています。次のページに進むには、テキスト ボックスに名前を入力してください。" などのメッセージなら理解できるでしょう。エンド ユーザーには、意味のある返答を、型式ばらない、わかりやすい表現で返す必要があります。ユーザーが理解できる方法で問題の解決を手助けします。

他には次のような例があります。

- ✓ "電話デバイスを同期してください": 友人にこのような言い方はしないでしょ。こうではなく、"電話を同期してください" のような表現を使用します。
- ✓ "Outlook を通じて明日の予定表イベントをスケジュールしてください": これも、親しみやすくなく、友人が話すような表現でもない、悪いテキストの例です。こうではなく、"Outlook で明日の予定を設定してください" のような表現にします。

上で示したのは簡単な例ですが、これらの例から、世界観やトーンを機械的で堅苦しいものではなく、親しみのあるものにすることの重要性がわかります。

## 大文字化

ユーザーが目にするテキストが一貫性を欠くものにならないように、アプリケーションでは大文字化に関する一貫した慣例を維持することをお勧めします。Windows Phone 7 の大文字化に関するガイドラインでは、多くの場所で小文字のレイアウトとすべて大文字のレイアウトを使用しますが、以下のレイアウトも使用します。

- ✓ **タイトルに関する大文字化の規則:** 語句の最初と最後の単語と、間にあるすべての単語を大文字で始めます。例外は、冠詞 (a、an、the)、等位接続詞 (and、but、for、not、or、so、yet)、および 4 文字以下の前置詞 (at、for、into) です。たとえば、"Vitamins in My California Raisins" のようになります。
- ✓ **文に関する大文字化の規則:** 語句の最初の単語のみ大文字で始めます。例外は、固有名詞、機能名など、テキスト内で通常大文字で始める単語です。たとえば、"Vitamins in my California raisins" のようになります。

Windows Phone 7 では、大文字化に関するガイドラインは次のとおりです。

**すべて小文字にする:**

- ✓ リストのタイトル
- ✓ リスト グループのタイトル
- ✓ プッシュ ボタン コントロールのテキストや、コマンドとして機能する単語
- ✓ リスト項目
- ✓ 検索ボックス内に例として表示されるテキスト
- ✓ 文中のリンク コントロール
- ✓ アプリケーションのページ タイトル

**文に関する大文字化の規則を適用する:**

- ✓ チェック ボックスおよびオプション ボタンのラベル
- ✓ 進行状況のインジケーター
- ✓ ステータス、通知、および説明のテキスト
- ✓ トグル スイッチ

**すべて大文字にする:**

- ✓ アプリケーションのタイトル
- ✓ 日付と時刻
- ✓ AM や PM

## 句読点

次の表では、UI 要素の句読点の標準的な規則を示します。

句読点	使用に関するガイドライン
アンパサンド (&)	設定やメニュー リストで使用してかまいません。たとえば、"日付 & 時刻"、"時計 & アラーム" のように使用します。
コロン (：)	<ul style="list-style-type: none"><li>✓ テキスト ボックス、ドロップダウン リスト、プログレスバーなどのコントロールのラベルの末尾には、コロンを使用しません。</li><li>✓ テキスト ボックスやドロップダウン リストが文中に組み込まれている場合、またはドロップダウン リストがメイン ウィンドウに表示される場合は、コロンを使用しません。</li><li>✓ グループの見出しや列見出しの末尾には、コロンを使用しません。</li><li>✓ 数値やその他の変数を後に示す場合にコロンを使用します。たとえば、"断片化の割合：XX%" のように使用します。</li></ul>
読点 (、)	<ul style="list-style-type: none"><li>✓ 3 個以上の要素を列挙する場合、要素を読点で区切ります。</li><li>✓ 3 個以上の要素を列挙する場合は必ず、最後の項目の前に読点を使用します。たとえば、"その車は赤く、小さく、速かった。" のように使用します。</li></ul>
省略記号 (...)	<ul style="list-style-type: none"><li>✓ 進行状況のインジケーターのラベルで省略記号を使用して、操作が継続中であることを示します (たとえば、ユーザーがファイルをダウンロードしている場合など)。進行状況のインジケーターが表示されていても、省略記号を使用することをお勧めします。</li><li>✓ 見出しには使用しません。</li><li>✓ ボタンのラベルには使用しません。</li></ul>
文末記号 (。 ? !)	<ul style="list-style-type: none"><li>✓ UI では、文末記号は説明のテキストでのみ使用します。説明のテキストがタイトル バーやボタンに表示される場合は、文末記号を使用しません。</li><li>✓ オプションやチェック ボックスのテキスト ラベルの末尾には、(ラベルが文の場合でも) 句点を使用しません。</li><li>✓ 文末記号が ? または ! の場合は、文末記号の後にスペースを (2 つではなく) 1 つ入れて文を区切ります。句点の後はスペースを入れません。</li><li>✓ 質問の末尾には疑問符を付けます。ただし、通常、ラベルの言葉を質問形式にすることは避けます。</li><li>✓ エラー メッセージやダイアログ ボックスのタイトルの末尾には疑問符を使用してかまいません。</li></ul>
かっこ ()	UI ではできるだけ使用しないようにします。ただし、略語やその他の短い情報を含める必要がある場合は使用してかまいません。

## UI テキストのデザインに関する考慮事項

世界観とトーンを適切に使用すると、エンド ユーザーにとって、より魅力のあるエクスペリエンスを実現で

きることを理解する必要があります。開発者は、**UI** インターフェイスに明確で親しみやすいテキストを使用してアプリケーションを構築する必要があります。

明確に定義された大文字化に関するガイドラインを使用することで、アプリケーションに一貫性を持たせ、エンド ユーザーが目にするテキストが一貫性を欠くものにならないようにすることができます。

**UI** 文字列では適切な句読点を正しく使用する必要があります。繰り返しになりますが、アプリケーションでは、一貫性に重点を置くことをお勧めします。