

## CL-06 メッセージ通知機能

### ■ 概要

本機能は、エラーメッセージや確認メッセージといった、ユーザへ通知するダイアログメッセージの表示処理の実装を一元管理することで、一貫性のある UI の提供を支援する。また、メッセージ表示処理の変更時にかかる維持管理コストを軽減する。

本機能では MessageNotifier クラスの static メソッドとして目的に合わせたメッセージ通知メソッドを提供している。開発者は、MessageNotifier が提供する抽象的なメソッドを呼び出すことで、内部の実装方式を意識することなくユーザへのメッセージ表示処理を実行できる。

MessageNotifier クラスは、「CM-02 インスタンス管理機能」により、メッセージ通知処理を実装した IMessageNotificationDisplayer インタフェース実装クラスのインスタンスを取得し、ユーザへのメッセージ表示処理を実行する。アーキテクトまたは業務共通開発者は、IMessageNotificationDisplayer インタフェースを実装したクラスのインスタンスを依存性注入 (DI 設定) することで、プロジェクトの UI 規約に合わせて、メッセージの表示方法を切り替えることができる。

デフォルトでは、System.Windows.Forms.MessageBox クラスを使用してメッセージを表示する MessageBoxNotificationDisplayer クラスが実行される。しかし、.NET Framework が標準提供する MessageBox は、オーナー画面 (ダイアログメッセージの表示元画面) の中央に表示できないという問題があるため、MessageBox と同等の機能を持ちつつ、オーナー画面の中央にもダイアログ表示可能な MessageFormNotificationDisplayer クラスも提供する。

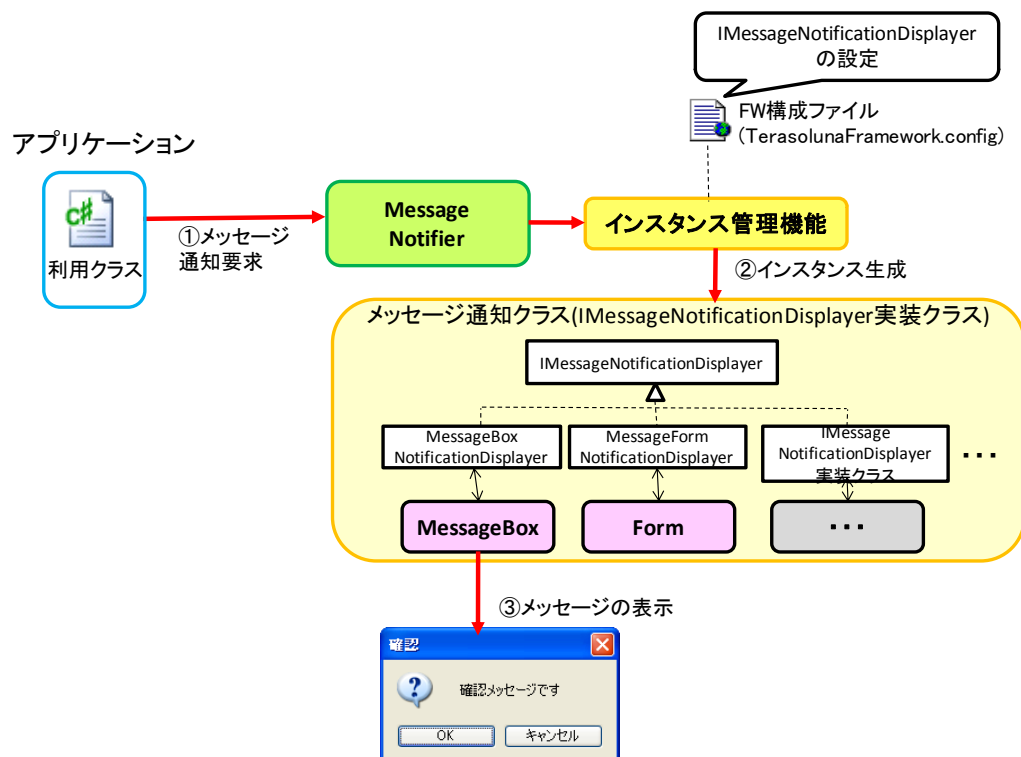


図 1 メッセージ通知機能の動作概念図

## ■ 使用方法

### ◆ MessageNotifier クラスの実行

本機能を利用するには、ユーザへのメッセージを通知する目的に合わせて MessageNotifier が提供する static メソッドを実行する。以下に、MessageNotifier クラスが提供するメソッドを示す。

表 1 MessageNotifier クラスの提供メソッド

| 項番 | メソッド   | 第 1 引数 | 第 2 引数 | 第 3 引数 | 第4引数 | 戻り値                            | 説明                    |
|----|--|--------|--------|--------|------|--------------------------------|-----------------------|
| 1  | <code>public static bool ShowConfirmMessage ( string message)</code>                                       | メッセージ  | —      | —      | —    | ユーザ確認の結果が OK や Yes の場合<br>true | ユーザへ確認を行うメッセージを表示する。  |
| 2  | <code>public static bool ShowConfirmMessage (IWin32Window owner, string message)</code>                    | オーナー画面 | メッセージ  | —      | —    |                                |                       |
| 3  | <code>public static bool ShowConfirmMessage (string message, string caption)</code>                        | メッセージ  | タイトル   | —      | —    |                                |                       |
| 4  | <code>public static bool ShowConfirmMessage (IWin32Window owner, string message, string caption)</code>    | オーナー画面 | メッセージ  | タイトル   | —    |                                |                       |
| 5  | <code>public static void ShowInformationMessage(string message)</code>                                     | メッセージ  | —      | —      | —    | —                              | ユーザへ情報提供するメッセージを表示する。 |
| 6  | <code>public static void ShowInformationMessage(IWin32Window owner, string message)</code>                 | オーナー画面 | メッセージ  | —      | —    |                                |                       |
| 7  | <code>public static void ShowInformationMessage(string message, string caption)</code>                     | メッセージ  | タイトル   | —      | —    |                                |                       |
| 8  | <code>public static void ShowInformationMessage(IWin32Window owner, string message, string caption)</code> | オーナー画面 | メッセージ  | タイトル   | —    |                                |                       |

| 項番 | メソッド  | 第 1 引数 | 第 2 引数 | 第 3 引数 | 第4引数 | 戻り値      | 説明                           |
|----|---|--------|--------|--------|------|----------|------------------------------|
| 9  | <code>public static void ShowErrorMessage (string message)</code>   | メッセージ  | —      | —      | —    | —        | ユーザへエラーメッセージを表示する。           |
| 10 | <code>public static void ShowErrorMessage (IWin32Window owner, string message)</code>                                       | オーナー画面 | メッセージ  | —      | —    | —        |                              |
| 11 | <code>public static void ShowErrorMessage (string message, string caption)</code>   | メッセージ  | タイトル   | —      | —    | —        |                              |
| 12 | <code>public static void ShowErrorMessage (IWin32Window owner, string message, string caption)</code>                       | オーナー画面 | メッセージ  | タイトル   | —    | —        |                              |
| 13 | <code>public static DialogResult ShowMessage (string dialogName, IWin32Window owner, string message, string caption)</code> | ダイアログ名 | オーナー画面 | メッセージ  | タイトル | ユーザ確認の結果 | 指定したダイアログ名に応じたユーザメッセージを表示する。 |

以下に、MessageNotifier クラスを用いて、情報メッセージを出力する実装例を示す。

```
MessageNotifier.ShowInformationMessage(this, "情報メッセージです", "タイトル");
```

リスト 1 情報メッセージを表示する例

## ■ FW 構成ファイル(TerasolunaFramework.config)の設定

本機能を利用する場合、「CM-02 インスタンス管理機能」を利用して、FW 構成ファイル (TerasolunaFramework.config) に、IMessageNotificationDisplay の実装クラスの DI 設定を行う。なお、DI 設定を省略した場合は、デフォルト実装である MessageBoxNotificaitonDisplay クラスが利用される。

以下に、TerasolunaFramework.config の記述例を示す。この例では、IMessageNotificationDisplay の実装クラスを MessageFormNotificaitonDisplay クラスに設定している。

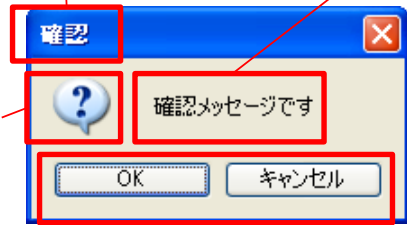
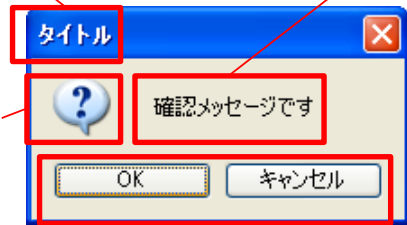
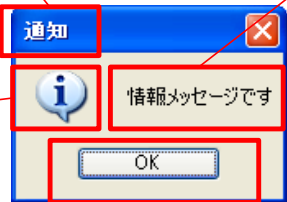
```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  . . .
  <unity>
    <typeAliases>
      <typeAlias alias=" IMessageNotificationDisplay "
        type="Terasoluna.Windows.Forms.MessageNotification.IMessageNotificationDisplay,
          Terasoluna.Windows.Forms"/>
      <typeAlias alias="MessageFormNotificationDisplay"
type="Terasoluna.Windows.Forms.MessageNotification.MessageFormNotificationDisplay,
          Terasoluna.Windows.Forms"/>
    </typeAliases>
    <containers>
      <container>
        <types>
          <!-- メッセージ通知機能の設定 -->
          <type type=" IMessageNotificationDisplay " mapTo="MessageFormNotificationDisplay"/>
        </types>
        . . .
      </container>
    </containers>
  </unity>
</configuration>
```

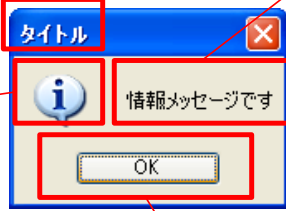
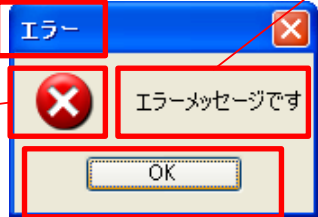
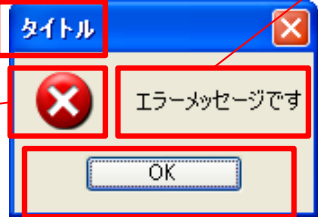
リスト 2 TerasolunaFramework.config の記述例

## ■ 画面イメージ

### ◆ MessageBoxNotificationDisplayer の画面イメージ

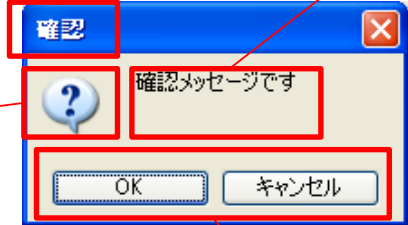
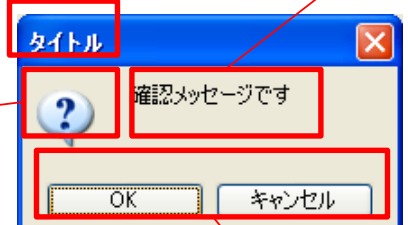
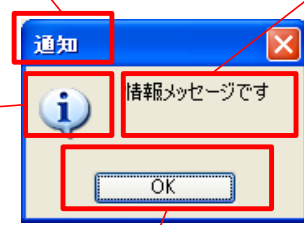
MessageBoxNotificationDisplayer クラスを使用した場合の画面イメージを示す。

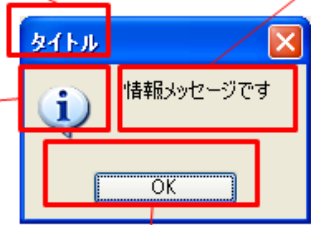
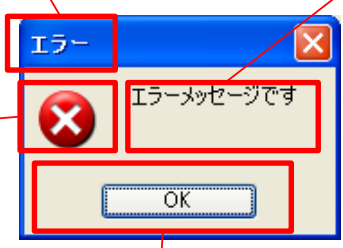
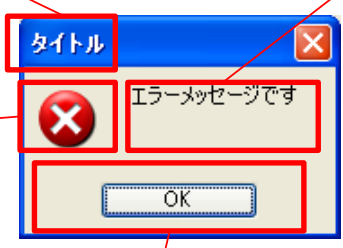
| 項番 | メソッド   | 画面イメージ  |
|----|--|---|
| 1  | <code>public static bool ShowConfirmMessage ( string message)</code><br><code>public static bool ShowConfirmMessage (IWin32Window owner, string message)</code>                                | <p>既定のcaption(="確認") 指定したmessage</p> <p>アイコンは、<br/>MessageBoxIcon.Question</p>  <p>ボタンは、<br/>MessageBoxButtons.OKCancel</p> |
| 2  | <code>public static bool ShowConfirmMessage (string message, string caption)</code><br><code>public static bool ShowConfirmMessage (IWin32Window owner, string message, string caption)</code> | <p>指定したcaption 指定したmessage</p> <p>アイコンは、<br/>MessageBoxIcon.Question</p>  <p>ボタンは、<br/>MessageBoxButtons.OKCancel</p>     |
| 3  | <code>public static void ShowInformationMessage(string message)</code><br><code>public static void ShowInformationMessage(IWin32Window owner, string message)</code>                           | <p>既定のcaption(="通知") 指定したmessage</p> <p>アイコンは、<br/>MessageBoxIcon.Information</p>  <p>ボタンは、<br/>MessageBoxButtons.OK</p> |

| 項番 | メソッド  | 画面イメージ  |
|----|---|---|
| 4  | <pre>public static void ShowInformationMessage(   string message,   string caption)</pre>   | <p>指定したcaption      指定したmessage</p>  <p>アイコンは、<br/>MessageBoxIcon.Information</p> <p>ボタンは、<br/>MessageBoxButtons.OK</p>  |
|    | <pre>public static void ShowInformationMessage(   IWin32Window owner,   string message,   string caption)</pre>                   |   |
| 5  | <pre>public static void ShowErrorMessage(   string message)</pre>   | <p>既定のcaption(="エラー")      指定したmessage</p>  <p>アイコンは、<br/>MessageBoxIcon.Error</p> <p>ボタンは、<br/>MessageBoxButtons.OK</p> |
|    | <pre>public static void ShowErrorMessage(   IWin32Window owner,   string message)</pre>   |   |
| 6  | <pre>public static void ShowErrorMessage(   string caption)</pre>   | <p>指定したcaption      指定したmessage</p>  <p>アイコンは、<br/>MessageBoxIcon.Error</p> <p>ボタンは、<br/>MessageBoxButtons.OK</p>       |
|    | <pre>public static void ShowErrorMessage(   IWin32Window owner,   string message,   string caption)</pre>                         |   |
| 7  | <pre>public static DialogResult ShowMessage(   string dialogName,   IWin32Window owner,   string message,   string caption)</pre> | <p>未実装のため、NotSupportedExceptionをスローする。<br/>ダイアログ名を指定して実行する場合には本クラスを継承し、メソッドをオーバーライドすること。</p>  |

## ◆ MessageFormNotificationDisplayer の画面イメージ

MessageFormNotificationDisplayer クラスを使用した場合の画面イメージを示す。

| 項番 | メソッド   | 画面イメージ   |
|----|--|--|
| 1  | <code>public static bool ShowConfirmMessage(string message)</code>                                     | <p>既定のcaption(="確認") 指定したmessage</p>  <p>アイコンは、SystemIcons.Question</p> <p>ボタンは、既定の文言("OK", "キャンセル")</p> |
|    | <code>public static bool ShowConfirmMessage(IWin32Window owner, string message)</code>                 |  |
| 2  | <code>public static bool ShowConfirmMessage(string message, string caption)</code>                     | <p>指定したcaption 指定したmessage</p>  <p>アイコンは、SystemIcons.Question</p> <p>ボタンは、既定の文言("OK", "キャンセル")</p>     |
|    | <code>public static bool ShowConfirmMessage(IWin32Window owner, string message, string caption)</code> |  |
| 3  | <code>public static void ShowInformationMessage(string message)</code>                                 | <p>既定のcaption(="通知") 指定したmessage</p>  <p>アイコンは、SystemIcons.Information</p> <p>ボタンは、既定の文言("OK")</p>     |
|    | <code>public static void ShowInformationMessage(IWin32Window owner, string message)</code>             |  |

| 項番 | メソッド   | 画面イメージ  |
|----|--|---|
| 4  | <pre>public static void ShowInformationMessage(string message, string caption)</pre>                                     | <p>指定したcaption</p> <p>指定したmessage</p>  <p>アイコンは、<br/>SystemIcons.Information</p> <p>ボタンは、<br/>既定の文言("OK")</p>   |
|    | <pre>public static void ShowInformationMessage(IWin32Window owner, string message, string caption)</pre>                 |   |
| 5  | <pre>public static void ShowErrorMessage(string message)</pre>   | <p>既定のcaption(="エラー")</p> <p>指定したmessage</p>  <p>アイコンは、<br/>SystemIcons.Error</p> <p>ボタンは、<br/>既定の文言("OK")</p>   |
|    | <pre>public static void ShowErrorMessage(IWin32Window owner, string message)</pre>                                       |   |
| 6  | <pre>public static void ShowErrorMessage(string message, string caption)</pre>   | <p>指定したcaption</p> <p>指定したmessage</p>  <p>アイコンは、<br/>SystemIcons.Error</p> <p>ボタンは、<br/>既定の文言("OK")</p>   |
|    | <pre>public static void ShowErrorMessage(IWin32Window owner, string message, string caption)</pre>                       |   |
| 7  | <pre>public static DialogResult ShowMessage(string dialogName, IWin32Window owner, string message, string caption)</pre> | <p>dialogName で指定した文字列に一致するダイアログを返す。<br/>もし、指定された dialogName のインスタンスが存在しない場合は、<br/>InvalidOperationException が発生する。</p> <p>既定では、MessageFormNotificationDisplayer クラスの定数を指定できる。</p> <ul style="list-style-type: none"> <li>・ConfirmMessageDialogName : ShowConfirmMessage メソッドと同等</li> <li>・InformationMessageDialogName : ShowInformationMessage メソッドと同等</li> <li>・ErrorMessageDialogName : ShowErrorMessage メソッドと同等</li> </ul> |



## ■ 内部構成

### ◆ 構成クラス

本機能を構成するクラスを以下に示す。

表 2 構成クラス一覧

| 項番  | クラス名                             | 説明   |
|---|----------------------------------|--|
| Terasoluna.Windows.Forms.MessageNotification 名前空間 |                                  |  |
| 1   | MessageNotifier                  | 本機能のエントリクラス。   |
| 2   | IMessageNotificationDisplayer    | メッセージを表示するためのインタフェース。  |
| 3   | MessageBoxNotificationDisplayer  | IMessageNotificationDisplayer インタフェースのデフォルト実装クラス。<br>MessageBox でメッセージを表示する。 |
| 4   | MessageFormNotificationDisplayer | IMessageNotificationDisplayer インタフェースの実装クラス。<br>Form によるダイアログにメッセージを表示する。    |
| 5   | IMessageDialog                   | メッセージを表示するダイアログを表すインタフェース。   |
| 6   | DefaultMessageDialog             | IMessageDialog の実装クラス。<br>Form を使ったダイアログを実装している。                             |

## ■ 拡張ポイント

### ◆ 標準提供のダイアログ上の固定メッセージの変更

本機能が標準で提供するダイアログのデフォルトメッセージは、フレームワーク内で定義されたリソースファイル(TerasolunaWindows.Forms.dll の DisplayResources.resx)で管理している。メッセージを変更する場合は、「CM-07 メッセージ管理機能」により、デフォルトメッセージの置き換えを実施する。

詳細な手順については、「CM-07 メッセージ管理機能」の機能説明書を参照のこと。

### ◆ 既存のダイアログの表示方法の変更

既存のダイアログのボタンの種類を変えたい場合やプロジェクトの UI 規約に合わせてメッセージ表示方法を追加したい場合は、IMessageNotificationDisplayer インタフェースの実装クラスを作成する。

ここでは MessageBoxNotificationDisplayer クラスを継承し、ShowConfirmMessage メソッドをオーバーライドすることで、MessageBox のボタンを MessageBox.YesNo(“はい”, “いいえ”)に変更する例を示す。

以下に実装例を示す。

```
public class SampleMessageBoxNotificationDisplay : MessageBoxNotificationDisplay
{
    public override bool ShowConfirmMessage(IWin32Window owner, string message, string caption)
    {
        ///アイコンをMessageBoxButtons.YesNoに変更
        DialogResult result =
            MessageBox.Show(owner, message, caption,
                MessageBoxButtons.YesNo, MessageBoxIcon.Question,
                MessageBoxDefaultButton.Button1, GetMessageBoxOptions(owner));
        return result == DialogResult.Yes;
    }
}
```

リスト 3 ShowConfirmMessage メソッドの変更例

また、TerasolunaFramework.config に IMessageNotificationDisplay 実装クラスの DI 設定を行う。

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    . . .
    <unity>
        <typeAliases>
            <typeAlias alias=" IMessageNotificationDisplay "
                type="Terasoluna.Windows.Forms.MessageNotification.IMessageNotificationDisplay,
                    Terasoluna.Windows.Forms"/>
            <typeAlias alias="MessageFormNotificationDisplay"
                type="Terasoluna.Windows.Forms.MessageNotification.MessageFormNotificationDisplay,
                    Terasoluna.Windows.Forms"/>
            <typeAlias alias="SampleMessageFormNotificationDisplay"
                type="BusinessApplication.SampleMessageBoxNotificationDisplay,
                    BusinessApplication"/>
        </typeAliases>
        <containers>
            <container>
                <types>
                    <!-- メッセージ表示機能のDI設定 -->
                    <type type=" IMessageNotificationDisplay "
                        mapTo="SampleMessageBoxNotificationDisplay"/>
                </types>
            </container>
        </containers>
    </unity>
</configuration>
```

リスト 4 TerasolunaFramework.config の記述例

実行結果は、以下の通り。

```
bool result = MessageNotifier.ShowConfirmMessage(this, "確認メッセージです");
```

リスト 5 MessageNotifier.ShowConfirmMessage メソッドの実行例

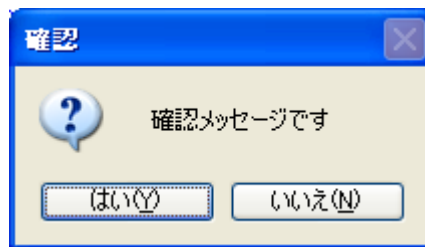


図 2 実行結果

## ◆ 独自のダイアログ名によるメッセージ表示方法の追加

独自のダイアログ名を追加したいなど、プロジェクトの UI 規約に合わせてメッセージ表示方法を追加したい場合も同様に、IMessageNotificationDisplayer インタフェースの実装クラスを作成する。

ここでは MessageFormNotificationDisplayer クラスを継承し、ShowMessage メソッドで指定可能な dialogName を追加した例(dialogName として、“WarningDilaogName”を指定すると警告アイコンのダイアログを生成)を示す。

MessageFormNotificationDisplayer クラスでは、RegsiterMessageDialog メソッドをオーバーライドして、UnityContainer.RegisterType メソッドで、IMessageDialog インタフェースを持つ実装クラスを DI で追加設定する。この時、引数に渡す登録名は、追加したいダイアログ名にする。

以下に実装例を示す。

```
public class SampleMessageFormNotificationDisplayer : MessageFormNotificationDisplayer
{
    /// 追加したダイアログ名
    public const string WarningDialogName = "WarningDialogName";

    /// ダイアログの作成
    protected override void RegisterMessageDialog(string dialogName)
    {
        base.RegisterMessageDialog(dialogName);

        /// 警告ダイアログを独自に追加
        switch (dialogName)
        {
            case WarningDialogName:
                Container.RegisterType<IMessageDialog, DefaultMessageDialog>(
                    WarningDialogName,
                    new InjectionProperty("IconType", MessageBoxIcon.Warning),
                    new InjectionProperty("ButtonsType", MessageBoxButtons.OK),
                    new InjectionProperty("DefaultButtonType", MessageBoxDefaultButton.Button1));
                break;
            default:
                break;
        }
    }
}
```

リスト 6 新しいダイアログ名の追加の例

また、TerasolunaFramework.config に IMessageNotificationDisplayer 実装クラスの DI 設定を行う。

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  . . .
  <unity>
    <typeAliases>
      <typeAlias alias=" IMessageNotificationDisplay "
        type="Terasoluna.Windows.Forms.MessageNotification.IMessageNotificationDisplay,
          Terasoluna.Windows.Forms"/>
      <typeAlias alias="MessageFormNotificationDisplay"
        type="Terasoluna.Windows.Forms.MessageNotification.MessageFormNotificationDisplay,
          Terasoluna.Windows.Forms"/>
      <typeAlias alias="SampleMessageFormNotificationDisplay"
        type="BusinessApplication.SampleMessageFormNotificationDisplay,
          BusinessApplication"/>
    </typeAliases>
    <containers>
      <container>
        <types>
          <!-- メッセージ表示機能のDI設定 -->
          <type type=" IMessageNotificationDisplay "
            mapTo="SampleMessageFormNotificationDisplay"/>
        </types>
      </container>
    </containers>
  </unity>
</configuration>
```

リスト 7 TerasolunaFramework.config の記述例

実行結果は、以下の通り。

追加したダイアログ名を表示するには、MessageNotifier.ShowMessage メソッドで、ダイアログ名を指定し実行する。

```
MessageNotifier.ShowMessage(SampleMessageFormNotificationDisplay.WarningDialogName,
  this, "警告メッセージです", "タイトル");
```

リスト 8 MessageNotifier.ShowMessage メソッドの実行例

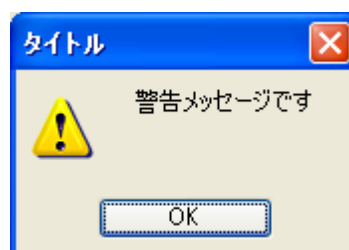


図 3 実行結果

## ■ 関連機能

- CM-02 インスタンス管理機能
- CM-07 メッセージ管理機能