

インストール時に FireWall の例外設定を構成する

注)このドキュメントは、InstallShield 2012 Premier Edition を基に作成しています。InstallShield 2012 以外のバージョンでは設定名などが異なる場合もあります。

概要

インストールを行ったアプリケーションが外部環境と通信する製品の場合、正しく実行させるためには Windows デフォルトの Firewall に対してアプリケーション許可の設定が必要な場合があります。InstallShield はデフォルトの機能として、この設定をサポートしていませんが、インストール時に netsh コマンドを実行することで Firewall の構成が可能です。

この記事では MSI 形式インストーラ・InstallScript 形式インストーラにて、カスタム処理として Firewall の例外設定を構成する場合の手順の一例をご紹介します。

A. 基本の MSI 形式インストーラの場合

基本の MSI 形式インストーラでは、EXE形式カスタムアクションより netsh コマンドを実行することで Firewall の構成が可能です。

I) インストール用カスタムアクションの作成

netsh コマンドを実行して Firewall の例外設定を行うカスタムアクションを追加します。

1. [動作とロジック]－[カスタム アクションとシーケンス]ビューにて、[カスタムアクション]を右クリックして[新しい EXE]－[ディレクトリを参照するパス]を選択します。



2. カスタムアクションの一覧に新規カスタムアクション「NewCustomAction1」が追加されるので、任意の名称に変更します。(この手順では「caAddFireWall」に設定)
3. Firewall の例外設定は netsh コマンドより実行可能です。以下はインストーラより **INSTALLDIR** 配下のファイル **SampleApp.exe** を表示名 **Sample** として例外設定する場合のコマンドの記述例です。

```
cmd.exe /c"netsh firewall add allowedprogram program = %"[INSTALLDIR]SampleApp.exe%" name = Sample"
```

上記の例に従って、インストールするアプリケーションに対する例外許可のコマンドを作成します。

4. カスタムアクション「caAddFireWall」にて、以下の設定を行います。

設定プロパティ名	設定内容	説明
作業ディレクトリ	SystemFolder	ターゲットシステム上の cmd.exe を使用するため SystemFolder 識別子を使用して、system32 フォルダを作業フォルダに指定
ファイル名とコマンドライン	cmd.exe /c"netsh firewall add allowedprogram program = ¥"[INSTALLDIR]SampleApp.exe¥" name = sample"	手順 I - 3 で作成したコマンドを指定します
スクリプト内実行	システムコンテキストで遅延実行	
インストール実行シーケンス	次の後:InstallFiles	
インストール実行条件	NOT Installed	初回インストール時のみ実行される条件

II) インストールのロールバック用カスタムアクションの作成

インストールがキャンセルされた場合にインストール処理をロールバックするカスタムアクションを追加します。

1. 同ビューにて[カスタムアクション]を右クリックして[新しいEXE]—[ディレクトリを参照するパス]を選択します。
2. もう一つ新規カスタムアクションが追加されるので、任意の名称に変更します。(この手順では「caAddFireWall_RollBack」に設定)
3. 追加した Firewall の例外設定についても netsh コマンドより削除可能です。以下は例外として追加されていたアプリケーション **SampleApp.exe** を例外から外す場合のコマンドの記述例です。

```
cmd.exe /c"netsh firewall delete allowedprogram program = ¥"[INSTALLDIR]SampleApp.exe¥""
```

上記の例に従って、アプリケーションに対する例外許可のコマンドを作成します。

4. カスタムアクション「caAddFireWall_RollBack」にて、以下の設定を行います。

設定プロパティ名	設定内容	説明
作業ディレクトリ	SystemFolder	ターゲットシステム上の cmd.exe を使用するため SystemFolder 識別子を使用して、system32 フォルダを作業フォルダに指定
ファイル名とコマンドライン	cmd.exe /c"netsh firewall delete allowedprogram program = ¥"[INSTALLDIR]SampleApp.exe¥""	手順 II - 3 で作成したコマンドを指定します
スクリプト内実行	システムコンテキストでロールバック実行	
インストール実行シーケンス	次の後:InstallFiles	
インストール実行条件	NOT Installed	初回インストール時のみ実行される条件

※ シーケンスとしては caAddFireWall の前に caAddFireWall_RollBack が配置されるようになります。

Ⅲ) アンインストール用カスタムアクションの作成

アンインストール時に Firewall の例外設定を削除するカスタムアクションを追加します。

1. Firewall の例外設定を削除するコマンドは手順 II で追加したロールバックのカスタムアクションと同等となりますので、カスタムアクションビューにて、カスタムアクション「caAddFireWall_RollBack」を右クリックして [クローン] を選択します。



2. 複製されたカスタムアクションを任意の名称に変更します。(この手順では「caDeleteFireWall」に設定)
3. カスタムアクション「caDeleteFireWall」にて、以下の設定を行います。

設定プロパティ名	設定内容	説明
スクリプト内実行	システムコンテキストで遅延実行	
インストール実行シーケンス	次の後: RemoveDuplicateFiles	
インストール実行条件	REMOVE="ALL"	アンインストール時のみ実行される条件

※ 上記以外の項目は複製後の状態をそのまま使用します。

Ⅳ) アンインストールのロールバック用カスタムアクションの作成

アンインストールがキャンセルされた場合にアンインストール処理をロールバックするカスタムアクションを追加します。

1. アンインストールのロールバックは Firewall の例外設定を再設定する動作となりますので、手順 I で作成したカスタムアクションと同様の処理となります。カスタムアクションビューにて、カスタムアクション「caAddFireWall」を右クリックして [クローン] を選択します。

2. 複製されたカスタムアクションを任意の名称に変更します。(この手順では「caDeleteFireWall_RollBack」に設定)

カスタムアクション「caDeleteFireWall_RollBack」にて、以下の設定を行います。

設定プロパティ名	設定内容	説明
スクリプト内実行	システムコンテキストでロールバック実行	
インストール実行シーケンス	次の後: RemoveDuplicateFiles	
インストール実行条件	REMOVE="ALL"	アンインストール時のみ実行される条件

※ シーケンスとしては caDeleteFireWall の前に caDeleteFireWall_RollBack が配置されるようになります。

B. InstallScript 形式インストーラの場合

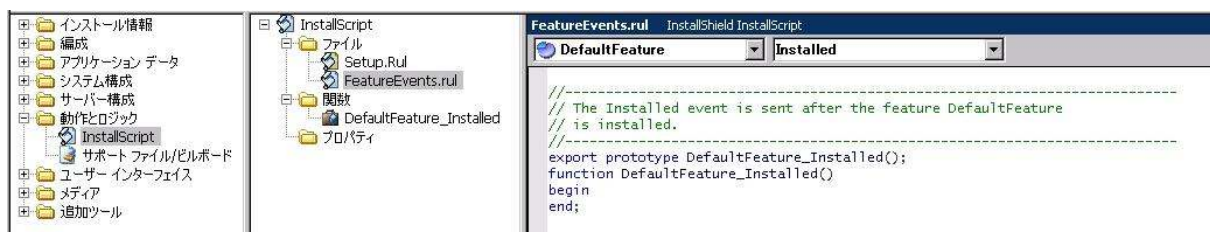
InstallScript 形式インストーラでは、例外設定を行うファイルが関連付けられた機能の機能イベントにて、LaunchApplication 関数より netsh コマンドを実行することで Firewall 例外設定の構成が可能です。

I) インストール用 InstallScript カスタムコードの作成

netsh コマンドを実行して Firewall の例外設定を行う InstallScript カスタムコードを追加します。

1. [動作とロジック] - [InstallScript]ビューにて、Setup.rul を選択します。スクリプトエディタ上部のコンボボックスを [※<任意の機能名>] - [Installed]に切り替えます。

(※ 任意の機能名の箇所は例外設定を行うファイルが含まれる機能をご指定ください。)



(この手順では DefaultFeature という機能の機能イベントを追加しています。)

2. 新しく追加されたスクリプトファイル FeatureEvents.rul を選択して、追加された機能イベント DefaultFeature_Installed に netsh コマンドを実行する InstallScript コードを追加します。

記述例:

```
export prototype DefaultFeature_Installed();
function DefaultFeature_Installed()

    NUMBER nResult;
    STRING szCmdLine;

begin

    // Firewall の例外設定を行うコマンドを作成します。
    // この例では、TARGETDIR に配置された SampleApp.exe を Sample という名称で設定しています。
    szCmdLine = "/c netsh firewall add allowedprogram program = " + "%"+TARGETDIR+"SampleApp.exe" +
        "%name= Sample ";

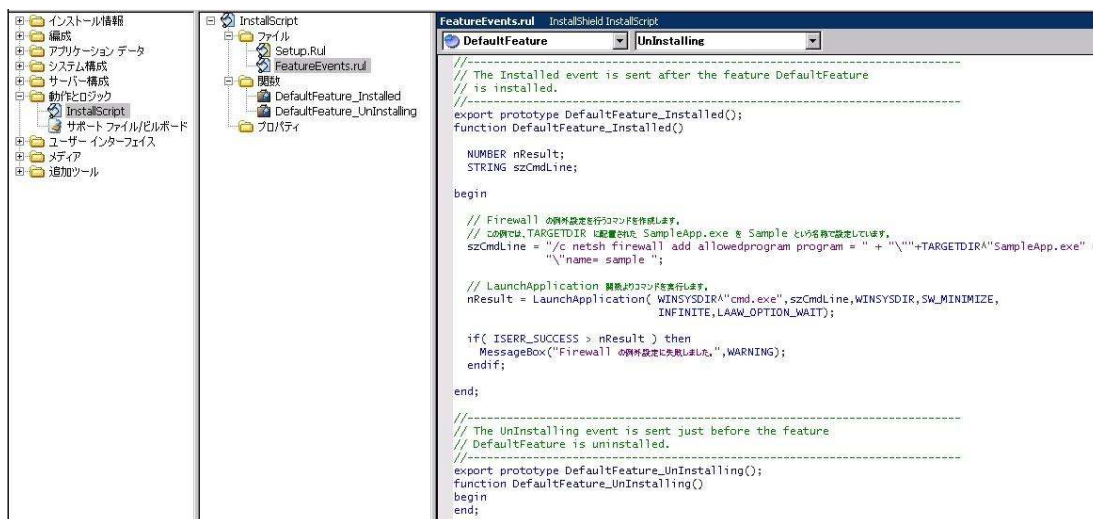
    // LaunchApplication 関数よりコマンドを実行します。
    nResult = LaunchApplication( WINSYSDIR+"cmd.exe",szCmdLine,WINSYSDIR,SW_MINIMIZE,
        INFINITE,LAAW_OPTION_WAIT);

    if( ISERR_SUCCESS > nResult ) then
        MessageBox("Firewall の例外設定に失敗しました。",WARNING);
    endif;
end;
```

II) アンインストール用 InstallScript カスタムコードの作成

アンインストール時に例外設定を削除する InstallScript カスタムコードを同じように追加します。

1. [動作とロジック]-[InstallScript]ビューにて、Setup.rul を選択します。スクリプトエディタ上部のコンボボックスを[DefaultFeature]-[Uninstalling]に切り替えます。



2. 追加された機能イベント DefaultFeature_UnInstalling に netsh コマンドを実行して、インストール時に追加した Firewall の例外設定を削除する InstallScript コードを追加します。

記述例:

```

export prototype DefaultFeature_UnInstalling();
function DefaultFeature_UnInstalling()

    NUMBER nResult;
    STRING szCmdLine;

begin

    // Firewall の例外設定を削除するコマンドを作成します。
    szCmdLine = "/c netsh firewall delete allowedprogram program = " + "%"+TARGETDIR+"SampleApp.exe" + "%";

    // LaunchApplication 関数よりコマンドを実行します。
    nResult = LaunchApplication( WINSYSDIR+"cmd.exe",szCmdLine,WINSYSDIR,SW_MINIMIZE,INFINITE,
        LAAW_OPTION_WAIT);

    if( ISERR_SUCCESS > nResult ) then
        MessageBox("Firewall の例外設定の削除に失敗しました。",WARNING);
    endif;
end;

```