

共享contract

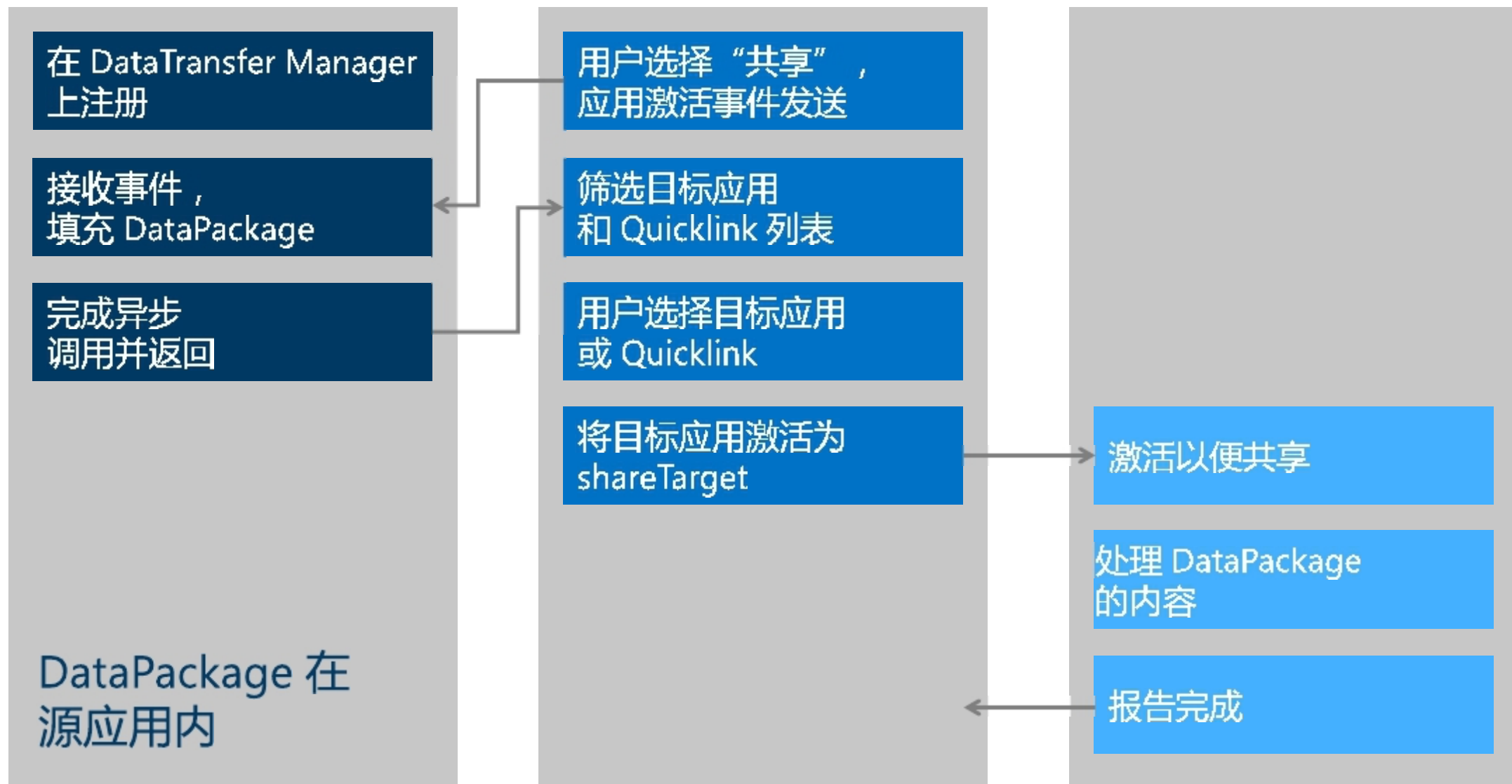
原理

从源应用到目标应用的共享

源应用

共享代理

共享目标应用



共享源

DataTransferManager

设置共享源应用

```
//创建DataManager, 添加Windows.ApplicationModel.DataTransfer
DataManager dtm;

//进入时注册DataRequested事件
protected override void OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);
    dtm = DataManager.GetForCurrentView();
    //创建event handler
    dtm.DataRequested += dtm_DataRequested;
}
//离开时移除DataRequested事件
protected override void OnNavigatingFrom(NavigatingCancelEventArgs e)
{
    base.OnNavigatingFrom(e);
    dtm.DataRequested -= dtm_DataRequested;
}
//通过该方法提供需要共享的信息
void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    /* todo */
}
```

设置共享源应用

//通过DataPackage共享Text

```
void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    //创建DataPackage对象，并设置相应属性
    string textSource = "共享具体内容";
    string textTitle = "共享标题";
    string textDescription = "共享描述（可选）";
    DataPackage data = args.Request.Data;
    data.Properties.Title = textTitle;
    data.Properties.Description = textDescription;
    data.SetText(textSource);

    /* todo list */
}
```

设置共享源应用

//通过DataPackage共享Url

```
void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    ...

    /* todo list 1 */

    Uri link = new Uri("http://msdn.microsoft.com/en-us/library/windows/apps/hh758314.aspx");

    ...

    data.SetUri(link);
}
```


设置共享源应用

//通过DataPackage共享HTML

```
void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    ...

    /* todo list 2
    1. 调用HtmlFormatHelper
    2. HtmlFormatHelper将另外一个原始的字符串转换为html 标记能够被解析和使用
    3. 当你选择了共享目标程序时,你会看到格式化的 html 显示在 body 中
    */

    String HTML = "<strong>HTML FORMATTED Text.</strong>";

    ...

    data.SetHtmlFormat(HtmlFormatHelper.CreateHtmlFormat(HTML));
}
```

设置共享源应用

```
//通过DataPackage共享File
async void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    ...

    /* todo list 3.1
    1. 异步方式获取文件
    2. 添加到 List<IStorageItem>集合
    */

    DataRequestDeferral GetFiles = args.Request.GetDeferral();

    ...

    try {
        StorageFile textFile = await Package.Current.InstalledLocation.GetFilesAsync("FileForSharing.txt");
        List<IStorageItem> files = new List<IStorageItem>();
        files.Add(textFile);
        data.SetStorageItems(files);
    }
    finally {
        GetFiles.Complete();
    }
}
```

设置共享源应用

//通过DataPackage共享图片

```
async void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    ...

    /* todo list 3.2
    1. 异步方式获取文件
    2. 添加到 List<IStorageItem>集合
    */

    DataRequestDeferral GetFiles = args.Request.GetDeferral();

    ...

    try {
        StorageFile imageFile = await Package.Current.InstalledLocation.GetFileAsync("Assets\\sample.jpg");
        //新方法：使用文件的方法也是可以的，但是要根据共享目标应用是否可用
        data.Properties.Thumbnail = RandomAccessStreamReference.CreateFromFile(imageFile);
        data.SetBitmap(RandomAccessStreamReference.CreateFromFile(imageFile));
    }
    finally {
        GetFiles.Complete();
    }
}
```

设置共享源应用

//通过DataPackage共享自定义格式

```
async void dtm_DataRequested(DataTransferManager sender, DataRequestedEventArgs args)
{
    ...

    /* todo list 4
    http://schema.org/docs/full.html
    http://msdn.microsoft.com/en-us/library/windows/apps/hh750306.aspx
    */
}
```

共享目标

作为共享目标应用需要考虑的事情

- 注册为共享目标，指定接受的格式



作为共享目标应用需要考虑的事情

- 注册为共享目标，指定接受的格式

数据格式	
数据格式:	Text
数据格式	
数据格式:	URL
数据格式	
数据格式:	HTML
数据格式	
数据格式:	

数据格式	
数据格式:	StorageItems
数据格式	
数据格式:	Bitmap
数据格式	
数据格式:	http://schema.org/DateTime
数据格式	
数据格式:	

设置共享目标应用

```
//App.xaml.cs
```

```
//共享目标应用被激活的时候调用
```

```
protected override void OnShareTargetActivated(ShareTargetActivatedEventArgs  
args)  
{  
    var rootFrame = new Frame();  
    //加载的不一定是MainPage,一般是单独一个页面获取  
    rootFrame.Navigate(typeof(MainPage), args.ShareOperation);  
    Window.Current.Content = rootFrame;  
    Window.Current.Activate();  
}
```


设置共享目标应用

```
//MainPage.xaml.cs || TargetPage.xaml.cs
```

```
//实现OnNavigateTo方法
```

```
ShareOperation share;  
string title;  
string description;  
string text;  
string customData;  
string customDataFormat = "http://schema.org/DateTime/";  
string formattedText;  
Uri uri;  
IReadOnlyList<IStorageItem> storageItems;  
IRandomAccessStreamReference bitmapImage;  
IRandomAccessStreamReference thumbImage;
```

设置共享目标应用

```
protected override async void OnNavigatedTo(NavigationEventArgs e)
{
    share = e.Parameter as ShareOperation;
    await Task.Factory.StartNew(async () =>
    {
        title = share.Data.Properties.Title;
        description = share.Data.Properties.Description;
        thumbImage = share.Data.Properties.Thumbnail;
        //Text检查
        if (share.Data.Contains(StandardDataFormats.Html))
        {
            formattedText = await share.Data.GetHtmlFormatAsync();
        }
    })
}
```

设置共享目标应用

//URI检查

```
if (share.Data.Contains(StandardDataFormats.Uri))  
{  
    uri = await share.Data.GetUriAsync();  
}
```

//无格式Text检查

```
if (share.Data.Contains(StandardDataFormats.Text))  
{  
    text = await share.Data.GetTextAsync();  
}
```

//File检查

```
if (share.Data.Contains(StandardDataFormats.StorageItems))  
{  
    storageItems = await share.Data.GetStorageItemsAsync();  
}
```

设置共享目标应用

//自定义数据检查

```
if (share.Data.Contains(customDataFormat))  
{  
    customData = await share.Data.GetTextAsync(customDataFormat);  
}
```

//图片检查

```
if (share.Data.Contains(StandardDataFormats.Bitmap))  
{  
    bitmapImage = await share.Data.GetBitmapAsync();  
}
```

设置共享目标应用

//对应属性

```
await Dispatcher.RunAsync(CoreDispatcherPriority.Normal, async () =>
{
    TitleBox.Text = title;
    DescriptionBox.Text = description;
    if (text != null)
        UnformattedTextBox.Text = text;
    if (uri != null)
    {
        UriButton.Content = uri.ToString();
        UriButton.NavigateUri = uri;
    }
    if (formattedText != null)
        HTMLTextBox.NavigateToString(HtmlFormatHelper.GetStaticFragment(formatted
Text));
```

设置共享目标应用

```
if (bitmapImage != null)
{
    IRandomAccessStreamW ithContentType bitmapStream = await
this.bitmapImage.OpenReadAsync();
    BitmapImage bi = new BitmapImage();
    bi.SetSource(bitmapStream);
    WholeImage.Source = bi;
    bitmapStream = await this.thumbImage.OpenReadAsync();
    bi = new BitmapImage();
    bi.SetSource(bitmapStream);
    ThumbImage.Source = bi;
}
```

设置共享目标应用

```
if (customData != null)
{
    StringBuilder receivedStrings = newStringBuilder();
    JsonObject customObject = JsonObject.Parse(customData);
    if (customObject.ContainsKey("type"))
    {
        if (customObject["type"].GetString() == "http://schema.org/Person")
        {
            receivedStrings.AppendLine("Type: " +
customObject["type"].Stringify());
            JsonObject properties = customObject["properties"].GetObject();
            receivedStrings.AppendLine("Image: " +
properties["image"].Stringify());
```

设置共享目标应用

```
        receivedStrings.AppendLine("Name: " +  
properties["name"].Stringify());  
        receivedStrings.AppendLine("Affiliation: " +  
properties["affiliation"].Stringify());  
        receivedStrings.AppendLine("Birth Date: " +  
properties["birthDate"].Stringify());  
        receivedStrings.AppendLine("Job Title: " +  
properties["jobTitle"].Stringify());  
        receivedStrings.AppendLine("Nationality: " +  
properties["Nationality"].Stringify());  
        receivedStrings.AppendLine("Gender: " +  
properties["gender"].Stringify());  
    }  
    CustomDataBox.Text = receivedStrings.ToString();  
} } }); }); }
```


教程

- [31 Days of Windows 8](http://31daysofwindows8.com)

<http://31daysofwindows8.com>

- [The Type Hierarchy](http://schema.org/docs/full.html)

<http://schema.org/docs/full.html>

资源

- [App contracts and extensions](http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh464906.aspx)

<http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh464906.aspx>

- [Sharing and exchanging data](http://msdn.microsoft.com/library/windows/apps/xaml/hh871373)

<http://msdn.microsoft.com/library/windows/apps/xaml/hh871373>

- [Quickstart: Sharing content](http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh871368)

<http://msdn.microsoft.com/en-us/library/windows/apps/xaml/hh871368>

- [《Programming Windows Sixth Edition》](http://shop.oreilly.com/product/0790145369079.do)

<http://shop.oreilly.com/product/0790145369079.do>

The image features the Microsoft logo in a white, italicized, sans-serif font, centered on a solid blue background. The logo is surrounded by several white-outlined squares of various sizes, some of which are partially cut off by the edges of the frame, creating a geometric pattern.

Microsoft®