

# Publish Error Detail

---

## Contents

1	Problem Statement .....	2
1.1	Goals, Requirements, Constraints.....	2
2	Solution Overview .....	2
3	Solution Details .....	2
3.1	Definitions.....	2
3.2	Design.....	2
3.2.1	Overview .....	2
3.2.2	PublishPluginXmlManifestFactory Class .....	3
3.2.3	Publish Plugins .....	3
3.2.4	PublishSummaryDialog Class .....	3
3.2.5	PublishSummaryErrorDialog Class .....	3
3.2.6	PublishJob Class .....	3
3.3	Assumptions, Dependencies, and Risks .....	4
3.4	Public API .....	4
3.5	Internal and External Special Values.....	4
3.6	Backward Compatibility .....	4
3.7	Performance .....	4
3.8	Scalability .....	4
3.9	Unit Testing.....	4
3.10	Privacy .....	4
3.11	Internationalization and Localization.....	4
3.12	Accessibility.....	4
3.13	Client-Specific Issues .....	4
3.14	Server-Specific Issues .....	4
3.15	Testable Units .....	4
3.15.1	PublishPluginXmlManifestFactory Testable Unit.....	4
3.15.2	Plugin Testable Unit .....	4
3.15.3	Publish Summary Dialogs Testable Unit .....	5

3.15.4	PublishJob Testable Unit .....	5
4	Open Issues .....	5
5	Future Considerations.....	5
6	Alternate Solutions .....	5
7	Review Feedback .....	5

## 1 Problem Statement

Windows Live Photo Gallery currently has no mechanism to report detail on partial successes and failures in the publish process. That is, if 95 of 100 pictures publish successfully, WLPG will report the number of success and failures but not the names of the pictures that failed to publish.

### 1.1 Goals, Requirements, Constraints

- Determine which items fail during the publish
- Capture the names and error detail for failed items
- Display the names and error detail to the user
- Provide an easy means to reselect the failed items once the query process is implemented in M2.

## 2 Solution Overview

Currently, each plugin detects when it is unable to successfully publish an item. To implement this feature, the plugins will be changed to also store the case of failure and any supplemental failure information into the session XML. At the completion of the publish process, the publish manager will inspect the session XML for failure information and display it to the user.

## 3 Solution Details

### 3.1 Definitions

Term	Definition

### 3.2 Design

#### 3.2.1 Overview

Each plugin will store failure information into the session XML when it is unable to successfully create an item. After the publish process has completed, the publish manager will inspect the session XML to see if any failures occurred. If so, the publish manager will display the failure information in the publish summary dialog.

### 3.2.2 PublishPluginXmlManifestFactory Class

- A new method will be added to the PublishPluginXmlManifestFactory class: `SetItemProperty()`. This new method represents the inverse of the existing `GetItemPropertyAs*()` methods.

### 3.2.3 Publish Plugins

- In the `PublishItem()` method, if the plugin is unable to create the item successfully, it must add a node to the session XML document via the new `SetItemProperty()` method. This node must be named “publishErrorDetail” and must contain the information to display to the user in the publish summary dialog.

### 3.2.4 PublishSummaryDialog Class

- The PublishSummaryDialog dialog must be changed to include a ListView control. This control must be hidden if there are no items in it. This control must display a maximum of five items.
- A new method, `AddErrorDetail()`, will be created. This method will accept a filename and error detail, and will add an item to the ListView control.

### 3.2.5 PublishSummaryErrorDialog Class

- The PublishSummaryErrorDialog dialog must be changed to include a ListView control. This control must be hidden if there are no items in it. This control must display a maximum of five items.
- A new method, `AddErrorDetail()`, will be created. This method will accept a filename and error detail, and will add an item to the ListView control.

### 3.2.6 PublishJob Class

- In the `PublishJob::OnJobCompleted()` method, the PublishJob class currently chooses to display either the publish summary or the publish error summary dialogs. This method must additionally retrieve error detail information from the session XML and pass it to the `AddErrorDetail()` method of the appropriate dialog.

### 3.3 Assumptions, Dependencies, and Risks

### 3.4 Public API

### 3.5 Internal and External Special Values

### 3.6 Backward Compatibility

### 3.7 Performance

### 3.8 Scalability

### 3.9 Unit Testing

### 3.10 Privacy

### 3.11 Internationalization and Localization

### 3.12 Accessibility

### 3.13 Client-Specific Issues

### 3.14 Server-Specific Issues

### 3.15 Testable Units

#### 3.15.1 PublishPluginXmlManifestFactory Testable Unit

##### 3.15.1.1 Work Items

Description	Days	Pri	Dev
Add SetItemProperty method		1	

##### 3.15.1.2 Dependencies

##### 3.15.1.3 Exit requirements

#### 3.15.2 Plugin Testable Unit

##### 3.15.2.1 Work Items

Description	Days	Pri	Dev
Add error detail in LiveProvider		1	
Add error detail in SoapboxPlugin		2	

### *3.15.2.2 Dependencies*

### *3.15.2.3 Exit requirements*

## **3.15.3 Publish Summary Dialogs Testable Unit**

### *3.15.3.1 Work Items*

Description	Days	Pri	Dev
Add ListView control to PublishSummaryDialog		1	
Add ListView control to PublishSummaryErrorDialog		1	

### *3.15.3.2 Dependencies*

### *3.15.3.3 Exit requirements*

## **3.15.4 PublishJob Testable Unit**

### *3.15.4.1 Work Items*

Description	Days	Pri	Dev
Retrieve error information from session XML and pass to publish summary dialogs		1	

### *3.15.4.2 Dependencies*

### *3.15.4.3 Exit requirements*

## **4 Open Issues**

## **5 Future Considerations**

## **6 Alternate Solutions**

## **7 Review Feedback**