



Process Frontend Custom Error Messages Developers Guide

Owner: Resultmaker Research & Development
Version: 6.0.0
Revision: 2013-11-27

Lars Pedersen, Charles Symons

Contents

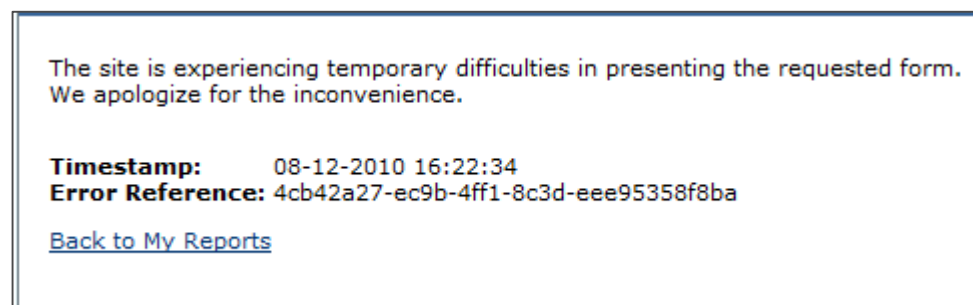
1	Introduction	3
1.1	The problem	3
1.2	The solution	3
1.3	Purpose	4
2	Schema for the Custom Error Message xml	4
2.1	Custom errors schema default namespace.....	4
2.2	Custom errors schema xml inside	4
3	Creating a Custom errors message	5
3.1	Creating a custom error class.....	5
3.1.1	Create derived custom error class	5
3.1.2	Catching and displaying the message	5
3.2	Xml definition for the translation file.....	5
3.2.1	Errors.....	5
3.2.2	Error	5
3.3	Xml definition for the custom errors file.....	6
3.3.1	ErrorMsgTranslations	6
3.3.2	Errors.....	6
3.3.3	Error	6
3.4	Specifying the language for custom errors files.....	6
3.5	Setting the error message language	6
3.6	Error Message Mode.....	6
3.7	Error message default usage.....	7
4	Location of custom errors file	7

1 Introduction

The Process Frontend Custom Error Messages is functionality that enables the Process Consultant or System Developer to provide human readable translations in the Process Frontend for the technical exceptions that could occur in the Process Engine.

1.1 The problem

Without a translation of the error, an exception that is not translated will end up in the Process Frontend, looking like this:



This does not explain to the end user what is wrong.

The experienced user or support staff could use the CTRL + SHIFT + Left Click to reveal the actual exception.



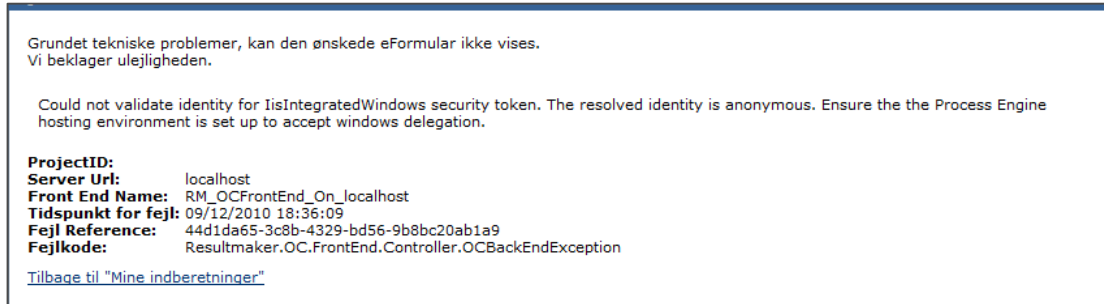
This information is useful for the support staff or System Developer, but not for the end user.

1.2 The solution

Process Frontend Custom Error Messages allows for translations of the technical errors into human readable messages.

This is done by specifying translations in a Custom Error Messages file. This is an xml file, which will allow the error messages to be displayed in a nicer way and according to the language that the Frontend is configured to run with.

The exception could look like this after the translation.



1.3 Purpose

The purpose of this document is to provide a guideline to the Process Consultant and the System Developer on how the Custom Error Messages work and how to use, configure and deploy the functionality.

2 Schema for the Custom Error Message xml

In order to technically communicate the format of the Custom Error Message xml file, a schema has been created. This schema is described in this section.

2.1 Custom errors schema default namespace

The default custom error schema name space is:

<http://www.resultmaker.com/schemas/CustomErrors/20101006>

2.2 Custom errors schema xml inside

```
<?xml version='1.0' encoding='utf-8'?>
<xs:schema attributeFormDefault="unqualified"
            elementFormDefault="qualified"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.resultmaker.com/schemas/CustomErrors/20101006">
  <xs:element name="ErrorMsgTranslations">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Errors">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="Error" >
                <xs:complexType>
                  <xs:attribute name="errorType" type="xs:string" use="required" />
                  <xs:attribute name="value" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Note: The attribute for the error message which is `errorType` and `Value` are required information so that the error can map and display the error message properly.

3 Creating a Custom errors message

This section describes how to create a custom errors message. Custom error messages are located in to different files. In the `Translations.[Language].xml` file, that comes with the Process Platform and in the `CustomError.[Language].xml` file that is for Resultmaker partners to use. See section “3.4 Specifying the language for custom errors” for the format of the `[Language]` string.

3.1 Creating a custom error class

The way in which the custom error exception messages work is by recognizing the unique custom exception of the classes that are derive from `Exception` class. Not only that the key which is used to identify error message to used is consist of namespace and error class

3.1.1 Create derived custom error class

A custom error class is needed to be thrown in order to display a custom error message and these custom error class must inherit from `Exception` as a minimum inherit class. For example:

```
namespace PPTest60_Scripts
{
    public class CustomException : Exception
    {
    }
}
```

Note: The name space must included as part of the `errorType` that is used for the key value. In this scenario the namespace for `CustomException` is `PPTest60_Scripts` so the combination which will be used as key is `PPTest60_Scripts.CustomException`.

3.1.2 Catching and displaying the message

The message is display and looked up from either one of these source files:

`CustomError.[Language].xml` and `Translations.[Language].xml`. If both have the same `errorType` key then the default file that the system will used to read its error messages is the `CustomError`.

3.2 Xml definition for the translation file

```
<Translations Language="English" >
  <Errors>
    <Error errorType="PPTest60_Scripts.CustomException" value="This is a
custom error exceptions."/>
  </Errors>
</Translations>
```

The root level for the xml translation file is `Translations`.

3.2.1 Errors

This element contains a list of `Error` elements, which will be used to display the error messages.

3.2.2 Error

This element is the lowest level where the error message will be compared and used.

The “errorType” attribute defines the key to determine which error text should be displayed when a specific error type of exception is thrown.

The “value” attribute will represent the text that will be display in the Frontend as an error message.

3.3 Xml definition for the custom errors file

```
<ce:ErrorMsgTranslations
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ce="http://www.resultmaker.com/schemas/CustomErrors/20101006">
  <ce:Errors>
    <ce:Error errorType=" PPTest60_Scripts.CustomException" value="
This is a custom error exception."/>
  </ce:ErrorMsgTranslations>
```

The name space xmlns:ce=http://www.resultmaker.com/schemas/CustomErrors/20101006 is used for validating when an existing customerror schema is used.

3.3.1 ErrorMsgTranslations

This element in the xml is the root level of the custom errors message file.

3.3.2 Errors

This element contains a list of Error elements, which will be used to display the error texts.

3.3.3 Error

This element is the lowest level where the error message will be compared and used.

The “errorType” attribute defines the key to determine which error text should be displayed when a specific error type of exception is thrown.

The “value” attribute will represent the text that will be display in the frontend as an error message.

3.4 Specifying the language for custom errors files

The name of the custom errors file always has to be “CustomErrors”. For specifying different languages, this can be done by suffixing the language identifier (i.e. en) to the “CustomErrors” file name.

For example an English version of CustomErrors has to be saved to the file called CustomErrors.en.xml, and the Danish one will be CustomErrors.da.xml and so on.

3.5 Setting the error message language

The language for the Custom Errors file is controlled by setting the key attribute of the Language setting in the Resultmaker.OC.FrontEnd.config file. Here is an example:

```
<configuration>
  <Resultmaker.OC.FrontEnd>
    <add key="Language" value="en" />
  </Resultmaker.OC.FrontEnd>
</configuration>
```

3.6 Error Message Mode

There are three ways to present error messages in the Resultmaker Process Frontend application. Each one presents different amounts of information. The is specified in the

Resultmaker.OC.Frontend.config file in the ErrorMessageDisplayMode key in the AppSettings section.

An example of the configuration of the error message mode is shown below:

```
<add key="ErrorMessageDisplayMode" value="detail"/>
```

There are three type of error message modes and they are *minimal*, *hidden* and *detail*.

ErrorMessageDisplayMode	Description
minimal	In this mode the minimal amount of error information is displayed to the user. Only the custom error message is displayed.
hidden	This displays the normal custom errors message and by pressing CTRL+SHIFT and clicking on the custom errors message, the usual error details will be displayed.
detail	This is display both custom error and detail error to the frontend where an error is occurred.

Note that the values must be specified in small caps.

3.7 Error message default usage

By default, the errors message that is displayed is taken from the customErrors.xml file. If the customErrors file doesn't exist, the error message will use the Translations.xml file to display the errors. If no translations exist for the error type, the normal exception is displayed.

4 Location of custom errors file

The custom errors file will be located at the translation path specified in the Resultmaker.OC.Frontend.config file.

```
<add key="TranslationsPath" value="Layouts/Presentation/Translations" />
```