OPENTEXT™

OpenText Professional Service Sample Application

Trademarks

OpenText is a registered trademark of OpenText Corporation. All other company names, brand names, and product names are the property and/or trademarks of their respective companies.

Copyright Notice

http://www.opentext.com

Contact Information

Open Text Corporation
275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1
Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544
International: +800-4996-5440
Fax: +1-519-888-0677
Email: IX-PS-NA@OpenText.com
FTP: ftp://ftp.opentext.com
Knowledge Center: https://knowledge.opentext.com
For more information, visit http://www.opentext.com

Notice

The material contained in this document represents proprietary, confidential information pertaining to OpenText products, software and services. By accepting delivery of this document you hereby agree that the information in this document shall not be disclosed outside of your organization and shall not be duplicated or disclosed without prior written consent from OpenText.

Synopsis

This document describes a simple request the OpenText Professional Services team might receive from a customer. Given the request, design and develop a solution that meets the customer's requirements.

Customer Requirements

The customer has a back-office system that generates pairs of files, an XML "control" file and a PDF "payload" file. When an XML and PDF file pair are detected, the solution should read and verify the XML file and perform several actions.

A sample XML file would contain the following sample content:

```xml
<?xml version="1.0" encoding="utf-8"?>
<content>
  <sender>sender@domain.com</sender>
  <recipients>
    <recipient>recipient01@domain.com</recipient>
    <recipient>recipient02@domain.com</recipient>
    <recipient>recipient03@domain.com</recipient>
  </recipients>
  <subject>This is the message subject.</subject>
  <messagebody>This is the message body.</messagebody>
</content>
```

The schema for the XML file is:

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="content">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="sender" type="xs:string" />
        <xs:element name="recipients">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="recipient" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="subject" type="xs:string" />
        <xs:element name="messagebody" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

The actions to perform on the XML and payload file pair are:

- Monitor the network folder where the XML and payload files are written to.
    - Allow the XML file to age a minimum of one minute before processing the XML file to ensure time for the PDF file to be created and/or written to the network folder.
    - When a new XML file is found and has aged the appropriate amount of time, validate that there is also a corresponding payload file with the same base file name. For example, “12345.xml” and “12345.pdf”.
    - If after one minute the XML file does not have a corresponding PDF payload file, move the XML file to an Error folder.
    - If the XML file does not match the specified schema, move the XML and PDF files to the Error folder.
- Once the XML and payload files are detected and determined to be valid, send an SMTP e-mail with the following parameters:
    - The SMTP sender address is the value from the XML `<sender>` node.
    - The SMTP recipients are the addresses from the XML `<recipients>` node.
    - The SMTP subject line is the value from the XML `<subject>` node.
    - The SMTP message body is the value from the XML `<messagebody>` node.
    - Add the PDF file as an attachment to the e-mail.
    - Once the SMTP e-mail has been sent, move the XML and PDF files to an Archive folder.
- Optional: Store an audit record in an SQL database (this can be MS SQL Server, MS SQL Server Express, or MS SQL Compact Edition (CE)) table.
    - The audit table should have the following columns (at a minimum):
        - Unique ID
        - Date/Time of entry
        - Sender address
        - Recipient address(es)
- Miscellaneous requirements
    - The monitored network folder, the Error folder, and the Archive folder should all be configurable.
    - Development language must be C#.

Application Design

Optionally provide a 2 to 3-page document detailing your application design. This type of document is called, not surprisingly, a design document and would be sent to the customer for them to review and approve before a custom solution project would enter the development phase.

Miscellaneous

There is no right or wrong way to build this type of solution. The purpose of this exercise is to gauge your understanding of the requirements and to examine how you would go about interpreting those requirements and generating the literature and software to support the customer's request.