Correspondence Editor Tool Guide

Version 1.3

06/08/2020

Contents

Document Control

Document Information	
Document Id	SPP-TECH-GUIDE-CORR-EDIT-TOOL-001
Document Owner	Sagitec Framework Team
Issue Date	26-Oct-18
File Name	SPP Technical Guide Correspondence Editor Tool

Version History

Version	Implemented By	Revision Date	Description of Change
1.0	Amol Kalekar	06/22/18	Initial Version
1.1	Sameer Pise	10/26/18	Added Data Flow
1.2	Sameer Dabir	11/14/2019	Added Instructions for Certificate installation
1.3	Sameer Dabir	06/08/2020	Corrected config file name

Introduction

This document provides a technical information about correspondence editor tool. It provides details about its architecture, installation steps for end users of the tool.

The correspondence Editor tool is a service that runs at the end user which enables them to view, edit, print and save the generated correspondences from the NeoSpin line of business application.

Audience

This document is prepared for an understanding of the application development teams, Infrastructure team and the end users

- Application Developers
- NeoSpin User
- Infrastructure team

Architecture Diagram

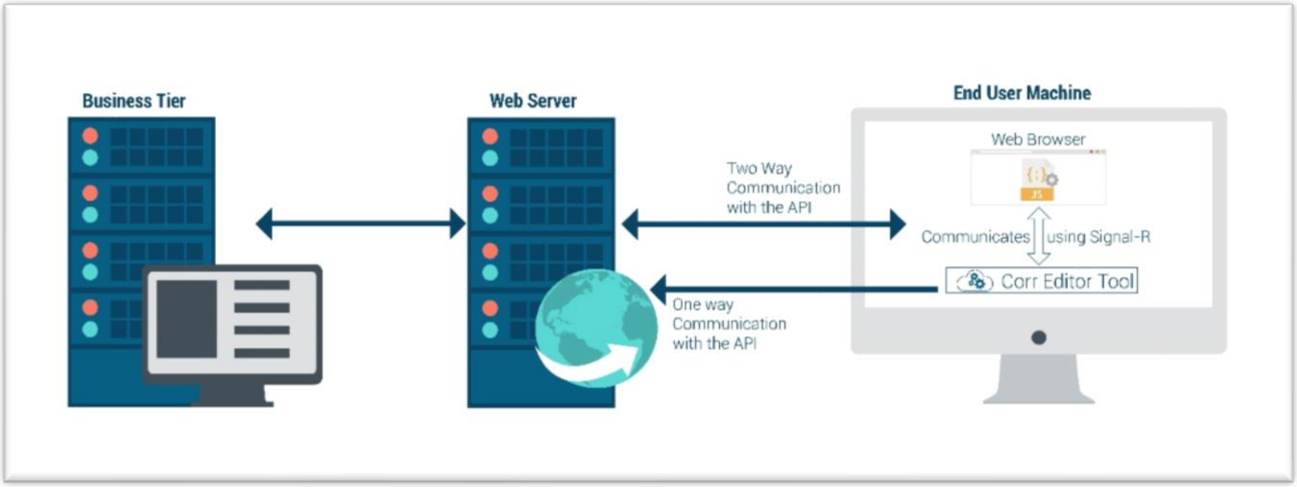

FIGURE 1

The Correspondence editor tool is service installed on the end user's machine.

This tool runs as a windows service using default ports (8082 for https and 8081 of HTTP). The ports can be configured based on the system requirements

The tool exchanges the correspondence data by interacting with the Browser using Signal-R (JS) to receive the generated correspondence data. The data is received in a Base64 format.

Advantages

- ✓ Users don't need to have any access to the folder where the correspondence is generated. This is a security risk, as users have access to this folder can modify any other letters (even generated by others) directly using the word.
- ✓ Browser Independent: Old Correspondence edit functionality (ActiveX control) would work only if they used Internet Explorer, with this approach they can use any browser.

Installation Steps

The below steps are to be used by Infrastructure Support staff to ensure the tool is installed correctly at each end user systems.

Tool Installation

Identify the Sagitec Framework version used by NeoSpin Application

- Get the correct version of SagitecWinCorrEditor.msi compatible with Sagitec Framework version
- Run the SagitecWinCorrEditor.msi file
- Update Custom configuration provided in CustomConfig.xml if override required to the default installation (Refer the Configuration properties in below table
- Run SagitecWinCorrEditor.msi to install PFX certificate

CustomSettings.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <appSettings>
    <add key="HTTPPORT" value="8081" />
    <add key="HTTPSPORT" value="8082" />
      <add key="SilentInstallation" value="false" />
      <add key="SkipCertificateInstallation" value="false" />
      <add key="CertificateLocation" value="" />
      <add key="CertificateHash" value="" />
      <add key="CertificatePassword" value="" />
      <add key="AddToTrustedRoot" value="true" />
      <add key="UseExistingCertificate" value="true" />
  </appSettings>
</configuration>
```

Field Name	Data Type	Description	Default Value
HTTPPORT	string	The port number for HTTP communication between client browser and the tool	8081
HTTPSPORT	string	The port number for HTTPS communication between client browser and the tool	8082
SilentInstallation	Boolean	Set the property as true to disable any alerts or pop-ups during installation	false
SkipCertificateInstallation	Boolean	Set the property as true to skip certificate installation for HTTPS port.	False
UseExistingCertificate	Boolean	Set the property as true for installation of self-signed certificate provided by Sagitec. Set the property as false for installation of certificate provided by client	True
AddToTrustedRoot	Boolean	Set the property as true to add certificate in trusted root repository.	true
CertificateLocation	string	Set the full directory path of client specific certificate. The path must include the certificate name with extension. This property is required when UseExistingCertificate is false	
CertificateHash	string	Set Hash or Thumbprint of the client specific certificate. This property is required when UseExistingCertificate is false	
CertificatePassword	string	Password of certificate the client specific certificate. This property is required when UseExistingCertificate is false	

Data Flow

On Click of Edit button

User Action	The user has generated the correspondence and wants to view or edit the specific context. He clicks an action button on the Line of Business application.
Systems Involved	Client-side Web browser Application Web Server Correspondence Editor Tool
Action Step Details	1) On clicking the edit button on Web Browser, a Client-side JavaScript fires an AJAX request to the Application Web Server to edit a generated Correspondence.

	2) Application Web Server validates the AJAX Request to ensure the authorized user has requested the information. The security checks are performed on the IP address, valid Session ID and authorized userid. 3) Post validation the web Server gives a Web API call to the Application Business Tier Server for getting the specific Correspondence document 4) The Application Server picks the requested correspondence file, converts it into a Base64 format string and sends to the Web Server along Additional Information required for Correspondence Editor Tool (for e.g. Custom Button List, Message details, Actions post button click) 5) The Client Web browser passes the Base64 data and Additional information to Correspondence Editor Tool via Signal-R interface 6) The Correspondence Editor Tool receives the Base64 data and creates a local file at the client machine. 7) The Correspondence Editor Tool opens a new window to shows the local file and to the end User

On Save Button Click

User Action	The user has opened the generated correspondence to edit the correspondence in the Correspondence Editor tool. The user chooses to save the document by clicking the Save button on the Correspondence Editor tool.
Systems Involved	• Correspondence Editor Tool • Application Business Tier Server
Action Step Details	1) The Correspondence Tool save the file changes in the local file version 2) It creates a Base64 string data and makes a Web API call (Update Correspondence Status) to Web Server for updating Correspondence document on server 3) e. On Success Correspondence Editor tool displays the configured message
Web API Method	Web API Method Name : UpdateCorrespondenceStatus Business Tier method: UpdateGeneratedCorrespondence Data Passed: (Formed, KeyField, TemplateName, FileName)

On Print Button Click

User Action	The user has opened the generated correspondence in the Correspondence Editor tool and chooses to save the document by clicking the Save button on the Correspondence Editor tool.
Systems Involved	• Correspondence Editor Tool • Application Business Tier Server
Action Step Details	1) The Correspondence Tool invokes the Print functionality within Word to print the local file version 2) Makes a Web API call to Web Server to update Correspondence Status

Web API Method	`Web API Method Name: UpdateCorrespondenceStatus`
	`Business Tier method: UpdateGeneratedCorrespondence`
	`Data Passed: (Formed, KeyField, TemplateName, FileName)`

On Custom Button Click

User Action	The user has opened the generated correspondence in the Correspondence Editor tool and chooses to perform a custom action determined by Line of Business application
Systems Involved	• Correspondence Editor Tool • Application Business Tier Server
Action Step Details	1) The Correspondence Tool Makes a Web API (CorrToolCustomButtom) 2) This Solution implements the Web API call and performs the required action at server side 3) On Success Correspondence Editor tool displays the configured message

Web API Method	`Web API Method Name: CorrToolCustomButtonClick`
	`Business Tier method: CorrToolCustomButton`

On Close Button Click

User Action	The user chooses to close the correspondence
Systems Involved	Correspondence Editor Tool
Action Step Details	1) The local copy of the file is deleted 2) The Correspondence Editor window is closed

Interface Data Fields

Below are the data fields which are exchanged between Correspondence Editor and Application Web Tier

Data Type	Field Name	Description
string	ServerFilePath	Server Path location of generated correspondence
string	WebAPIURL	Web API URL to be invoked

string	FormID	Form ID where the Correspondence is edited
string	KeyField	Primary Key of Business Object
string	TemplateName	Name of Template
string	UserID	Client End user ID
string	MainSessionID	Session ID
string	PrinterName	PrinterName to which the document will be printed

Security Considerations

1) The secure certificate is required to be installed at the end user machine, so that the tool can work
2) The self-signed certificates are restricted in the Newer version of browser (esp. Mozilla), so use of client specific certificates is recommended.
3) Use of secure https ports for communication between Signal-R Hub on the local machine and Correspondence Editor Tool
4) Only encrypted Data is sent from Correspondence Editor Tool to Application Tier.
5) The data contains the IP and userid which is encrypted using Session ID as a key
6) The Web Server validates the request by decrypting the IP and user-id to uniquely identify the request has come for a valid client
7) Same Web browser Session ID is maintained in Correspondence Editor. In case the Session-Expires the Web API request from Correspondence will be rejected by the Application Server.