# UPP WebSocket Test Tool

Contents

[UPP WebSocket Test Tool 1](#_Toc2588676)

[Introduction 1](#_Toc2588677)

[Important usage notes 1](#_Toc2588678)

[Connection configuration 1](#_Toc2588679)

[Actions 1](#_Toc2588680)

[Form Entry 1](#_Toc2588681)

[Form Update 1](#_Toc2588682)

[File 1](#_Toc2588683)

[Line Display 1](#_Toc2588684)

[Device 1](#_Toc2588685)

[Payment 1](#_Toc2588686)

[Configuration 1](#_Toc2588687)

[Security 1](#_Toc2588688)

[Printing 1](#_Toc2588689)

[Audio 1](#_Toc2588690)

[JSON message tracer 1](#_Toc2588691)

## Introduction

The **U**PP **W**ebSocket **T**est **T**ool (**UWTT**) is a browser based application that interfaces with Tetra Devices operating the iConnect WS application using UPP plugin. UWTT is like RBATT which uses WebSockets and iConnect-WS UPP JSON API.

It allows for predefined action set to be executed. The UWTT can be used as a testing and developer tool to test transactions.

The UWTT operates using Chrome or Firefox and can communicate to the Tetra device using Ethernet.

UWTT allows you to test communications between the Point of Sale (POS) and the PIN pad device without having to run live transactions. It enables you to send messages to the PIN pad device and simulate credit or debit transactions. When running the simulation, a bank response to the terminal can be forced by selecting the response you would like to test (e.g., "Approved," "Declined").

## Important usage notes

A local webserver is required for running the UWTT. Execute run.exe to start the local server on port 8080. User can also execute run.exe from the command prompt after changing directory into UWTT folder to change the default port.

run.exe --port=X

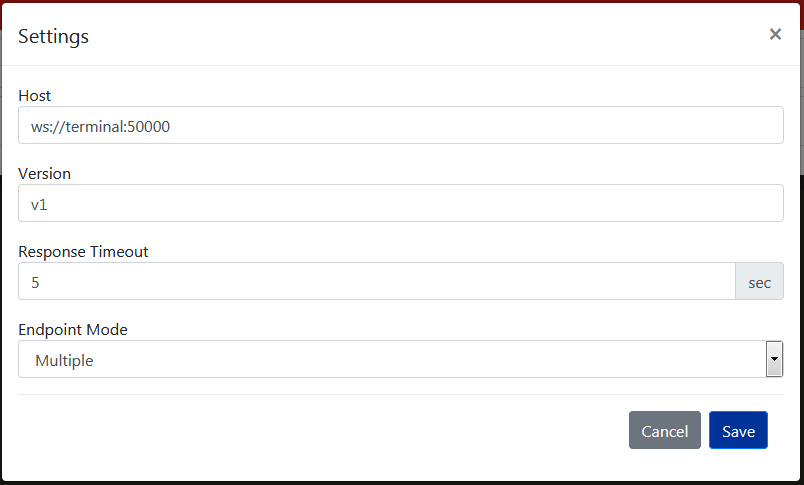
IE is not supported.

## Connection configuration

In the upper right corner UWTT has a Settings link where the user is able to set connection settings.

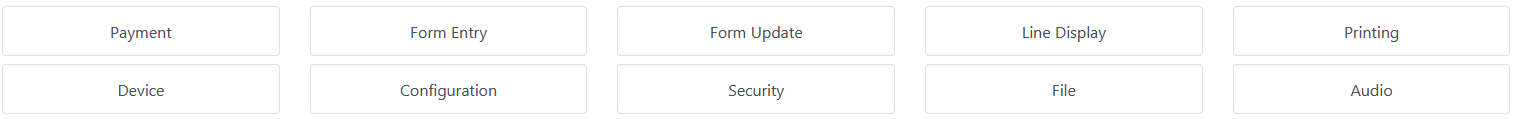


Settings contain the host address and port of the iConnect-WS terminal, the UPP plugin version, the response timeout and endpoint mode. Response timeout is the time gap in which the **response** to a **request** should be delivered and also the gap in which the **event\_ack** to an **event** should be delivered. To connect to host with SSL *wss* schemeshould be used instead of *ws* and CA certificate (the same CA with which the terminal server certificate is signed) should be installed on the PC in the trusted root store. Endpoint mode can be single or multiple. The single mode is not supported by ‘file’ endpoint, therefore multiple mode is used in this case.



## Actions

Actions are user friendly commands for iConnect-WS to make a transaction, show form, upload file, etc. without any knowledge of iConnect-WS UPP JSON API.

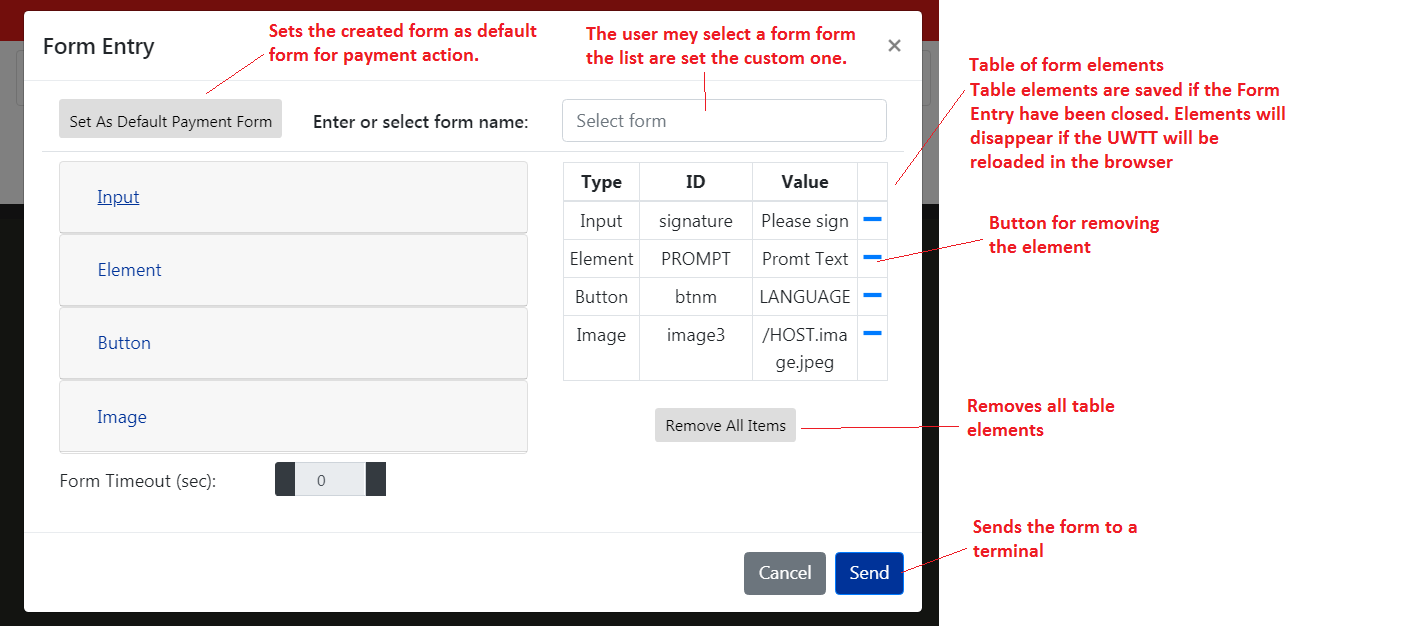


### Form Entry

Form Entry action is the form building tool. It have common part for the whole form as well as separate parts for specific form elements.

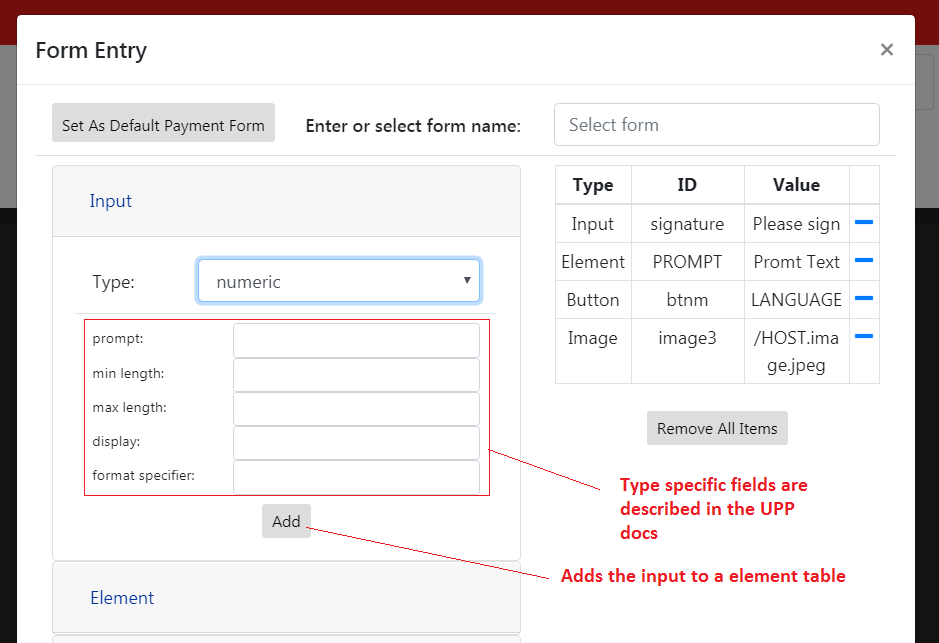
#### Common

Common Form Entry elements are described in the image below.



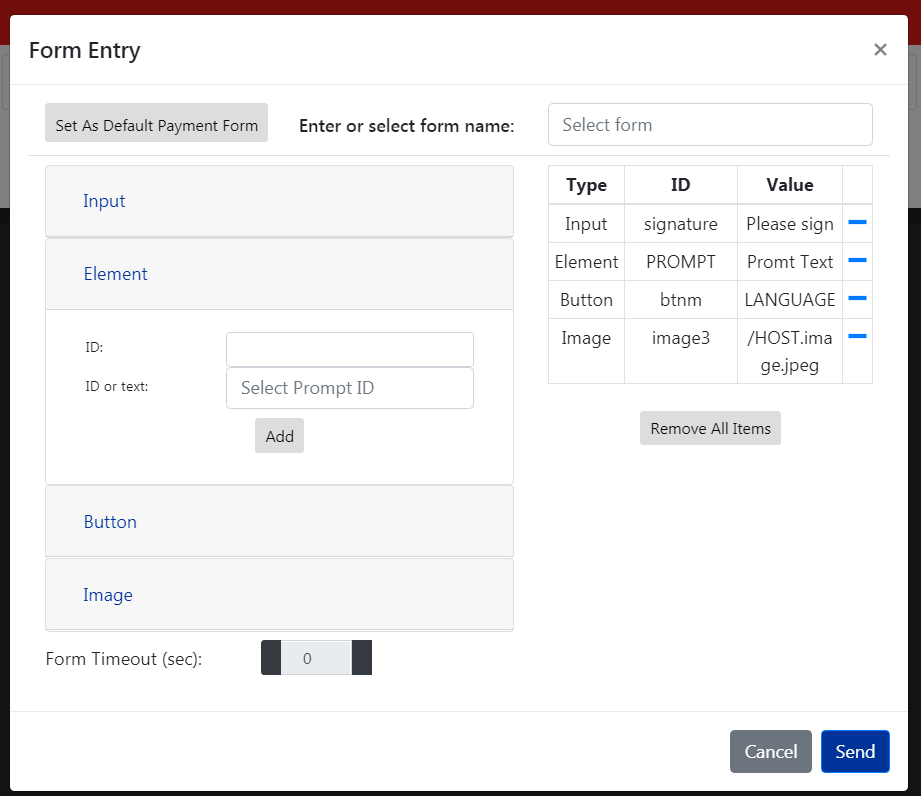
#### Input

Form Input has several types: numeric, alphanumeric, signature, pin and menu. Each type has a subset of elements. Subset description is available in the UPP docs.

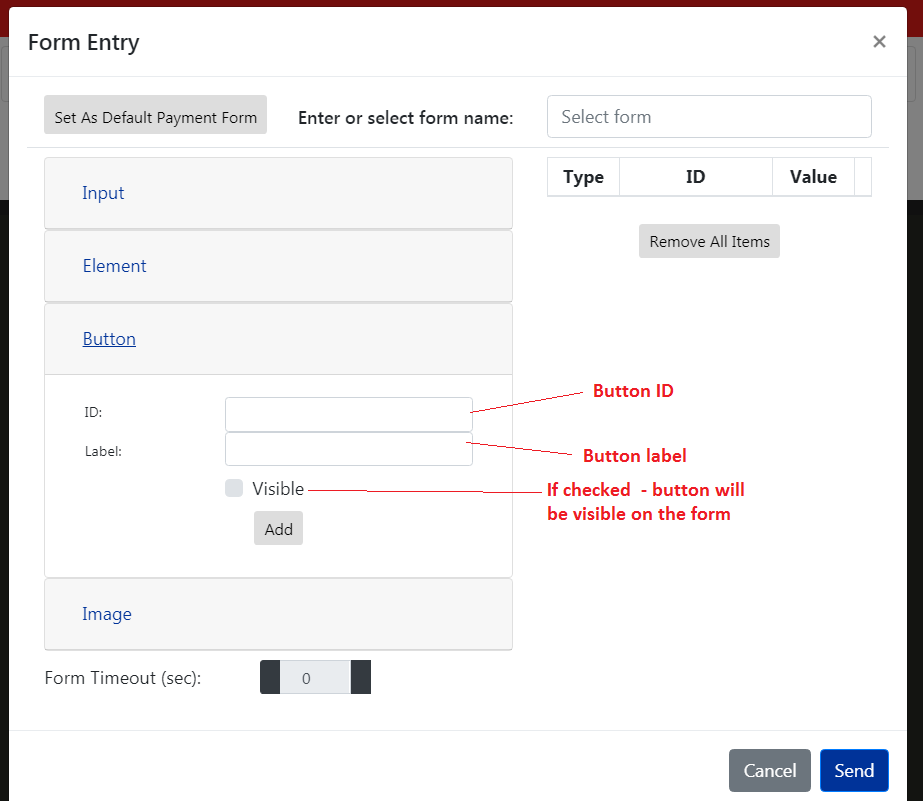


#### Element

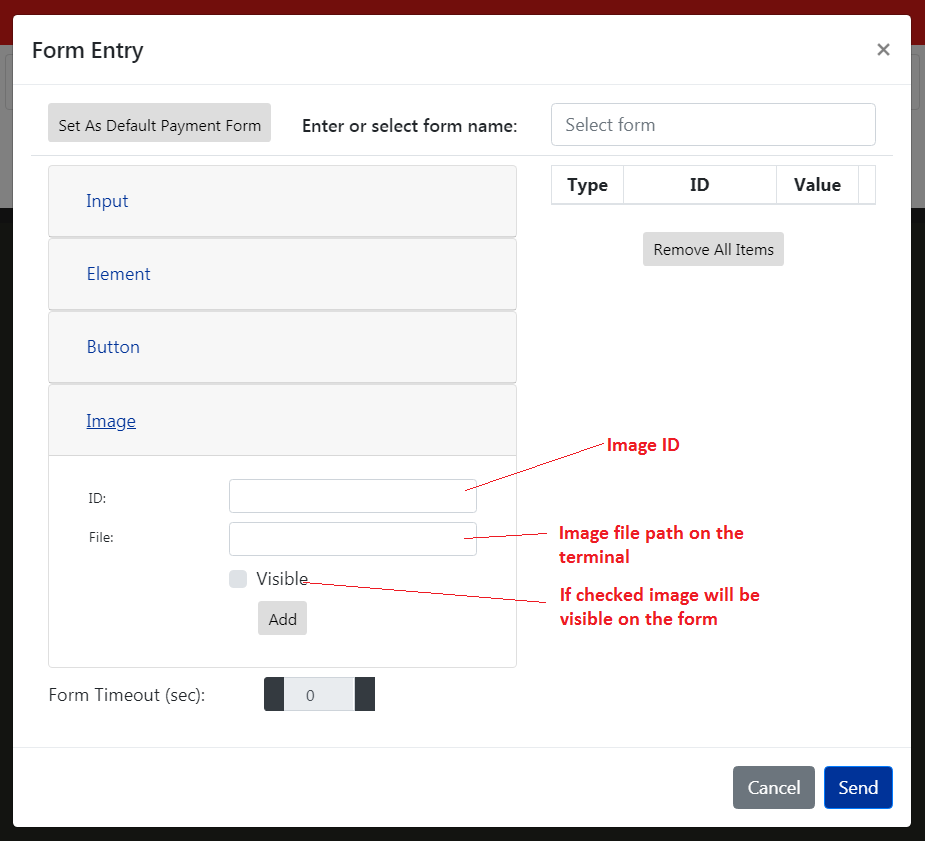
Element is form’s variable. *ID* is form variable name and *ID or text* is its value.



#### Button

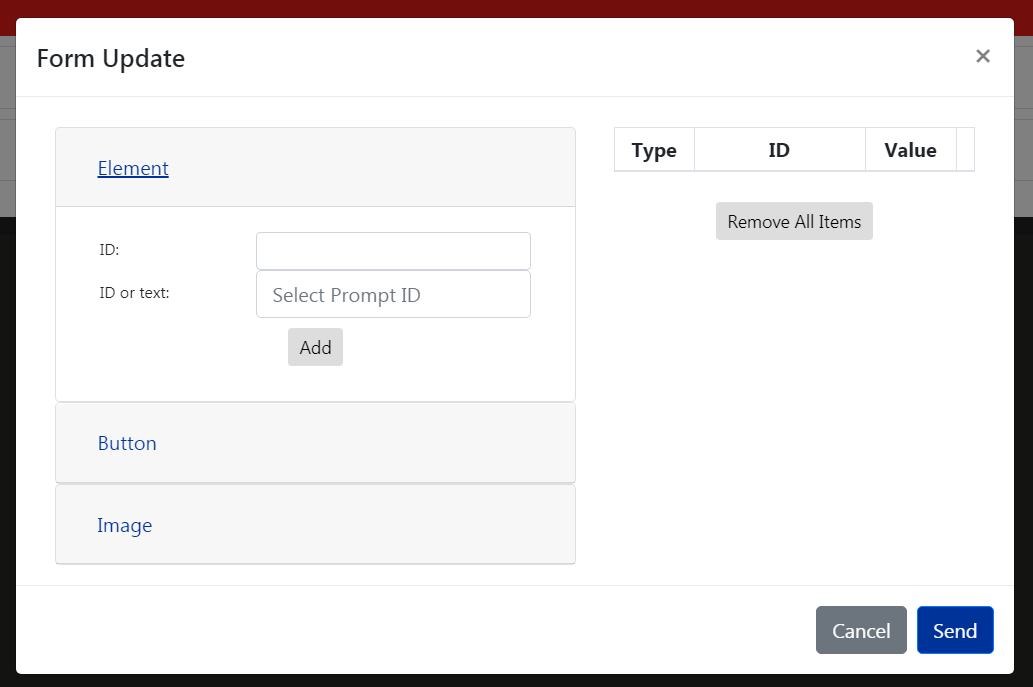


#### Image



### Form Update

Action is used to update the current form. Form elements are exactly the same as if [Form Entry](#_Form_Entry) Action except it doesn’t have i**nput, form name and form timeout**.



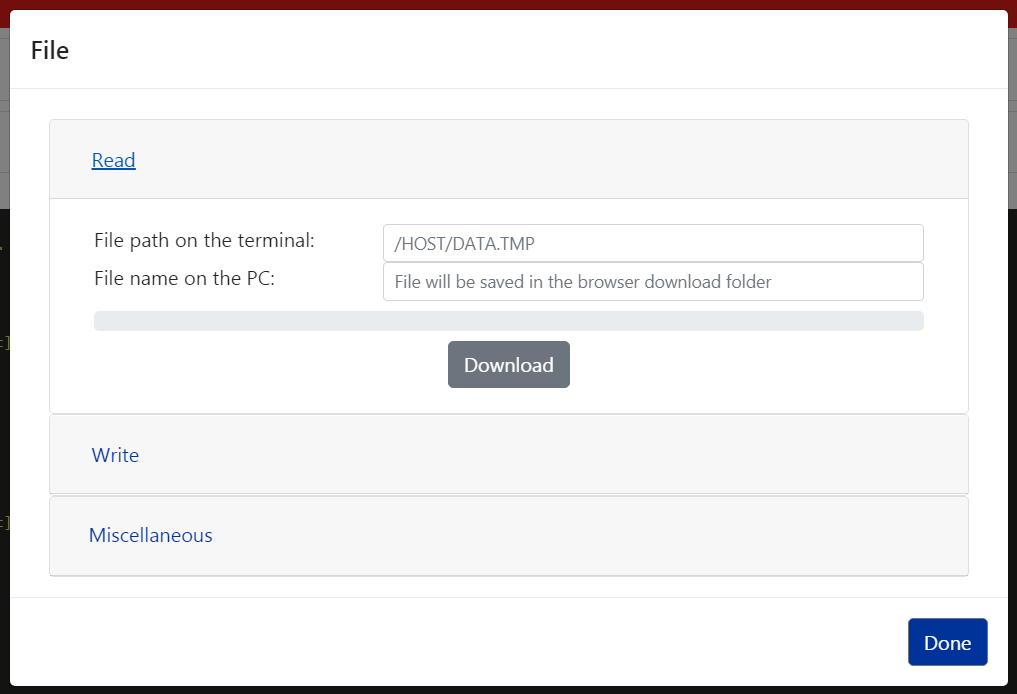
### File

Action is used to **upload**, **download**, **remove**, **find** and **install** the file on/from to the terminal.

#### Read

File read card is for setting download parameters to be able to download the file from the terminal to the PC.

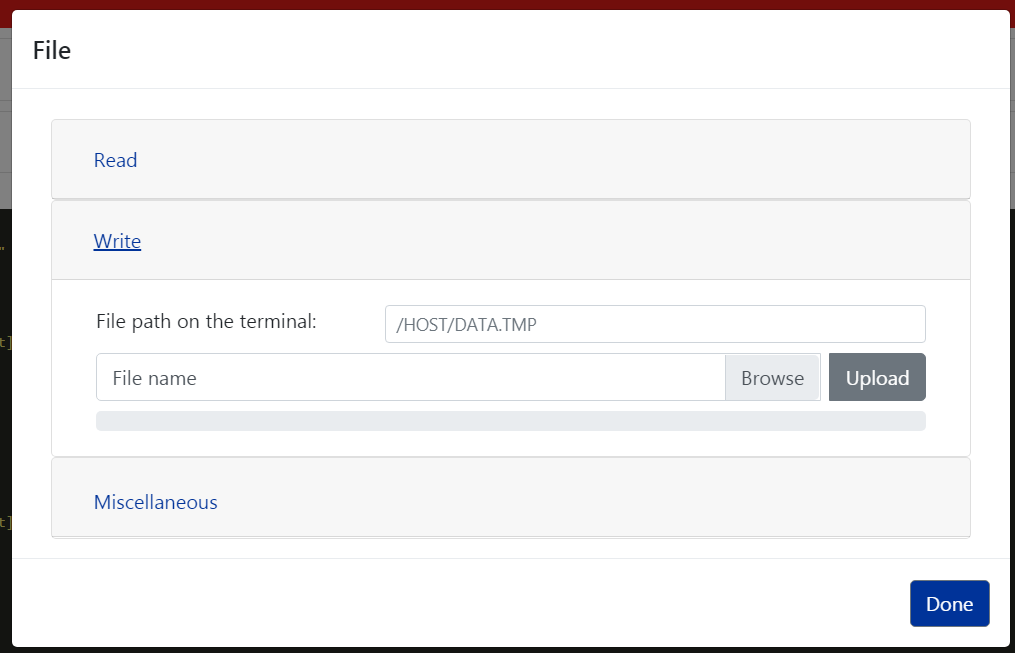
* **File path on the terminal** is a file path for the file to be downloaded.
* **File name on the PC** is the file name in the browser’s default download folder where the downloaded file will be saved.



#### Write

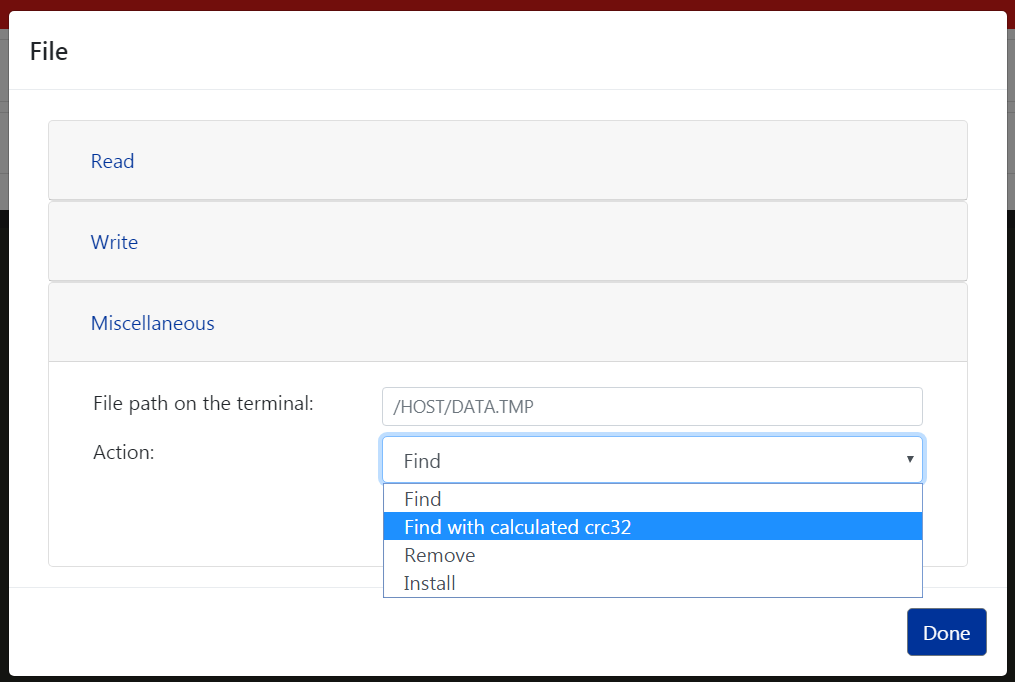
Allows the user to upload the files to the terminal.

* **File path on the terminal** is the path on the terminal where the uploaded file will be saved.
* **File name** lets the user to pick up the file form PC to upload



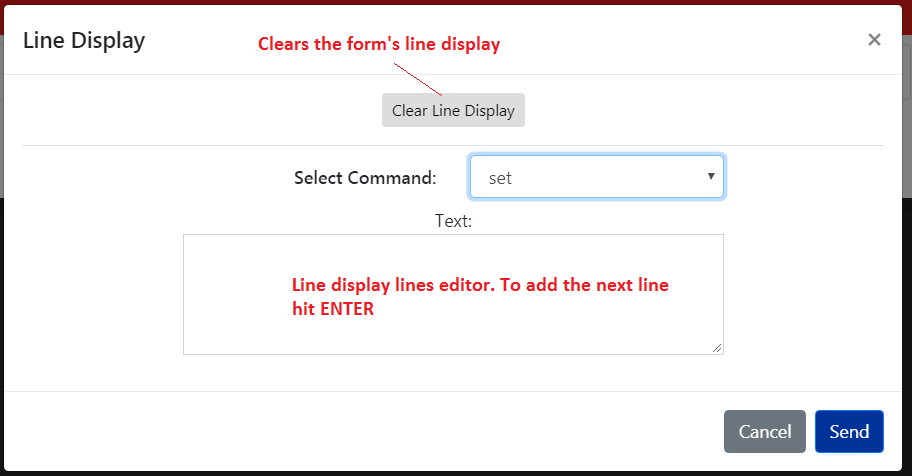
#### Miscellaneous

This section lets the user to send **find**, **find with the calculated crc32**, **remove** and **install** request for the file set in the **File path on the terminal.**



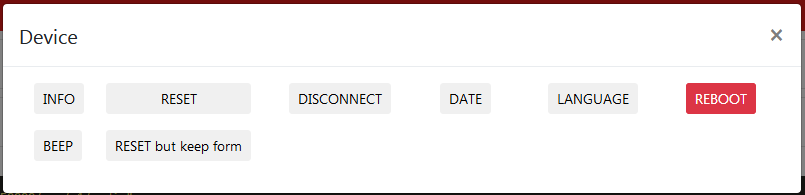
### Line Display

Clears, Sets, Adds the lines and set the Total line to the line display



### Device

* INFO – to get the information about the terminal ( SW versions, etc );
* RESET – sets the terminal in to idle state;
* DISCONNECT – disconnects the iConnect-WS UPP plugin form the UPP application;
* DATE – sets/gets date and time to/from the terminal;
* LANGUAGE – sets/gets default and current languages to/from the terminal;
* BEEP – allows to produce beep on the terminal with tone and duration;
* REBOOT – reboots the terminal.



### Payment

Payment can be started w/ or w/o the form. To set up the payment form please see [common](#_Common) part of the Form Entry description.

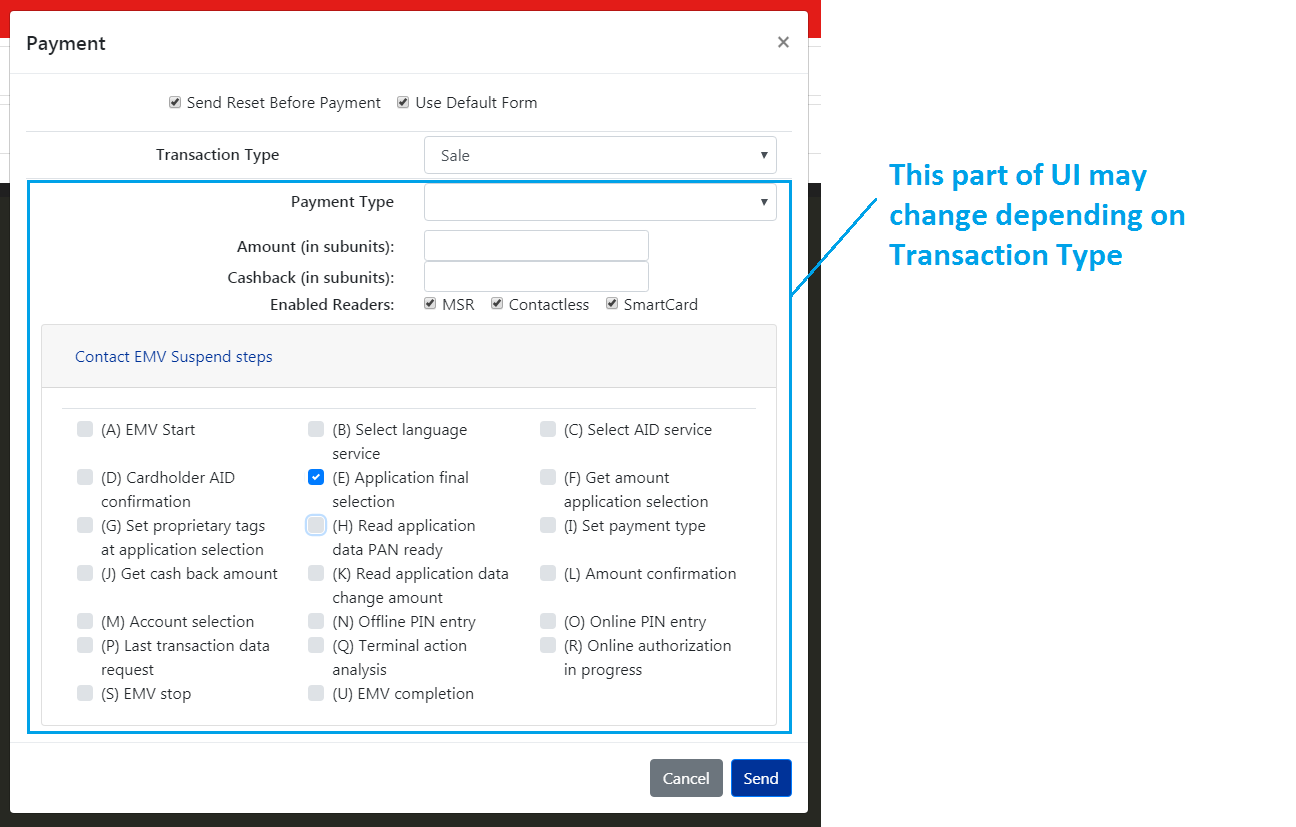
The first step is payment initiation by selecting **Transaction Type**. Depending on transaction type some UI components may change.

Before running the transaction user may choose to send reset (idle) with **Send Reset Before Payment** and to send previously defined form in Form Entry action.

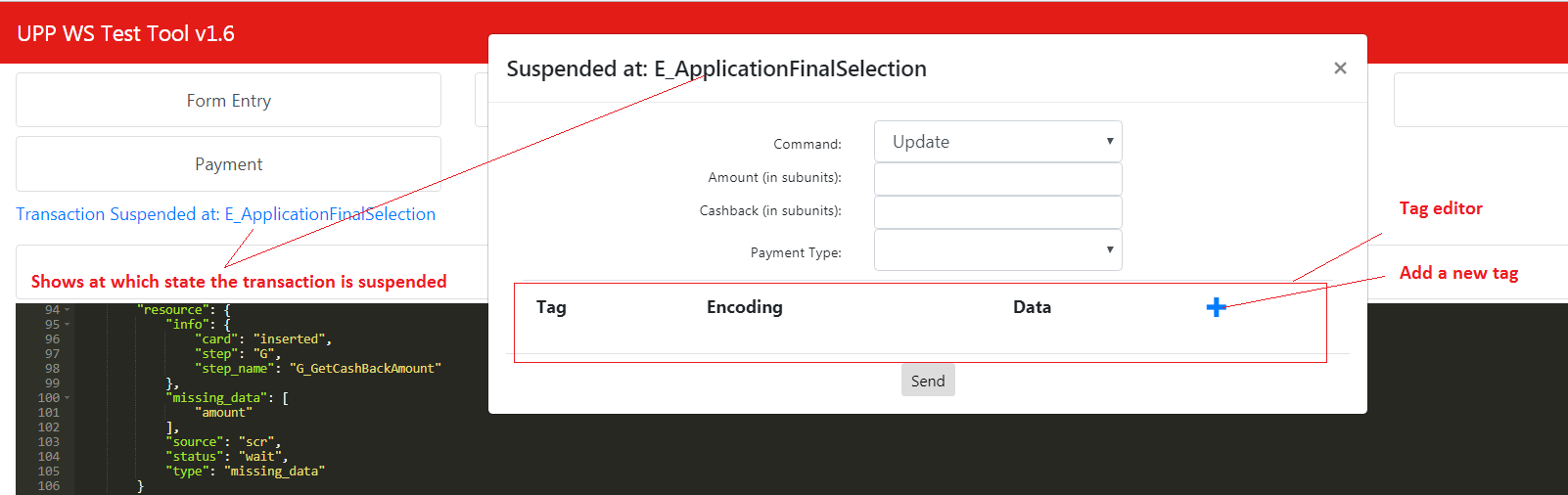
#### Sale transaction example

**Payment Type, Amount, Cashback** fields are optional that might be requested during transaction.

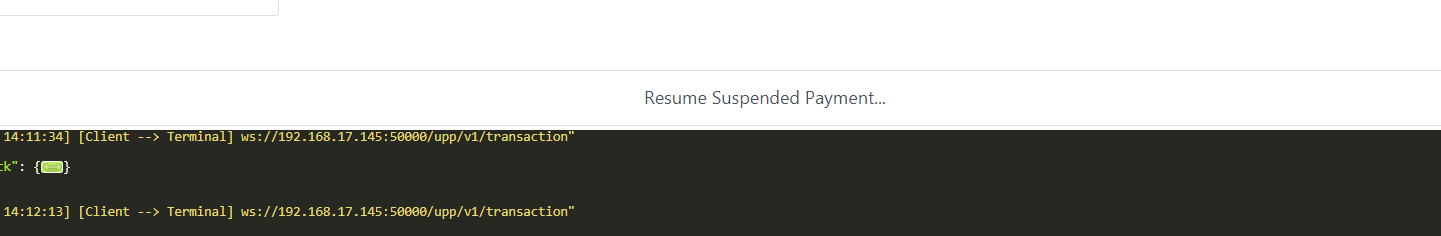
**Contact EMV Suspend steps** will be shown only if **SmartCard** reader will be enabled.



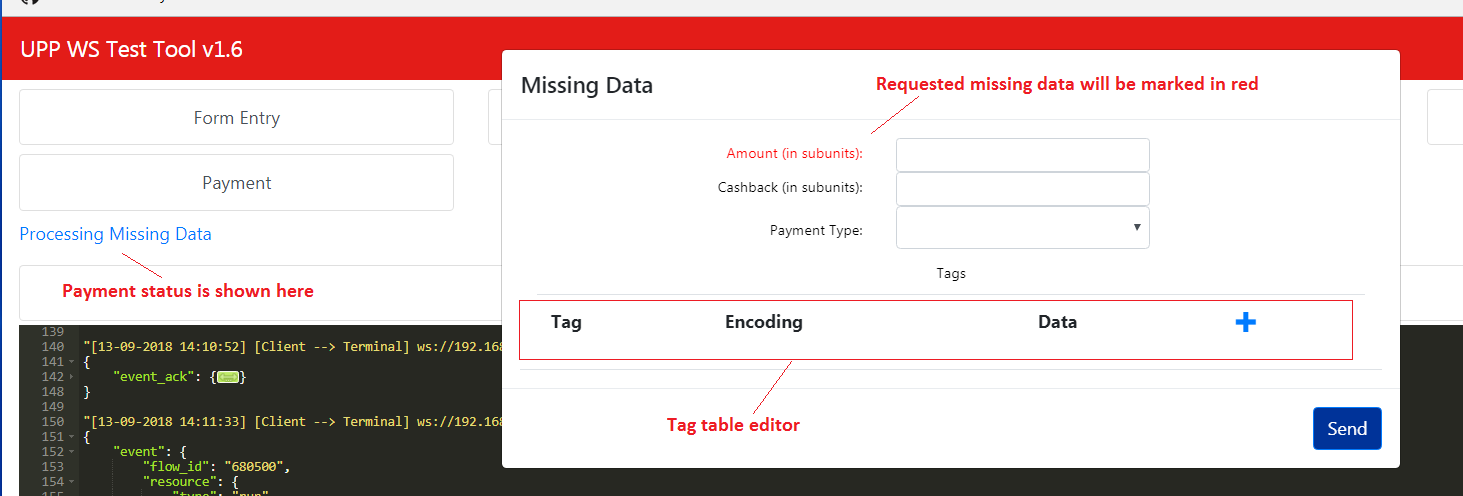
If there are selected suspended steps then during payment processing suspend dialog will appear:



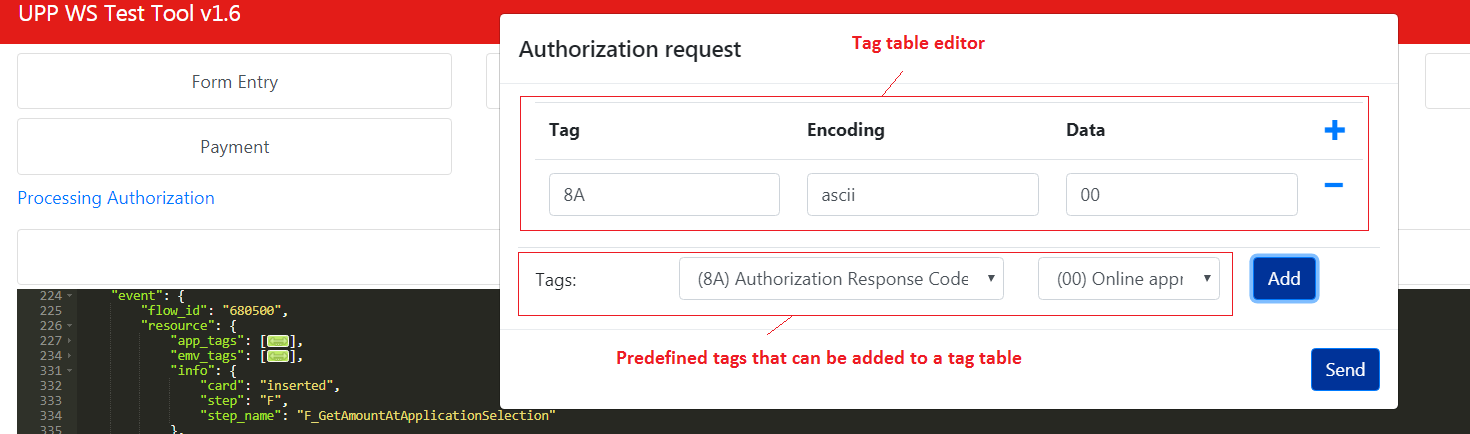
Suspend dialog may be closed to do other actions. And to resume the transaction “Resume Suspended Payment…” should be clicked:



When suspend is finished without updating mandatory fields Missing Data dialog will appear:



After all requested data will be updated Authorization Request dialog will show which is like payment host simulator:



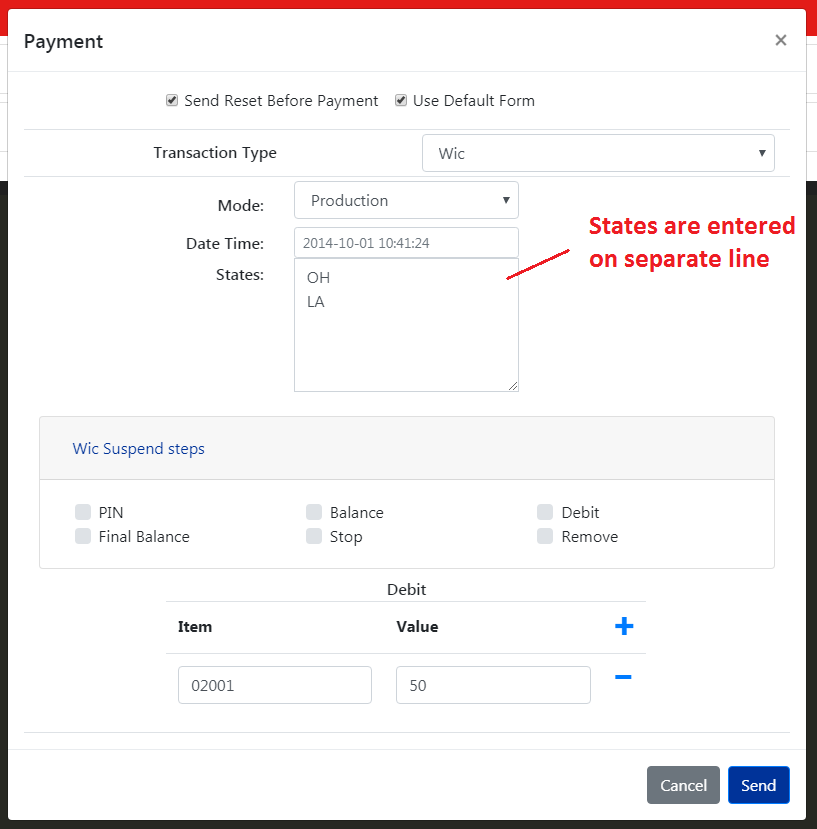
And in the end payment status will also be shown in the status field:



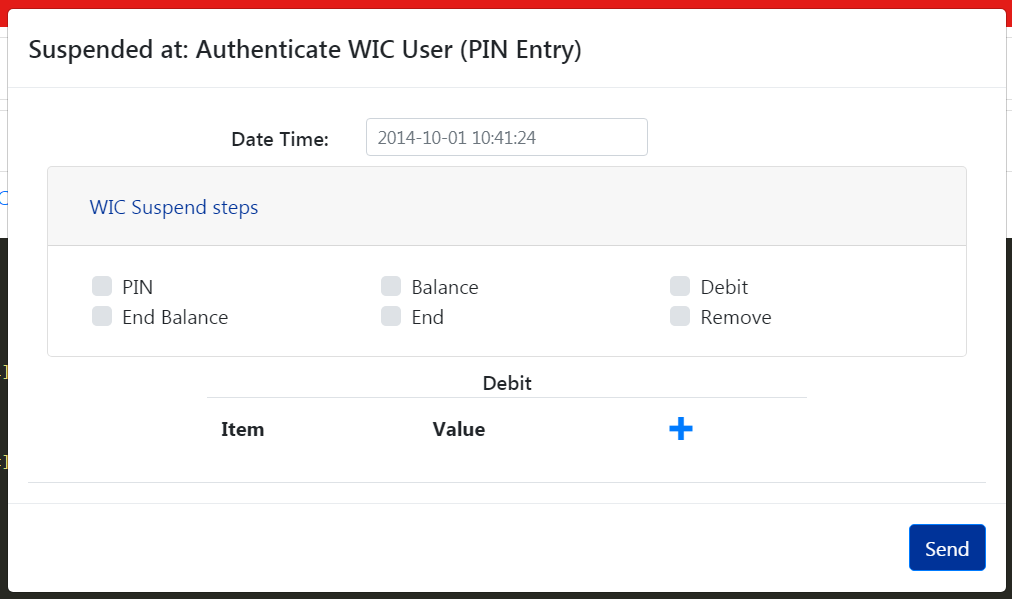
#### WIC transaction

WIC transaction similar to EMV sale transactions has suspend steps and also during transaction may request for a missing data.

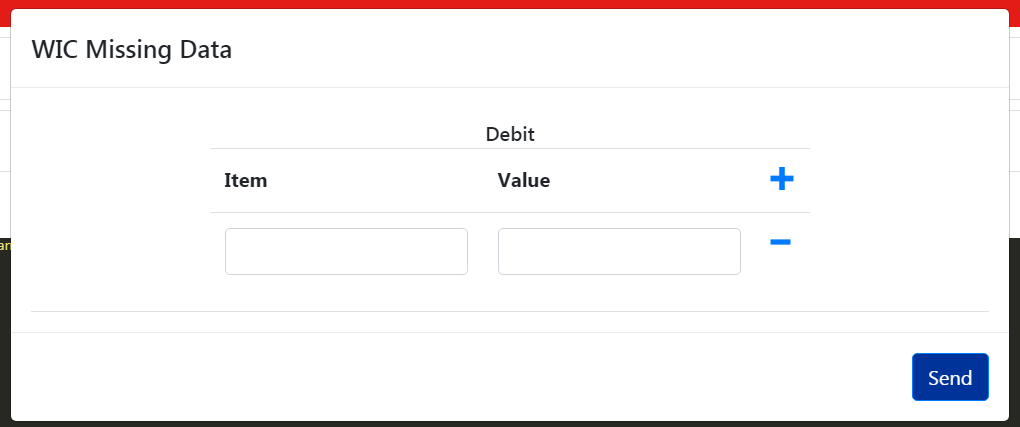
Date Time, States and WIC Suspend steps data will be saved between transactions while the browser won’t reload the UWTT page.



When the transaction will be suspended user will be able to update date and time, add more suspend steps and also debit items.

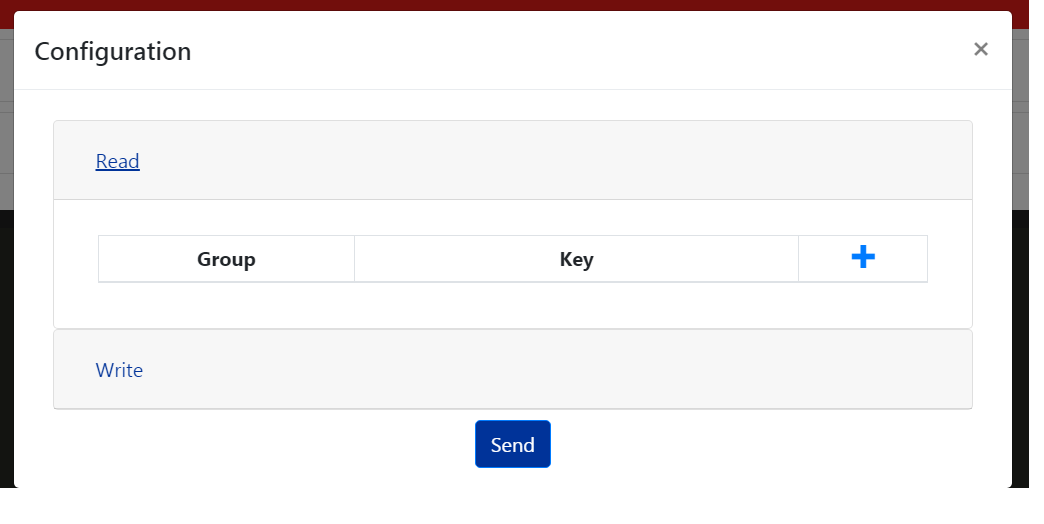


Missing data prompt will appear if there wasn’t set any debit items in the transaction initiation or suspend steps. User may send no items form missing data prompt.



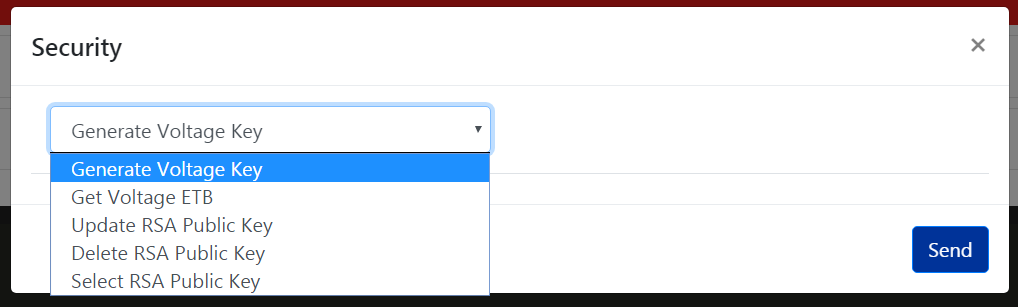
### Configuration

Allows the user to read and write UPP configuration settings. More on these setting can be found in UPP manuals.



### Security

Section provides Voltage and RSA key operations for UPP



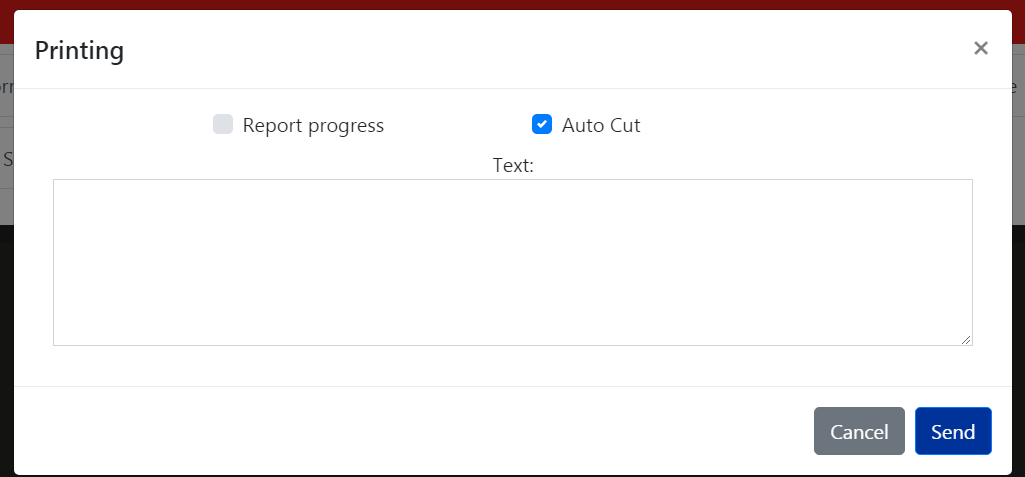
### Printing

Action provides possibility to print receipts on terminals having printer i.e. MOVE 5000.

**Report progress** is flag indicating that printing progress should be sent to a client and is OFF by default to speed up printing.

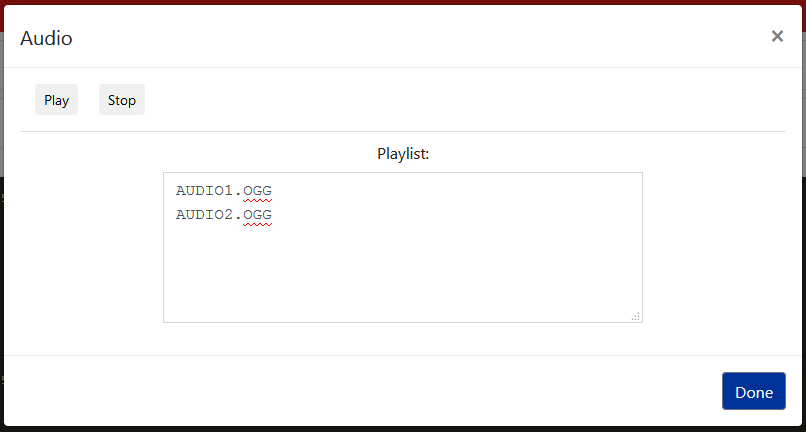
**Auto Cut** property is set to ON by default which instructs the terminal to roll out the paper to see all printed information.

**Text** is printing text/command container where lines are separated with ENTER.



### Audio

Action provides possibility to play preloaded .OGG audio files on the terminal.



**Playlist** is a list of the .OGG audio files to be played. Each audio file name has 15-character limit including extension. Must be given in upper case.

**Play** Start to play list of audio files in the order listed.

**Stop** Stops playback. Playback stops when the current audio file finishes playing.

## JSON message tracer

JSON message tracer window contains all JSON messages sent/received from the terminal. They might be from/to different endpoints.

To expand/collapse JSON object F2 key may be used.

By default JSON objects “event\_ack”, “app\_tags”, “emv\_tags” and “balance” (in WIC type) will be collapsed as they pollute the traces. They can be expanded if the user want that.

