

---

## Swiftium Event Wizard

---

### Index:

- i. [General Event Details](#)
    - a. [Event Name](#)
    - b. [Time Zone](#)
    - c. [Start/End Dates](#)
    - d. [Demo](#)
    - e. [Transaction Lifespan](#)
  - ii. [Event Type](#)
    - a. [Lead Retrieval](#)
    - b. [Attendance Tracking](#)
    - c. [Attendance Tracking + Already Tagged](#)
  - iii. [Capture Source Type](#)
  - iv. [Capture Source Encoded](#)
  - v. [Capture Source Meta Data](#)
    - a. [Field Delimiter](#)
    - b. [Character Encoding](#)
    - c. [Discard Characters](#)
  - vi. [Capture Source Fields](#)
  - vii. [External Data Source Type](#)
    - a. [Swiftium Event Reference](#)
    - b. [Manual File Upload](#)
    - c. [API Integration](#)
  - viii. [External Data Source Fields](#)
    - a. [Field Sequence](#)
    - b. [Field Properties](#)
      - i. [Image/Picture](#)
        - 1. [Convert to JPG](#)
        - 2. [Image Source](#)
          - a. [URL](#)
          - b. [Base64](#)
          - c. [API](#)
        - 3. [Image Height](#)
- ix. [Capturing Method](#)
- x. [Printing](#)
- xi. [Field Display Sequences](#)
  - a. [SwiftLeads app Fields](#)
  - b. [Printer Print-out Fields](#)
  - c. [Reporting Portal Fields](#)
- xii. [Local Snapshot Cache](#)
- xiii. [Scanner Options](#)
  - a. [Timeout](#)
  - b. [Advanced Options](#)
- xiv. [Mobile App Options](#)
  - a. [Advanced Options](#)

- xv. [Offline/Online Network Mode](#)
  - a. [Offline Mode](#)
  - b. [Online Mode](#)
- xvi. [SwiftLeads Licenses](#)
- xvii. [SwiftLeads Options](#)
  - a. [App Banner](#)
  - b. [Scanner Instructions](#)
    - i. [Illustration](#)
    - ii. [Written Text](#)
  - c. [Mobile App Instructions](#)
    - i. [Illustration](#)
    - ii. [Written Text](#)
- xviii. [E-Literature Options](#)
  - a. [What is E-Literature?](#)
  - b. [Email Fields](#)
  - c. [Emailing Mode](#)
  - d. [Delayed E-Literature](#)
  - e. [SMTP Account](#)
- xix. [MapYourShow](#)
- xx. [Onsite Inventory](#)
- xxi. [Summary](#)
  - a. [Default License](#)
  - b. [Standard Qualifiers](#)
  - c. [Default License Log-in](#)

### **General Event Details**

Enter an event name. The name must be unique to the system. An error message will appear below the name textbox if the name already exists. A green message will appear if the name is valid.

Select the time zone where the event is being held. The time zone will be applied to the transaction's capture timestamp, thus ensuring the timestamp reflects the selected time zone.

Start and End Dates should be the event's dates with some room for leeway. It is recommended to extend the end date past the actual day the event ends. The same applies to the start date; set the start date a day or two before the actual day the event begins. This allows for production testing before the event goes live. By default, any event syncs will run at a shorter interval than while in production. For example, a registration data sync might sync once an hour during testing, versus every 1-10 minutes while the event is live.

The demo option should only be used for demo events and never for production events. Demo events are limited to 200 transactions.

Transaction lifespan is the number of days after the event's end date that the transactions are accessible to exhibitors. The default is 90 days.

## **Event Type**

The event type dictates how the Swiftium system will capture and handle transactions.

Lead retrieval gives the ability to capture transactions as leads. These leads can contain notes and be qualified by the user capturing the lead.

Attendance tracking gives the ability to tag all capture transactions with a session name. This tag can later be filtered in the transaction reports to determine which transactions were captured in a session.

Attendance tracking + Already Tagged gives the ability to tag all captured transactions within a session name. This is the same functionality as standalone attendance tracking. The added feature Already Tagged provides is in not allowing an attendee into a session they had previously been scanned into. The SwiftLeads app will alert the user that the attendee had already been scanned for the given session. All transactions are recorded in the system regardless of whether the transaction was Already Tagged. In the transaction reports, a column will be provided to filter Already Tagged transactions from the initial tagged transactions.

## **Capture Source Type**

Capture source type defines which data source transactions will be generated from. The barcode types are not an indication of what the system is capable of scanning. This is merely an indicator of how the event wizard will progress. Choose the options that best match your event.

## **Capture Source Encoded**

Captured data source defines what information is available in the data source that is being captured. In some cases, there might be only a unique ID; meaning additional information must be uploaded to the system to match the ID against. If the capture source is a vCard/Me Card, choose the vCard/Me Card option. At a later step, you will be able to define the information encoded in the vCard.

## **Capture Source Metadata**

Field delimiter defines a single character that separates data fields in the captured data source. The delimiter can be anything as long as the captured data source is a unique ID only; since it is a single field, nothing delimits it.

Character encoding defines how the data in the captured data source is encoded. Most of the time the encoding should be left as UTF-8. If you're experiencing character loss when capturing a data source, the character encoding might need to be changed. Character loss will typically display itself as black question marks in the middle of the captured data or the captured data is scrambled and unreadable.

Discard characters are any unwanted characters that need to be removed from the captured data source. For example, a START OF TEXT character might be appended to the captured data source and this character is not wanted.

## **Capture Source Fields**

Capture source fields means the field sequence produced when the captured data is split by the delimiter character previously selected. This field sequence should not define any field that is going to be uploaded to the system as a look up field. Only fields that are encoded in the captured data source should be defined.

## External Data Source Type

The External data source defines the source of data that will be used to look up a captured unique ID's corresponding external data.

Swiftium Event Reference allows data to be referenced from another event within your account. The fields in the referenced event do not have to match the current event fields. There will be a step to bind the reference fields to the current event fields.

Manual database upload gives the user the ability to upload a flat file, csv file, and/or an excel sheet to the system.

API Integration allows the user to create syncs to obtain data from a third party data site that Swiftium is integrated with.

## External Data Source Fields

The external data source field sequence defines which fields will be appended to captured data. The external data fields are joined on the captured unique ID field (Badge ID). The captured data source field sequence will be listed above the external data field sequence. These captured data source fields will be labeled with a name badge icon. The external data source must include the captured data source fields when uploaded to the system; the two data sources will not be merged.

Field properties give the ability to transform an external data text field into an image/picture field. Transforming a field to an image/picture requires additional field properties to be completed. These include the image source and image height.

By default, images will be converted to JPG to save space and bandwidth. There is the option to select another image format. This option is typically enabled when the image is a logo or other form of graphic. Pictures of attendees should always be converted to the JPG as quality in the photo is not recognizable by the human eye.

The image source defines where the Swiftium system will obtain the images.

- **URL** image source is the simplest way to get images into the Swiftium system. Simply set the external data source field value to an image URL. The Swiftium system will fetch the image from the URL automatically a few minutes after uploading the data to the system.
- **Base64** image source allows for the images to be directly encoded into the external data field.
- **Integration partner** image source will fetch the images in the third party event sync automatically.

Image height is the height of the image when displayed in the SwiftLeads app. 200-400px is the recommended size. Anything smaller than 200px might be hard to see on a mobile device. A tablet could use a larger height if desired. This height is purely personal preference.

## **Capturing Method**

Capturing method defines what medium will be used to capture transactions. There are currently two options available: Handheld Scanner coupled with a smartphone running the SwiftLeads app, and the SwiftLeads mobile app from the app stores. The mobile app utilizes the user's device camera to capture transactions.

NFC capture data sources are also available for the mobile app but not recommended for general population use. NFC is only for use in a controlled mobile device environment, as NFC is currently only compatible with Android NFC enabled devices.

## **Printing**

Printing transactions can be utilized with the Swiftium Bluetooth printers. This requires the use of a handheld scanner. There are two printing options to choose from to trigger the system to print the captured transaction: Print on Scan, and Print on Save.

Print on Scan will print the transaction immediately upon capturing the transaction.

Print on Save will print the transaction when the user or the app saves the transaction.

## **Field Display Sequences**

SwiftLeads App Display defines which fields will be displayed on the app screen upon capturing a transaction. Fields can be joined together to fit on a single line. For example, first and last name might be on the same line.

Printer Print-Out defines which fields will be displayed on the transaction print out.

Reporting Portal Reports defines the fields that are displayed in the reporting portal tables. Typically, all fields will be displayed in this field sequence. To display limited fields for exhibitors/sub-accounts, refer to the field views FAQ.

## **Local Snapshot Cache**

Local Database Snapshot defines which external data fields will be cached in the SwiftLeads app upon setup. It is labeled a snapshot as it a copy of the current data in the system at the time of app setup. The data on the system could change after the app has been setup, rendering the snapshot data out of date.

The data in the snapshot is only used in the app and will be discard and replaced with the most up to date external data on the server when viewing the transactions in the reporting portal.

This snapshot gives the user a better experience by removing the process of fetching data from the server every time a transaction is being captured. In low/non-existent network conditions, where the app would not be able to fetch external data from the server, the snapshot gives the user a seamless experience by always displaying information upon capturing a transaction.

The snapshot does not have to contain all the available fields. It may contain as few or as many fields as needed.

### **Scanner Options**

Scanner timeout defines how long the scanner must be inactive to power off. This does not turn off the coupled Smartphone, only the scanner portion of the handheld. This is to conserve battery life of the scanner. The scanner should last upwards of 4-5 days of moderate scanning. The scanner timeout is defined in minutes. The default is 120 minutes (2 hours)

Advanced options for the scanner can be configured to dial in the SwiftLeads app to your liking. Many of the options will have already been selected, based on the previous selections made in the wizard. These options are most likely required and should not be changed. The options can be reset to the required options by selecting the reset button.

### **Mobile Options**

Advanced options for the mobile app can be configured to dial in the SwiftLeads app to your liking. Many of the options will have already been selected based on the previous selections made in the wizard. These options are most likely required and should not be changed. The options can be reset to the required options by selecting the reset button. The capture method should be set to camera for scanning barcodes. Capture method scanner is for the handheld scanner. Camera symbologies define the type of barcode being captured.

### **Online/Offline Mode**

SwiftLeads App network mode defines how the app will act upon capturing a transaction if no external data is found in the local snapshot database.

In online mode, the app will attempt to fetch external data from the server if external data was not found in the local snapshot.

In offline mode, the app will never attempt to fetch external data from the server if external data was not found in the local snapshot. If no external data could be found in either data source, the app will create a transaction from the captured data.

Regardless of the network mode, the captured transactions data in the reporting portal will always be backfilled/replaced with the most up to date external data.

### **SwiftLeads Licenses**

SwiftLeads Licenses option is currently only for reference and does not affect the event in any way. In the future, this step will allow for license features to be configured here.

### **SwiftLeads Options**

App Banner Logo gives the event creator the ability to upload an image to the system that will be displayed at the top of the SwiftLeads app. The height of the image plays the biggest role in how it looks in the app as the width will auto scale with the height.

App Instructions is a short written note and illustration of how to operate the scanner and the mobile app camera. The instructions are displayed conditionally on what type of app it is—either the handheld app or the mobile app. Either set of instructions can be overwritten. There are no restrictions on what the illustration can be or what the written text can say. For example, these settings could be overwritten with an event logo and/or a custom message from show management.

App colors can be overwritten for a look that matches an event's color scheme. Pressing the red x button above each color box will revert the color back the default SwiftLeads app color.

### **E-Literature Options**

What is E-Literature? E-Literature allows exhibitors to attach files to their custom qualifier choices. Upon qualifying a captured transaction with an E-Literature enabled qualifier choice, an email will be sent to the captured transaction's email. This email message will contain links to access the E-Literature the exhibitor had qualified the captured transaction with. Afterwards, from the reporting portal, exhibitors can run reports on which captured transactions have opened their E-Literature.

E-Literature options define the email fields in the event's data, which email fields should be sent e-literature and when e-literature should be sent out to the captured transactions.

By default, the e-literature emails are sent through the Swiftium SMTP relay. This can be overwritten by specifying a different SMTP relay by adding the account to the Sender SMTP setting panel.

### **MapYourShow**

MapYourShow integration allows exhibitors to SSO into the Swiftium reporting portal to download their captured transactions. Simply specify your MYS API account credentials in the form. This feature requires the use of the Swiftium Onsite Inventory. The Onsite Inventory event must import exhibitors using the MYS exhibitor import tool, located in the Onsite Inventory Management tab, instead of the default flat file upload.

### **Onsite Inventory**

Enabling the Onsite Inventory will create an event in the inventory system and auto link it to the current Swiftium event, eliminating the process of linking the two systems together.

**Summary**

Default license setup code is used to set up handheld devices with the event's configuration and the standard custom qualifiers (if applicable).

Standard custom qualifiers can be configured automatically for handheld devices by pressing the Set System Default button. A custom set of standard custom qualifiers can be configured by clicking the link in the Default Qualifiers panel. These standard qualifiers will be automatically configured when the app is set up using the default license setup code.

The default license login will provide access to all transactions captured by devices that were set up using the default license setup code.