



Upgrade Guide

CommerceDriver™ for *Android™*
Version 2.33 to 2.34

CommerceDriver™ Upgrade Guide

EVO Snap* has released the v2.34 version of CommerceDriver™. This guide is designed to assist partners in the migration process from integrations using v2.33 to v2.34 of CommerceDriver™.

The new v2.34 version of CommerceDriver™ includes the new feature enhancement of Manual Card Entry for transactions.

Enhancements & New Features

Manual Keyed Entry

Manual Keyed Entry (MKE) is the process of running what would typically be a card present transaction without a payment terminal by manually entering the card information in order to process a transaction.

To support the new Manual Keyed Entry feature, the following has been added to Commerce Driver

Added Keyed Entry Property to Pos Request Builder

```
public final class PosRequest implements Serializable {  
  
    public Builder setKeyedEntry(boolean keyedEntry) {  
        this.keyedEntry = keyedEntry;  
        return this;  
    }  
}
```

Setting Keyed Entry Boolean in Pos Request

```
//Example Code  
AuthType authType = (AuthType) transactionTypeSpinner.getSelectedItem();  
PosRequest.Builder builder = new PosRequest.Builder(authType);  
builder.setKeyedEntry(keyedEntryCheckBox.isChecked());
```

Added KeyedEntryData Object

```
package com.evosnap.commercedriver.transaction.keyedEntry;

public class KeyedEntryData {
    private String cardPan;
    private String cardExpiration;
    private String cardCvd;

    public void setCardPan(String cardPan) {
        this.cardPan = cardPan;
    }

    protected String getCardPan() {
        return this.cardPan;
    }

    public void setCardExpiration(String cardExpiration) {
        this.cardExpiration = cardExpiration;
    }

    protected String getCardExpiration() {
        return this.cardExpiration;
    }

    public void setCardCvd(String cardCvd) {
        this.cardCvd = cardCvd;
    }

    protected String getCardCvd() {
        return this.cardCvd;
    }
}
```

Added Manual Keyed Entry Transaction Event

```
//Example Code
class MyTransactionClass implements TransactionEventListener {

    // new callback passes Keyed Entry handler. Has exposed submit and cancel interface
    // methods as shown bellow
    @Override
    public void onRequestKeyedEntryData(KeyedEntryDataHandler keyedEntryDataHandler) {

        AlertDialog.Builder builder = new AlertDialog.Builder(context);
        builder.setTitle("Enter Card Data");
        builder.setPositiveButton(android.R.string.yes, new
        DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                KeyedEntryData data = new KeyedEntryData();
                data.setCardPan(<Card Pan as String>);
            }
        });
    }
}
```

```
        data.setCardExpiration(<Card Exp as MMY String>);
        data.setCardCvd(<Card cvd as String>);
        keyedEntryDataHandler.onKeyedEntryDataEntered(data);
    }
});

builder.setNegativeButton(android.R.string.no, new
DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        keyedEntryDataHandler.onKeyedEntryDataCancelled();
    }
});
}
}
```

StartTransaction thrown SnapValidationError Changed

The Boolean parameter `isKeyed` will now allow for transactions to be processed without the need of a valid terminal connection. Keyed Entry transactions can still be run with a terminal connected, but `StartTransaction` will no longer throw a connection error if called with keyed entry enabled in transaction Pos Request.