Notes on Cerner CCL Interface Scripts

>> Script pipeline for Inbound and Outbound messages:

Inbound: Modify Original → Map To Library → Modify Object

Outbound: Modify Object → Map From Library → Modify Original

>> Order of Events:

Upon receiving an inbound message, the *Modify Original* script is applied to the message to alter its initial format. For example, to remove unsupported segments.

The message will then be broken up via the *Map To Library* script and stored in an object (i.e., data structure) called *oen_reply* that uses Cerner's HL7 record structure.

The *oen_reply* data structure contains a number of fields referenced via:

oen_reply->[library structure]

Map To Library uses this structure to store the message data so it is easily manipulated by the Cerner CCL scripts.

Modify Object can use *oen_reply* to access parts of the message string (e.g., segment, field, component, sub-component) easily and modify the message without the need for complicated string manipulation, just like you would using good old OOP.

Map From Library extracts the information from data fields in *oen_reply* and stores it in *oen_reply->out_msg* for transmission on an outbound interface. In general, *oen_reply>out_msg* is used to pass modified messages out of scripts.

* Note that a *Cerner Typescript* may be applied if multiple message types are expected on an inbound interface. The *Typescript* determines the type of message before any other scripts are applied.

* If an ACK is required on an inbound interface, an ACK script is called by the interface once formatting is complete or a transaction received in the case that there is no formatting.

Finally, the *ComServer Route Script* is run. It is used in custom routing interfaces. The script uses the *oen_route* data structure and the *oen_route->route_list* field to change the inbound interface's route list after meeting certain criteria based on the information contained in the message.

>> Example Modify Original script:

```
record work
(
        1       test_msg=VC   ;variable char
)
set work->test_msg = oen_request->org_msg
set len = movestring("Inbound",1,work->test_msg,1,7)
set oen_reply->out_msg = work->test_msg
```

The above script modifies the original message by adding "Inbound" to the first seven bytes of the message. The modified message is passed out of the script via *oen_reply->out_msg*.