**Service Repair Module 2.0**

Author: Alex Littlejohn

alittlejohn@truecloud.com

Last Update: 8/28/2016, v1.0

# Overview

The Service Repair module is a series of Custom Records, scripts, and standard configuration elements within NetSuite to support a company’s ability to track Customer and Internal repairs, record historical work performed, and to integrate the billing of services into a standard NetSuite document flow.

This original bundle (v1.0) was created for American Barcode and was iterated on over the course of a few years. This bundle has been installed in the following accounts:

* American Barcode
* Coridian
* Retail Technology Group
* And others

This updated bundle (v2.0) was a refresh of the original bundle on behalf of MidFlorida Golf Carts. This refresh was performed to enhance the code readability, improve on NetSuite’s recommended best-practices, speed up the processing of the custom development, and in general enhance the functionality of the bundle.

# Terminology

* Service Request / Work Order
  + Refers to the SRV Request custom record. This is a document similar in nature to a Sales Order; it relates an Entity (Customer, Internal Customer) with their Equipment with status/issue. This document is the source for viewing statuses, billing, etc.
* Service Technician
  + An Employee with NetSuite access who is associated to the SRV Request record and is assumed to be responsible for the repair, communication, and creation of a Sales Order.
* FAM-ONLY
  + Noted next to components that should only be included in the bundle to a production environment that has the Fixed Asset Management module.

# Script Overview

The scripts written for the v2.0 bundle are object-oriented in an attempt to make the code easier to read, support, and enhance. The general code format is as follows:

* NetSuite primary functions are defined in each script and should call sub-functions to perform specific actions. In some cases the sub-functions themselves refer to other sub-functions in an attempt to ensure that each function perform a very specific action.
* Utilization of a library file helps to normalize the functions being called. The best example of this is the log() function which ensures that all logging performed is handled uniformly.

The individual scripts referenced below can be found in the “Development Account – Alex” environment (where the bundle was created) as well as in MidFlorida Golf Cart’s production environment. This document will step through the major functions present, if more clarification is needed please reach out to the author and if necessary the document will be updated.

# Record Overview

The following custom-records are included in the Service Module v2.0 bundle:

* Fixed Asset Records (FAM-ONLY)
  + FAM – Asset
  + FAM – Asset Type
  + FAM – Repair & Maint Category
  + FAM – Depreciation Method
  + FAM – Repair & Maint Sub B
  + FAM – Repair & Maint Sub A
  + FAM – Alternate Depreciation
  + FAM – Depreciation History
  + FAM – Alternate Methods
  + BG – Summary Records
  + SRV FAM Maintenance Schedule
* SRV Request
  + Primary record that represents a Service Request/Work Order. This is the primary record and most closely resembles a Sales Order.
* SRV Work
  + Used to relate a SRV Request to a SRV Equipment record in a many-to-one relationship. The act of repairing a piece of Equipment causes this record’s status to be updated along with notes as to what was repaired.
  + This record is necessary to ensure that the Entity’s history is preserved if the SRV Equipment is repaired on multiple occasions. Without this record the SRV Equipment would be directly related to the SRV Request and any historical references and repairs would be overwritten.
* SRV Equipment
  + Refers to an Entity’s equipment (Customer, Internal Customer). Generally has a Serial Number associated with it to ensure its uniqueness, this record is what is being repaired.
* SRV Part / Time
  + This record is created and associated to a SRV Request if the technician needs to associate an Inventory Item (replacing a broken component) or a Non-Inventory Item (charging for an additional service). These records do not allocate Inventory until they are placed on a Sales Order, if allocation is an issue it is recommended that the Service Technician create a Sales Order from their SRV Requests regularly.

# User Event Scripts

* [SRV : Part / Labor – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=19&c=TSTDRV1537159&h=14eb27283e6e1f45ca3e&_xt=.js)
  + Record Type: SRV Part / Time
  + NetSuite Primary Functions: afterSubmit
  + Sub-functions:
    - createChangeOrderTask
      * If the SRV Part / Time is flagged as a Change Order, the script will generate a Task (NetSuite Standard Record) and update the SRV Part / Time with the Task. The Task will be assigned to the Customer’s Sales Rep if one exists otherwise it will be assigned to the Service Technician.
* [SRV : Request – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=10&c=TSTDRV1537159&h=77c0aba459a57b6d5f23&_xt=.js)
  + Record Type: SRV Request
  + NetSuite Primary Functions:
  + Sub-functions: beforeLoad, beforeSubmit, afterSubmit
    - approveChangeOrders
      * If the user has pushed a button to approve all Change Orders, this will iterate through any unapproved Change Orders associated with the Service Request and approve them. Users could otherwise approve Change Orders individually at the Task-level.
    - getConfiguration
      * Pulls dynamic information that is either setup within a script-parameter or pulled from other configuration in NetSuite. This information is stored as a session-object to speed up the processing in future processes.
    - buildUI
      * Writes the buttons, custom-address functionality, and writes the sublist that displays the SRV Equipment/SRV Work records related to the Service Request.
* [SRV : Asset – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=165&c=TSTDRV1537159&h=484b2ca335135e9af047&_xt=.js) (FAM-ONLY)
  + Record Type: FAM - Asset
  + NetSuite Primary Functions: beforeLoad, beforeSubmit, afterSubmit
  + Sub-functions:
    - serviceFormOrganization
      * Create a custom-tab and moves Service-specific fields to this tab to better organize the form.
    - normalizeServiceDate
      * Takes the date the user enters in the standard date-field and ensures it’s normalized to always be the first of the month. This is to ensure that the records are eligible for service at the beginning of each month.
    - addEquipment
      * If the Asset has had a maintenance date defined and a SRV Equipment record has not been created for the record, this generates the SRV Equipment record which will allow this to be put on a SRV Request when the maintenance date occurs.
* [SRV : Fulfillment – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=164&c=TSTDRV1537159&h=c3ea41a59719a8d86758&_xt=.js)
  + Record Type: Item Fulfillment
  + NetSuite Primary Functions: beforeLoad, afterSubmit
  + Sub-functions:
    - disableServiceFields
      * Disables the service body and column fields to ensure users are not altering them on the Fulfillment-level. If changes are needed the user should alter it on the Sales Order.
    - completeServiceWork
      * When an Item is fulfilled that has a SRV Work record related to it at the line level it will cause the SRV Work record to be updated to “Completed” (or equivalent status).
* [SRV : Work – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=16&c=TSTDRV1537159&h=3cea282f28e43ec6251a&_xt=.js)
  + Record Type: SRV Work
  + NetSuite Primary Functions: beforeLoad
  + Sub-functions:
    - buildUI
      * Puts a button on the page when viewing the record that sends the user to a Time Tracking record.
* [SRV : Service Task – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=21&c=TSTDRV1537159&h=3b4c0ad47b27aa2971a5&_xt=.js)
  + Record Type: Task
  + NetSuite Primary Functions: beforeSubmit, afterSubmit
  + Sub-functions:
    - rejectChangeOrder
      * Rejecting a Change Order Task will remove the Item from the related Sales Order and Estimates.
    - updateServiceWorkStatus
      * If all change orders related to the SRV Work (that is associated to the service Task) are approved then update the SRV Work status to reflect this; if all change orders are rejected update the status to reflect this.
* [SRV : Transaction – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=17&c=TSTDRV1537159&h=da7a8484fb60ade169c8&_xt=.js)
  + Record Type: Estimate, Sales Order
  + NetSuite Primary Functions: beforeLoad, beforeSubmit, afterSubmit
  + Sub-functions:
    - setValues
      * Gets values from the server-side as the page is loading, stores data in a custom-field written by the script. This data is consumed by the client-script to set the values.
    - rejectTranscation
      * Rejecting a transaction causes a Sales Order to be generated with the default Item (defined in Company Preferences) set as a line-item., the quantity should match the number of SRV Work records.
    - serviceParts
      * Builds a button on the form that when pressed will set the SRV Part / Time records associated to the SRV Request that have not been placed on a transaction yet.
    - updateServiceWork
      * Sets the transaction on the SRV Work record, once this field is set it is unable to be placed on another transaction. If the transaction is deleted it should unset the value in the SRV Work record.
    - updateServiceParts
      * Sets the transaction on the SRV Part / Time record, once this field is set it is unable to be placed on another transaction. If the transaction is deleted it should unset the value in the SRV Work record.
* [SRV : Equipment – User Event](https://system.na1.netsuite.com/core/media/media.nl?id=15&c=TSTDRV1537159&h=4ea88d4de4a46863318c&_xt=.js)
  + Record Type: SRV Equipment
  + NetSuite Primary Functions: beforeLoad, beforeSubmit, afterSubmit
  + Sub-functions:
    - getConfiguration
      * Pulls dynamic information that is either setup within a script-parameter or pulled from other configuration in NetSuite. This information is stored as a session-object to speed up the processing in future processes.
    - buildUI
      * Writes a button to support the generation of a new SRV Request from this Equipment record.
    - updateExternalId
      * Normalizes the External ID of the record, if the Customer is changed (for example) the External ID needs to be updated.

# Client Scripts

* [SRV : Request - Client](https://system.na1.netsuite.com/core/media/media.nl?id=14&c=TSTDRV1537159&h=b81bec319ad96dcc5c84&_xt=.js)
  + Record Type: SRV Request
  + NetSuite Primary Functions: pageInit, saveRecord, fieldChanged, lineInit, validateDelete
  + Sub-functions:
    - addressLibrary functions
      * The addressLibrary set of functions is what recreates the custom address-field functionality. Refer to library file referenced below.
    - saveRecord
      * Ensures at least one Service Work line exists
    - fieldChanged
      * Changing the customer causes the Address-select list to be refreshed.
      * Changing an address-value causes the distance to be recalculated.
      * Changing the Serial Number in the Work sublist causes the system to lookup fields and source into other columns.
    - lineInit
      * Selecting a line that has previously been saved disables the line so the user does not override any important information. If a key-value needs to be altered on a line they should remove the line and add a new line.
    - validateDelete
      * Removing a line will alert the user if the SRV Work record has been placed on a transaction and will warn them that the record will be inactivated. Confirming will inactivate the record removing it from future views after the record has been saved.
    - setInternalCustomer
      * Sets the Global Preference defined internal customer.
* [SRV : Transaction - Client](https://system.na1.netsuite.com/core/media/media.nl?id=18&c=TSTDRV1537159&h=f379a69d113b77d7172b&_xt=.js)
  + Record Type: Estimate, Sales Order
  + NetSuite Primary Functions: pageInit
  + Sub-functions:
    - setServiceRequestValues
      * Sets SRV Work lines if data exists. This data is written to the page by the User Event deployed to the record and stores the values in a JSON object that is hidden.
    - addServiceParts
      * Sets SRV Part / Time lines if data exists. This data is written to the page by the User Event deployed to the record and stores the values in a JSON object that is hidden.
* [SRV : Service Receipt - Client](https://system.na1.netsuite.com/core/media/media.nl?id=13&c=TSTDRV1537159&h=4171d75528780863d133&_xt=.js)
  + Record Type: Suitelet – SRV : Service Receipt
  + NetSuite Primary Functions: saveRecord
  + Sub-functions:
    - saveRecord
      * Grabs sublist data and stores in a hidden field.
      * **Note:**  this function can potentially be written following best-practices; issue was encountered where the sublist data wasn’t present when the form was submitted, this was created as a work-around.

# Scheduled Scripts

* [SRV : Sch. Maintenance – Scheduled](https://system.na1.netsuite.com/core/media/media.nl?id=166&c=TSTDRV1537159&h=913d0ee1cc35f7e690fe&_xt=.js) (FAM-ONLY)
  + Record Type: SRV FAM Maintenance Schedule
  + NetSuite Primary Functions:
  + Sub-functions:
    - -

# Suitelet Scripts

* [SRV : Service Receipt - Suitelet](https://system.na1.netsuite.com/core/media/media.nl?id=13&c=TSTDRV1537159&h=4171d75528780863d133&_xt=.js)
  + Record Type: N/A
  + NetSuite Primary Functions: execute
  + Sub-functions:
    - getAssetsToService
      * Returns a list of Asset-data where the Service Date is within the current month.
    - processRequests
      * Creates a SRV Request for every ten assets that need to be processed (if there are 23 Assets there would be 3 SRV Requests created). This function uses the buildServiceRequest function which is included in the library file.
    - updateAssets
      * If a SRV Request was created successfully for an Asset, the Service Date is moved forward according to its assigned Maintenance Schedule.

# Library Scripts

* [TC\_SRV\_Library.js](https://system.na1.netsuite.com/core/media/media.nl?id=9&c=TSTDRV1537159&h=fd50447cf555afe385ea&_xt=.js)
  + Sub-functions:
    - log
      * Shorthand function to perform nlapiLogExecutions(), assumes level to be “DEBUG” unless otherwise stated. stringifies objects.
    - buildServiceREquest
      * A function to generate a Service Request with Service Work, performs UPSERTS of Equipment as needed.
    - distance
      * Set of functions to calculate distance between two points (latitude and longitude points) powered by MapQuest.
    - yield
      * Shorthand for yielding scheduled scripts.
    - getParameters
      * Get’s parameters from URLs for use on client-side scripts.
    - getTabs
      * Stores all custom-tabs related to a form in a session object. Used to ensure correct sublist placement.
    - getCustomRecordId
      * Gets the Internal ID of custom-records to build dynamic links in sublists and buttons.
    - businessDays
      * Takes the current date and returns a new date beyond the number of business days defined, business days calculated as Monday through Friday.
    - removeSpecialCharacters
      * Used to ensure normalized serial numbers, removes all non alphanumeric values from a string.
    - addressLibrary
      * Used in User Events and Clients to generate and maintain the custom address fields on a record.

# Workflows

* Service Request Email