

[Documentation](#)**MODULES**

ss_cs_clear_pdf_group
_line
ss_ue_commission_cal
c_line_item

Home

Commission and DMF Line Item Management Script

Description

This script manages the addition of commission and DMF (Dealer Marketing Fund) line items in NetSuite records. It automatically updates these line items based on specific conditions and parameters.

Details

- **API Version:** 2.x
- **Script Type:** UserEventScript
- **Title:** Commission and DMF Line Item Management Script

Requirements

The script requires the following NetSuite modules:

- `IN/runtime`
- `IN/log`

Functionality

Main Function: `beforeSubmit`

The `beforeSubmit` function is triggered before a record is submitted. It performs the following tasks:

1. Retrieves the new record and its line items.
2. Calculates the commission and DMF amounts based on media budget and billing type.
3. Updates line items with the calculated commission and DMF amounts.
4. Sets PDF group numbers for these line items.
5. Adjusts line items based on specific conditions.

Key Sub-functions

`UpdateOriginalLineItem`

Updates the original line item with the specified value and group number.

Parameters:

- `currentRecord` : The current NetSuite record being processed.
- `sublistName` : The name of the sublist being modified.
- `lineIndex` : The line index to update.
- `value` : The value to set for the line item.
- `groupNumber` : The group number to assign to the line item.

Returns: The set PDF group number value.

`addCommissionLineItem`

Adds or updates a commission/DMF line item in the transaction record.

Parameters:

- `currentRecord` : The current NetSuite record being processed.
- `sublistName` : The name of the sublist being modified.
- `mediaCommissionItem` : The ID of the media commission item.
- `commissionAmount` : The amount of commission to apply.
- `currentItemValue` : The current item's value object.
- `pdfGroupNumberInitValue` : The initial value of the PDF group number.

`calculateCommission`

Calculates the commission amount based on media budget and commission rate.

Parameters:

- `mediaBudget` : The media budget amount.
- `bodymediaCommission` : The body media commission rate.
- `isPostbill` : Flag indicating if it is post-bill.
- `hasDMF` : Indicates if the item has DMF.

Returns: The calculated commission amount.

`calculateDmf`

Calculates the DMF amount based on media budget and commission rate.

Parameters:

- `mediaBudget` : The media budget amount.
- `bodymediaCommission` : The body media commission rate.
- `isPostbill` : Flag indicating if it is post-bill.

Returns: The calculated DMF amount.

`calculatePrebill`

Calculates the prebill amount.

Parameters:

- `mediaBudget` : The media budget amount.
- `bodymediacommission` : The body media commission rate.
- `x` : Additional multiplier or parameter.

Returns: The calculated prebill amount.

`getLineItemValue`

Retrieves line item values from a given record and line index.

Parameters:

- `newRecord` : The NetSuite record object.
- `sublistId` : The ID of the sublist.
- `line` : The line number to retrieve values from.

Returns: An object containing line item values.

extractItemDetail

Extracts item details from a given line item value object.

Parameters:

- `currentItemValue` : The current line item value object.
- `runtime` : The runtime object for getting script parameters.

Returns: An object containing extracted item details.

Error Handling

The script includes a try-catch block to handle any errors that occur during execution. Errors are logged using `log.error`.

Usage

1. Deploy this script as a User Event Script in NetSuite.
2. Ensure the necessary custom fields (`custbody_moore_media_commission`, `custbody_moore_media_billing_type`, etc.) are present in the records.
3. Configure the script parameters (`custscript_ss_media_commission_item_para`, `custscript_ss_dmf_item_para`) as required.

Notes

- The script relies on specific field IDs and custom fields in NetSuite. Ensure these fields are correctly set up in your environment.
- Adjust the script parameters as needed to match your specific use case.

Client Script for Line Copy

To prevent conflicts when users copy line items, there is an associated client script. This script clears the PDF group field on the new line to avoid issues and to save the user from manually clearing it each time.

```
/*
 * @NApiVersion 2.x
 * @NScriptType ClientScript
 */
define(['N/record', 'N/currentRecord', 'N/log'], function(record, currentRecord, log) {

    function pageInit(context) {
        var makeCopyButton = document.getElementById('item_copy');
        if (makeCopyButton) {
            makeCopyButton.onclick = function() {
                try {
                    // Perform the line copy
                    item_machine.copyline();

                    setWindowChanged(window, true);

                    var rec = currentRecord.get();

                    var sublistId = 'item';

                    var currentLine = rec.getCurrentSublistIndex({
                        sublistId: sublistId
                    });

                    // Clear the 'custcol_pdf_group' field on the new line
                    rec.setCurrentSublistValue({
                        sublistId: sublistId,
                        fieldId: 'custcol_ss_pdf_group',
                        value: ''
                    });

                    console.log('line copied')

                    return false;
                } catch (e) {
                    console.error('Error in Make Copy Click Handler', e.message);
                    return false;
                }
            };
        }
    }

    return {
        pageInit: pageInit
    };
});
```



Documentation

MODULES

ss_cs_clear_pdf_group
_line
ss_ue_commission_cal
c_line_item

MODULE

ss_ue_commission_calc_line_item

This script manages the addition of commission and DMF (Dealer Marketing Fund) line items in NetSuite records. It automatically updates these line items based on set conditions and parameters.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 1

Requires

- module:N/runtime
- module:N/log

Members

INNER

bodyCommission

Type: number

The body commission rate from the SO transaction record

[View Source](#)

ss_ue_commission_calc_line_item.js, line 68

Example

```
newRecord.getValue({fieldId: 'custbody_moore_media_commission'}) = 15%  
var bodyCommission = newRecord.getValue({fieldId: 'custbody_moore_media_commission'}) / 100;
```

INNER

Type: number

commissionAmount

if the line item is commissionable the commission amount is calculated using the calculateCommission function

[View Source](#)

ss_ue_commission_calc_line_item.js, line 222

Example

```
var commissionAmount = calculateCommission(extractItemDetails.mediaBudget, bodyCommission, isPostBill, extractItemDetails.itemLineDmf);  
addCommissionLineItem(newRecord, sublistName, extractItemDetails.mediaCommissionItem, commissionAmount, currentItemValue, pdfGroupNumberInitValue);
```

INNER

Type: object

currentItemValue

the lines item value object

[View Source](#)

ss_ue_commission_calc_line_item.js, line 131

Example

```
var currentItemValue = [currentLineMediaMonth: '1', currentLineItemMediaDmfItem: '1', currentItemValue: '1', currentLineMediaDate: '1', currentLineDmfCheckbox: '1', currentli
```

INNER

Type: number

dmfAmount

if the line item is DMF the DMF amount is calculated using the calculateDmf function

[View Source](#)

ss_ue_commission_calc_line_item.js, line 251

Example

```
var dmfAmount = calculateDmf(extractItemDetails.mediaBudget, bodyCommission, isPostBill);  
addCommissionLineItem(newRecord, sublistName, extractItemDetails.mediaDmfItem, dmfAmount, currentItemValue, pdfGroupNumberInitValue);
```

INNER

Type: object

extractItemDetails

The extracted item details object

[View Source](#)

ss_ue_commission_calc_line_item.js, line 141

Example

```
var extractItemDetails = [currentLineMediaMonth: '1', itemLineDmf: '1', currentItemValue: '1', currentLineMediaDate: '1', itemLineOverrideMediaCalc: '1', currentLineDmfGroup:
```

MEMBERS

bodyCommission
commissionAmount
currentItemValue
dmfAmount
extractItemDetails
isPostBill
lastCommissionDmfIndex
lastCommissionDmfIndex
lineCount
mediaBillType
newRecord
nextDmfNumber
pdfGroupNumberInitValue
pdfGroupNumberInitValue
pdfGroupNumberInitValue
preBill
sublistName
totalCommissionAndDmf
uniquePdfGroupNumbers

METHODS

UpdateAllLineItemMedia

INNER Type: boolean

isPostBill

Flag indicating if the media billing type is post-bill

[View Source](#)

ss_ue_commission_calc_line_item.js, line 90

Example

```
var isPostBill = mediaBillType === '2';
return true if mediaBillType is post-bill
```

INNER Type: number

lastCommissionDmfIndex

The last commission DMF index is set to the current index

[View Source](#)

ss_ue_commission_calc_line_item.js, line 235

Example

```
lastCommissionDmfIndex = i
```

INNER Type: number

lastCommissionDmfIndex

The last commission DMF index is set to the current index

[View Source](#)

ss_ue_commission_calc_line_item.js, line 262

Example

```
lastCommissionDmfIndex = i
```

INNER Type: number

lineCount

The number of lines in the sublist

[View Source](#)

ss_ue_commission_calc_line_item.js, line 50

Example

```
var lineCount = newRecord.getLineCount({sublistId: sublistName});
```

INNER Type: string

mediaBillType

The media billing type from the SO transaction record

[View Source](#)

ss_ue_commission_calc_line_item.js, line 80

Example

```
var mediaBillType = newRecord.getValue({fieldId: 'custbody_moore_media_billing_type'});
```

INNER Type: object

newRecord

The new record object from the context

[View Source](#)

ss_ue_commission_calc_line_item.js, line 39

Example

```
let newRecord = context.newRecord;
```

INNER Type: number

nextPdfNumber

The next available PDF group number

[View Source](#)

ss_ue_commission_calc_line_item.js, line 151

Example

```
var nextPdfNumber = getNextAvailablePdfNumber(uniquePdfGroupNumbers);
```

INNER

Type: string

pdfGroupNumberInitValue

The PDF group number value is set using the `UpdateOriginalLineItem` function

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 168

Example

```
var pdfGroupNumberInitValue = UpdateOriginalLineItem(newRecord, sublistName, i, preBill, nextPdfNumber);
```

INNER

Type: string

pdfGroupNumberInitValue

if the line item is post bill and the item is either commissionable or DMF the PDF group number is set using the `UpdateOriginalLineItem` function

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 203

Example

```
var pdfGroupNumberInitValue = UpdateOriginalLineItem(newRecord, sublistName, i, extractItemDetails.mediaBudget, nextPdfNumber);
```

INNER

Type: string

pdfGroupNumberInitValue

The PDF group number value is set using the `setPDFGroupNumber` function

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 277

Example

```
var pdfGroupNumberInitValue = setPDFGroupNumber(newRecord, sublistName, i, nextPdfNumber);
```

INNER

Type: number

preBill

if the line item is not overridden and the item is either commissionable or DMF the prebill amount is calculated using the `calculatePrebill` function

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 174

Example

```
var preBill = calculatePrebill(extractItemDetails.mediaBudget, extractItemDetails.itemLineCommission ? bodyCommission : 0, extractItemDetails.itemLineDmf ? 1 : 0);
```

INNER

Type: string

sublistName

The name of the sublist being processed

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 49

Example

```
var sublistName = 'item';
```

INNER

Type: number

totalCommissionAndDmf

The total commission and DMF amount

[View Source](#)

`ss_ue_commission_calc_line_item.js`, line 117

INNER

Type: Array

uniquePdfGroupNumbers

An array of unique PDF group numbers

[View Source](#)

ss_ue_commission_calc_line_item.js, line 102

Example

```
var uniquePdfGroupNumbers = [ '1', '2', '3', '4', '5' ];
```

Methods

INNER**UpdateAllLineItemMediaMonth(currentRecord, sublistName, lineindex)**

Updates all line items with the media month based on the start date. The function updates the media month field for all line items in the sublist.

Parameters:

Name	Type	Description
currentRecord	Object	The NetSuite record object.
sublistName	string	The name of the sublist.
lineindex	number	The line index to update.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 973

Example

```
UpdateAllLineItemMediaMonth(currentRecord, sublistName, lineindex)
```

INNER**UpdateOriginalLineItem(currentRecord, sublistName, lineindex, value, groupNumber) → {string}**

This Function is for Updates the original line item with the specified value and group number. The Function is called when the line item is not overridden and the item is either commissionable or DMF. The function also sets the PDF group number for the line item. The function returns the PDF group number value.

Parameters:

Name	Type	Description
currentRecord	Object	The current NetSuite record being processed.
sublistName	string	The name of the sublist being modified.
lineindex	number	The line index to update.
value	number	The value to set for the line item.
groupNumber	number	The group number to assign to the line item.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 308

Returns:

The set PDF group number value.

Type: string

Example

```
UpdateOriginalLineItem(currentRecord, sublistName, lineindex, value, groupNumber)
```

INNER**addCommissionLineItem(currentRecord, sublistName, mediaCommissionItem, commissionAmount, currentItemValue, pdfGroupNumberInitValue) → {void}**

Adds a commission/DMF line item to the current Transaction record. The function also sets the PDF group number for the line item (Commission/DMF) The function checks if the line item already exists and updates the existing line item if found.

Parameters:

Name	Type	Description
currentRecord	Object	The current NetSuite record being processed.
sublistName	string	The name of the sublist being modified.
mediaCommissionItem	number	The ID of the media commission item.
commissionAmount	number	The amount of commission to apply.
currentItemValue	Object	The current item's value object.
pdfGroupNumberInitValue	string	The initial value of the PDF group number.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 355

Returns: Type: `void`

INNER

adjustAmountOnSpecificCondition(newRecord, lineIndexToAdjust)

Adjusts the amount of a specific line item based on a specific condition. The function adjusts the amount of a line item if the total amount is not a whole number.

Parameters:

Name	Type	Description
<code>newRecord</code>	<code>Object</code>	The NetSuite record object.
<code>lineIndexToAdjust</code>	<code>number</code>	The line index to adjust.

[View Source](#)

`ss_ue_commission_calc_line_item.js, line 896`

Example

```
adjustAmountOnSpecificCondition(newRecord, lineIndexToAdjust)
```

INNER

beforeSubmit(context) → {void}

Handles the before submit event for the transaction records - This function manages the addition of commission and DMF line items based on the current state of each line item in the transaction. - This function also manages the sequencing of PDF group numbers for commission and DMF line items.

Parameters:

Name	Type	Description
<code>context</code>	<code>Object</code>	The context object containing the new record.

[View Source](#)

`ss_ue_commission_calc_line_item.js, line 16`

Throws:

Throws an error if an error occurs.

Type: `error`

Returns:

Type: `void`

Example

```
beforeSubmit(context);
```

INNER

calculateCommission(mediaBudget, bodymediaCommission, isPostBill) → {number}

Calculates the commission amount based on media budget, body media commission, and post-bill flag. The function returns the calculated commission amount.

Parameters:

Name	Type	Description
<code>mediaBudget</code>	<code>number</code>	The media budget amount.
<code>bodymediaCommission</code>	<code>number</code>	The body media commission rate.
<code>isPostBill</code>	<code>boolean</code>	Flag indicating if it is post-bill.

[View Source](#)

`ss_ue_commission_calc_line_item.js, line 492`

Returns:

The calculated commission amount.

Type: `number`

Example

```
calculateCommission(1000, 0.15, false) → 150
```

INNER

calculateDmf(mediaBudget, bodymediaCommission, isPostBill) → {number}

Calculates the DMF amount based on media budget, body media commission, and post-bill flag. The function returns the calculated DMF amount.

Parameters:

Name	Type	Description
<code>mediaBudget</code>	<code>number</code>	The media budget amount.
<code>bodymediaCommission</code>	<code>number</code>	The body media commission rate.
<code>isPostBill</code>	<code>boolean</code>	Flag indicating if it is post-bill.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 532

Returns: The calculated DMF amount.

Type: `number`**Example**

```
calculateDmf(1000, 0.15, false) --> 250
```

INNER**calculatePrebill(mediaBudget, bodymediacommission, x) → {number}**

Calculates the prebill amount. The function returns the calculated prebill amount.

Parameters:

Name	Type	Description
mediaBudget	<code>number</code>	The media budget amount.
bodymediacommission	<code>number</code>	The body media commission rate.
x	<code>number</code>	Additional multiplier or parameter.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 568

Returns: The calculated prebill amount.

Type: `number`**Example**

```
calculatePrebill(1000, 0.15, 1) --> 1000
```

INNER**extractItemDetail(currentItemValue, runtime) → {Object}**

Extracts item details from a given line item value object. The function returns an object containing extracted item details from the getLineItemValue.

Parameters:

Name	Type	Description
currentItemValue	<code>Object</code>	The current line item value object.
runtime	<code>Object</code>	The runtime object for getting script parameters.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 694

Returns: An object containing extracted item details.

Type: `Object`**Example**

```
extractItemDetail(currentItemValue, runtime) --> { mediaBudget: '1', itemLineCommission: '1', itemLineDmf: '1', itemLineOverrideMediaCalc: '1', currentLineItemValue: '1', ... }
```

INNER**findExistingCommissionDmfLine(currentRecord, sublistName, groupNumber, mediaCommissionItem, mediaDmfItem) → {number}**

Finds an existing commission or DMF line based on group number and item IDs. The function also checks if the line item is already commissionable or DMF. The function returns the line index if found, or -1 if not found.

Parameters:

Name	Type	Description
currentRecord	<code>Object</code>	The current NetSuite record being processed.
sublistName	<code>string</code>	The name of the sublist.
groupNumber	<code>number</code>	The group number to look for.
mediaCommissionItem	<code>number</code>	The ID of the media commission item.
mediaDmfItem	<code>number</code>	The ID of the media DMF item.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 768

Returns: The line index if found, or -1 if not found.

Type: `number`**Example**

```
findExistingCommissionDmfLine(currentRecord, sublistName, groupNumber, mediaCommissionItem, mediaDmfItem) --> -1
```

INNER**getLineItemValue(newRecord, sublistId, line) → {Object}**

Retrieves line item values from a given record and line index. The function returns an object containing line item values.

Parameters:

Name	Type	Description
newRecord	Object	The NetSuite record object.
sublistId	string	The ID of the sublist.
line	number	The line number to retrieve values from.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 598

Returns:

An object containing line item values.

Type: object

INNER**getMonthNumber(dateString) → {number}**

Gets the month number from a given date string. The function returns the month number (1-12) from the date string.

Parameters:

Name	Type	Description
dateString	string	The date string to extract the month number from.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 955

Returns:

The month number (1-12).

Type: number

Example

getMonthNumber(dateString) --> 6

INNER**getNextAvailablePdfNumber(uniquePdfGroupNumbers) → {number}**

Determines the next available PDF group number based on existing numbers for sequencing. The function returns the next available PDF group number

Parameters:

Name	Type	Description
uniquePdfGroupNumbers	Array	An array of existing PDF group numbers.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 854

Returns:

The next available PDF group number.

Type: number

Example

getNextAvailablePdfNumber(uniquePdfGroupNumbers) --> 6

INNER**getUniquePdfGroupNumbersFromLines(newRecord, sublistId) → {Array}**

Gets unique PDF group numbers from line items in a given sublist. The function returns an array of unique PDF group numbers.

Parameters:

Name	Type	Description
newRecord	Object	The NetSuite record object.
sublistId	string	The ID of the sublist.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 821

Returns:

An array of unique PDF group numbers.

Type: Array

Example

getUniquePdfGroupNumbersFromLines(newRecord, sublistId) --> ['1', '2', '3', '4', '5']

INNER**setPDFGroupNumber(currentRecord, sublistName, lineIndex, groupNumber) → {string}**

Sets the PDF group number for a given line item. The function returns the set or existing PDF group number. The function also checks if the PDF group number is null, undefined, empty, or 'unset' and sets the PDF group number if true.

Parameters:

Name	Type	Description
currentRecord	Object	The current NetSuite record being processed.
sublistName	string	The name of the sublist being modified.
lineIndex	number	The line index to update.
groupNumber	number	The group number to assign.

[View Source](#)

ss_ue_commission_calc_line_item.js, line 728

Returns:

The set or existing PDF group number.

Type: string

Example

```
setPDFGroupNumber(currentRecord, sublistName, lineIndex, groupNumber) -> '1'
```

Documentation generated by [JSDoc 4.0.3](#)



Documentation

MODULES

ss_cs_clear_pdf_group_line
ss_ue_commission_cal_c_line_item

MODULE

ss_cs_clear_pdf_group_line

Members

INNER

currentLine

Get the current sublist index

[View Source](#)

ss_cs_clear_pdf_group_line.js, line 76

Type: `number`

Example

```
var currentLine = rec.getCurrentSublistIndex({  
    sublistId: sublistId  
});
```

Type: `HTMLElement`

makeCopyButton

take the element with the id 'item_copy' and assign it to the variable makeCopyButton

[View Source](#)

ss_cs_clear_pdf_group_line.js, line 22

Example

```
var makeCopyButton = document.getElementById('item_copy');
```

Type: `Record`

rec

Get the current record

[View Source](#)

ss_cs_clear_pdf_group_line.js, line 56

Example

```
var rec = currentRecord.get();
```

Type: `string`

sublistId

The sublist id

[View Source](#)

ss_cs_clear_pdf_group_line.js, line 66

Example

```
var sublistId = 'item';
```

Methods

INNER

onclick() → {boolean}

Click handler for the makeCopyButton

[View Source](#)

ss_cs_clear_pdf_group_line.js, line 33

Throws: Error in Make Copy Click Handler

Type: `error`

Returns: Type: `boolean`

Example

```
makeCopyButton.onclick = function() {  
    try {  
        // Perform the line copy  
        item_machine.copyline();  
        setWindowChanged(window, true);  
        var rec = currentRecord.get();  
    } catch (e) {  
        // Handle error  
    }  
}
```

MEMBERS

currentLine
makeCopyButton
rec
sublistId

METHODS

onclick
pagelink
setCurrentSublistValue

INNER

pageInit(context)

Client script pageInit function

Parameters:

Name	Type	Description
context	object	

[View Source](#)

ss_ss_clear_pdf_group_line.js, line 13

INNER

setCurrentSublistValue()

Clear the 'custcol_ss_pdf_group' field on the new line

[View Source](#)

ss_ss_clear_pdf_group_line.js, line 90

Example

```
rec.setCurrentSublistValue({  
  sublistId: sublistId,  
  fieldId: 'custcol_ss_pdf_group',  
  value: ''  
});
```

Documentation generated by [JSDoc 4.0.3](#)