University of Colorado
Boulder | Colorado Springs | Denver | Anschutz Medical Campus

OnBase
by Hyland

OnBase Guide - Unity Script - MIKG Formatter

Goal: To format all or selected rows from a multi-instance keyword group, using a Workflow script.

Complexity Level: Departmental Workflow Developers

7/7/2023

Table of Contents

Background

Multi-Instance Keyword Groups (MIKGs) allow for multiple keyword records to be added to a document, while maintaining the relationship between values in each row.

Outside of a MIKG, multiple values for keyword types can be added to a document. For example, if a request refers to more than one semester it could have two Term Code values (2237 and 2241).

However, if multiple values need to be stored in a way that represents multiple records, a MIKG is required so the values aren't "mixed up". For example, on a form where students request enrollment in one or more classes, it's important that the subject, catalog # and section number align.

Forms have built-in functionality to support MIKGs (repeaters/tables) but using the data in Workflow can be more challenging. The script documented here allows for easier formatting of MIKG data for use in Workflow, especially in notifications where you want the data formatted in a way that's easy to read and understand.

Prerequisites

You must have OnBase Studio installed and know how to configure a life cycle. Refer to the OnBase Client Guides for instructions on installation and to the Workflow MRG for more details as necessary.

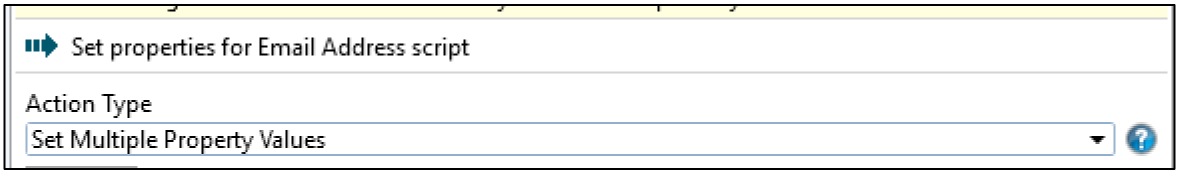
Refer to the handbook for general information on Unity script usage in Workflow. You should be familiar with using properties in Workflow as well.

Contact UIS_DM_Support@cu.edu for assistance if needed.

Workflow Configuration

Set Required Property Values for Script Inputs

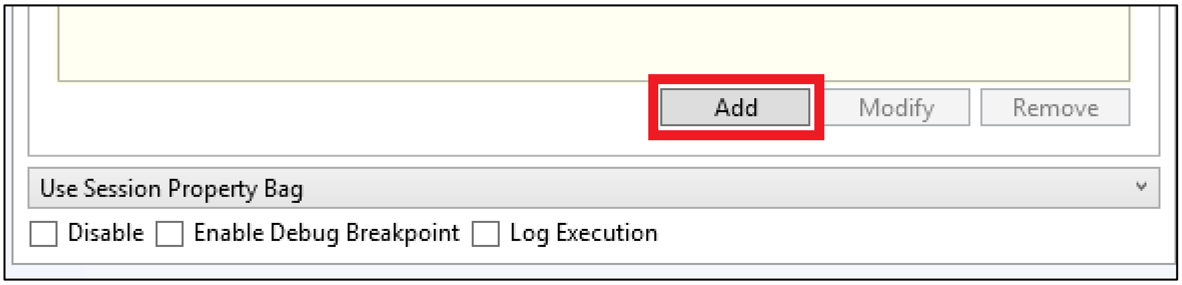
Create an action and choose the **Set Multiple Property Values** action type.



Set properties for Email Address script

Action Type: Set Multiple Property Values

Click **Add** at the bottom of the panel at the right side to add each new property value.



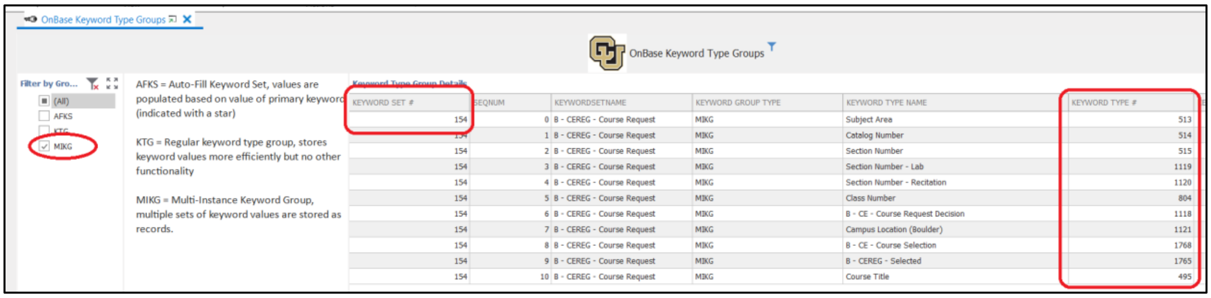
Use Session Property Bag

□ Disable □ Enable Debug Breakpoint □ Log Execution

These inputs are required for the MIKG Formatter script to run.

- propMIKGnum
- propFormatString

The **propMIKGnum** property needs to be set to the unique ID assigned to the MIKG in Configuration. Our recommendation is to use the *OnBase Keyword Type Groups* dashboard in the *Configuration - Keywords* category.


OnBase Keyword Type Groups
OnBase Keyword Type Groups
Filter by Gro...
Keyword Type Group Details

■ (All)
□ AFKS
□ KTG
☑ MIKG

AFKS = Auto-Fill Keyword Set, values are populated based on value of primary keyword (indicated with a star)

KTG = Regular keyword type group, stores keyword values more efficiently but no other functionality

MIKG = Multi-Instance Keyword Group, multiple sets of keyword values are stored as records.

KEYWORD SET #	SEQNUM	KEYWORDSETNAME	KEYWORD GROUP TYPE	KEYWORD TYPE NAME	KEYWORD TYPE #
154	0	B - CEREG - Course Request	MIKG	Subject Area	513
154	1	B - CEREG - Course Request	MIKG	Catalog Number	514
154	2	B - CEREG - Course Request	MIKG	Section Number	515
154	3	B - CEREG - Course Request	MIKG	Section Number - Lab	1119
154	4	B - CEREG - Course Request	MIKG	Section Number - Recitation	1120
154	5	B - CEREG - Course Request	MIKG	Class Number	804
154	6	B - CEREG - Course Request	MIKG	B - CE - Course Request Decision	1118
154	7	B - CEREG - Course Request	MIKG	Campus Location (Boulder)	1121
154	8	B - CEREG - Course Request	MIKG	B - CE - Course Selection	1768
154	9	B - CEREG - Course Request	MIKG	B - CEREG - Selected	1765
154	10	B - CEREG - Course Request	MIKG	Course Title	495

Keep in mind that with new MIKG and keyword type configuration, the ID may be different between environments.

The **propFormatString** property represents the pattern of how each row in the MIKG should be returned in the output. This is similar to configuring an auto-name string on a document type. The *OnBase Keyword Type Groups* dashboard will be helpful for this as well to get the necessary keyword type IDs. propFormatString should reference keyword types by their ID, with curly braces ({ and }) around each keyword ID.

If the desired output looks like:

SOCY 1001-100, Intro to Sociology, Class: 12345

This is represented by the keyword types and static text of:

{Subject Area} {Catalog Number}-{Section Number}, {Course Title}, Class: {Class Number}

Replacing keyword type names with their IDs, we would use the following as propFormatString:

{513} {514}-{515}, {495}, Class: {804}

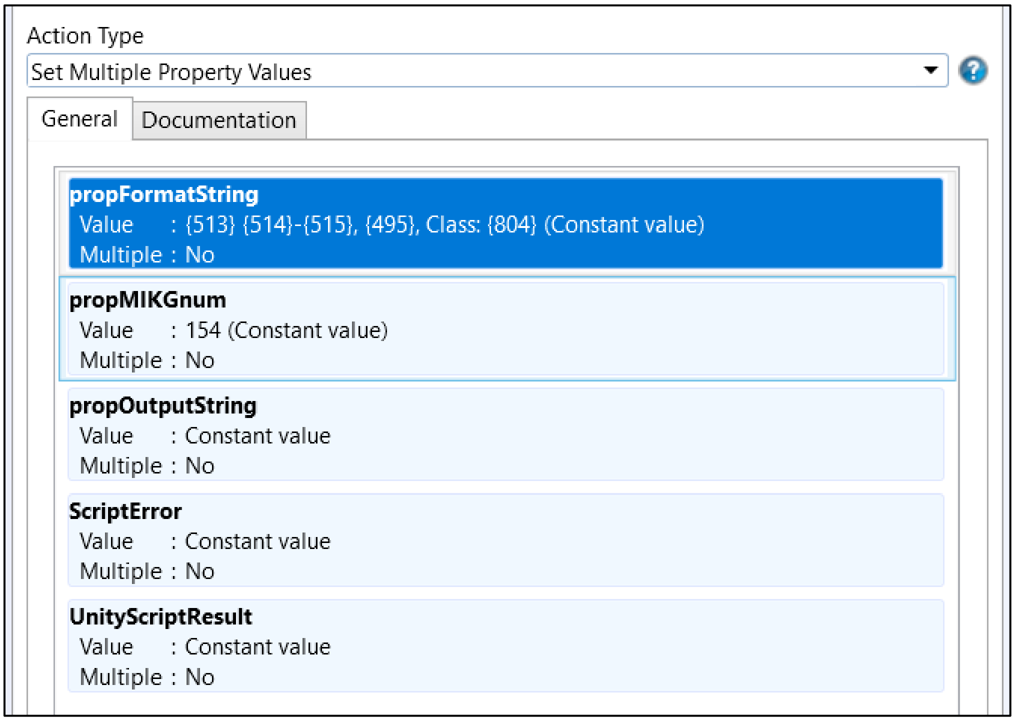
If you want to set capitalization on certain keyword values, you can include C (capitalize everything, default for most keyword types already), W (capitalize words), or L (lowercase everything) within the curly braces before and after the keyword ID. Ex:

- {C495C}
- {L495L}
- {W495W}

Nothing except for keyword IDs or C, L, W should be inside curly braces in propFormatString.

The action to set input property values should look something like this:

- propFormatString = {513} {514}-{515}, {495}, Class: {W804W}
- propMIKGnum = 154
- propOutputString = (blank)
- ScriptError = (blank)
- UnityScriptResult = (blank)



Action Type
Set Multiple Property Values

propFormatString
Value : {513} {514}-{515}, {495}, Class: {804} (Constant value)
Multiple : No

propMIKGnum
Value : 154 (Constant value)
Multiple : No

propOutputString
Value : Constant value
Multiple : No

ScriptError
Value : Constant value
Multiple : No

UnityScriptResult
Value : Constant value
Multiple : No

Set Optional Property Values

Three additional properties are available for optional use with this script.

- propSeparator
- propSearchKWType
- propSearchKWValue

The **propSeparator** property can be set if you want to specify the separator used between MIKG rows in the script's output. By default, " | " is used. Specifying a separator would be helpful if you want to use the pipe character as part of the format and also want to manipulate the formatting of the script output.

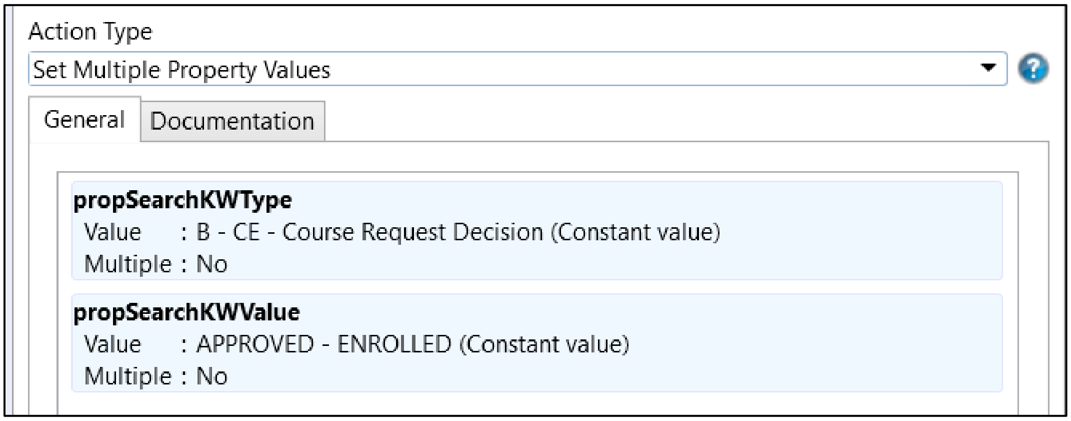
The **propSearchKWType** property can be set to the name(s) of the keyword type(s) you want to search on to limit which rows are returned. If using more than one keyword type to search, separate the names with "|" (no space), ex: propSearchKWType = First Name|Last Name|Date of Birth.

The **propSearchKWValue** property can be set to the value(s) of the other keyword type(s) you are using to search. The number of keyword values must match the number of keyword types. Separate the values with "|" (no space), ex: propSearchKWType = John|Smith|1901-01-01. For dates, use the format yyyy-mm-dd.

For example, if I only want rows from this MIKG where the request has been approved, I would set these additional property values:

- propSearchKWType = B - CE - Course Request Decision
- propSearchKWValue = APPROVED - ENROLLED

Since no separator is specified, the default will be used.

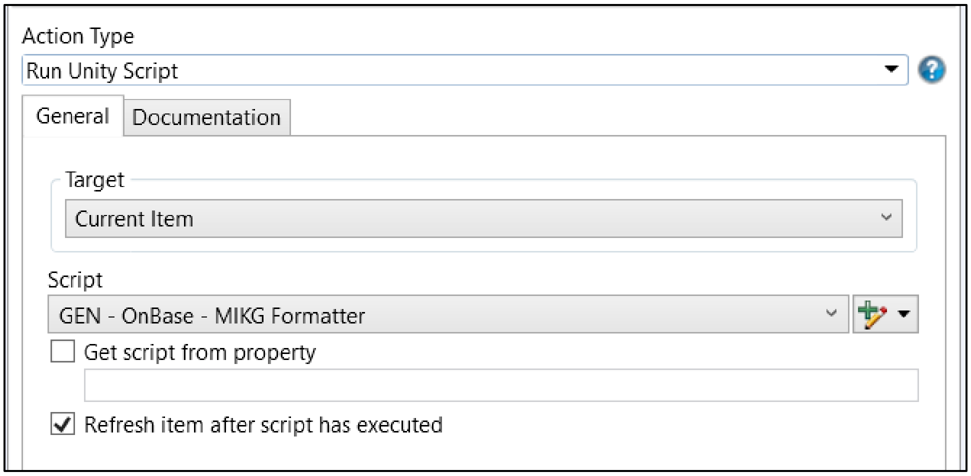


propSearchKWType
Value : B - CE - Course Request Decision (Constant value)
Multiple : No

propSearchKWValue
Value : APPROVED - ENROLLED (Constant value)
Multiple : No

Run the Script

Create a "Run Unity Script" action. Select "GEN - OnBase - MIKG Formatter" from the list of available scripts. Check the box to Refresh item after script has executed.


Action Type
Run Unity Script
General | Documentation
Target
Current Item
Script
GEN - OnBase - MIKG Formatter
☐ Get script from property
☑ Refresh item after script has executed

Use the Script Results

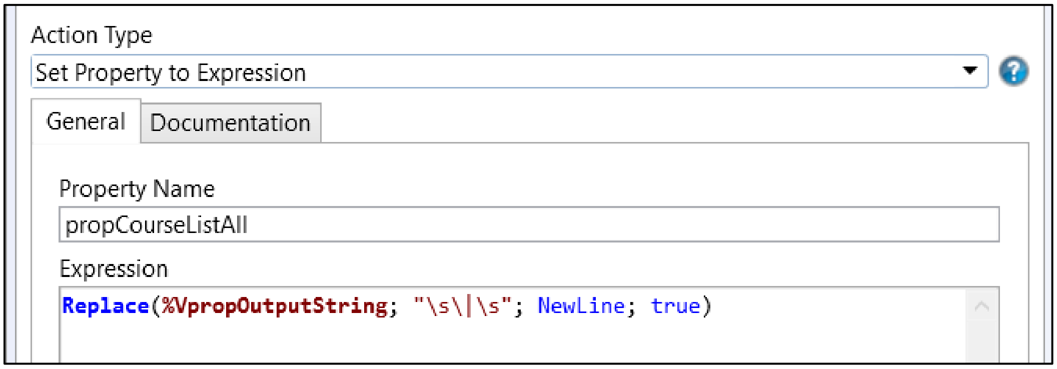
Once the script has set the output property value, **propOutputString**, you can then use it in whatever rules or actions suit your process. In most cases this will probably be used as a token in a notification template.

If the supplied criteria return more than one result from the MIKG, they will be listed with a separator between rows. If a separator was defined using propSeparator, that would be used in place of " | ".

If you'd like to put each row on a new line, you can use the Set Property to Expression action type to reformat the property using the expression:

```
Replace(%VpropOutputString; "\s\|\s"; NewLine; true)
```

This expression assumes the use of the default separator; adjust the Replace() function as needed if using a different separator.


Action Type
Set Property to Expression
General | Documentation
Property Name
propCourseListAll
Expression

```
Replace(%VpropOutputString; "\s\|\s"; NewLine; true)
```