Created by MyCRM Ltd

eWorkflow 2011

User and function guide

Alistair Dickinson
5/8/2012

Function Description

The main purpose of this document is to help end users gain insight and knowledge when using the latest release of eWorkflow for Microsoft Dynamics CRM 2011, from MyCRM.

The new eWorkflow function Library contains 36 new workflow functions that can be used to enhance business automation and productivity.

Table of Contents

eWorkflow – Marketing

In the new eWorkflow Marketing Helper Library we have included 4 additional function calls that can be used against an Account, Contact or Lead.

Function – GetLastCampaignForMember

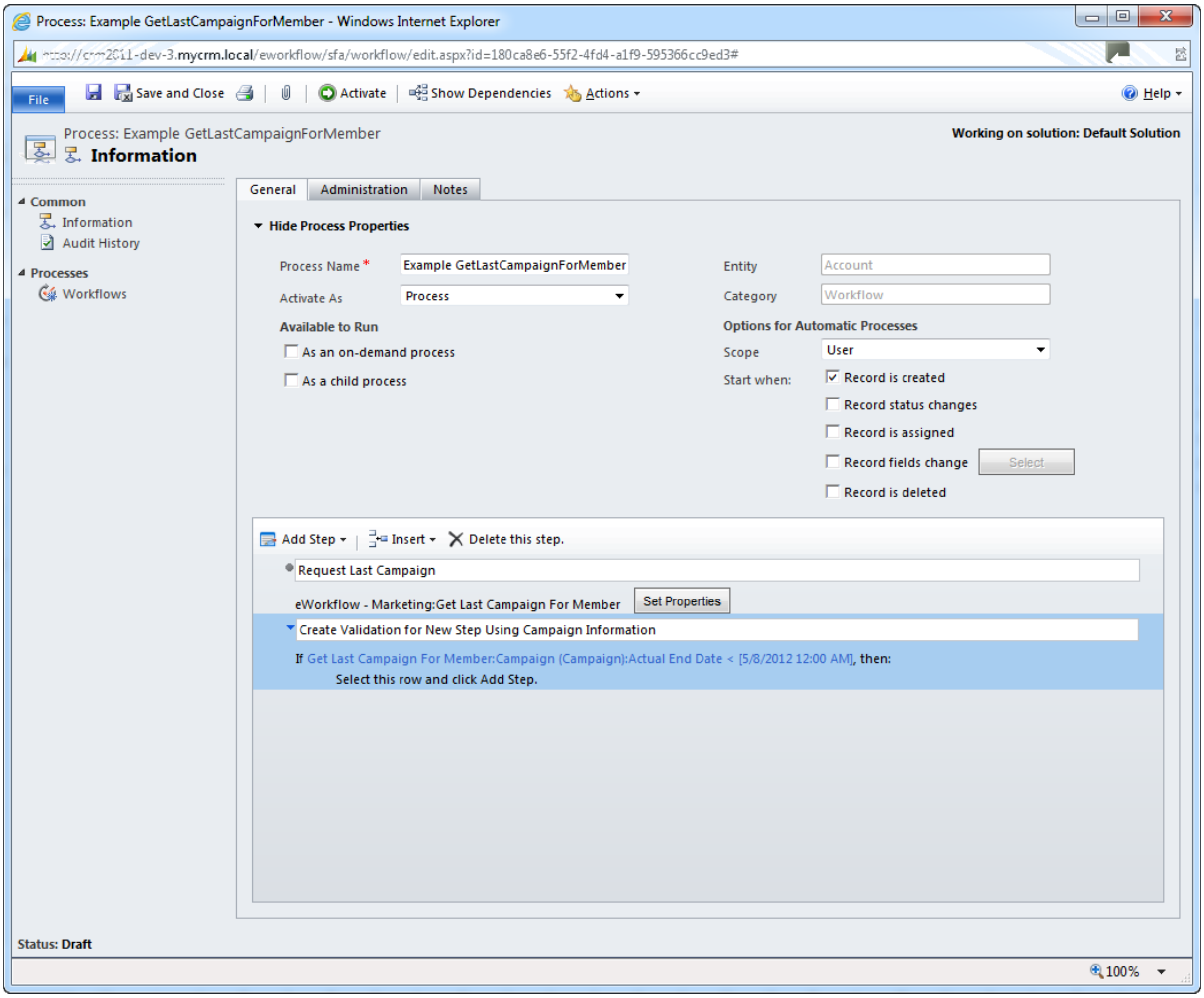
This function call enables a user to return the last campaign that an Account, Contact or Lead was entered into. By having this information via a workflow a business process can easily be created to see if a customer type is responding to campaigns or if a customer has been involved in too many campaigns.

This function will return the actual campaign or return null if no campaigns exist.

If a campaign is returned information about that campaign can be used on the Account, Contact or Lead Record an example may be the return of the campaign last date

Function Call :- Input
Select Contact
Select Account
Select Lead

Example 1.1



The above example shows the call to the last campaign which returns a full Campaign data set that can then be used for validation in the workflow.

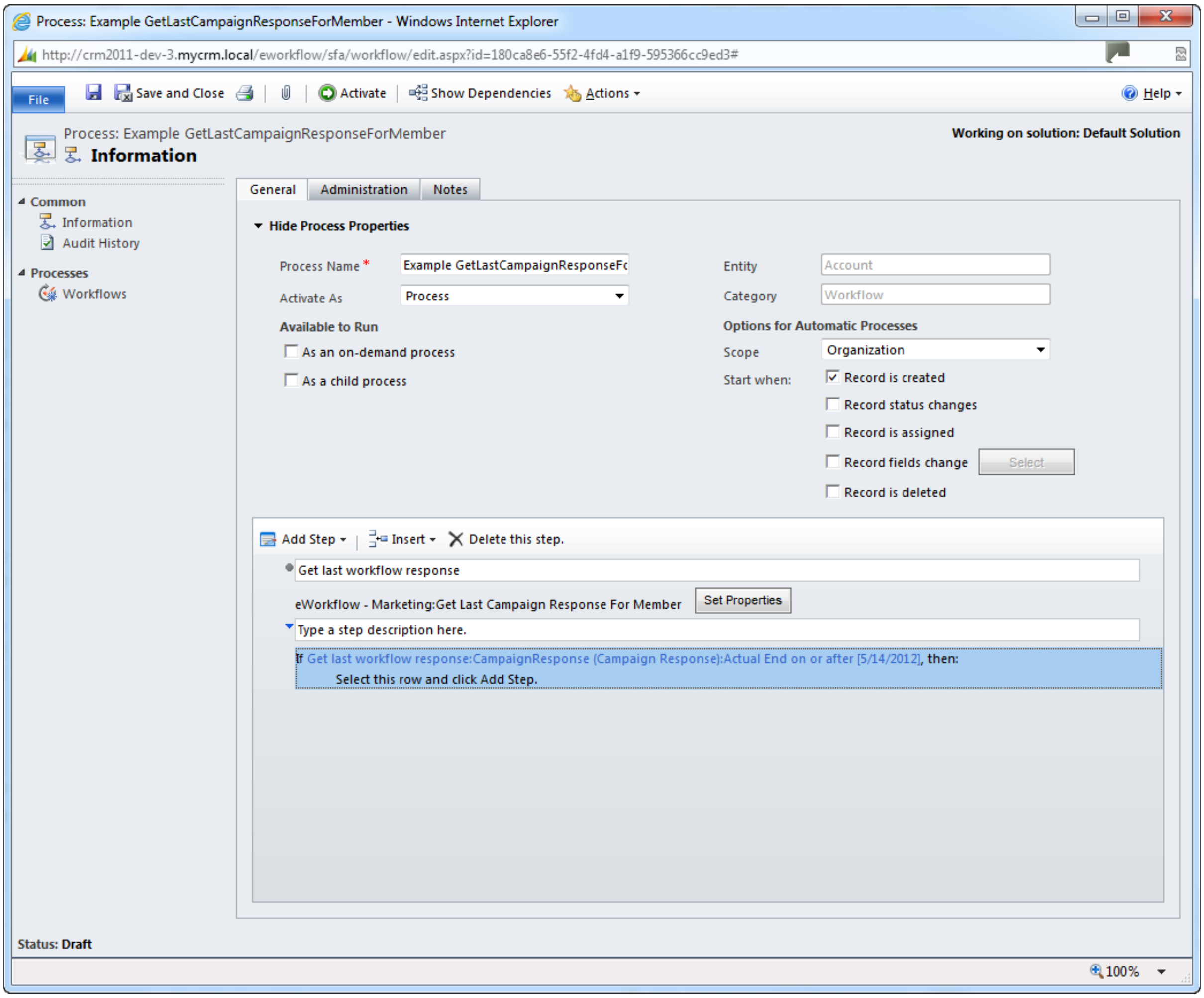
Function Call :- Output
All Campaign Fields on the Published Form

Function - GetLastCampaignResponseForMember

This function call can be used in a workflow process to return the last campaign activity response that was created for an Account, Contact or Lead record. The data from the Campaign Response Activity can be used in verification of the workflow process and steps.

Function Call :- Input
Select Contact
Select Account
Select Lead

Example 2.1



You can see as an example we have used the function to get the last activity and then used a check condition on one of the activity end date, to then create or update a record.

Function Call :- Output
All Campaign Response Fields on the Published Form

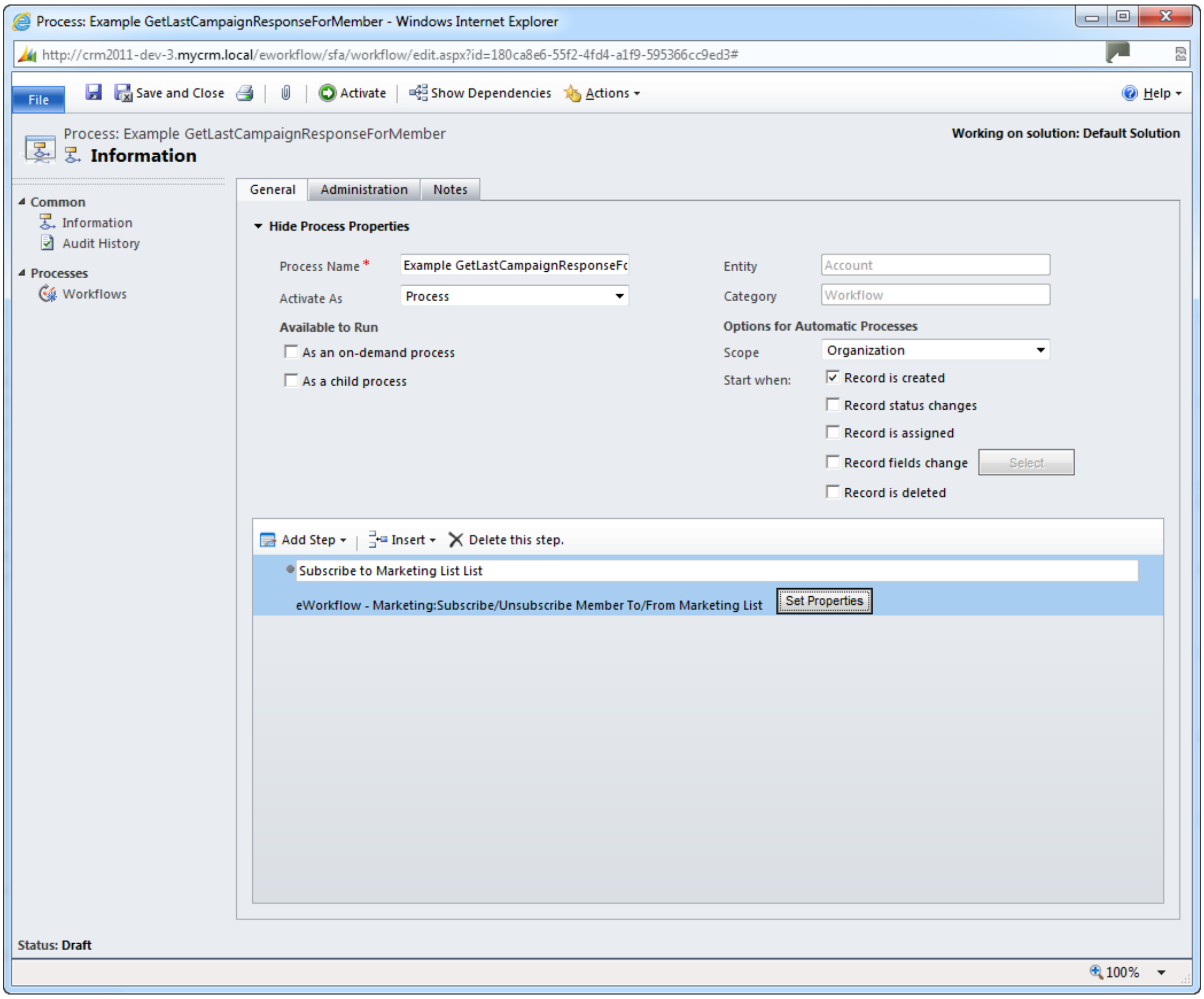
Function – MarketingSubscribeUnSubscribeMember

This function call is a great asset to enable the creation of Marketing Lists through Workflow, Using this you can automate the adding of members i.e. Accounts, Contacts or Leads to a defined list via workflow, you can also use the **IsAMarketingListMember** function to see if a lead or contact is already on a list.

The advantage of having this function means that when new Accounts, Contacts or Leads are created in the CRM these records are added to a list or can be added to separate lists, without delay or expecting Marketing to manually build lists for new contacts.

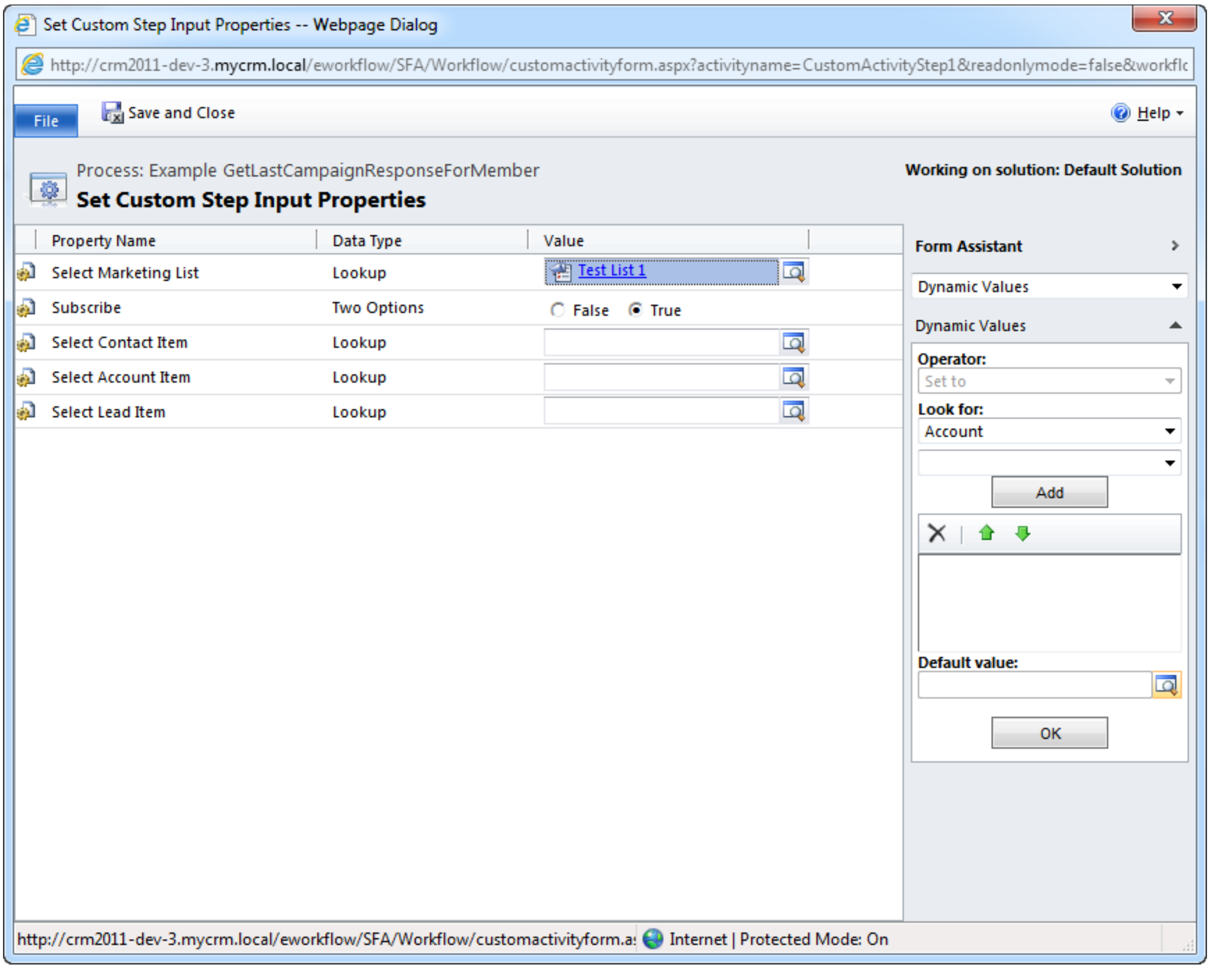
Function Call :- Input
Select Marketing List
Subscribe (True / False)
Select Contact
Select Account
Select Lead

Example 3.1



To set this function the properties will need to be selected for the Marketing List and if this is an add or a remove.

Example 3.2



The same function is used to remove members from a Marketing list so if a contact or an Account or even a lead is disabled the automation of Marketing List management is done in a workflow process.

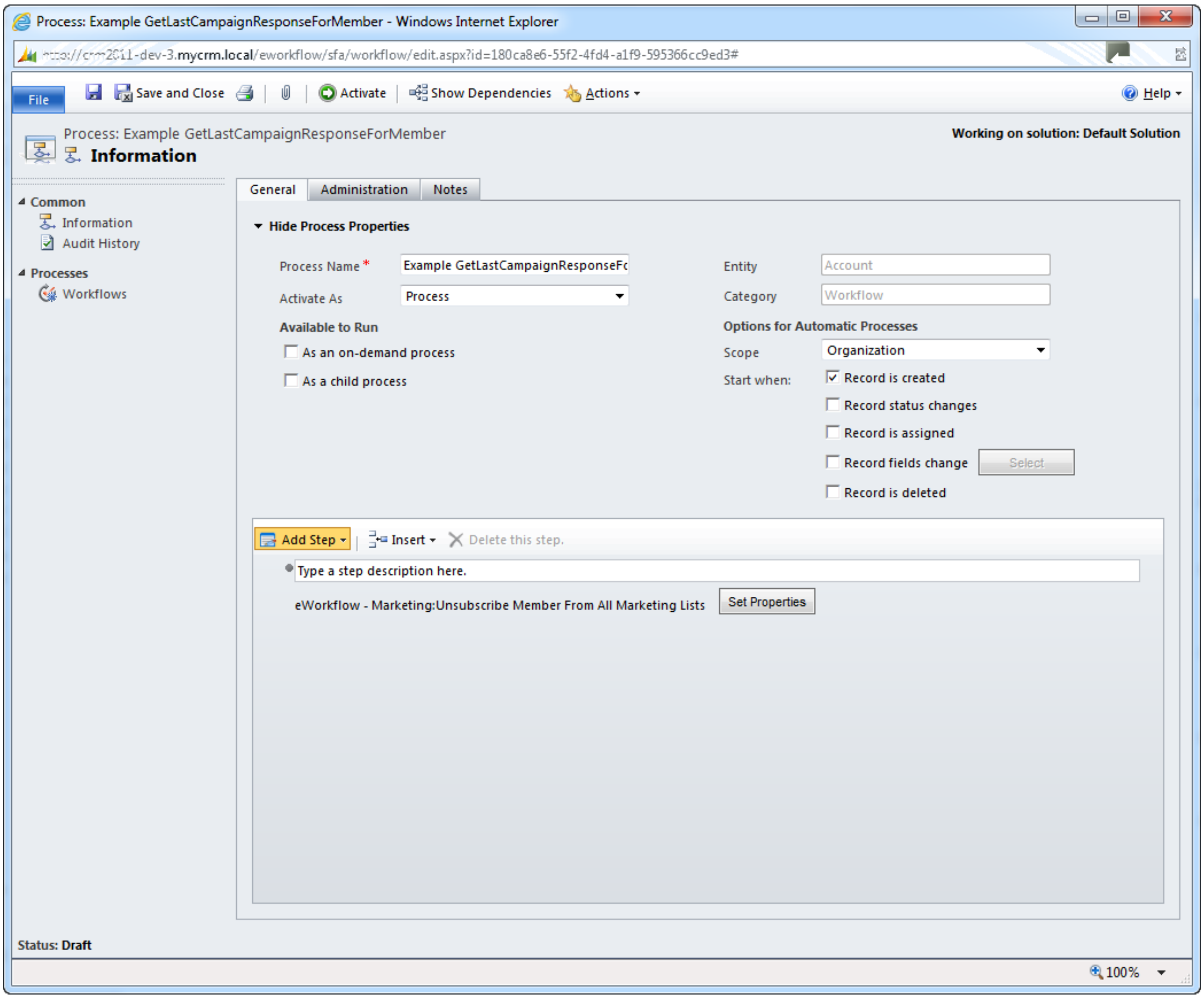
Function – UnsubscribeMemberFromAllMarketingLists

Another great marketing function that enhances the management of Marketing Lists in CRM for Account, Contacts or Leads is the ability to enable a single function call to be used against these record types and have them removed from any Marketing List in the CRM System.

This is likely to be required if an Account, Contact or Lead is disabled, as an example this is likely to be used when a Lead is qualified and a lead becomes a contact and a requirement to move to a different list is needed.

Function Call :- Input
Select Contact
Select Account
Select Lead

Example 4.1



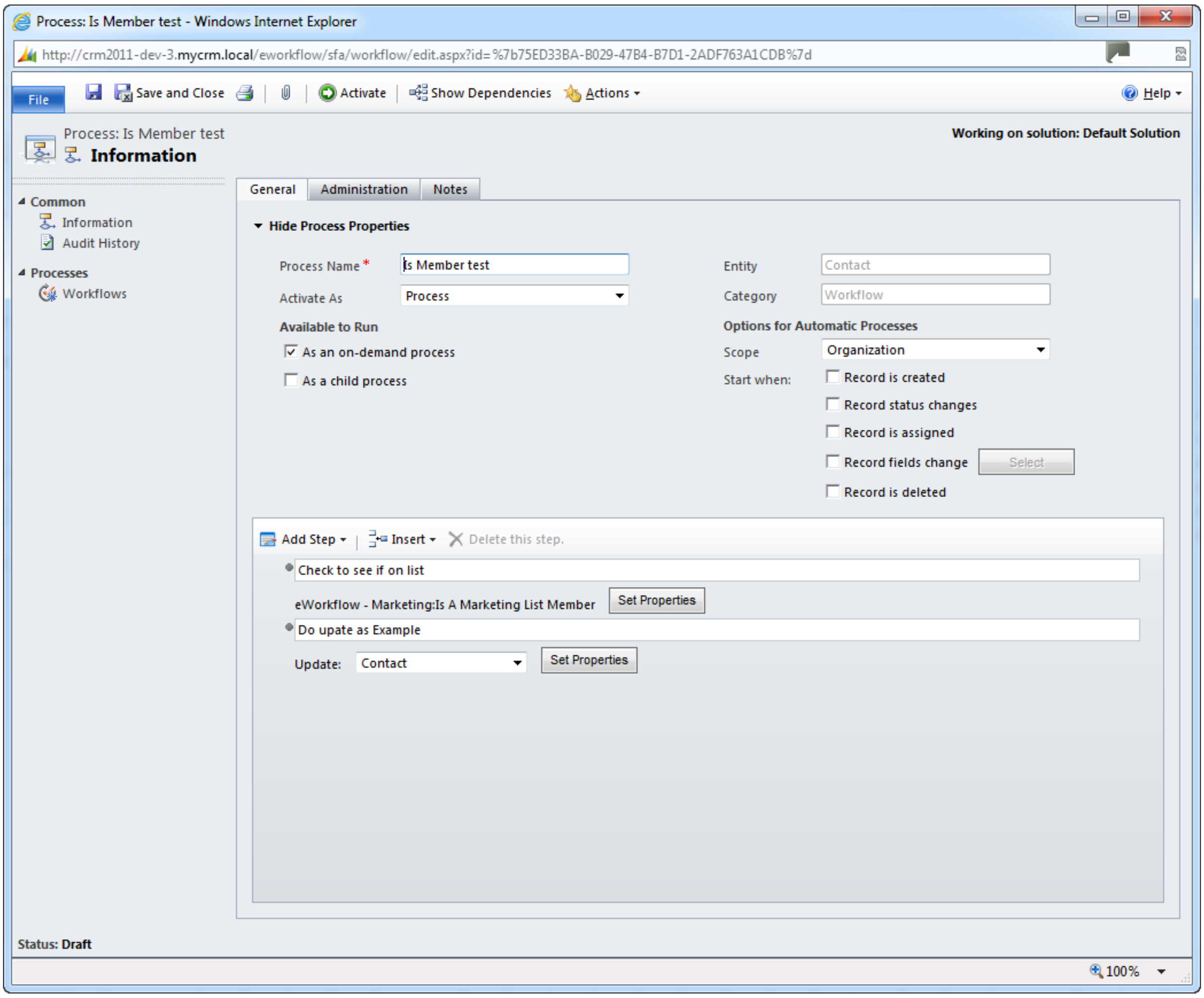
Function – IsAMarketingListMember

This function can be used to check if an Account, Contact or Lead is already on an existing Marketing list. This helps with business process where one of the Account, Contact or Lead records changes its status or in the case of a Lead progresses through the Sales Cycle.

The workflow function will return a True or False and from there a process can be completed.

Function Call :- Input
Select Contact
Select Account
Select Lead
Marketing List

Example 5.1



eWorkflow – Calculations

Our Calculation set extends the workflow capabilities even further with the option to perform standard calculations within workflow when either a record is created, updated or a field is changed. A number of new workflow function calls are available

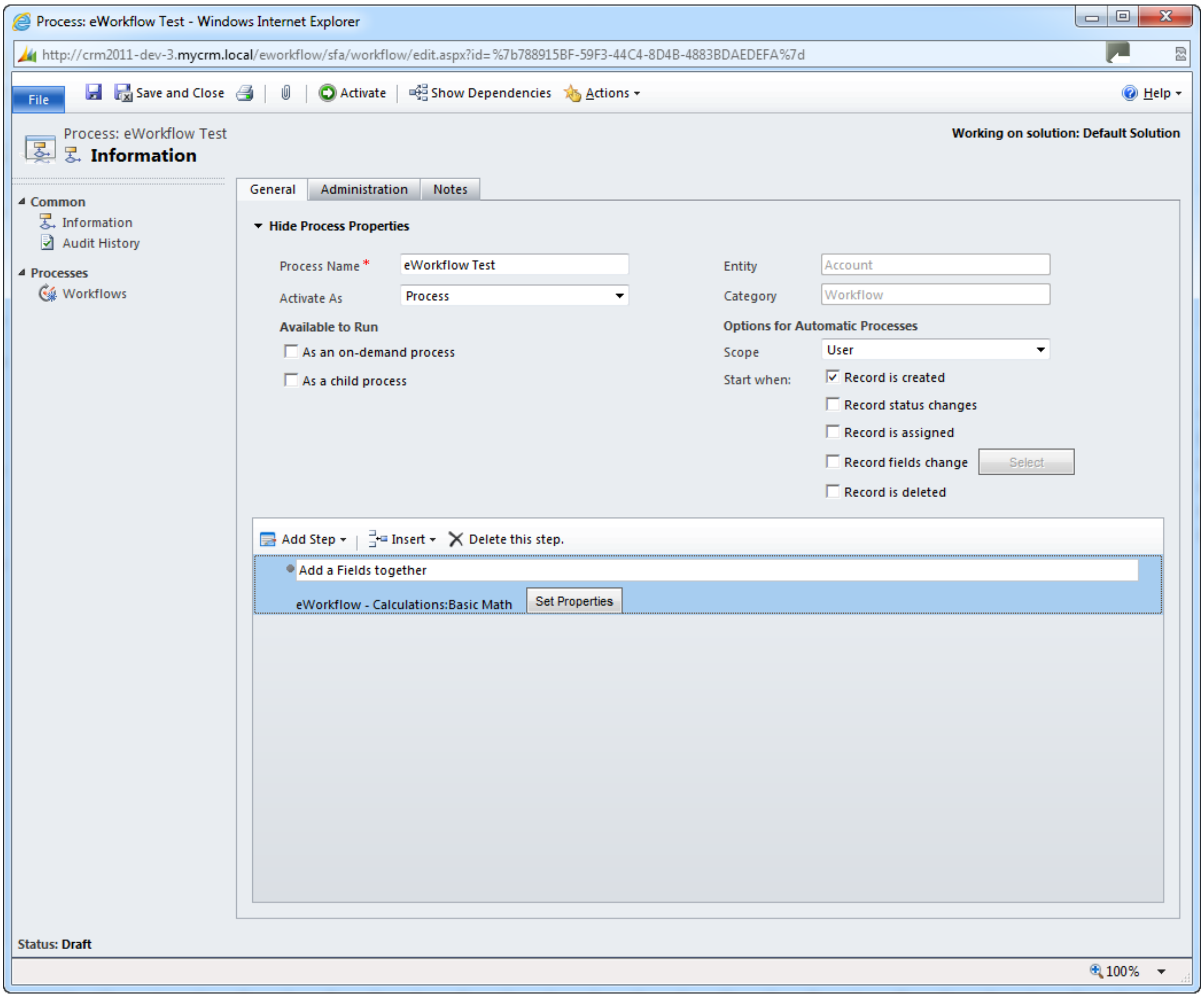
Function – BasicMath

This function allows you to automate calculation on different record types, as an example a credit limit could be set on an account and the available balance be set by taking the total of unpaid invoices minus the limit

This would be just one example but this workflow call extends number manipulation even further with CRM workflow.

Function Call :- Input
First Number
Second Number
Symbol

Example 6.1



Basic math calculation enables +, - / and * functions to be applied to number fields.

Function Call :- Output
Result (Number)

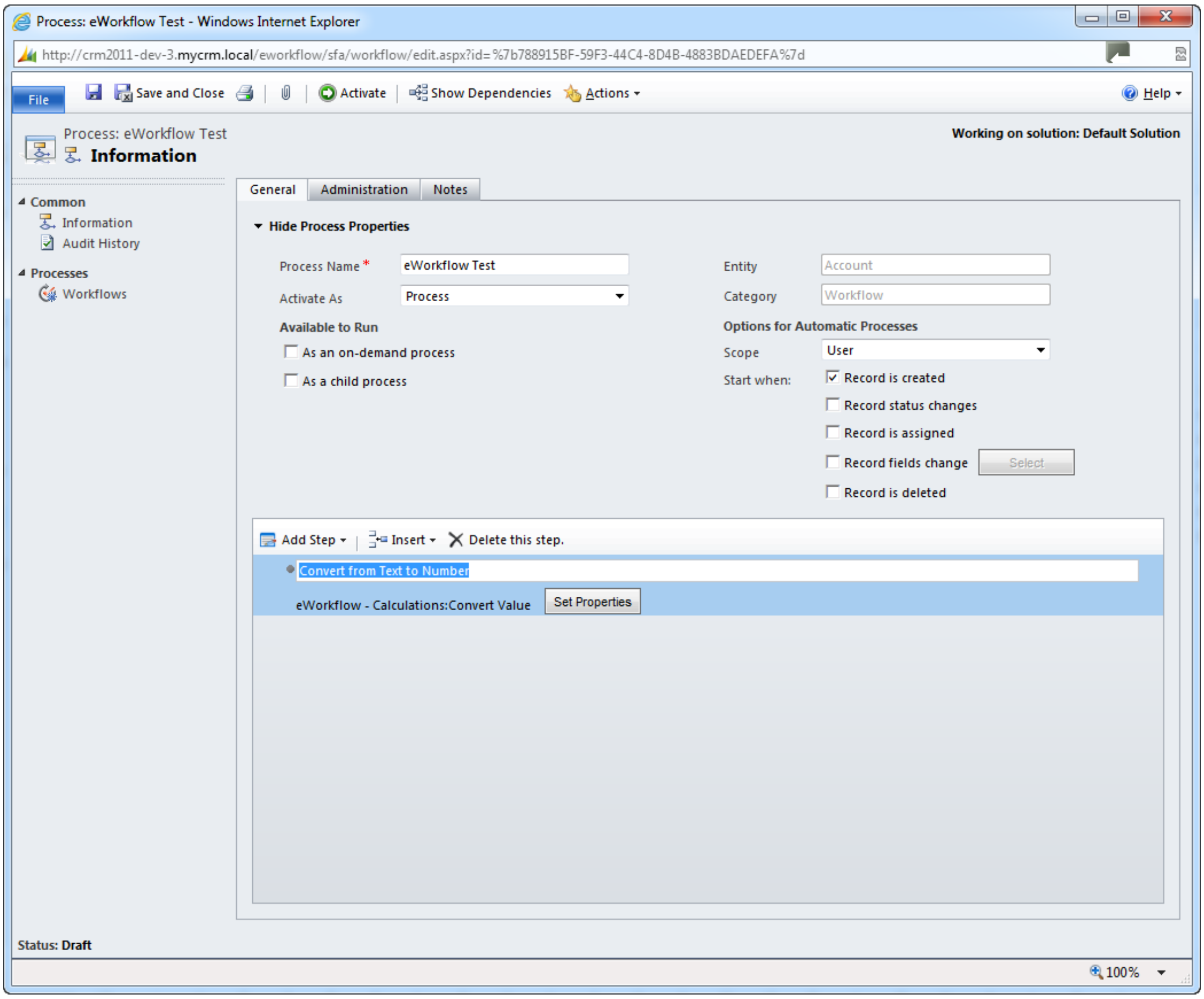
Function – ConvertTextToNum

This workflow function enables the conversion of a text string to a number, with several CRM number types to choose as output.

The number based in the Text fields is output in the following formats after the function is run

Function Call :- Input
Value to Convert

Example 7.1



Function Call :- Output
Decimal (Number)
Money (Money)
Float (Float)
Rounding Number
Truncated Number
Processing Error (error)

Function – EquationSolver/SumUp

This great extended function enables an end user to perform complex or standard calculations on a given number in the CRM record where the workflow is applied

This work flow allows you to enter a calculation either fixed or based on values from other CRM fields ad return a calculated value which can be used to update different field in the record.

An example may be calculating a commission value from the value of an opportunity or an Order.

Function Call :- Input
Formula (text)
@a (text)
@b (text)
@c (text)
@d (text)
@e (text)
@f (text)
@g (text)
@h (text)
@i (text)
@x (text)
@y (text)
@z (text)

The Equation Solver uses Reverse Polish Notation to perform calculations based on the input values. It supports multiple Operators, Constants and Functions:

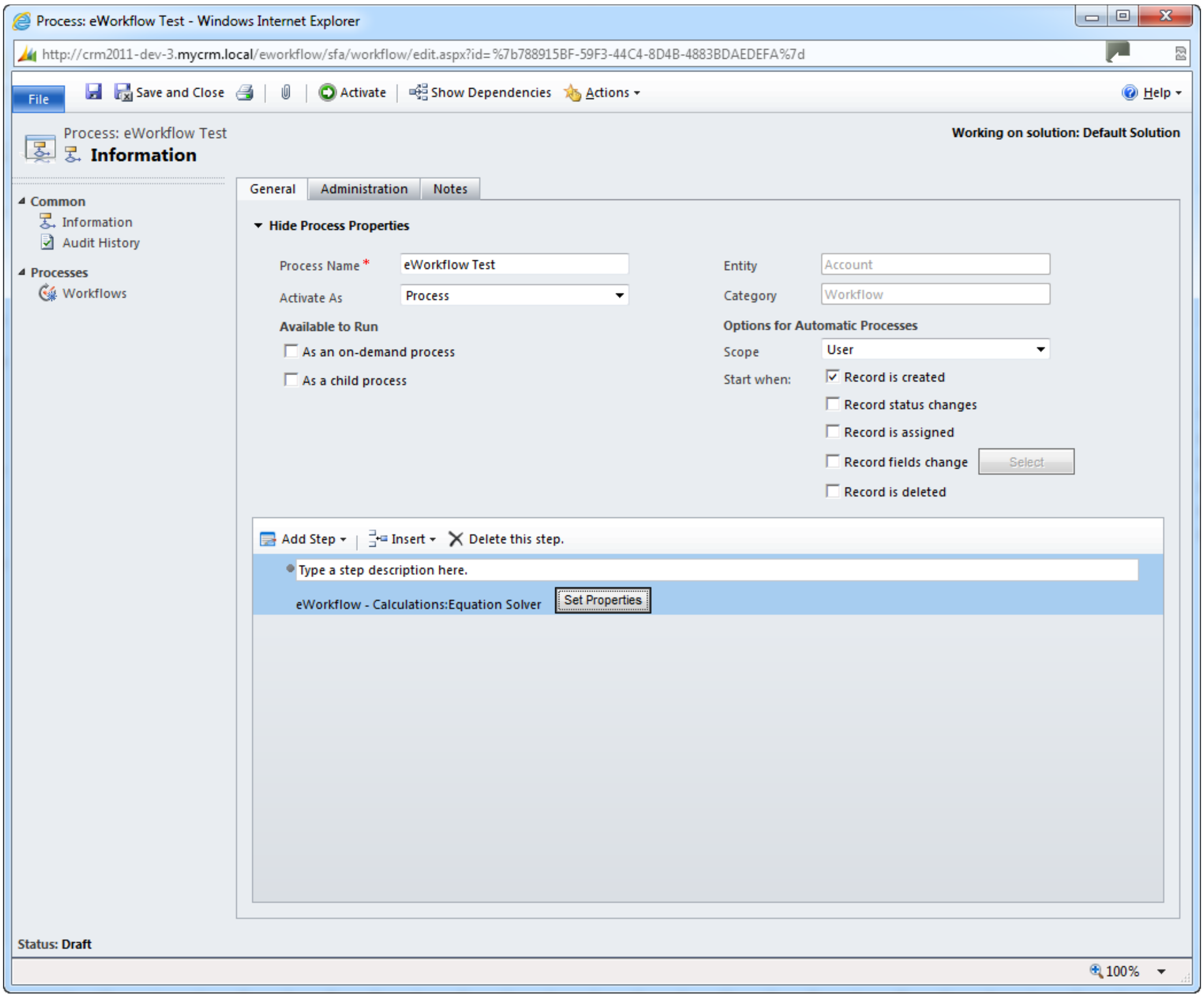
Operator	Example	Function
	12	Add: 1+2 = 3
-	3-4	Subtract: 3-4 = -1
*	1*2	Multiply: 1*2 = 2
/	2/1	Divide: 2/1 = 2
%	20%6	Mod: 20 mod 6 = 2
^	2^3	Power: 2 to the 3rd power = 8

Constant	Value
e	2.71828182845904523536
pi	3.14159265358979323846
rand	A Random Number between 0 and 1

Function	Description
abs	Absolute Value
acos	Arccosine
asin	Arcsine
atan	Arctangent
cos	Cosine
degrees	Convert radians to degrees
even	Rounds up to the nearest even integer
fact	Factorial – Example: fact(5) = 5 * 4 * 3 * 2 * 1
gcf	Greatest Common Factor (Included as an example math function)
log	log(number,base)
log10	Log Base 10 of a number
max	Maximum Value – Can take multiple parameters
min	Minimum Value – Can take multiple parameters
neg	Internal Use A negative number neg 1 = -1
odd	Rounds up to the nearest odd integer
power	power(a,b) = a^b
radians	Convert degrees to radians
randbetween	A random number between the two numbers passed – Example: randbetween(1,10)

round	Rounds the number 1 to the number 2's number of decimal places.
sign	1 if positive, 0 if 0, -1 if less than 0
sin	Sine
sqrt	Square Root
tan	Tangent
trunc	Remove anything after the decimal place.

Example 8.1



Function Call :- Output
String (Text)
Money (Money)

Float (Float)
Rounding Number
Truncated Number
Processing Error (error)
Error Message (text)

Function – CalculateMaximum

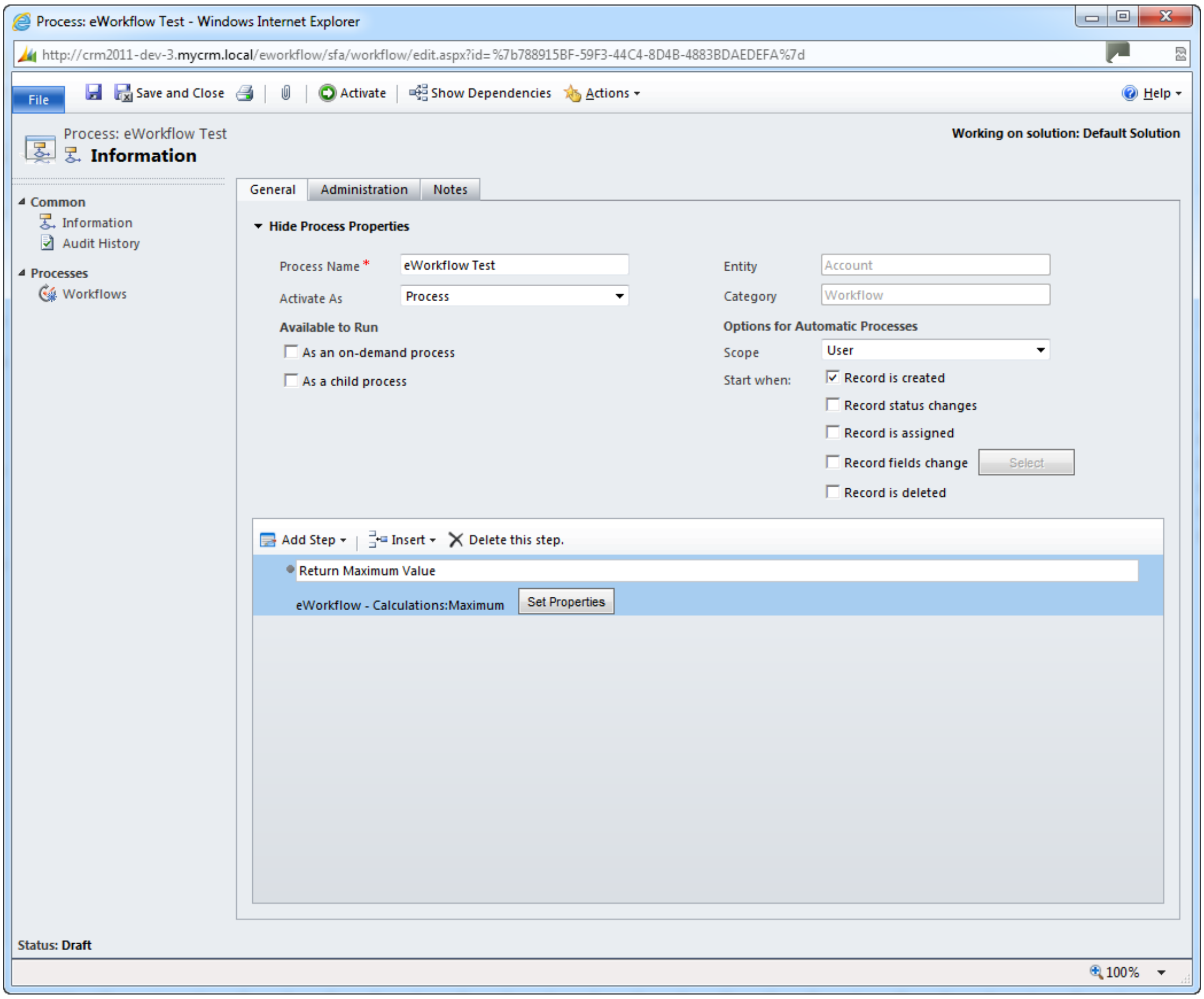
This function compares two numbers and returns the highest value.

An example might be that as Part of an SLA the Maximum wait time is 8 Hours, a business user creating a workflow you may want a wait statement on the account record that polls for the maximum amount of Time.

Another example may be a validation check for a credit limit, and if the Maximum is returned then a task is performed as an alert.

Function Call :- Input
First Number
Second Number

Example 9.1



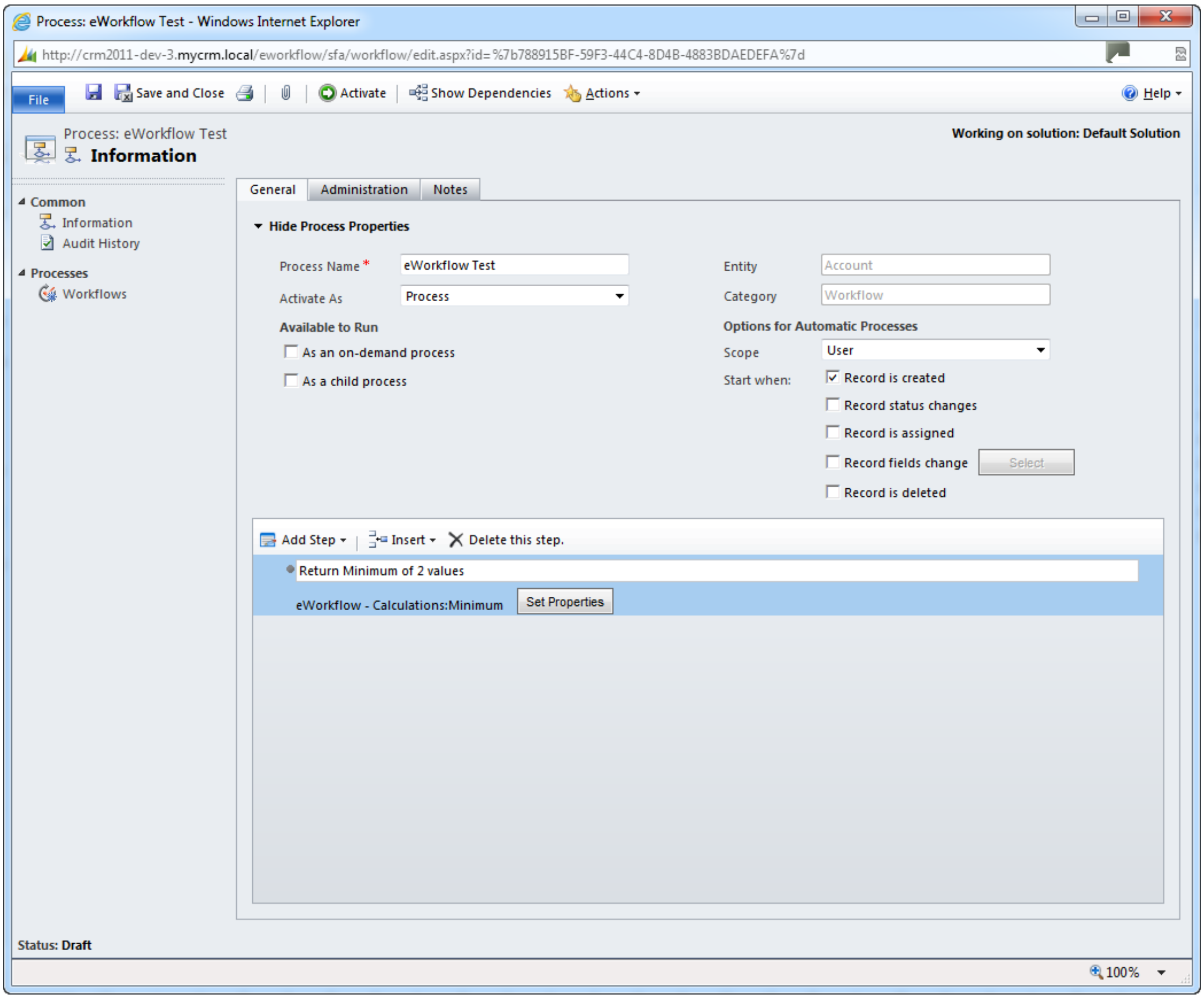
Function Call :- Output
Result (Number)

Function – CalculateMinimum

This workflow function will return the lower value of two numbers so again could be used as part of a wait statement for an account, contact to reach a certain level.

Function Call :- Input
First Number
Second Number

Example 10.1



The validation of the wait check used could then run additional steps or call child workflows.

Function Call :- Output
Result (Number)

eWorkflow – Dates

As part of the eWorkflow library we have included a number of popular date manipulation functions that are supported and easy to use.

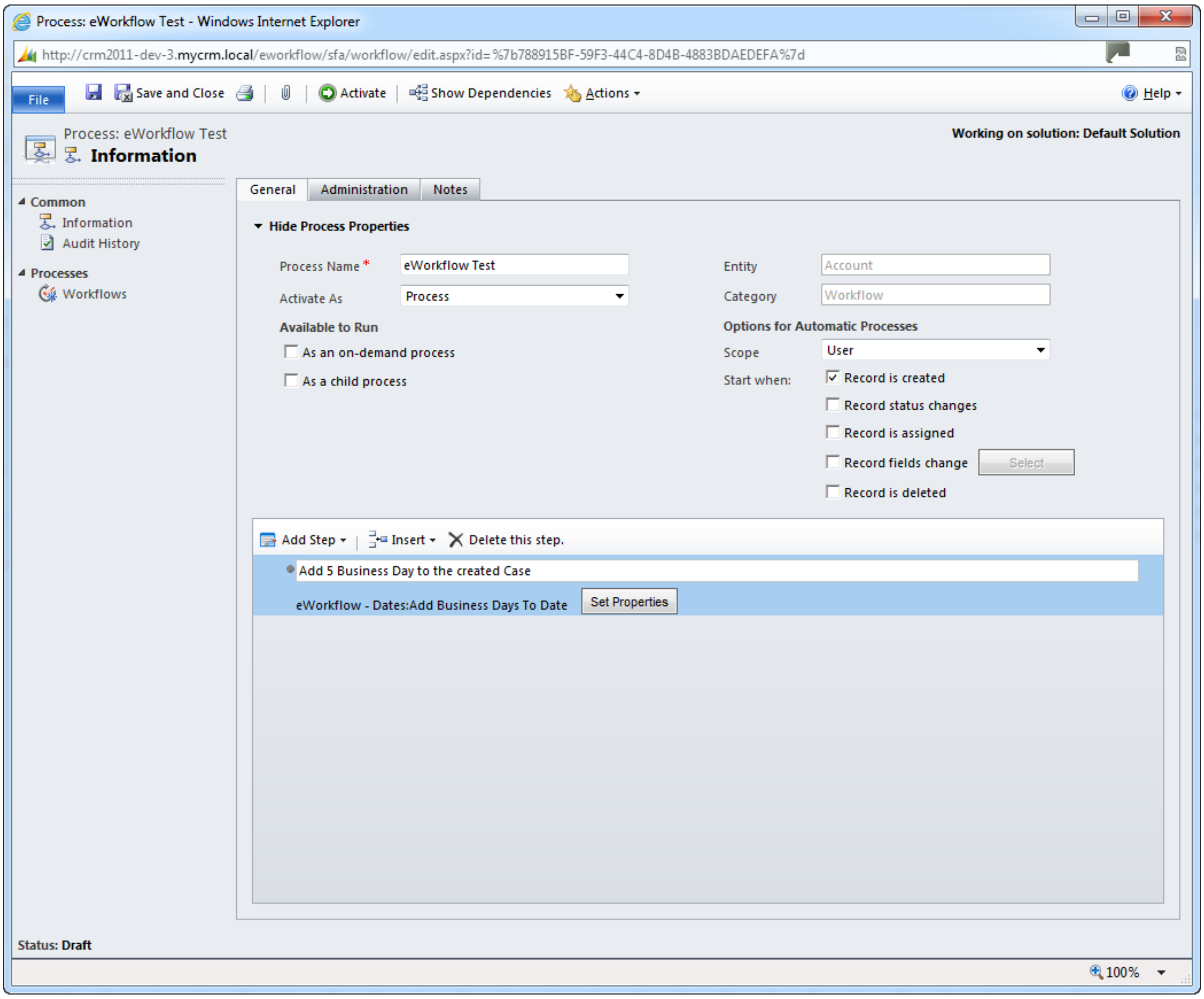
Function – AddBusinessDaysToDate

There are many examples where in a business process you would only want to add the related business days to a required part of a calculation, in most cases these would include follow up activities or the creation of Cases that are under SLA.

As an organisation you may have a standard business process to respond to an enquiry within 3 Business Days, now with eWorkflow you can perform this calculation.

Function Call :- Input
Start Date (Date)
Check to validate if Date is Business Day (True / False)
Days To Add (Whole Number)
Hours To Add (Whole Number)
Minutes To Add (Whole Number)

Example 11.1



Note this function can also be used to add Hours and Minutes, to a date calculation

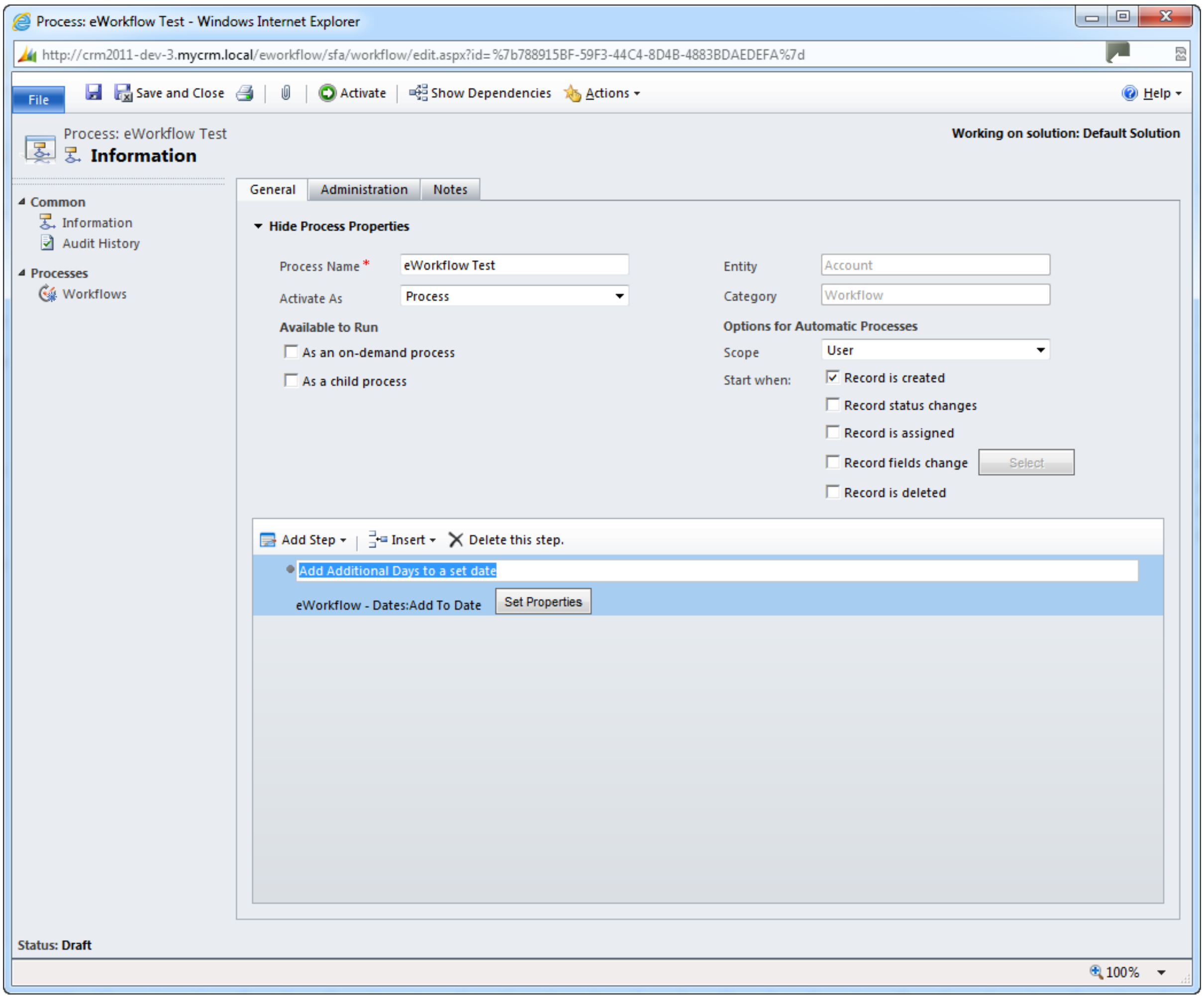
Function Call :- Output
Result (Number)

Function – AddToDate

Add a number of Years, Months, Weeks, Days, Hours and/or Minutes to a given date/time.

Function Call :- Input
Start Date (Date)
Years to Add (Whole Number)
Months to Add (Whole Number)
Weeks to Add (Whole Number)
Days To Add (Whole Number)
Hours To Add (Whole Number)
Minutes To Add (Whole Number)

Example 12.1



Function Call :- Output
Result (Number)

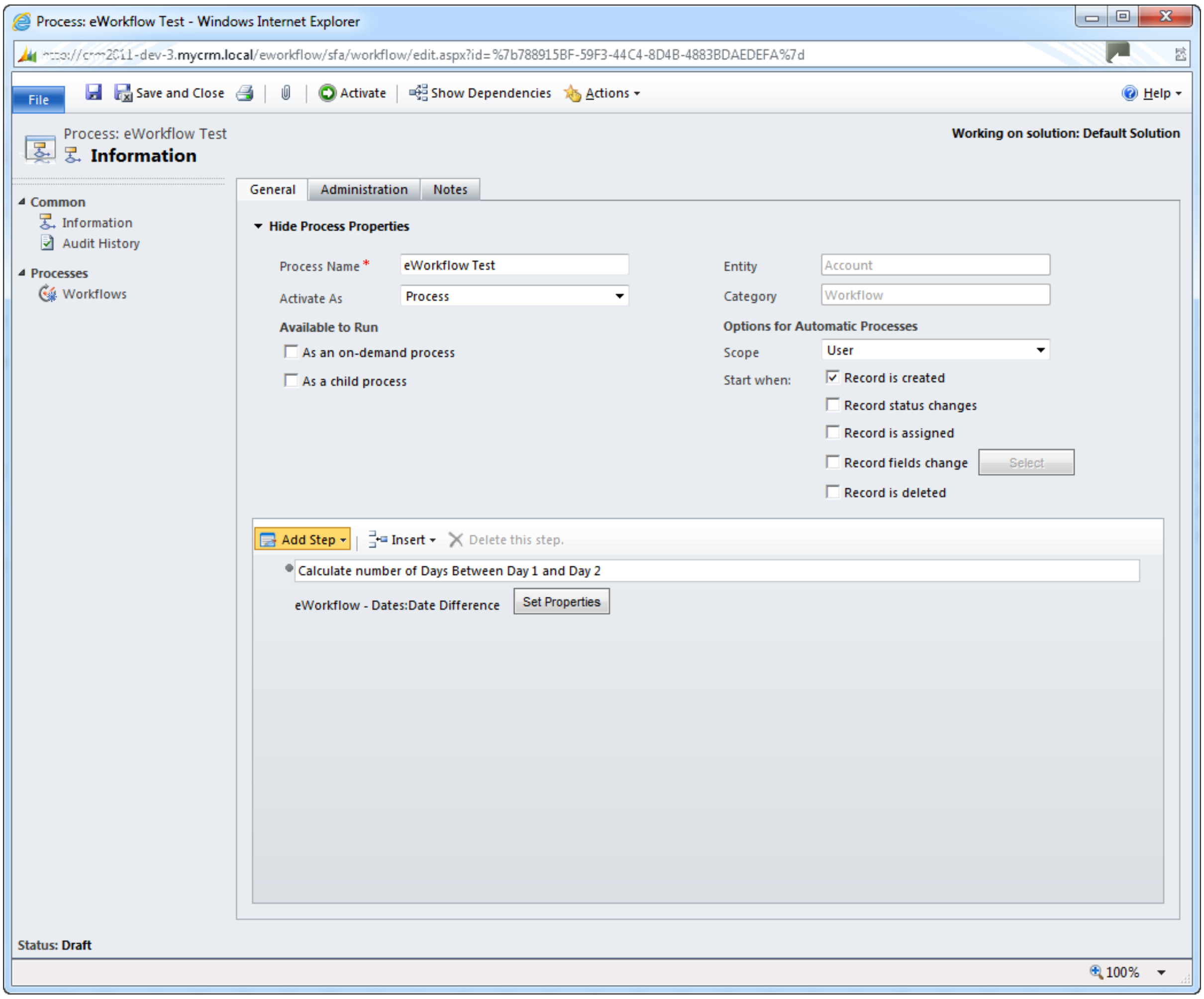
Function – DateDifference

This is a great feature to calculate the number of actual days between two dates given. Dates given as part of an equation or check statement will return an integer number of actual days.

You can also use the **DateDifferenceBusinessDays** which will calculate the difference but only return the business days beween the two dates.

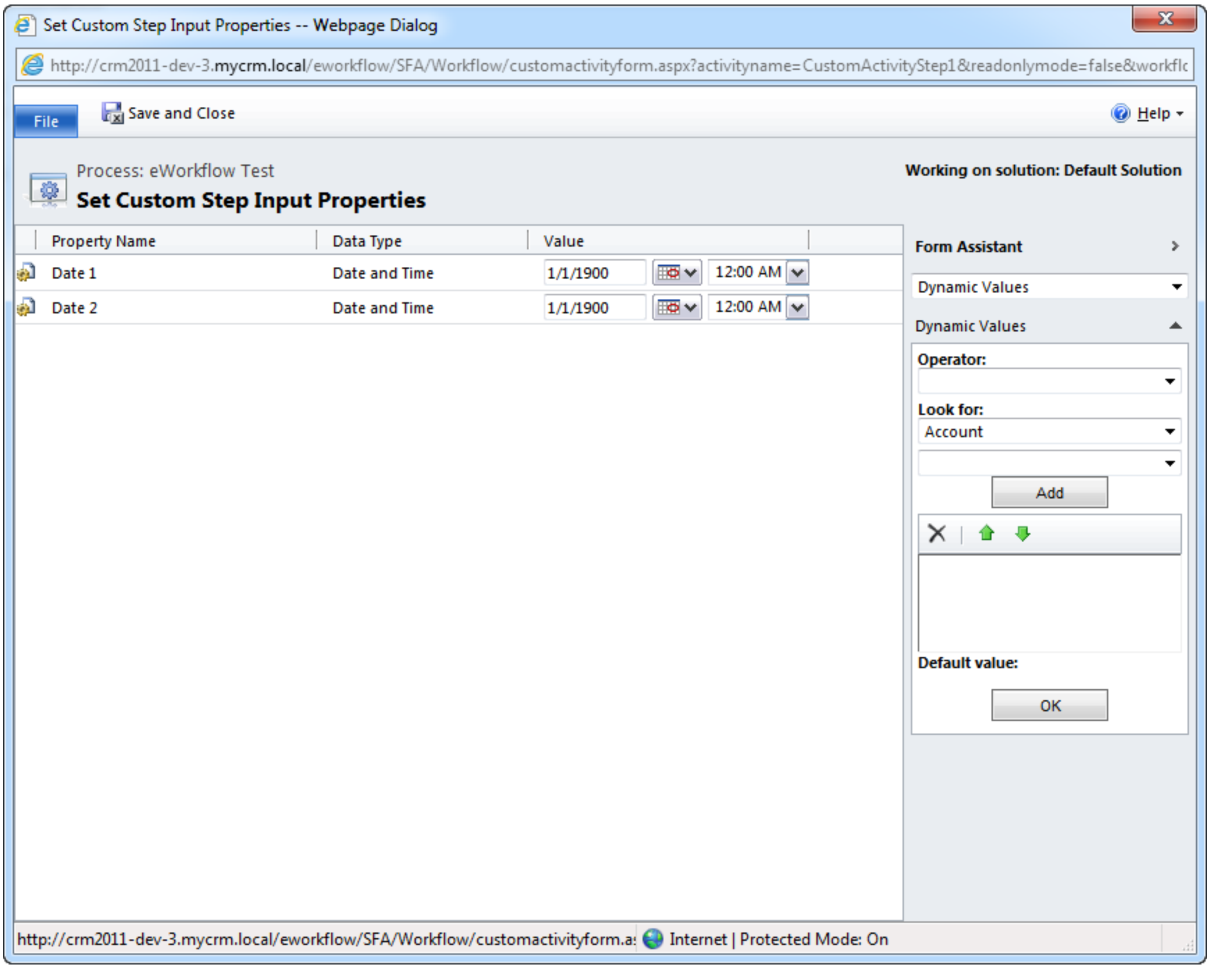
Function Call :- Input
Date 1 (Date)
Date 2 (Date)

Example 13.1



Dates are entered by setting the parameter on the settings screen

Example 13.2



Function Call :- Output
Days (Whole Number)
Hours (Whole Number)
Minutes (Whole Number)
Total Days (Whole Number)
Total Hours (Whole Number)
Total Minutes (Whole Number)

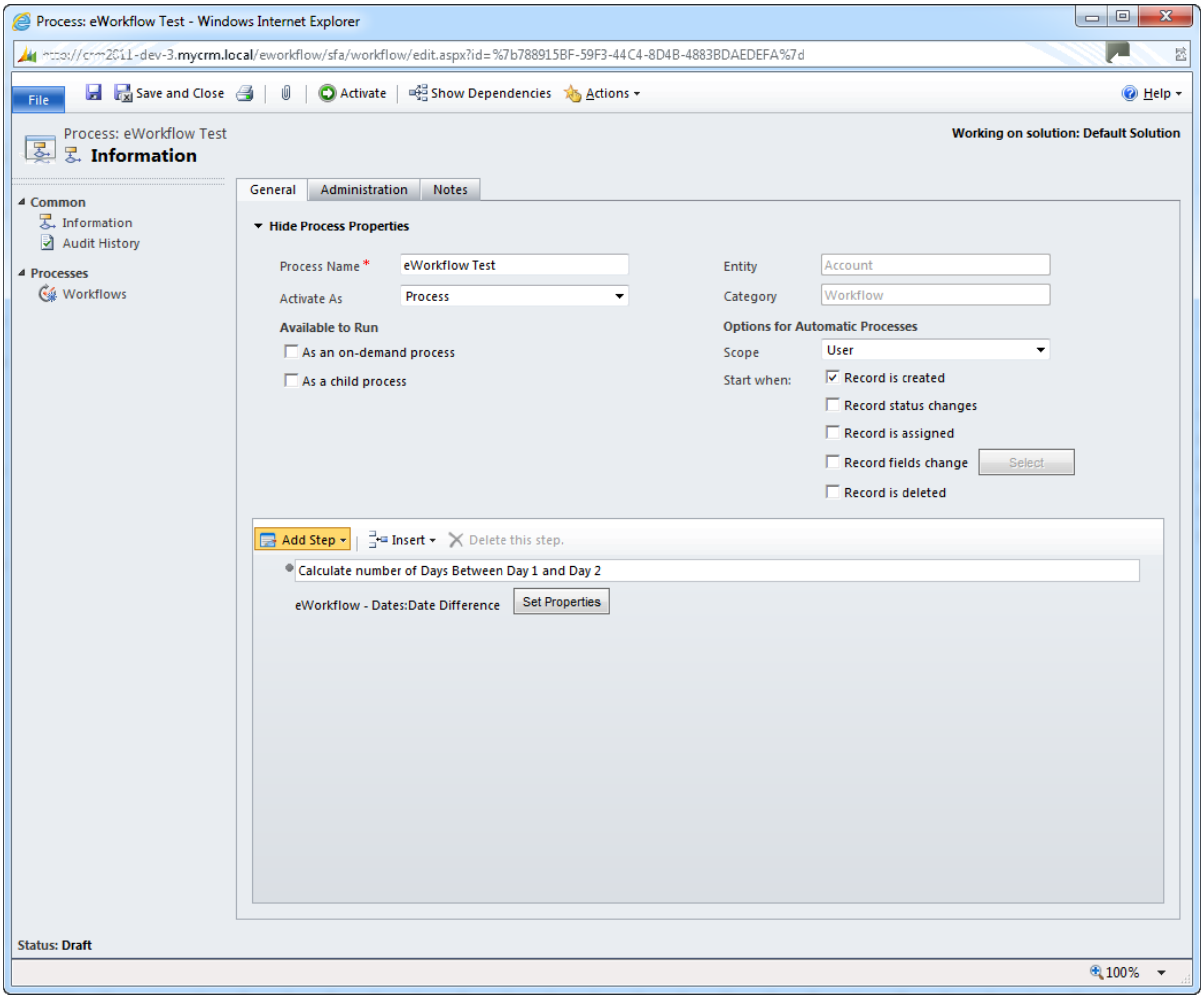
Function – DateDifferenceBusinessDay

Similar to the DateDifference function but this calculation will only return the number of business days left between 2 dates. The function uses the business day rules as defined in CRM.

This is a create function for running in SLA style Workflows that will update a record if a case or incident has not been matched.

Function Call :- Input
Date 1 (Date)
Date 2 (Date)

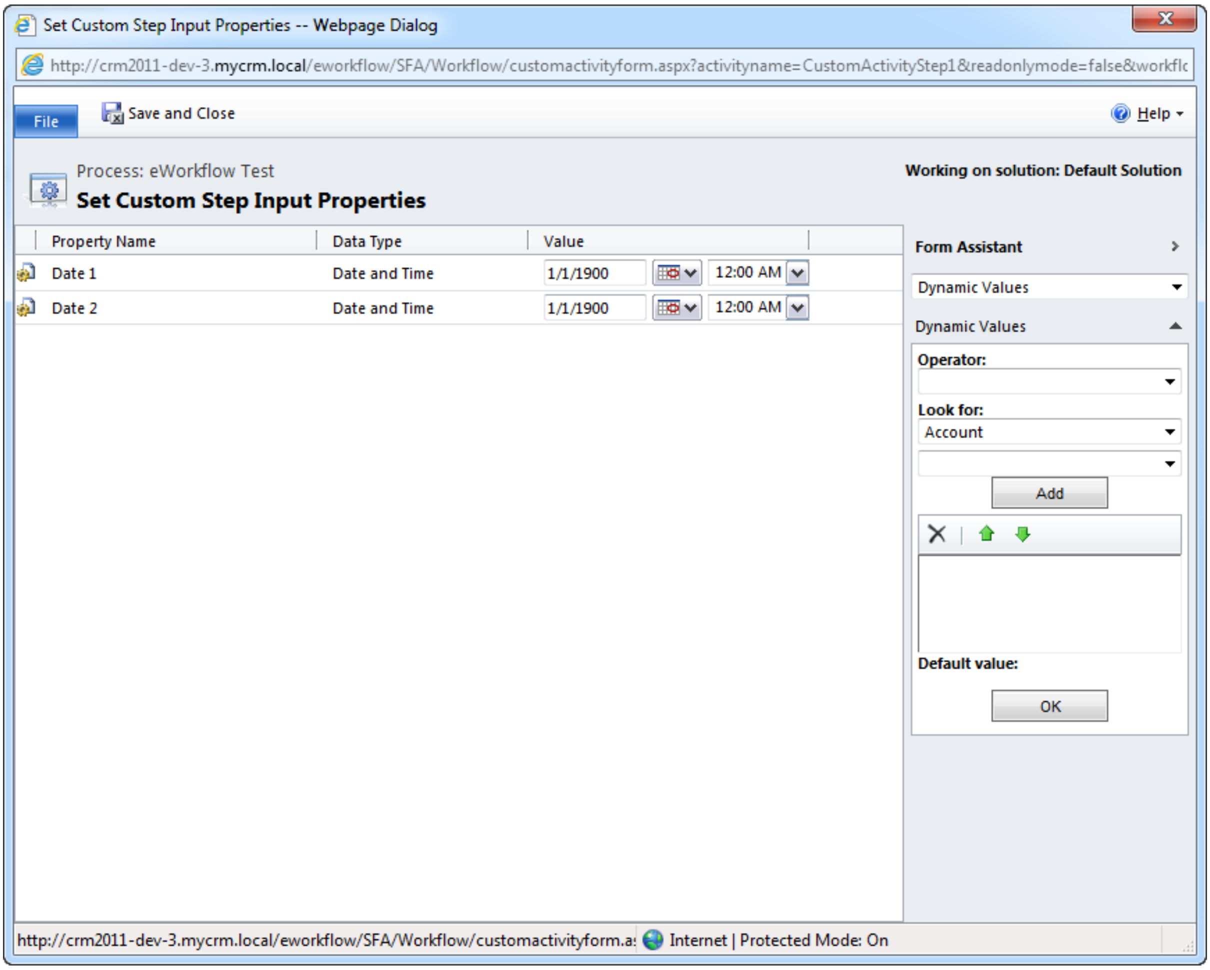
Example 14.1



Function Call :- Output
Days (Whole Number)
Hours (Whole Number)
Minutes (Whole Number)
Total Days (Whole Number)
Total Hours (Whole Number)
Total Minutes (Whole Number)

As before with Date Difference you will need to set the parameters on the setting screen, which can either be fixed dates or data fields (Data Slugs) from CRM.

Example 14.2

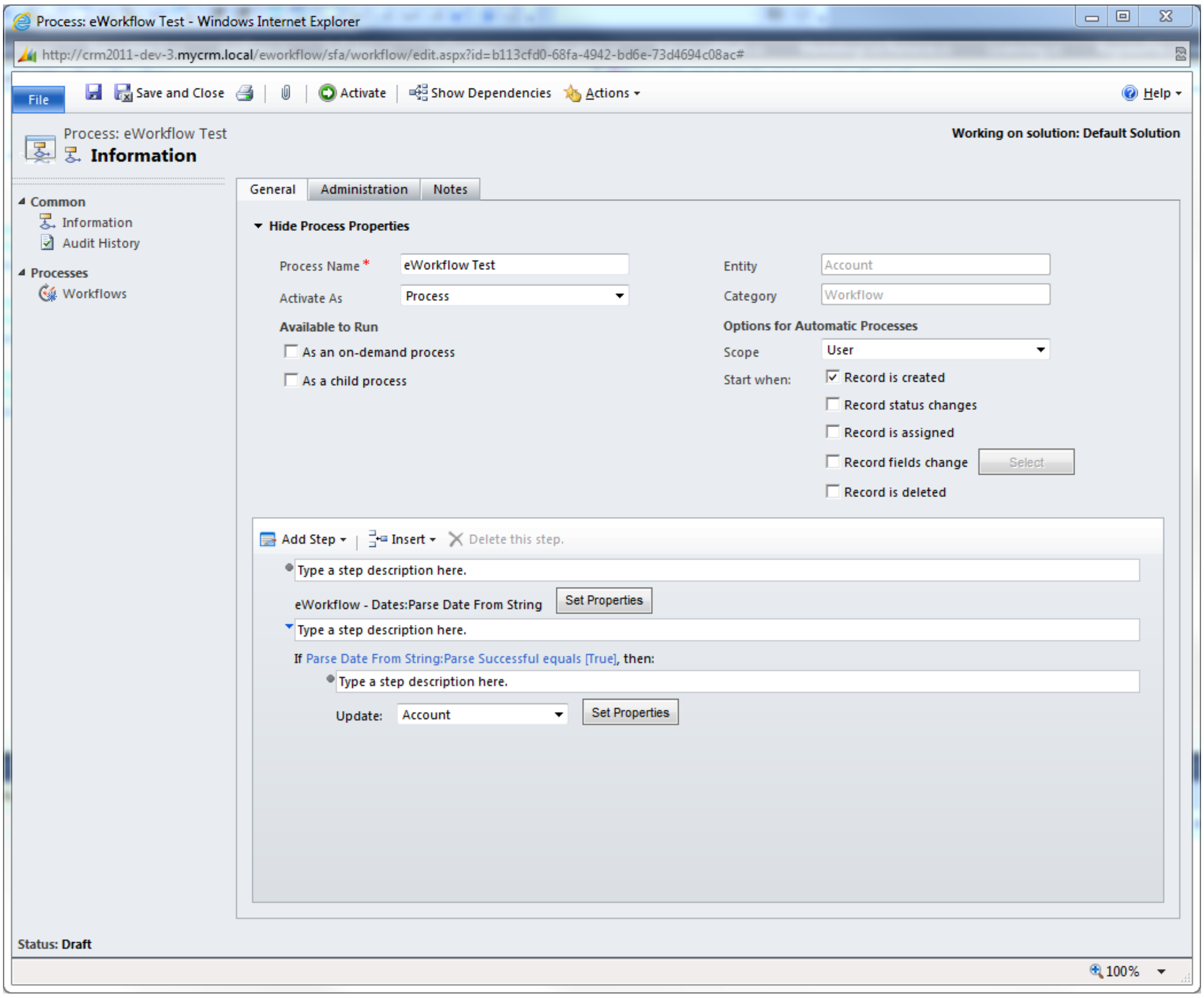


Function – ParseDate

This function takes a string of text and attempts to convert it to a date/time value. Examples would be "15 Jan 2012", "01/04/2010" etc.

Function Call :- Input
Date as Text (a string representing a date)

Example 15.1



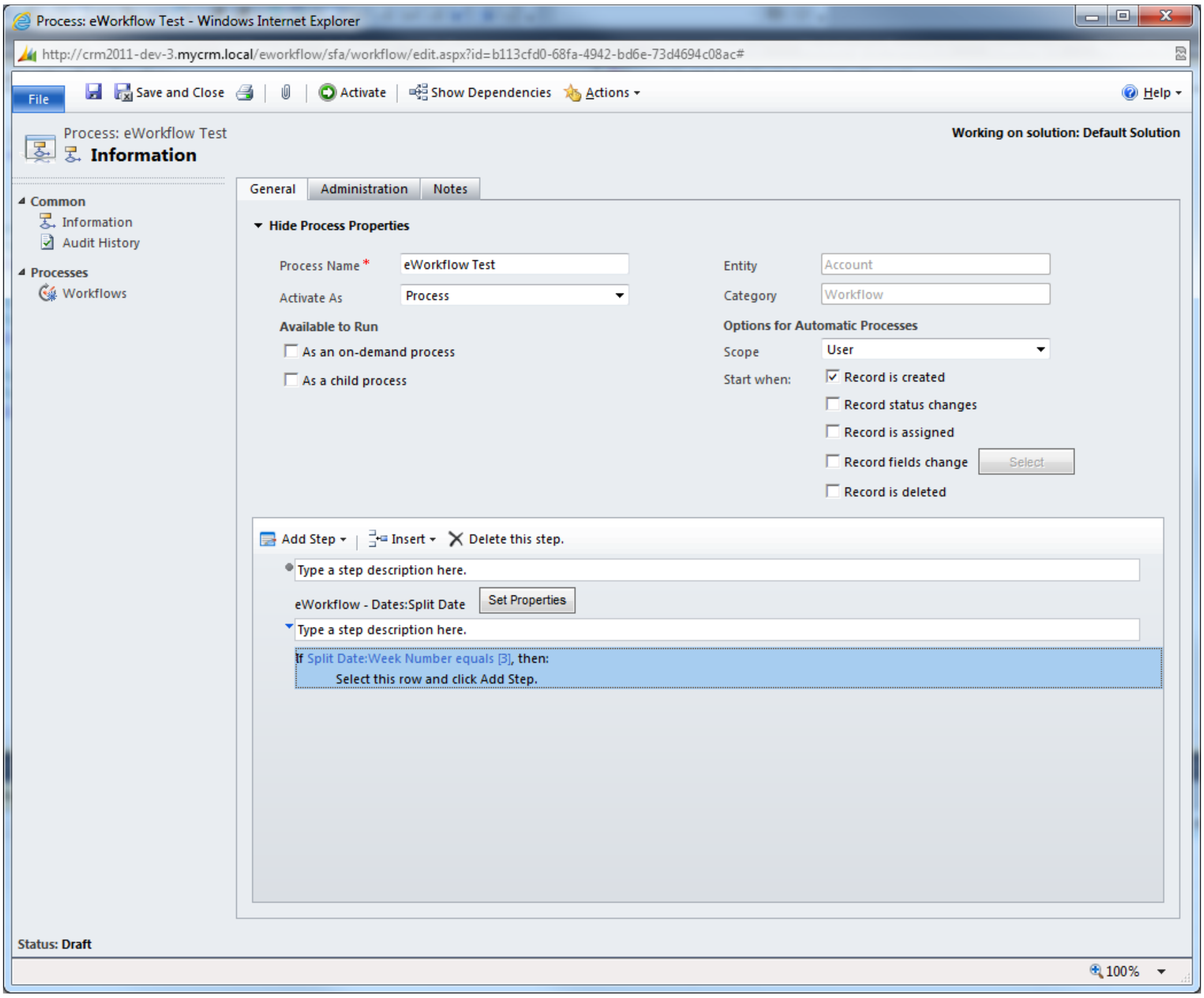
Function Call :- Output
Date (date/time)
Parse Successful (boolean) – should be used to check if the string was parsed successfully before updating a field with the Date value

Function – SplitDate

Takes a date/time as input, and splits it into useful component output parts.

Function Call :- Input
Date 1 (CRM date field or fixed date field from picker)

Example 16.1



A number of fields are output which can then be used for further validation

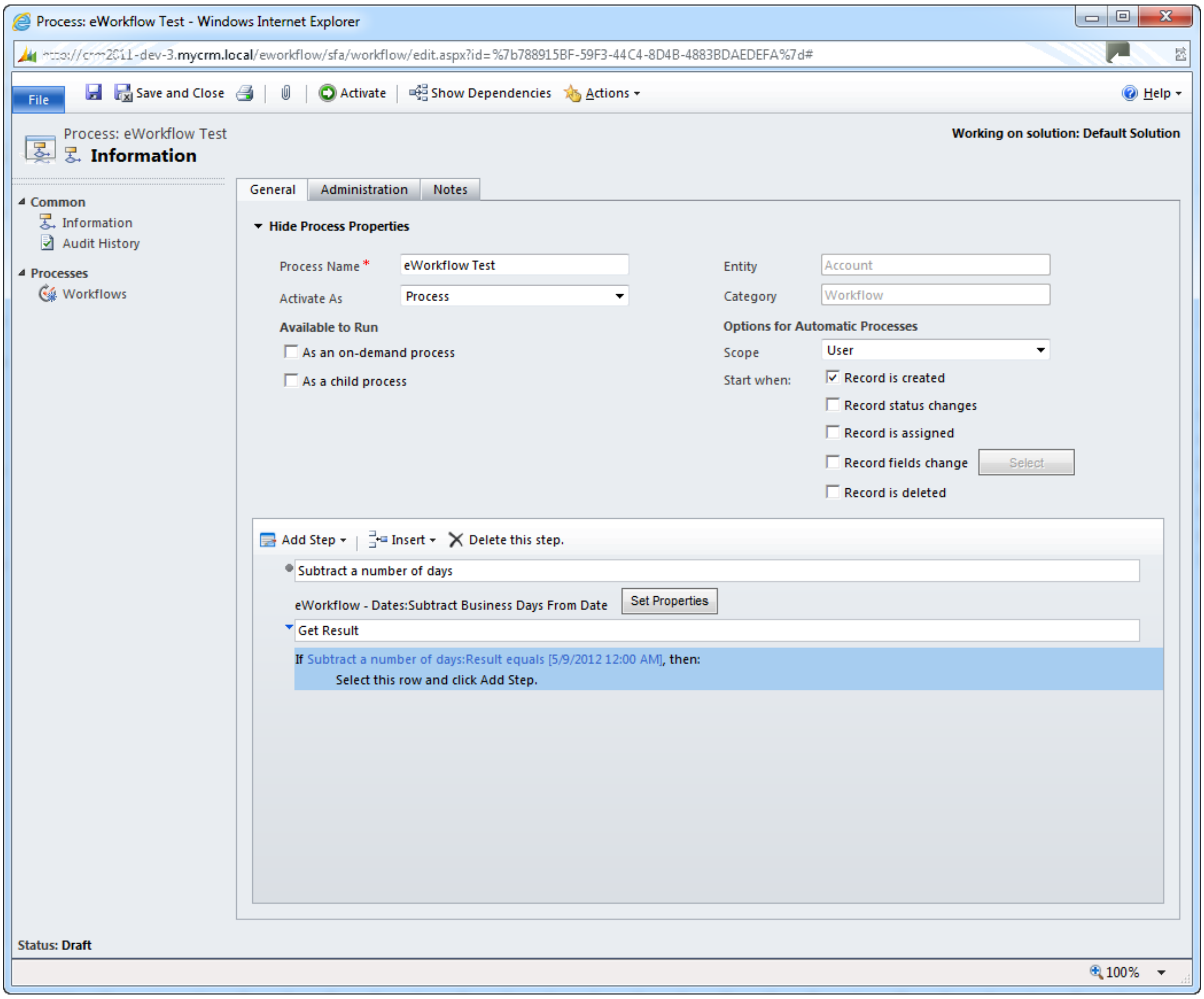
Function Call :- Output
Day of Week (Number)
Day of Week (Text)
Day of Year
Week Number
Year
Month
Day
Hour (12 hour clock)
Hour (24 hour clock)
Minute
AM
PM

Function – SubtractBusinessDaysFromDate

Similar to the above function that enables the adding of a number of Business Days, you can use this function to subtract days from a given date.

Function Call :- Input
Date 1 (CRM date field or fixed date field from picker)
Check to see if Date is Business Day
Days To Subtract
Hours to Subtract
Minutes to Subtract

Example 17.1



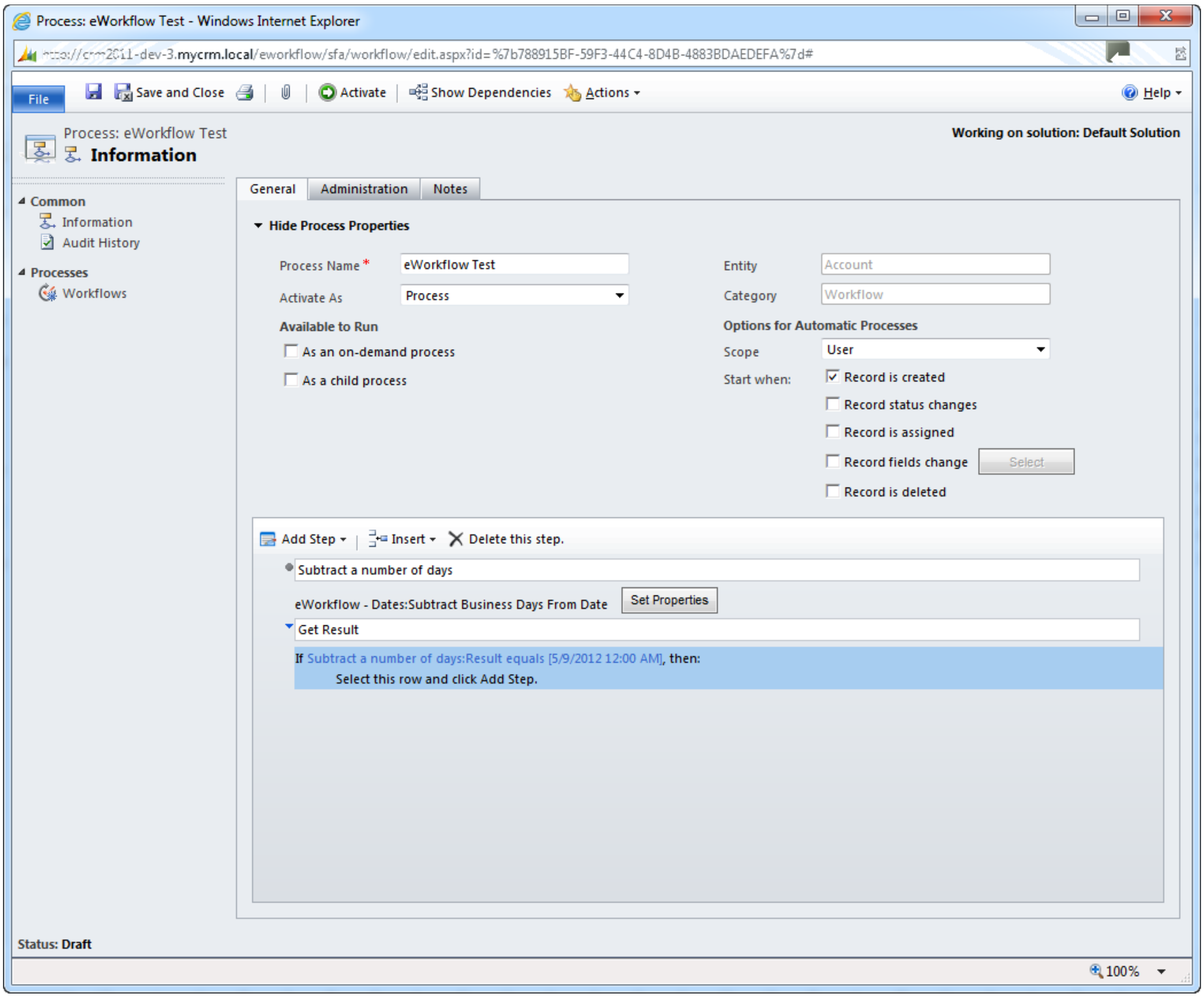
Function Call :- Output
Result (CRM date)

Function – SubtractDaysFromDate

A similar function to subtracting business days but this function includes all week days, additional input parameters also exist.

Function Call :- Input
Date 1 (CRM date field or fixed date field from picker)
Years to Subtract
Months To Subtract
Weeks to Subtract
Days To Subtract
Hours to Subtract
Minutes to Subtract

Example 18.1



Function Call :- Output
Result (CRM date)

eWorkflow – RegEx

Another set of useful tools and function calls to be included in the workflow Library is that for handling Regular Expressions.

These function calls enable an end user to complete a number of predefined functions

Function – RegexMatch

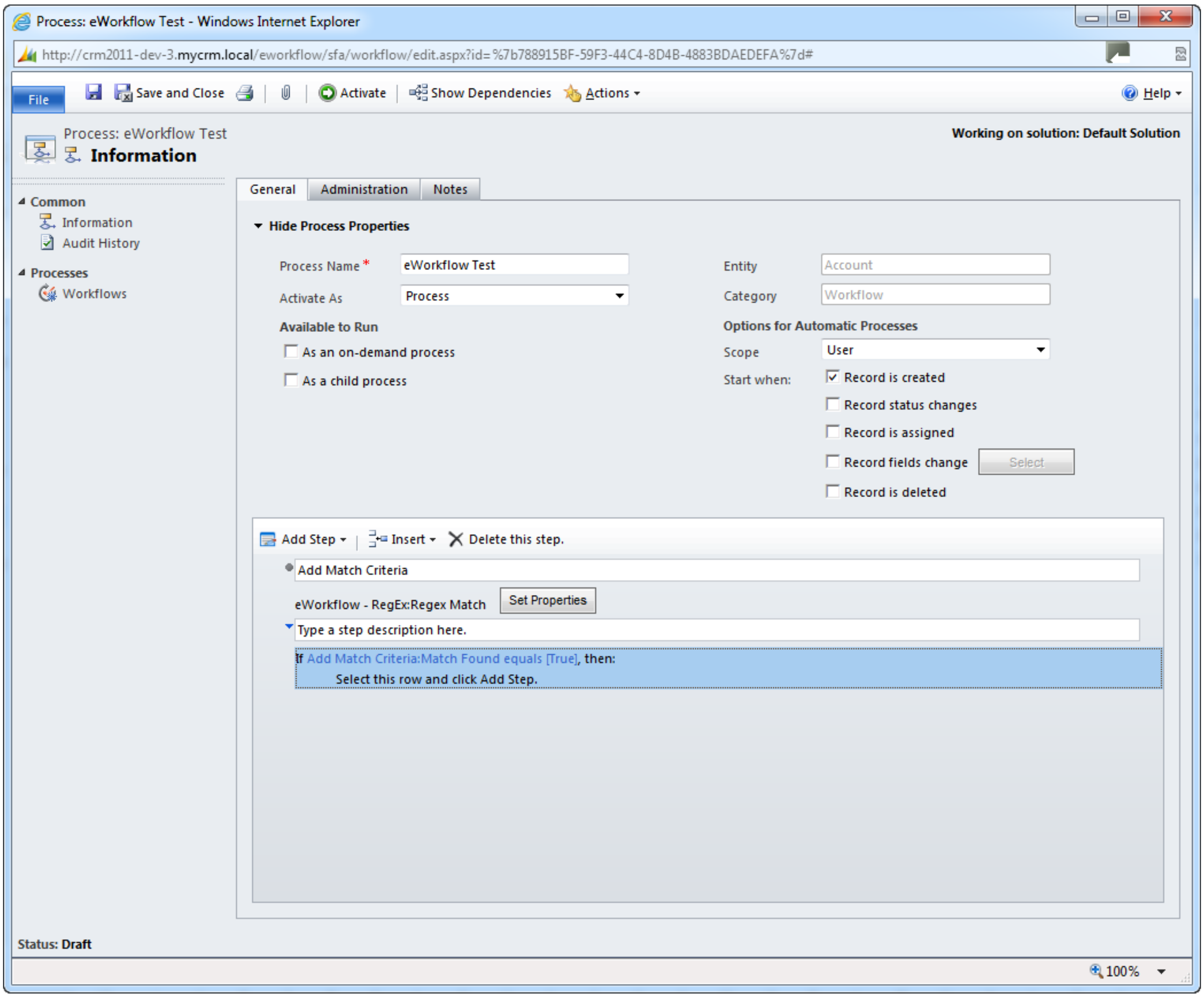
With this function call a workflow can be used to Match text in a body or field of data, regular expressions can be complex but are very powerful when usines in combination with workflow and CRM data Fields

The link below can be used to see example of how to use Regular Expressions against Data

http://www.regular-expressions.info/examples.html

Function Call :- Input
Text (String)
Regular Expression (String)

Example 19.1



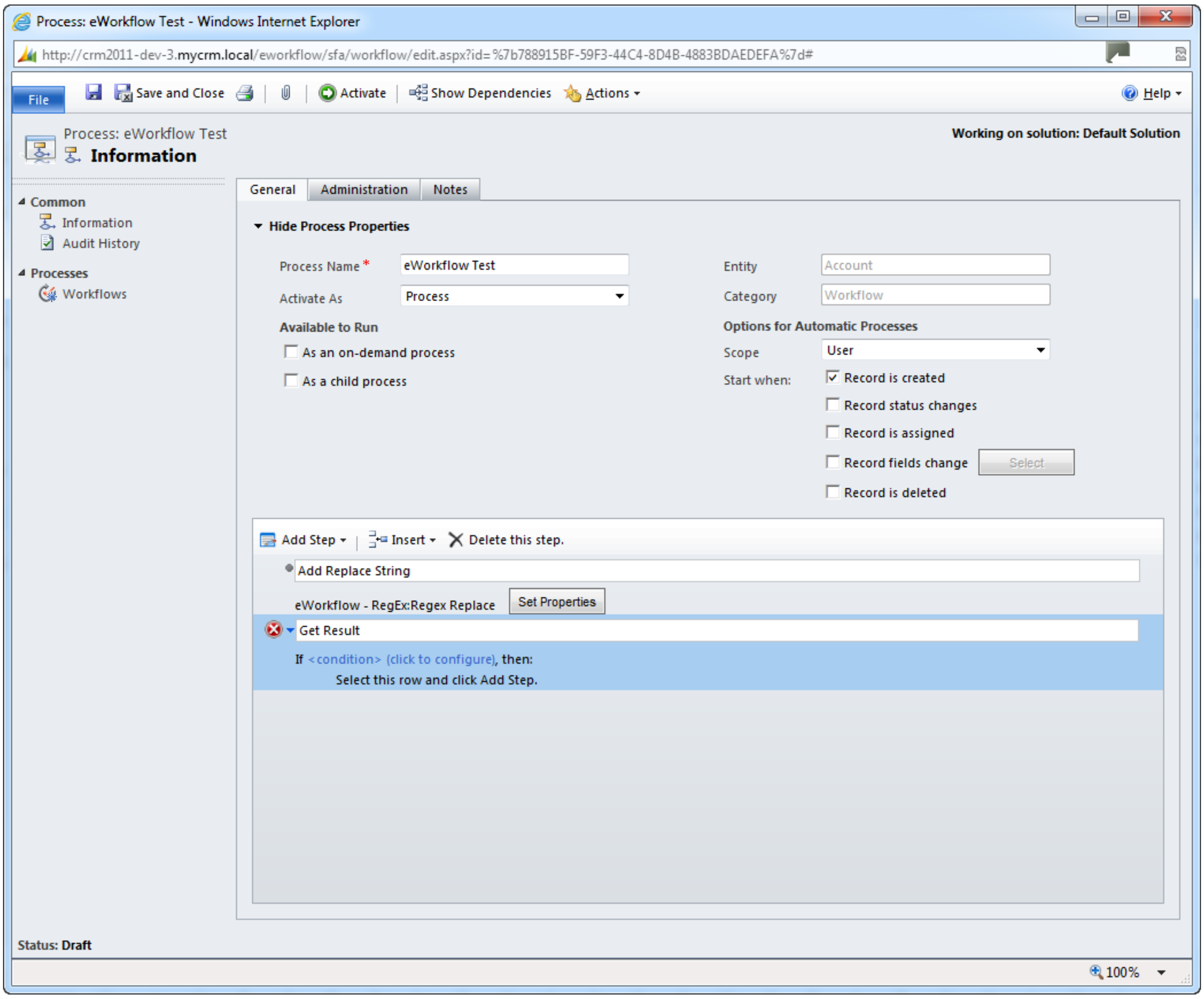
Function Call :- Output
Match (True / False)
Invalid Regular Expression

Function – RegexRelpace

Another great use of the regex functionality is having the ability to replace strings in a block of text or a data field, which can then be used for updating a record.

Function Call :- Input
Text (String)
Replacement (String)
Regular Expression (String)

Example 20.1



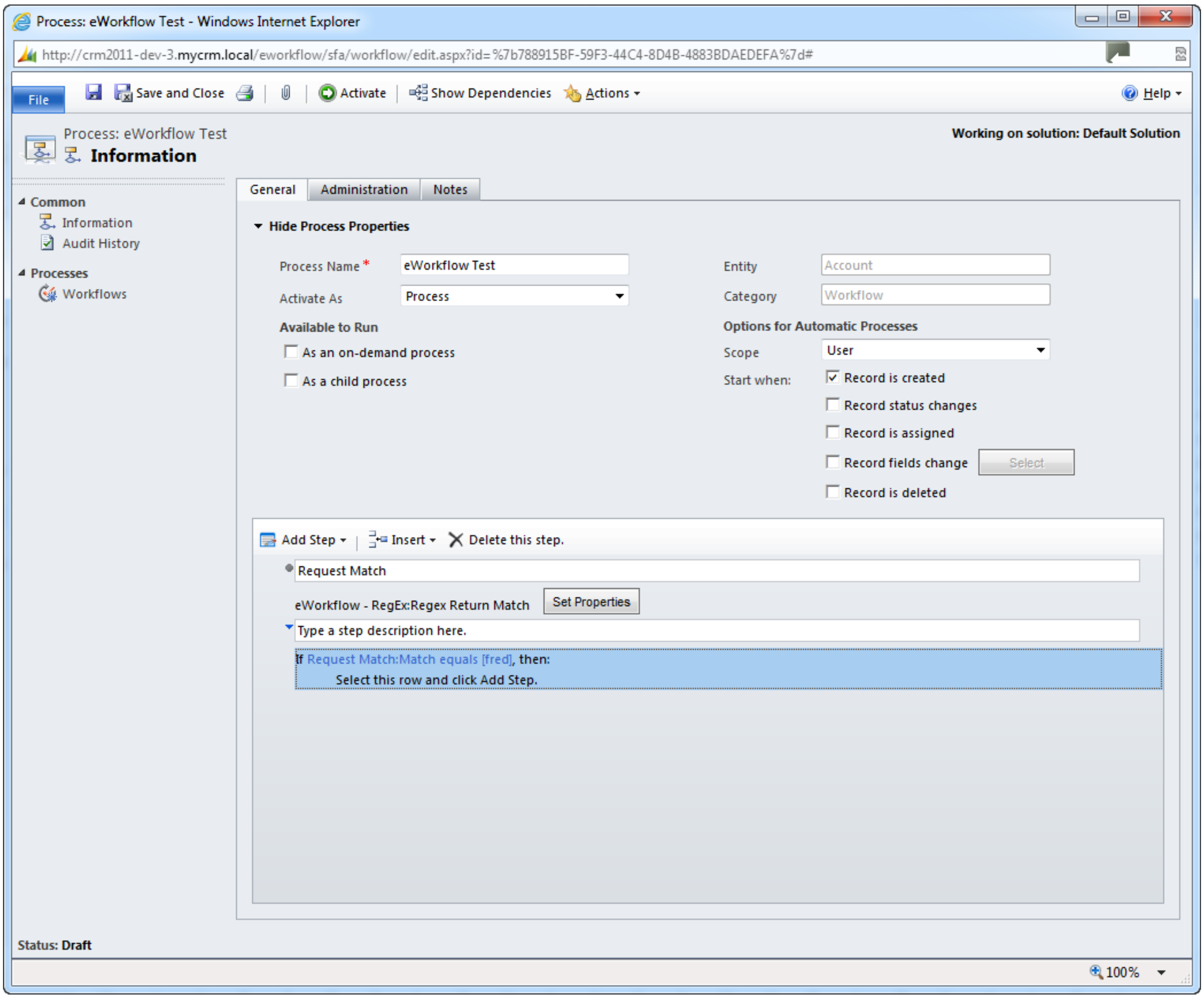
Function Call :- Output
Result (String)
Invalid Regular Expression

Function – RegexReturnMatch

Similar to the match function but this workflow call will return the match it has found

Function Call :- Input
Match Index (Whole Number)
Text (String)
Regular Expression (String)

Example 21.1



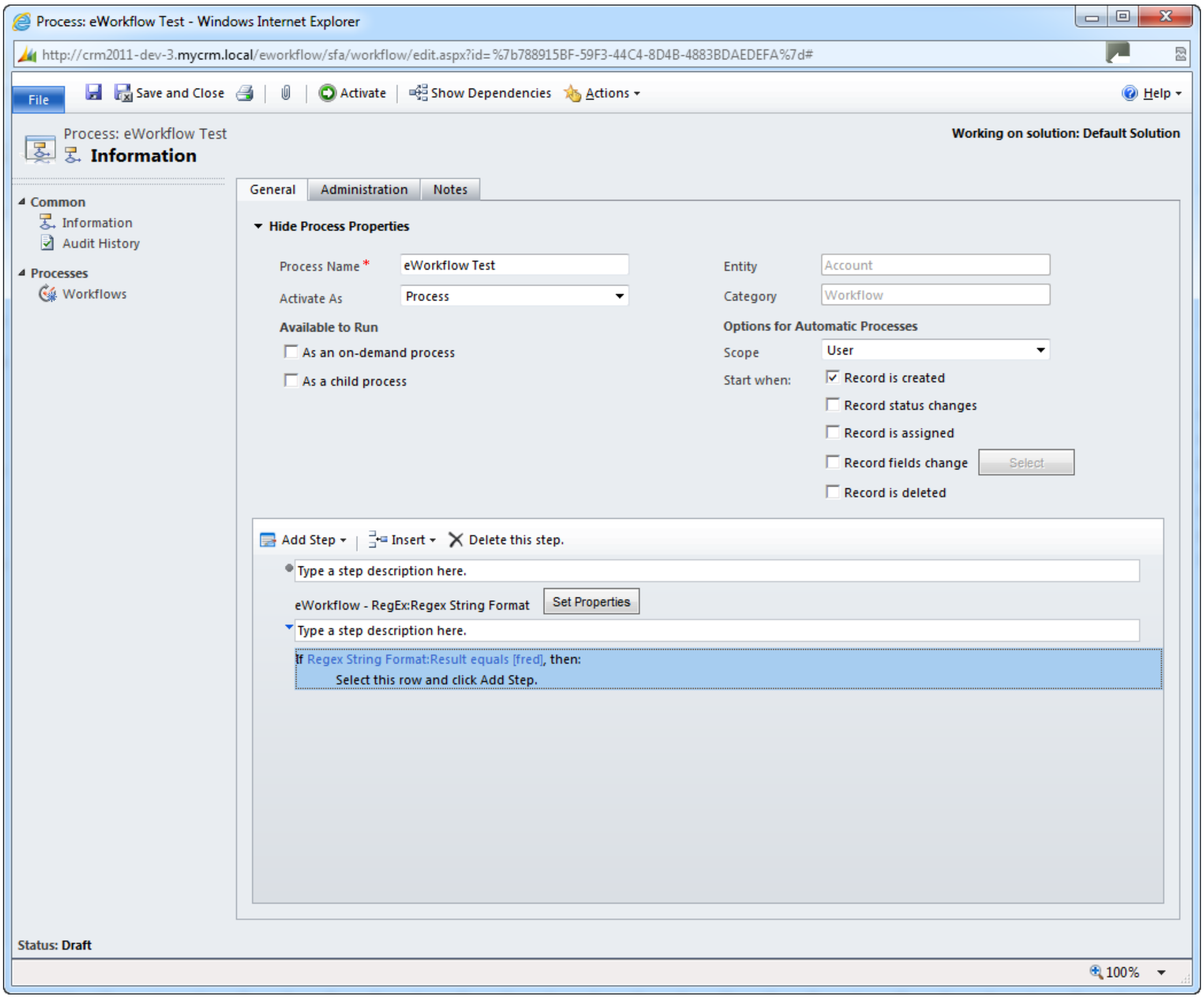
Function Call :- Output
Match Found (True / False)
Match (String)
Invalid Regular Expression

Function – RegexStringFormat

Similar to the match function but this workflow call will return the match that was found

Function Call :- Input
Format (String)
Text (String)
Regular Expression (String)

Example 22.1



Function Call :- Output
Match Found (True / False)
Result (String)
Invalid Regular Expression

eWorkflow – Strings

This function set for Microsoft CRM workflow enables data strings in text and notes fields to be manipulated and updated based on workflow conditions.

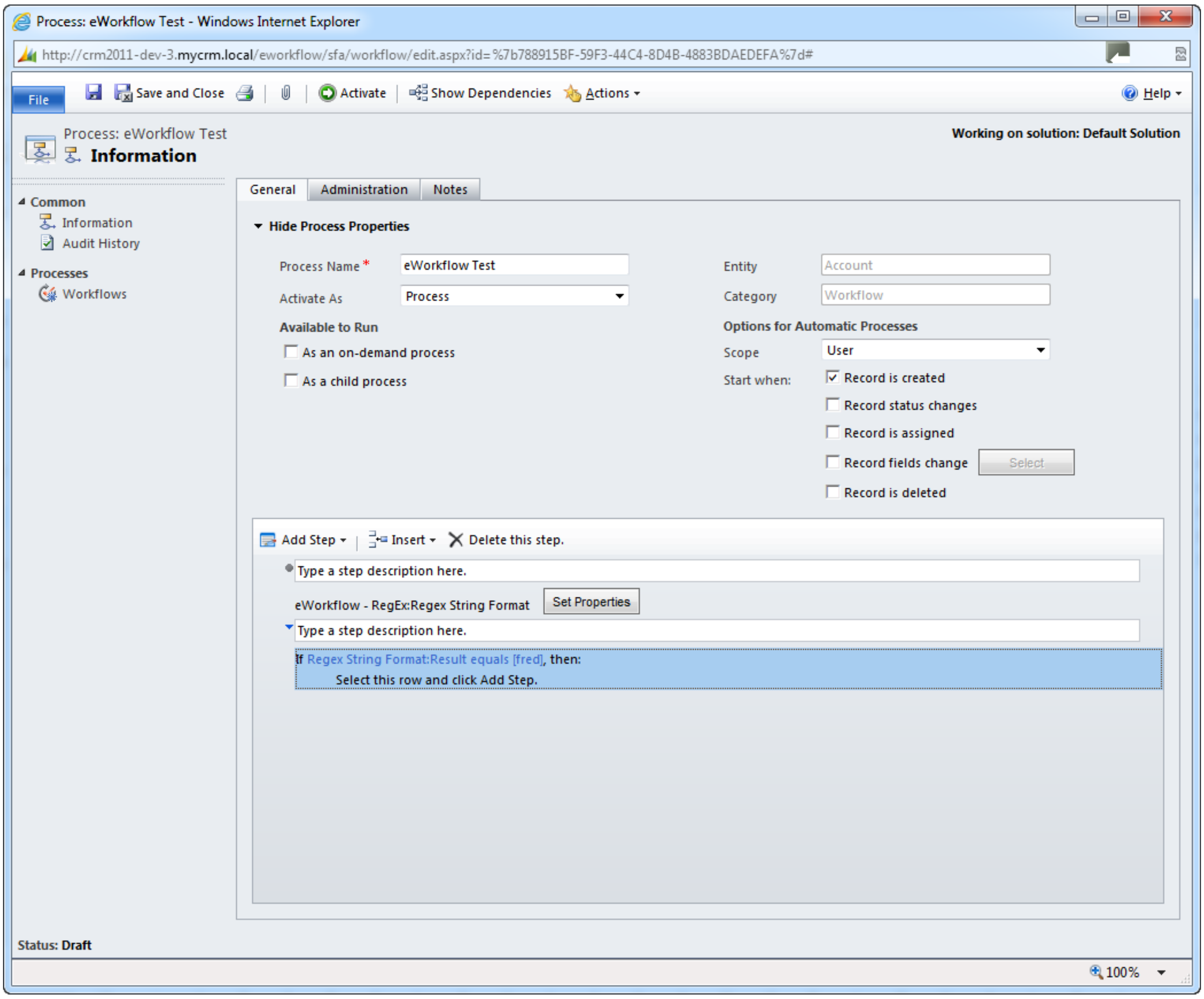
Included are a number of standard string manipulations functions.

Function – CapitaliseString

Takes a string and capitalises either just the first word or all words, depending on the input settings:

Function Call :- Input
Text (String)
Capitalize all Words (True / False)

Example 23.1



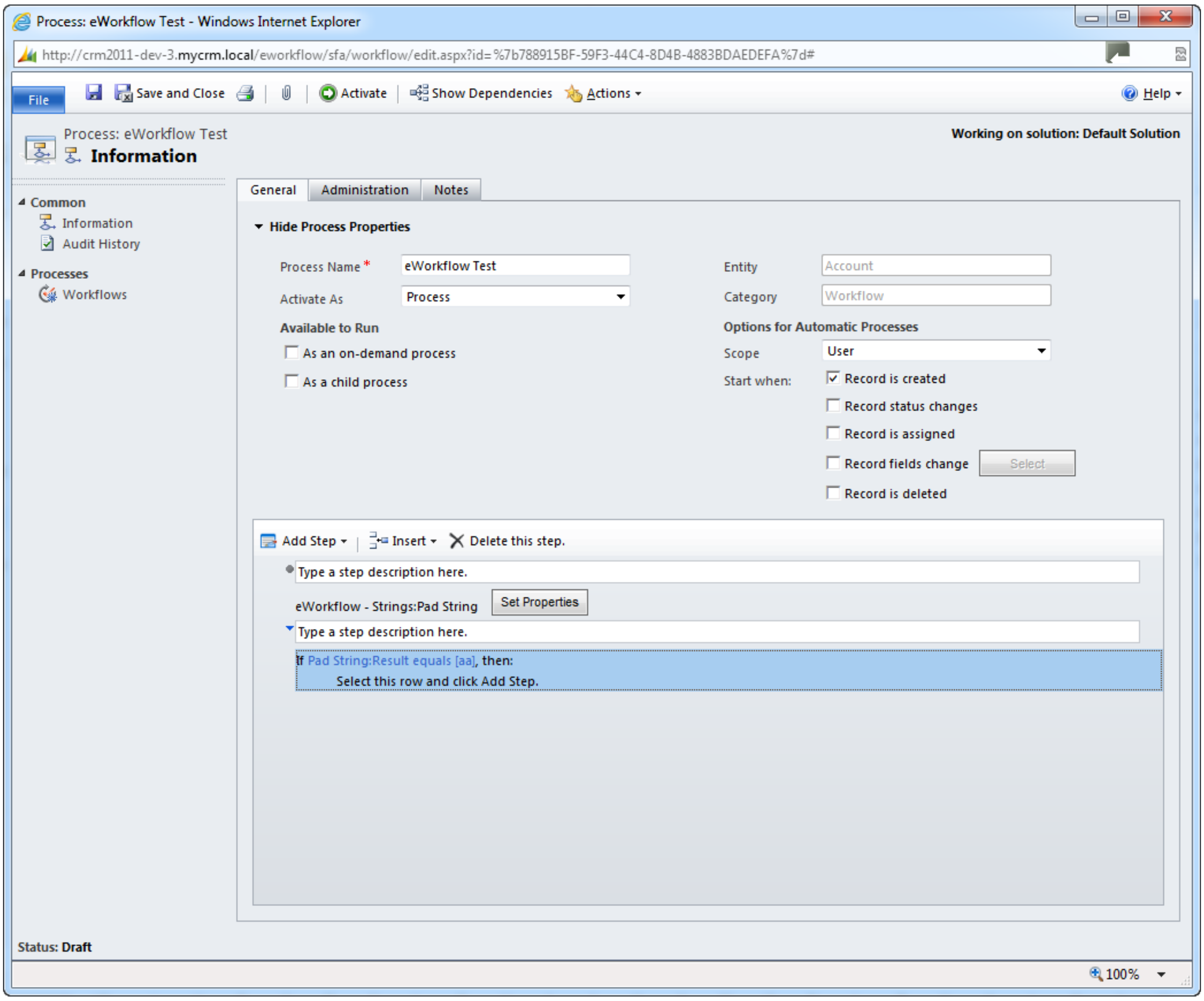
Function Call :- Output
Result (String)

Function – PadString

This function takes a string and pads the left or right side with a specified character, until the full length of the string is equal to the Final Length input parameter.

Function Call :- Input
Text (String)
Pad Character (String)
Pad to Left (True / False)
Final Length (Whole Number)

Example 24.1



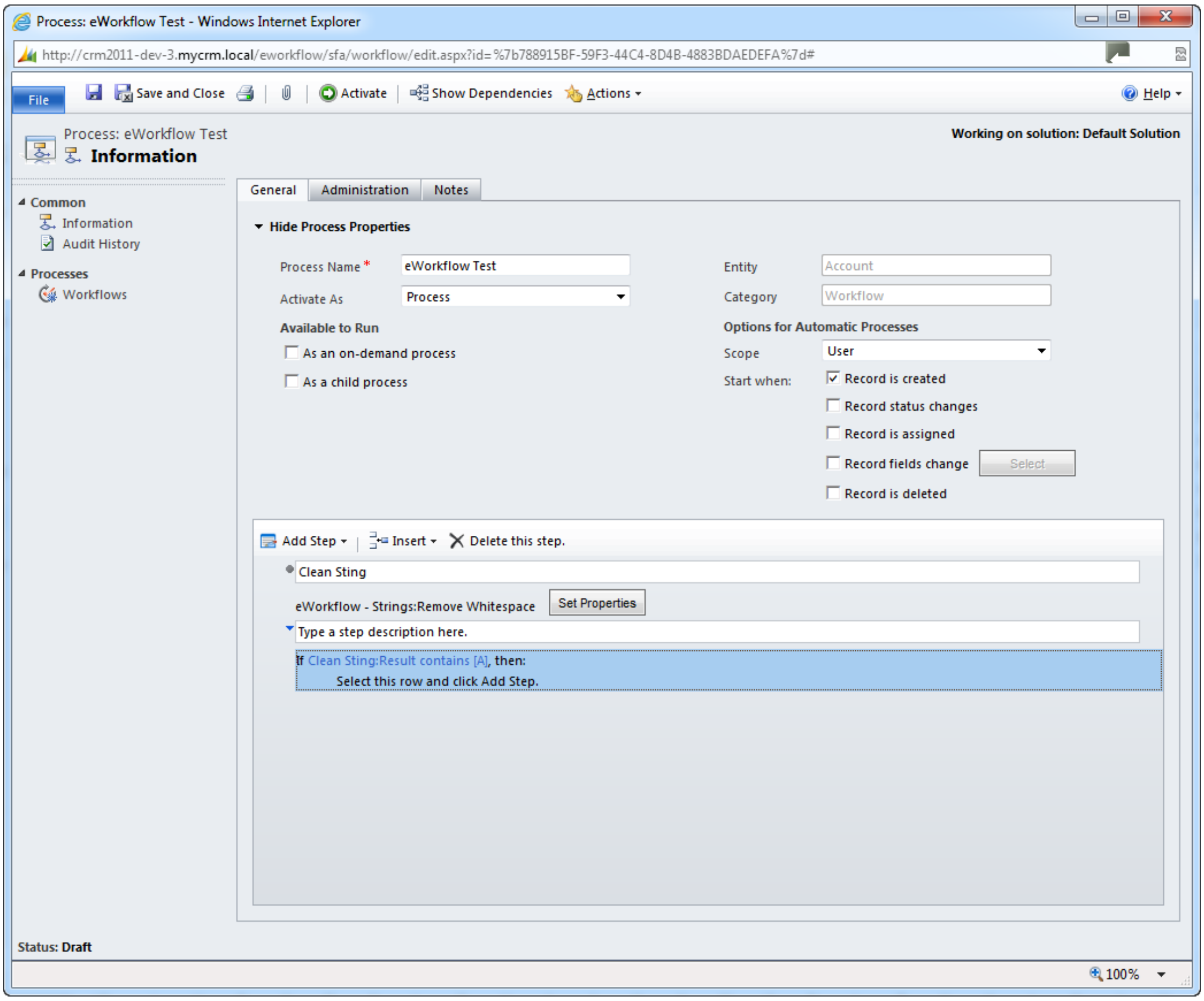
Function Call :- Output
Result (String)

Function – RemoveWhitespace

Removes whitespace from the start or end of a specified input string.

Function Call :- Input
Text (String)

Example 25.1



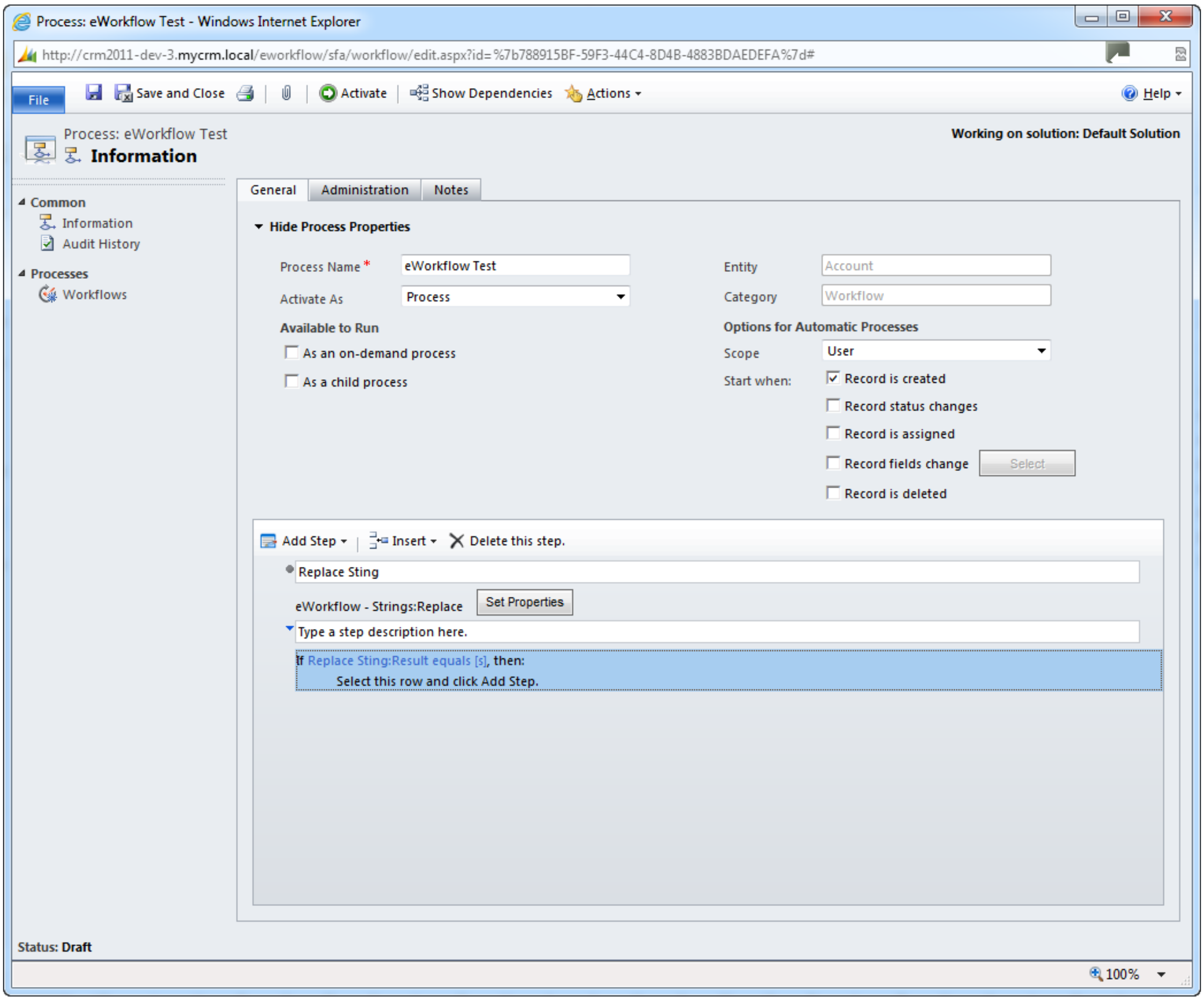
Function Call :- Output
Result (String)

Function – ReplaceString

This function gives the ability to an end user or workflow designer to change string values and replace strings within a body of text in the CRM database

Function Call :- Input
Text (String)
Old Value (String)
New Value (String)
Case Sensitive (True / False)

Example 26.1



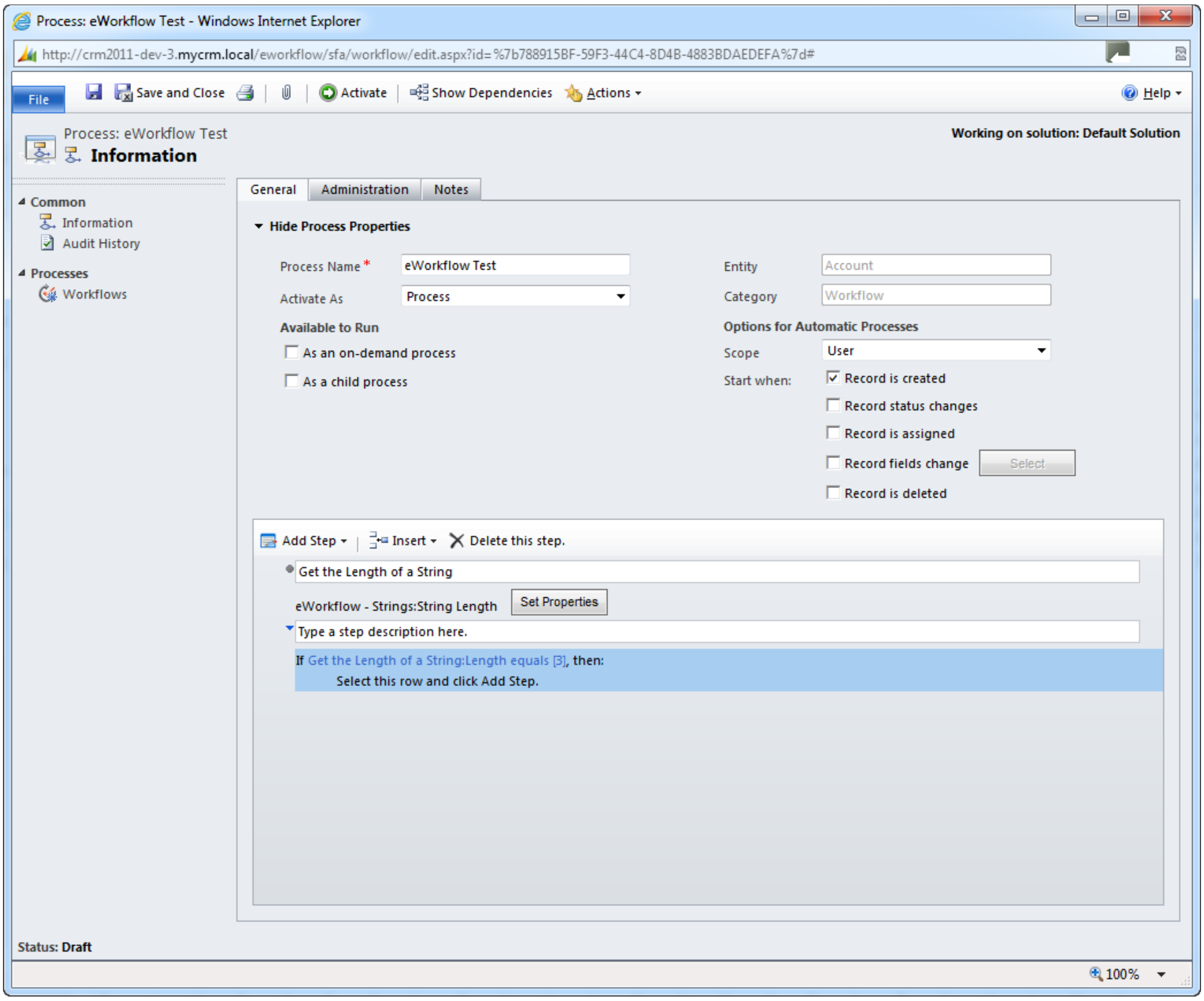
Function Call :- Output
Result (String)

Function –StringLength

Now in the workflow string lengths can be tested to make sure a string is of the right length.

Function Call :- Input
Text (String)

Example 27.1



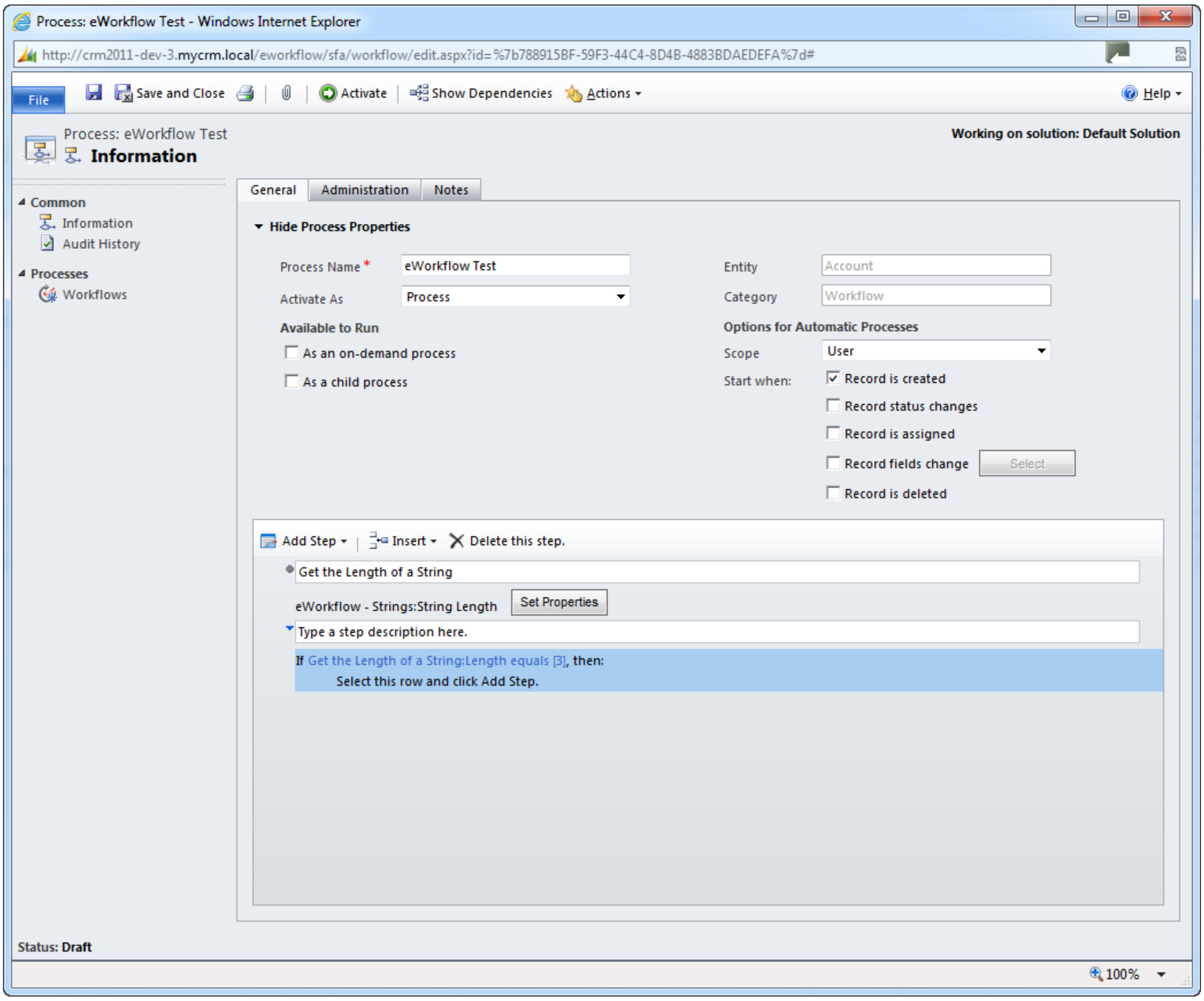
Function Call :- Output
Length (Whole Number)

Function –SubString

This substring check will identify if a block of text of certain keywords exist in a text field, which is great for processing requests based on formatted incoming emails.

Function Call :- Input
Text (String)
From Left To Right (True / False)
Start Index Point
Length

Example 28.1



Function Call :- Output
Result (String)

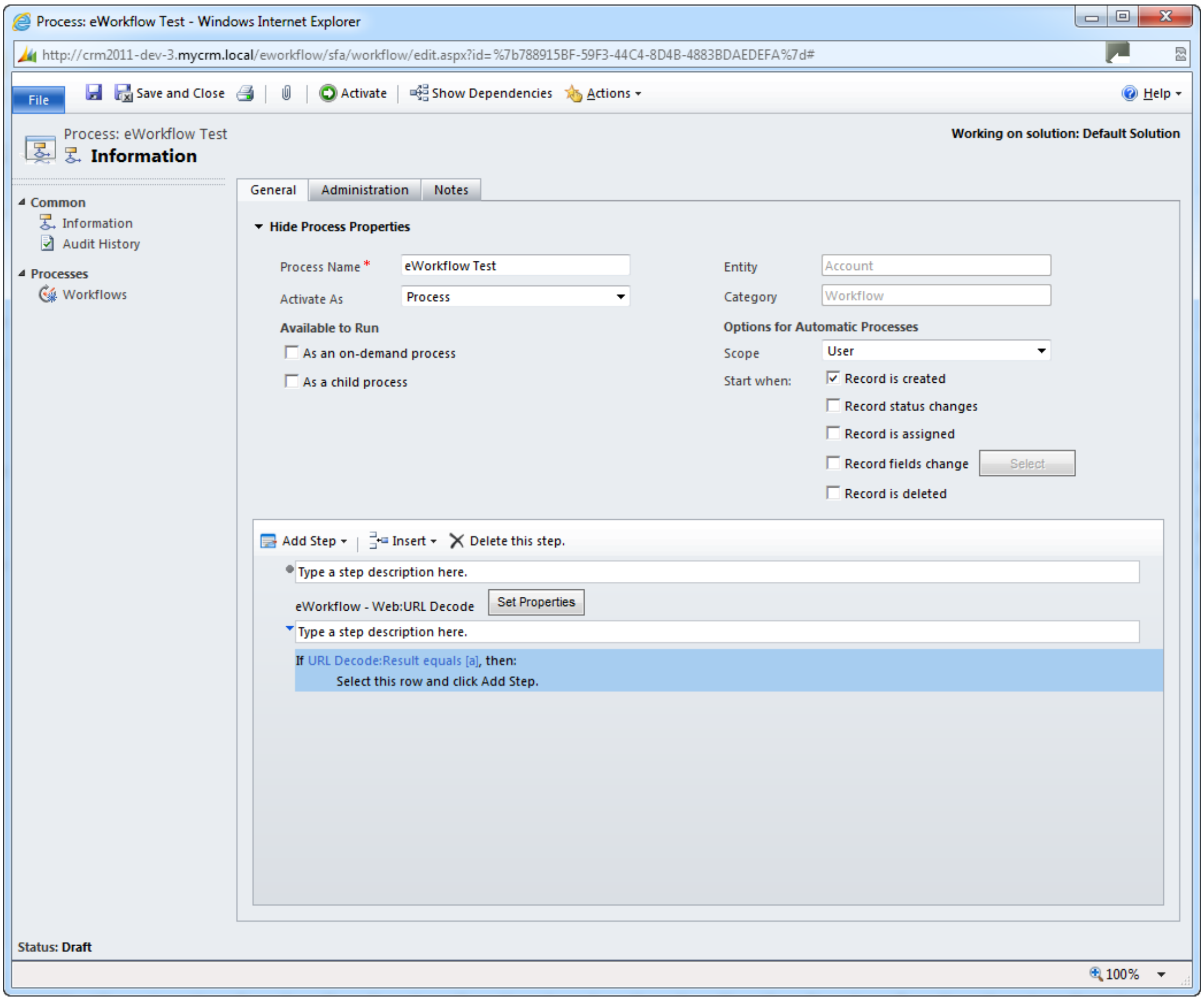
eWorkflow – Web

Two additional functions for URL encoding have been included as part of the eWorkflow Library. For more information on URL encoding, please see: http://en.wikipedia.org/wiki/Percent-encoding

Function –WebURLDecode

Function Call :- Input
Text (String)

Example 29.1

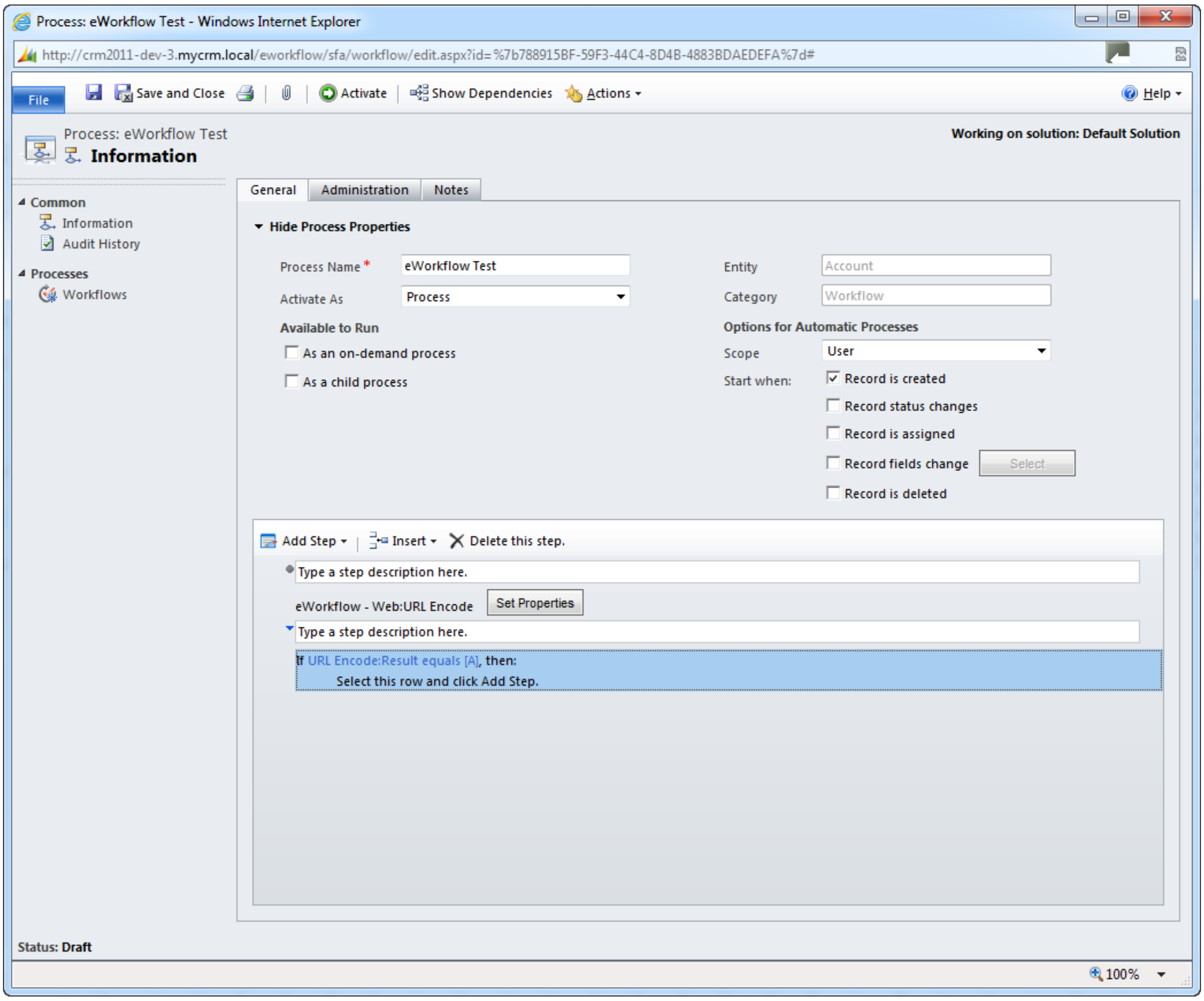


Function Call :- Output
Result (String)

Function –WebURLEncode

Function Call :- Input
Text (String)

Example 30.1



Function Call :- Output
Result (String)

eWorkflow – Sales

We have created a number of useful and extensible functions that may be useful within Sales and Business workflow, these functions enable counts and totals to be created to help gather analytical data

Function – AddInvoiceLineItem

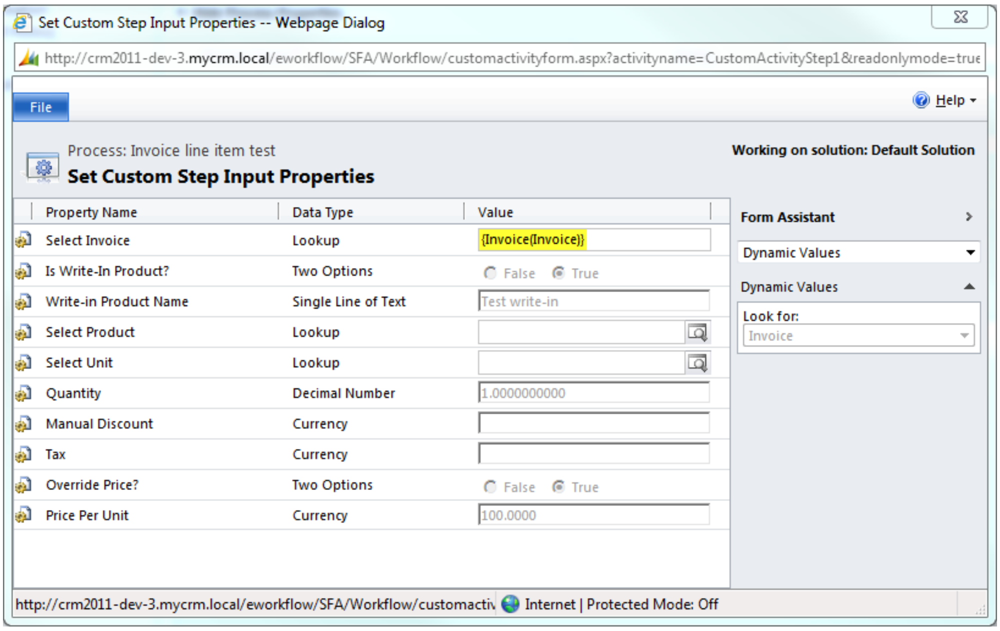
This function allows creation of Line Items on Invoices. CRM2011 lacks the ability to add a Line Item out of the box, but eWorkflow enables this.

The input parameters for the function mirror the standard CRM2011 Line Item form, and must be treated in the same way. The line item may be a write-in with a supplied name, and the price may be overridden if an actual product is supplied. In the case of a write-in product or an overridden price, the item price should be entered into the Price Per Unit field.

The Unit field is required when an actual product is supplied.

Function Call :- Input
Invoice (Object)
Is Write-In Product? (yes/no)
Write-in Product Name (string)
Product (Object)
Unit (Object)
Quantity (decimal)
Manual Discount (currency)
Tax (currency)
Override Price? (yes/no)
Price Per Unit (currency)

Example 31.1



Function – AddOpportunityLineItem

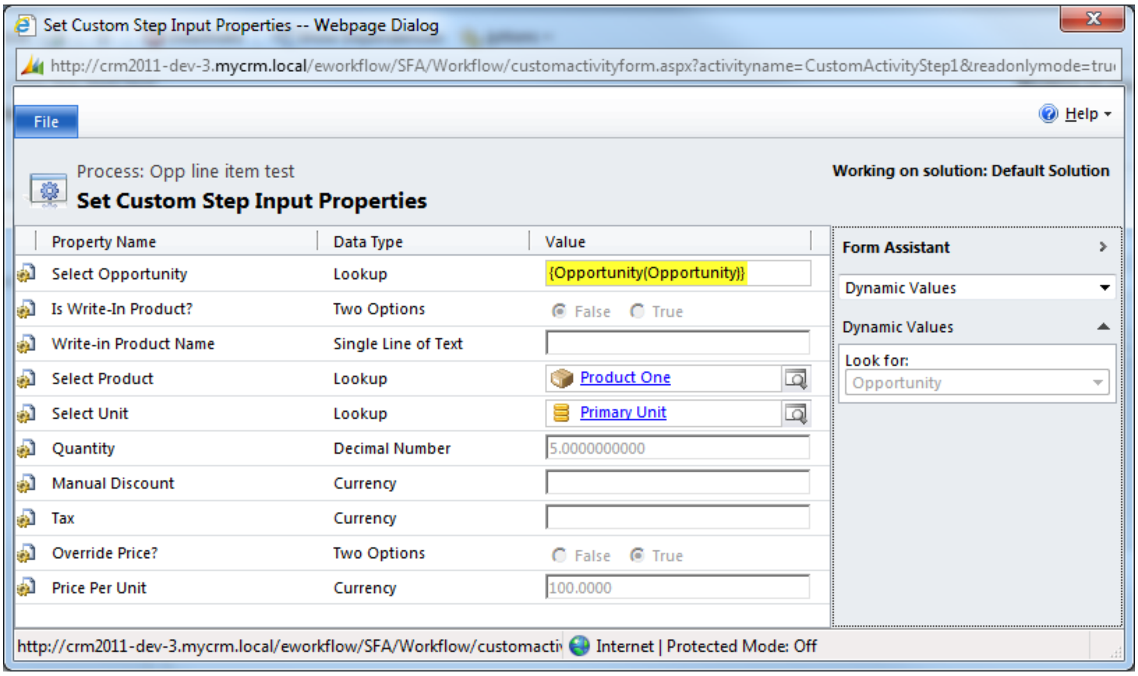
This function allows creation of Line Items on Opportunites.

The input parameters for the function mirror the standard CRM2011 Line Item form, and must be treated in the same way. The line item may be a write-in with a supplied name, and the price may be overridden if an actual product is supplied. In the case of a write-in product or an overridden price, the item price should be entered into the Price Per Unit field.

The Unit field is required when an actual product is supplied.

Function Call :- Input
Opportunity (Object)
Is Write-In Product? (yes/no)
Write-in Product Name (string)
Product (Object)
Unit (Object)
Quantity (decimal)
Manual Discount (currency)
Tax (currency)
Override Price? (yes/no)
Price Per Unit (currency)

Example 32.1



Function – AddOrderLineItem

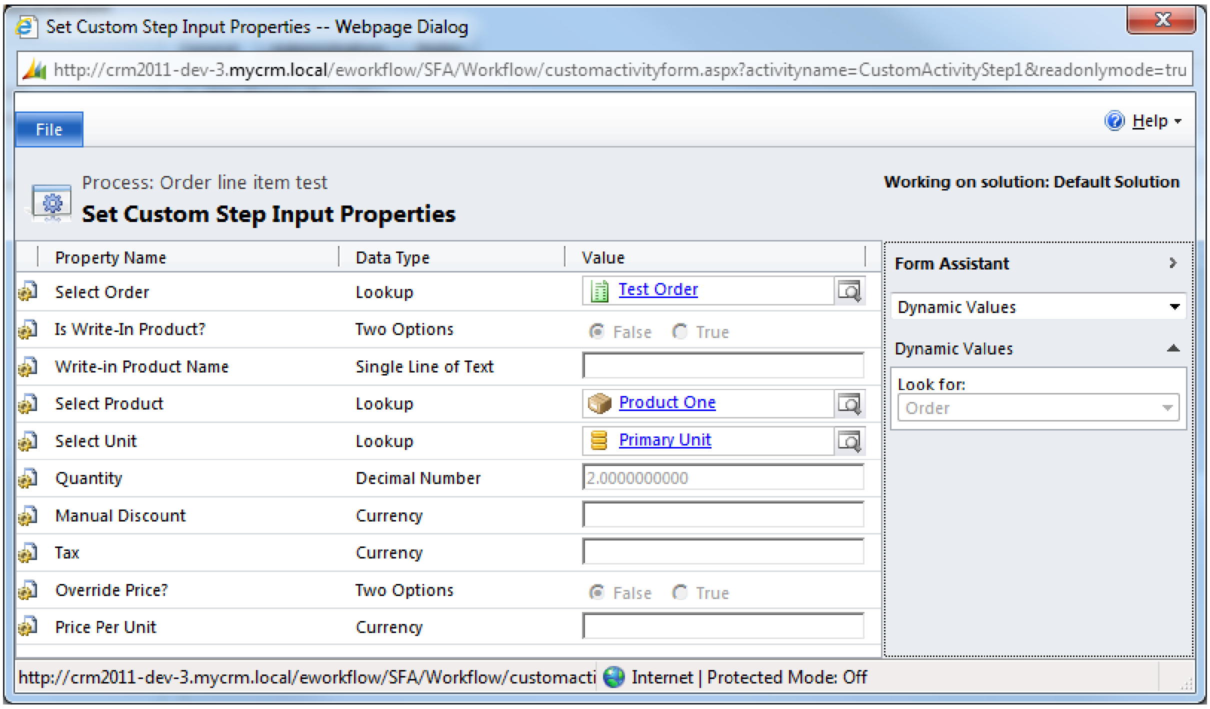
This function allows creation of Line Items on Orders.

The input parameters for the function mirror the standard CRM2011 Line Item form, and must be treated in the same way. The line item may be a write-in with a supplied name, and the price may be overridden if an actual product is supplied. In the case of a write-in product or an overridden price, the item price should be entered into the Price Per Unit field.

The Unit field is required when an actual product is supplied.

Function Call :- Input
Order (Object)
Is Write-In Product? (yes/no)
Write-in Product Name (string)
Product (Object)
Unit (Object)
Quantity (decimal)
Manual Discount (currency)
Tax (currency)
Override Price? (yes/no)
Price Per Unit (currency)

Example 33.1



Function – AddQuoteLineItem

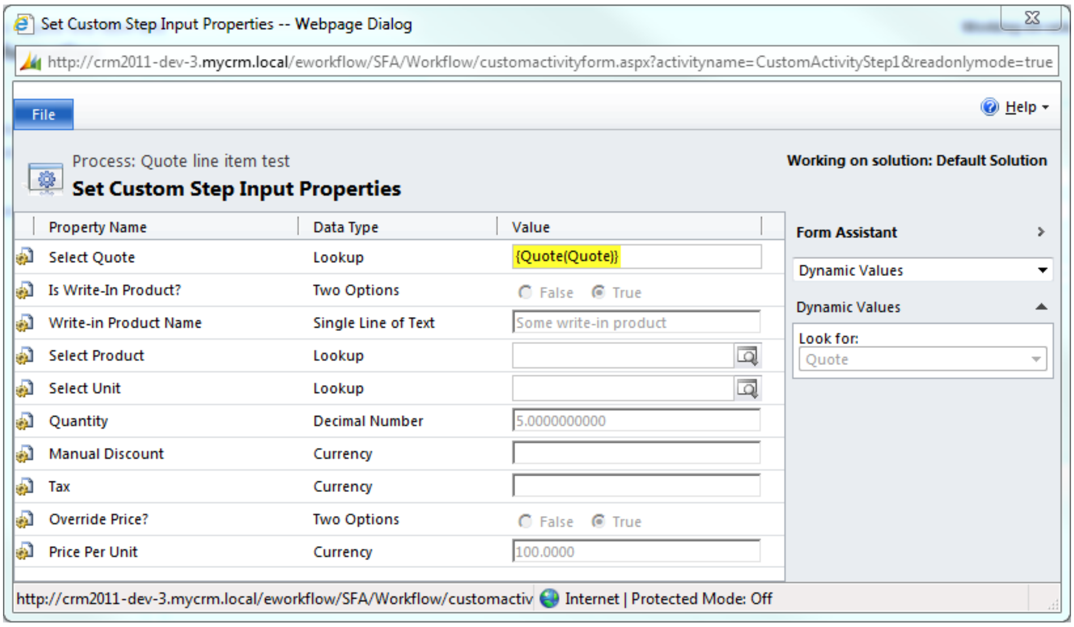
This function allows creation of Line Items on Quotes.

The input parameters for the function mirror the standard CRM2011 Line Item form, and must be treated in the same way. The line item may be a write-in with a supplied name, and the price may be overridden if an actual product is supplied. In the case of a write-in product or an overridden price, the item price should be entered into the Price Per Unit field.

The Unit field is required when an actual product is supplied.

Function Call :- Input
Quote (Object)
Is Write-In Product? (yes/no)
Write-in Product Name (string)
Product (Object)
Unit (Object)
Quantity (decimal)
Manual Discount (currency)
Tax (currency)
Override Price? (yes/no)
Price Per Unit (currency)

Example 34.1

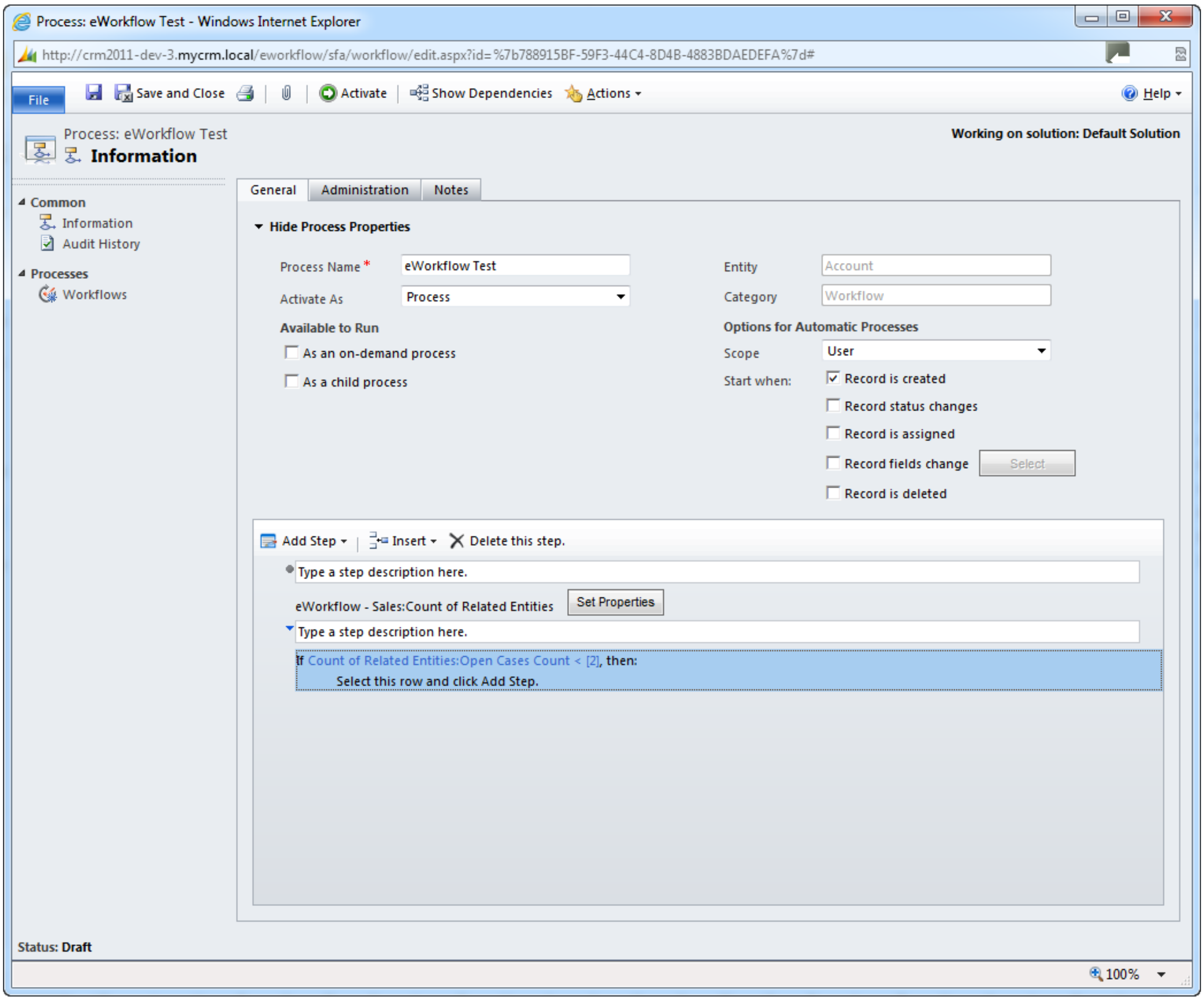


Function – CountRelatedRecords (Entities)

This function enables an end user to count associated records. This is a great addition if you need to build a Summary of totals for the Account or Contact.

Function Call :- Input
Account (Object)
Contact (Object)

Example 35.1



Function Call :- Output
Open Case Count (number)
Closed Case Count (number)
Opportunities Count (number)
Active Quotes Count (number)
Active Invoice Count (number)

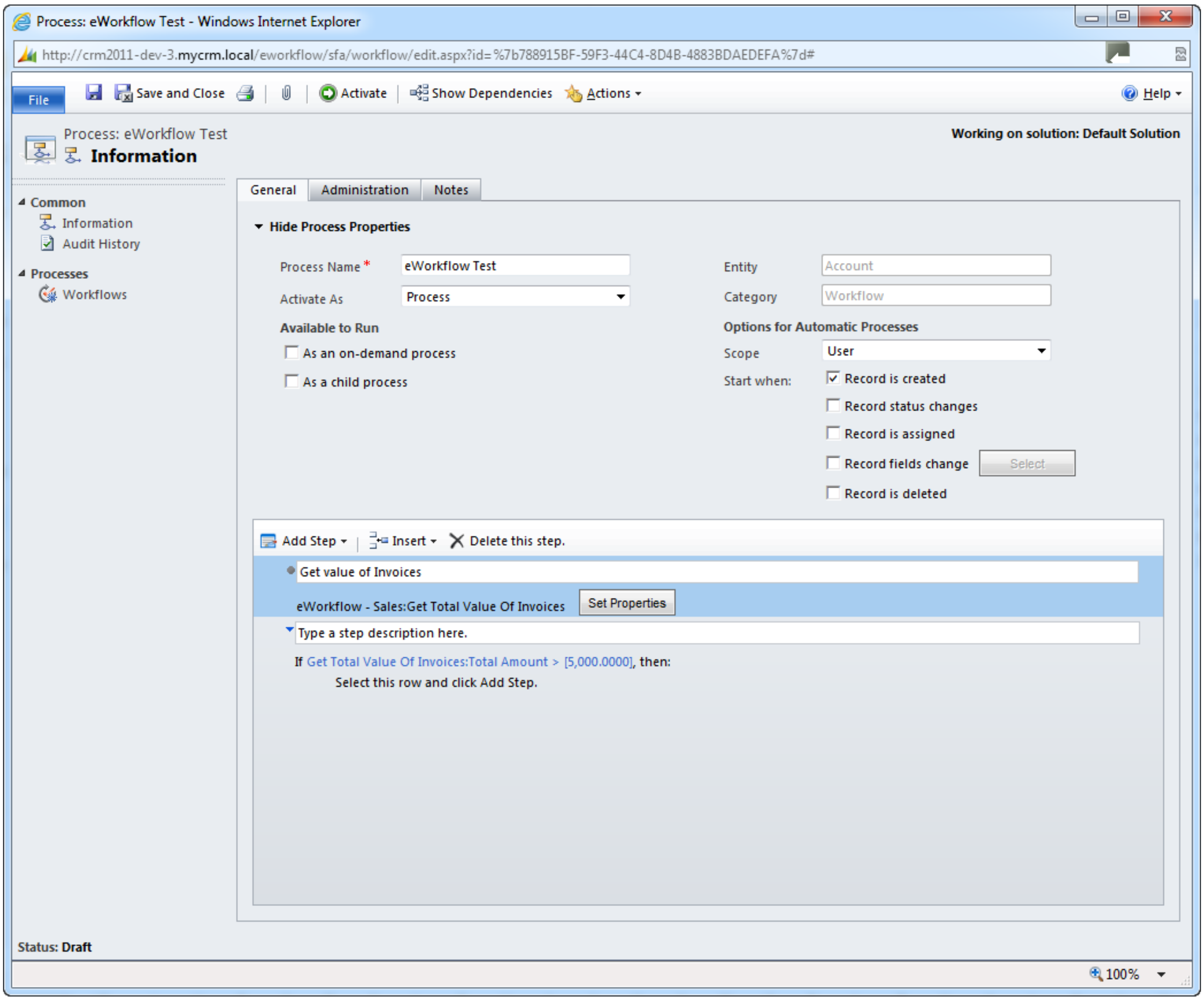
Active Orders Count (number)
Related Contacts Count (number)

Function – TotalValueOfInvoices

With the Total Invoice value count, you can now perform workflow actions based on the total value of current invoices for a customer, in the example we test to see if an account has reached the £5000 GBP spend

Function Call :- Input
Account (Object)
Contact (Object)

Example 36.1



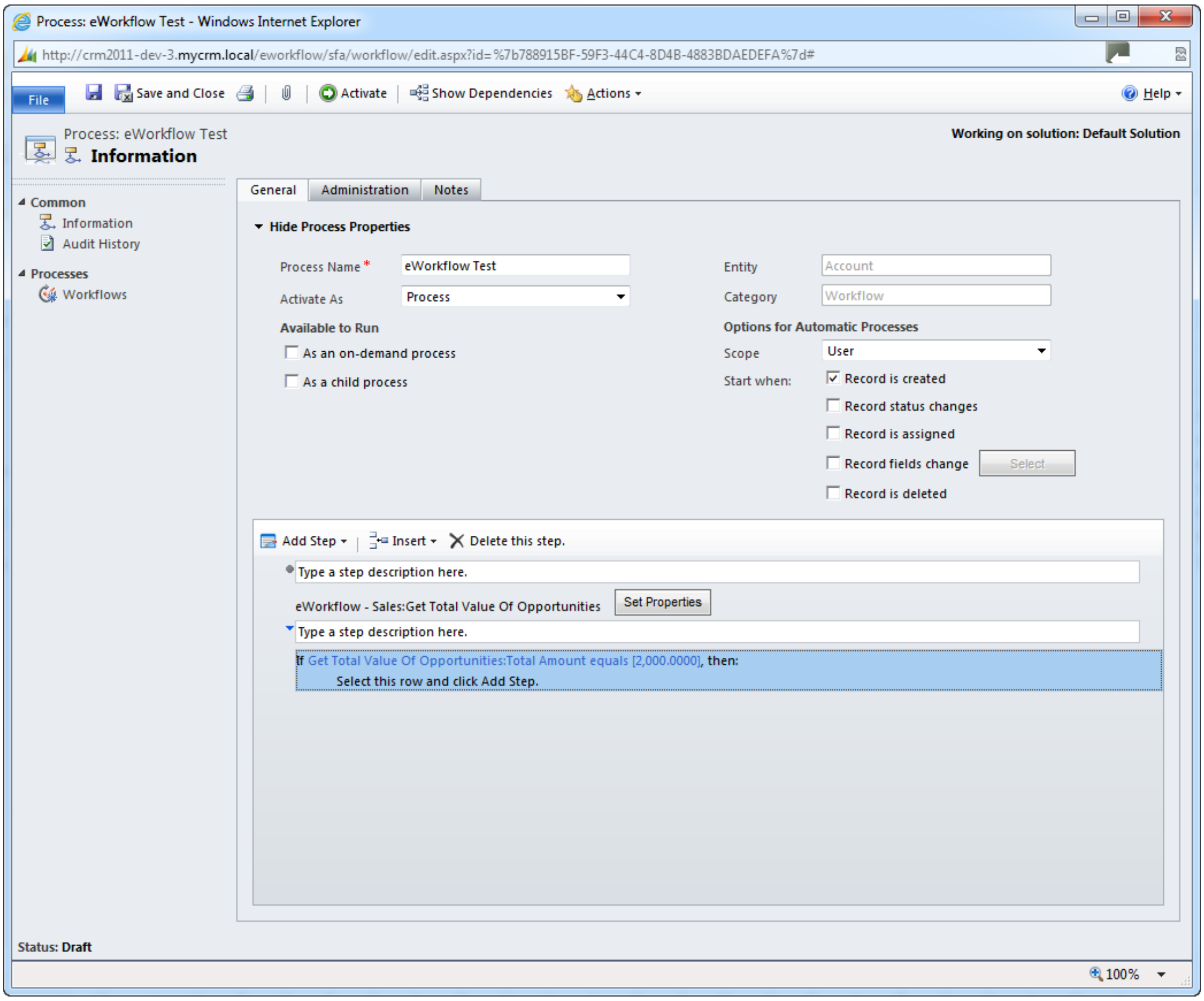
Function Call :- Output
Total Amount (Money)

Function – TotalValueOfOpportunities

With the Total Opportunity value count, you can now perform workflow actions based on the total value of current Opportunities for a customer, in the example we test to see if an account has reached the £2000 GBP Actual Value

Function Call :- Input
Account (Object)
Contact (Object)

Example 37.1



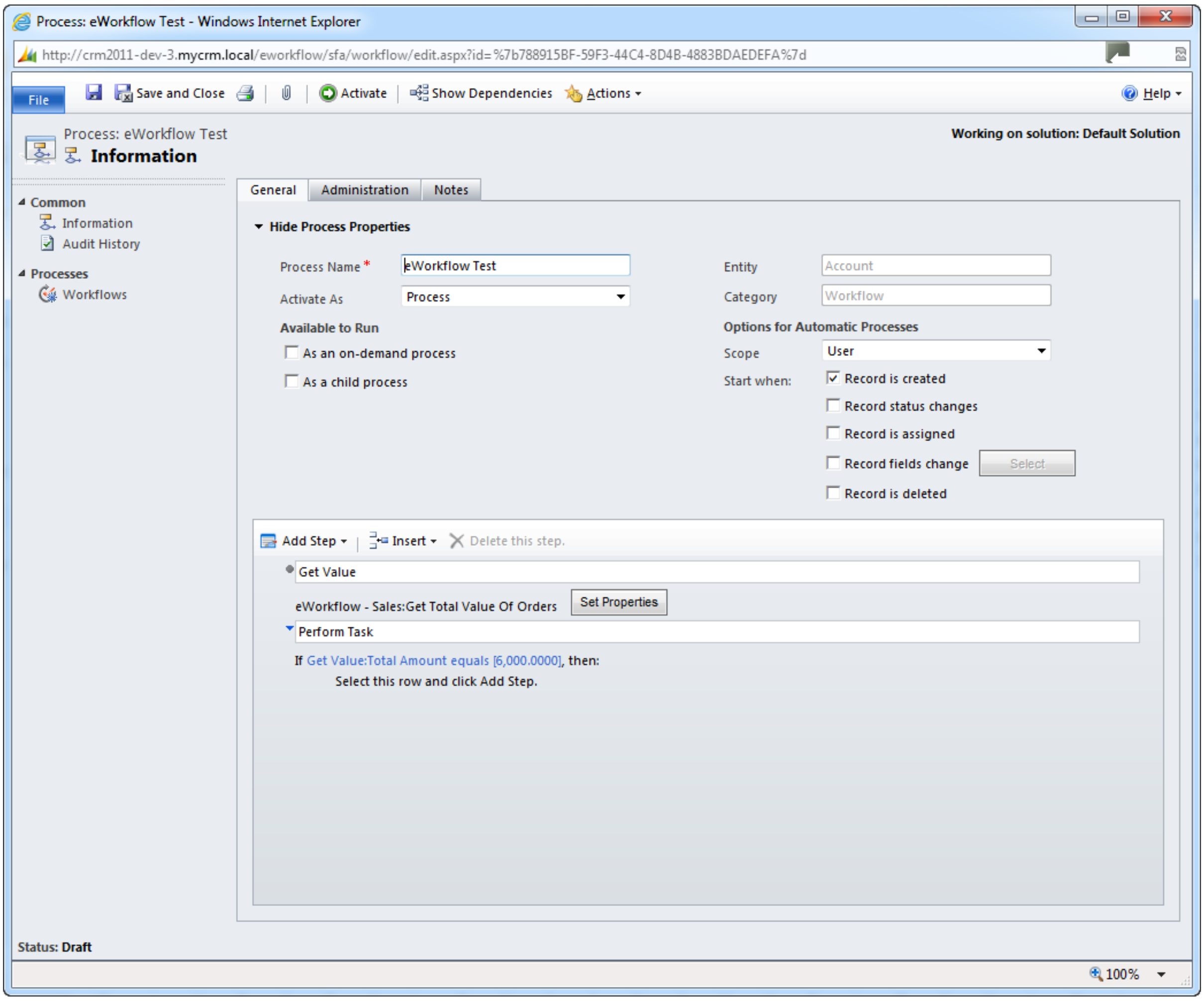
Function Call :- Output
Total Amount (Money)
Total Estimated Revenue (Money)

Function – TotalValueOfOrders

With the Total Order value count, you can now perform workflow actions based on the total value of current Orders for a customer, in the example we test to see if an account has reached the £6000 GBP Value

Function Call :- Input
Account (Object)
Contact (Object)

Example 38.1



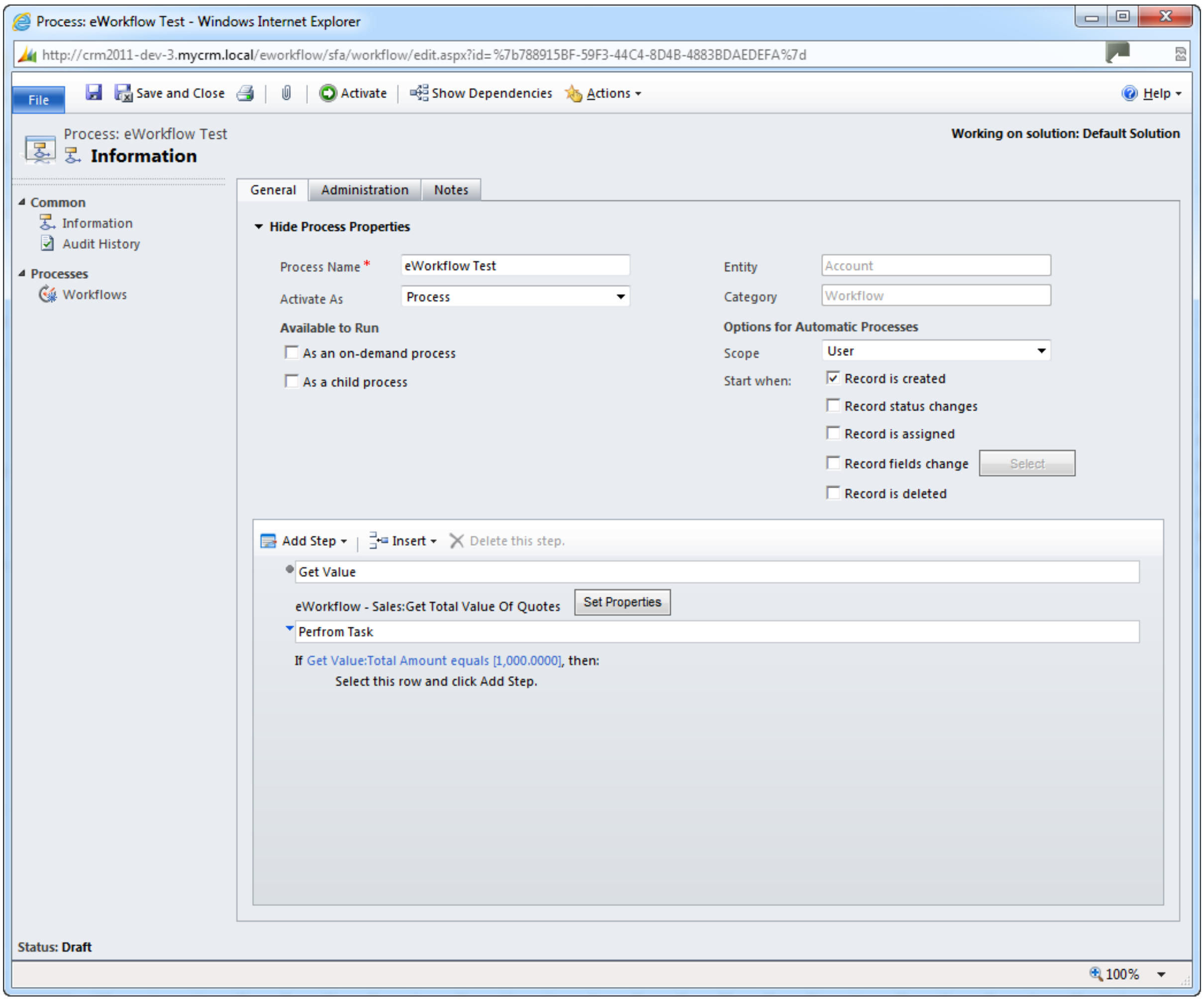
Function Call :- Output
Total Amount (Money)

Function – TotalValueOfQuotes

With the Total Quote value count, you can now perform workflow actions based on the total value of current Quotes for a customer, in the example we test to see if an account has reached the £1000 GBP Value

Function Call :- Input
Account (Object)
Contact (Object)

Example 39.1



Function Call :- Output
Total Amount (Money)

eWorkflow – Other

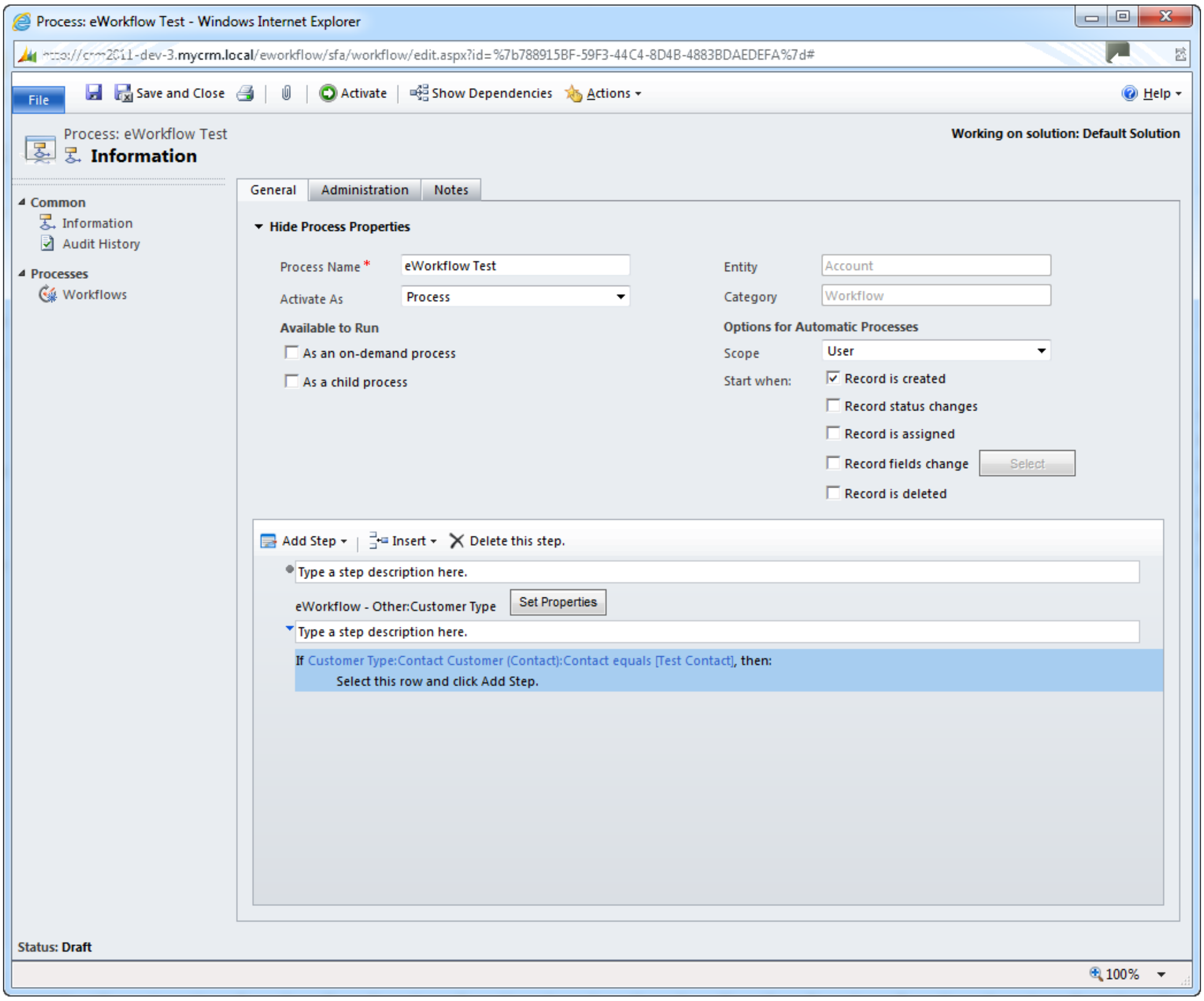
Function – CustomerType

When configuring and setting up business workflows there are a number of occasions that you may need to validate the customer type, currently this is not possible in CRM so this extended function can be used to validate a customer type and expose the related data for that customer (Account / Contact) in the workflow process.

An example may be when running workflows off the back of an Opportunity or Order and having to validate if the order is for a contact or an account

Function Call :- Input
N/A

Example 40.1



Function Call :- Output
CustomerType:Contact
CustomerType:Account
CustomerType (String)