Credit Hound – Codeless BPA Platform – Email Pack

Pre-requisites

- **Minimum Codeless BPA Platform version:** Codeless BPA Platform 2020 Update 1
- **Installed version of Credit Hound v6 (or above)**
    - System Requirements: https://draycir.com/credithound/spec#requirements

Email Configuration

These tasks all send emails, as such, they will all need a suitable Send Email configuration in the Output tools section of BPA Platform.

The email server must be configured with a suitable account and email address that can send these emails. This address and sender will also need to be set in the global variables section (See below for more details) so emails are sent from the correct address.

Data Connectors

You can import the '**Credit Hound Databases.connection**' file and configure connection to the two databases below:

Database Query OLEDB

There must be a link to your Credit Hound Company SQL database. Each task will need its query to be linked to this OLEDB connection.

Database Query ODBC

Some tasks will need to be linked to an ODBC connection for the Configuration.mdb Microsoft Access database. This is used to output and query user information in the outgoing email.

The Configuration.mdb Credit Hound database resides in C:\Program Data\Credit Hound Data by default. This is the default value when you import the task, but you can amend this. Please see the **Configuration Guide** section below.

Global Variables

In order to make the email alerts flexible to different needs, the Tasks make use of global variables to configure how the tasks work.

These global variables are described below.

Email Variables

EmailForCreditHoundAlerts

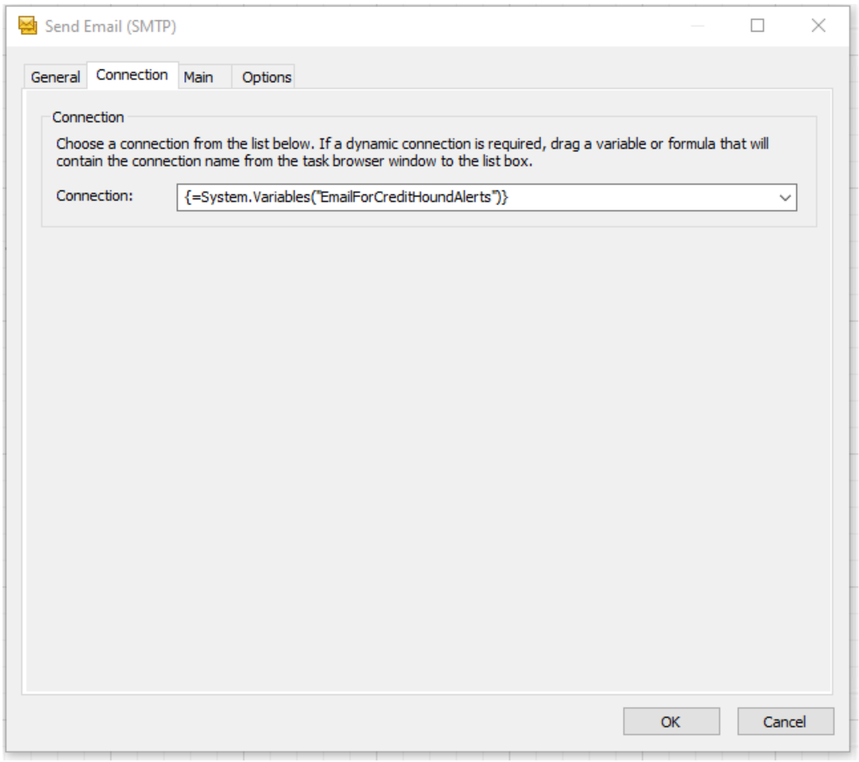
Email connection used for Credit Hound tasks.

This variable stores the name of the email connector. This allows the connection used to be reflected across all tasks without needing to edit individual tasks.

For example, in BPA Platform Client > Tools > Output > Send Email (SMTP), you may have a connection to your Office 365 Server. The name of the connection could be **Office 365**.

You can then provide this value to the EmailForCreditHoundAlerts variable as **Office 365** also.

Each task is configured to use the EmailForCreditHoundAlerts in its Send Email (SMTP) connection tab.



Connection

Choose a connection from the list below. If a dynamic connection is required, drag a variable or formula that will contain the connection name from the task browser window to the list box.

Connection: {=System.Variables("EmailForCreditHoundAlerts")}

This means you do not have to edit individual tasks but can just update the Global Variable in one place and all tasks will use this value.

AlertFromEmailAddress

The email address that sends out the alerts.

This must be a valid email address in the email server linked to by your global Email configuration to avoid SPAM filters and other security.

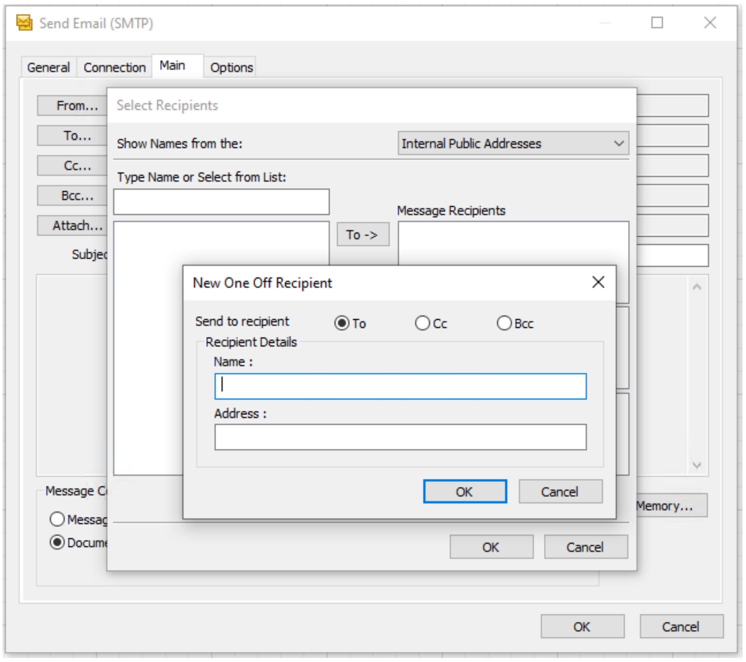
AlertFromName

The name of the sender of the alerts (also see AlertFromEmailAddress).

CreditHoundAddressList

List of email recipients for Credit Hound tasks. This can be used to determine which recipient email address you want to the email to be sent to. For example, currently this variable is defined as the BCC for each respective task.

You can configure a static email recipient in the Send Email (SMTP) step under 'New' and the To field:



Show Names from the: Internal Public Addresses

Send to recipient ● To ○ Cc ○ Bcc

Or you can specify the email recipient values in this Global Variable and make use of the **{=System.Variables("CreditHoundAddressList")}** field in the Name and Address above.

These are used in the BCC field of all outgoing reports.

Each email address in the list can be separated with a semi-colon:

address1@address.com; address2@address.com; address3@address.com;

EmailAddressToName

The 'To' name for Credit Hound task emails.

All addresses receiving the email reports are on the BCC. This name will be shown on each person's inbox as a generic 'To' name.

Other Variables

TransactionOverDays

The threshold for the number of days old a transaction is. Used in conjunction with LastChaseOverDays.

This variable is used in the task '*7. Credit Hound - Transactions over x days old have not been chased in the last n days.*'

LastChaseOverDays

Variable for specifying the number of days to use since a transaction was last chased.

This variable is used in the task '*7. Credit Hound - Transactions over x days old have not been chased in the last n days.'*

DisputeOverDays

The threshold for the number of days a transaction has been in dispute.

This variable is used in the task '*8. Credit Hound - A transaction has been in dispute for over x days.'*

InitialiseRepoFilePath

Path for the Save File step for tasks that use repositories. This will allow the task to populate the repository before the task is run and sends emails to recipients for the first time.

Repositories

For frequently run tasks, the email alerts are limited to new or changed data. These repositories store the previous results. These can be reset when required.

Credit Hound Emails

Once unpacked, the Alerts will be stored in a Task Folder called Credit Hound Alerts.

By default, all tasks will be set to disabled to avoid incorrect emails being sent while the system is being configured.

No.	Alert description
1	**Credit Hound – New customer has been synchronised into Credit Hound** This task checks if new customers have been synchronised to Credit Hound and sends an email notification accordingly. Allows users to know if they need to assign a group or category to the new customer in Credit Hound.
2	**Credit Hound - A customer has been put On Stop** This task checks if an account has been put On Stop within Credit Hound and sends an email reporting on those customers.
3	**Credit Hound – Customers over Credit Limits** This task generates an email alert to specified users if a customer has exceeded their Credit Limit. This task should be scheduled frequently so any necessary action can be taken promptly.

	Any customers currently over their credit limit will be added to an email. If the customer was included in a previous alert, they will not be included again.
4	**Credit Hound Notes - Separated into 4 scenarios according to the different Credit Hound note types:** **1. A new transaction note created by another Credit Hound user** This task sends an email reporting if new transaction notes has been created by another Credit Hound user. This can be used, for example, if a BI User or Limited user has added a note to a transaction. The user(s) will be sent the email based on the Group of customers they are assigned to.
	2. A new dispute note created by another Credit Hound user This task sends an email reporting if new dispute notes have been created by another Credit Hound user. The user(s) will be sent the email based on the Group of customers they are assigned to.
	3. A new callback note created by another Credit Hound user This task sends an email reporting if new callback notes have been created by another Credit Hound user. The user(s) will be sent the email based on the Group of customers they are assigned to.
	4. A new automated chasing note created by another Credit Hound user This task sends an email if new notes generated by letters produced by Rules and Actions (Chase Letters), or manual chasing (Follow Up Letters/Dispute Letters) have been created by another Credit Hound user. The user(s) will be sent the email based on the Group of customers they are assigned to.
5	**Credit Hound - Payment has been made for outstanding To-Do** This task sends an email reporting on when payments for a transaction are synchronised into Credit Hound that cleared To-Do items. The email is sent to the user assigned to the To-Do item.
6	**Credit Hound - A new task has been assigned to me**

	This task generates an email notification of all the To-Do items that have been assigned to a specified user. The email is sent to the user assigned to the To-Do item.
7	**Credit Hound - Transactions over x days old have not been chased in the last n days.** This task alerts relevant users if outstanding transactions have not been chased for a number of days. For example, transactions that are over 30 days old that have not been chased in the last 7 days. The user(s) will be sent the email based on the Group of customers they are assigned to. This task uses the global variables 'TransactionOverDays' and 'LastChaseOverDays'. If the days passed are over these values, the details are added to the email.
8	**Credit Hound - A transaction has been in dispute for over x days** This task alerts relevant users if outstanding transactions have been in dispute for a number of days. This task uses the global variable 'DisputeOverDays'. If the days passed are over this value, the details are added to the email.
9	**Credit Hound - A new category has been created** If a new Category has been created, this task generates a report to be sent to specified users (e.g., a manager). They can then query this and understand why before customers are assigned to the new category.
10	**Credit Hound - Summary of when rules were run** This alert sends a report email showing a summary of when the Rules configured in Credit Hound were last run.
11	**Credit Hound – Account notes added** This task reports on when Account notes have been added in the Credit Hound chase screen, so users can be alerted on these notes before having to go into Credit Hound.
12	**Credit Hound – Customer default contact changed**

	This task sends an email to specified users if a default contact of a customer has been changed.

Configuration Guide

- Import the **'Credit Hound Databases.connection'** file first and realign all the connectors for your environment
    a. Navigate to Data Connectors > Database Query (OLEDB)
        i. Double click 'Connection to Credit Hound Company Database'
        ii. Connection Tab > Reconfigure button
        iii. Change the Server name
        iv. Amend the authentication details
        v. Pick your Credit Hound COMP000x database
    b. Navigate to Data Connectors > Database Query (ODBC)
        i. Double click 'Connection to Configuration Database'
            1. This will point to C:\ProgramData\Draycir\Credit Hound Data\Configuration.mdb by default
            2. Change this path if required
        ii. Test connection
    c. Navigate to Output > Send Email (SMTP)
        i. Click Add
        ii. Call the connection something like Office 365
        iii. Server Address: smtp.office365.com
        iv. Check 'Enable Authentication'
        v. Put your Username and Password in
            1. The password may have to be an App Password on your Office 365 portal
        vi. Advanced
            1. Enable 'Use Secure SMTP if available'
            2. Port Number: 587
        vii. Test Connection

Then import the **'Credit Hound Email Alerts Export v1.0.tks'** file

- Configure your Variables
    - **EmailForCreditHoundAlerts**: Set this value to your Send Email (SMTP) connection name
        - E.g., Office 365
    - **AlertFromEmailAddress**
        - E.g., accounts@company.com
    - **AlertFromName**

        - E.g., John Doe
    - **CreditHoundAddressList**
        - List of email recipients of alerts which go to specific users
        - You can separate the email addresses by semi-colon
        - E.g., accounts@draycir.com; manvir.singh@draycir.com
        - These are initially configured as BCC email addresses in the respective tasks that use this variable but you can configure this as the To field also.
    - **EmailAddressToName**
        - All addresses receiving the email reports are on BCC. This variable can be used as the name will be shown on each person's inbox as a generic 'To' name.
    - **LastChaseOverDays**
        - Default value is 7
    - **TransactionOverDays**
        - Default value is 30
    - **DisputeOverDays**
        - Default value is 2
    - **InitialiseRepoFilePath**
        - If you want to populate the repositories with the existing Credit Hound data for first time use, you can replace the *Send Email (SMTP)* step with the *Save File – Initialise Repository* step in each task respectively. This variable is used to define the path for where you want to these Files to be saved to. See the 'Populate repositories for first time use' section below.

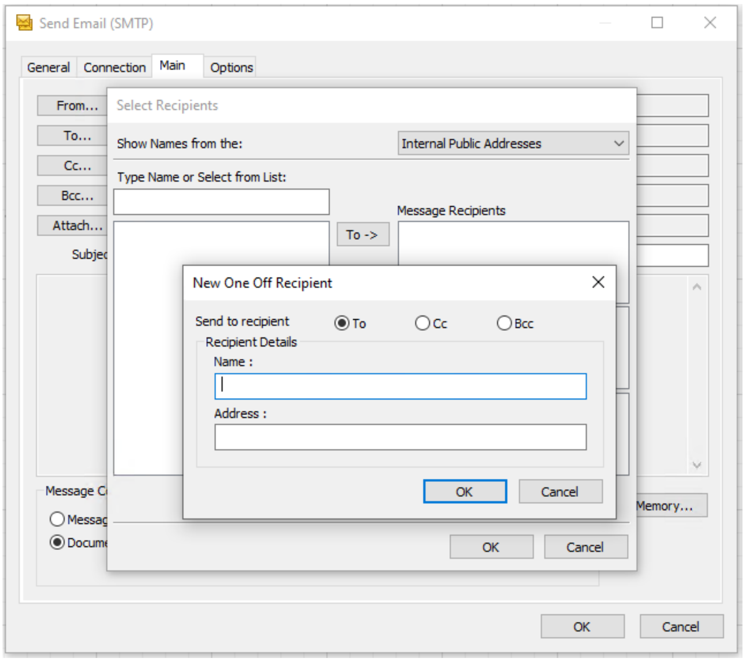
Tasks

- Go into **Database Query (OLEDB)** for each task and click the drop down for the Connection selecting the 'Connection to Credit Hound Company Database'.

Email recipients

- For tasks 4.1, 4.2, 4.3, 4.4 and 7, the emails are configured to be sent to the user(s) which is part of a Credit Hound Group. You will need to ensure your Credit Hound users are assigned to Groups in Credit Hound Manager to allow the email to go to the correct recipient.
    - The To field in each of these tasks will show the following:
        - Name: {=ThisStep.DocumentSource.RecordSet("Row_UserName")}
        - Address: {=ThisStep.DocumentSource.RecordSet("Row_UserEmail")}
- For tasks 5 and 6, the emails are configured to be sent to the user who owns the To-Do item in Credit Hound
    - The To field in both tasks will show
        - {=ThisStep.DocumentSource.RecordSet("Row_Username")}
        - {=ThisStep.DocumentSource.RecordSet("Row_Email")}
- For the other tasks that do not send an email to a specific user that is part of a Group in Credit Hound, or going to the owner of the To-Do item, configure your Email Recipients.

- You can configure a static email recipient in the Send Email (SMTP) step under 'New' and the 'To' field:



Show Names from the: Internal Public Addresses

Send to recipient ● To ○ Cc ○ Bcc

- Or you can specify the email recipient values in this Global Variable and make use of the **{=System.Variables("CreditHoundAddressList")}** field in the Name and Address above.

Populate repositories for first time use

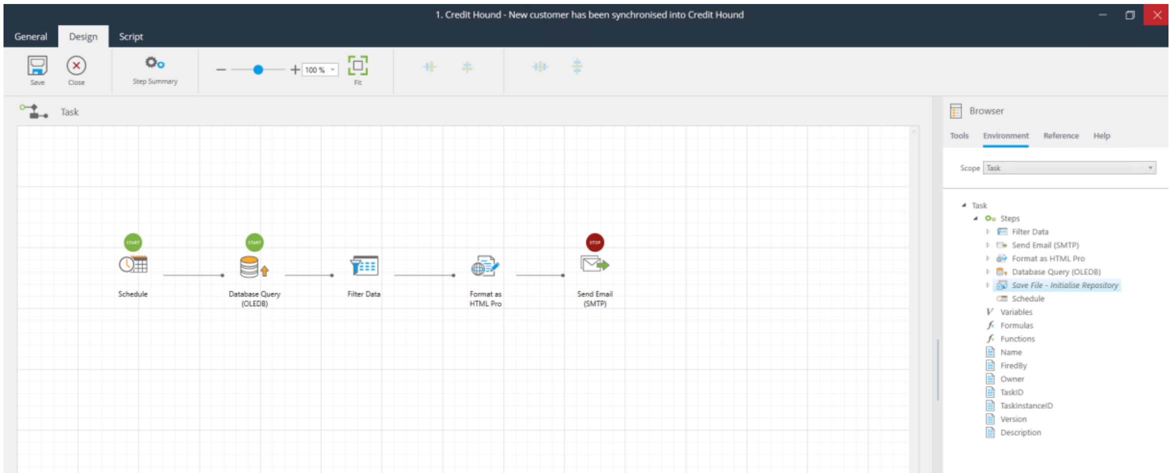
As the Tasks make use of Internal Memory and Repositories to filter out data so they don't appear on subsequent emails, the first time you run a task would populate the repository with all data accordingly.

If you do not want an email going out to your recipients with all this existing Credit Hound data, you can amend the task and the last step to *Save File – Initialise Repository* first so that the repositories are populated with this initial data.

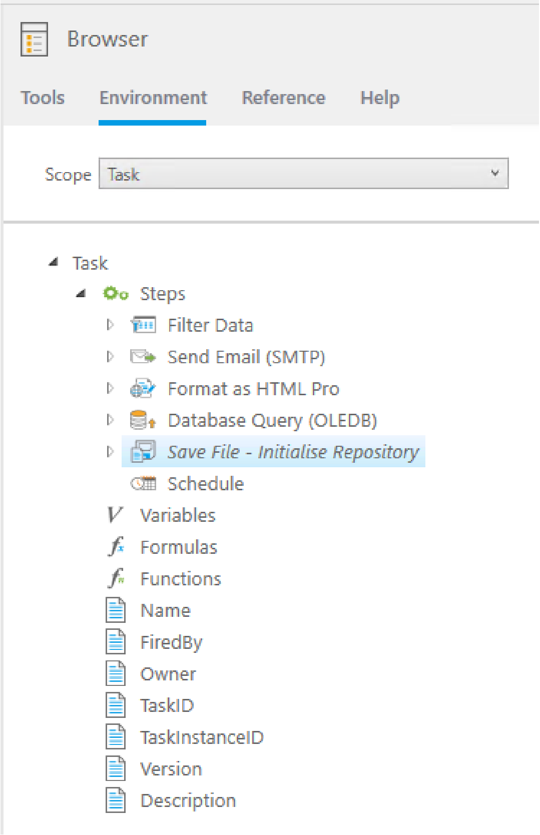
Once you have used Save File, you can re-configure the link to the Send Email (SMTP) step so that subsequent runs of the task only contain new data in the email to your recipients.

Steps

1. Go to your Task e.g., 1. Credit Hound – New customer has been synchronised into Credit Hound
   a. Your pre-defined task will be setup with a final step to **Send Email (SMTP)**

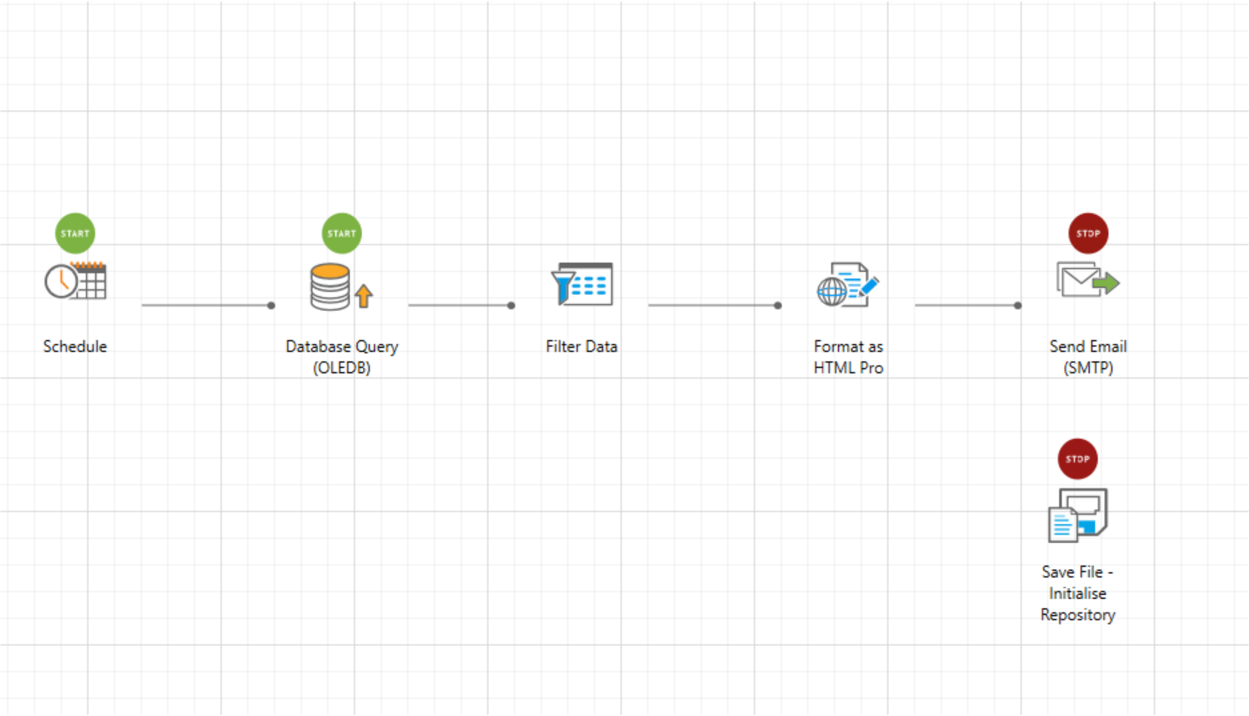


2. Go to the Browser > Environment and under 'Scope' select Task
3. Click on the ***Save File – Initialise Repository*** step and drag it into the main tasks area

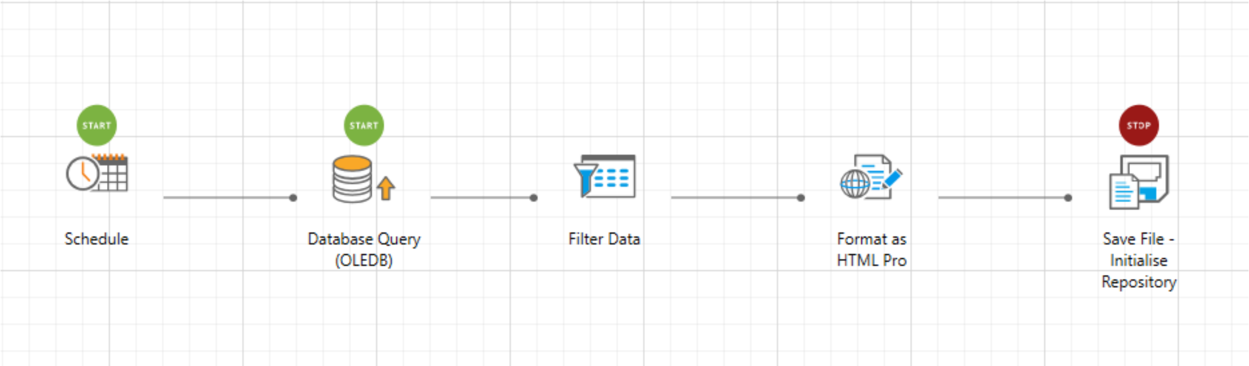


Scope Task

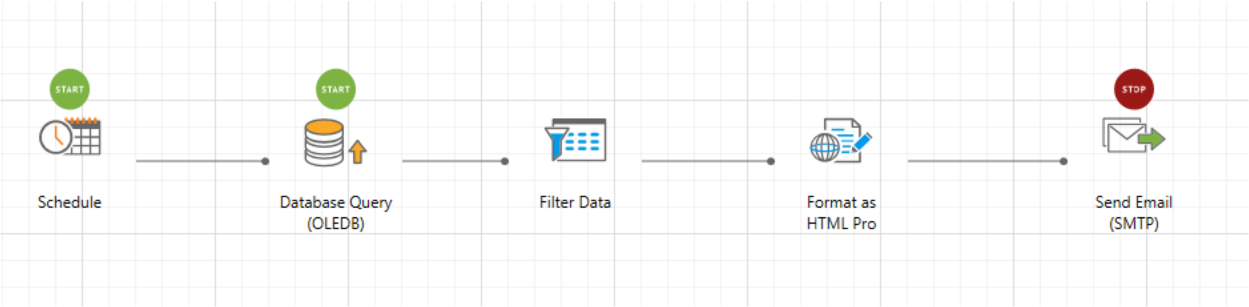
4. So it appears as below



5. Delete the **Send Email (SMTP)** step
6. Connect the Format as HTML Pro step to **Save File - Initialise Repository** like below

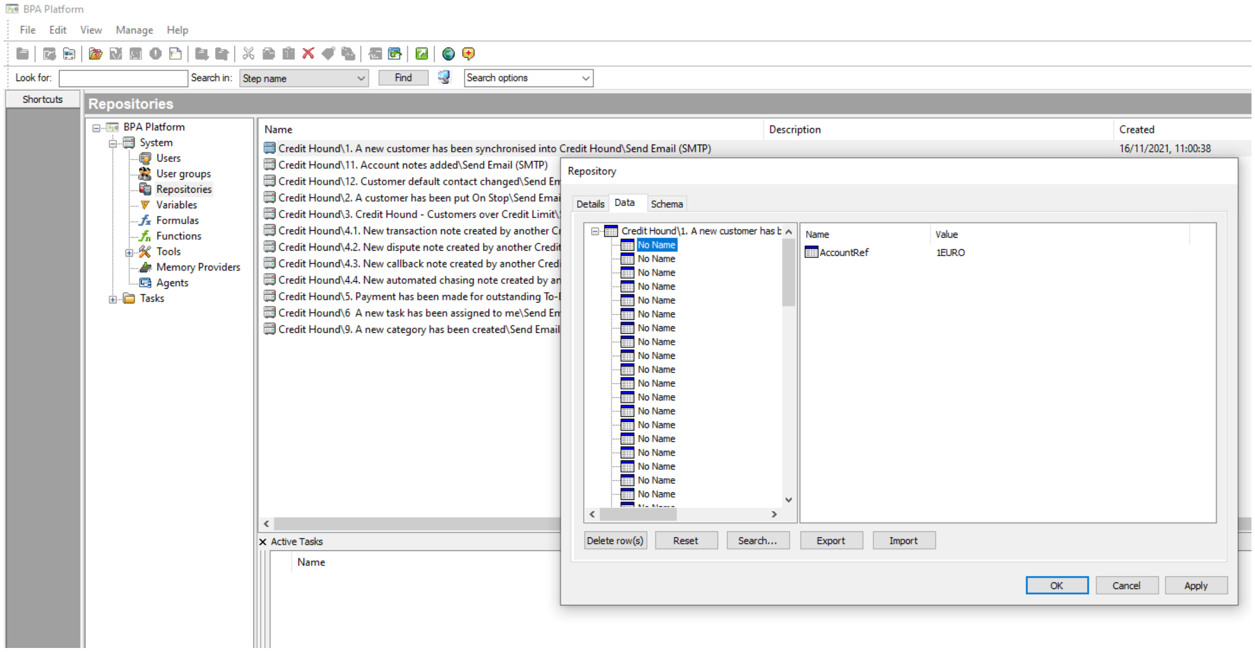


7. Enable the task and click Queue Now to start the Task process which creates the file and therefore populates the repository.
8. You can now go back into the Task and reconfigure it to send the email.
9. Drag in the 'Send Email (SMTP)' step back from the Browser > Under Scope select Task.
    a. Delete the 'Save File - Initialise Repository' step.
    b. Reconnect it 'Send Email (SMTP)' to 'Format as HTML Pro' like below



Repeat these steps for each Credit Hound Task that you would like to populate their respective repository with for the initial data.

Note: You can see the data that has been populated in the Repository by launching the **BPA Platform Client (Classic) application** and going to System > Repositories to and double clicking the Repository.



Name	Description	Created
Credit Hound\1. A new customer has been synchronised into Credit Hound\Send Email (SMTP)		16/11/2021, 11:00:38
Credit Hound\11. Account notes added\Send Email (SMTP)		
Credit Hound\12. Customer default contact changed\Send Em		
Credit Hound\2. A customer has been put On Stop\Send Emai		
Credit Hound\3. Credit Hound - Customers over Credit Limit\		
Credit Hound\4.1. New transaction note created by another Cr		
Credit Hound\4.2. New dispute note created by another Credit		
Credit Hound\4.3. New callback note created by another Credi		
Credit Hound\4.4. New automated chasing note created by an		
Credit Hound\5. Payment has been made for outstanding To-I		
Credit Hound\6 A new task has been assigned to me\Send Em		
Credit Hound\9. A new category has been created\Send Email		

Repository

Name	Value
AccountRef	1EURO

You can 'Reset' the repository from here also, to clear the repository of any data.