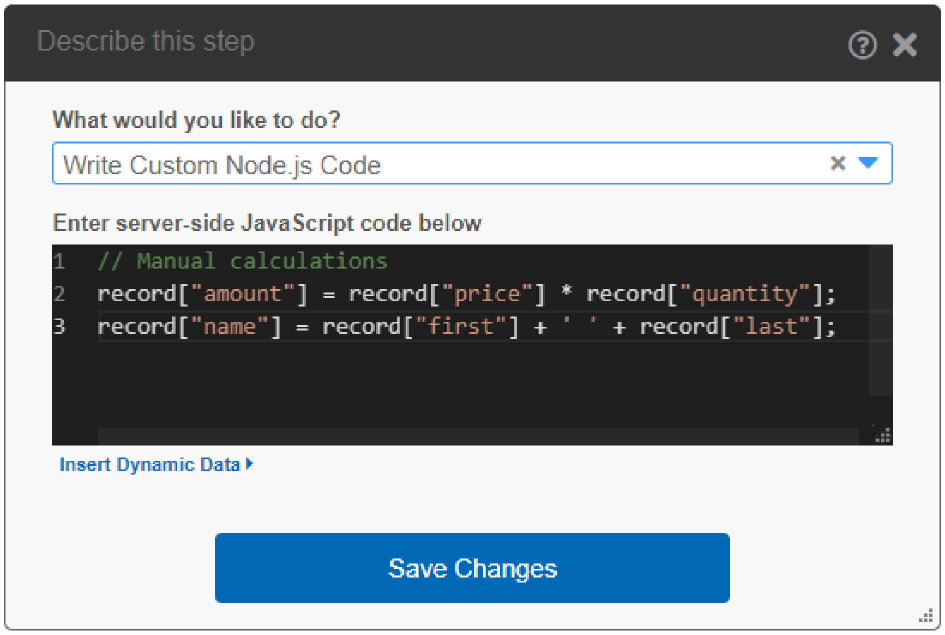
# Custom Code

In the Profound.js low-code environment, much of the functionality can be created without writing any code.

However, if you need to add functionality that is more advanced than what is provided by the plugins, any step can then be written as server-side or client-side JavaScript code. You can mix Code-based steps with No-Code steps at a granular level, without having to work with the entire application as code.

## Writing Server-side Code

To write Node.js (server-side JavaScript) code, select the “Write Custom Node.js Code” plugin under the “Custom” category when adding a step.

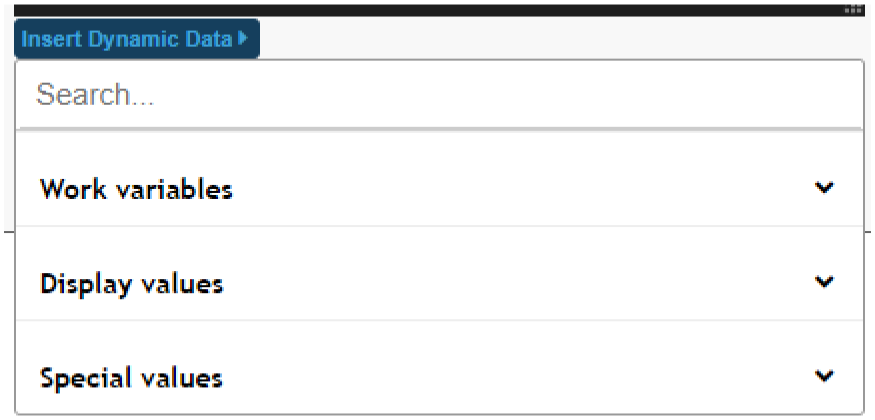

Describe this step
What would you like to do?
Write Custom Node.js Code
Enter server-side JavaScript code below
Insert Dynamic Data
Save Changes

```
// Manual calculations
record["amount"] = record["price"] * record["quantity"];
record["name"] = record["first"] + ' ' + record["last"];
```

You can use any JavaScript code, including direct calls to any of the pjs API.

If you need to write code for the start of a condition or loop that will encompass other steps, select the “Node.js Loop/Condition Partial”.

As you’re writing code, you can insert references from the low-code environment by clicking on the Insert Dynamic Data icon.



Work variables can be referred to directly by name in your code.

Screen fields can be referenced using the following syntax:

```
screenName["fieldName"]
```

Global properties can be accessed by using the following format:

```
globals["property name"]
```

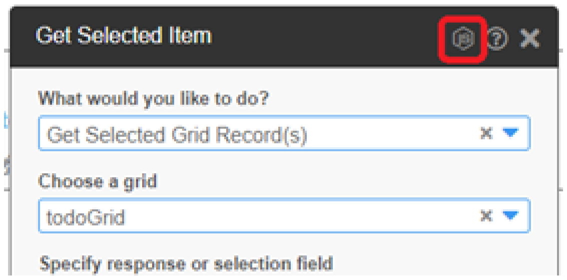
## Writing Client-side Code

To write client-side JavaScript code, select the “Write Client-side Code” plugin under the “Client-side” category when adding a step. If you need to write client-side code for the start of a condition or loop that will encompass other client-side steps, select the “Client-side Loop/Condition Partial”.

If the entire routine is custom code, it may be easier to just write the code directly in the widget event property, rather than creating a routine.

## Converting Existing Steps to Code

You can convert any existing low-code step to code by clicking the JS icon on the action dialog.


Get Selected Item

What would you like to do?
Get Selected Grid Record(s)

Choose a grid
todoGrid

Specify response or selection field

You can then customize the code and save your changes.

## Code Analysis

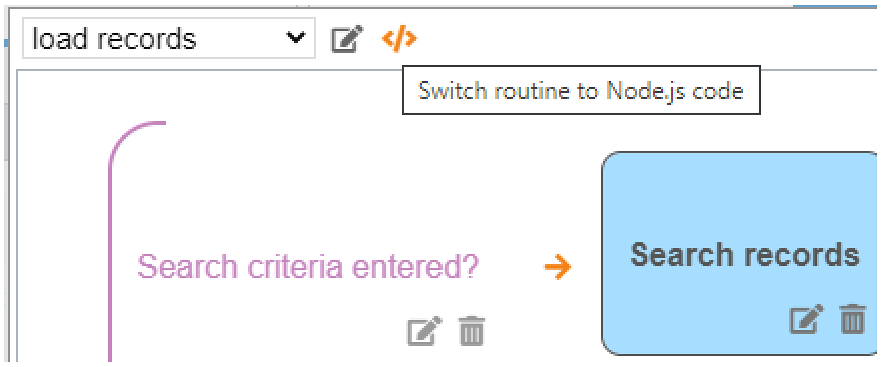
The Profound.js low-code environment will analyze the code you write in order to provide future suggestions for work variables and global properties. Only outer-level declarations will be analyzed. For example, if you write this line of code:

```
var maxRecords = 10;
```

Profound.js may start suggesting *maxRecords* as a value for answers to other low-code plugins.

## Convert a Routine to Code

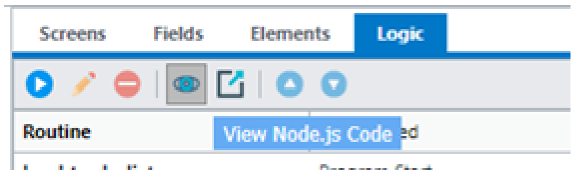
In addition to converting individual steps to code, you can also convert an entire routine to code. This is done by clicking the "Switch routine to Node.js code" icon. Client-side steps are not converted as part of this process.



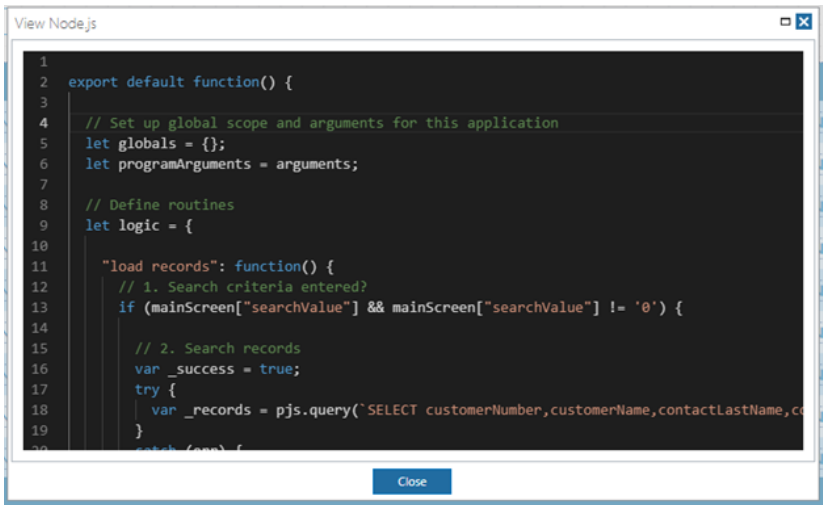
When you do this, the routine will have to be maintained manually with code going forward.

## Viewing the Entire Program as Code

You can view or export the auto-generated code for the entire application from the Logic tab by clicking on the corresponding toolbar icon.



The application will then be translated to Node.js.



```

export default function() {

  // Set up global scope and arguments for this application
  let globals = {};
  let programArguments = arguments;

  // Define routines
  let logic = {

    "load records": function() {
      // 1. Search criteria entered?
      if (mainScreen["searchValue"] && mainScreen["searchValue"] != '0') {

        // 2. Search records
        var _success = true;
        try {
          var _records = pjs.query(`SELECT customerNumber,customerName,contactLastName,c
        }
```

This can be useful in gaining an understanding of how low-code applications are structured and will help you in writing your custom routine steps.