

Resource  
Data Management

# JavaScript Editor Change Guide

File View About

Page 0

Internal Parameters

ON  
Off  
Zero  
One

Ten  
Twelve  
Seventeen  
One

Intruder Set

Air Temp

Pack 700

BCAST

Current Selection: T

Miscellaneous

Functional

I/O

Diagnostic

Mathematical

Time

Logic

## Contents

Introduction .....	3
New Features .....	3
Changes.....	3
Canvas Size.....	4
Using the Editor .....	4
File Tab .....	5
New Page.....	5
Item Order .....	5
Revision History .....	5
Save to Controller .....	5
Settings .....	5
View Tab .....	5
About Tab .....	5
Block Count.....	6
Online Mode .....	6
Search .....	6
Toolbox.....	6
Pages .....	6
Current Selection Indicator .....	6
Naming Blocks.....	7
Password Protected Programs.....	7
Code Block .....	8
Code Block Use.....	8



Please ensure all power is switched off before installing or maintaining this product.

## Introduction

This document should provide a quick guide to the changes within the JavaScript editor which is present in software versions Intuitive TDB 4.2 or later and DMTouch 4.0 or later.

This should allow existing users of previous editor software to be able to have a brief overview of the differences to be aware of.

The full user guide providing information on creating programs, individual blocks etc. can be found within the documentation for the Intuitive TDB controller and it is advisable to review that document also before using the editor due to the changes in functionality.

## New Features

Below is a collection of some of the new features to be aware of when using the editor:

- One of the main new features of this editor is that it does not require Java to run, so therefore internet browsers such as Chrome or Edge for example, can be used to access it. In order to see what the appearance will look like in more detail. Please see section [Using the Editor](#).
- Alongside its implementation of the TDB Desktop Editor V2.3.6, the code block has been added for use with this editor. The code block utilises LUA programming language and allows the user to create their own logic inside the block. This block can be utilised to help reduce the number of blocks used in a program by allowing for processes that are used multiple times in a program to be carried out within one block. Please see section [Code Block](#) for further information.
- The editor will now warn while trying to save if there are any naming clashes, which could lead to erroneous programming. Any blocks affected will need to be renamed before the program is able to save. Please see section [Naming Blocks](#) for more information on how this operates.
- Navigation of the canvas has been changed to incorporate different uses. The mouse scroll will now zoom in and out, while pressing and holding the mouse scroll button down and dragging the mouse, will now allow the user to freely navigate the canvas. The vertical and horizontal scroll bars will allow the user to move the canvas page as needed.
- A search bar has now been implemented as can be seen in [Using the Editor](#). This can be used to search for any items placed on the canvas using their given name. This can be used in aiding to resolve any naming clashes that may be present.
- A snap to grid option is present within the View Tab which, when enabled, will cause blocks when being placed or moved, to snap to the top left section of the closest square block on the canvas. This can be useful to keep the canvas organised and uniform.

## Changes

Below are a few of the changes to be aware of in functionality.

- The way in which menus are interacted with and areas such as adding pages, using item ordering etc. have been revamped. Menus are now accessed via the tabs displayed on the screen, the use of right clicking on the canvas is no longer required to access areas such as the toolbox. The right click of the mouse is now utilised only on blocks for accessing properties or deleting. These can be viewed within the [Using the Editor](#) section.
- The appearance of block icons have been updated
- Setting blocks when set to units such as "hrs:min" or "min:sec" the inputs for Value, Min and Max must be entered in the format of '---:--'. For example, if "hrs:min" is selected and a value of 2 hours and 30 minutes is required, this should be entered as "002:30". This will only show as such in the editor itself, will be displayed as "02:30" when viewing the controller parameters list.
- The way in which you unlock a password protected with an Edit password has been changed. This can be viewed within the [Password Protected Programs](#) section.

There are other changes incorporated also, which can be viewed within the full user guide as stated previously.



Please ensure all power is switched off before installing or maintaining this product.

Please be advised, that some features which were present within the previous editor, currently are not available within this editor. These include:

- Simulation Mode
- Split Wire
- Copy and Paste
- Find and Replace

These features are being looked at to possibly be included in for future development.

The option to rename some of the blocks that do not require a unique name, such as digital edge and Analog switch blocks, has also been removed in this editor.

## Canvas Size

Due to the functionality of the web browser, the canvas size per page on the editor is restricted in comparison to the V2.x.x PC Desktop Editor. If there is a program that requires using a large canvas size, the program logic should be spread over multiple pages rather than a single page to ensure it is within the canvas size. This may only affect users with legacy versions of large programs, created in older versions of the editor which did not allow for multiple pages.

When fully zoomed in on the page within the TDB Desktop Editor, the canvas space on any one page used for programs being uploaded to the controller, should be no more than 340cm by 340cm. If out with this, the program will function, however, the canvas space for each page will be scaled to fit within the allowable limits on the controller editor. This may impact the placement of blocks on the original program created, with blocks having their position adjusted to fit.

## Using the Editor

The first change to be aware of is the appearance of the editor. It also does not require Java to run so can be accessed via internet browsers such as Chrome for example.

After accessing the editor, it will look as below.

The screenshot shows the Intuitive Editor interface for 'Editor Test - Control Editor'. The interface includes a top navigation bar with a 'File' menu, 'View' and 'About' options, and a 'Pages' section. The main workspace is a grid-based canvas labeled 'Page 0' containing a ladder logic diagram with various blocks (e.g., 0:1, 0:2, 0:3, etc.). The right side of the interface features a 'Block Count' (40), 'Online Mode', and a 'Search' function. The bottom left corner has a 'Current Selection Indicator' showing a selected block. Annotations with arrows point to various parts of the interface: 'View Tab' (File, View, About), 'File Tab', 'Page Navigation' (left arrows), 'Canvas' (the main grid), 'Current Selection Indicator' (bottom left), 'About Tab', 'Pages' (top right), 'Block Count' (40), 'Online Mode', 'Search', and 'Toolbox' (right side icons).



Please ensure all power is switched off before installing or maintaining this product.

## File Tab

### New Page

Click this to create a new page on the program. This will bring up the pages tab where a new page name can be entered and then the return key pressed on the keyboard to create the new page.

### Item Order

This allows the user to manually change the item order of all visible blocks as they will appear within the "Control" section of the controller. This can be changed by choosing the required section from the header at the top, i.e. whether it is an input, output or parameter and then left clicking and dragging the desired item up or down the list as required. Click "Save" to keep changes, or "Cancel" to exit.

### Revision History

Revisions can be added, edited or deleted from here. Clicking on "New" will bring up the fields required to add a new revision line. By clicking on a previous revision line, "Edit" can be used to make any changes required. To use the delete button, click on a revision line first and then click this to remove it. Clicking on "Save" will store the changes, "Cancel" will exit the menu.

### Save to Controller

This will allow the program to be saved to the controller. Once clicked, below will be shown.

**Save to Controller ?**

Performing this action will overwrite the currently installed program.  
Enter the passwords desired for controlling file access.  
Leave the fields blank to disable password access.

Enter "Read" password  
|

Retype "Read" password

Enter "Edit" password

Retype "Edit" password

Confirm Cancel

### Settings

After selecting this option, the settings tab will open displaying "General" and "Expansion Boards". Click on these options to expand them, as shown to the right. The dropdown menus are then used to choose selections.

### View Tab

Within this dropdown menu, the user can:

- Choose to show the names of each block or not
- Choose to show the grid or not
- Enable/disable Snap To Grid.

While enabled, when placing or moving blocks, they will snap to the closest point of the grid on the canvas.

Each of the options above are selected or deselected by left-clicking on them, a check mark will denote if they are active.

### About Tab

Will display information regarding the TDB such as the current version.

**Settings**

General ^

Description  
PLC Program

Temperature Units  
Deg. C v

Refresh Timeout  
180  
Seconds

Status Input Type  
OV Return v

Expansion Boards ^

Board 1  
Not Installed v

Board 2  
Not Installed v

Board 3  
Not Installed v

Save Cancel



Please ensure all power is switched off before installing or maintaining this product.

## Block Count

Shows the current number of blocks used in the program. The maximum number of blocks an Intuitive TDB device can have is 10,000. Of which a maximum of 2000 items (I/O, states & parameters) can be made visible.

## Online Mode

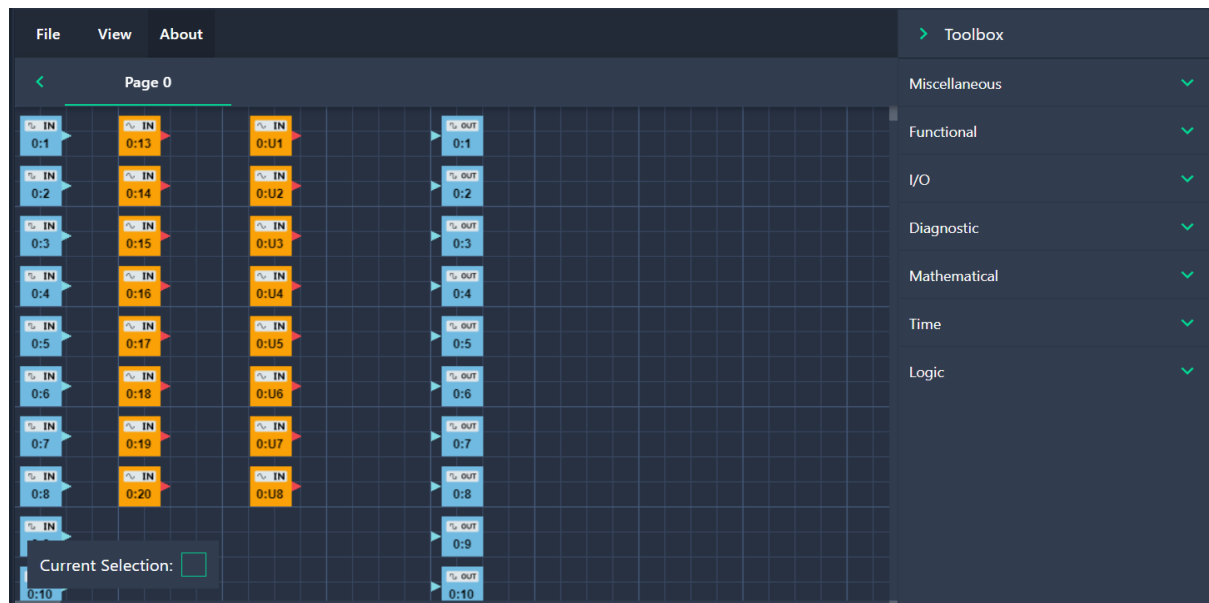
Left click on this icon to use 'Online Mode' which allows the user to see how the program is operating in real time. This item will only function with 'running' programs. Once online, rolling the mouse over start or end points will return the value at that point. Left clicking on this button while in online mode will switch back to edit mode. This can be a useful tool to prove program logic or in any fault finding that may need to be carried out.

## Search

Will bring up a search bar which can be used to find specific blocks or static text placed using its full name. Type the name of the block into the search bar and then press the return key on the keyboard. The requested block will then be highlighted in the program. Click the "X" to close.

## Toolbox

Brings up the toolbox, which contains all of the blocks that can be placed in the program.



Clicking on each menu header will expand that menu and display the blocks in that section, which are detailed further in this manual. Clicking on the header again will minimise that section.

Click the arrow beside the 'Toolbox' header to close the toolbox.

## Pages

The user can view, add, delete and rename the pages from this section.

All pages will be displayed here, right clicking on one will allow the page to either be renamed or deleted. Please note, that if only one page present, delete cannot be used.

To add a new page, the "+" to the top right of the 'Pages' menu is used. This will create a new line on the page where the name should be entered, the return key pressed on the keyboard and then the user will be taken to the newly created page. The arrow next to the 'Pages' header is used to close the menu.

## Current Selection Indicator

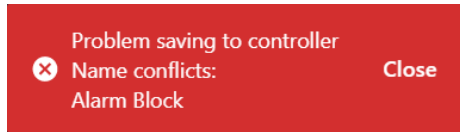
Displays what block is currently selected and what will be placed if the user left-clicks on the canvas with their mouse. With the box showing as blank, no block is currently selected and the mouse pointer can be used for general editing and navigation. In order to set this to blank after placing blocks previously, left click on the box.



Please ensure all power is switched off before installing or maintaining this product.

## Naming Blocks

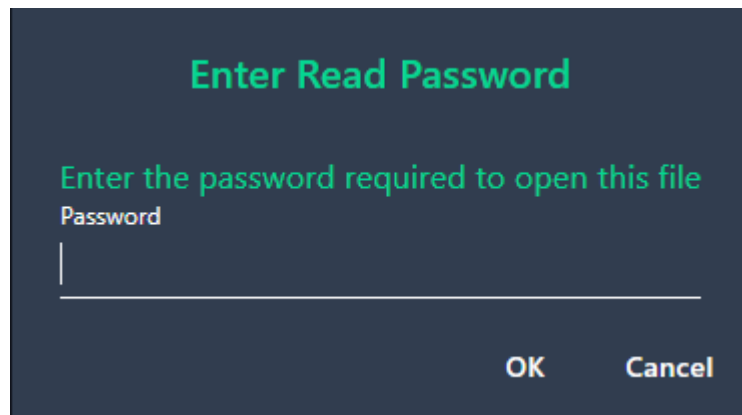
Any block that generates an external input or output, parameter or certain blocks that carry out saves internally that are placed in the program, will require a unique name given. If two or more blocks are placed with the same name, a warning message similar to below will be displayed when trying to save the program.



Any naming conflicts will need to be corrected before the program can be saved. This could be done by using the search bar to find the blocks in question and changing the name as required. Any block that has the option to set to 'Internal' and is set as such, will not be subject to a naming conflict check.

## Password Protected Programs

If a program has been given a read or edit password, this will need to be entered upon clicking on the 'Edit' option within the 'TDB' menu of the controller. If a read password has been set, the following screen will be shown after doing so.



The password should be entered and then 'OK' left clicked to view the program.

If an edit password has been set, the 'File' tab button will be unavailable to interact with and the toolbox icon will display with a red padlock next to it as shown.



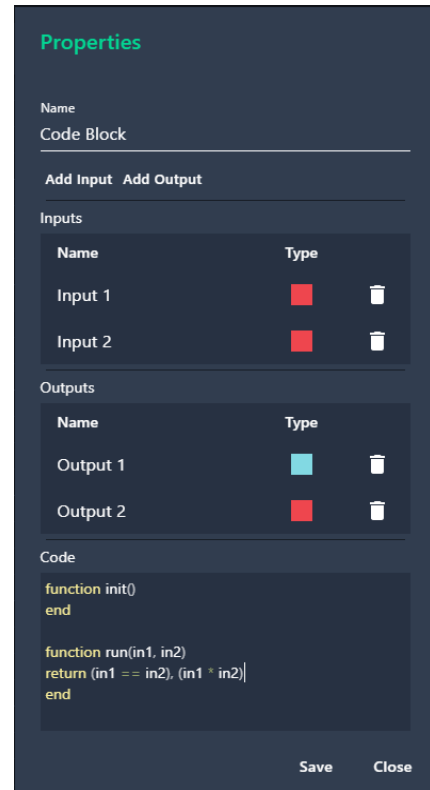
Left-click on this icon where the user will then be prompted to enter the edit password and click 'OK' to allow editing.



Please ensure all power is switched off before installing or maintaining this product.

## Code Block

Icon	Properties
<b>Code Block</b>	<p>The code block can be used to carry out mathematical functions using multiple inputs to return multiple or singular outputs. This block can be utilised to help reduce the amount of blocks used in a program by allowing for processes that are used multiple times in a program to be carried out within one block, as shown in section below.</p> <p><b>Inputs and Outputs</b></p> <p>Inputs and outputs are added to the block using the "Add Input" and "Add Output" buttons. They can then be set to either 'Digital' or 'Analog' via clicking on the type colour-blue for Digital, red for Analog. The recycle bin icon beside each input or output removes it. Please note, that in order to delete an input or output, all others in that category below them must be disconnected from wires.</p> <p><b>Code</b></p> <p>There are two functions within the block, 'init' and 'run'. The 'init' function is called when the program starts so it can be used to do a calculation needed at the start or to allow for a value to be used that is not an input to the block.</p> <p>The 'run' function is called each time the program loops. The parameters to the function need to correlate with the inputs that have been set up in the block i.e. 'in1' will be the first input, 'in2' the second input and so on. The inputs can still be named as required. The outputs come from what is returned from the function and again need to correlate with what is returned.</p> <p>In the example above, this would take in1 and in2 and return 'true' if they are the same, 'false' if not for the first output. The two values for in1 and in2 would be multiplied for the second output.</p> <p>The language used for this block is LUA, for the full functions available, please refer to <a href="https://www.lua.org/manual/5.1/manual.html">https://www.lua.org/manual/5.1/manual.html</a>. Please be advised that the only standard library utilised by the block is 'math'.</p> <p><b>Note 1:</b> Block only supported in TDB Desktop Editor V2.3.6 or later and in Plant TDB V4.2 or later and DM 4.0 or later.</p>



## Code Block Use

In addition to the [Code Block](#) section above, this part of the User Guide provides a more complex example of how the code block can be used and further information on its functionality.

As stated previously, the main use of the block is to help reduce the number of blocks used within a program, by allowing the user to perform processes that would previously need to be carried out multiple times within the program, to be done within a single block. In the example below, the code block is used to take 10 Analog inputs and either returns one of them or the value within "Chain", depending on the "Sel" and "Tens" values. A similar, albeit not exact same, piece of logic can be created utilising other blocks within the TDB Editor, however it would require the use of 147 blocks, as opposed to just 29 if placed as shown on the next page.



Please ensure all power is switched off before installing or maintaining this product.

The code block has been set up with 13 inputs:

"Input 1" through to "Input 10" which will create a list of inputs from which the block can use as an output.

"Sel" which acts as an input selector for the block, i.e., this value is the starting point for the logic to choose the output. Depending on what this value is, will determine which input from the first 10 will be given to the output.

"Chain" could be used to daisy chain blocks. For example, another code block could be setup similar to above but with different values for the 10 inputs and therefore providing a different output which could then be fed into this block.

"Tens" is used via the code block's mathematical capabilities in order to allow multiples of ten to be used in the input selector.

It has also been given one Output.

**Note:** Currently, there is a maximum of 16 inputs and 16 outputs that can be added to a single Code block.

Each Input has then been connected to a setting block as shown and an Analog display block connected to the output.

### Setting Up a Run Function

In the image to the right is the function used in order to carry out the desired action as described above.

In the first line of the "run" function, the parameters for the function are set out by listing the inputs that are being used.

In the second line, this is defining a parameter "a" as a list of those ten inputs to be used later in the function.

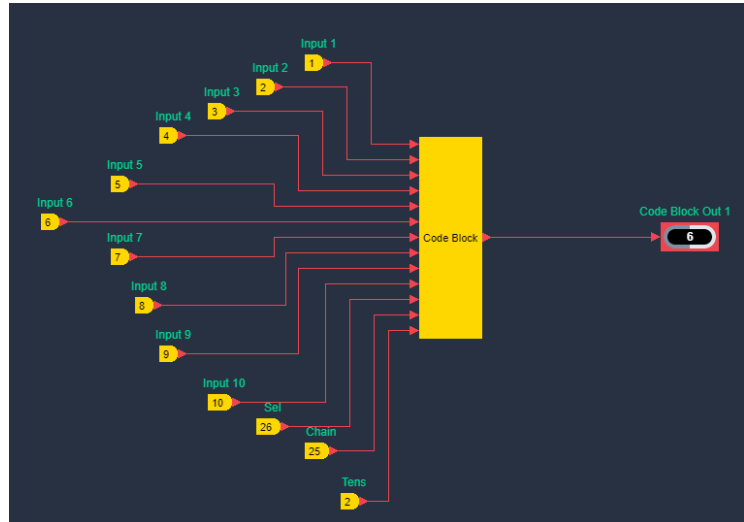
The third line is carrying out a mathematical equation. The function of `math.floor` is to return the largest integer smaller than or equal to  $x$  in the equation "`math.floor(x)`". Therefore, this part of the function is identifying that "t" should equal the largest integer smaller than or equal to the value represented by "sel" when divided by 10. In this example, the result would be 2.

The fourth line defines that "i" will equal the value "sel" modulo 10. In this example, this would be 6, as 26 divided by 10 has a quotient of 2 and a remainder of 6.

The rest of the function is performing an if statement using the parameters set out beforehand, in order to determine what the output will be.

Therefore, in this example, it will be saying if the value "t" is equal to the value in the input "tens", then the output will return "i" from the variables set up in "a". This is because in LUA, square brackets are used to index a table or list. So in the example, "`a[i]`" will return 6 as it is being asked to return the sixth entry "[i]" from the list "a" which happens to be the value 6 for the purposes of the example.

The second part of the if statement, advises that if the first part is not true, i.e. "t" does not equal the value "tens", then the output should return the value represented by "chain" which in the example would be 25.



### Properties

Input 10	■	🗑️
Sel	■	🗑️
Chain	■	🗑️
Tens	■	🗑️

**Outputs**

Name	Type
Output 1	■

**Code**

```

function init()
end

function run(in1, in2, in3, in4, in5, in6, in7, in8, in9, in10, sel, chain, tens)
  a = {in1, in2, in3, in4, in5, in6, in7, in8, in9, in10}
  t = math.floor(sel / 10)
  i = sel % 10

  if (t == tens) then
    return a[i]
  else
    return chain
  end
end

```

Save
Close



Please ensure all power is switched off before installing or maintaining this product.

As seen above, the function has returned the value in Input 6 as "t" had a value of 2 which was equal to the value of "tens" which was also 2. For the use of the tens to work properly in this example, the tens would need to match the multiple of ten entered into the input select. I.e. to get an output of 6 with the "sel" set to 36, tens would need to be 3.

The function must always have "end" in order to signify the end of the run function, but it must be used to show the end of if statements etc also. As shown in the example above, two ends are used to denote the end of the if statement and then also the function itself.

### Init Function

When using the init function, anything here should be added into the body of the function to then be used in the run function. This is to allow variables to be used that are not inputs to the block. So for example, if the value of 5 and 14 were needed for calculations within the run function but were not inputs, they could be added to the init function as so:

```
Function init()  
    a = 5  
    b = 14  
end
```

These could then be used in the run function by calling for either "a" or "b".

### Programming

The use of loops within a function should be used with caution as these could cause problems with block functionality and could stop the function within the code block from running. This would be statements such as "While true do".

There is a maximum of 16 Code Blocks that can be added to a program, however performance may be affected dependent on the functionality within each block.

Care should be taken when programming this block as currently, there is no simulation mode available to test functionality, although the Online Mode can be used to aid in programming. Therefore, a knowledge of LUA is advisable in order to be aware of the basic functions. As advised previously, the manual can be found at <https://www.lua.org/manual/5.1/manual.html> and there are also videos that can be found online in order to aid understanding of how to use LUA effectively.



Please ensure all power is switched off before installing or maintaining this product.

Group Offices

RDM Group Head Office  
80 Johnstone Avenue  
Hillington Industrial Estate  
Glasgow  
G52 4NZ  
United Kingdom  
+44 (0)141 810 2828  
[support@resourcedm.com](mailto:support@resourcedm.com)

RDM USA  
9441 Science Center Drive  
New Hope  
Minneapolis  
MN 55428  
United States  
+1 612 354 3923  
[usasupport@resourcedm.com](mailto:usasupport@resourcedm.com)

RDM Asia  
Sky Park at One City  
Jalan USJ 25/1  
47650 Subang Jaya  
Selangor  
Malaysia  
+603 5022 3188  
[asiatech@resourcedm.com](mailto:asiatech@resourcedm.com)



Visit [www.resourcedm.com/support](http://www.resourcedm.com/support) for more information on RDM solutions, additional product documentation and software downloads.

While every effort is made to ensure the information given within this document is accurate, Resource Data Management Ltd shall not be liable for errors or omissions, for incidental or consequential damages, directly or indirectly, in connection with the furnishing, performance or misuse of this product or document. All specifications are subject to change without notice.

See [www.resourcedm.com](http://www.resourcedm.com) for terms and conditions of sales.

Copyright © Resource Data Management