

Surface Automation of Terminal Emulators

Introduction

When interfacing with a Terminal Emulator installed locally the Blue Prism Mainframe interface, using an API called Hllapi, can be used to easily read and send data to the emulator session.

For a non-Hllapi compliant emulator or an emulator accessed via a thin client technology such as Citrix, Surface Automation techniques will need to be used instead. How to use Surface Automation to interface with Terminal Emulators is the subject of this guide.

There are a number of techniques available for interfacing with a terminal emulator using Surface Automation, these are:

- Using the Windows Clipboard to copy all the text from the Mainframe screen. Data can then be easily extracted from the full screen text.
- If Clipboard is not available in the Mainframe interface Character Matching or OCR can be used to read text fields.
- Data can be keyed into the Mainframe by clicking where you want to enter text and sending keystrokes. If click events are not recognised by the emulator they tabbing keystrokes will be required.

- ★ If your Mainframe emulator is installed locally (not using a thin client technology) but is not Hllapi compliant, the best option than using Surface Automation is to consider purchasing licenses for a Hllapi compliant emulator to enable easier Robotic Process Automation.
- ★ As with any Surface Automation interface you build in Blue Prism it is recommend, if possible, that you maximise Windows you want to interface with. This will make the behaviour of the application more predictable and reliable. Once you launch or attach your mainframe emulator, the next thing you should do is maximise it.

Recommended Training

It is recommend that the following training is taken prior to using the techniques outlined in this guide:

- Surface Automation Training

Training guide collateral

This training guide is distributed alongside the following:

- Mainframe Surface Automation Interface. This is a base object that you can call from your own object interface to interact with a terminal emulator. This object will need configuring for your own mainframe and screen resolution, and may need additional bespoke actions creating for your mainframe.
- Emulator Example Object – Basic Actions. This is an example emulator object to show how the Mainframe Surface Automation Interface object could be used for your own interfaces. It contains an example of an Attach action and a Log In action.

info@blueprism.com • +44 (0)870 879 3000 • Centrix House, Crow Lane East, Newton-le-Willows, WA12 9UY

Reading Mainframe Text using Clipboard

Copy All Text

Most terminal emulators have the functionality to select and copy all text using keyboard or menu shortcuts.

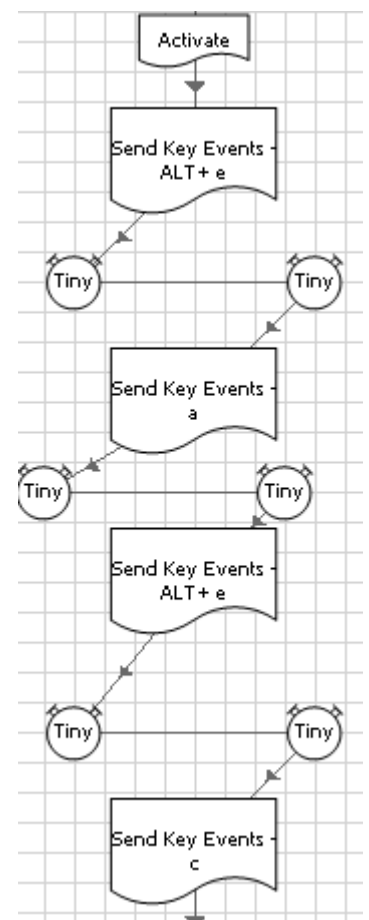
If you can copy all the text to the Windows Clipboard, you can then extract the exact data fields you want from within the full screen text. For example, the value you want might always be in row 10, column 20 and be up to 15 characters wide.

The action 'IBM PComm - Read All Screen Text (Copy All)' in the provided Mainframe Surface Automation Interface shows how text is copied to clipboard for the IBM PComm emulator. For this action the following steps needed to be taken:

- Activate the emulator window
- Send Alt+e keystrokes to open the Edit menu
- Send an 'a' keystroke to select the Select All menu option
- Send Alt+e keystrokes to open the Edit menu
- Send a 'c' keystroke to select the Copy Text menu option
- Use a calculation stage with the GetClipboard() function to retrieve the clipboard.

The Mainframe SA interface supplied with this guide only contains logic for copying text from the IBM PComm and PowerTerm emulator applications.

If you have a different mainframe emulator you will need to add a similar action to the object based upon the copy all text keystrokes that work for your emulator.



Getting data from mainframe clipboard

Once you have read all the text from the mainframe session into Blue Prism, you can extract the specific data you need to read from the full screen text.

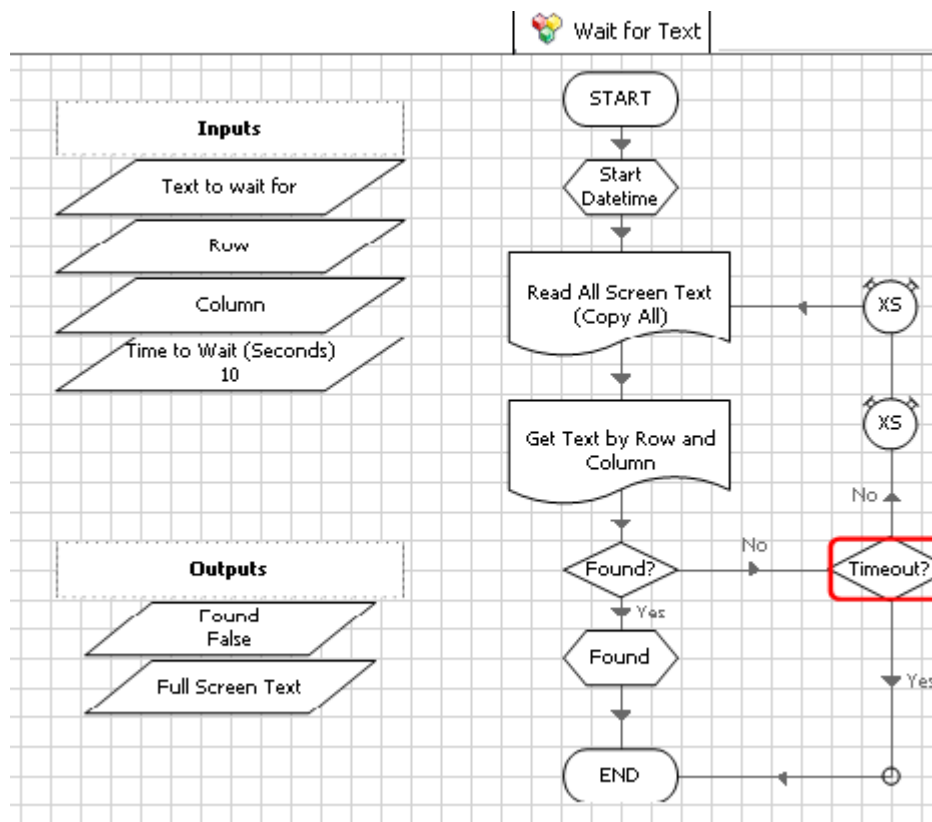
To assist with this data extraction the Mainframe Surface Automation Interface object includes an action called 'Get Text by Row and Column'. For each field you want to extract you can simply use this action, providing it with the column, row, and length of the field you want to extract from the already captured full screen text.

Getting data from mainframe clipboard

Additional 'Wait for Text' and 'Wait for Text to NOT exist' actions are provided that use the logic of copying all the text from the screen and extracting text to wait for a specific text condition on the screen.

These waiting for text actions allow you to include Best Practice wait stage logic in your interface.

This screenshot shows the Wait for Text action. It simply copies all the text from the mainframe screen and then looks for some text, such as a screen title, to exist at a specific location:



Character matching and OCR

If clipboard is not available to copy all the screen text from your mainframe emulator, character matching or OCR technology will need to be employed instead.

Using Read Text with OCR

The 'Read Text with OCR' feature in Blue Prism usually works very well with the simple and clear fonts used within a Mainframe Emulator application.

It is recommend that the following training is taken prior to attempting this technique:

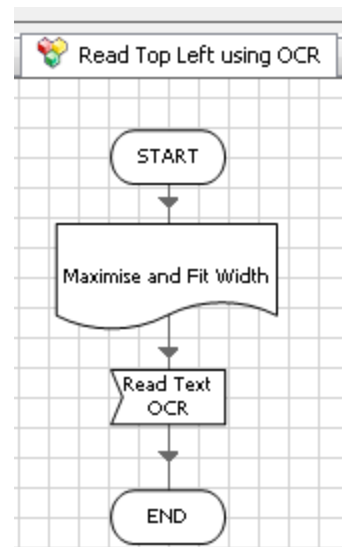
- Guide to Reading Text with OCR

For this technique you need to use Region Editor in Application Modeller to teach Blue Prism the regions of the mainframe screen from which you want to read your text.

Your interface will typically require the following steps:

- Launches or attaches to your mainframe application
- Waits for an image of a screen title or image to exist
- Uses the Read Text with OCR functionality to read text from the screen regions you have identified in Region Editor.

★ Note: the OCR functionality within Blue Prism works best when used upon smaller field regions rather than on larger screen areas.



Using Character Matching

If both Clipboard and the 'Read Text with OCR' methods cannot be used, the Character Matching features within Blue Prism may be used instead. The Identify Font feature in Region Editor may be used to attempt to identify your mainframe font, but it is more common that Mainframe Fonts need to be manually created in System Manager. A basic description is given of how to create a new font in section 5 of the Surface Automation training course.

★ Note: Some mainframe emulators add shading within the displayed text characters. Such shading would be incompatible with Character Matching.

Once you have identified or created your Mainframe Font, you can use Recognise Text in read stages to read text from the regions you have identified in Region Editor.

Writing text to Mainframe Emulators

There are two methods of navigating and entering text into the screens:

1. Clicking in the mainframe cell you want to enter text into, and then sending the keystrokes to enter text

Or

2. Moving around the screen using tabs and enter keystrokes to get to the cell you want to enter text into, then sending the keystrokes to enter text.

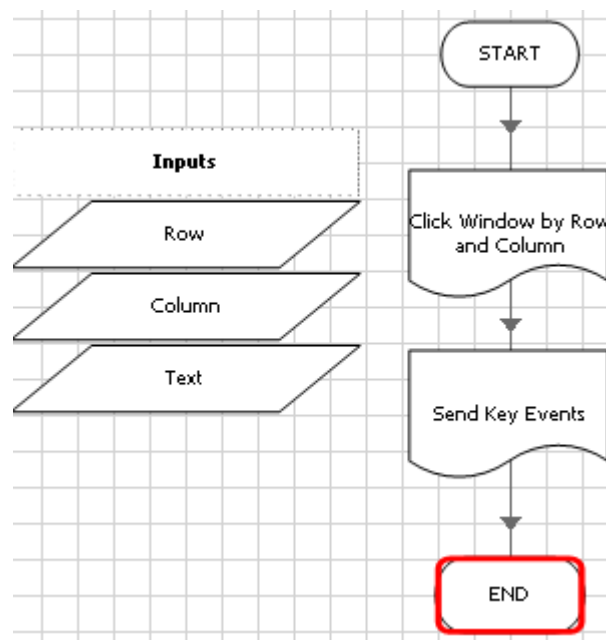
The first method, clicking in the cell first, is a quicker and more reliable method of interfacing with the mainframe emulator and should therefore be attempted first.

Clicking into the Mainframe Cell to enter text

The quickest method of entering text into a mainframe emulator using surface automation is to click in the mainframe cell where you want to enter text, and then using Send Keys or Send Key Events to enter the required text.

This functionality has been included in the 'Set Text by Row and Column' action in the provided Mainframe Surface Automation Interface object. However, some configuration needs to be done for this interface to work (see the Configuring Mainframe Surface Automation Interface section at the end of this guide).

The logic is simple, imagine that your mainframe screen is made up of lots of cells, usually 80x24 cells in size. Once configured, the interface simply clicks in the cell you want to start entering text in and then sends the text keystrokes you require.



- * Note: Any interface that clicks at pre-defined screen coordinates the screen resolution of the Windows operating system must be reliably fixed. For this technique of clicking in a mainframe cell all your virtual desktops must be set up with the same screen resolution.

Clicking into the Mainframe Cell to enter text

The second option for entering text into the mainframe screen is to send the correct number of tab or enter keystrokes to the screen to get to the field where data needs to be entered. Then sending the required text.

With this method of text entry take care to ensure that the tab order of the mainframe screen is always the same, and does not change depending upon different business scenarios. For example, for some account types additional fields might be displayed on a customer screen.

Configuring the Mainframe Surface Automation Interface object

The object distributed with this training guide is provided as an example, and it contains many of the interface techniques mentioned within this guide. The object is unlikely to work for you 'out of the box', it is provided 'as is' as the basis upon which might develop your own terminal emulator interface.

To configure the object to work for your terminal emulator and desktop screen resolution the following steps will need to be taken:

1. Add an action to copy all text from your mainframe emulator

The object comes already configured for use with PowerTerm and IBM PComm mainframe emulators. For any other emulator you wish to use you will need to develop a new action to copy all text from the screen.

The following will need to be done:

- Add a new action called '<Emulator Name> Read All Screen Text (Copy All)'. This action does not need to be published. This new action needs to include all the steps to copy all the screen text from your emulator.
- Amend the Choice Stage on the 'Read All Screen Text (Copy All)' action to accept the name of your emulator as an option, and route that option to call the sub-page you created.

2. Configure the keyboard locale you are using

When using Send Key Events some characters require additional keys to be pressed. For example, to send a * character on a UK or US keyboard we need to hold down the Shift key and the number 8 key together.

The logic of when to hold down the shift key has been included within the object for UK and US keyboards, look at the 'Text Conversion for UK Keyboard' as an example.

If your Blue Prism robot is to be used with another keyboard layout the following will need to be done:

- Create a new action called 'Text Conversion [Country Name] Keyboard]. Add similar logic to the US and UK keyboard actions to configure Shift key presses for the new keyboard.
- Amend the Choice Stage on the 'Text Conversion' action to accept the name of the country you have added, and route that option to call the sub-page you created.

3. Use the Configuration action to set the parameters for your emulator

The Configuration action should be called by your process first to set all the parameters required for the interface to work.

The following input parameters need to be set on the Configuration page:

- **Mainframe Emulator.** This needs to be set to the name of the Emulator you are using. For example, it needs to be set to IBM PComm if that is the emulator you are using. If you have updated the object to work with a new emulator you will need to set this input parameter to the name you have given to it.
- **Keyboard.** This needs to be set to the keyboard location you are using. It defaults to US, see above if you need to configure a new keyboard layout.
- **Screen Columns and Screen Rows.** This needs setting to the number of characters wide and high your mainframe emulator is set to use. By default this is 80 and 24.

The next four input parameters may take a bit of experimentation and testing to get correct. Sending text using the interface is a good test of if you have configured this correctly or if it requires more adjusting.

- **0,0 x coordinate** – needs to be set to the x coordinate of the top left position of the very first character cell within the mainframe emulator.
- **0,0,y coordinate** – needs to be set to the y coordinate of the top left position of the very first character cell within the mainframe emulator.
- **Cell height** – needs to be set to the height of each cell within the mainframe emulator (i.e. how high in desktop pixels are each of the 24 rows within the mainframe)
- **Cell width** – needs to be set to the width of each cell within the mainframe emulator (i.e. how wide in desktop pixels are each of the 80 columns within the mainframe)

Remember the following tips:

- ★ For application consistency, when using Surface Automation it is good to always maximise an application window (if possible) before interfacing it.
- ★ For the techniques used this guide the constancy of desktop screen resolution is important because a change in resolution may impact the 0,0 x and y position or the cell width and height.

The following image shows how to set the provided objects Configuration parameters:

