Velocity/MARK Display

Integration Guide

About this guide

This **Integration Guide** describes the ProGlove MARK Display-specific functions that are available in the Velocity demo projects. These functions can be used to integrate the possibilities of the MARK Display scanner in any telnet or browser-based application with Velocity. The initial configuration of a Velocity project is defined in the documents *Browser Profile Configuration in Velocity – QSG vx.y.pdf* and *Telnet Profile Configuration in Velocity – QSG vx.y.pdf* that are part of the Velocity/MARK Display Demo Kit. The functions that are covered in this guide are used in the Advanced Configuration section of the Velocity Console. The last part of this guide describes how to create a configuration file for the ProGlove Insight Mobile app via the ProGlove Insight web page.

Contents

Functions used in the Telnet Demo

Press Key

Definition and usage

Enters keyboard data after a set timeout. When linked to a template scope, information can be automatically entered when a screen is displayed or refreshed. Usefull for automatically navigation through menus.

Syntax

pressKey(*timeOut*)

Parameter Values

Parameter	*Description*
timeOut	Timeout in milliseconds

PG Connect

Definition and usage

Starts the ProGlove Insight Mobile service, if not running, and shows the ProGlove connect pairing barcode.

Syntax

PG_CONNECT(*keysOnConnect*)

Parameter Values

Parameter	*Description*
keysOnConnect	Optional. A (string of) character(s) that will be pressed after a connection with the scanner is made. With this, the right menu options are selected to start scanning. Example: - 1{enter} - 1{enter}{pause:200}2{enter}{pause:200}3{enter}

PG Disconnect

Definition and usage

Disconnect ProGlove Scanner.

Syntax

PG_DISCONNECT()

Parameter Values

Parameter	*Description*
none	

PG Disable Scan Button

Definition and usage

Disables the scan button of the ProGlove Scanner.

Syntax

PG_DISABLE_SCAN_BUTTON()

Parameter Values

Parameter	*Description*
none	

PG Enable Scan Button

Definition and usage

Enables the scan button of the ProGlove Scanner.

Syntax

PG_ENABLE_SCAN_BUTTON(*sHexCode*)

Parameter Values

Parameter	*Description*
sHexCode	Optional. A key can be defined to enable the scan button. The syntax is xxxx when xxxx is the hex code of the key.

PG Play Feedback

Definition and usage

Trigger a feedback on the connected MARK. This feedback is a combination of LED light and sound.

Syntax

PG_PLAY_FEEDBACK(*feedbackID*)

Parameter Values

Parameter	*Description*
feedbackID	Feedback ID: 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)

PG Play Feedback Sequence

Definition and usage

Trigger a feedback sequence on the connected MARK. The sequence can contain a minimum of one and a maximum of five sound/light notifications.

Syntax

PG_PLAY_FEEDBACK_SEQUENCE*(feedbackID1, feedbackID2, feedbackID3, feedbackID4, feedbackID5*)

Parameter Values

Parameter	*Description*
feedbackID1	Feedback ID: 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID2	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID3	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID4	Feedback ID: 0: None

	1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID5	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)

PG Scan Button Double Click

Definition and usage

Event handler that responds to a double click of the scan button.

Syntax

PG_SCAN_BUTTON_DOUBLE_CLICK(*action, command, coordinates, confirmKey*)

Parameter Values

Parameter	*Description*
action	Action to perform when a double click is detected. The default action is 'Send key(s)'.
command	If action is Send Key(s): the (string of) character(s) to enter. For instance - {enter}, (tab}{enter} to confirm selection - y or y{enter} to confirm a question by default.
coordinates	If action is Send Screen Information. The coordinated of the information on the screen to send. These should be entered as y,x,l.

	y is the row, x is the column and l is the length of the text. x and y start at 0. For instance: 2,10,4 uses the text on the screen at column 10, row 2 and length 4
confirmKey	If action is Send Screen Information. Key to confirm the data sent.

PG Set Display Screen

Definition and usage

Display information on the ProGlove display device. This function is not linked to a scope but called from the other scripts.

Syntax

PG_SET_DISPLAY_SCREEN(*templateID*, *data*, *separator*, *refreshType*)

Parameter Values

Parameter	*Description*
templateID	String that describes which template to use (PG2, PG3, PG1I, PG1E, PG1C, PG2I, PG2E, PG2C).
data	String that encodes the information you want to display. Each template field has an id, a header and a content: - The id is a positive integer, which identifies the template field. We number each field in normal reading direction, starting with 1 on the top-left - The content is the string, that will be displayed inside the field. - The header is the string, that will be displayed on the top of the field. It should give context to what is shown in this field. The data string is constructed by appending template field descriptions with a ';' character between them. Each template field description looks like this: "id;header;content" full example: "1;header1;content1;2;header2;content2"
separator	The ProGlove Connect App will split the DATA string along the separator character.

	Possible values are ; and |
refreshType	This parameter controls the display device's strategy of screen refresh. The display uses an E-Ink display, which has two modes of refreshing. Full and partial refreshes. Possible values are FULL_REFRESH, PARTIAL_REFRESH and DEFAULT

PG Display Data (On Screenupdate)

Definition and usage

If a specific text is display is displayed, information is read from the telnet screen and send it to the ProGlove Display using either the PG1A, PG2 or PG3 template. The data to use for header and data can be fixed, variable (read from the screen) or combined.

Format:

pgTemplate = "PG1A"

pgDisplayText = "1|Header1|Data1"

pgTemplate = "PG2"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"

pgTemplate = "PG3"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2|3|Header3|Data3"

Syntax

pgDisplayDataOnScreenUpdate(*searchText, searchTexXPos, searchTexYPos, showDisplayText, pgTemplate, header1Text, data1Text, header2Text, data2Text, header3Text, data3Text*)

Parameter Values

Parameter	*Description*
searchText	Text that must be displayed to identify the screen that contains the data to send.
searchTextXpos	X-position of the search text, starting from 0
searchTextYpos	Y-position of the search text, starting from 0
showDisplayText	Show the text that is on the screen at the search text coordinates. This helps to verify if the search text actually matches the text on the screen while testing your project.
pgTemplate	PG template to use for displaying the data. PG1A uses Header 1, Data1

	PG2 uses Header 1, Data1, Header 2, Data2 PG3 uses Header 1, Data1, Header 2, Data2, Header 3, Data3
header1Text	Text to use for Header 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data1Text	Text to use for Data 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
header2Text	Text to use for Header 2 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data2Text	Text to use for Data 2 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
header3Text	Text to use for Header 3 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data3Text	Text to use for Data 3

	If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '

PG Display Data (Link to scope)

Definition and usage

Read information from the telnet screen and send it to the ProGlove Display using either the PG1A, PG2 or PG3 template. The data to use for header and data can be fixed, variable (read from the screen) or combined.

Format:

pgTemplate = "PG1A"

pgDisplayText = "1|Header1|Data1"

pgTemplate = "PG2"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"

pgTemplate = "PG3"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2|3|Header3|Data3"

Syntax

pgDisplayDataOnLinkToScope (*pgTemplate, header1Text, data1Text, header2Text, data2Text, header3Text, data3Text*)

Parameter Values

Parameter	*Description*
pgTemplate	PG template to use for displaying the data. PG1A uses Header 1, Data1 PG2 uses Header 1, Data1, Header 2, Data2 PG3 uses Header 1, Data1, Header 2, Data2, Header 3, Data3
header1Text	Text to use for Header 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance:

	'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data1Text	Text to use for Data 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
header2Text	Text to use for Header 2 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data2Text	Text to use for Data 2 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
header3Text	Text to use for Header 3 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data3Text	Text to use for Data 3 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '

PG Info, Error, Confirm Display Data (On Screenupdate)

Definition and usage

If a specific text is display is displayed, an Information, Error or Confirmation message with either 1 or 2 data fields is sent. The data to use for header and data can be fixed, variable (read from the screen) or combined.

Format:

pgTemplate = "PG1I", "PG1E", "PG1C"

pgDisplayText = "1|Header|Data"   (Header data is ignored)

pgTemplate = "PG2I", "PG2E", "PG2C"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"   (Header data is ignored)

Syntax

pgIECDisplayDataOnScreenUpdate(*searchText, searchTexXPos, searchTexYPos, showDisplayText, messageType, nrOfDataFields, data1Text, data2Text*)

Parameter Values

Parameter	*Description*
searchText	Text that must be displayed to identify the screen that contains the data to send.
searchTextXpos	X-position of the search text, starting from 0
searchTextYpos	Y-position of the search text, starting from 0
showDisplayText	Show the text that is on the screen at the search text coordinates. This helps to verify if the search text actually matches the text on the screen while testing your project.
messageType	Type of message: Error, Information, Confirmation
nrOfDataFields	Number of data fields
data1Text	Text to use for Data 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data2Text	Text to use for Data 2

	If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '

PG Info, Error, Confirm Display Data (Link to scope)

Definition and usage

Display an Information, Error or Confirmation message with either 1 or 2 data fields is sent. The data to use for header and data can be fixed, variable (read from the screen) or combined.

Format:

pgTemplate = "PG1I", "PG1E", "PG1C"

pgDisplayText = "1|Header|Data"   (Header data is ignored)

pgTemplate = "PG2I", "PG2E", "PG2C"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"   (Header data is ignored)

Syntax

pgIECDisplayDataLinkToScope(*messageType, nrOfDataFields, data1Text, data2Text*)

Parameter Values

Parameter	*Description*
messageType	Type of message: Error, Information, Confirmation
nrOfDataFields	Number of data fields
data1Text	Text to use for Data 1 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used. For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '
data2Text	Text to use for Data 2 If pos:y,x,l is used, the text on the screen at row y, column x and length l is used.

	For instance: 'Error is: pos:2,10,20' adds the text on the screen at row 2, column 10 and length 20 to the string 'Error is: '

Functions used in the Web Demo

Create Web Scopes (Link To Session)

Definition and usage

Create scopes for every webpage and its elements. Scripts can then be linked to these scopes.

Syntax

createWebScopes()

Parameter Values

Parameter	*Description*
None	

Show Scopes

Definition and usage

Display the names of the available scopes for the current web page. This script is used during testing of a web application and shows the scopes that are created by the Create Web Scopes (Link To Session) script.

Syntax

showScopes()

Parameter Values

Parameter	*Description*
None	

PG Connect

Definition and usage

Starts the ProGlove Insight Mobile service, if not running, and shows the ProGlove connect pairing barcode.

Syntax

PG_CONNECT(*keysOnConnect*)

Parameter Values

Parameter	*Description*
keysOnConnect	Optional. A (string of) character(s) that will be pressed after a connection with the scanner is made. With this, the right menu options are selected to start scanning.

PG Disconnect

Definition and usage

Disconnect ProGlove Scanner.

Syntax

PG_DISCONNECT()

Parameter Values

Parameter	*Description*
none	

PG Disable Scan Button

Definition and usage

Disables the scan button of the ProGlove Scanner.

Syntax

PG_DISABLE_SCAN_BUTTON()

Parameter Values

Parameter	*Description*
none	

PG Enable Scan Button

Definition and usage

Enables the scan button of the ProGlove Scanner.

Syntax

PG_ENABLE_SCAN_BUTTON(*sHexCode*)

Parameter Values

Parameter	*Description*
sHexCode	Optional. A key can be defined to enable the scan button. The syntax is xxxx when xxxx is the hex code of the key.

PG Play Feedback

Definition and usage

Trigger a feedback on the connected MARK. This feedback is a combination of LED light and sound.

Syntax

PG_PLAY_FEEDBACK(*feedbackID*)

Parameter Values

Parameter	*Description*
feedbackID	Feedback ID: 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)

PG Play Feedback Sequence

Definition and usage

Trigger a feedback sequence on the connected MARK. The sequence can contain a minimum of one and a maximum of five sound/light notifications.

Syntax

PG_PLAY_FEEDBACK_SEQUENCE*(feedbackID1, feedbackID2, feedbackID3, feedbackID4, feedbackID5*)

Parameter Values

Parameter	*Description*
feedbackID1	Feedback ID: 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID2	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID3	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)

feedbackID4	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)
feedbackID5	Feedback ID: 0: None 1: ID 1 - Positive (ACK) 2: ID 2 - Negative (NACK) 3: ID 3 - Special1 (Yellow) 4: ID 4 - Special2 (Purple) 5: ID 5 - Special3 (Cyan)

PG Set Display Screen

Definition and usage

Display information on the ProGlove display device. This function is not linked to a scope but called from the other scripts.

Syntax

PG_SET_DISPLAY_SCREEN(*templateID*, *data*, *separator*, *refreshType*)

Parameter Values

Parameter	*Description*
templateID	String that describes which template to use (PG2, PG3, PG1I, PG1E, PG1C, PG2I, PG2E, PG2C).
data	String that encodes the information you want to display. Each template field has an id, a header and a content:

	- The id is a positive integer, which identifies the template field. We number each field in normal reading direction, starting with 1 on the top-left - The content is the string, that will be displayed inside the field. - The header is the string, that will be displayed on the top of the field. It should give context to what is shown in this field. The data string is constructed by appending template field descriptions with a ';' character between them. Each template field description looks like this: "id;header;content" full example: "1;header1;content1;2;header2;content2"
separator	The ProGlove Connect App will split the DATA string along the separator character. Possible values are ; and |
refreshType	This parameter controls the display device's strategy of screen refresh. The display uses an E-Ink display, which has two modes of refreshing. Full and partial refreshes. Possible values are FULL_REFRESH, PARTIAL_REFRESH and DEFAULT

PGx Display Data (Web)

Definition and usage

Information is read from the web page screen and send it to the ProGlove Display using the PG1A, PG2 or PG3 template. All fields can contain the data of an element. if id:xxxx or name:xxxx is used in the field, the content of the element with that id or name is used.

Format:

pgTemplate = "PG1A"

pgDisplayText = "1|Header1|Data1"

pgTemplate = "PG2"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"

pgTemplate = "PG3"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2|3|Header3|Data3"

Syntax

PGxDisplayDataWeb(*templateID*, *header1Text*, *data1Text*, *header2Text*, *data2Text*, *header3Text*, *data3Text*)

Parameter Values

Parameter	*Description*
templateID	PG template to use for displaying the data. PG1A uses Header 1, Data1 PG2 uses Header 1, Data1, Header 2, Data2 PG3 uses Header 1, Data1, Header 2, Data2, Header 3, Data3
header1Text	Text to use for Header 1 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
data2Text	Text to use for Data 2 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
header2Text	Text to use for Header 2 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
data2Text	Text to use for Data 2 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance:

	'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
header3Text	Text to use for Header 3 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
data3Text	Text to use for Data 3 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '

PG1IEC-PG2IEC Display Data (Web)

Definition and usage

An Information, Error or Confirmation message with either 1 or 2 data fields is sent. All fields can contain the data of an element. if id:xxxx or name:xxxx is used in the field, the content of the element with that id or name is used.

Format:

pgTemplate = "PG1I", "PG1E", "PG1C"

pgDisplayText = "1|Header|Data"   (Header data is ignored)

pgTemplate = "PG2I", "PG2E", "PG2C"

pgDisplayText = "1|Header1|Data1|2|Header2|Data2"   (Header data is ignored)

Syntax

pgIECDisplayDataWeb(*messageType*, *nrOfDataFields* , *data1Text*, *data2Text*)

Parameter Values

Parameter	*Description*
messageType	Type of message: Error, Information, Confirmation
nrOfDataFields	Number of data fields
data1Text	Text to use for Data 1 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '
data2Text	Text to use for Data 2 if id:xxxx or name:xxxx is used, the content of the element with that id or name is used. For istance: 'Error is: id:errorTxt' adds the content of the element with id errorTxt to the string 'Error is: '

PG Scan Button Double Click

Definition and usage

Event handler that responds to a double click of the scan button.

Syntax

PG_SCAN_BUTTON_DOUBLE_CLICK(*action, command*)

Parameter Values

Parameter	*Description*
action	Action to perform when a double click is detected. The default action is ‘Send key(s)’.
command	If action is Send Key(s): the (string of) character(s) to enter. For instance - {enter}, (tab}{enter} to confirm selection - y or y{enter} to confirm a question by default.

	Note: In this web demo, when the button is double-clicked, the pick quantity is confirmed. This is done by copying the value in the text field with ID ‘qtyOnQty’ in the text field with ID ‘quantity’ and then confirming that value by emulating an enter key press.

Create Element Scope

Definition and usage

Create a scope with the name of an element when this element is changed. For instance, when the text in the element changes. to show error messages. Other scripts can then be linked to this scope and will be executed when the text changes.

Syntax

createMessageElementScope(*elementToObserve*)

Parameter Values

Parameter	*Description*
elementToObserve	name or ID of the element to observe.

Show Keyboard

Definition and usage

Show or hide the on-screen keyboard.

Syntax

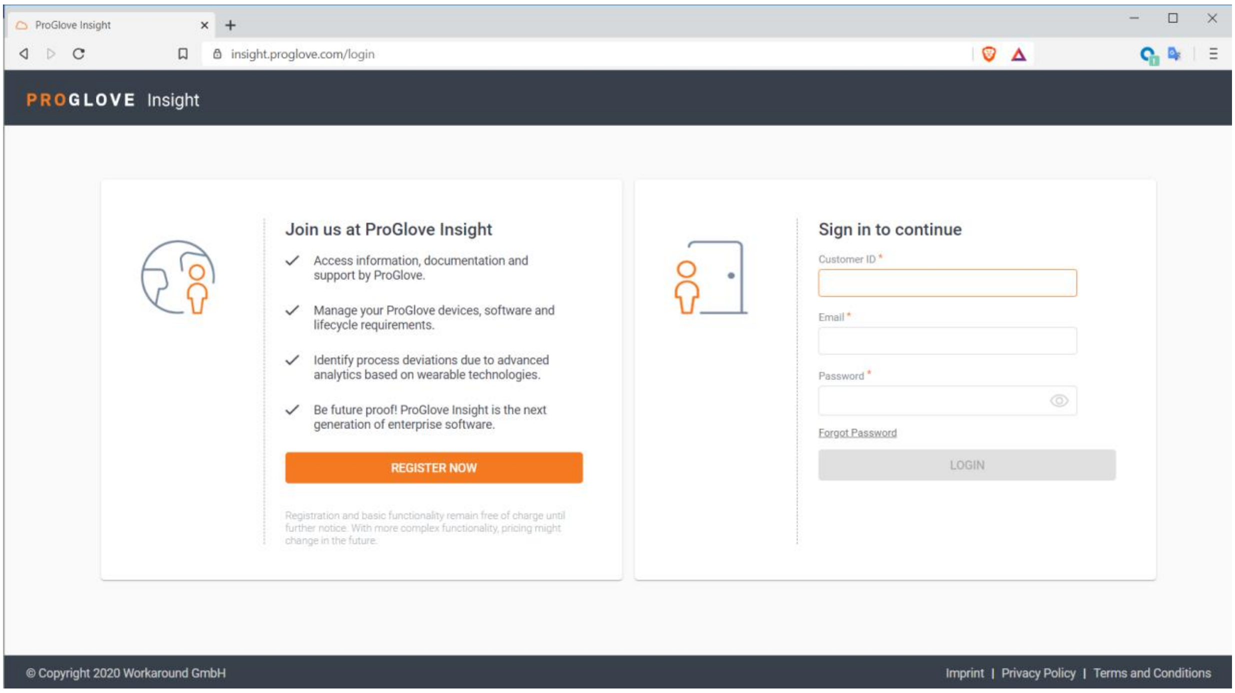
ShowKeyboard(*bShow, keyboardName, timeOut*)

Parameter Values

Parameter	*Description*
bShow	Define the visual state of the Velocity keyboard
keyboardName	Optional. Name of the keyboard to show
timeOut	Number of milliseconds to wait before the keyboard is shown

ProGlove Insight – Scanner Profile

The demo kit contains a configuration file for the Insight Mobile app that contains the necessary scanner settings. You can change these settings, or create a new configuration file, via the website http://insight.proglove.com/



Join us at ProGlove Insight

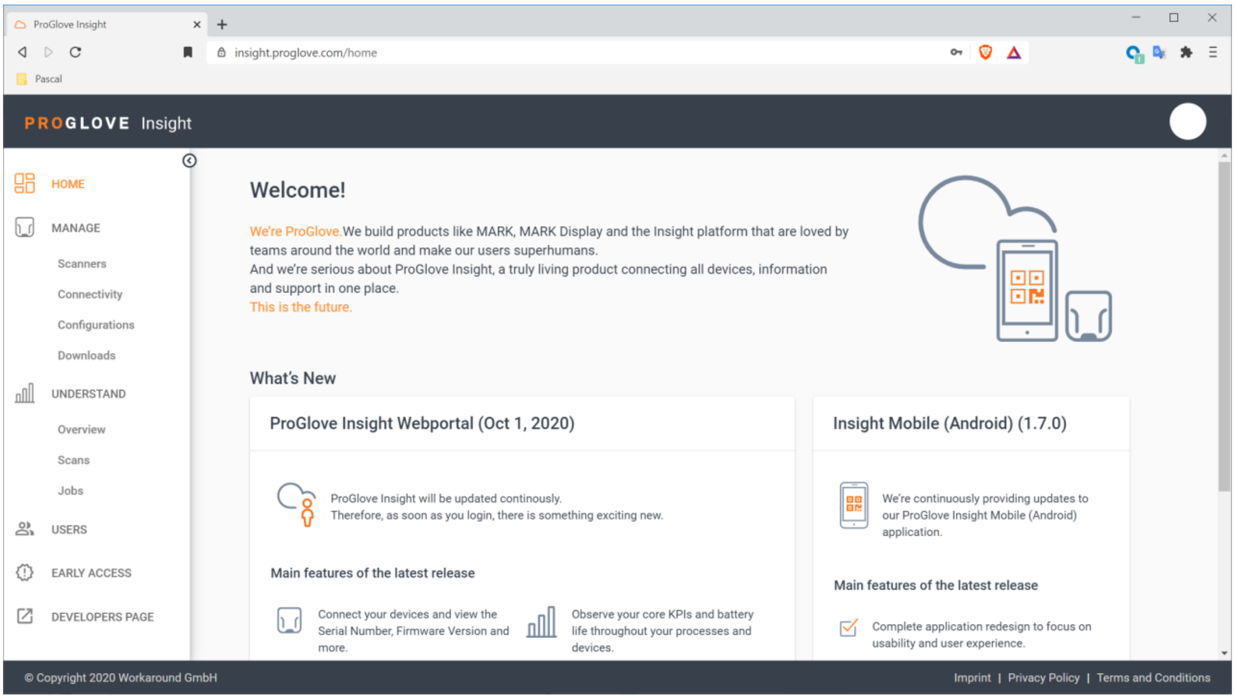
- ✓ Access information, documentation and support by ProGlove.
- ✓ Manage your ProGlove devices, software and lifecycle requirements.
- ✓ Identify process deviations due to advanced analytics based on wearable technologies.
- ✓ Be future proof! ProGlove Insight is the next generation of enterprise software.

Registration and basic functionality remain free of charge until further notice. With more complex functionality, pricing might change in the future.

Sign in to continue

© Copyright 2020 Workaround GmbH

Register first to create an account. ProGlove will send an Email with the required login information. When logged in, you can download the latest version of the Insight Mobile app, create scanner configuration files, and have access to the KPI data that can be stored on-line:



Welcome!

We're ProGlove. We build products like MARK, MARK Display and the Insight platform that are loved by teams around the world and make our users superhumans.
And we're serious about ProGlove Insight, a truly living product connecting all devices, information and support in one place.
This is the future.

What's New

ProGlove Insight Webportal (Oct 1, 2020)

ProGlove Insight will be updated continously.
Therefore, as soon as you login, there is something exciting new.

Main features of the latest release

Connect your devices and view the Serial Number, Firmware Version and more.

Observe your core KPIs and battery life throughout your processes and devices.

Insight Mobile (Android) (1.7.0)

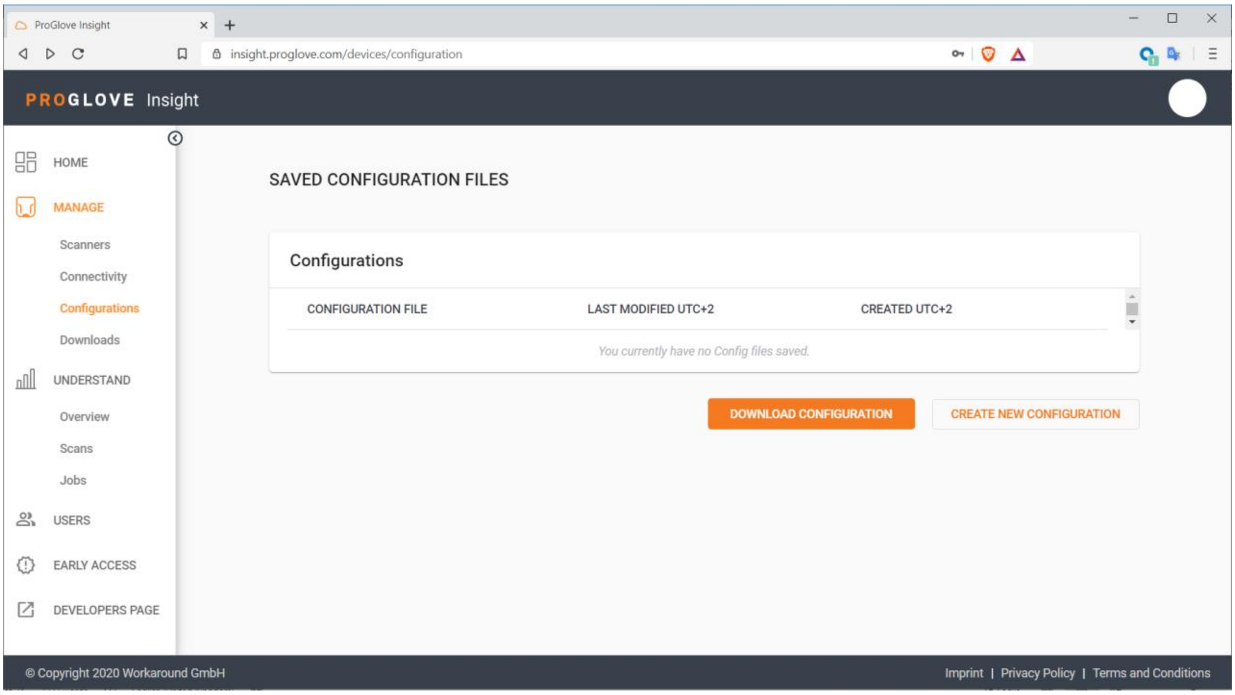
We're continuously providing updates to our ProGlove Insight Mobile (Android) application.

Main features of the latest release

Complete application redesign to focus on usability and user experience.

© Copyright 2020 Workaround GmbH

From the menu, select Configurations:



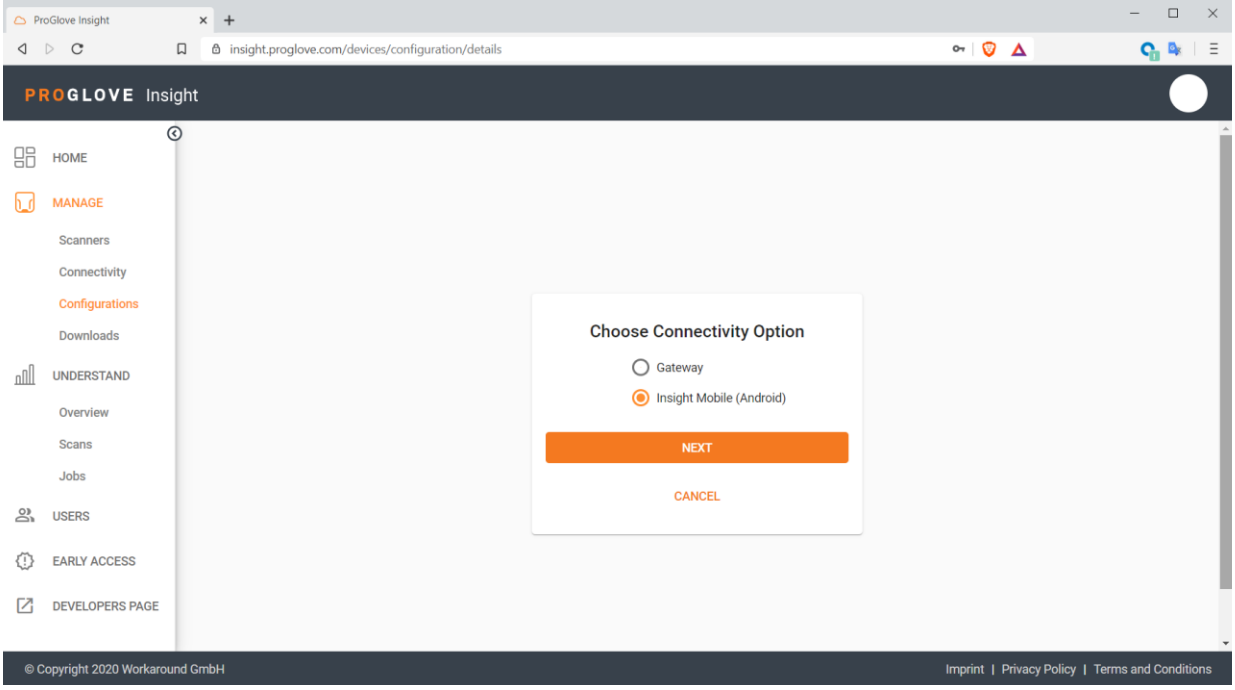
SAVED CONFIGURATION FILES

Configurations

CONFIGURATION FILE	LAST MODIFIED UTC+2	CREATED UTC+2

You currently have no Config files saved.

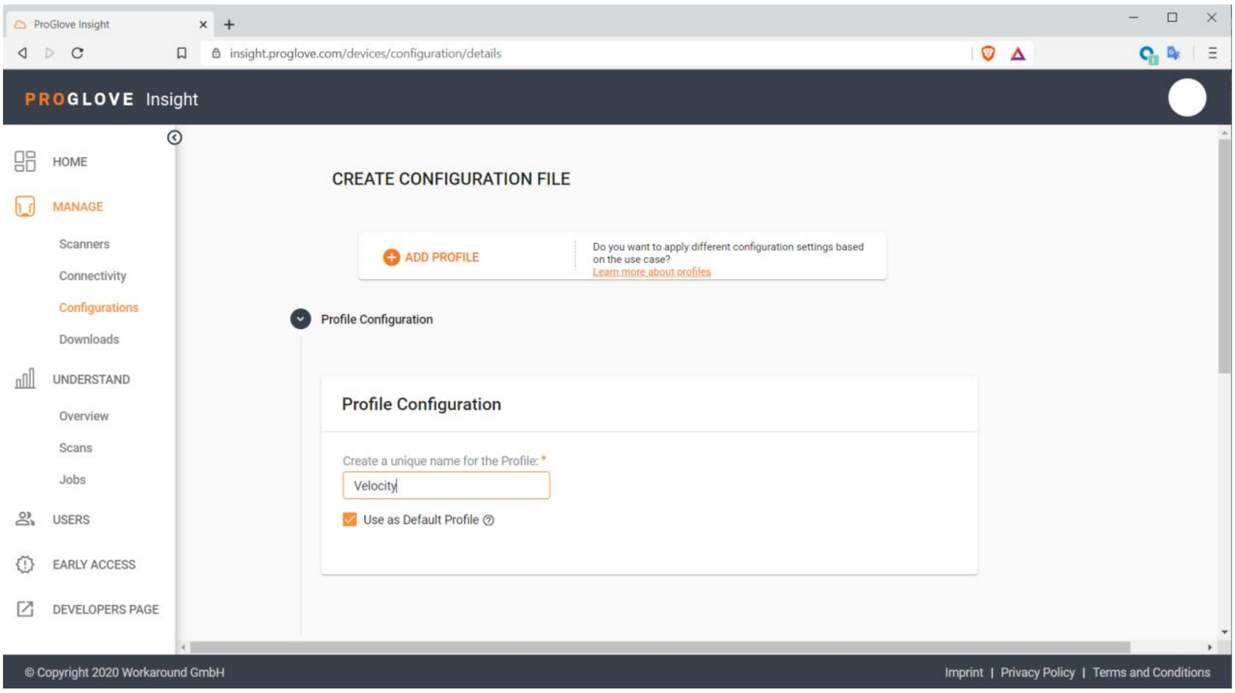
Here we can create the configuration file that is needed for the use with Velocity. Click Create New Configuration, then select Insight Mobile (Android) and click Next:



Choose Connectivity Option

○ Gateway
● Insight Mobile (Android)

Enter a name for the configuration:



CREATE CONFIGURATION FILE

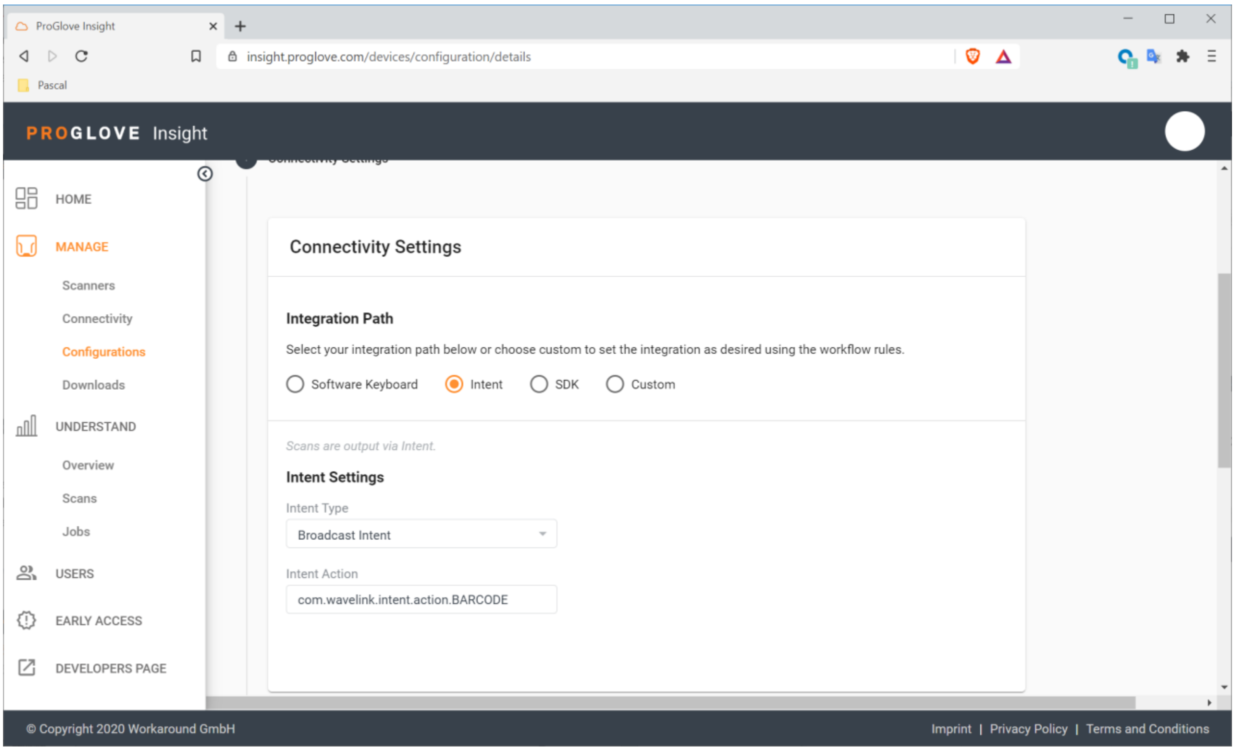
Do you want to apply different configuration settings based on the use case?
Learn more about profiles

Profile Configuration

Create a unique name for the Profile: *
Velocity

☑ Use as Default Profile

Scroll down to the Connectivity Settings section. Select Intent and enter com.wavelink.intent.action.BARCODE for the intent:



Connectivity Settings

Integration Path

Select your integration path below or choose custom to set the integration as desired using the workflow rules.

○ Software Keyboard ● Intent ○ SDK ○ Custom

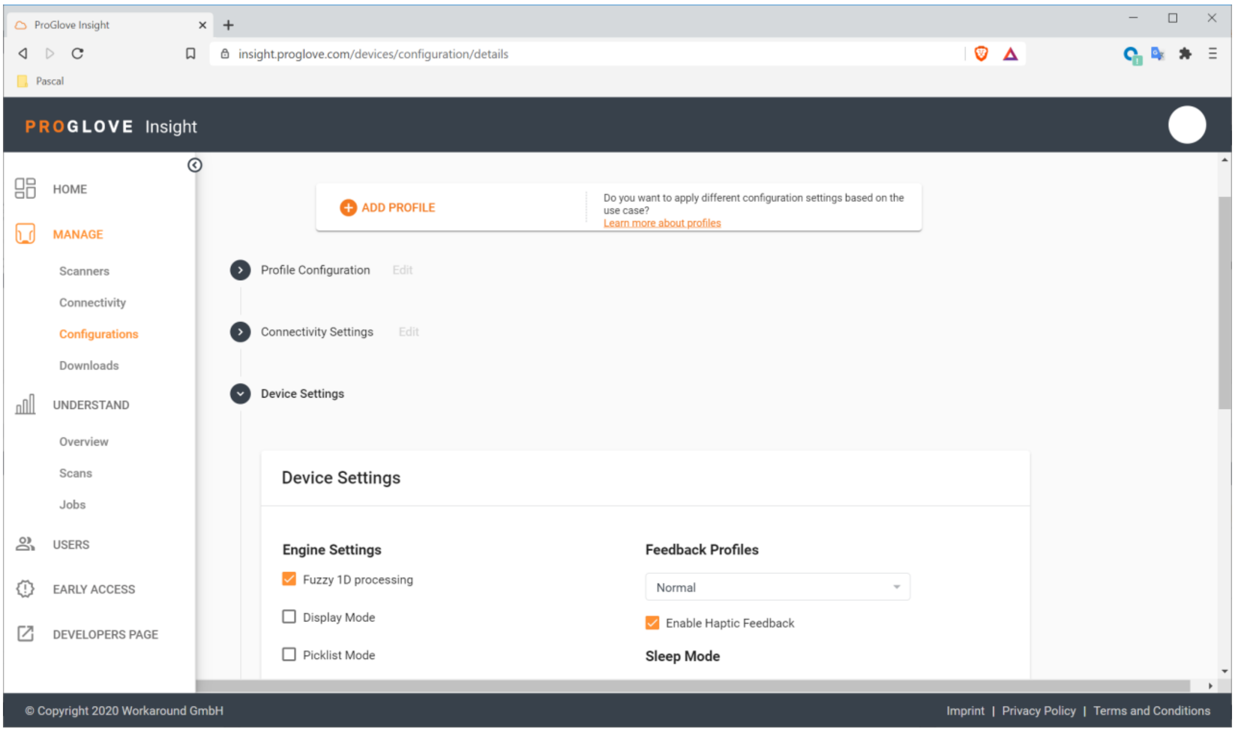
Scans are output via Intent.

Intent Settings

Intent Type
Broadcast Intent

Intent Action
com.wavelink.intent.action.BARCODE

Open Device Settings:



Do you want to apply different configuration settings based on the use case?
Learn more about profiles

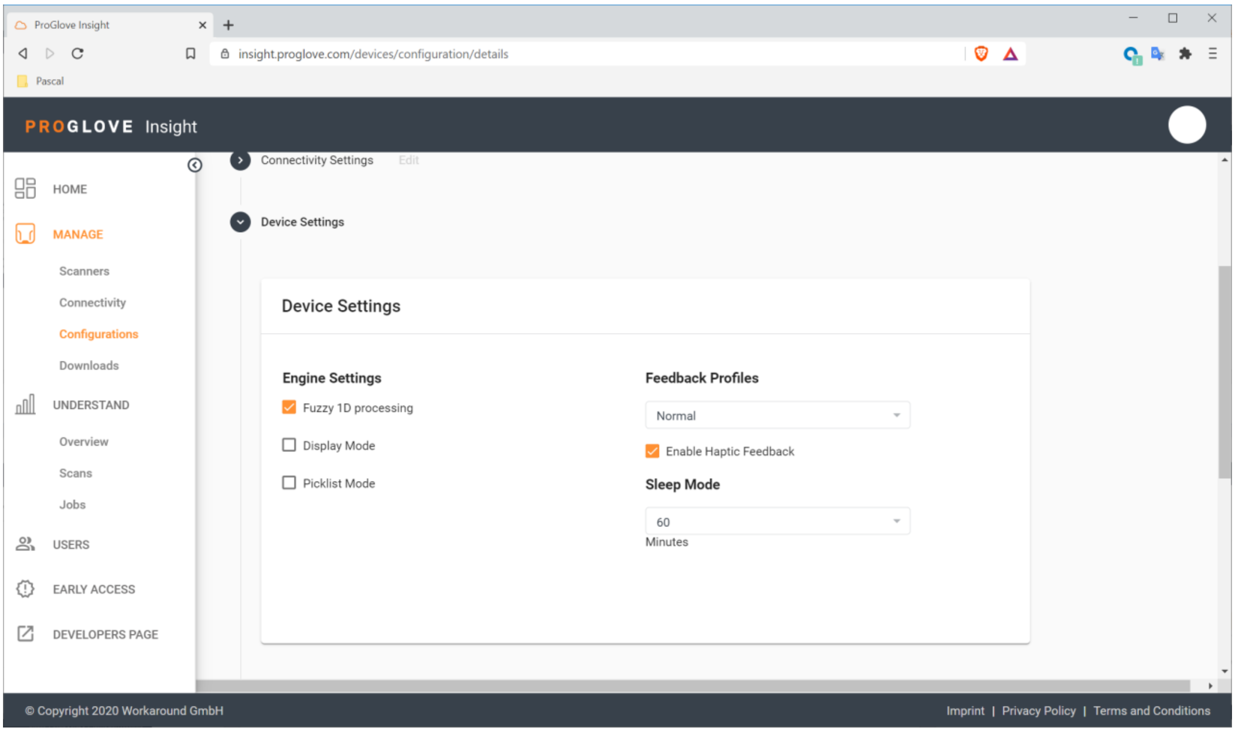
Device Settings

Engine Settings
- ☑ Fuzzy 1D processing
- ☐ Display Mode
- ☐ Picklist Mode

Feedback Profiles
Normal
- ☑ Enable Haptic Feedback

Sleep Mode

Scroll down and change the value for Sleep Mode, to for instance 60 Minutes:



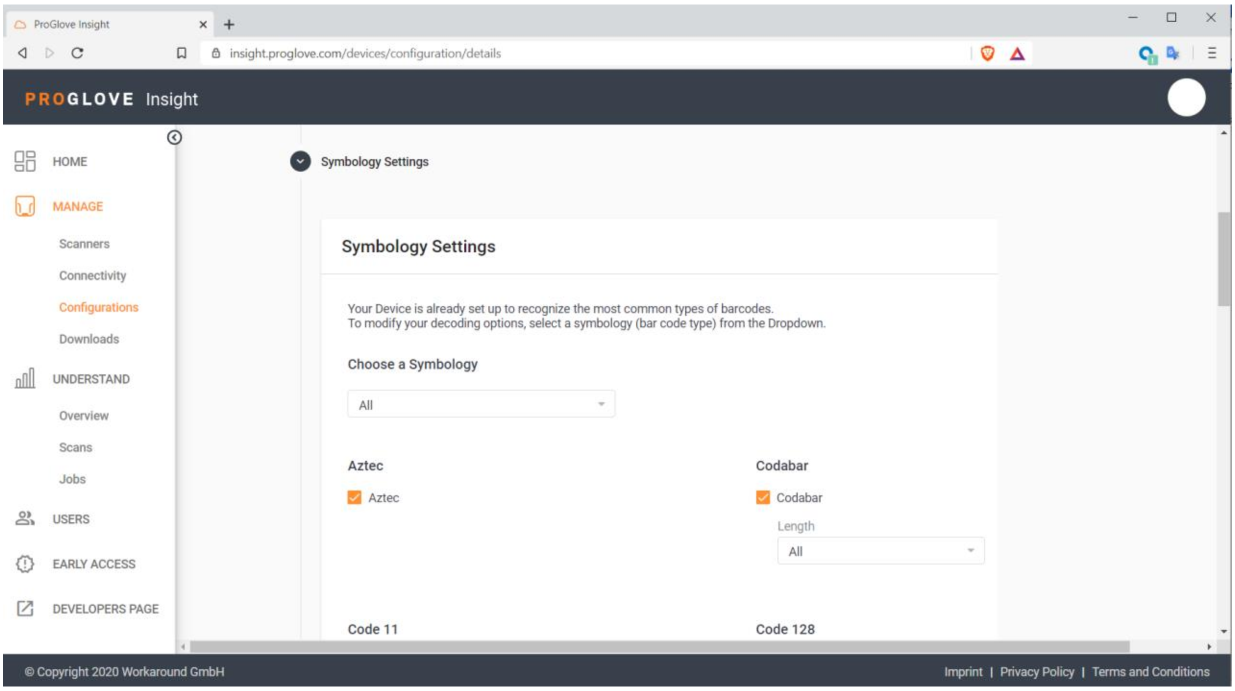
Device Settings

Engine Settings
- ☑ Fuzzy 1D processing
- ☐ Display Mode
- ☐ Picklist Mode

Feedback Profiles
Normal
- ☑ Enable Haptic Feedback

Sleep Mode
60
Minutes

Scroll down to the Symbology Settings section. Here you can enable/disable barcode types and configure barcode type specific settings:



Symbology Settings

Your Device is already set up to recognize the most common types of barcodes.
To modify your decoding options, select a symbology (bar code type) from the Dropdown.

Choose a Symbology: All

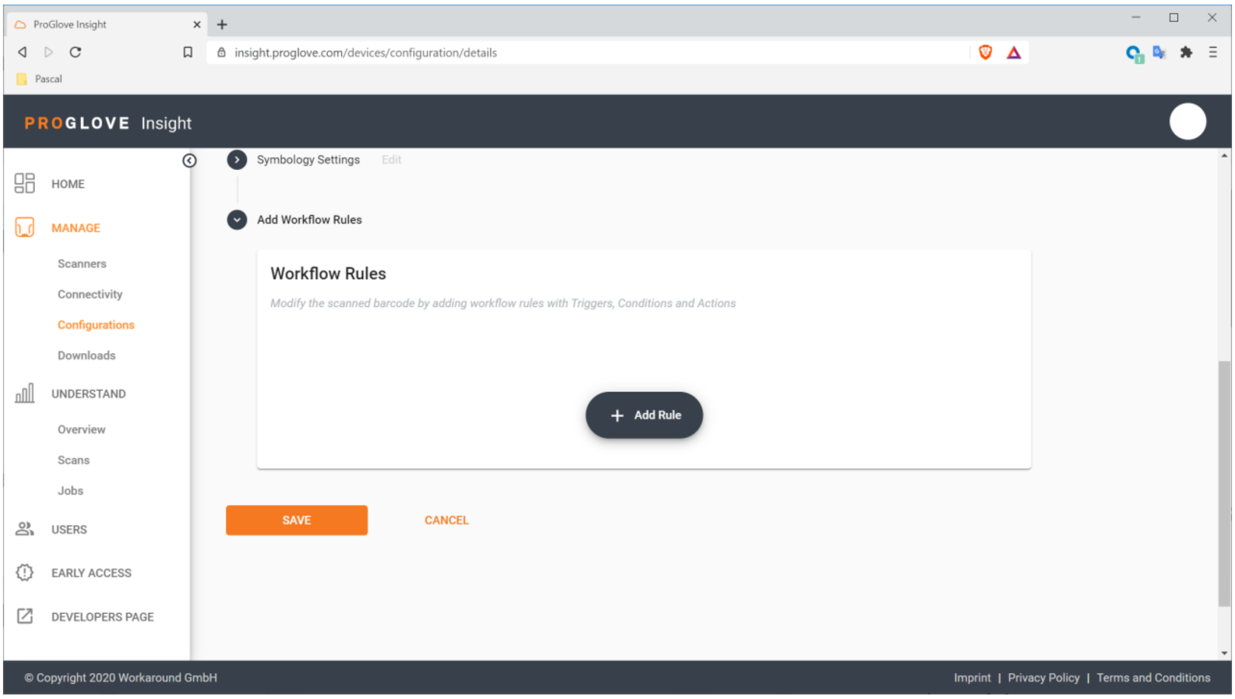
Aztec
☑ Aztec

Codabar
☑ Codabar
Length: All

Code 11

Code 128

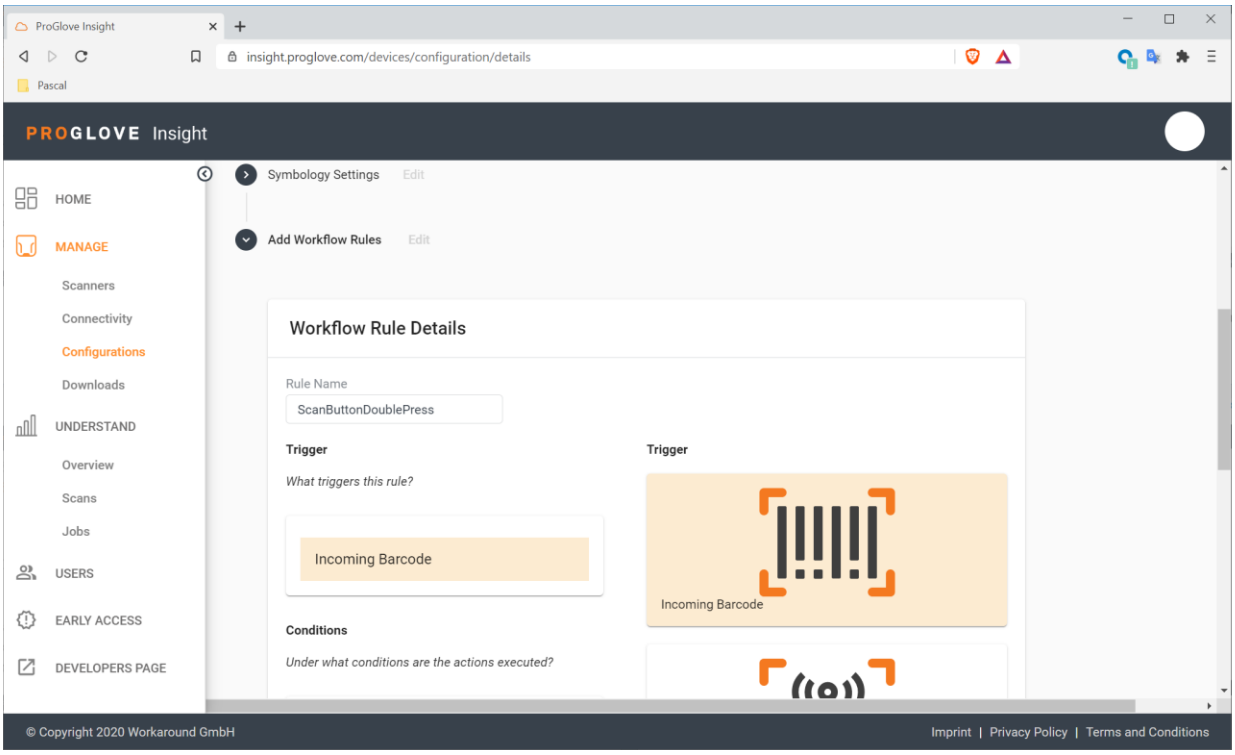
Scroll down to the Add Workflow Rules section. Here we can configure a double click on the scan button as an event in Velocity:



Workflow Rules

Modify the scanned barcode by adding workflow rules with Triggers, Conditions and Actions

Click Add Rule and add a Rule Name:



Workflow Rule Details

Rule Name: ScanButtonDoublePress

Trigger

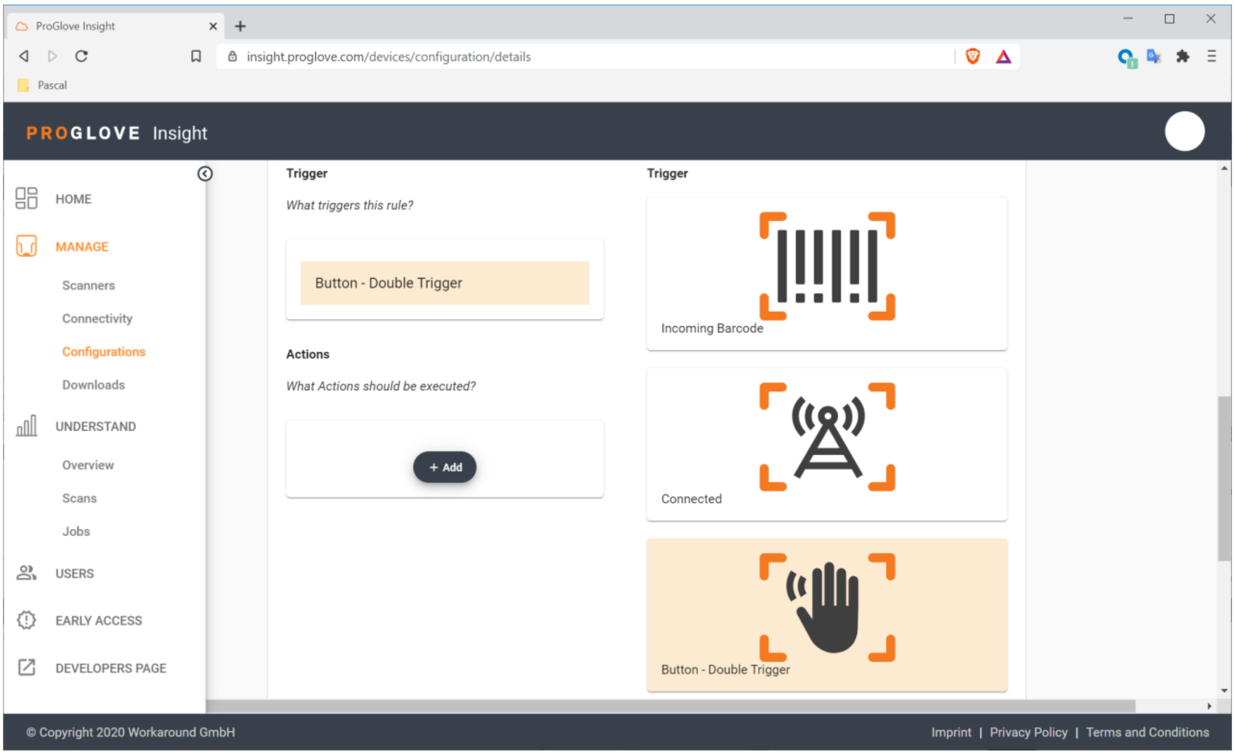
What triggers this rule?

Incoming Barcode

Conditions

Under what conditions are the actions executed?

Under Trigger, Scroll down and select Button - Double trigger:



Trigger

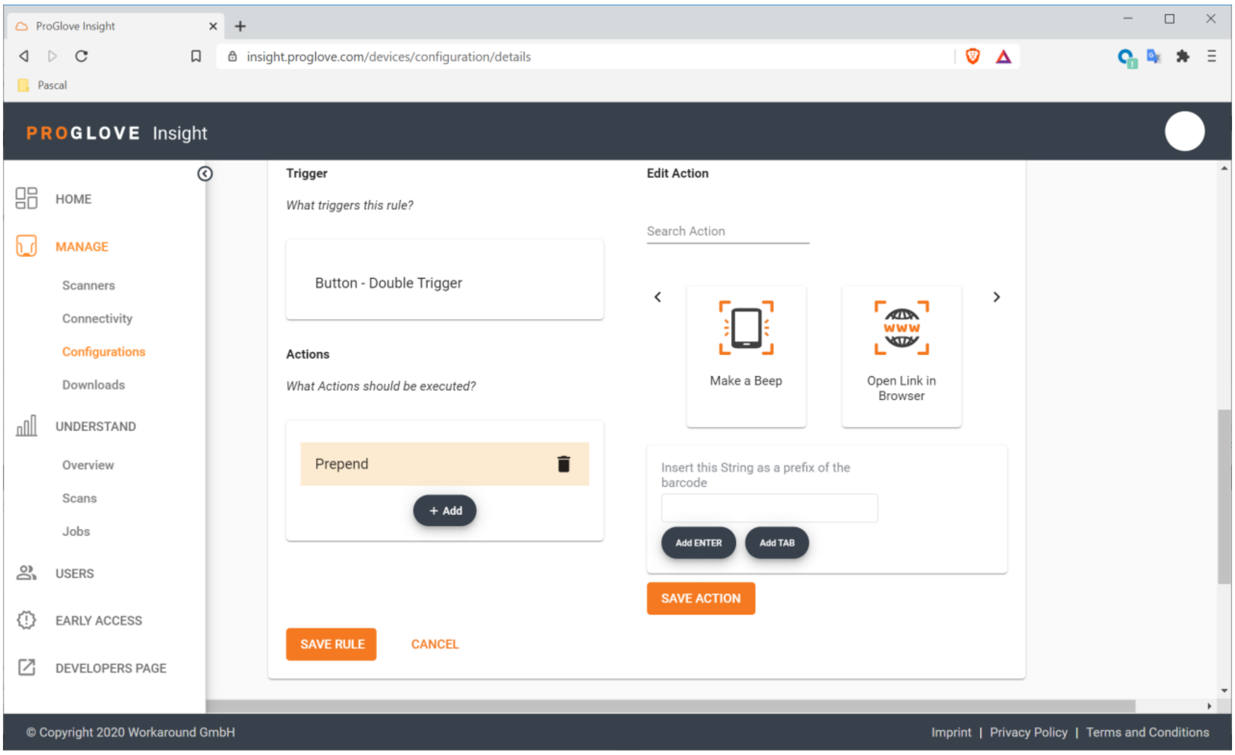
What triggers this rule?

Button - Double Trigger

Actions

What Actions should be executed?

Under Actions, click Add:



Trigger

What triggers this rule?

Button - Double Trigger

Actions

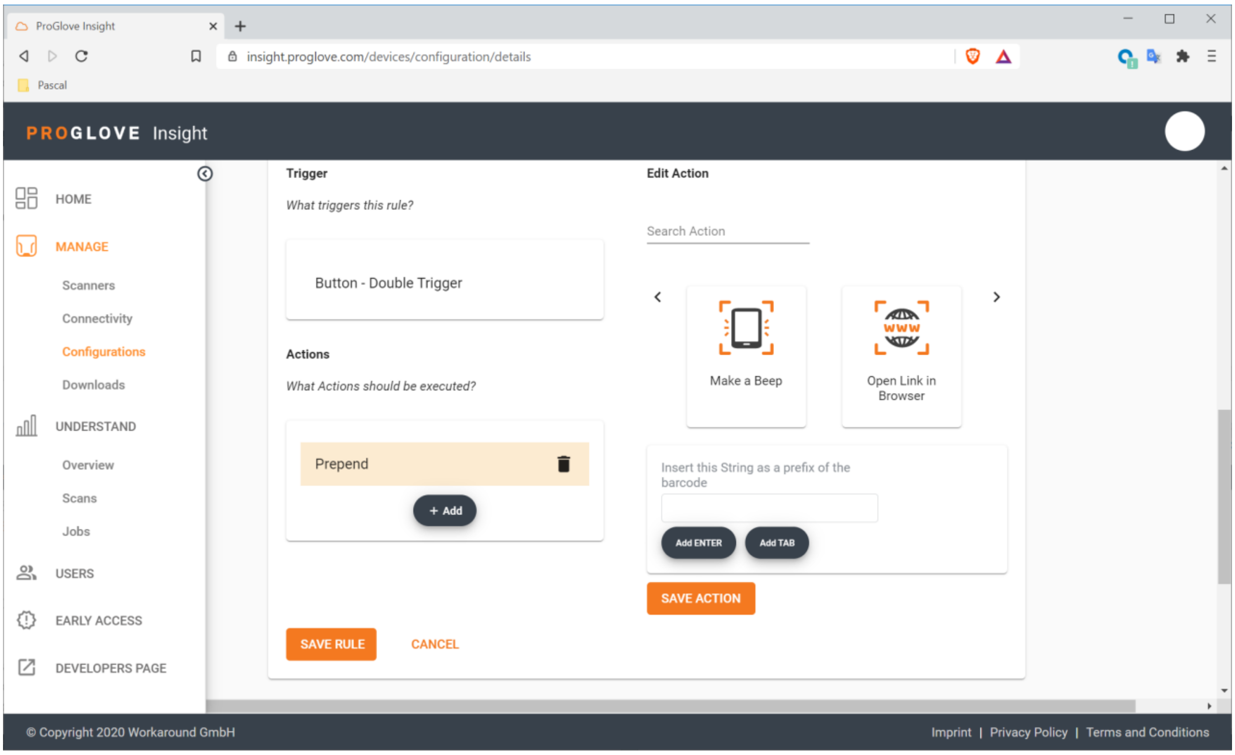
What Actions should be executed?

Prepend

Edit Action

Insert this String as a prefix of the barcode

Click the > button to scroll to the available options. Select Output via Intent:



Trigger

What triggers this rule?

Button - Double Trigger

Actions

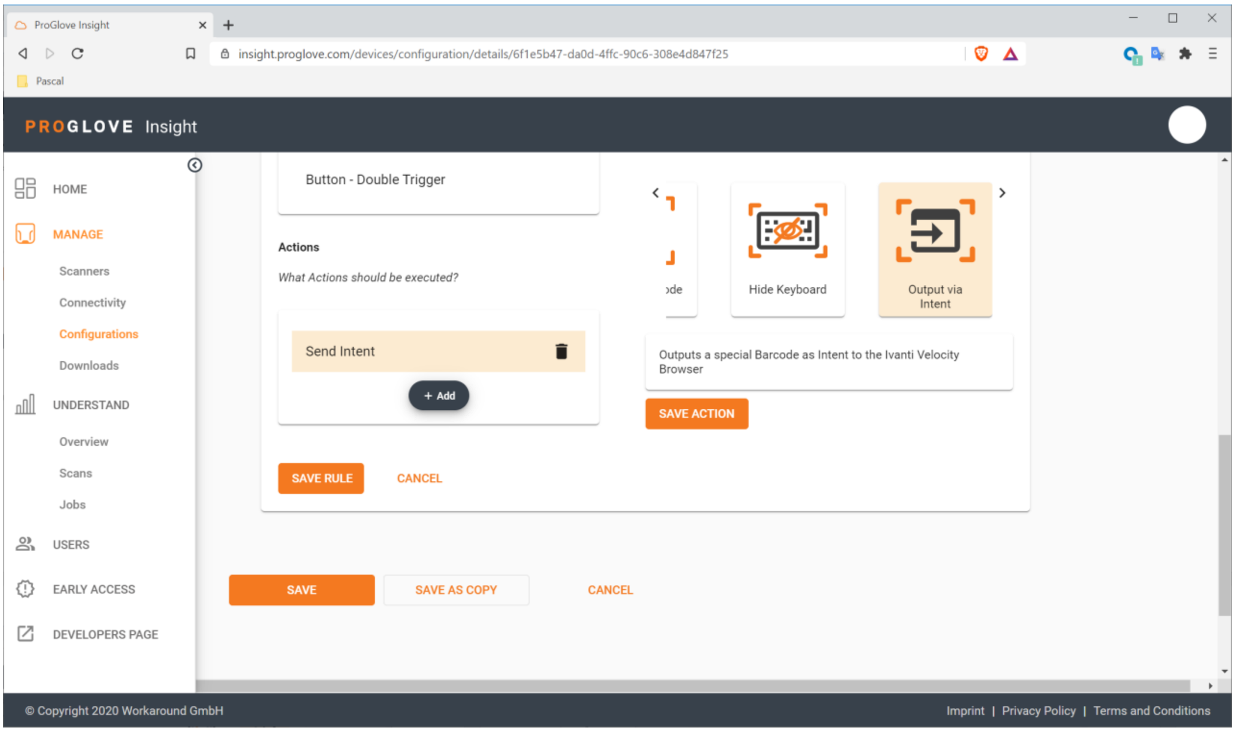
What Actions should be executed?

Prepend

Edit Action

Insert this String as a prefix of the barcode

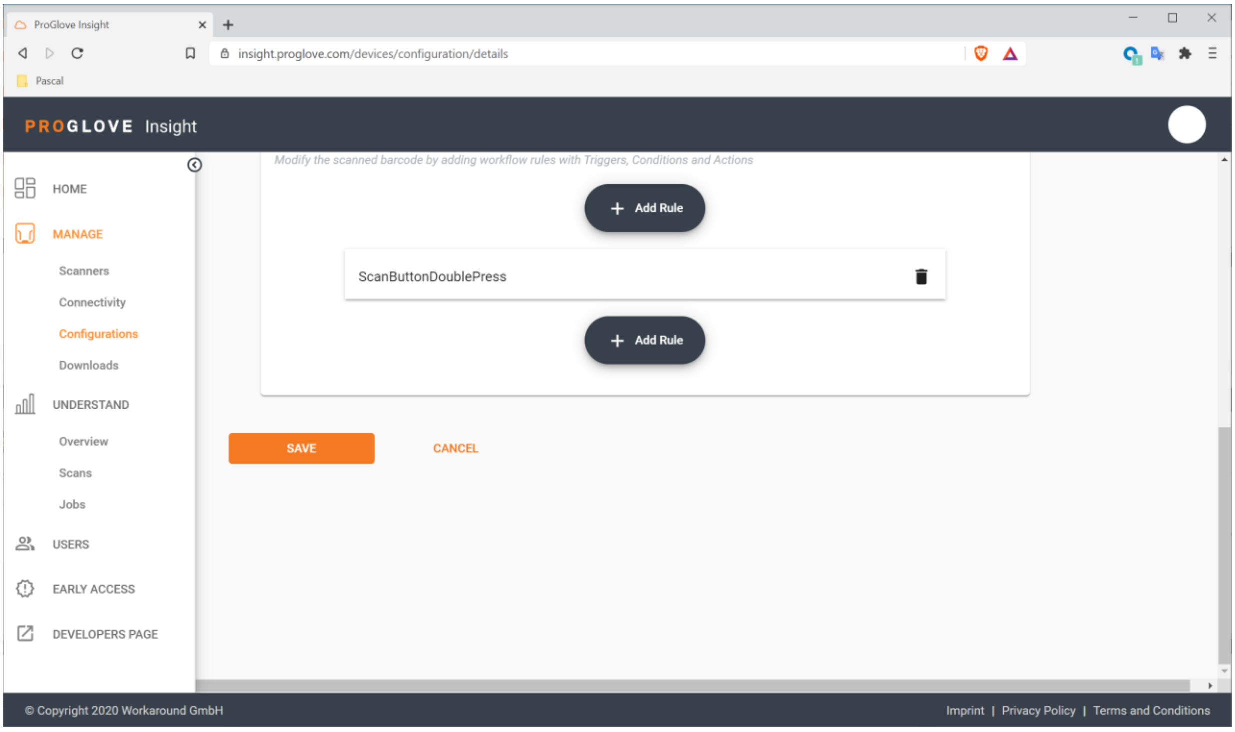
Click Save Action:



What Actions should be executed?

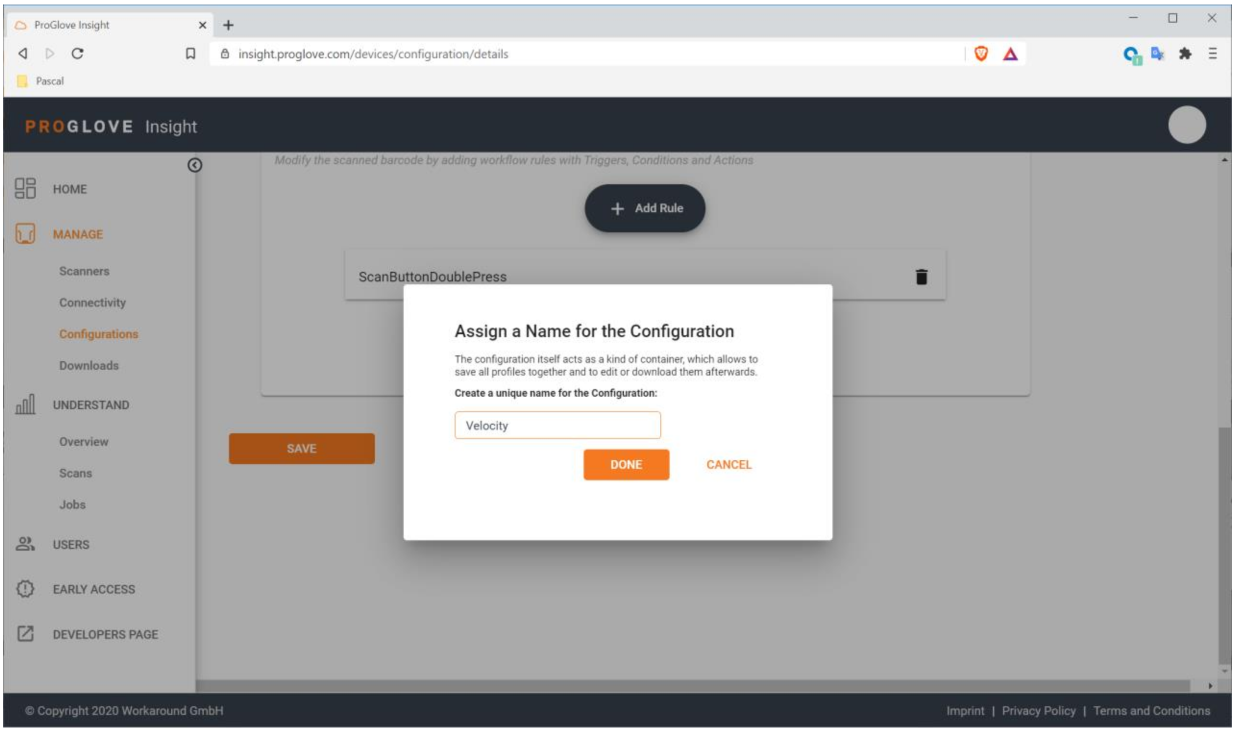
Outputs a special Barcode as Intent to the Ivanti Velocity Browser

Click Save Rule.



Modify the scanned barcode by adding workflow rules with Triggers, Conditions and Actions

Click Save. Enter a name for the configuration and click Done:



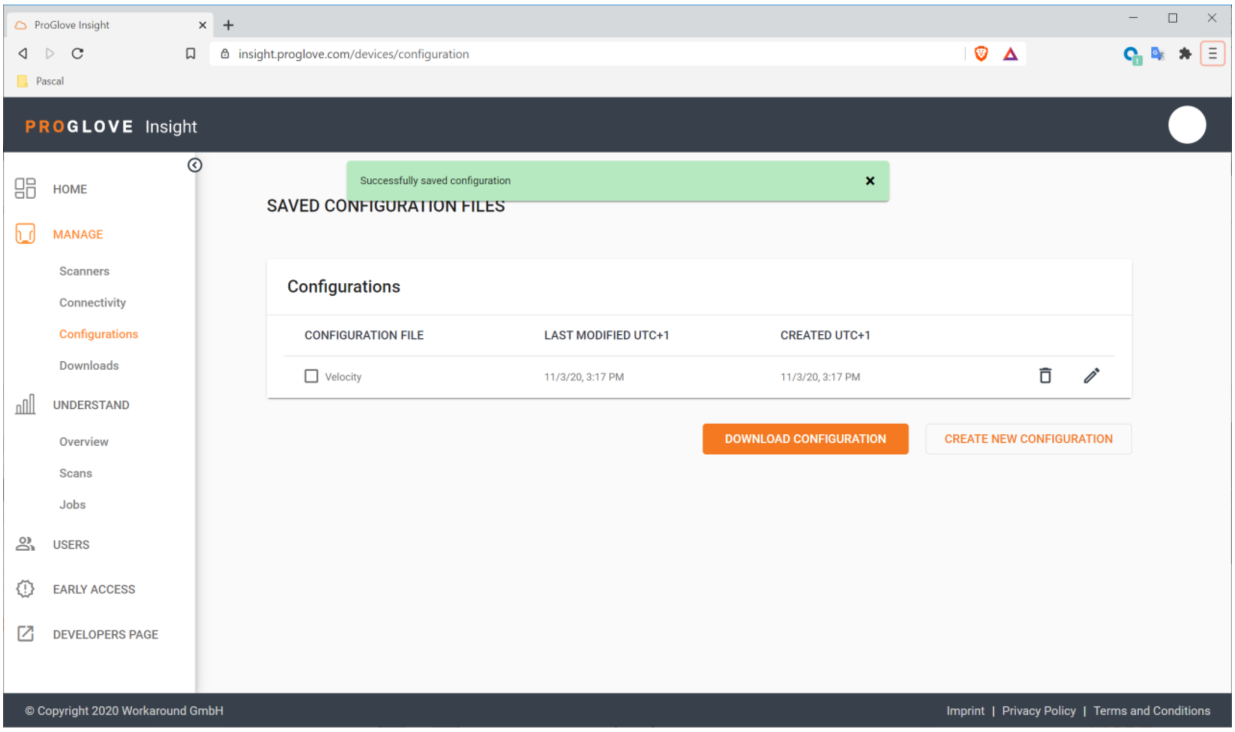
Modify the scanned barcode by adding workflow rules with Triggers, Conditions and Actions

Assign a Name for the Configuration

The configuration itself acts as a kind of container, which allows to save all profiles together and to edit or download them afterwards.

Create a unique name for the Configuration: Velocity

The configuration is now saved and can be downloaded to be copied to your mobile device.



Successfully saved configuration

SAVED CONFIGURATION FILES

Configurations

CONFIGURATION FILE	LAST MODIFIED UTC+1	CREATED UTC+1
☐ Velocity	11/3/20, 3:17 PM	11/3/20, 3:17 PM

ivanti | Wavelink

ACCELERATE DATA CAPTURE

Benefit from advanced control of device-specific barcode scanning parameters

PRESERVE HOST SYSTEMS

Get the latest mobile experience without changes to your enterprise systems

MAINTAIN USER EXPERIENCE

Preserve worker familiarity by easily porting web app

Ivanti Wavelink **helps organizations leverage modern mobile technology in the warehouse and across supply chain to accelerate productivity** without modifying WMS or backend systems. Industries around the world such as manufacturing, retail, hospitality, healthcare, warehousing and field force automation rely on Ivanti Wavelink’s full portfolio of enterprise mobility, device management and voice-enablement solutions - improve productivity, speed operations, and save costs

PROGLOVE - *for a smarter workforce*

MORE EFFICIENCY

Operators can save up to 6 seconds per scan

BETTER ERGONOMICS

Hands-free scanning to streamline workflows

RETURN ON INVESTMENT

Efficiency gains using wearable scanners

The human workforce is at the core of what we do and we aspire to be the leader in industrial wearables and software to enable a smarter workflow.

ProGlove builds the **lightest, smallest and toughest wearable barcode scanners** in the world, connecting workers to actionable information. More than 500 renowned organizations in manufacturing, production, logistics, aerospace, and retail use these smarter workforce solutions. ProGlove was founded in December 2014 after winning the Intel "Make it Wearable"

Challenge in Silicon Valley and by 2020 was named a Technology Pioneer by the World Economic Forum. ProGlove employs 200 people from over 40 countries with offices in Munich, Chicago, and Belgrade.