

MX-Browser

Configuring the MX Browser

MX Browser has a variety of settings allowing you to configure it to perform the best for your website. URL specific configurations contain the following settings (from 2.0.0 version):

Browser Settings

- **Home URL** - The URL address that opens when the browser launches (Default configuration) or when Home button is pressed.
- **Lock user to home URL** - User cannot navigate away from the given URL, sub-pages can still be accessed. Make sure to use this feature if mobile device management tool is used (e.g. SOTI).
- **Enable navigation buttons** - Controls the access of back and forward buttons.
- **Enable address bar** - Controls the access of address bar.
- **Hide toolbar when scrolling down a page** - Hides the toolbar to use the screen more efficiently.
- **Soft scan button** - User can have a button on the screen that can trigger the MX mobile terminal.
- **Show keyboard for input** - Show the default soft-keyboard when tapping on an input field within a web page.
- **Fixed zoom** - Zoom settings (manual, range and default zoom level) that applies of the page when it is opened. The first focused field on your page is considered when zooming. If this settings is changed MX Browser requests a reload on actual page to apply its effect (this operation may result loss of data on the current page).

Navigation

- **Define input types** - Select which input types will be focused on the web page for inserting a read result.
- **Initially focused input field** - Configure which input field will be initially focused when web page is loaded.
 - first input field on the website
 - last input field on the website
 - or custom
 - either to select the field from a list about available field on the defined web page
 - or configure the focus by name or DOM attribute
- **Move focus after read** - Configure how the focus changes after reading a bar code successfully.
 - move to the next input field
 - move to the previous input field
 - or don't change the focus
- **Allow focus on touch or from JS** - Allows to move the focus when an input field is tapped by the user (users can select different field to focus than it is), move the cursor via JavaScript or pressing TAB on a keyboard. If the selected input field's type is selected in "**Define input types**", then the read result will be inserted there from MX mobile terminal.
- **Automatically submit forms** - It is possible to submit a form when all the users are ready with a form. There are two possible ways to achieve this:
 - Last scan was performed on the last input field and "**Move focus after read**" is set to "**Next input field**"
 - Last scan was performed on the first input field and "**Move focus after read**" is set to "**Previous input field**"
- **Confirm auto-submit forms** - If this setting is enabled, a dialog is shown to the user to get a confirmation before submitting the form.

Advanced

- **Keyboard wedge mode** - MX Browser emulates keyboard when a barcode is entered into an input field. If it is tuned on, all other navigation settings are ignored.
- **HTML inserts** - Definition of HTML code that will be added to HEAD or BODY section of the page. If this settings is changed MX Browser requests a reload on actual page to apply its effect (this operation may result loss of data on the current page).
- **Send event on read result received** - After scanning with an MX device, the browser will send an event to the web page instead of filling the configured input field. The event can be handled on the website with having an event-listener that will listen to the

"readResultReceived" event.

- **Fill out focused input field with read result** - Read result is also inserted into input field after the "readResultReceived" event is triggered. It means that the web page may implement additional logic with the read result additionally to input it into the field.
- **Send event if connection status changed** - The browser sends an event to the web page when an MX device is connected or disconnected. The event can be handled on the website with having an event-listener that will listen to the "connectionStateChanged" event.

Input fields navigation

MX Browser allows you to configure how you want to navigate through the input fields of your web sites. If you enable keyboard wedge mode these settings are ignored.

Defining input types

You can define which type of input fields may be focused on when the browser is processing successful read result. This configuration can be done in **"Define input types"** from the **"Navigation"** group. Supported types are (bold ones are set by default): hidden, **email**, **number**, **password**, **search**, **tel**, **text**, **textarea**, **url**.

Scanner focus

There are many settings for fine-tuning how focus changes after a successful read. When you first open the web page the scanner focus is on the first input field that is enabled in the "Define input types" setting by default. That means this input field will be the first to receive a scan result when scanning a bar code with a Cognex MX mobile terminal. If there are no such input fields on the page or all input types are disabled, the scanned bar code will be copied on your devices clipboard.

It is always possible to read a code into address bar of MX Browser.

The initially focused field can be changed by going in the MX Browser settings and selecting **"Initially focused input field"**:

1. **First input field** - The first input field on the web page that's type is enabled in the **"Define input types"** setting will be the first to focus initially.
2. **Last input field** - The last input field on the web page that's type is enabled in the **"Define input types"** setting will be the first to focus initially.
3. **Custom input field** - Provide additional ways for configuring the initially focused input field.
 - **Web page URL** - You will be given a choice to pick an input field from any web page (only the types enabled in **"Define input types"** will be shown), and that is the first to focus initially.
 - **Define focused input field** - You can configure which input field will be the first to focus initially by configuring the input name or defining a DOM attribute name and value for that input.

It is also possible to define where the focus should move after a successful read was performed in **"Move focus after read"** settings:

1. **Next input field** - The scanner will focus on the next available input field after a successful read.
2. **Previous input field** - The scanner will focus on the previous input field after a successful read.
3. **Do not move focus** - The focus will not change, which means that the scanner will extend the text in the same input field, unless the focus change happens manually (with **"Allow focus on touch or from JS"** enabled and tapping on another input field or if another input field is focused using JavaScript).

When tapping on an input field on the web page, if that input fields type is enabled in the **"Define input types"** setting, by default the scanner focus will move to that input field. This can be changed in the MX Browser settings by changing the **"Allow focus on touch or from JS"** switch. When this setting is disabled, the scanner focus changes will happen only as it is defined by the **"Initially focused input"** and **"Move focus after read"** settings. Keep in mind, if this setting is enabled and you select an input field that's type is disabled in the **"Define input types"** setting, the scanner focus will not go into that input field even though a cursor will appear there. This allows you to type something into that input field, but you cannot scan a bar code, and have the result inserted since that type is disabled.

Automatically submit HTML forms

MX Browser can automatically submit an HTML form. To enable this feature you have to change the **"Automatically submit forms"** switch. When this setting is enabled, the form will be automatically submitted in two cases:

1. After scanning a bar code in the last input field on the web page, with the **"Move focus after read"** setting set to **"Next input field"**.
2. After scanning a bar code in the first input field on the web page, with the **"Move focus after read"** setting set to **"Previous input field"**.

To prevent accidental form submissions, right below the **"Automatically submit forms"** setting you will find the **"Confirm auto-submit forms"** setting. If this is enabled, then a confirmation is displayed every time before automatically submitting the form.

Integration with Cognex Hardware

It is possible to interact with a Cognex Mobile terminal from MX Browser using JavaScript on your webpage. There are two events that caught from JavaScript code: *"readResultReceived"* and *"connectionStateChanged"*.

readResultReceived

The *"readResultReceived"* event is triggered when the MX device is triggered and **"Send event on read result received"** is enabled. The read result that is sent to that event has the following format:

```
{
  "readResults" : [
    {
      "goodRead" : 1,
      "readString" : "2112345678900",
      "symbology" : 4,
      "symbologyString" : "UPC-EAN",
      "xml" : "...",
    }
  ],
  "subReadResults" : [
  ],
  "xml" : "...",
}
```

The main *readResult* object contains three key-value pairs:

1. *readResults* (Array): An array of read result objects that were scanned with an MX Device. If multicode is not enabled on the device, then only one result is in this array. If multicode is configured, then this array contains the main result and all the sub read results.
2. *subReadResults* (Array): An array of read result objects excluding the main result. For single code scanning, this array will be empty.
3. *xml* (String): The raw XML string received by the reader.

To listen for *"readResultReceived"* event on your website, add the JavaScript event listener like in the example below:

```
document.addEventListener('readResultReceived', function(barcode_result){
  alert(JSON.stringify(barcode_result.detail));
});
```

connectionStateChanged

The *"connectionStateChanged"* event is triggered when the MX device changes its connection state and **"Send event if connection status changed"** is enabled. The attached object has a number representing the new connection state:

- 0 - Disconnected, the readerDevice is not connected to any remote system.
- 1 - Connecting, the readerDevice is in the process of establishing a connection to a remote system.
- 2 - Connected, the readerDevice is connected to a remote system.
- 3 - Disconnecting, the readerDevice is in the process of disconnecting from a remote system.
- 4 - Unknown, the readerDevices connection state is unavailable.

To listen for *"connectionStateChanged"* event on your website, add the following JavaScript event listener:

```
document.addEventListener('connectionStateChanged', function(connection_state){
  alert(JSON.stringify(connection_state.detail));
});
```

Other possibilities to control MX mobile terminal

The MX Browser allows communication from the web site to the MX Device as well. An object named as "cmbSDK" is available from Javascript after the web page has been loaded. By using this object it is possible to:

- start scanning without having to press a button on the MX mobile terminal
- stop scanning if nothing is being scanned
- trigger userEvent1 and userEvent2 (configurable events on MX mobile terminals)

All you have to do is add the following functions to your web site and call them whenever you need (using timer, on button press, etc.).

```
function startScanning() {
    cmbSDK.triggerStartScan();
}
function stopScanning() {
    cmbSDK.triggerStopScan();
}
function triggerUserEvent1() {
    cmbSDK.triggerUserEvent1();
}
function triggerUserEvent2() {
    cmbSDK.triggerUserEvent2();
}
```

Disable trigger buttons

MX Browser provides functions that can be called from the source of a web page to disable trigger buttons.

```
/*
 * The following functions configure the trigger buttons of an MX mobile terminal.
 * @param state      - 0 disable
 *                   - 1 enable triggering
 */
cmbSDK.setLeftTriggerButton(int state);
cmbSDK.setRightTriggerButton(int state);
cmbSDK.setGripTriggerButton(int state);
```

Caution! Use this function carefully as you can create a state where MX is not responding to your users.

MX Browser sets all of your trigger buttons to enable triggering, when MX Browser is reopened or a new page is loaded.

Mass configuration - MDM

MX Browser provides possibility to push configuration via MDM and prevent that the end user may change settings. This functionality is supported for Android and planned to be available in next release for iOS.

- [Android](#)
- [iOS](#)

1. Create a configuration that serves your needs in the best way. We would suggest to involve end users in testing if it is really the best solution.
2. If you are satisfied with the result, then create a configuration file with "**Export configuration**" (available from action menu on the list of Configurations screen) . Get the file from the device in your preferred way (e-mail, download...)
3. Push this file from your EMM/MDM software to all device you want to configure and make sure that it is located in "*Internal Storage/Android/data/com.cognex.mxbrowser/*".
4. It is done. All settings will be effective from the next start of MXBrowser.

Important to know that in this case you are not able to modify any setting on the configured device (so does your end users).

1. Create a configuration that serves your needs in the best way. We would suggest to involve end users in testing if it is really the best solution.
2. If you are satisfied with the result, then create a configuration file with "**Export configuration**" (available from action menu on the list of Configurations screen) . Get the file from the device in your preferred way (e-mail, download...)
3. Upload your file to a host that is accessible by your managed phones.
4. Set the *App configuration* for *MX Browser* to the following:

```
<dict>
  <key>configurationURL</key>
  <string>https://myserver.com/firmwares/firmware_file_name</string>
</dict>
```

5. It is done. All settings will be effective from the next start of MXBrowser.

Important to know that in this case you are not able to modify any setting on the configured device (so does your end users).