Emergency Contact List Application

**Software Architecture Document**

**Version 1.1**

[1 Introduction 3](#_Toc354594497)

[1.1 Overview 3](#_Toc354594498)

[1.2 Acronyms / Definitions 3](#_Toc354594499)

[1.3 References 4](#_Toc354594500)

[2 Software Architecture 5](#_Toc354594501)

[2.1 Architecture Overview 5](#_Toc354594502)

[2.2 Application Components 6](#_Toc354594503)

[2.3 Contact List Format 6](#_Toc354594504)

[3 Testing 7](#_Toc354594505)

[3.1 Test Plan and Results 7](#_Toc354594506)

[4 Possible Future Enhancements 7](#_Toc354594507)

[4.1 Using an SQL Database for storing the Emergency Contact List 7](#_Toc354594508)

[5 Revision History 8](#_Toc354594509)

# Introduction

### Overview

The Emergency Contact List (ECL) Application manages a list of emergency contacts on the device.

The applications displays the contact list to the user, where the user can scroll through the list or search for specific contacts. If a contact contains a backup contact, the user can jump to that contact information. If the device is online, the application provides easy access to contact the list members by phone or email.

The contact list is accessible to the user whether or not the device is online. The contact list is stored in local storage when the application is closed.

The contact list is updated by pushing a new list through the BlackBerry Enterprise Server (BES) MDS Connection Service.

The application receives data through the BlackBerry VPN secure connection and stores data on the device in the secured Work Perimeter.

The application is built with BlackBerry 10 WebWorks HTML5, CSS, and JavaScript. It uses bbui.js to style and render the screens presented to the user.

### Acronyms / Definitions

|  |  |
| --- | --- |
| **Acronym /Definition** | **Description** |
| UI | User Interface |
| XML | eXtensible Markup Language |
| API | Application Programming Interface |
| BB10 | BlackBerry 10 |
| BDS | BlackBerry Device Service |
| MDS | Mobile Data System |
| BES | BlackBerry Enterprise Server |
| ECL | Emergency Contact List Application |
| EIS | Enterprise Information System |
| CRUD | **C**reate, **R**ead, **U**pdate, **D**elete |

### References

The Emergency Contact List Application is a BlackBerry 10 Webworks Application that uses the following main features and APIs.

* Bbui.js – toolkit for generating BB10 style HTML content for the user interface. https://github.com/blackberry/bbUI.js/wiki
* HTML5 WebWorks PushService – Receives push message from the BES. <https://developer.blackberry.com/html5/apis/blackberry.push.pushservice.html>
* HTML5 WebWorks Notifications – Create a notification for the Hub (universal inbox). <https://developer.blackberry.com/html5/apis/notification.html>
* HTML5 WebWorks Invoke – invoke the application from a notification. <https://developer.blackberry.com/html5/apis/blackberry.invoke.html>
* HTML5 Local Storage – save application state when closed. <http://developer.blackberry.com/html5/apis/localstorage.html>

# Software Architecture

### Architecture Overview



Figure 1. Architecture Overview

1. The Push Test Service sends a request to the BES MDS Connection Service to push a new contact list to the device.
2. The push message is delivered through the WebWorks Push Service API to the application’s Push Handler.
3. The Push Handler adds the request to the Request List, and sends a notification to the Hub to alert the user that a new contact list has arrived. The application can be used to view the contact list.

### Application Components

The application consists of the following main components.

* Screens – HTML5 using bbui.js screens and the JavaScript for generating screen content and handling screen interactions.
* Push Handler – This component registers to receive request push messages.
* Contact List – The data for the list of contacts. It uses local storage to persist this data when the application is closed.

### Contact List Format

The contact list is pushed to the device as JSON string (text). Here is an example in JavaScript.

var teamList = [

{

"title": "US Team",

"list": [

{

"name": "Joseph Smith",

"title": "Software Business Analyst",

"officePhone": "999-488-1111",

"cellPhone": "519-555-2222",

"email": "joseph.smith@company.com",

"bbPin": "FFFF9999",

"backup": "Dennis Jones"

},

{

"name": "Dennis Jones",

"title": "System Specialist",

"officePhone": "999-488-1111",

"cellPhone": "519-555-2222",

"email": "dennis.jones@company.com",

"bbPin": "3333333F",

"backup": undefined

},

]

},

{

"title": "India Team",

"list": [

{

"name": "Rajesh Doe",

"title": "Systems Analyst",

"officePhone": "999-555-1111",

"cellPhone": "519-555-2222",

"email": "rajesh.doe@company.com",

"bbPin": undefined,

"backup": "Ricky"

}

]

}

];

This JavaScript data will look like this when encoded as a JSON string:

"[{"title":"US Team","list":[{"name":"Joseph Smith","title":"Software Business…

# Possible Future Enhancements

### Using an SQL Database for storing the Emergency Contact List

* The present implementation of the ECL application stores the emergency contact list details in localStorage.
* This can be enhanced to create a storage module that uses a SQL Database. This Module could expose CRUD methods to create, read, update and delete contact list entries.