Tizen Web Device API

Kisub Song
Samsung Electronics
Contents

• Introduction
  • Why Tizen Web Device APIs?
• Tizen Web Device API
  • Tizen Web Device API Background
  • 25 Modules of Tizen Web Device API
  • Updates from Tizen 1.0
• Device APIs in Detail
  • Application Control / Hybrid Application / Searching with Filter
• Summary
• Appendix
Introduction
Introduction

• Tizen is a **Web-based platform**

• A **Web application** is composed with Web technologies, such as HTML5, JavaScript, and CSS

• Tizen Web device APIs provide the characteristic features of the Tizen platform
Web Device APIs in Tizen Architecture
Tizen Web APIs

Device API
- Application
- Bluetooth
- Calendar
- Contact
- Messaging

W3C
- HTML5
- CSS3
- Geolocation
- Touch Event
- File

Miscellaneous
- Web GL
- Typed Array
- Full Screen API
- JSON
- URI Scheme

...
Why Tizen Web Device APIs?

• Limitations of standard specifications
  • Most of them are still working drafts
  • No full support for the Tizen characteristic features

• Tizen Web device APIs
  • Device APIs give the characteristic features of the Tizen platform
  • Device APIs have been updated agilely
Tizen Web Device APIs Design

• Follows W3C specification API style
  • Numerical constants avoided but string enumerations used
  • Most methods are asynchronous
  • Success and error callback, and constructors used

• Based on standard technologies
  • Tizen Web device API specifications are written in WebIDL
  • Implementation follows the fundamental rule of ECMAScript 5.1 and W3C WebIDL recommendation (for example, type conversion)

• Defined tizen namespace
  • All Tizen device APIs exist in the tizen namespace
  • tizen is the global object to which all Tizen device APIs are bound
Tizen Web Device API
Web Runtime

Web Runtime

Web application

WebKit

WebKit2

WebCore

JavaScriptCore

config.xml

Web Runtime

AppFW

Tizen Device APIs

Tizen platform core

WebKit

JavaScriptCore

JSC API

JSC API

JSC API
Configuration for Web Applications

- **config.xml file**
  - Defined in the W3C *Packaging and XML Configuration* specification
  - Tizen namespace is defined for describing Tizen specific features

- **Properties related with device APIs**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>tizen:application</td>
<td>Describes the application ID and package ID.</td>
<td><code>&lt;tizen:application id=&quot;UJedH641XL.Test&quot; package=&quot;UJedH641XL&quot; required_version=&quot;2.1&quot; /&gt;</code></td>
</tr>
</tbody>
</table>
| tizen:app-control | Defines the application control that this application provides. | `<tizen:app-control>  
  <tizen:src name="controller.html" />
  <tizen:operation name="http://tizen.org/appcontrol/operation/view" />
</tizen:app-control>` |
| tizen:privilege   | Describes the privilege that this application needs.      | `<tizen:privilege name="http://tizen.org/privilege/appmanager.kill" />` |
Application Signing

- Applications must be signed with 2 signatures:
  - **Author signature**
    - Determines the integrity of an application package as intended by the developer
    - Confirms that all applications that are signed with the same author certificate are trustworthy
  - **Distributor signature**
    - Generated by an application publisher
    - Determines the privilege level of the application.
Privilege Levels

- Privileges are categorized into 3 levels
  - **public**: open to all Tizen application developers
  - **partner**: for partners registered on the Tizen store
  - **platform**: for managing the Tizen platform

<table>
<thead>
<tr>
<th>Application distributor certification</th>
<th>API privilege level</th>
<th>Non-privileged</th>
<th>public</th>
<th>partner</th>
<th>platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>public</td>
<td></td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>partner</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>X</td>
</tr>
<tr>
<td>platform</td>
<td></td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Privileges

- Some APIs require privileges for using them from an application
  - Privilege level: application needs to have higher privilege level than the required one to use the API
  - Privilege: needs to be described in `config.xml`

```java
void launch(ApplicationId id, optional SuccessCallback? successCallback, optional ErrorCallback? errorCallback);
```

- Required privileges are described in API specification documents
Tizen Web Device API Modules

Tizen common
- Tizen

Content
- Content
- Download

Social
- Bookmark
- Calendar
- Call History
- Contact
- Data Synchronization

Application
- Alarm
- Application
- Data Control
- Package

System
- Power
- System Information
- System Setting
- Time

User interface
- Notification

Communication
- Bluetooth
- Messaging
- Network Bearer Selection
- NFC
- Push
- Secure Element

Input/Output
- File System
- Message Port
# Updates from Tizen 1.0

<table>
<thead>
<tr>
<th>Newly introduced</th>
<th>Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Download</td>
</tr>
<tr>
<td></td>
<td>Data Control</td>
</tr>
<tr>
<td></td>
<td>System Setting</td>
</tr>
<tr>
<td></td>
<td>Push</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Obsolete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geocoder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Renamed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
<td>Call History</td>
</tr>
<tr>
<td>Media-Content</td>
<td>Content</td>
</tr>
</tbody>
</table>
Device APIs in Detail
Application Control (AppControl)
Application Control

- Application control enables launching other applications based on the functionalities needed
  - For example: Contacts app uses the Gallery app to select an image file
Application Control

**Tizen application**

Request launch with the ApplicationControl instance.

**AppControl registry**

Resolve app ID to launch from given app control.

**AppControl manager**

Launch the selected application and pass the ApplicationControl instance.

**Base application**

Caller application gets the response data.

**Reply the result**

operation: "http://tizen.org/appcontrol/operation/view"
url: "http://example.com"
mime: "text/html"
key: values
...
Example: Requesting Application

- Requesting AppControl:

```javascript
var appControl = new tizen.ApplicationControl(
    'http://tizen.org/appcontrol/operation/pick', null, 'image/*', null);

tizen.application.launchAppControl(appControl, null,
    function() { console.log("Launching AppControl succeeded"); },
    function(e) { console.log("Launching AppControl failed"); },
    { onsuccess : function(data) {
        console.log("AppControl returned success");
        data.forEach(function(v, i) {
            console.log("[\"+i\"] key : \"+v.key);
            v.value.forEach(function(vv, vi) {
                console.log("---- value\#\"+vi\" : \"+vv");
            });
        });
    },
    onfailure : function() { console.log("AppControl returned failure"); } });
```

Printed log:

- Launch Reply
- Onsuccess function is invoked when the base application returns the result data.
- "AppControl returned success"
- "[0] key : http://tizen.org/appcontrol/data/selected"
- "---- value#0 : /opt/usr/media/Images/image2.jpg"
Providing AppControl in Your Application (1/2)

- Edit AppControl in the IDE:

```xml
<tizen:app-control>
  <tizen:src name="control.html" />
  <tizen:operation name="http://tizen.org/appcontrol/operation/view" />
  <tizen:mime name="image/*" />
</tizen:app-control>
```
Providing AppControl in Your Application (2/2)

- Handle passed ApplicationControl instance and reply the result

```javascript
function initAppControl() {
    var currentApp = tizen.application.getCurrentApplication();
    var reqedAppControl = currentApp.getRequestedAppControl();
    var appControl = reqedAppControl.appControl;
    console.log("operation:" + appControl.operation + "/ uri:" + appControl.uri + "/ mime:" + appControl.mime);
    // Do something
    reqedAppControl.replyResult([tizen.ApplicationControlData('key0', ['value0_0']),
                                 tizen.ApplicationControlData('key1', ['value1_0'])]);
} window.addEventListener('load', initAppControl, false);
```
Hybrid Application with Message Port
Message Port

• Features
  • Application identifies message ports by application ID and port name
  • Application can open a message port that restricts access from unidentified application
  • Native API also has a message port that is compatible with the Web message port

• Limitations
  • Only string data can be sent through a message port
  • It can send a limited amount of data at a time (4KB)
Communication through Message Port

- **LocalMessagePort** is used to receiving messages.
- **RemoteMessagePort** is used to send messages.
- Application can send messages using the `sendMessage()` method of `RemoteMessagePort` object.
Example: Communication through Message Port

```javascript
var remoteMsgPort =
    tizen.messageport.requestRemoteMessagePort('gC5tbRRSOx.SvcApp', 'PORT_A');
var localMsgPort =
    tizen.messageport.requestLocalMessagePort('PORT_R');
remoteMsgPort.sendMessage( [
    { key:'CMD', value:'openWindow' },
    { key:'OPTION', value:'bx' }
],
    localMsgPort
);

var localMsgPort =
    tizen.messageport.requestLocalMessagePort('PORT_A');
var watchId = localMsgPort.addMessagePortListener( function (data, remoteMsgPort) {
    console.log('Received data is...');
    // ......
    remoteMsgPort.sendMessage( [
        { key:'ANSWER', value:'OK' },
        { key:'PARAM1', value:'WFWQ' }
    ];
} );
```
Trusted Message Port

- Trusted message port allows access from the application signed with the same certificate with the current application.

Tizen application 1
[PRHZQtLNMX.UIApp]
Signed with cert A

Tizen application X
[ORLQb3Wf0q.AppX]
Signed with cert X

Trusted remote message port

Connection established

Trusted local message port name:'PORT_A'
Signed with cert A

Tizen application 2
[gC5tbRRSOx.SvcApp]
Signed with cert A

Trusted remote message port

Connection refused
Hybrid Application Package

- Combines a web application and native applications in a package
- Create in the SDK IDE: Project > Properties > Project References

Tizen application package
[pkgID: X1A9waVdw0]

Web UI application
[appID: X1A9waVdw0.SampleApp]

Native Service Application
[appID: X1A9waVdw0.SampleAppService]
Searching with Filters
Filters

- Filters are Tizen common interfaces to make complex queries
- Filter types:
  - Attribute filter, attribute range filter, and composite filter
- Filters are used with various Tizen device API modules

<table>
<thead>
<tr>
<th>Module</th>
<th>Function [Interface::Method()]</th>
<th>Search for (result type interface)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact</td>
<td>ContactManager::find()</td>
<td>Person</td>
</tr>
<tr>
<td></td>
<td>AddressBook::find()</td>
<td>Contact</td>
</tr>
<tr>
<td>Calendar</td>
<td>Calendar::find()</td>
<td>CalendarItem (CalendarTask or CalendarEvent)</td>
</tr>
<tr>
<td>CallHistory</td>
<td>CallHistory::find()</td>
<td>CallHistoryEntry</td>
</tr>
<tr>
<td>Content</td>
<td>ContentManager::find()</td>
<td>Content (VideoContent or AudioContent or ImageContent)</td>
</tr>
<tr>
<td>Messaging</td>
<td>MessageStorage::findMessage()</td>
<td>Message</td>
</tr>
<tr>
<td></td>
<td>MessageStorage::findConversations()</td>
<td>MessageConversation</td>
</tr>
<tr>
<td></td>
<td>MessageStorage::findFolders()</td>
<td>MessageFolder</td>
</tr>
</tbody>
</table>
Attribute Names

- **Name of the attribute to be searched**
  - Generally, matches with the name of the attribute.
    - For example: to search the Contact objects `birthday` attribute, use the attribute name `birthday`.
  - The attribute name can have 2 or more levels separated by a period.
    - For example: `phoneNumbers.number`.
  - Available attribute names are described on the SDK documents.

**Contact Filter Attributes**

The following table lists the filter types you can use with specific contact attributes in the methods of the `AddressBook` interface.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Attribute filter supported</th>
<th>Attribute range filter supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>personId</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>lastUpdated</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>isFavorite</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Filter Types

- **Attribute filter**
  
  ```javascript
  var filter = new tizen.AttributeFilter('name.firstName', 'EXACTLY', 'Chris');
  ```
  
  Describe the filter with the SQL `where` clause:
  ```sql
  WHERE name.firstName = 'Chris'
  ```

- **Attribute range filter**
  
  ```javascript
  var filter = new tizen.AttributeRangeFilter('name.firstName', null, new Date(1983, 0, 1));
  ```
  
  Describe the filter with the SQL `where` clause:
  ```sql
  WHERE birthday < 410227200
  ```

- **Composite filter**
  - Combines several filters into a set
Example: Filters in Pseudo-SQL Expression

```javascript
var filter = new tizen.CompositeFilter({
  type: "UNION",
  filters:[
    new tizen.CompositeFilter({
      type: "INTERSECTION",
      filters:[
        new tizen.AttributeFilter("name.firstName", "CONTAINS", "Chris"),
        new tizen.AttributeFilter("name.lastName", "EXACTLY", "Smith")
      ]
    }),
    new tizen.AttributeFilter("phoneNumbers.number", "ENDSWITH", "5555")
  ]
});

addressBook.find(filter, null,
    function(contacts){ console.log("Retrieved : " + contacts.length); },
    function(err){ console.log("Error : " + err.message); });
```

Describe the filter with the pseudo-SQL `where` clause:

```
( ( ( name.firstName LIKE '%Chris%' ) AND ( name.lastName = 'Smith' ) )
  OR
  ( phoneNumbers.number LIKE '5555%' )
)```
Summary

- **Tizen Web Device API supports Tizen characteristic features.**
  - Tightly coupled with the structure of the Web Runtime
  - Privileges have been defined to control API access authority
- **Tizen Web Device API provides 25 modules**
  - 12 have been newly introduced since Tizen 1.0
- **Tizen Web Device API provides various special features**
Appendix
<table>
<thead>
<tr>
<th>Privilege</th>
<th>Module</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partner</td>
<td>Application</td>
<td>kill / getAppCert</td>
</tr>
<tr>
<td></td>
<td>Data Control</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>Network Bearer Selection</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>Secure Element</td>
<td>ALL</td>
</tr>
<tr>
<td></td>
<td>System Information</td>
<td>imei, msisdn, msin</td>
</tr>
<tr>
<td>Platform</td>
<td>Package</td>
<td>install / uninstall</td>
</tr>
<tr>
<td></td>
<td>Bluetooth</td>
<td>setVisible</td>
</tr>
<tr>
<td></td>
<td>Bookmark</td>
<td>ALL</td>
</tr>
</tbody>
</table>
# Application Control Model

<table>
<thead>
<tr>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation</strong></td>
<td></td>
</tr>
<tr>
<td>• Defines the action to be performed by the application control</td>
<td><a href="http://tizen.org/appcontrol/operation/view">http://tizen.org/appcontrol/operation/view</a></td>
</tr>
<tr>
<td>• Mandatory item for application control</td>
<td><a href="http://tizen.org/appcontrol/operation/pick">http://tizen.org/appcontrol/operation/pick</a></td>
</tr>
<tr>
<td>• IRI style</td>
<td><a href="http://tizen.org/appcontrol/operation/call">http://tizen.org/appcontrol/operation/call</a></td>
</tr>
<tr>
<td><strong>URI</strong></td>
<td></td>
</tr>
<tr>
<td>• Data on which the action is performed</td>
<td>http, tel, mailto ...</td>
</tr>
<tr>
<td><strong>MIME</strong></td>
<td></td>
</tr>
<tr>
<td>• Specific URI type</td>
<td></td>
</tr>
<tr>
<td>• If MIME is not set and URI has a file scheme, the MIME is automatically determined by local file</td>
<td>audio/<em>, video/</em> ...</td>
</tr>
<tr>
<td><strong>Application control data</strong></td>
<td></td>
</tr>
<tr>
<td>• Key-value pairs providing additional information for the service request</td>
<td></td>
</tr>
<tr>
<td>• Dataset</td>
<td></td>
</tr>
</tbody>
</table>

*AppControl Manager resolves the application whose operation, URI, and MIME type match the requested ApplicationControl instance exactly.*
### Base AppControls

<table>
<thead>
<tr>
<th>Base application</th>
<th>Operation</th>
<th>URI</th>
<th>MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet (Browser)</td>
<td><a href="http://tizen.org/appcontrol/operation/view">http://tizen.org/appcontrol/operation/view</a></td>
<td>http</td>
<td>-</td>
</tr>
<tr>
<td>MusicPlayer</td>
<td><a href="http://tizen.org/appcontrol/operation/view">http://tizen.org/appcontrol/operation/view</a></td>
<td>-</td>
<td>audio/*</td>
</tr>
<tr>
<td>VideoPlayer</td>
<td><a href="http://tizen.org/appcontrol/operation/view">http://tizen.org/appcontrol/operation/view</a></td>
<td>-</td>
<td>video/*</td>
</tr>
<tr>
<td>Phone</td>
<td><a href="http://tizen.org/appcontrol/operation/dial">http://tizen.org/appcontrol/operation/dial</a></td>
<td>tel</td>
<td>-</td>
</tr>
<tr>
<td>Call</td>
<td><a href="http://tizen.org/appcontrol/operation/call">http://tizen.org/appcontrol/operation/call</a></td>
<td>tel</td>
<td>-</td>
</tr>
<tr>
<td>Camera</td>
<td><a href="http://tizen.org/appcontrol/operation/create_content">http://tizen.org/appcontrol/operation/create_content</a></td>
<td>-</td>
<td>image/jpeg, video/3gpp</td>
</tr>
<tr>
<td>FileManager</td>
<td><a href="http://tizen.org/appcontrol/operation/pick">http://tizen.org/appcontrol/operation/pick</a></td>
<td>-</td>
<td><em>/</em>, image/<em>, audio/</em>, video/*</td>
</tr>
<tr>
<td>Email</td>
<td><a href="http://tizen.org/appcontrol/operation/compose">http://tizen.org/appcontrol/operation/compose</a></td>
<td>mailto</td>
<td>-</td>
</tr>
<tr>
<td>ImageViewer</td>
<td><a href="http://tizen.org/appcontrol/operation/view">http://tizen.org/appcontrol/operation/view</a></td>
<td>-</td>
<td>image/*</td>
</tr>
</tbody>
</table>

* The base AppControls are provided by the system default applications
Example: Requesting Application

```javascript
var appControl = new tizen.ApplicationControl(
  "http://tizen.org/appcontrol/operation/view",
  "file:///opt/usr/media/Images/image1.jpg",
  null,
  null);

tizen.application.launchAppControl(appControl,
  null,
  function() {
    console.log("launch appControl succeed");
  },
  function(e) {
    console.log("launching appControl failed : " +
    e.message);
  });
```

Operation

URI

MIME is null, but would be determined to be image/jpeg based on the URI value

Launch

ImageViewer
Providing AppControl in Your Application

• 3rd party application can provide a new AppControl by describing a new AppControl in the config.xml file

```xml
<tizen:app-control>
  <tizen:src name="[PAGE]" />
  <tizen:operation name="[OPERATION]" />
  <tizen:uri name="[URI_SCHEME]" />
  <tizen:mime name="[MIME_TYPE]" />
</tizen:app-control>
```

The path of HTML file launched initially when this AppControl is requested.
### Attribute Filter - Flags

<table>
<thead>
<tr>
<th>String only</th>
<th>Case-sensitive</th>
<th>SQL ‘WHERE’ expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXACTLY</td>
<td>X</td>
<td>ATTR = 'VALUE'</td>
<td>Match exactly with the specified value</td>
</tr>
<tr>
<td>FULLSTRING</td>
<td>O</td>
<td>ATTR LIKE 'VALUE'</td>
<td>Match with the whole string but case-insensitive</td>
</tr>
<tr>
<td>CONTAINS</td>
<td>O</td>
<td>ATTR LIKE '%VALUE%'</td>
<td>Contain the specified string and case-insensitive</td>
</tr>
<tr>
<td>STARTSWITH</td>
<td>O</td>
<td>ATTR LIKE 'VALUE%'</td>
<td>Start with the specified string and case-insensitive</td>
</tr>
<tr>
<td>ENDSWITH</td>
<td>O</td>
<td>ATTR LIKE '%VALUE%'</td>
<td>End with the specified string and case-insensitive</td>
</tr>
<tr>
<td>EXISTS*</td>
<td>-</td>
<td>IS NOT NULL or ATTR &lt;&gt; ''</td>
<td>Have any value not null nor empty string</td>
</tr>
</tbody>
</table>

* If the EXISTS flag is set, the matchValue is not necessary and can be ignored*
Modules
Tizen APIs (1/2)

• Tizen
  • Provides Tizen’s common functionalities
    • Contains object interfaces that are commonly used throughout the other modules
  • The `tizen` object is the topmost object providing the foundations for accessing Tizen device features
Tizen APIs (2/2)

• Tizen’s common object interfaces
  • Generic asynchronous callback handling
    • Success callback for methods that do not require a return value
    • Error callback for methods that require an error as input parameter
  • Generic error and exception handling
    • Tizen APIs throw a WebAPIException object and returns a WebAPIError object through error callbacks
    • These follow the style and types of error of DOM4 specification.
  • Filters and sort modes
    • Filters are used to limit query results and compose complex queries
    • Generally, filters are used with the find() method in Calendar, Call History, Contact, Content, and Messaging modules
Application APIs (1/2)

• Alarm
  • Schedules an application to be launched at a specific time

• Application
  • Manages current application (this application)
  • Manages other applications
    • Retrieves information of applications
  • Launches other applications
    • Application control
Application APIs (2/2)

- **Package**
  - Retrieves information of installed packages
  - Checks the updates of the installed package list
  - Installs or uninstalls packages
  - **partner privileged**

- **Data Control**
  - Exchanges specific data with the data control provider application
    - Data control provider application can be implemented as a native application
  - Data control types:
    - SqlDataControl: SQL-type data control – data consists of rows and columns
    - MapDataControl: key-value-type data control
  - **partner privileged**
Communication APIs (1/2)

• **Bluetooth**
  • Manages Bluetooth devices
  • Discovers nearby devices, and bonds or pairs with found devices
  • Connects to devices to exchange data with them

• **NFC**
  • Manages NFC devices
  • Detects NFC tag and peer
  • Exchanges NDEF data

• **Messaging**
  • Sends or receives SMS, MMS, or email messages
  • Retrieves the message storage
Communication APIs (2/2)

- **Network Bearer Selection**
  - Sets a network bearer for a specific IP address or domain name
  - partner privileged

- **Secure Element**
  - Provides functionality to communicate with applications in several secure elements, such as UICC/SIM, embedded Secure Element, or Secure SD card
  - partner privileged

- **Push**
  - Receives push notifications from the Tizen push server
Content APIs (1/2)

• **Content**
  - Discovers and manages images, video, music, and other files
  - Scans content or directory metadata in the device and updates the content database
  - Retrieves content from content database using filters
  - Browses content by getting a list of content directories
  - Views and edits content item details
Content APIs (2/2)

• **Download**
  - Downloads files from a specific URL
  - Enables applications to manage download operation details
    • Sets the stored location of a downloaded file
    • Sets the network type: cellular, Wi-Fi, or default
    • Get operation status: current received size and events
    • Gets the file MIME type
Input/Output APIs

• Filesystem
  • Accesses the file system virtual root locations
    • Documents, downloads, images, music, videos, ringtones, wgt-package, wgt-private, and wgt-private-tmp
  • Manages file storage
    • Internal and external
    • Mounted, removed, or unmountable
  • Accesses files and directories
  • Creates, reads, edits or deletes files and directories

• Message Port
  • Communicates with other applications: IPC based on application ID
Social APIs (1/3)

• **Contact**
  • Manages contacts in device address books
    • Including adding, searching, updating, and deleting
  • Manages persons
    • Person is the aggregation of one or more contacts that are the information of the same person
  • Supports vCard 3.0
Social APIs (2/3)

• Calendar
  • Manages events and tasks
    • Including adding, searching, updating, and deleting
    • Each event or task has a series of attributes, such as purpose, starting time, and duration
  • Monitors the changes in events and tasks
  • Supports iCalendar v2.0, based on RFC 5545

• Call History
  • Browses the call history of a device
  • Removes call history entries
  • Monitors changes
Social APIs (3/3)

<table>
<thead>
<tr>
<th>Social</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookmark</td>
<td>new</td>
</tr>
<tr>
<td>Manages bookmarks and bookmark folders</td>
<td></td>
</tr>
<tr>
<td>Browses bookmark folders</td>
<td></td>
</tr>
<tr>
<td>platform privileged</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data Synchronization</th>
<th>new</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronizes device data to the server using the OMA DS 1.2 protocol</td>
<td></td>
</tr>
<tr>
<td>Contact data or calendar data</td>
<td></td>
</tr>
<tr>
<td>Manages the OMS DS profile slots</td>
<td></td>
</tr>
<tr>
<td>Tizen platform sets a limitation on the number of supported profiles</td>
<td></td>
</tr>
<tr>
<td>Adds, updates and removes profiles</td>
<td></td>
</tr>
</tbody>
</table>
System APIs (1/2)

• **Time**
  - Provides TZDate type that is an extended type of ECMAScript date
    - It contains the time zone information as well as time information
  - Provides utility functions for managing system time and duration
    - Methods for getting local time or time zone
    - Methods for calculating time duration

• **System Information**
  - Enables access various properties of the system
    - Bluetooth, NFC, Wi-Fi, Front/back Camera, Flash, GPS, Sensors, Platform and API version, USB host/accessory
  - Monitors the change of system information
System APIs (2/2)

- **System Setting**
  - Sets or gets the system setting values
    - `HOME_SCREEN`: Home screen background image
    - `LOCK_SCREEN`: Lock screen background image
    - `INCOMING_CALL`: incoming call ringtone
    - `NOTIFICATION_EMAIL`: email notification alert tone

- **Power**
  - Manages the power state for the screen resource
    - Sets or gets the state of the screen: off, dim, normal, or bright
    - Monitors the changes in the state of the screen
  - Sets the CPU not to sleep
User Interface APIs

- Notification
  - Posts UI notifications about application events

![Notification Example](image)