Android Sample project

To implement the Android library into your project, you must first download it from the DynConD portal and then extract the Android library into your project's folder.

The minimum required version for the Android library to run is Android 6.0, API level 23.

For the Android library to be functional, we need Internet so the first and foremost thing to do is to grant internet and wifi permissions in the AndroidManifest.xml file

After we have granted internet permission, we can import the Android library into our project in two different ways (depending on the version of the Android studio that you are using).

The first way is to Go to the top menu and click File → New → New Module → Import .JAR/.AAR Package and then go Next → Find DynConD Android library file and click Finish

The second way is to go to the top menu and click on File → Project Structure → Dependencies + → JAR/AAR Dependency → type in the path of the Android library folder and press Ok

It is important to note that in the last part of the build.gradle script there must be an implementation of the specified library, and if it is not there, it is necessary to add it and synchronize the gradle.

For a successful call of DynConD's **DynConDInetAddress** class, *import dyncond.dyncondlib.DynConDInetAddress* statement must be placed into the activity.

DynConD's **DynConDInetAddress** class is a replacement for the standard InetAddress class and has the same basic methods and results as the InetAddress class. If no IP address is returned by the **DynConDInetAddress** class, the standard non DynConD DNS A/AAAA resolving can be used.

DynConD's **DynConDInetAddress** class by default uses local Android host DNS resolvers. By using the **getAllByNameADNS** method with a defined service value of "100", authoritative DynConD DNS servers (ADNS) are used instead of local Android host DNS resolvers. This way, the real-time DSS parameters are obtained from servers, avoiding using cached values in the process whose accuracy depends on the TTL value of the TXT RRs. If no IP address is returned when using the ADNS query, the getAllByNameADNS class automatically performs a standard recursive DNS query after the Timeout period defined in TXT RR.

DynConD Android library implementation guide

- Download the DynConD Android library from http://dyncond.com/downloads/
- For both, new or existing Android studio projects, extract the DynConD Android library into the project folder
- DynConD Android library requires a minimum API level 23 (Android 6.0 - Marshmallow)
- In the AndroidManifest.xml, permissions for network (Internet) and WiFi must be set as:

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- In the Android Studio, go to File -> New -> New Module -> Import .JAR/.AAR Package -> Next -> Find DynConD Android library file -> Finish
- In the application's *build.gradle* use *dependencies* to implement the library as follows:

```
implementation project(":DynConD_Library")
```

- For a successful call of the DynConDInetAddress function set

```
import dyncond.dyncondlib.DynConDInetAddress;
```

in Activity

- Call DynConDInetAddress function in Activity as follows:

```
DynConDInetAddress dynConDInet = new
DynConDInetAddress(getApplicationContext());
InetAddress[] dyncond = DynConDInetAddress.getAllByName(…);
or
InetAddress[] dyncond = DynConDInetAddress.getAllByNameADNS(…);
or
InetAddress[] dyncond = DynConDInetAddress.getbyName(…);
```

- DynConD's ***DynConDInetAddress*** class is a replacement for the standard *InetAddress* class and has the same methods, results and error codes as the *InetAddress* class

iOS Sample project

To implement the iOS framework into your project, you must first download it from the DynConD portal and then extract the iOS framework into your project's folder.

The minimum required version for the iOS framework to be fully functional is iOS 10.0

Open the project editor in the project navigator, select Target, and click on the + sign in Frameworks, Libraries, and Embedded Content. In the Framework search window, click on Add Other -> Add Files and select DynConD.xcframework

By clicking on the Open button, DynConD.xcframework will be included in the project and will be visible in Frameworks, Libraries, and Embedded Content. On the right side of the included framework select Embed & Sign option, if it's not already marked

For a successful call of DynConD's dyncondgetaddrinfo function, import DynConD statement must be placed into the controller (ViewController)

DynConD's dyncondgetaddrinfo function is a replacement for the standard getaddrinfo function and has the same parameters, results, and error codes as getaddrinfo function. If no IP address is returned by the dyncondgetaddrinfo function, the standard non DynConD DNS A/AAAA resolving can be used

DynConD's dyncondgetaddrinfo function by default uses local iOS host DNS resolvers. By using the dyncondgetaddrinfo function with a defined service value of "100", authoritative DynConD DNS servers (ADNS) are used instead of local iOS host DNS resolvers. This way, the real-time DSS parameters are obtained from servers, avoiding using cached values in the process whose accuracy depends on the TTL value of the TXT RRs. If no IP address is returned when using the ADNS query, the dyncondgetaddrinfo function automatically performs a standard recursive DNS query after the Timeout period defined in TXT RR

DynConD iOS framework implementation guide

- Download the DynConD iOS framework from http://dyncond.com/downloads/

- For both new or existing Xcode projects, extract the DynConD iOS framework into the project folder

- Open the project, click on Target, open General tab and click on the + sign in Frameworks, Libraries and Embedded Content. In the Framework search window, click on Add Other -> Add Files and select DynConD.xcframework

- By clicking on the Open button, DynConD.xcframework will be included into the project and will be visible in Frameworks, Libraries and Embedded Content. On the right side of the included framework select Embed & Sign option, if it's not already marked

- For a successful call of DynConD's dyncondgetaddrinfo function, import the DynConD statement and it must be placed into the controller (ViewController)

- Calling the DynConD's dyncondgetaddrinfo function:

let status = DynConDGAI().dyncondgetaddrinfo(...)

- DynConD's dyncondgetaddrinfo function is a replacement for the standard getaddrinfo function and has the same parameters, results and error codes as getaddrinfo function

Load Agent installation procedure on Linux server

1. Download dyncond.conf (DynConD Load Agent configuration file) from https://my.dyncond.com/User/LoadSetupList -> Get configuration file

2. Download Load Agent Linux binary from https://my.dyncond.com/User/LoadSetupList or https://my.dyncond.com/Downloads/Downloads

3. Copy dyncond.conf to /etc directory

Example /etc/dyncond.conf file:

UserID:user

Credentials:password

NS:ns1.dyncond.net

NS:ns2.dyncond.net

NS:ns3.dyncond.net

NS:ns4.dyncond.net

IP:198.51.100.100

IPP:172.16.100.100

Refresh:3

CPU:1

RAM:1

LogFile:1

Explanation of dyncond.conf file:

UserID: your username provided by DynConD portal, cannot be changed by the user

Credentials: your username provided by DynConD portal, cannot be changed by the user

NS: DynConD name servers to which data is sent

IP: server public IP address

IPP: optional server private IP address, if multiple servers share the same public IP (cluster)

Refresh: refresh interval for sending data to DynConD NS

CPU: if 1 CPU data is sent, if 0 CPU data is not sent

RAM: if 1 RAM data is sent, if 0 RAM data is not sent

LogFile: if sent data is logged to /var/log/dyncond/agent_info.log, if 0 sent data is not logged

4. Copy Load Agent dyncondagent to /usr/bin directory

5. Create dyncondagent service:

- create service configuration file:

/etc/systemd/system/dyncondagent.service

[Unit]

Description=DynConD agent service

After=network.target

StartLimitIntervalSec=0

[Service]

Type=simple

Restart=always

RestartSec=1

User=root

ExecStart=/usr/bin/env /usr/bin/dyncondagent

[Install]

WantedBy=multi-user.target

- start dyncondagent service:

systemctl start dyncondagent

- enable dyncondagent service at server boot:

systemctl enable dyncondagent

Created symlink /etc/systemd/system/multi-user.target.wants/dyncondagent.service → /etc/systemd/system/dyncondagent.service.

- check status of dyncondagent service:

systemctl status dyncondagent

● dyncondagent.service - DynConD agent service

Loaded: loaded (/etc/systemd/system/dyncondagent.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2021-06-12 10:24:04 CEST; 7min ago

Main PID: 62877 (dyncondagent)

Tasks: 1 (limit: 2340)

Memory: 668.0K

CGroup: /system.slice/dyncondagent.service

└─62877 /usr/bin/dyncondagent

Jun 12 10:24:04 test2.localdomain systemd[1]: Started DynConD agent service.

Jun 12 10:24:04 test2.localdomain env[62877]: AGENT STARTED

5. DynConD can use two log files, info and error log files

- info log example, used if LogFile:1 value is set in dyncond.conf:

/var/log/dyncond/agent_info.log

2021-06-12 10:24:04.157 AGENT STARTED

2021-06-12 10:24:04.167 IPV4 address: 75.119.133.81 (ns1.dyncond.net)

2021-06-12 10:24:04.366 IPV4 address: 45.61.53.180 (ns2.dyncond.net)

2021-06-12 10:24:04.524 IPV4 address: 52.78.162.64 (ns3.dyncond.net)

2021-06-12 10:24:04.688 IPV4 address: 159.89.232.248 (ns4.dyncond.net)

2021-06-12 10:24:07.708
UserID:user;Credentials:password;IP:198.51.100.100;CPU:10;RAM:30

2021-06-12 10:24:07.708
UserID:user;Credentials:password;IP:198.51.100.100;CPU:10;RAM:30

2021-06-12 10:24:07.709
UserID:user;Credentials:password;IP:198.51.100.100;CPU:10;RAM:30

2021-06-12 10:24:07.709
UserID:user;Credentials:password1;IP:198.51.100.100;CPU:10;RAM:30

- log file contains information about Load Agent service start time, resolved IP addresses of DynConD Name servers and information about all data sent to every IP address

- error log example:

/var/log/dyncond/agent_error.log

2021-06-12 10:21:45.305 config file not exists