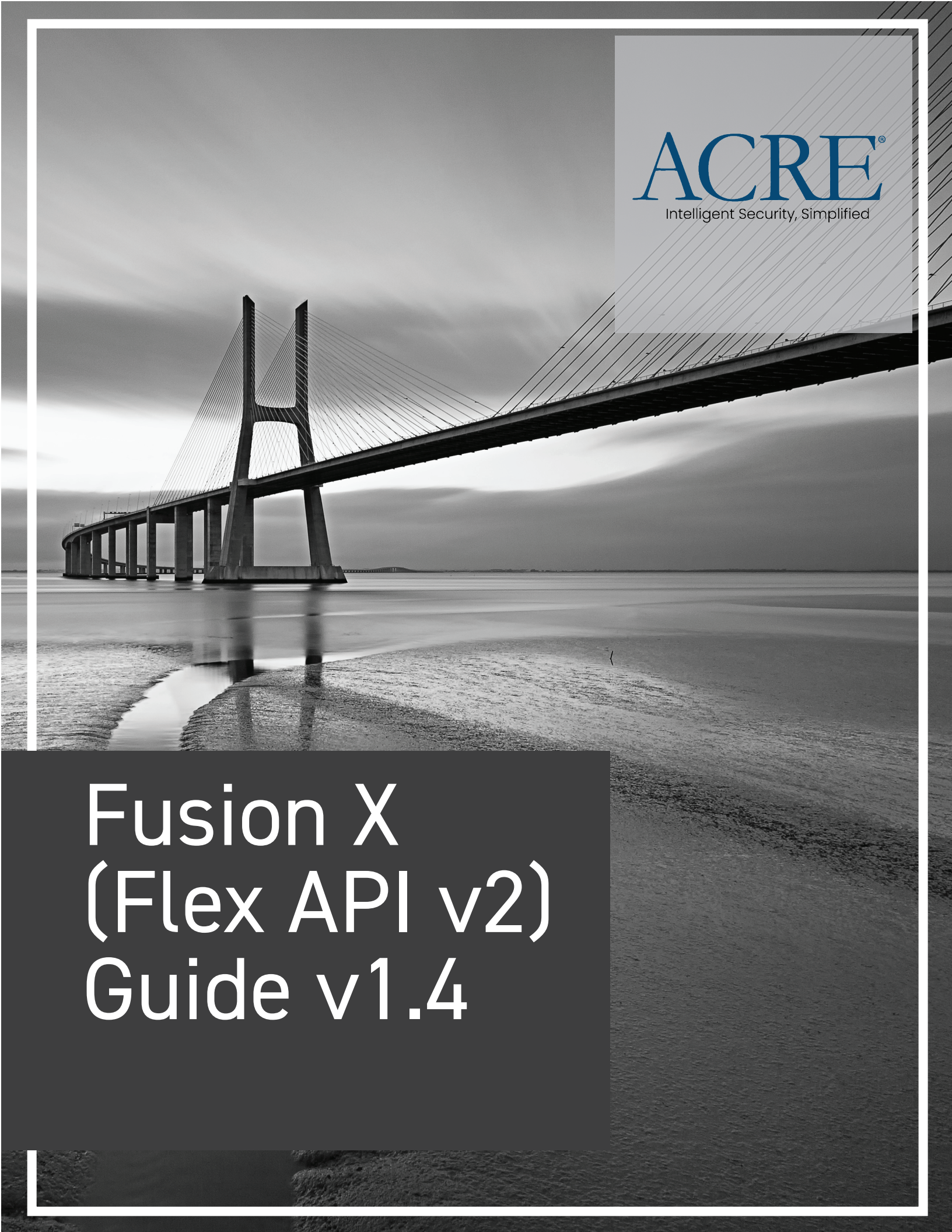


Fusion X (Flex API v2) Guide v1.4

This manual is proprietary information of ACRE, LLC. Unauthorized reproduction or distribution of this manual is strictly forbidden without the written consent of ACRE, LLC. The information contained in this manual is for informational purposes only and is subject to change at any time without notice. ACRE, LLC. assumes no responsibility for incorrect or outdated information that may be contained in this publication.

DNA Fusion™ and SSP™ are trademarks of ACRE, LLC.

The DNA Fusion™ Access Control Software and SSP™ Security System Processor use equipment that generates, uses, and radiates radio frequency energy. If not installed and deployed in accordance with the guidelines of this installation manual, they may cause harmful interference to radio communications. Operation of this equipment in a residential area may cause harmful interference, in which case the user will be required to correct the interference at their own expense.

The DNA Fusion™ Access Control Software and SSP™ Security System Processor shall be installed in accordance with this installation manual and in accordance with the National Electric Code (N.E.C), ANSI and NFPA 70 Regulations and recommendations.

Publish Date: May. 2022
Manual Number: FusionX(FlexV2API)v1.2

© Copyright 2002-2022 ACRE, LLC. All rights reserved.

Warranty

All ACRE products are warranted against defect in materials and workmanship for two years from the date of shipment. ACRE will repair or replace products that prove defective and are returned to ACRE within the warranty period with shipping prepaid. The warranty of ACRE products shall not apply to defects resulting from misuse, accident, alteration, neglect, improper installation, unauthorized repair, or acts of God. ACRE shall have the right of final determination as to the existence and cause of the defect. No other warranty, written or oral is expressed or implied.

ACRE®
Intelligent Security, Simplified
16650 Westgrove Dr | Suite 150
Addison, TX 75001
Phone: (972) 818-7001
Fax (972) 818-7003
https://acre-co.com/

ACRE NA Software License Agreement

THE ENCLOSED SOFTWARE PACKAGE IS LICENSED BY ACRE NA. TO CUSTOMERS FOR THEIR NON-EXCLUSIVE USE ON A COMPUTER SYSTEM PER THE TERMS SET FORTH BELOW.

DEFINITIONS: ACRE NA shall mean ACRE NA, which has the legal right to license the computer application known as DNA Fusion herein known as the Software. Documentation shall mean all printed material included with the Software. Licensee shall mean the end user of this ACRE NA Software. This Software Package consists of copyrighted computer software and copyrighted user reference manual(s).

LICENSE: ACRE NA, grants the licensee a limited, non-exclusive license (i) to load a copy of the Software into the memory of a single (one) computer as necessary to use the Program, and (ii) to make one (1) backup or archival copy of the Software for use with the same computer. The archival copy and original copy of the Software are subject to the restrictions in this Agreement and both must be destroyed or returned to ACRE NA if your continued possession or use of the original copy ceases or this Agreement is terminated.

RESTRICTIONS: Licensee may not sub license, rent, lease, sell, pledge or otherwise transfer or distribute the original copy or archival copy of the Software or the Documentation. Licensee agrees not to translate, modify, disassemble, decompile, reverse engineer, or create derivative works based on the Software or any portion thereof. Licensee also may not copy the Documentation. The license automatically terminates without notice if Licensee breaches any provision of this Agreement.

TRANSFER RIGHTS: Reseller agrees to provide this license and warranty agreement to the end user customer. By installation of the software, the end user customer and reseller agree to be bound by the license agreement and warranty.

LIMITED WARRANTY: ACRE NA warrants that it has the sole right to license the Software to Licensee. Upon registration by the Licensee, ACRE NA further warrants that the media on which the Software is furnished will be free from defects in materials and workmanship under normal use for a period of twelve (12) months following the delivery of the Software to the Licensee. ACRE NA's entire liability and your exclusive remedy shall be the replacement of the Software if the media on which the Software is furnished proves to be defective. EXCEPT AS PROVIDED IN THIS SECTION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN PARTICULAR, EXCEPT AS PROVIDED IN THIS SECTION, WITH RESPECT TO ANY PARTICULAR APPLICATION, USE OR PURPOSE, LICENSOR DOES NOT WARRANT THAT THE PRODUCTS WILL MEET THE LICENSEE'S REQUIREMENTS, THAT THE PRODUCTS WILL OPERATE IN THE COMBINATIONS OF 3RD PARTY SOFTWARE WHICH THE LICENSEE MAY SELECT TO USE, OR THAT THE OPERATION OF THE PRODUCTS WILL BE UNITERRUPTED OR ERROR FREE. NEITHER OPEN OPTIONS, NOR ITS VENDORS SHALL BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF USE, INTERRUPTION OF BUSINESS, NOR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND WHETHER UNDER THIS AGREEMENT OR OTHERWISE. IN NO CASE SHALL OPEN OPTIONS' LIABILITY EXCEED THE PURCHASE PRICE OF THE SOFTWARE.
The disclaimers and limitations set forth above will apply regardless of whether you accept the Software.

TERMINATION: ACRE NA may terminate this license at any time if licensee is in breach of any of its terms or conditions. Upon termination, licensee will immediately destroy the Software or return all copies of the Software to ACRE NA, along with any copies licensee has made.

APPLICABLE LAWS: This Agreement is governed by the laws of the State of Texas, including patent and copyright laws. This Agreement will govern any upgrades, if any, to the program that the licensee receives and contains the entire understanding between the parties and supersedes any proposal or prior agreement regarding the subject matter hereof.

Flex V2 API w/ Fusion X User Guide Part A

In Part A

- FusionX Flex V2 API Requirements
- FusionX Flex V2 API Installations
- Configuring Fusionx Flex V2 API
- Creating and Importing Certificates
- Adding Site Bindings
- Creating API Client Key
- Host Name Information

Introduction

Flex v2.2+ is the latest API w/ Fusion X for DNA Fusion version 8+.

Flex v2 API w/ FusionX Requirements

Server Requirements

• .NET 5.0 Framework

• Must be installed on the DNAFusion server.

• Windows 10+, Windows Server 2012 R2, Windows Server 2016 or Windows Server 2019.

Flex v2 API w/ FusionX Installation

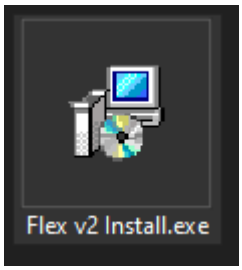
Flex v2 API w/ FusionX is meant to be used in conjunction with DNA Fusion v8+. The Installation will preform pre-checks as outlined in the installation in the following page. If the installation detects any previous version it will update any necessary components that may be out of date after stopping the services . Please contact DNA Fusion customer support if you experience any issues or problems in installing the latest version of Flex v2 w/ FusionX.

Note: You will need to be logged in as an Administrator prior to installing the Flex API.

Installing the Latest version of Flex v2 w/ FusionX

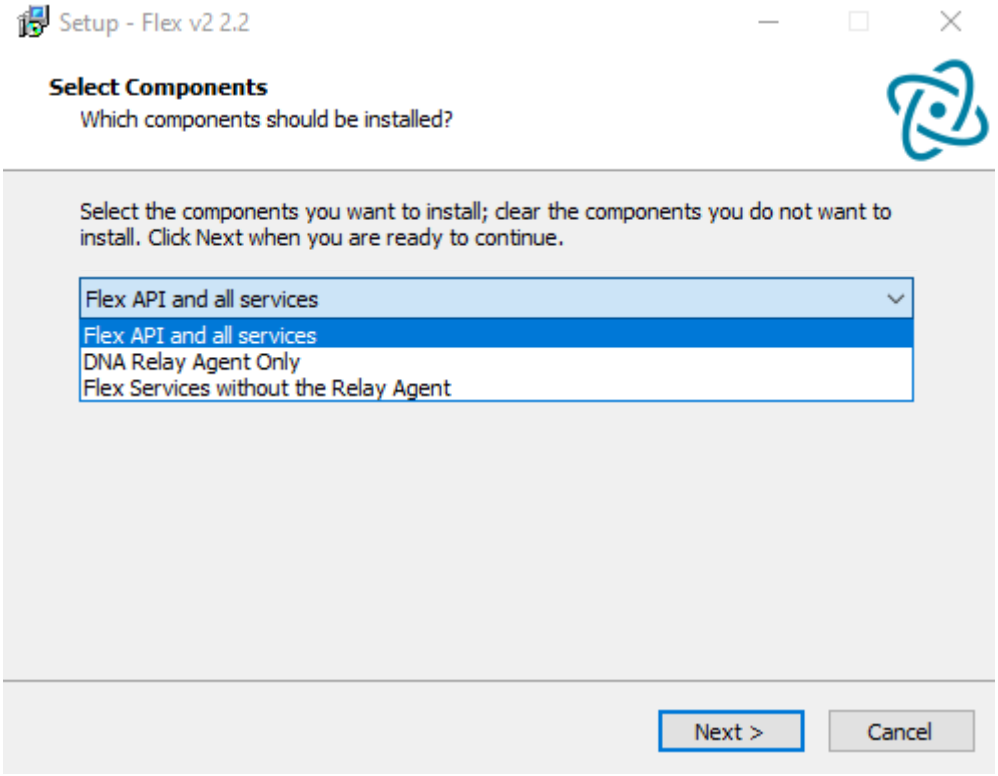
You will need to start by **Downloading** the latest version of the Flex v2 API, the latest version can be found by visiting the Acre Knowledge Base at:

kb.ooaccess.com/en

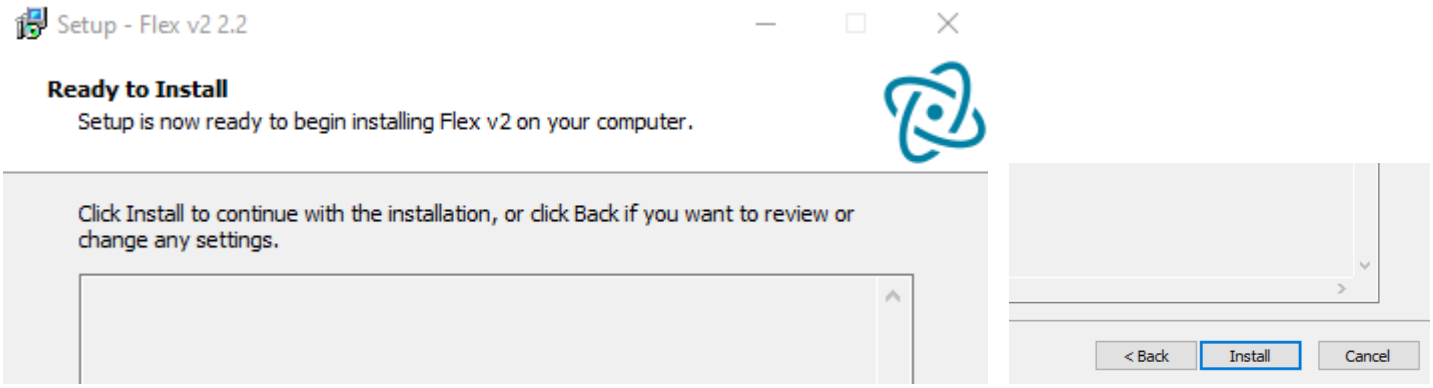


1. **Double-Click** the Installation file once it has been downloaded.

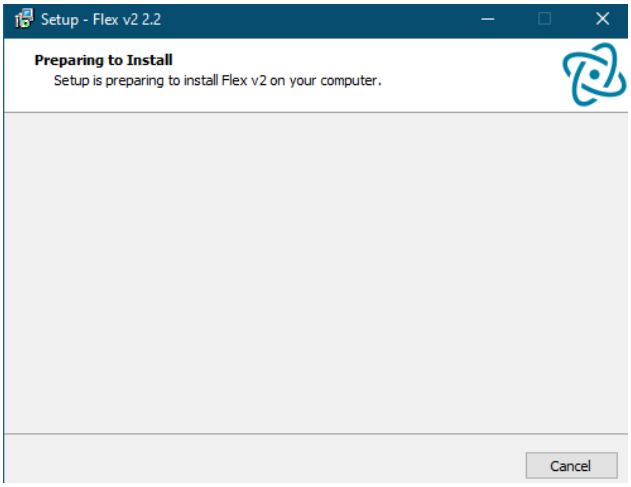
2. **Click** Next on the initial installation screen once you have selected the components that are right to you.

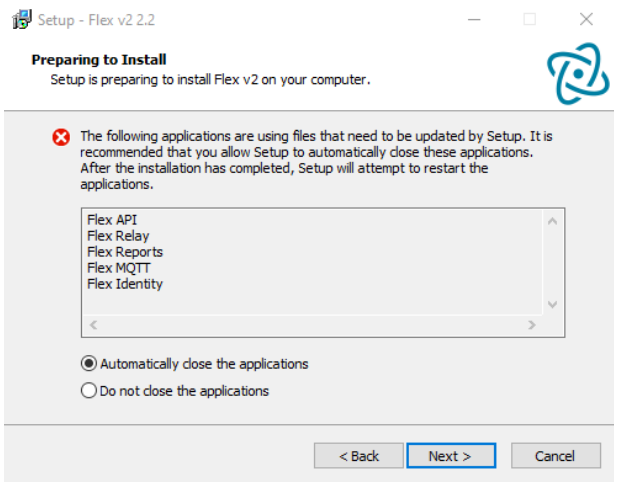


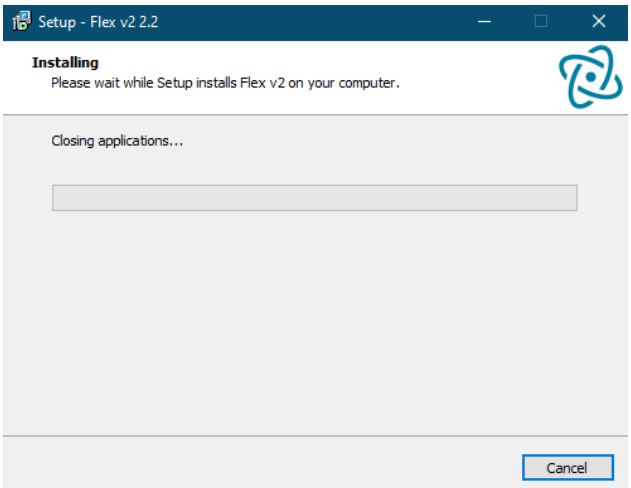
2. **Click** the Install button on the next screen to begin the installation. The Install file will begin by detecting your DNA Fusion Installation environment.

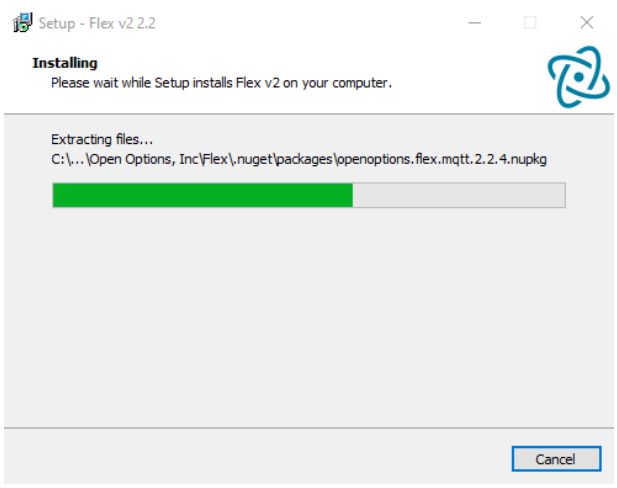


If the Installation detects any prior installations of the Flex API, the install will prompt you to confirm that you want to update the installation. Otherwise the installation will begin to install the latest version of the Flex v2 API w/ FusionX.

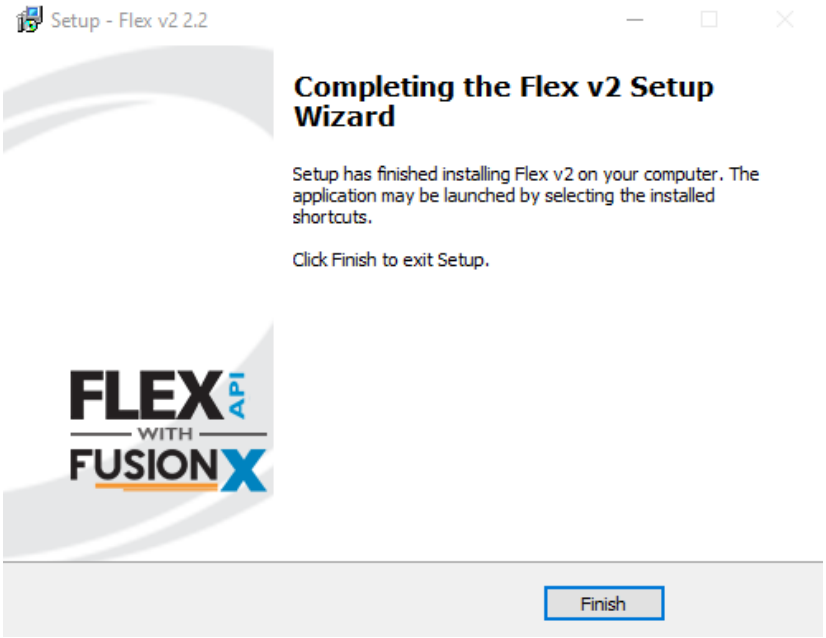








3. **Click** Finish Once completed to finalize the installation.



Setting File Uses:

Settings.json

- Has shared settings and all v2 API related settings
- Contains database connection
- Authority URL
- Bindings
- Application specific logging

Identity.settings.json

- See Part B for Formatting Identity Settings

Reports.settings.json

- Application specific logging
- If you restart, "Flexible", and select, "Flex API", you will be able to see the default settings.

Once completed, a service restart is required for all changes to go into effect.

Connection String

Connection String should be built from the DNA Reports DSN if not specified in

> **Note**: Settings Must be manually saved via, "File>Save", Menu Bar Item.

the, *settings.json,* file. If not properly configured you will be able to override it through the Connection String settings.

Authority URL

Authority URL is the IRL that the Flex API will redirect for authentication. This in turn must be hard-coded to a valid URL that the bindings setting will accept.

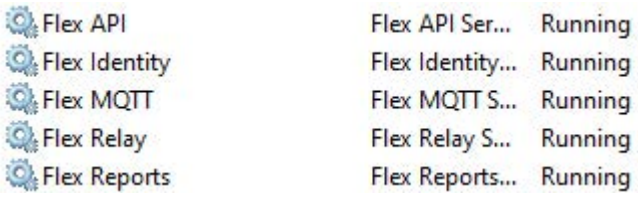
Bindings

Similar to Flex API v1, the bindings are a semi-colon separated binding lists. For additional information regarding the bindings, since the API is built and functions using readily available .NET web format, you can find many base and more advance commands by searching for .NET API binding lists.

Flex Services:

Running Services after Installation

Once you have completed the installation, there is a requirement for the initial services to be started in order to run the Flex v2 API. The services are shown below:



You will need to initiate the services by **Right-Clicking** on the service, and **Selecting** *Start* or by **Clicking** on the service and **Selecting** *Start* from the upper left part of the *Services* window.

Flex Configuration Guide Part B

This Guide will provide some insight into the working processes that help run the Flex v2 API, FlexID, and FusionX application. While this guide is not intended to assist with any one topic in particular, it allows for the solving of several issues and assists in shedding some light on various features.

In Part B
• Settings.Json FileConfiguration • JSend Message Structure • Alarm Processing • Flex Identity OAuth • Flex API MQTT Broker • Fusion X Code Samples

Note: In this style of box, notes or concerning matters will be inserted to ensure that all pertinent notes are completed in an orderly manner allowing for important Items to stand out.

Settings

General Overview

The Settings files provides an outline for general operation of the Flex API when it comes to the settings for user login and site location. The focus of this first part will be on the identity.settings.json file that can be opened as a text file and adjusted to configure a variety of the settings in the Flex v2 API w/ FusionX.

Identity.settings.json File

1. **Open** the "Identity.Settings.json" file which can be found at the following location.

C:\Users\Public\Documents\Open Options, Inc\Flex

Please note that the location may change based on the drive selected as the installation location for your specific configuration.

2. **Double-Click** to open the file using the Microsoft Notepad application.

The File settings should be as shown below:

```
{
  "signingCertificate": {
    "useTemporaryKey": true
  },
  "SmartInspect": {
    "Level": "Debug",
    "File": {
      "Filename": "Flex.Identity.sil"
    },
    "Pipe": {
      "Reconnect": {
```

```
      "Enabled": true,
      "Interval": "1s"
    }
  },
  "Tcp": {
    "Level": "Message",
    "Reconnect": {
      "Enabled": true,
      "Interval": "1s"
    }
  }
 },
 "claim": {
  "includeRights": false
 },
 "WsFederation": {
  "DisplayName": "Island Rentals",
  "MetadataAddress": "https://*islandrentals.local/
FederationMetadata/2007-06/FederationMetadata.xml:443/",
  "Wtrealm": "https://islandrentals.local/"
 },
 "Google": {
  "DisplayName": "Island Rentals",
  "ClientId": "IslandRentals",
  "ClientSecret": "password"
 }
}
```


While these settings will need to be configured to your local network requirements, they can be used as a baseline for understanding what is needed to make the Flex v2v API's outward facing or web based functions work.

Configuring the Identity.Settings File

When dealing with a Flex v2 API w/ FusionX System that refuses to launch, the first step in troubleshooting will be to check the "Authority" settings in the "Settings.json" file.

The Following can be used as a template for formatting the ".json" files.

EXAMPLE configuration Firewall Port Forwarding Public IP to Private IP

Firewall IP from | to		Port from | To	
HTTP	PUBLICIP>PRIVATEIP	80>80	
HTTP	PUBLICIP>PRIVATEIP	443>443	HTTPS

"Identity.settings.json" Section:

},

"RedirectUris": [

"https://HOSTNAME",

"http://HOSTNAME",

"https://PRIVATEIP"

]

}

In this example, the "Authority" settings would need to be adjusted based on the requirements for a secure IP configuration using port 443 and Port 80 for non secure functions. When using the HTTPS or secured functions of the web browser, the "Settings.Json" file would need to be configured to allow for a certificate to be used or not used in the validation process for HTTPS as shown below:

If using personnel self-signed certificate

"validateCertificate": "False"

//

"urls": "http://*:80;https://*:443",

"ConnectionStrings": {

"DefaultConnection": ""

},

"authority": {

"url": "https://PUBLICIP/identity",

"validateCertificate": "False"

},

From inside the network you would also use the "PUBLICIP"

HTTP://PUBLICIP

HTTPS://PUBLICIP

Companion Application

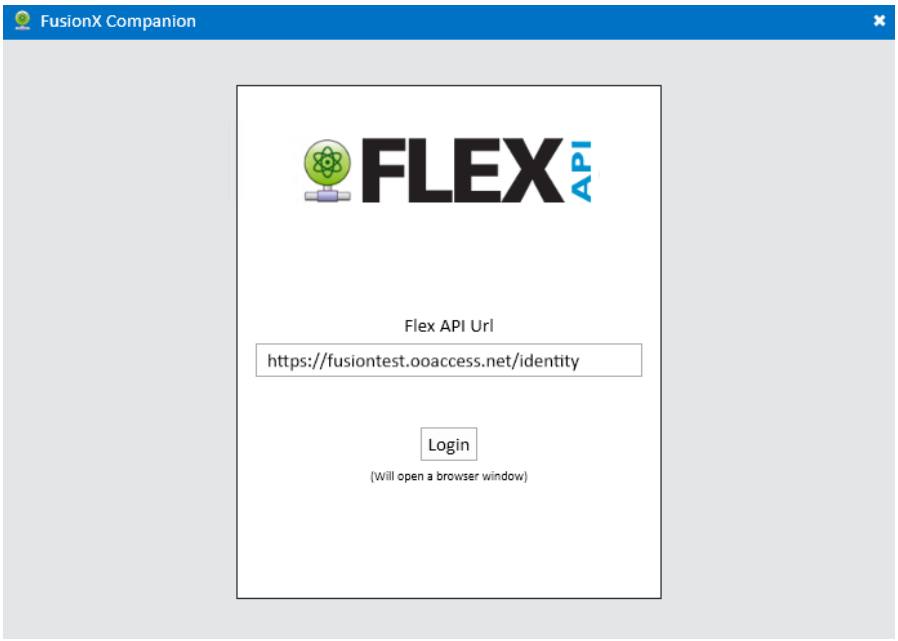
Initial Setup of the companion app will require the modification of the Settings Json file. You can test the default configuration using the following address:

https://fusiontest.ooaccess.net/identity

While you can attempt to connect to it using the base address configuration found in the Settings.Json file located at:

C:\Users\Public\Documents\Open Options, Inc\Flex.Companion\Settings.json

Initial modification of the address settings may be required to conform to your local network and system naming and networking conventions.



{

“Badging”: {

“ProxComPort”: 4

“iClassComPort”: 3

“SmartCardComPort”: 2,

“iClassTimeOut”: 10000,

“SmartCardTimeOut”:10000,

“ProxTimeOut”: 10000,

“ProxCardFormat”: 1,

“ProxReaderType”: 0,

“BitsOnCard”: 26,

“FacilityCodeOffset”: 1,

“FacilityCalcBits”: 8,

“CardNumberBitOffset”: 9,

“CardNumberBits”: 16,

“Printer”: “\\\\oofiles\\Toshiba ES3055C PS3”.

“PrinterType”: 0,

“PrinterStation”: 0,

“CardMask”: null,

“Parity1Mask”: null,

“Parity2Mask”: null,

“Parity3Mask”: null,

}

“Bioscrypt”: {

“ComPort”: 3,

“DeviceId”: 0,

“BaudRate”: 115200,

“IpAddress”: “”,

“ConnectByNetwork”: true

},

“HandKey2” : {

“ComPort”: 3,

“BaudRate”: 9600,

“IpAddress”: “”,

“ConnectByNetwork”: true

},

“Identity”: {

“Authority”: “https://fusiontest.ooaccess.net/identity”,

“RefreshToken”: null,

“Expires”: “0001-01-01T00:00:00”,

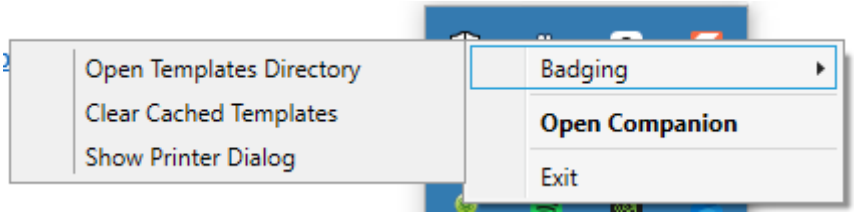
“ValidateEndpoints”: true,

“ValidateIssuerName”: true

}

}

Additionally, any changes made may require a clearing of the companion app cache. This may also be performed to fore a re-download of essential data that is not updating on the system.



SAML Information

Operator Profiles

The Flex v2 API has the ability to house multiple users based on the DNA Fusion Operator settings. These can be found by going to:

DNA>Administrative>Operator Maintenance>Operator Privileges...

Once there you can edit the operator email address and information tied to the Flex API Service.

FusionX SAML2

Once the Operator email is properly set to the correct one, you will now need to have the operator log into FusionX.

1. Click on the Upper Right-Hand corner and select “Profile”.
2. Scroll to the bottom of the page.
3. Click the “Remove” button for the SAML2 extension.
4. Log out of FusionX.
5. Close Browser.

This Page Intentionally Left Blank

JSend

General Overview

JSend is the FlexV2API's method of sending pertinent information regarding DNA Fusion and user activities within the program to the FlexAPI. While this is not a new method of retrieving information within the realm of API's (or Application Program Interfaces), it is essential for the function and features within the Flex v2 API.

JSend General Operations

The JSend aspect of the Json files used to run the Flex API in both V1 and V2 use a general format for the issuing of "Status" information and "Data" to be returned or ferried between DNA Fusion and the application to be displayed in a variety of mediums to include the web page interface and the HHD (Hand Held Device) application. Below is an example of the communications format.

```
{
   "status": "{ success | fail | error }",
   "data": "{ a wrapper for any data to be returned }"
}
```

While the data format is very simple, it is essential for the transporting of information. Next we will go over how successful, Failed and Error files look like.

JSend Success

In the following examples we will be using a command for calling a list of "Company" records from the SQL table in FlexV2API with FusionX.

Command: ***GET*** http(s)://localhost/api/v2/companies

Return File:


```
{
   “pageNumber”: 1,
   “pageSize”: 3,
   “totalNumberOfPages”: 1,
   “totalNumberOfRecords”: 2,
   “status”: “success”,
   “data”: [
      {
         “address1”: “12345 Adams Lane”,
         “city”: “Addition “,
         “country”: “USA”,
         “name”: “Software Security Experts Exchange”,
         “state”: “TX”,
         “uniqueKey”: 3,
         “zipcode”: “75254”
      },
      {
         “address1”: “12345 New Address”,
         “address2”: “Suite 234”,
         “name”: “Chess Enterprises”,
         “uniqueKey”: 4
      }
   ]
}
```


As indicated after “status”, the request was “success”(fully) completed and returned two records from a single page followed by the number of records.

JSend Fail

In this instance, there will be a failure to return information resulting from there not being an 11th file in the “companies” table in the SQL database.

GET http://localhost/api/v2/companies/11

{
“status”: “fail”,
“data”: “Unable to locate members record with the unique key of 11”
}

JSend Error

The last instance will request the deletion of a non-existent record resulting in an error message.

DELETE http://localhost/api/v2/companies/11

{
“status”: “error”,
“message”: “Could not delete company record with the unique key of 11 due to the following error: System.Exception: Unexpected delete error.”
}

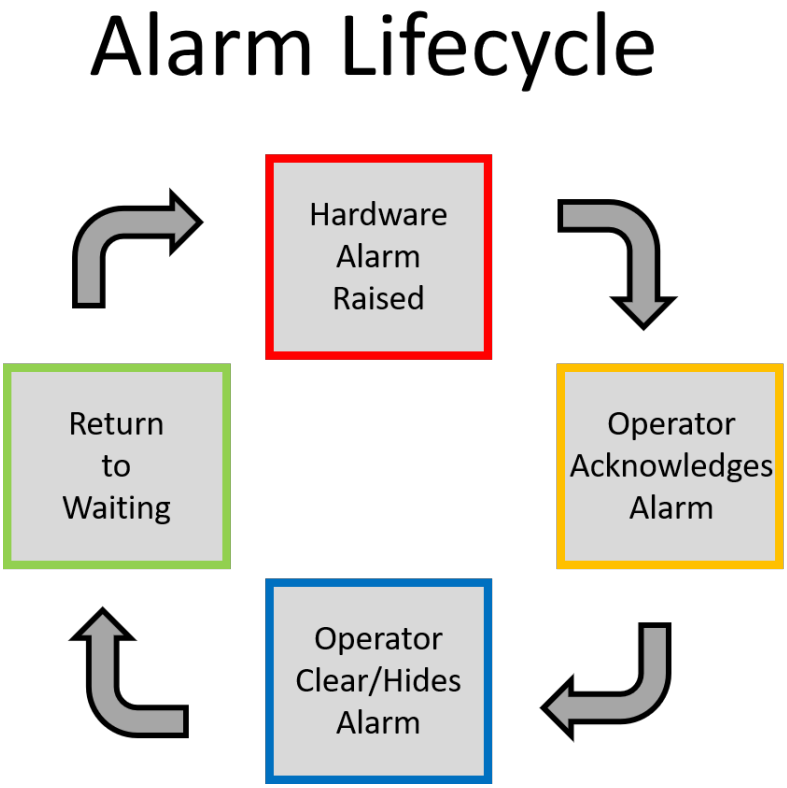
While the last two instances no actual information was returned, there was data still returned to display a message detailing the nature of the “status” message.

Alarm Processing

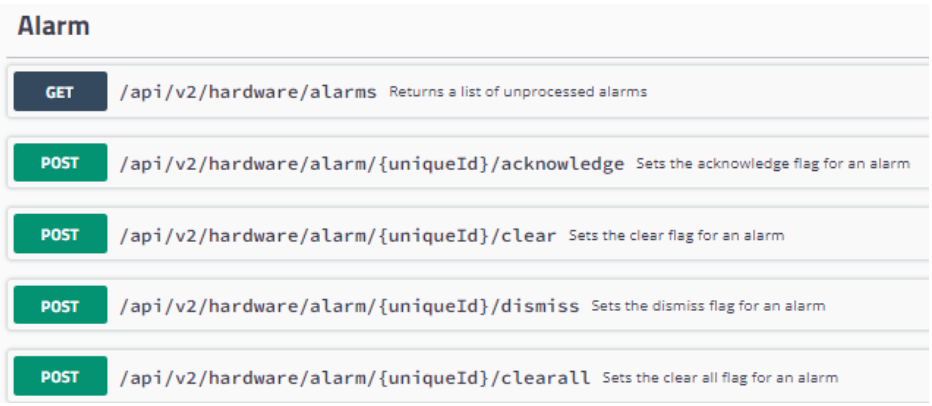
The Flex v2 API w/ Fusion X has the capability of alerting you whenever there are alarms or events that come up in DNA Fusion. While this is a great feature to take advantage of for remote detection and alerting to possible significant events happening on your security grids there are settings that must be put in place prior to the events replicating in the Flex v2 API and alerting logged in users of events. and alarms.

Alarm Life Cycle

Below is a diagram displaying the typical alarm life-cycle in the Flex API



In order to be able to process alarms in the Flex API, you will need to first configure the Alarm Hardware API as the example shown below.



The Alarm Hardware API is a capable and flexible utility within the Flex API. Additionally you can use the Alarm Processing project code shown below to test the functionality for demonstration purposes.

```
g Common;       You, a week ago • FusionX sample code; update comments in DataObjec…
g System.Text;
g System.Text.Json.Nodes;

ble.WriteLine("Alarm Processing");
ble.WriteLine("------------------------");
ble.WriteLine("\tThis sample will demonstrate how to process alarms by:");
ble.WriteLine("\t\t1. Getting an alarm");
ble.WriteLine("\t\t\t read alarmId");
ble.WriteLine("\t\t\t read hardwareState");

ble.WriteLine("\t\t2. Acknowledging an alarm");
ble.WriteLine("\t\t3. Clearing an alarm");
ble.WriteLine("------------------------");

ble.WriteLine("");

client = await FlexApiExtensions.GetHttpClientViaResourceOwnerPassword();

ble.WriteLine("\tSTEP 1) Getting alarms ....");

OTE: If you do not have any active alarms, you will get an error when you try to access the a

response = await client.GetAsync($"{FlexApiExtensions.GetBaseUrlFromConfig()}/hardware/alarms
ng content = await response.Content.ReadAsStringAsync();

item = JsonObject.Parse(content);
```

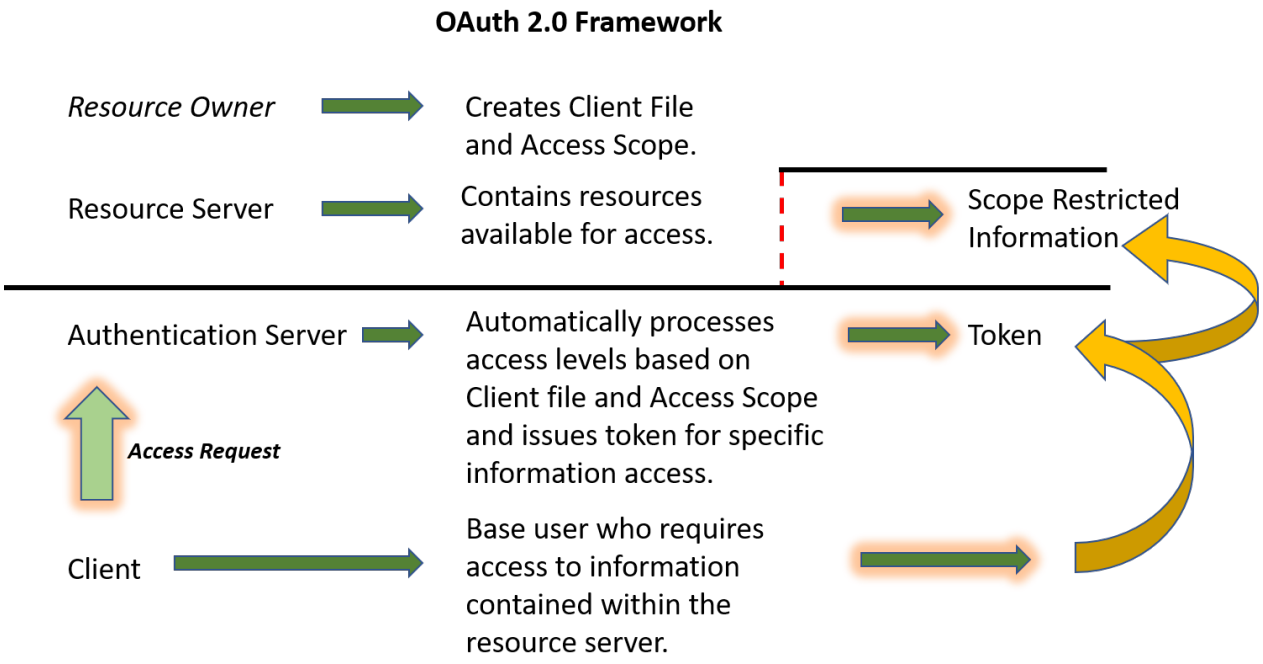
Note: You must have alarms setup in the DNA System in order for the Alarm Processing project code to work.

FlexID

General Overview

FlexID is a program built into the Flex API that allows for authorized access to individuals that is separate from the Owner (Resource Owner) or Admin account equivalent to the DNA Server housing the information populated within DNA Fusion. This is accomplished through the use of the OAuth 2.0 Framework. While it is essentially an authentication method for access control to resources contained within the server accessible by the Flex API, it does so through the issuing of credential "Tokens" by the "Authorization Server" on behalf of the "Resource Owner". The allocated resources are given to the client by the "Resource Owner" through the use of allocated "Scopes" within the client file in FlexID.

OAuth

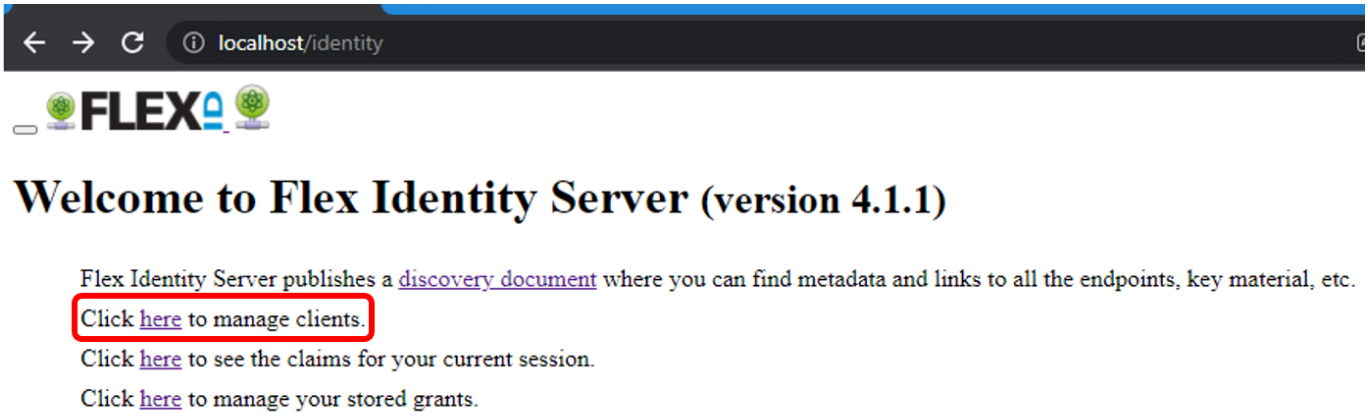


This figure shows the standard flow of information within the OAuth 2.0 Framework.

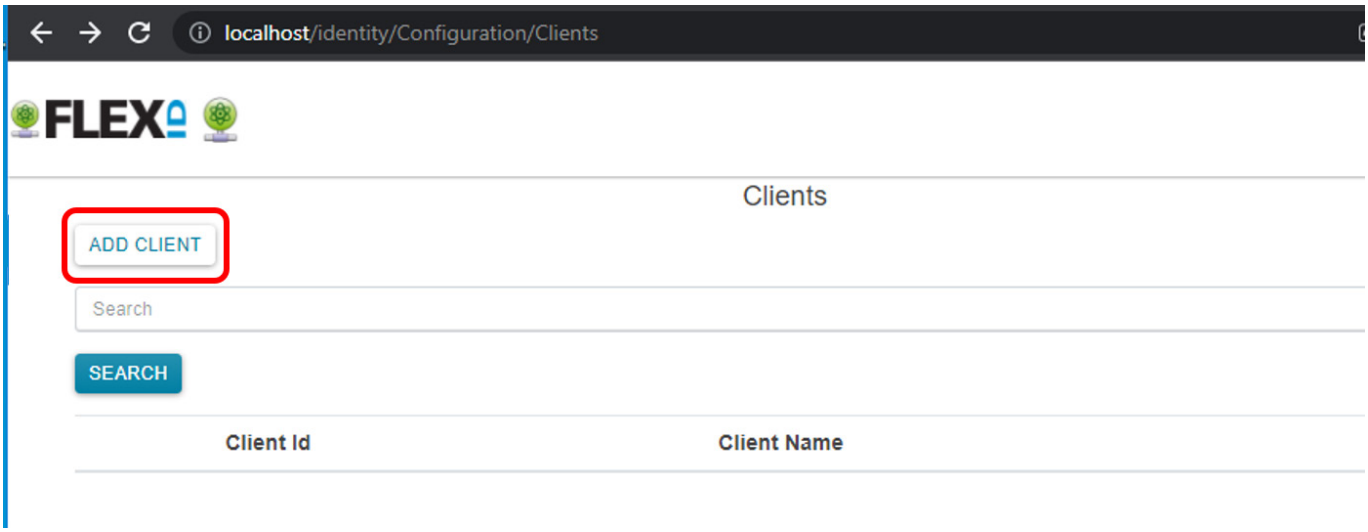
Flex ID Server

In order to set up the Flex ID Server you will first need to set up clients. This can be accomplished by using the following steps.

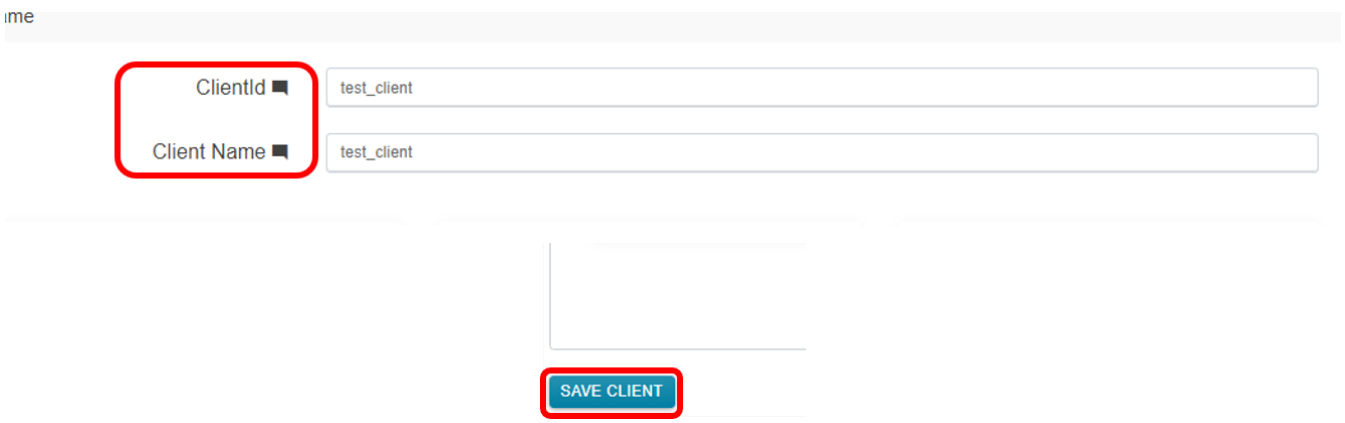
1. Open the Flex Identity Server.
2. Click on the “Here” on the “Manage Clients” line.



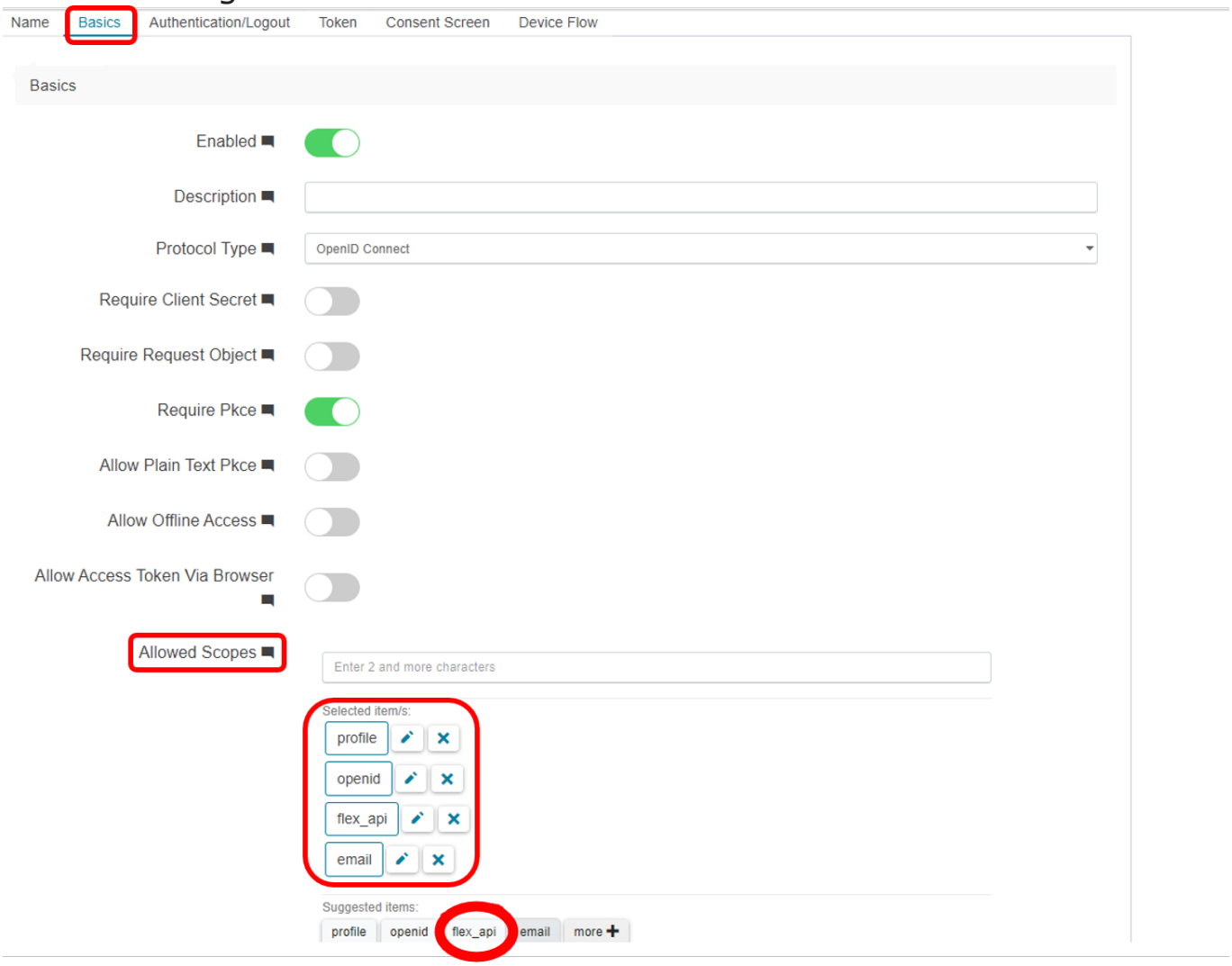
3. Click “Add Client” on the next screen.



4. Enter the “Client ID” and the “Client Name” and Click “Save Client”.

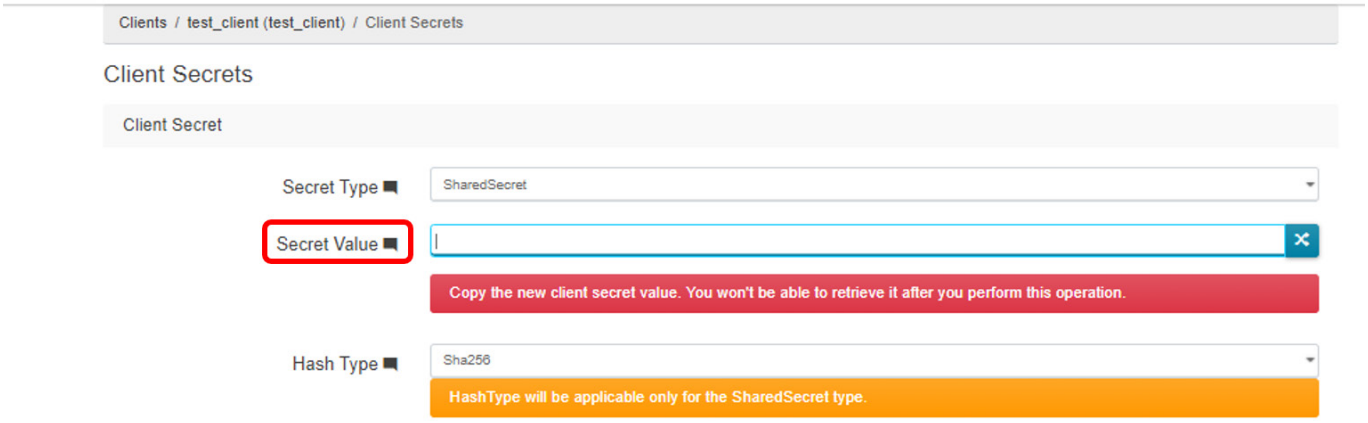


5. Click “Basics” on the tabs at the top of the next screen once the client has been created. You will need to define the “scope”(s) of their access and save the settings.



6. Click the “Basic” tab on the site bar and scroll down and click on “Manage Client Secrets”.

7. Create a "Secret Value" and take note of it in a private file. DO NOT SHARE THE SECRET VALUE. A random value generator such as a "cryptographically secure pseudo-random number generator" can be used to create this value.



8. Click "Add Client Secret" once completed to save the generated secret value.

As the "Resource Owner" you will need to create each clients secret. The client secret is a phrase used to create the key used in the encryption and decryption of the information within the system. The characters in the phrase are meant to be used by the encryption operations or signatures to create sufficient entropy or randomness to create a strong key.

FlexAPI MQTT Broker

MQTT Broker

The MQTT Broker is a system integrated in the Flex API that allows for the interaction through TCP/IP and Web-Sockets to a variety of features located within DNA. These can include alarms, events and status updates. In order to perform this you will first need to "Subscribe" to events, status and alarm updates. In order to do this you can follow the simple format of:

flex/{source system}/{hardware type}/{unique key}/{alarm /event /status}

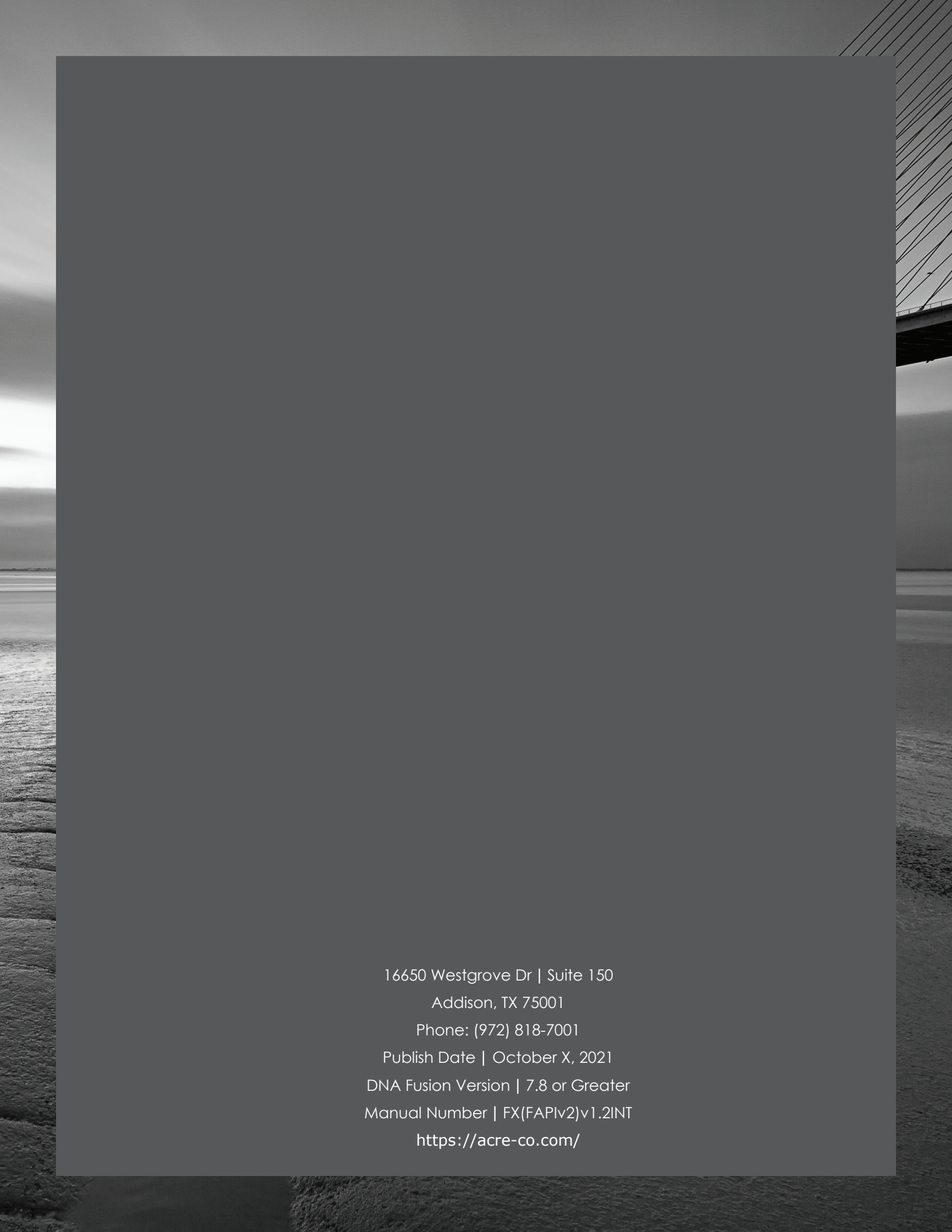
The process is completed by inserting of the various wildcards needed to make the system work. Some examples of these various wildcards and how they are formated has been shown below.

WILDCARD TYPE	FLEX LINE SAMPLE CODE
all alarms	flex/+/+/+/alarm
all events	flex/+/+/+/event
all status	flex/+/+/+/status
specific hardware unique key	flex/+/+/a20fe7a3-d842-4ba2-b43a-fa234f87aa0f/+
a type of hardware	flex/+/Door/+/+

Once you have connected to the MQTT broker and when messages are raised and received, you can receive them in your client handler and process them. The “Json” data will be provided within the “ApplicationMessage”. From the “Payload” you will be able to read any messages transmitted and perform any additional processes required. A sample of the event “Payload” data is shown below.


```
{
   “CameraId”:0,
   “CardholderKey”:null,
   “GSACredential”:””,
   “Display”:true,
   “EventDescriptionId”:2,
   “FirstName”:””,
   “HardwareAddress”:”1.5.D4”,
   “HardwareDescription”:”ACM 4”,
   “HardwareType”:”0:Door”,
   “HardwareUniqueKey”:”a20fe7a3-d842-4ba2-b43a-fa234f87aa0f”,
   “LastName”:””,
   “CardNumber”:0.0,
   “FacilityCode”:0,
   “Panel”:”2022-01-07T04:04:25Z”,
   “Transaction”:”2022-01-06T22:04:25Z”,
   “TransactionData”:”(4) Door: 0x01 (was 0x01), AccPt: 0x14 (was 0x04)”,
   “TransactionId”:0
}
```


The MQTTMessage product samples are readily available online at places such as MQTTnet for ease of use and formating.



16650 Westgrove Dr | Suite 150
Addison, TX 75001
Phone: (972) 818-7001
Publish Date | October X, 2021
DNA Fusion Version | 7.8 or Greater
Manual Number | FX(FAPIv2)v1.2INT
https://acre-co.com/