Location Intelligence
Infrastructure Asset Management

# Confirm®

## Confirm Configuration Service

v26.30a.AM

Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor or its representatives. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, without the written permission of Confirm.

© 2026 Confirm. All rights reserved.

Products named herein may be trademarks of their respective manufacturers and are hereby recognized. Trademarked names are used editorially, to the benefit of the trademark owner, with no intent to infringe on the trademark.

**Open Source Attribution Notice**

The Confirm suite of products contain the following open source software:

- Feature Data Objects v 3.5.0, which is licensed under GNU Lesser General Public License, Version 2.1, February 1999 with the unRAR restriction. The license can be downloaded from: **http://fdo.osgeo.org/licenceAndGovernance.html**. The source code for this software is available from **http://fdo.osgeo.org/content/fdo-350-downloads**
- MrSID software (specifically the mrsid32.dll) is used under license and is Copyright © 1995-2002, LizardTech, Inc., 1008 Western Ave., Suite 200, Seattle, WA 98104. All rights reserved. MrSID is protected by U.S. Patent No. 5,710,835. Foreign patents are pending. Unauthorized use or duplication prohibited.

  Patented technology in the Software was developed in part through a project at the Los Alamos National Laboratory, funded by the U.S. Government and managed by the University of California. The U.S. Government has reserved rights in the technology, including a non-exclusive, nontransferable, irrevocable, paid-up license to practice or have practiced throughout the world, for or on behalf of the United States, inventions covered by the patent, and has other rights under 35 U.S.C. § 200-212 and applicable implementing regulations.

  For further information, contact Lizardtech.

- NodaTime, version number 1.3.10, which is licensed under the Apache license, version number 2.0. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0 . The source code for this software is available from http://nodatime.org/.
- Chromium Embedded Framework, version 3, which is licensed under the New BSD License. The license can be downloaded from http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://code.google.com/p/chromiumembedded/downloads/list.
- Xilium.CefGlue, version 3, which is licensed under the MIT License (with portions licensed under the New BSD License). The licenses can be downloaded from http://opensource.org/licenses/MIT and http://opensource.org/licenses/BSD-3-Clause. The source code for this software is available from http://xilium.bitbucket.org/cefglue/.
- D3 Data Driven Documentation, version 3.4.1, which is licensed under the New BSD License. The license can be downloaded from from https://github.com/mbostock/d3/blob/master/LICENSE. The source code for this software is available from http://d3js.org/.
- OpenLayers, version 8.1, which is licensed under the BSD 2-Clause Licence. The license which can be downloaded from https://github.com/openlayers/openlayers/blob/master/LICENSE.md The source code for this software is available from https://github.com/openlayers/openlayers.
- Proj4js, version 1+, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://trac.osgeo.org/proj4js/.
- requireJS, version 2.1.2, which is licensed under the MIT License or the New BSD License. The license can be downloaded from https://github.com/jrburke/requirejs/blob/master/LICENSE. The source code for this software is available from http://requirejs.org/.
- Apache Cordova, version 11.1.0, which is licensed under the Apache License, Version 2, January 2004. The license can be downloaded from http://www.apache.org/licenses/LICENSE-2.0.html. The source code for this software is available from http://phonegap.com/download/.
- Xilium.CefGlue, version 75.1, which is unlicensed. The source code for this software is available from https://gitlab.com/xiliumhq/chromiumembedded/cefglue.

- Chromium Embedded Framework, version 75.0, which is licensed according to the following criteria:

  Copyright (c) 2008-2014 Marshall A. Greenblatt. Portions Copyright (c) 2006-2009 Google Inc. All rights reserved.

  Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

  * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

  * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

  * Neither the name of Google Inc. nor the name Chromium Embedded Framework nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

  THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

  The source code for this software is available from http://opensource.spotify.com/cefbuilds/index.html#

October 02, 2026

# Table of Contents

**Confirm Configuration Service**

Installation, Configuring and Upgrading the Confirm Configuration Service 6

Configuring and Using the Amazon Web Service Key Management 14

Configuring and Using Active Directory User Management 18

Using Confirm Configuration Service from Confirm Components 19

FAQ 20

MFA FAQ 21

# Confirm Configuration Service

Confirm's Identity and access management can be made more secure by enabling two step isolation and encryption of database and third party passwords. The first step isolates encryption to a newly created Confirm Configuration Service. This enables IT administrators and security personals to have centralized control over the encryption mechanism. The second step gets all encryption done by Amazon Web Services' Key Management Service which makes the system highly secure.

Migration to this enhanced security is optional but recommended. Also, users can choose to migrate only to the isolated Confirm Configuration Service or further on to Amazon Web Service encryption, which will need access to AWS over the Internet all the time.

## In this section


Installation, Configuring and Upgrading the Confirm Configuration Service 6
Configuring and Using the Amazon Web Service Key Management 14
Configuring and Using Active Directory User Management 18
Using Confirm Configuration Service from Confirm Components 19
FAQ 20
MFA FAQ 21

# Installation, Configuring and Upgrading the Confirm Configuration Service

### Prerequisites

Ensure that the following prerequisites are installed prior to configuring the Confirm Configuration Serivce.

- If installing the Web Service on a 32 bit system please refer to the IIS Configuration Required for 32 bit Systems page for more details.
- Ensure that Web Server IIS role is installed.
- Visual C++ 2010 redistributable.
- .Net Framework 4.8

**Note:** It is a best practice, and highly recommended, to use HTTPS deployment for Confirm Configuration Service.

### Installation

Run the following steps to install Confirm Configuration Service:

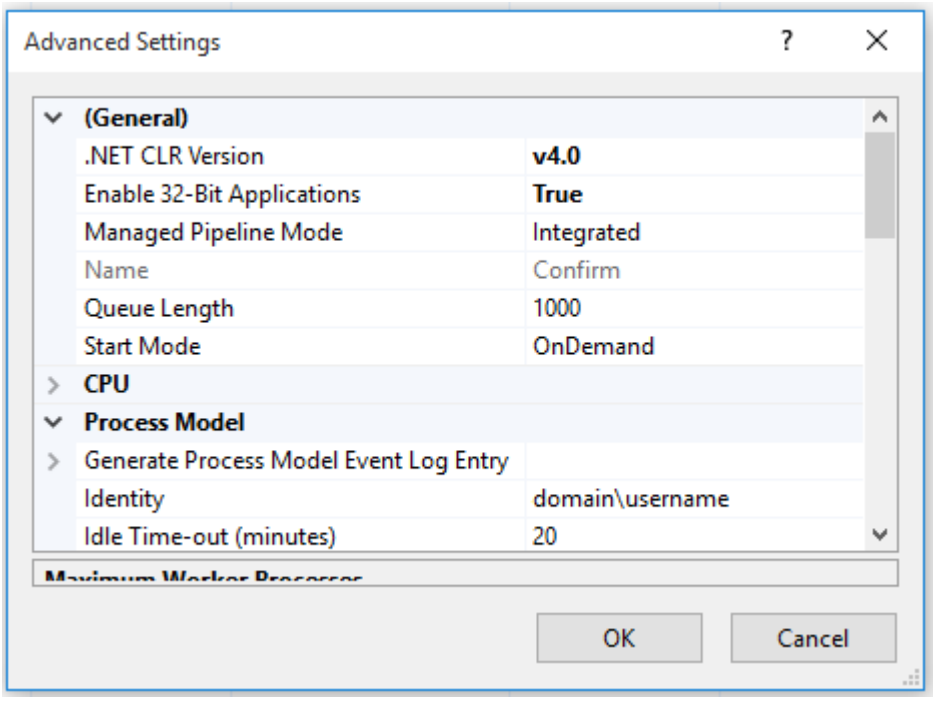
- Run the ConfigurationService.exe and specify a location to install the web service. The default location is set to C:\inetpub\wwwroot\confirm\ConfigurationService.

### Configuring an Application Pool

It is advisable to create a new application pool for the Confirm Configuration Service. If setting up more than one service, ensure that a unique application pool is created for each instance.

To create an application pool, go to the IIS Manager and right click on the Application Pools option and select Add Application Pool. Ensure that the following settings are applied to the new application pool.

| | |
|---|---|
| **.Net Framework** | Set to 'v4.0' |
| (Applicable for 64 bit machines only)<br>**Enable 32-Bit Applications** | Set to ‘True’ (This is set in the ‘Advanced Settings'.). |
| **Managed Pipeline** | Set to 'Integrated'. |
| **Identity** | Set to a domain user. |
| **Load User Profile** | Set to 'True'. |



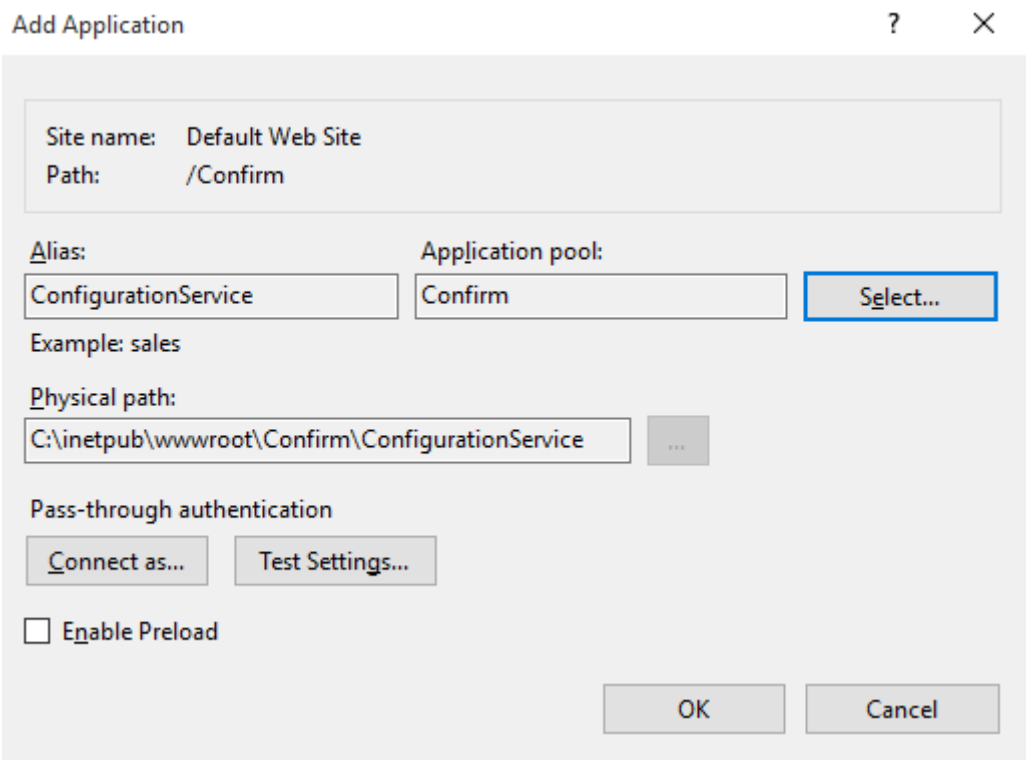

Note: It is recommended to use a domain user and set the password policy such that its password never expires. Otherwise, the password should be updated every time it is changed failing which the Application Pool will get disabled and cause Confirm Configuration Service to stop.

**Creating the Web Service Application**

Open the Internet Information Services (IIS Manager) and navigate to Sites. Then right click on Default Web Site and select Add Application.

The Add Application window will appear.

- Set the Alias to the name you would like for your Web Service.
- Set the Application pool to the application pool initially created.
- Set the Physical Path as the folder where the Web Service files are stored.




- Finally click on OK to create the application.

### Registry Configuration

Following registry entry are to be configured on the web server where Confirm Configuration Service is installed:

| Registry Location | Key | Description |
|---|---|---|
| *32 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Confirm\ConfigurationService<br>*64 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Confirm\ConfigurationService | LoggingEnabled | Enable logging on or turn it off (Yes/ No). Default set to 'No'. |
| | LogFile | Path and Filename of the log file. |
| | KeyManagementEnabled | To determine whether to use External Key Management Service (Yes/ No). |
| | KeyManagementTool | External Key management tool to be used. Set to 'AWS' |
| | SamlConfigFile | Provide the path of SAML config file. Required only in case of Single sign-on for ConfirmWeb. |
| | AdConfig | Stores the path of the Active directory configuration file. Enable/Disable Mixed Mode. |
| | UserStoredInActivedirectory | To determine whether a user stored in Active Directory (Yes/No). |

### SAML Configuration

In case of Single Sign-on, additional SAML Information needs to be configured in saml configuration file.

To create a saml configuration file, copy and paste the following text in text editor:

```
<?xml version="1.0" encoding="UTF-8"?>
<SAMLConfigurations xmlns="urn:componentspace:SAML:2.0:configuration">

 <!-- Below section is for Confirm web interface -->
  <SAMLConfiguration xmlns="urn:componentspace:SAML:2.0:configuration"
ID="{Database Id}">
    <ServiceProvider Name="{ACS URL}" AssertionConsumerServiceUrl="{ACS
URL}" />
    <PartnerIdentityProviders>
      <!-- You can get below information from your Identity Provider.
Please refer to your Identity Provider setup instructions for more details
 -->
      <PartnerIdentityProvider SignAuthnRequest="false"
        Name="{Identity Provider Issuer}"
        Description="{Email Domain}"
        SingleSignOnServiceUrl="{Identity Provider Single Sign-On URL}"
        PartnerCertificateFile="{X.509 Certificate file path}"/>

    </PartnerIdentityProviders>
  </SAMLConfiguration>

 <!-- Below section is for ConfirmConnect -->
  <SAMLConfiguration xmlns="urn:componentspace:SAML:2.0:configuration"
ID="Connect">
    <ServiceProvider Name="{ACS URL}" AssertionConsumerServiceUrl="{ACS
URL}" />
    <PartnerIdentityProviders>
      <PartnerIdentityProvider SignAuthnRequest="false"
        Name="{Identity Provider Issuer}"
        Description="{Email Domain}"
        SingleSignOnServiceUrl="{Identity Provider Single Sign-On URL}"
        PartnerCertificateFile="{X.509 Certificate file path}"/>

    </PartnerIdentityProviders>
  </SAMLConfiguration>

</SAMLConfigurations>
```

Populate the saml configuration file as below:

| Parameter | Value |
|---|---|
| PartnerIdentityProvider SignAuthRequest | If set to True, provide a value for ServiceProvider LocalCertificateSubject. |
| ServiceProvider Name | Replace {ACS URL} with hosted ConfirmWeb URL suffixed by /api/ProfileName/samllogin<br>For e.g.: https://confirm.com/confirm/web/api/ProfileName/samllogin |
| ServiceProvider AssertionConsumerServiceUrl | Same as above |

| Parameter | Value |
|---|---|
| ServiceProvider LocalCertificateSubject | Optionally provide the certificate name to digitally sign the outgoing SAML requests to the Identity provider.<br>For e.g.: "CN=My Secure Certificate Name"<br>The certificate can be generated by running the 'PushCertificate.bat' utility.<br>Steps to populate this field:<br>**1.** 'PushCertificate.bat' file can be found in Resources folder within Confirm media.<br>**2.** Run the PushCertifcate.bat and enter a Certificate Name in the prompt.<br>**3.** Use this certificate name as value here. |
| PartnerIdentityProvider Name | Replace {Identity Provider Issuer} with Identity Provider Issuer Name<br>For e.g.: http://www.identityprovider.com/ek5v5y0355 |
| PartnerIdentityProvider Description | Replace {Email Domain} with email domain that needs to be configured for single sign-on<br>For e.g.: @confirm.com |
| PartnerIdentityProvider SingleSignOnServiceUrl | Replace {Identity Provider Single Sign-On URL} with Single Sign-On URL for your Identity Provider.<br>For e.g.: https://identityprovider.com/app/confirm/ek5v5y0355/sso/saml |
| PartnerIdentityProvider PartnerCertificateFile | Replace {X.509 Certificate file path} with the location of your Certificate file. |
| ID | {Database Id} or {Connect}<br>• Replace {Database Id} with the database Id specified in the database configuration file in case of Confirm Web Interface.<br>• Use Connect in case of ConfirmConnect. |

Save the saml configuration file created above as 'saml.config'. The path of saml configuration file needs to be specified as the value for 'SamlConfigFile' in registry.

**Note:** If there is a space in the Profile Name, replace the space with %20.

For e.g. the profile name is 'Client - Development', it should be 'Client%20-%20Development'.

### Auth0 Configuration

In case of Single Sign-on, additional Auth0 Information needs to be configured in OIDC configuration file.

To create a OIDC.json file, copy and paste the following text in text editor:

```
{
        "domain": "{Domain Name}",
        "applications": [
```

```
    {
       "application_type": "web",
       "client_id": "client_id",
       "client_secret": "client_secret",
      "consumer_service_url": "http://localhost/confirm/confirmweb/api/oid-
clogin"
    },
    {
      "application_type": "connect",
      "client_id": "{client_id}",
      "client_secret": "{client_secret}",
      "consumer_service_url": "http://localhost/ConnectorWebService/Con-
firmMobileService.svc/idplogin"
    }
  ],
      "organizations": [
    {
      "org_name": "{organization_name}",
      "org_id": "{organization_id}",
      "profileName": "{Profile_name}",
      "connections": [
        {
          "domain": "{@xyz.com}",
          "connection": "{connection}"
        },
        {
          "domain": "{@abc.com}",
          "connection": "{connection}"
        }
      ]
    },
    {
      "org_name": "{organization_name}",
      "org_id": "{organization_id}",
      "profileName": "{Profile_name}d",
      "connection": "{connection}",
      "connections": []
    }
  ]
}
```

Populate the OIDC.json file as below:

| Parameter | Value |
|---|---|
| Domain | In Auth0, the "domain" refers to the unique URL that identifies your Auth0 tenant or account. |
| client_id | The "client_id" is used by the client application to identify itself to the Auth0 authorization server when making requests for authentication, authorization, or resource access. |
| client_secret | The "client_secret" is a confidential piece of information known only to the client application and the authorization server.<br><br>It is used to authenticate the client application when it requests tokens from the authorization server. |
| organization_name | An Organization in Auth0 represents a logical container that can include one or more tenants. Each tenant is a separate and isolated environ- |

| Parameter | Value |
|---|---|
| | ment, often representing a distinct application or project. |
| organization_id | Id can be obtained on the creation of an organization in auth0 under the Organization tab. |
| consumer_service_url | • Redirect URI is used to redirect the user back to the application after they log in and authorize the application.<br>• It's important to note that the Redirect URI must be a valid URL registered in the Auth0 Dashboard for your application.<br>• You need to whitelist the Redirect URI to prevent unauthorized redirects and ensure that Auth0 only redirects to specified valid URLs.<br>• To whitelist the Redirect URI we can use Allowed the callback URLs section of Auth0 under application. |
| Connection | Connections allow us to setup federated login. We can setup database connections or Enterprise connections (SAML, OIDC, AD).<br>For e.g.: @nda.com |
| Profile_name | This corresponds to the profile name of tenant configured in database setting. |

Save the Auth0 configuration file created above as 'oidc.json'. The path of OIDC file needs to be specified as the value for 'ProfilePath' in registry.

**Note:** If there is a space in the Profile Name, replace the space with %20.

For e.g. the profile name is 'Client - Development', it should be 'Client%20-%20Development'.

### Configuring Mixed Mode Authentication

Mixed mode is a combination of Active Directory(AD) and Confirm authentication i.e. AD users will be authenticated using their AD credentials and non-AD users will be authenticated using their Confirm credentials

A configuration file is required to enable Mixed mode. Please follow steps below:

- Open a text editor and create a new file.
- Copy and paste the text below this section into the file
- Save the file.
- Store the full path of the file in the registry key AdConfig (please refer to the section **Registry Configuration** in this document).

```
{"AuthProviderType":"Mixed"}
```

**Note:** The possible values for AuthProviderType are as below:

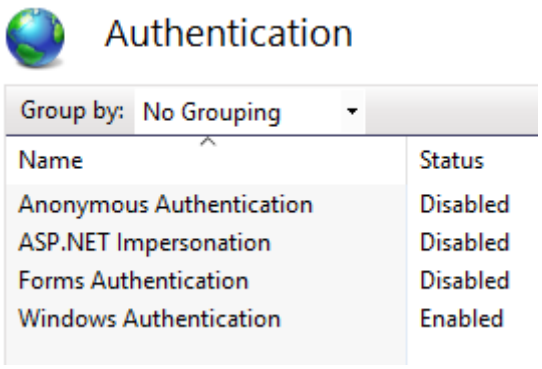
- Confirm - To Use Confirm Authentication.
- ActiveDirectory - To Use ActiveDirectory.
- Mixed - To Use both Authentication.

**Note:** Once the Mixed Mode authentication is enable, it will be applicable to all the Components( Confirm web, Connect, Connector) as the AdConfig file is shared file.

### Access Control and Authentication

Follow the steps below to setup the access control and authentication for Confirm Configuration Service:

- Open Internet Information Services Manager.
- Select the Confirm Configuration Web Site/Application.
- In Features View, double click Authentication.
- Enable Windows Authentication and disable Anonymous Authentication.




If Windows Authentication is not available there, you can install it by going to Add Roles and Features within Server Manager.

- Restart the Web Service.

All Confirm components need to run under a domain user to use Confirm Configuration Service. The details of this can be found under section **Using Confirm Configuration Service**.

Additionally, it is recommended to restrict the access to Confirm Configuration Service to a specific domain group.

### High Availability

It is recommended that Confirm Configuration Service is deployed in clustered environment to handle system failures.

### Health Check

To check the successful configuration of Confirm Configuration Service browse the URL

'https://<IP Address: Port Number>/api/healthcheck'

after replacing 'IP Address' and 'Port Number' with actual values from IIS.

The response returned should be:

```
<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Confirm Configuration Service working correctly.</string>
```

If the response differs from above then reverify the configuration steps and refer to the **FAQ Section**.

# Configuring and Using the Amazon Web Service Key Management

Once the Confirm Configuration Service has been installed and configured successfully, IT administrators can choose to use Amazon Web Service's Key Management Service to encrypt and decrypt all database and 3rd party passwords within Confirm. This will make the system access highly secure.

**Note:** Once migrated to Amazon Web Service, Confirm will always use it for encryption/ decryption of database & third party passwords and other sensitive data. In case of non-availability or connectivity to Amazon Web Service, the encryption and decryption will not be possible, thereby inhibiting Confirm's operation. However, Amazon claims very high availability of AWS.

IT administrators should get good understanding of Amazon Web Service's AWS Identity & Access Management and AWS Key Management Service at https://aws.amazon.com before migrating Confirm to use AWS based encryption.

**AWS Costing**

Confirm Configuration Service uses two Amazon Web Services "AWS Identity and Access Management" and "AWS Key Management Service".

There is a cost associated with AWS Key Management Service. To know the details check the pricing on https://aws.amazon.com.

Prerequisites to use AWS based encryption:

- An Amazon Web Service account.
- An Identity and Access Management (IAM) user with Access Key ID and Secret Access Key.

**Note:** Download the credentials file as you create the Access keys. If not downloaded at this time, the Secret Access key will be lost and a new one will have to be generated.

Please ensure that the following is ready

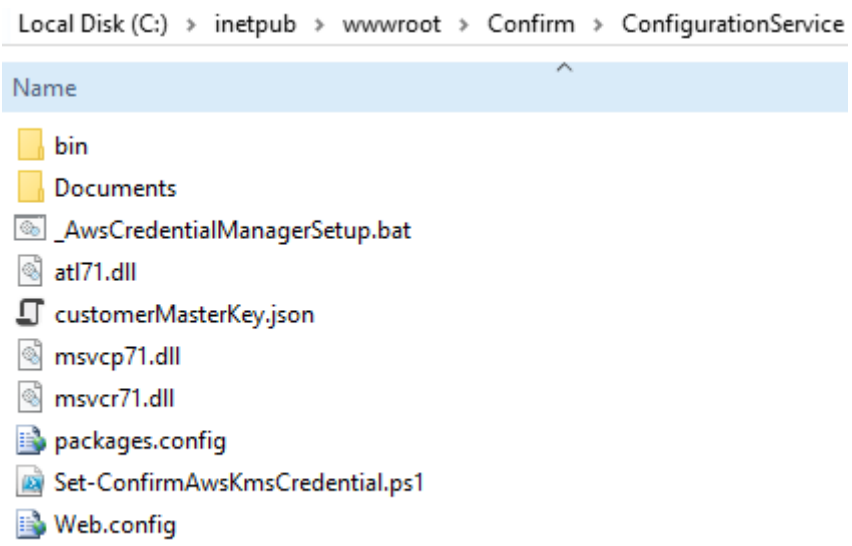
- Identify an AWS region where you wish to have the Customer Master Key (CMK) created.
- An enabled Customer Master Key (CMK) for a region.
- An IAM User having permissions on the newly generated CMK with at least Encryption and Decryption Permissions.
- Downloaded credentials file

**Configuring the AWS credentials**

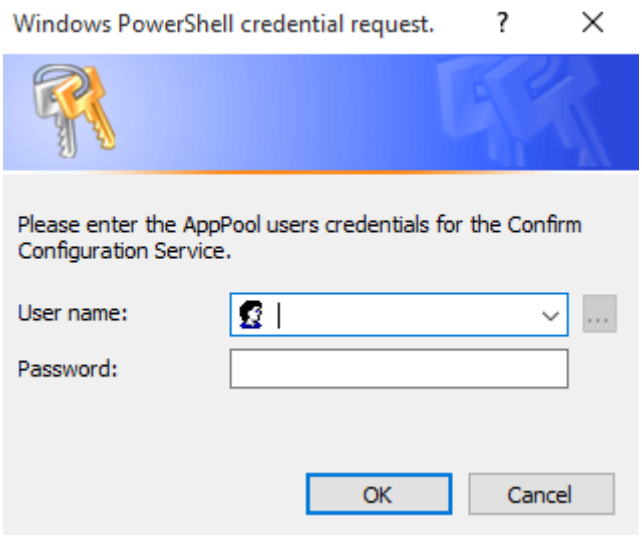
The Confirm Configuration Service reads the AWS credentials from Credential Manager which is part of the Windows Operating System.

Follow these steps to store the AWS credentials in Credential manager:

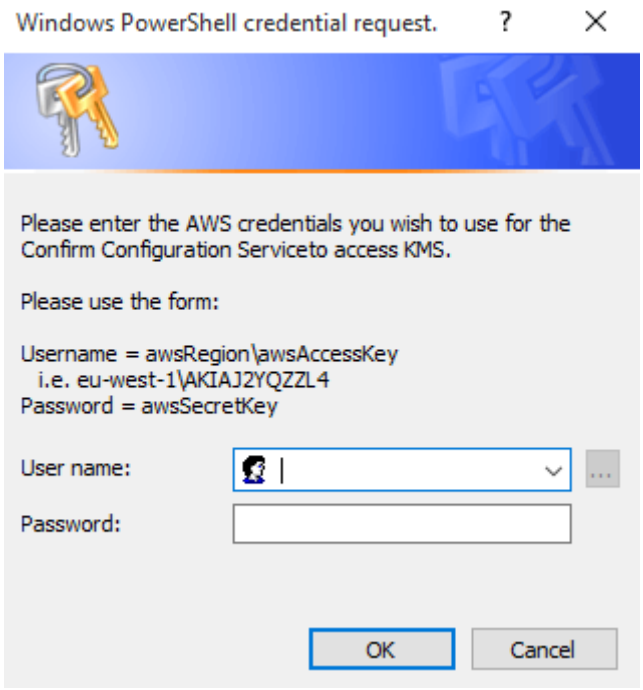
- Run the '_AwsCredentialManagerSetup.bat' file available in the Confirm Configuration Service installation directory.




- In the first step, it prompts for credential request.




  Provide the same credentials that was provided in ApplicationPool Identity. Then Click OK.
- In second step, provide AWS region, Access Key Id and Secret Access key for the Identity and Access Management (IAM) user in use. Use downloaded credentials file to find the key values.




Verify that the credentials have been created under ‘Generic Credentials’ in Window Credential Manager.

**Configuring the Confirm Configuration Service to use AWS Customer Master Key (CMK)**

- Open the ‘customerMasterKey.json’ file available in the Confirm Configuration Service installation directory.
- Update the following fields with the value from AWS Customer Master Key (CMK) assigned to the Identity and Access Management (IAM) user in use.
- "KeyId": "<Key ID value>",

  "Alias": "<Alias Name>",

  "KeyArn": "<ARN Value> "

- Taking a backup of ‘customerMasterKey.json’ file is recommended. This can be used on upgrading Confirm Configuration Service to avoid setting it up again.

**Credential Manager backup**

The credentials stored in Credential Manager can be stored in a backup file to restore in case system failure or avoiding setting up credentials again if not changed.

Backup using Credential Manager -

The Credential Manager allows to backup and restore the saved credentials.

Following are the steps:

- Open Credential Manager
- Click on the Backup Vault or Back up Credentials (in case of Window server 2012 R2) button
- Browse a file path and provide a file name
- Follow the on screen instruction to safe guard your backup file with password
- A .crd file is created on supplied path
- This same file can be used to restore your saved credentials in the event of a problem.

Using Command line -

Another way to access the Credential Manager is through the command line.

Following are the steps:

- Open a command prompt Type command rundll32.exe keymgr.dll, KRShowKeyMgr
- Press Enter
- The Stored User Names and Passwords window will open, allowing to perform the same functions as described above using Credential Manager.

It is recommended to store the backup file to a system on which Confirm Configuration Service is not installed. This will help to recover the file in case of system failure.

### Registry Configuration

Following registry entry are to be configured on the web server where Confirm Configuration Service is installed:

| Registry Location | Key | Description |
|---|---|---|
| *32 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Confirm\ConfigurationService<br>*64 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Confirm\ConfigurationService | KeyManagementEnabled | To determine whether to use External Key Management Service (Yes/ No). Default set to 'No'. |
| | KeyManagementTool | External Key management tool to be used. Set to 'AWS' |

Confirm is now configured to use AWS Key Management Service. Any subsequent access to passwords used to access Confirm database or third party applications from within Confirm or other component will be encrypted and decrypted using AWS Key Management Service.

# Configuring and Using Active Directory User Management

Once the Confirm Configuration Service has been installed and configured successfully, IT administrators can choose to manage users in Active Directory from Confirm. This is relevant for scenarios where Confirm runs in Full or Mixed Integrated Security mode.

### Registry Configuration

Following registry entry are to be configured on the web server where Confirm Configuration Service is installed:

| Registry Location | Key | Description |
|---|---|---|
| *32 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Confirm\ConfigurationService | UserStoredInActivedirectory | To determine whether to manage users in Active Directory from Confirm (Yes/ No). Default set to 'No'. |
| *64 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Confirm\ConfigurationService | AdConfig | The path of the JSON config file which stores configuration information for managing users in Active Directory (this can be a UNC path). |

Confirm is now configured to manage users in Active Directory.

# Using Confirm Configuration Service from Confirm Components

### Introduction

Confirm Components like Confirm Client should know about Confirm Configuration Service to use it. Every Confirm application requires the following registry settings to establish connection with the Confirm Configuration Service.

**Note:** Confirm Connector will not use Confirm Configuration Service for Encryption/Decryption. 'Connector Key' registry setting value would need to be in place at the requisite registry path for Connector to Encrypt/Decrypt sensitive data within the connector configuration file. If this key is not provided then default key will be used for Encryption/Decryption by the Connector.

### Registry Configuration

Following registry entry are to be configured on each machine where a Confirm component (Client, Connector, Web services, Task Processor):

| Registry Location | Key | Description |
|---|---|---|
| *32 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Confirm<br>*64 bit Machines*<br>HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Confirm | UseConfigurationWebService | To determine whether to use Confirm Configuration Service (Yes/ No). Default set to 'No'. |
| | ConfigurationServiceUrl | The URL of Confirm Configuration Service. |

### Access Control and Authentication

Every request made to the Confirm Configuration Service has to be Windows Authenticated. This means that every Confirm component will have to be running under a domain user to use Confirm Configuration Service.

Confirm Client already run under domain user. Also, Task Processor is recommended to be run under a domain user context.

Web Applications run under application pool, which in turn run under default user context of AppPoolIdentity. Point them to run under a domain user.

# FAQ

**Q: I am unable to login into Confirm after migrating to Confirm Configuration Service and it prompts the message 'Unable to process your request. Please try later or contact System Administrator.', how do I troubleshoot?**

A: Follow these steps:

- Use healthcheck api to verify Confirm Configuration Service deployment and configuration is successful.
- If configuration is unsuccessful, verify the ConfigurationServiceUrl value in Registry where Confirm is installed.
- Check the Confirm Client Provider log and Confirm Configuration Service log for more details.

*If migrated to AWS, follow these additional steps:*

- Verify that IAM User has at least Encryption and Decryption permissions on CMK.
- If Confirm Configuration Service log file reports that it is unable to read from Credential Manager, make sure that AWS credential configuration has been done with the ApplicationPool user.
- If Confirm Configuration Service log file reports that AWS access key is disabled, enable AWS access key by login into AWS account.

**Q: Can I define IAM users regionally?**

A: Users are global entities. Users can use AWS services in any geographic region.

**Q: From where can I get more info about AWS Key Management Service (KMS)?**

A: Visit URL: https://aws.amazon.com/kms/ and https://aws.amazon.com/kms/faqs/ for more details.

**Q: From where can I get more info about AWS Identity and Access Management (IAM)?**

A: Visit URL: https://aws.amazon.com/iam/ for more details.

**Q: Once migrated to AWS for encryption, how can I go back to use Confirm's encryption?**

A: Once migrated to use AWS encryption, manual intervention is required to get the database back to using Confirm's encryption. This will require cleaning up the passwords from all encrypted locations and setting them back again. You may contact Confirm support team for details.

# MFA FAQ

**Q: What is Multi-Factor Authentication (MFA)?**

A: MFA simply means you're using more than one way to prove it's really you when you log in. It's an extra layer of security that makes your account much harder for others to access, even if they somehow get your password. Here's how it usually works:

- **Something You Know:** First, you'll enter your username and password, just like always. This is the "something you know" part.
- **Something You Have:** Then, you'll be asked for a special, one-time code that comes from an app on your phone. This is the "something you have" part – because only you have your phone!

**Q: Your First Login: What to Expect!**

A: Since you're a "non-SSO user," your login experience will be a little different from those who use their company's single sign-on system (like Okta or Azure AD). Don't worry, it's straightforward! When you log in for the very first time:

1. The system will ask for your username and password as usual.
2. Then, it will guide you through setting up MFA using your authenticator app. You'll see:
    - A QR code displayed on your screen. This is like a special barcode your phone will scan.
    - You might also see a secret key (a long string of letters and numbers). This is a backup in case you can't scan the QR code for any reason.

**Q: What If Something Goes Wrong? Don't Panic!**

A: Technology can sometimes be tricky, but we've got your back! If you run into any of these situations, please don't hesitate to reach out to our support team:

- You lose your phone (oh no!).
- You accidentally uninstall or lose access to your Authenticator app.
- The 6-digit code from your app keeps getting rejected.
- You can't see the QR code or the setup screen when you're trying to set up MFA.

Our support team is ready to help you regain access quickly and smoothly!

**Q: What Does "Non-SSO User" Really Mean?**

A: Simply put, if your organization doesn't use a central "Identity Provider" (IdP) for logging into all its systems, then you're a "Non-SSO user" in this context. Now, even you have the option to enable that extra layer of MFA security!

**Q: How MFA Will Be Set Up for You**

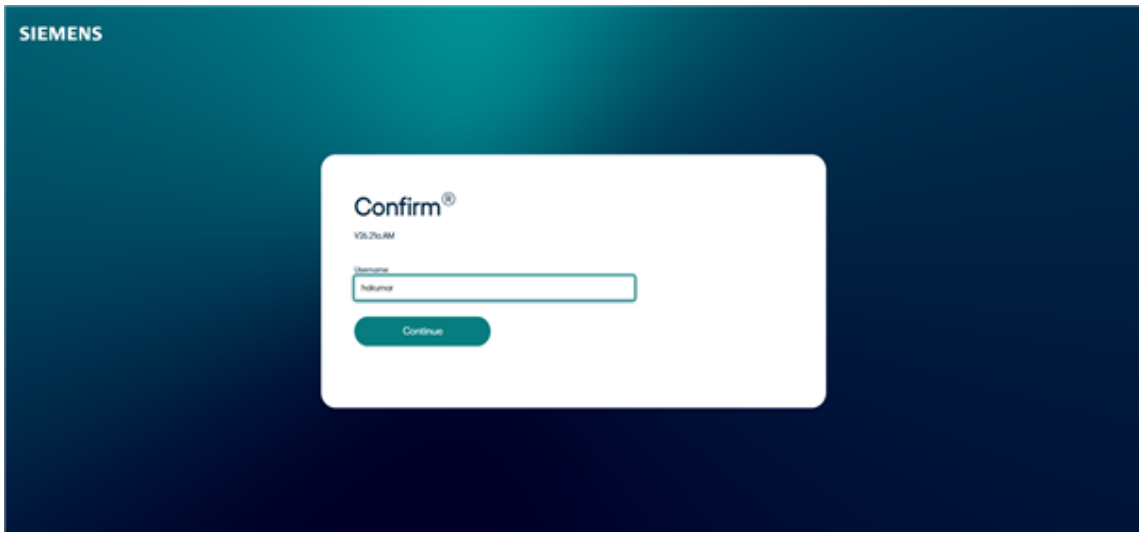
A: To get MFA enabled for your account, here's the process:

1. **Request MFA:** You'll need to raise a technical support ticket to let us know you want to use MFA.
2. **Behind the Scenes:** Our tech team will then get to work! They'll make sure your user account is set up correctly in our modern authentication system (Auth0) if it isn't already, and then they'll enable MFA for you.

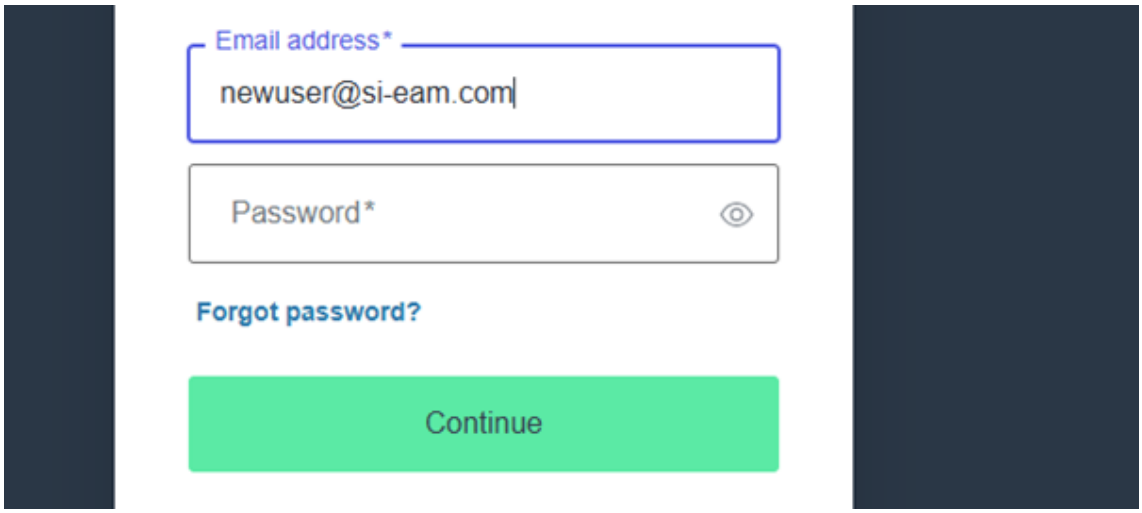
**Q: How to Set Up MFA Once It's Enabled (Your First Login)**

A: Once our team has enabled MFA for your account, here's how you'll complete the setup during your next login:

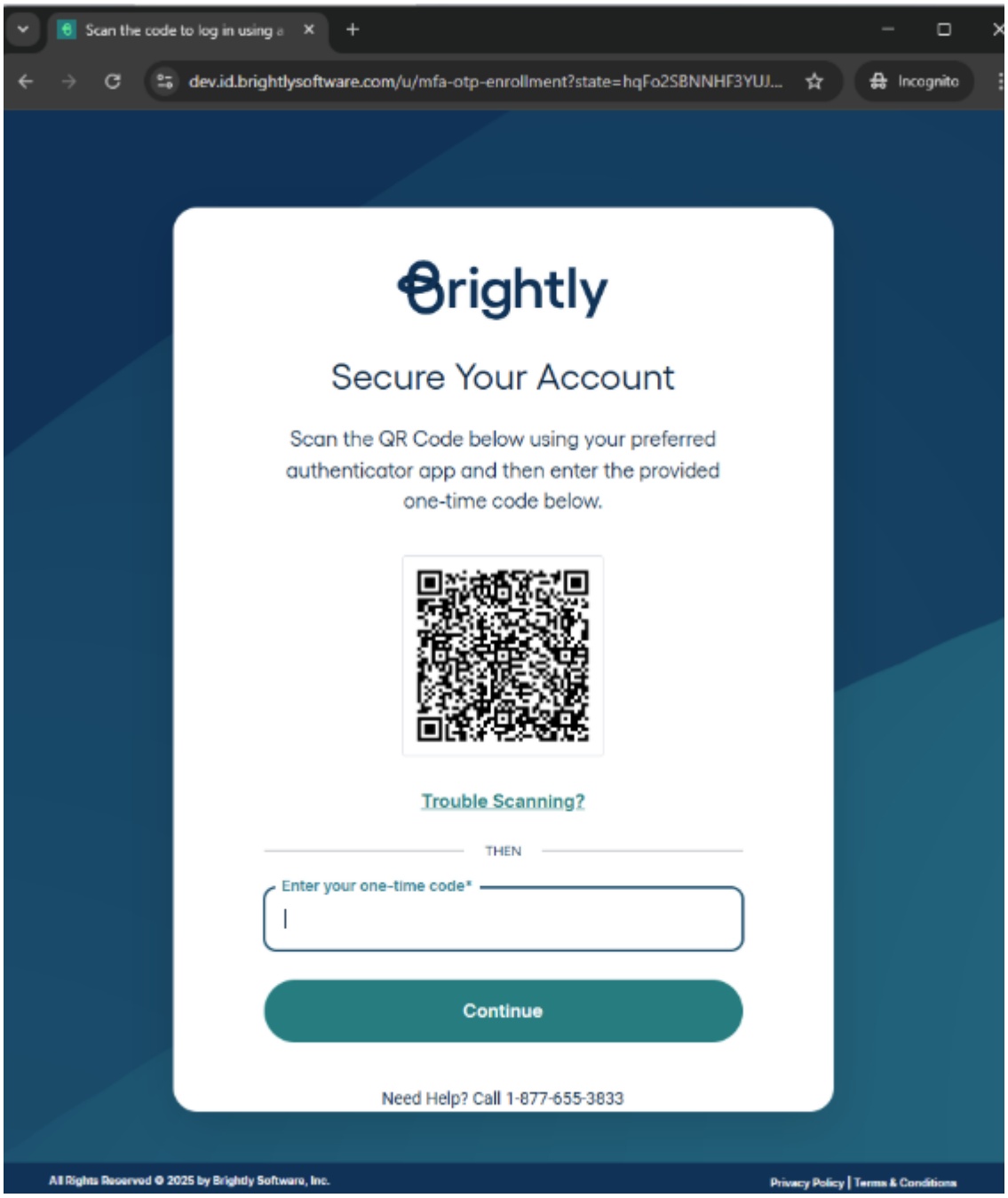
1. **Go to Confirm Web:** Open your web browser and navigate to the Confirm login page.




2. **Enter Email:** Type in your email address and click "CONTINUE."




3. **Enter Password:** Type in your password and click "CONTINUE."
4. **See the QR Code:** You'll then be taken to a page displaying a QR code. This is the magic part!




5. **Scan with Your App:**
   - Open your Microsoft Authenticator app (or Google Authenticator) on your mobile phone.
   - Look for an option to "Add Account" or a "+" sign to scan a QR code.
   - Point your phone's camera at the QR code on your computer screen.
6. **Get the One-Time Code:** Once scanned, your authenticator app will now display a constantly changing 6-digit code.
7. **Enter the Code:** Type this 6-digit code from your mobile app into the field on your computer screen and click "CONTINUE."
8. **Welcome In!** If everything matches, you'll be successfully logged into the Confirm application!


SIEMENS
Feature
Confirm
Filters
Search for Feature
Status
Live
Actions
Batch
Create
Dashboard
Map
Sites
Assets
Contracts
Inspections
Enquiries
Maintenance
Workzone
Orders
Variations
Info Centre
Help Centre