

**Oracle® Enterprise Single Sign-on  
Logon Manager**

How-To: Understanding the ESSO-LM Secondary  
Authentication API

Release 11.1.1.5.0

**21014-01**

March 2011

## Oracle Enterprise Single Sign-on Logon Manager How-To: Understanding the ESSO-LM Secondary Authentication API

Release 11.1.1.5.0

21014-01

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

# Table of Contents

---

Table of Contents.....	3
Introduction .....	4
About This Guide.....	4
Prerequisites .....	4
Terms and Abbreviations.....	4
Accessing ESSO-LM Documentation .....	4
Understanding the ESSO-LM Secondary Authentication API.....	5
Overview .....	5
The <code>SecondaryAuthKey</code> Method .....	6
The <code>FreeSecondaryAuthKey</code> Method.....	6
Example Implementation.....	7
Switching Secondary Authentication Methods .....	8
Switching from Built-In Secondary Authentication to External Secondary Authentication .....	8
Switching from External Secondary Authentication to Built-In Secondary Authentication .....	10
Switching from One External Secondary Authentication Library to Another.....	10

# Introduction

---

## About This Guide

This document describes the ESSO-LM Secondary Authentication API. The API allows you programmatically supply the passphrase answer to the `MSAuth` (Windows Authenticator v2) authenticator.

## Prerequisites

Readers of this document should have a thorough understanding of software development using the Microsoft .NET framework, including the Component Object Model (COM) technology, and related concepts.

## Terms and Abbreviations

The following table describes the terms and abbreviations used throughout this guide:

Term or Abbreviation	Description
ESSO-LM	Oracle Enterprise Single Sign-on Logon Manager
Agent	ESSO-LM client-side software
Console	ESSO-LM Administrative Console
WinAuth v2	Windows Authenticator Version 2

## Accessing ESSO-LM Documentation

We continually strive to keep ESSO-LM documentation accurate and up to date. For the latest version of this and other ESSO-LM documents, visit [http://download.oracle.com/docs/cd/E21040\\_01/index.htm](http://download.oracle.com/docs/cd/E21040_01/index.htm).

# Understanding the ESSO-LM Secondary Authentication API

---

## Overview

The secondary authentication API allows a third party application to programmatically supply a passphrase to the Windows Authenticator v2 (a.k.a. `MSAuth`) during an authentication session. This eliminates the need for interaction with the user and automates the authentication process.

The API consists of the following functions:

- **SecondaryAuthKey** – allocates the passphrase answer buffer, fills the buffer with the passphrase answer, and returns a pointer to the answer buffer.
- **FreeSecondaryAuthKey** – clears the answer buffer once the answer is no longer needed by third party code.

**Note:** The custom secondary authentication library must be validated and digitally signed by Oracle; otherwise, it will not be accepted by ESSO-LM. For assistance with this process, please contact Oracle Support.

## The SecondaryAuthKey Method

This method is used to obtain the user's passphrase answer (in our example, the user's AD SID) and store it in memory at a specified address for later retrieval.

```
BOOL SecondaryAuthKey( LPBYTE* pbAnswer, LPDWORD pdwSize )
{
    BOOL fRetVal = FALSE;

    // check for invalid parameters
    if ( NULL != pbAnswer )
    {
        // obtain user's SID - it will be used as passphrase answer
        CSid sid;
        CString strSid( sid.Sid() );

        // allocate the memory buffer
        LPBYTE pByte = new BYTE[strSid.GetLength() + 1];

        // copy the SID to the buffer
        ::memcpy( pByte, strSid.GetBuffer(), strSid.GetLength() );

        // save the address of the buffer to the passed pointer
        *pbAnswer = pByte;

        // save the size of the buffer to the passed pointer
        if ( NULL != pdwSize )
        {
            *pdwSize = strSid.GetLength() + 1;
        }

        // set successful return code
        fRetVal = TRUE;
    }

    return fRetVal;
}
```

## The FreeSecondaryAuthKey Method

This method is used to clear the passphrase answer buffer after SecondaryAuthKey has been successfully called.

```
void FreeSecondaryAuthKey( LPBYTE pbAnswer )
{
    // free the memory buffer
    delete[] pbAnswer;
}
```

## Example Implementation

Below is an example of using the secondary authentication API to programmatically supply the passphrase answer to the authenticator.

```
BOOL CResetDlg::SecondaryAuth( LPCTSTR pszDllPath )
{
    BOOL fRetVal = FALSE;

    // load SecondaryAuth.dll
    HMODULE hSecondaryAuth = LoadLibrary( pszDllPath );

    If ( NULL != hSecondaryAuth )
    {
        SECONDARYAUTHKEY pfnSecondaryAuthKey = (SECONDARYAUTHKEY)
        GetProcAddress( hSecondaryAuth, "SecondaryAuthKey" );
        if ( NULL != pfnSecondaryAuthKey )
        {
            LPBYTE pbByte = NULL;
            DWORD dwAnswerSize = 0;

            // call SecondaryAuthKey to get the passphrase answer
            BOOL bAnswerResult = pfnSecondaryAuthKey( &pbByte,
            &dwAnswerSize );

            // use the returned answer - pbByte
            // ...

            // call FreeSecondaryAuthKey to let the library free the
            memory
            FREESECONDARYAUTHKEY pfnFreeSecondaryAuthKey =
            (FREESECONDARYAUTHKEY) GetProcAddress( hSecondaryAuth,
            "FreeSecondaryAuthKey" );
            if ( NULL != pfnFreeSecondaryAuthKey )
            {
                pfnFreeSecondaryAuthKey( pbByte );
            }

            // set successful return code
            fRetVal = TRUE;
        }

        // unload SecondaryAuth.dll
        FreeLibrary( hSecondaryAuth );
    }

    return fRetVal;
}
```

## Switching Secondary Authentication Methods

You have the ability to change the method used by Windows Authenticator v2 (WinAuth v2) to verify the user's identity to another method if necessary. The following scenarios are supported:

- WinAuth v2 built-in secondary authentication to external secondary authentication
- External secondary authentication to WinAuth v2 built-in secondary authentication
- One external secondary authentication library to another

## Switching from Built-In Secondary Authentication to External Secondary Authentication

To configure WinAuth v2 for recovery via custom secondary authentication library, do the following:

1. Start the ESSO-LM Administrative Console.
2. In the tree in the left pane, right-click the **Global Agent Settings** node and select **Import** → **From Live HKLM** from the context menu.
3. Under the "Live" settings set, navigate to **Authentication** → **Windows v2**.  
If you have previously configured ESSO-LM to use either the user's AD SID or a secure random key as a secondary authentication method, revert back to interactive passphrase by deselecting the check box next to the **Recovery Method** option. (This reverts the option to its default value, **User passphrase**.)
4. Create a directory named identically to the GUID of your custom library in the following directory:

```
<oracle_install_dir>\v-GO SSO\AUI\Recovery\
```

**Note:** Substitute the full path of the directory in which Oracle ESSO products are installed for <oracle\_install\_dir>.

For example, if your library's GUID is {B623C4E7-A383-4194-A719-7B17D074A70F}, you would create the following directory:

```
<oracle_install_dir>\v-GO SSO\AUI\Recovery\{B623C4E7-A383-4194-A719-7B17D074A70F}
```

5. Place your custom library file in the directory you created in step 4.

6. Add a GUID entry to ESSO-LM's secondary authentication methods list for your custom library.
  - a. Create a key named identically to the GUID of your custom library under the following registry location:

- On 32-bit systems:

```
HKEY_LOCAL_MACHINE\Software\Passlogix\AUI\MsAuth\
RecoveryMethods\
```

- On 64-bit systems:

```
HKEY_LOCAL_MACHINE\Software\Wow6432Node\Passlogix\AUI\MsAuth\
RecoveryMethods\
```

For example, if your library's GUID is {B623C4E7-A383-4194-A719-7B17D074A70F}, you will create the following key on a 32-bit system:

```
HKEY_LOCAL_MACHINE\Software\Passlogix\AUI\MsAuth\RecoveryMethods\
{B623C4E7-A383-4194-A719-7B17D074A70F}
```

- b. Under the key you created in step 6a, create a string value named `Path` and set it to the full path and file name of your custom library. In our example, you would set it to:

```
<oracle_install_dir>\v-GO SSO\AUI\Recovery\{B623C4E7-A383-4194-
A719-7B17D074A70F}\<MyCustomLibrary.dll>
```

Where `<oracle_install_dir>` is the full path of the directory in which Oracle ESSO products are installed and `<MyCustomLibrary.dll>` is the file name of your custom library.

7. Set ESSO-LM's recovery method to your custom secondary authentication library. If it does not already exist, create a string value named `ResetMethodGUID` under `HKEY_LOCAL_MACHINE\Software\Passlogix\AUI\MsAuth\RecoveryMethods\` and set it to the GUID of your custom library.
8. Reinitialize the WinAuth v2 settings with the newly selected configuration:
  - a. Launch ESSO-LM, double-click its system tray icon, and select **Settings** in the left-hand pane of the window that appears.
  - b. Select the **Authentication** tab, then click **Change**. The Setup Wizard appears.
  - c. Follow the prompts in the wizard. When prompted to select your primary logon method, make sure that **Windows Logon v2** remains selected.
  - d. Complete the remaining steps in the wizard.

## Switching from External Secondary Authentication to Built-In Secondary Authentication

To configure WinAuth v2 for recovery via one of ESSO-LM's built-in secondary authentication methods, do the following:

1. Start the ESSO-LM Administrative Console.
2. In the tree in the left pane, right-click the **Global Agent Settings** node and select **Import → From Live HKLM** from the context menu.
3. Under the "Live" settings set, navigate to **Authentication → Windows v2**.
4. Select the check box next to the **Recovery Method** option and do one of the following:
  - To use the interactive passphrase prompt with a user-supplied passphrase for secondary authentication, select **User passphrase** from the drop-down list
  - To use silent secondary authentication using the user's AD SID as the passphrase answer, select **Passphrase suppression using user's SID** from the drop-down list
  - To use silent secondary authentication with a secure random key as the passphrase answer, select **Passphrase suppression using secure key** from the drop-down list
5. Save your changes locally or publish them to the repository, as applicable.
6. Reinitialize the WinAuth v2 settings with the newly selected configuration:
  - a. Launch ESSO-LM, double-click its system tray icon, and select **Settings** in the left-hand pane of the window that appears.
  - b. Select the **Authentication** tab, then click **Change**. The Setup Wizard appears.
  - c. Follow the prompts in the wizard. When prompted to select your primary logon method, make sure that **Windows Logon v2** remains selected.
  - d. Complete the remaining steps in the wizard.

## Switching from One External Secondary Authentication Library to Another

If you are currently using one external secondary authentication library and want to switch to a different external library, repeat the steps in [Switching from WinAuth v2 Built-In Passphrase Support to External Secondary Authentication](#).