One Identity Manager 8.1.4

HTML5 Development Guide

Copyright 2020 One Identity LLC.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of One Identity LLC .

The information in this document is provided in connection with One Identity products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of One Identity LLC products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, ONE IDENTITY ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ONE IDENTITY BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ONE IDENTITY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. One Identity makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. One Identity does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

One Identity LLC.
Attn: LEGAL Dept
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our Web site (http://www.OneIdentity.com) for regional and international office information.

Patents

One Identity is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at http://www.OneIdentity.com/legal/patents.aspx.

Trademarks

One Identity and the One Identity logo are trademarks and registered trademarks of One Identity LLC. in the U.S.A. and other countries. For a complete list of One Identity trademarks, please visit our website at www.OneIdentity.com/legal. All other trademarks are the property of their respective owners.

Legend

WARNING: A WARNING icon highlights a potential risk of bodily injury or property damage, for which industry-standard safety precautions are advised. This icon is often associated with electrical hazards related to hardware.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

One Identity Manager HTML5 Development Guide
Updated - 19 October 2020, 07:42
Version - 8.1.4

Contents

About this guide **4**

Architecture of One Identity Manager HTML applications **5**

HTML5 development **6**

Installing the One Identity Visual Studio Code extension 7

Setting up working environment 7

Connecting to the One Identity Manager database 7

Creating a development folder 8

Using the One Identity Visual Studio Code extension 9

Compilation branches 9

Selecting a compilation branch 9

Creating compilation branches 10

Deleting compilation branches 10

Updating the API client 10

Creating HTML applications automatically 11

Compile HTML applications and save to database 11

Check translations 12

Creating applications 12

About us **15**

Contacting us 15

Technical support resources 15

1

About this guide

This guide shows web developers how to view One Identity Manager HTML applications as code and how to understand their internal functionality. For that purpose, One Identity provides you with an extension for Visual Studio Code.

Available documentation

The online version of One Identity Manager documentation is available in the Support portal under Technical Documentation. You will find videos with additional information at www.YouTube.com/OneIdentity.

2

Architecture of One Identity Manager HTML applications

One Identity Manager manages a folder structure that contains the source files for all HTML applications. In the data model, this folder structure is stored as part of automatic software updates and it is assigned the machine role **HTML Development**.

You can store this folder structure locally in a development folder. In the development folder, the folders for database modules are stored at the top level. Below that you will find a folder for each HTML application. See section Creating a development folder on page 8 to find out how to create a development folder.

The HTML applications are structured as nodeJS applications that use the **Angular** framework. Generally, any HTML applications that can be compiled as nodeJS applications are supported.

HTML applications use the API Client to communicate with the One Identity Manager API. API Client is an npm library that is automatically generated and stored to the database during API compilation. API Client controls all network access on the API Server.

Much of the logic for HTML applications is realized by plugins that can be used independently of a specific HTML application. You can also use this logic for your own applications.

3

HTML5 development

In this section, you will find information on how to deal with the One Identity Visual Studio Code extension.

You can use the Visual Studio Code extension to launch the ImxClient command line program directly in Visual Studio Code without any in-depth knowledge of the syntax:

- Connect to the database
- Create the development folder
- Manage and use compilation branches
- Update HTML applications with the same setup
- Compile a HTML application
- Update the base libraries (API Client)
- Check for missing translations

The Visual Studio Code extension also provides you with the following helpful information:

- Status of the current database connection
- Information about the ImxClient used
- Information about the API Client used
- Status of the development order
- Compilation branch display
- Information on the last translation check

Related topics

- Installing the One Identity Visual Studio Code extension on page 7
- Setting up working environment on page 7
- Using the One Identity Visual Studio Code extension on page 9
- Creating applications on page 12

Installing the One Identity Visual Studio Code extension

To install the One Identity Visual Studio Code extension

1. Start Visual Studio Code.
2. In Visual Studio Code, click **Extensions** in the left navigation.
3. In **Extensions**, click **...** | **Install from VSIX**.
4. In the file browser, select the file `vcode-extension.vsix` in the One Identity Manager installation folder and click **Install**.
5. Restart Visual Studio Code.
6. In Visual Studio Code, click **Explorer** in the left navigation.
7. In **Explorer**, navigate to **One Identity** | **Configuration**.
8. Next to **Imx client**, click **One Identity Manager: edit imx client path**.
9. In the file browser, select the file `ImxClient.exe` in the One Identity Manager installation folder and click **Open**.

Setting up working environment

In this section, you will learn how to use the One Identity Visual Studio Code extension to set up your working environment. After setup, you will be able to use the Visual Studio Code extension and its functions without any issues.

Related topics

- Connecting to the One Identity Manager database on page 7
- Creating a development folder on page 8

Connecting to the One Identity Manager database

Many of the functions available to you in the Visual Studio Code extension require a connection to the One Identity Manager database.

To connect to the One Identity Manager database

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Workspace**.
3. Next to **Database**, click **One Identity Manager: connect to database**.

   The dialog to select the database connection is opened.
4. In the dialog, perform one of the following actions:
   - to use an existing connection to the One Identity Manager database, select it in the **Select a database connection** menu.

     - OR -
   - to create a new connection to the One Identity Manager database, click **Add new connection** and enter a new connection.
5. Select the authentication method and, under **Authentication method**, enter the login data for the database.
6. Click **Log in**.

Creating a development folder

You must perform the following actions to create a development folder:

- Download the HTML application folders from the database and save them as sub-folders in the development folder.
- Download the libraries from the database and save them in the sub-folder `imx-modules` in the development folder.
- The newest libraries are downloaded to the sub-folder `imx-modules` in the development folder. Skip this step if a connection to the database has not been established (also see Connecting to the One Identity Manager database on page 7).

To set up a One Identity Manager development folder

1. On the hard drive, create a folder that you want to use as a development folder.

   NOTE: The path to this folder and the folder itself, can only contain UTF-8 characters.
2. In Visual Studio Code, click **File** | **Open folder** in the toolbar.
3. In the file browser, select the folder that you created previously to use as the development folder.
4. In Visual Studio Code, click **Explorer** in the left navigation.
5. In **Explorer**, navigate to **One Identity** | **Workspace**.
6. Next to **Development folder**, click **One Identity Manager: set up development folder**.
7. Confirm this prompt with **Continue setup**.

Using the One Identity Visual Studio Code extension

In this section, you will learn how to use the One Identity Visual Studio Code extension.

Related topics

- Compilation branches on page 9
- Updating the API client on page 10
- Creating HTML applications automatically on page 11
- Compile HTML applications and save to database on page 11
- Check translations on page 12

Compilation branches

Use compilation branches to manage different versions of your compiled project and to store the versions in the database. You can select the required compilation branch, show available compilation branches, create compilation branches and delete compilation branches.

Related topics

- Selecting a compilation branch on page 9
- Creating compilation branches on page 10
- Deleting compilation branches on page 10

Selecting a compilation branch

To select and use a compilation branch

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Configuration**.
3. Next to **Branch ID**, click **One Identity Manager: edit branch id**.

   TIP: If a connection to the One Identity Manager database has not yet been established, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

   A menu with available compilation branches is opened.
4. In the menu, click the compilation branch you would like to use.

Creating compilation branches

To create a compilation branch

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Configuration**.
3. Next to **Branch ID**, click **Add branch**.

   An input field opens.
4. In the input field, enter a name for the compilation branch and press **Enter**.

   TIP: If a connection to the One Identity Manager database has not yet been established, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

Deleting compilation branches

To delete a compilation branch

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Configuration**.
3. Next to **Branch ID**, click **One Identity Manager: edit branch id**.

   TIP: If a connection to the One Identity Manager database has not yet been established, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

   A menu with available compilation branches is opened.
4. In the menu, click the configuration branch you would like to delete.
5. In **Explorer** next to **Branch ID**, click **Delete branch**.
6. Confirm this prompt with **OK**.

Updating the API client

You can update the base libraries and, therefore, the API client at any time.

To update the API Client

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Workspace**.
3. Next to **API Client**, click **One Identity Manager: update api client**.

   TIP: If a connection to the One Identity Manager database has not yet been estab-

lished, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

Creating HTML applications automatically

To recognize and create HTML applications automatically, check the folders available in your development folder. It involves the following steps:

- The sub-folder `assets` in the working directory is connected to the sub-folder `src/assets` in the application.
- The relevant plugins are connected and integrated.

NOTE: HTML applications are also automatically set up before each compilation.

To automatically create HTML applications

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Workspace** | **Development folder**.
3. Next to **Apps**, click **reload existing apps**.

   TIP: If a connection to the One Identity Manager database has not yet been established, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

Compile HTML applications and save to database

You can compile your HTML applications and save the results automatically to the One Identity Manager database.

To compile HTML applications and save to the database

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Workspace** | **Development folder** | **Apps**.
3. Click **Compile app** next to the HTML application that you would like to compile and save to the database.

   TIP: If a connection to the One Identity Manager database has not yet been established, you must set up the connection now (see Connecting to the One Identity Manager database on page 7).

   A debug compilation is executed and the changes are saved to the enabled Compilation branch.

Check translations

In your HTML application, you can check the development folder for missing translations.

To check the development folder for missing translations

1. In Visual Studio Code, click **Explorer** in the left navigation.
2. In **Explorer**, navigate to **One Identity** | **Workspace** | **Development folder**.
3. Next to **Translations**, click **Check translations**.

   A report opens. The report shows you any files containing texts that have not yet been translated or have not been translated in full.

Creating applications

You can the use One Identity Visual Studio Code extension to create your HTML applications and link them into the existing system.

Follow these steps to create your HTML application:

1. Create the new HTML application.
2. Import the source files into the database.
3. Add the HTML application to the database.

To create a new HTML application

1. Install the One Identity Visual Studio Code extension (see Installing the One Identity Visual Studio Code extension on page 7).
2. Set up the work environment with the help of the One Identity Visual studio Code extension (see Setting up working environment on page 7),
3. Add a new CCC folder to your work environment.
4. Ensure that you are using the correct version of Angular (6.9.2):
   a. Open a command line prompt.
   b. In the CCC folder, run the **ng version** command.
   c. Check the version number returned. The version must be 6.2.9.

      TIP: If nothing is returned it means that Angular is not installed locally or globally.

      To install Angular globally, run the command: **npm install -g @angular/cli@6.2.9**.

      To install Angular locally, run the command: **npm install @angular/cli@6.2.9**.
5. In the CCC folder, run the command: **ng new <your HTML application name>**

6. In the QBM/OpsWeb folder, copy the imx-plugin-config.json file and add it to the CCC/<your HTML application name> folder.
7. In the CCC/<your HTML application name> folder, edit the (freshly copied) imx-plugin-config.json file:
   - Remove all apart **common** and **shared**.
8. In the CCC/<your HTML application name> folder, edit the package.json file:
   a. In the scripts section, add "build:debug": "ng build --prod --source-map".
   b. In the dependencies section, add dependencies to imx-api, imx-qbm-components, and imx-qbm-dbts.
   c. In the dependencies | rxjs section, change the version to ^6.3.3.
9. In the CCC/<your HTML application name> folder, edit the angular.json file:
   - In the projects | <your HTML application name> | architect | build | options subsection, add "outputPath": "dist".
   - In the projects | <your HTML application name> | architect | build | options subsection, add "preserveSymlinks": true.
10. In the CCC/<your HTML application name>/src folder, edit the index.html file:
    - In the head section, add <base href=".">.
11. In the CCC/<your HTML application name>/src folder, edit the environment.<your domain name>.ts and environment.ts files:
    - Enter one entry for each client URL:

      For environment.<you domain name>.ts: clientUrl: ''

      For environment.ts: clientUrl: 'http://localhost:8182'
12. In the CCC/<your HTML application name> folder, edit the tsconfig.json file:
    - In the compilerOptions | paths subsection, add "@shared/*": ["src/imx-plugins/QBM/shared/*"].
13. Use the One Identity Visual Studio Code extension to make sure that you are not using complication branches (see Selecting a compilation branch on page 9).
14. Use the One Identity Visual Studio Code extension to compile your HTML application (see Compile HTML applications and save to database on page 11).
15. Open your compiled HTML application over the URL: **$<App server URL>/html/<your HTML application name>/**.

    NOTE: Here you must use the same name for the HTML application as stored in the package.json file.

To import source files in to the database

1. Go to your work environment.
2. Start the Software Loader program.
3. Click **Import into database**.
4. Click **Next**.

5. On the **Connect to database** page, select a database and enter the user credentials.
6. Click **Next**.
7. Select your work environment.
8. Mark all the files in the `CCC/<your HTML application name>` folder but exclude the following folders:
    - `node_modules`
    - `dist`
    - `.git`
    - `src/assets`
    - `src/imx-modules`
9. Click **Next**.
10. Confirm the prompt with **Yes**.
11. On the **Assign machine roles** page, mark all the file and enable the **HTML Development** machine role.
12. Click **Next**.
13. On the **Select change label** page, perform the following actions:
    - To not use a change label, click **Do not assign the files to a change label**.
    - To use a change label, click **Assign files to following change label**. Then click **...** and select the desired changed labels.
14. Click **Next**.
15. On the **Transferring files** page, once the files have been successfully transferred to the database, click **Next**.
16. On the **Wizard complete** page, click **Finish**.

To add the HTML application to the database

1. Start the Designer program.
2. Navigate to **Base data** | **Security settings** | **HTML applications**.
3. Select the **Object** | **New** menu item to add a new item.
4. Enter a display name and the **CCC/<your HTML application name>** path.

Related topics

- Installing the One Identity Visual Studio Code extension
- Selecting a compilation branch
- Compile HTML applications and save to database

About us

One Identity solutions eliminate the complexities and time-consuming processes often required to govern identities, manage privileged accounts and control access. Our solutions enhance business agility while addressing your IAM challenges with on-premises, cloud and hybrid environments.

Contacting us

For sales and other inquiries, such as licensing, support, and renewals, visit https://www.oneidentity.com/company/contact-us.aspx.

Technical support resources

Technical support is available to One Identity customers with a valid maintenance contract and customers who have trial versions. You can access the Support Portal at https://support.oneidentity.com/.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request
- View Knowledge Base articles
- Sign up for product notifications
- Download software and technical documentation
- View how-to videos at www.YouTube.com/OneIdentity
- Engage in community discussions
- Chat with support engineers online
- View services to assist you with your product