Copyright 2019 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 indicates a potential for property damage, personal injury, or death.

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

**IMPORTANT, NOTE, TIP, MOBILE, or VIDEO**: An information icon indicates supporting information.

One Identity Manager API Designer User and Development Guide
Updated - August 2019
Version - 8.1.1
# Contents

The API Designer ............................................................. 7

Basic principles of API development .................................. 8

Examples and help – Software Development Kit .................... 9

Quick start – Creating an API ............................................. 10

Working with the API Designer ........................................ 11

Starting the API Designer .................................................. 11

Project types .................................................................. 12

Changing the project type ............................................... 12

Solution projects ................................................................ 14

  Importing solution project changes into a database ............. 14

User interface .................................................................... 15

  Menu bar ....................................................................... 16

  Toolbar .......................................................................... 19

  Status bar ..................................................................... 21

Navigation ....................................................................... 21

  Opening the navigation ............................................... 22

Home page ...................................................................... 23

  Opening the homepage ................................................. 23

Global settings .................................................................. 24

  Changing global settings .............................................. 26

  Changing language ....................................................... 26

Database objects ............................................................ 27

  Object properties ......................................................... 27

    Displaying object properties ..................................... 28

  Managing objects in a project (solution) ......................... 29

    Opening the solution ................................................. 30

  Editing database objects (definition tree view) .................. 30

    Opening the definition tree view ............................... 31

    Context menu in the definition tree window ................. 31

Extensions ...................................................................... 33
Viewing and saving generated code belonging to a node ........................................ 34
Node editor view ........................................................................................................ 35
Labeling changes ........................................................................................................ 37
Using change labels .................................................................................................... 37
Creating a change label ............................................................................................... 38
Change management ..................................................................................................... 39
Deleting change labels ................................................................................................. 40
Find and replace ........................................................................................................... 40
Performing a search ....................................................................................................... 43
Multilingual captions ................................................................................................... 44
Creating multilingual captions .................................................................................... 45
Searching for a multilingual caption ............................................................................ 45
Editing a multilingual caption ....................................................................................... 46
Deleting a multilingual caption ..................................................................................... 47
Editing database queries ............................................................................................. 48
Displaying database queries ......................................................................................... 49
Creating a database query ............................................................................................ 49
Editing a database query ............................................................................................... 50
Testing a database query .............................................................................................. 51
Deleting a database query ............................................................................................. 51
Managing tabs ............................................................................................................... 52
Managing layouts .......................................................................................................... 53
Displaying the change history (command list) ............................................................ 53
Opening the command list ......................................................................................... 54
Bookmarks ................................................................................................................... 54
Editing bookmarks ....................................................................................................... 55
Setting bookmarks ........................................................................................................ 56
Deleting bookmarks ..................................................................................................... 56
Compiling an API ......................................................................................................... 57
Testing a compilation .................................................................................................... 58
Starting a compilation ................................................................................................. 59
Managing compilation errors and warnings (task window) ......................................... 59
Opening the task window ............................................................................................. 60
Resolving errors and warnings .................................................................................... 61
Opening the compiling log .......................................................................................... 61
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing versions (compilation branches)</td>
<td>62</td>
</tr>
<tr>
<td>Selecting and using compilation branches</td>
<td>62</td>
</tr>
<tr>
<td>Creating compilation branches</td>
<td>63</td>
</tr>
<tr>
<td>Editing compilation branches</td>
<td>63</td>
</tr>
<tr>
<td>Deleting compilation branches</td>
<td>64</td>
</tr>
<tr>
<td>Testing an API</td>
<td>64</td>
</tr>
<tr>
<td>Linking C# projects to the API Designer</td>
<td>65</td>
</tr>
<tr>
<td><strong>API projects</strong></td>
<td>68</td>
</tr>
<tr>
<td>Creating API projects</td>
<td>68</td>
</tr>
<tr>
<td>Editing API projects</td>
<td>69</td>
</tr>
<tr>
<td>Deleting API projects</td>
<td>70</td>
</tr>
<tr>
<td>Configuring authentication</td>
<td>71</td>
</tr>
<tr>
<td>Configuring the compilation</td>
<td>74</td>
</tr>
<tr>
<td>Assigning API files to an API project</td>
<td>75</td>
</tr>
<tr>
<td>Importing API projects</td>
<td>75</td>
</tr>
<tr>
<td><strong>API files</strong></td>
<td>77</td>
</tr>
<tr>
<td>Creating API files</td>
<td>78</td>
</tr>
<tr>
<td>Editing API files</td>
<td>78</td>
</tr>
<tr>
<td>Deleting API files</td>
<td>79</td>
</tr>
<tr>
<td>Assigning API files to an API project</td>
<td>80</td>
</tr>
<tr>
<td>Importing API files</td>
<td>80</td>
</tr>
<tr>
<td><strong>API Server</strong></td>
<td>82</td>
</tr>
<tr>
<td><strong>ImxClient command line program</strong></td>
<td>83</td>
</tr>
<tr>
<td>Starting the ImxClient command line program</td>
<td>83</td>
</tr>
<tr>
<td>ImxClient command overview</td>
<td>83</td>
</tr>
<tr>
<td>help</td>
<td>84</td>
</tr>
<tr>
<td>compile-app</td>
<td>84</td>
</tr>
<tr>
<td>repl</td>
<td>85</td>
</tr>
<tr>
<td>branch</td>
<td>85</td>
</tr>
<tr>
<td>connect</td>
<td>86</td>
</tr>
<tr>
<td>compile-api</td>
<td>86</td>
</tr>
<tr>
<td>fetch-files</td>
<td>88</td>
</tr>
<tr>
<td>push-files</td>
<td>88</td>
</tr>
<tr>
<td>get-apistate</td>
<td>89</td>
</tr>
</tbody>
</table>
The API Designer

The API Designer allows you to create, record, compile and publish a REST-API (Representational State Transfer Application Programming Interface) in the quickest way possible. This API is based on the OpenAPI Specification and the One Identity Manager database model.

The main benefits of API Designer:

- Easy and fast operation
- The finished API ‘understands’ the One Identity Manager database model.
- Modifications to the API are visible.
- Supports the principles of good API design.
- OpenAPI support: APIs that you create using the API Designer are based upon the OpenAPI specification as standard. This allows you to make use of other tools:
  - Swagger: Use Swagger to create code, documentation and test cases.
  - Postman: Use Postman to test the various methods of your API.
Basic principles of API development

The main components of an API created using the API Designer are files and projects.

Basic information on API files:

- You can use API files to send data to the application or request data from the application.
- One API file can belong to more than one project.

The basics of API projects:

- API projects combine multiple API files into logical sections. The API project includes the configuration.

Related topics

- Quick start – Creating an API on page 10
Examples and help – Software Development Kit

To make it easier for you to start developing your API with the API Designer, One Identity provides a Software Development Kit (SDK) with lots of commented code example.

The SDK can be found on the installation medium in the directory `QBM\dvd\AddOn\ApiSamples`. 
Quick start – Creating an API

The following provides a rough list of the steps you must take to create an API in the API Designer.

1. Start the API Designer.
2. Create API files (in which you can define an API method, for example).
3. Create an API project.
4. Configure the authentication for the API project (e.g. Single Sign-On).
5. Assign the created API files to the API project.
6. Test the API.
7. Compile the API.
8. Save the changes to the database.

Related topics

- Starting the API Designer on page 11
- Creating API files on page 78
- Creating API projects on page 68
- Configuring authentication on page 71
- Assigning API files to an API project on page 80
- Testing an API on page 64
- Compiling an API on page 57
- Importing solution project changes into a database on page 14
Working with the API Designer

This section provides you with general information on working with the API Designer. For example, you will learn how

- the API Designer interface is structured
- to edit your projects - locally or within the database itself
- to edit database objects
- to compile an API

and much more.

Starting the API Designer

Before you start developing your API, you must start the API Designer.

To start the API Designer

1. Go to the installation path for API Designer.
2. Run the ApiDesigner.Editor.exe application.
   The API Designer opens.
3. Perform one of the following tasks:
   - 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.
4. Select the authentication method and, under Authentication method, enter the login data for the database.
5. Click Connect.
6. In the Select project save type dialog, perform one of the following tasks.
To save the project in the database, click **Database project**.
- OR -
To save the project within a solution, click **Solution project**.

**TIP:** For more information on the tables, see **Project types** on page 12.

7. (Optional) To save the selected storage location for future projects, select the **Use selected project as default** option.
8. Click **OK**.

The API Designer opens and displays the **start page**.

**Related topics**
- Home page on page 23
- Project types on page 12

**Project types**

The API Designer distinguishes between two different project types for saving your changes.

- **Database project:** The project will be stored in the database. Your saved changes are visible to others.
- **Solution project:** API Designer allows file-based editing of custom objects **locally** on your own computer. For more information, see **Solution projects** on page 14.

**Related topics**
- Changing the project type on page 12
- Solution projects on page 14
- Importing solution project changes into a database on page 14
- Menu bar on page 16

**Changing the project type**

You can save your project as a **database project** or as a **solution project**. For more information, see **Project types** on page 12.
To save the project as a database project

1. Click Edit | Set project type for projects on the menu bar.
2. Click on Database project in the Select project save type dialog.
3. (Optional) To save the selection and use it next time you start the API Designer, enable the option Use selected project as default.

   TIP: To undo this setting:
   1. Click on Connection | Settings on the menu bar.
   2. Expand the Editor area in the Global settings dialog box.
   3. In the Editor area, enable the option Ask for save type after every startup.
      Next time the program is started, the Select project save type dialog will be displayed.
4. Click OK.

To save the project as a solution project

1. Click Edit | Set project type for projects on the menu bar.
2. Click on Solution project in the Select project save type dialog.
3. Perform one of the following actions:
   - Click Create solution and use the wizard to create a new solution.
   - OR -
   - Click Load solution and enter the path to an existing solution.
   - OR -
   - Click on the solution most recently used.
4. (Optional) To save the selection and use it next time you start the API Designer, enable the option Use selected project as default.

   TIP: To undo this setting:
   1. Click on Connection | Settings on the menu bar.
   2. Expand the Editor area in the Global settings dialog box.
   3. In the Editor area, enable the option Ask for save type after every startup.
      Next time the program is started, the Select project save type dialog will be displayed.
5. Click OK.

Related topics

- Project types on page 12
- Menu bar on page 16
Solution projects

API Designer allows for custom objects to be edited **locally** in file-based format on your own computer. To do this you will need to export any custom objects stored within the database to the API Designer, i.e. you must use the ‘solution project’ project type (see Changing the project type on page 12).

To import the changes made to the solution project back to the database, follow the instructions in Importing solution project changes into a database on page 14.

After exporting these objects from the database, you can edit and remove them on your computer, or supplement them with additional objects. Local object editing is no different than editing within the database project.

During export, any objects contained in the selected module are copied to the hard drive:

- API projects
- API files

Related topics

- Changing the project type on page 12
- Importing solution project changes into a database on page 14
- Menu bar on page 16

Importing solution project changes into a database

Changes that have been executed within a solution project must be transferred to the One Identity Manager database.

**To import objects into the One Identity Manager**

1. Click Edit | Import objects into database on the menu bar.
2. In the Import objects into database dialog, enable the option Remove deleted objects from database. Objects that you have deleted from your solution project are now also deleted from the database.
3. Go to Solution objects to select the objects that have been changed and are to be exported into the database.

   **TIP:** To select or deselect listed objects quickly, enable the Select all/deselect all option.

4. Click Next.
5. Click Finished.
Related topics

- Solution projects on page 14
- Project types on page 12

User interface

This section will give you a rough overview of the graphical user interface of the API Designer.

Use the mouse and keyboard to interact with the API Designer's graphical user interface. Use a screen resolution of at least 1280 x 1024 pixels and a minimum color depth of 16 bits for optimum display.

ℹ️ TIP: The API Designer view can be customized at any time to suit your needs by moving, closing or hiding items.

Main user interface components

- **Top – Title bar**
  The title bar shows the program icon, program name and connected database (including the current user).

- **Top – Menu bar**
  You can use the menu bar menus to call up submenus and quickly access and execute many functions in API Designer. For more information, see Menu bar on page 16.

- **Top – Toolbar**
  The toolbar contains different icons that you can click on to access other functions. For more information, see Toolbar on page 19.

- **Left – Navigation**
  Use the navigation to manage (create, open, delete) API configurations, API files and API projects. You can also access the most recently edited file here. For more information, see Navigation on page 21.

- **Center – Work area**
  The center area is where other work areas for editing are shown such as the Definition tree view. When you open objects, they are shown in the work area in separate Tabs.

- **Bottom – Status bar**
  The status bar shows information such as database activity, project information, the connected database and the current user. For more information, see Status bar on page 21.
**Homepage**

The homepage is displayed upon startup of the API Designer. On the homepage, you can:

- use compilation branches to store different versions of your API in the database.
- Test your API locally
- compile the API and then write it to the database
- use the API Designer to connect and edit C# projects

For more information, see Home page on page 23.

**Detailed information about this topic**

- Menu bar on page 16
- Toolbar on page 19
- Status bar on page 21
- Navigation on page 21
- Home page on page 23

**Menu bar**

You can use the menus on the menu bar to access submenus and to access and execute many functions in API Designer quickly. The following table provides you with information about each menu.

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection</strong></td>
<td></td>
</tr>
<tr>
<td>Settings</td>
<td>Opens the Global settings dialog in the API Designer. Here you can configure the basic settings for API Designer.</td>
</tr>
<tr>
<td>Exit</td>
<td>Closes the program.</td>
</tr>
<tr>
<td><strong>Edit</strong></td>
<td></td>
</tr>
<tr>
<td>Set project type</td>
<td>Opens the Select the project type dialog. Here you can select whether the project should be saved as a database or solution project.</td>
</tr>
</tbody>
</table>

For more information, see Global settings on page 24.

For more information, see Project types on page 12 and Changing the project type on page 12.
<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import objects into database</td>
<td>Opens the <strong>Import objects into database</strong> dialog. Here you can import new and modified objects from your solution project into your One Identity Manager database. For more information, see Importing solution project changes into a database on page 14.</td>
</tr>
<tr>
<td>Change standard project file</td>
<td>From here you can change the solution project that is to be used here.</td>
</tr>
<tr>
<td>Find and replace</td>
<td>Opens the <strong>Find and replace</strong> dialog. Here you can search through documents using certain terms or strings and replace where necessary. For more information, see Find and replace on page 40.</td>
</tr>
<tr>
<td>Find next</td>
<td>Performs the search with the search parameters specified in the <strong>Find and replace</strong> dialog. The search is performed without opening a dialog.</td>
</tr>
<tr>
<td>Captions</td>
<td>Opens the <strong>Multilingual captions</strong> dialog. Here you can create and edit multilingual captions. For more information, see Multilingual captions on page 44.</td>
</tr>
<tr>
<td>Import object</td>
<td>You can use this menu item to import API projects and API files into the API Designer. For more information, see Importing API projects on page 75 and Importing API files on page 80.</td>
</tr>
<tr>
<td>Edit database queries</td>
<td>Opens the <strong>Edit database queries</strong> dialog. Here you can add, change or delete database queries. For more information, see Editing database queries on page 48.</td>
</tr>
<tr>
<td>View</td>
<td></td>
</tr>
<tr>
<td>Tabs</td>
<td>Opens the <strong>Tabs</strong> dialog. You can manage open tabs here. For more information, see Managing tabs on page 52.</td>
</tr>
<tr>
<td>Restore standard layout</td>
<td>Restores the default layout. For more information, see Managing layouts on page 53.</td>
</tr>
<tr>
<td>Restore standard layout (including size)</td>
<td>Restores the default layout and the default window size. For more information, see Managing layouts on page 53.</td>
</tr>
<tr>
<td>Menu</td>
<td>Menu item</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td></td>
<td>Restore saved layout</td>
</tr>
<tr>
<td></td>
<td>Save layout</td>
</tr>
<tr>
<td></td>
<td>Start page</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Solution</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Node editor</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tasks</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command list</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Compilation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bookmarks</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Navigation</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Help**
Menu | Menu item | Description
--- | --- | ---
Community | Opens the One Identity forum.
Support portal | Opens the support portal website.
Training | Opens the training website.
Online documentation | Opens One Identity’s documentation website.
Info | Opens a dialog. Here you get detailed information on the API Designer (system information, version number, information about the software producer, installed modules and so on).

Related topics
- User interface on page 15
- Global settings on page 24
- Changing the project type on page 12
- Find and replace
- Multilingual captions
- Importing API projects
- Editing database queries on page 48
- Managing tabs on page 52
- Managing layouts on page 53
- Home page on page 23
- Managing objects in a project (solution) on page 29
- Node editor view on page 35
- Managing compilation errors and warnings (task window) on page 59
- Displaying the change history (command list) on page 53
- Bookmarks on page 54
- Navigation on page 21
- Opening the compiling log on page 61

Toolbar

The toolbar contains different icons that you can click on to access other functions.
Table 2: Toolbar functions

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Change label" /></td>
<td>In the list, select the label under which changes made to the One Identity Manager database are to be saved. Use the label to make it easier to identify and assign changes to the database. For more information, see Labeling changes on page 37.</td>
</tr>
<tr>
<td><img src="image" alt="Managing change labels" /></td>
<td>Opens the Change label dialog box. You can select, add, change or delete change labels here. For more information, see Labeling changes on page 37.</td>
</tr>
<tr>
<td><img src="image" alt="Use the current change label as default" /></td>
<td>Specify that the current change label see Change label) is to be used as the default change label. This change label will automatically be selected upon API Designer startup. This setting is bound to the client and does not affect other users of the One Identity Manager database. For more information, see Using change labels on page 37.</td>
</tr>
<tr>
<td><img src="image" alt="Captions" /></td>
<td>Opens the Multilingual captions dialog. You can create and edit multilingual captions here. For more information, see Multilingual captions on page 44.</td>
</tr>
<tr>
<td><img src="image" alt="Testing a compilation" /></td>
<td>Tests the compilation for either the Debug or Release version. For more information, see Testing a compilation on page 58.</td>
</tr>
<tr>
<td><img src="image" alt="Save" /></td>
<td>Saves changes to the object currently being edited. If you want to save changes to other database objects, select the corresponding tab in the definition tree view and click the button.</td>
</tr>
<tr>
<td><img src="image" alt="Save all" /></td>
<td>Saves changes to all objects currently being edited.</td>
</tr>
<tr>
<td><img src="image" alt="Previous/next node" /></td>
<td>Navigates backwards and forwards within the history of the selected objects. This displays the selected database object in the definition tree view. If an object or node has been deleted, the next object is shown. If an object that no longer exists is selected in the history, the previous object is shown.</td>
</tr>
</tbody>
</table>

**TIP:** On the menu bar under Connection | Settings, you can specify the number of database objects shown in the history. The history can be deleted in the context menu using Clear history.

**Related topics**

- [User interface](#) on page 15
- [Labeling changes](#) on page 37
- [Multilingual captions](#) on page 44
- [Testing a compilation](#) on page 58
Status bar

The status bar displays different status data. Some status data is shown by way of icons. Which icons are displayed is partially dependent on the program settings selected. The status bar comes in different colors.

Table 3: Meaning of the Colors

<table>
<thead>
<tr>
<th>Color</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Development environment database is connected.</td>
</tr>
<tr>
<td>Red</td>
<td>Simulation mode is enabled.</td>
</tr>
<tr>
<td>Green</td>
<td>Test environment database is connected.</td>
</tr>
<tr>
<td>Yellow</td>
<td>Productive environment database is connected.</td>
</tr>
</tbody>
</table>

Table 4: Status bar icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧑🏼‍♀️</td>
<td>Shows the current user.</td>
</tr>
<tr>
<td>📖</td>
<td>Shows information about the project.</td>
</tr>
<tr>
<td>📁</td>
<td>The database is connected.</td>
</tr>
<tr>
<td>🔄</td>
<td>Shows database access.</td>
</tr>
<tr>
<td>🟢</td>
<td>The database must be compiled.</td>
</tr>
<tr>
<td>🛠️</td>
<td>The program is in simulation mode.</td>
</tr>
</tbody>
</table>

On the status bar, the database is also shown in the following format:

<Server>/<Database (description)>

TIP: To copy the database path into the clipboard, double-click on the database on the status bar.

Double-click on the user on the status bar to see more information about the current user.

Related topics

- User interface on page 15

Navigation

Open the navigation via View | Navigation (see Opening the navigation on page 22).
Use the navigation to manage (create, open, delete) API files and API projects. You can also access the most recently edited file here.

At the bottom of the navigation area you can select whether you would like to manage API files, or API projects.

The following table provides an overview of the various features available within the navigation.

**Table 5: Navigation features**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Add](image) Add | Adds a new database object. The type of database object changes according to the area you are in (API files or API projects). For detailed information, see the following sections:  
  - Creating API projects on page 68  
  - Creating API files on page 78 |
| ![Delete](image) Delete | Deletes a database object. For detailed information, see the following sections:  
  - Deleting API projects on page 70  
  - Deleting API files on page 79 |
| ![Reload data](image) Reload data | Refreshes the view/reloads the data. |
| ![Properties](image) Properties | Opens the Object properties dialog box. Allows you to view the properties for the selected database object. For more information, see Object properties on page 27. |
| Searching | Use the search feature to find specific database objects. Enter the required search term and then press Enter. |

**Related topics**

- User interface on page 15
- Creating API projects on page 68
- Creating API files on page 78
- Deleting API projects on page 70
- Deleting API files on page 79
- Object properties on page 27

**Opening the navigation**

By default, the navigation remains open (on the left of the window) until you close it. You can open the navigation again at any time.
To open the navigation

- Click on View | Navigation on the menu bar.

Related topics

- Navigation on page 21
- User interface on page 15

Home page

Open the homepage via View | Home (see Opening the homepage on page 23).

The homepage is displayed upon startup of the API Designer.

On the homepage, you can:

- use compilation branches to store different versions of your API in the database.
- Test your API locally
- compile the API and then write it to the database
- use the API Designer to connect and edit C# projects

Related topics

- Managing versions (compilation branches) on page 62
- Testing an API on page 64
- Compiling an API on page 57
- Linking C# projects to the API Designer on page 65
- Opening the homepage on page 23
- User interface on page 15

Opening the homepage

You can open the homepage again at any time.

To open the homepage

- Click on View | Home on the menu bar.
  The homepage opens in the work area.
Global settings

Open the global settings via Connection | Settings (see Changing global settings on page 26).

Use the Global settings dialog box to configure basic settings for API Designer. The following table gives an overview of the various features within the Global settings dialog box.

Table 6: Global settings

<table>
<thead>
<tr>
<th>Area</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td>Current language</td>
<td>Shows the language currently in use.</td>
</tr>
<tr>
<td></td>
<td>Language at next startup</td>
<td>Specify the language for the API Designer. The API Designer will open using this language the next time the API Designer starts. For more information, see Changing language on page 26.</td>
</tr>
<tr>
<td></td>
<td>Resolution</td>
<td>Specify the resolution.</td>
</tr>
<tr>
<td></td>
<td>Maximum history length</td>
<td>Specify how many changes can be undone.</td>
</tr>
<tr>
<td></td>
<td>Maximum number of recently opened items</td>
<td>Specify how many items are to be shown under “Recently edited files”</td>
</tr>
<tr>
<td>Editor</td>
<td>Shorten long property values</td>
<td>If you would like to shorten long property values, enable this option.</td>
</tr>
<tr>
<td></td>
<td>Show an error message if syntax errors are found after editing</td>
<td>Enable this option if you want to receive a notification when syntax errors have been found after editing.</td>
</tr>
<tr>
<td></td>
<td>Save and load API Designer size within layout</td>
<td>If you would like to save the size settings for the API Designer view and reload them upon restarting, enable this option.</td>
</tr>
<tr>
<td></td>
<td>Ask for save type after every startup</td>
<td>Enable this option if you would like to select how you would like to save your project each time you start API Designer (see Changing the project type on page 12).</td>
</tr>
<tr>
<td>Area</td>
<td>Setting</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Branch ID</strong></td>
<td></td>
<td>Use compilation branches to store different API versions in the database. Manage you compilation branches. You can select the compilation branch that you want to use, display the existing compilation branches, create compilation branches, edit compilation branches, and delete compilation branches. For more information, see Managing versions (compilation branches) on page 62.</td>
</tr>
<tr>
<td><strong>Compiler</strong></td>
<td>Display the following warnings as errors</td>
<td>Specify which warnings are displayed as errors during compilation. Enter the codes for the warnings, separated by commas. The error codes refer to the Microsoft C#Sharp compiler. For more information about compilation, see Compiling an API on page 57. TIP: For information about error codes that occur during compilation, you can refer, for example, to the Error code table of the Tasks view (see Managing compilation errors and warnings (task window) on page 59).</td>
</tr>
<tr>
<td></td>
<td>Ignore the following errors</td>
<td>Specify which errors should be ignored during compilation. Enter the codes for the warnings, separated by commas. The error codes refer to the Microsoft C#Sharp compiler. For more information about compilation, see Compiling an API on page 57. TIP: For information about error codes that occur during compilation, you can refer, for example, to the Error code table of the Tasks view (see Managing compilation errors and warnings (task window) on page 59).</td>
</tr>
<tr>
<td><strong>Save and backup</strong></td>
<td>Ask for change label</td>
<td>Enable this option if you would like to be asked for a change label when saving.</td>
</tr>
<tr>
<td></td>
<td>Keep backup of unsaved objects on local machine</td>
<td>If you would like to backup unsaved changes on your local hard drive, enable this option. This ensures that your changes are not lost if the program crashes.</td>
</tr>
<tr>
<td><strong>Keyboard layout</strong></td>
<td>Display next bookmark</td>
<td>Specify the shortcut to be used for the next bookmark. TIP: To specify a Ctrl or Shift shortcut, enable the Alt and/or Ctrl options. Finally, select a button from the list.</td>
</tr>
</tbody>
</table>
### Changing global settings

You can change the global/general settings for the API Designer at any time.

**To adjust API Designer settings**

1. Click on **Connection | Settings** on the menu bar.
2. Click on an area in the **Global settings** dialog box to collapse it or expand.
3. Configure the required settings (see **Global settings** on page 24).
4. Click **Apply**.

**Related topics**

- **Global settings** on page 24

### Changing language

The API Designer is available in German and English. You can switch between these languages at any time. Changing the language requires the API Designer to be restarted.

**To change the API Designer language**

1. Click on **Connection | Settings** on the menu bar.
2. In the **Global settings** dialog box, select the required language from the **New startup language** list.
3. Click **Apply**.
4. Confirm the prompt by selecting **Yes** in the **Restart API Designer** dialog box. The API Designer will restart in the selected language.

---

<table>
<thead>
<tr>
<th>Area</th>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Turn bookmarks on/off</td>
<td>Specify the shortcut you want to use to set or remove a bookmark.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIP: To specify a Ctrl or Shift shortcut, enable the Alt and/or Ctrl options. Finally, select a button from the list.</td>
</tr>
</tbody>
</table>

**Related topics**

- **Changing global settings** on page 26
- **Menu bar** on page 16
Database objects

API projects and API files are defined as database objects in API Designer. In this section, you will learn how to work with these database objects API Designer.

Related topics
- Editing database objects (definition tree view) on page 30
- Managing objects in a project (solution) on page 29
- Object properties on page 27

Object properties

Right-click on an object to open the Object properties dialog box (see Displaying object properties on page 28).

The Object properties dialog box displays the properties of objects in the API Designer. The dialog box is divided into five tabs which are explained below.

General

Under the General tab, the general properties of the object such as ID, status, or primary key are displayed.

Properties

Under the Properties tab, all columns in the database object are displayed in a table alongside their values. You can choose between a simple column view and the advanced view with additional data for column definitions.

Permissions

The permissions for each database object are displayed under the Permissions tab. The first entry shows the basic permissions for the table. The permissions for this particular database object are displayed below it. The other entries show the column permissions.

TIP: Double-click on the table entry, the object entry, or a column entry to display the permissions group from which the permissions were determined.
### Table 7: Icon used for permissions

<table>
<thead>
<tr>
<th>Icon</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>✅</td>
<td>Permissions exist.</td>
</tr>
<tr>
<td>●</td>
<td>Permissions have been removed by the object layer</td>
</tr>
<tr>
<td>🍀</td>
<td>Permissions limited by conditions.</td>
</tr>
</tbody>
</table>

#### Label

Under the **Change labels** tab, all the change labels assigned to the database object are displayed. You can also assign a change label to the object.

#### Export

Under the **Export** tab, you can export the database object:

- to the clipboard
- as a file
- via drag&drop
- as an SQL INSERT statement or SQL UPDATE statement

#### Related topics

- [Displaying object properties](page 28)

### Displaying object properties

You can display the properties of a database object at any time.

**To show the properties of a database object**

1. Click on **View | Navigation** on the menu bar.
2. In the navigation, click on the database object whose properties you would like to display.
3. Click 📑.

   The **Object properties** dialog box opens. This dialog box contains several tabs displaying the properties of the selected database object. For more information, see **Object properties** on page 27.

#### Related topics

- [Object properties](page 27)
Managing objects in a project (solution)

Open the Solution view via View | Solution (see Opening the solution on page 30). The solution can be used to display all the objects in the API Designer that can be opened via the navigation.

The solution view depends on your choice of project. If you select a solution project when starting the API Designer, the listed entries will be grouped according to their respective database modules and you will be able to disable the compilation of individual database modules. Database modules whose status has not changed, are shown as collapsed nodes. If you mark a subnode, you can run additional tasks.

You can also identify the objects whose status has changed in API Designer. Differences between changes in the API Designer and changes to resources in the database or on the hard disk are marked.

To open an object, double-click on it in the solution.

The following table provides an overview of the various features available within the Solution window.

Table 8: Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 Update list</td>
<td>Refreshes the list of database objects.</td>
</tr>
<tr>
<td>🔄 Reload selected objects</td>
<td>Reloads the selected objects.</td>
</tr>
<tr>
<td>🔄 Mark all modified objects to be reloaded.</td>
<td>Any objects that you have modified will be reloaded after you click on 🔄 Reload selected objects</td>
</tr>
<tr>
<td>🔄 Disable module compilation.</td>
<td>Disables compilation of the selected module.</td>
</tr>
<tr>
<td>🔄 Only display modified objects</td>
<td>Only modified objects will be displayed in the list.</td>
</tr>
</tbody>
</table>

Table 9: Solution columns

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object identifier</td>
<td>Name of the object</td>
</tr>
<tr>
<td>DB status</td>
<td>Up to date is displayed when the database has been refreshed. Otherwise the user is who made the change is displayed.</td>
</tr>
<tr>
<td>Status drive</td>
<td>Up to date is displayed when the file has been updated. Otherwise a time when the changes will be made is shown.</td>
</tr>
<tr>
<td>Status API Designer</td>
<td>Displays object modifications. The status is shown as Changed if the object has been changed.</td>
</tr>
</tbody>
</table>
**Related topics**

- Opening the solution on page 30
- Menu bar on page 16

**Opening the solution**

You can open the solution at any time.

To open the solution

- Click on View | Solution on the menu bar.

**Related topics**

- Managing objects in a project (solution) on page 29

**Editing database objects (definition tree view)**

Open the definition tree view via View | Navigation | Click the required database object (see Opening the definition tree view on page 31).

Use the definition tree view to display and edit different database objects in API Designer. A separate tab containing a unique definition tree view will be opened for each database object that you edit. A definition tree view opens when adding or editing database objects, for example.

A database object contains what are known as nodes that you can display in a tree structure within the definition tree view and edit in the node editor view (see Node editor view on page 35). If you add a new database object to your project (for example, an API project), this database object is created with a predefined number of nodes and is shown in the object definition. The object definition is a type of view found within the definition tree view.

The predefined nodes contained in a database object form the basic structure of the database object. You cannot delete these nodes. In addition to the predefined nodes, you can add extra nodes via the context menu (right-click). You can delete these nodes.
The following table gives an overview of the various features contained on the toolbar within the definition tree view.

### Table 10: Definition tree view toolbar

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Object definition" /></td>
<td>An overview of nodes contained in a database object can be found here.</td>
</tr>
<tr>
<td><img src="image" alt="Generated code" /></td>
<td>You can view the code generated for a selected node here. For more information, see Viewing and saving generated code belonging to a node on page 34.</td>
</tr>
<tr>
<td><img src="image" alt="Find and replace" /></td>
<td>You can search for terms here and replace them where necessary. For more information, see Find and replace on page 40.</td>
</tr>
</tbody>
</table>

**Related topics**

- [Opening the definition tree view](#) on page 31
- [Viewing and saving generated code belonging to a node](#) on page 34
- [Find and replace](#) on page 40

### Opening the definition tree view

You can open the definition tree view at any time.

**To open the definition tree view of a database object**

1. Click on View | Navigation on the menu bar.
2. In the navigation, double-click on the database object which holds the definitions you would like to display.
   
   The definition tree view for the selected object opens in a separate tab.

**Related topics**

- [Editing database objects (definition tree view)](#) on page 30
- [Navigation](#) on page 21

### Context menu in the definition tree window

To access the context menu, right-click a node in the definition tree window. The options available in the menu depend on the type of node selected (context-sensitive).
Structure

The top section of the context menu lists the node types that can be added to the selected node.

Some node types do not allow additional node types to be pasted into them. In such cases, the Object in extension function is at the top of the context menu. All other functions are either available or grayed out, depending on the type of node and its position in the definition tree window.

Table 11: Functions in the context menu

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object in extension</td>
<td>Allows you to create sub-elements of the selected node in an extension instead of an object.</td>
</tr>
<tr>
<td>Move to extension</td>
<td>Moves the selected node into a new or existing extension.</td>
</tr>
<tr>
<td>✂️ Cut</td>
<td>Cuts the highlighted node and copies it to the clipboard. Any child nodes for the selected node are also cut.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You cannot cut any nodes that have been automatically added to the definition tree.</td>
</tr>
<tr>
<td>✉️ Copy</td>
<td>Copies the highlighted and child nodes to the clipboard.</td>
</tr>
<tr>
<td>✂️ Paste</td>
<td>Pastes the node contained in the clipboard.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You can only paste nodes contained in the clipboard if they are also permitted at the selected point. Example: You cannot paste a &quot;Plugin&quot; node below the &quot;Compilation settings&quot; node.</td>
</tr>
<tr>
<td>✂️ Delete</td>
<td>Deletes the highlighted node and all child nodes.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You cannot cut any nodes that have been automatically added to the definition tree.</td>
</tr>
<tr>
<td></td>
<td><strong>TIP:</strong> You can select multiple nodes by holding down the Ctrl key to apply the function once to all selected nodes.</td>
</tr>
<tr>
<td>Delete node as an extension</td>
<td>Deletes the highlighted node as an extension.</td>
</tr>
<tr>
<td>✓️ Set/remove bookmark</td>
<td>Sets or removes a bookmark on the highlighted node. For more information, see Bookmarks on page 54</td>
</tr>
<tr>
<td>Expand all</td>
<td>Expands all nodes.</td>
</tr>
<tr>
<td>Collapse all</td>
<td>Collapses all nodes.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Export</td>
<td>Saves the selected node and their child nodes in XML format to your hard drive.</td>
</tr>
<tr>
<td>Import</td>
<td>Pastes previously exported child nodes at the selected point.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You can only paste nodes if they are also permitted at the selected point. Example: You cannot paste a &quot;Plugin&quot; node below the &quot;Compilation settings&quot; node.</td>
</tr>
<tr>
<td>Move up</td>
<td>Moves the selected node upwards.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You cannot move any nodes that have been automatically added to the definition tree.</td>
</tr>
<tr>
<td>Move down</td>
<td>Moves the selected node downwards.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> You cannot move any nodes that have been automatically added to the definition tree.</td>
</tr>
<tr>
<td>Undo</td>
<td>Undoes the previous action.</td>
</tr>
<tr>
<td>Redo</td>
<td>Redoes the last action that has been undone.</td>
</tr>
<tr>
<td>Search object references</td>
<td>Searches for all references that refer to the selected node.</td>
</tr>
<tr>
<td></td>
<td><strong>NOTE:</strong> If searching for a reference within a collection, the results are grouped by related columns.</td>
</tr>
<tr>
<td>Search</td>
<td>Opens the <strong>Find and replace</strong> dialog. For more information, see Find and replace on page 40.</td>
</tr>
<tr>
<td>Search next</td>
<td>This function continues the search using the current search parameters and highlights the next matching node. The search is executed even if the <strong>Search</strong> dialog is not open.</td>
</tr>
</tbody>
</table>

**Related topics**

- Editing database objects (definition tree view) on page 30
- Bookmarks on page 54
- Find and replace on page 40

**Extensions**

Use extension to indirectly edit base objects (standard database objects that are contained in your project). You can add as many extensions as you wish to the base objects in the API Designer.

Extensions are edited when you configure the base objects. This means you add an extension to the object and change the property value. The base object is subsequently
compiled and the modification resulting from the extension is highlighted in the API Designer.

Base objects to which you want to add an extension can be identified as colored nodes in the Object definitions view in the Definition tree view.

**Figure 1: Workflow in the API Designer**

![Workflow diagram](image)

**Viewing and saving generated code belonging to a node**

In the Definition tree view, you can view a node’s generated code (within a database object) and copy or save where necessary. This function is particularly useful if you are looking at code in detail and want to reuse parts of the code, or you want to examine an error in more detail.

- **NOTE:** You cannot edit the code.
- **NOTE:** If you want to search for a specific point in the code, you can use the shortcut Ctrl + F.

**To view a node’s generated code**

1. Click on **View | Navigation** on the menu bar.
2. In the navigation, double-click on the database object that contains the node you require.
3. Click on the node in the definition tree view whose code you want to see.
4. Click **охранить**. The code will be displayed in a code view.

**To save the generated code as a file**

1. Execute the steps previously described.
2. In the code view, select the code that you want to save.
3. Right-click on the highlighted code.
4. Click **Copy**.
5. Click **Save as** in the context menu.
6. Select a name and location for the file in the save dialog box and click **Save**.

**Related topics**
- Editing database objects (definition tree view) on page 30
- Resolving errors and warnings on page 61

**Node editor view**

Open the node editor view via **View | Node editor** (see Opening the node editor view on page 36).

The **Node editor** view allows you to edit the properties of a node that you have selected in the Definition tree view (see Editing nodes on page 36).

**NOTE:** The settings shown depend on the type of node selected.

The following table provides an overview of the various features available within the **Node editor** window.

**Table 12: Controls**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![And name / icon of the base object](image) | Displayed if a file-based database object (**Solution project**) has been selected.  
Click on ![Open corresponding folder](image) to open the folder on your drive where the database object is located.  
Right-click on ![Open corresponding folder](image) to open a context menu.  
For more information, see **Solution projects** on page 14. |
| ![Database](image)                          | Displayed if a database object is selected.                                  |
| ![Delete filter](image)                     | Deletes the filter entered into the next field.                             |
| ![Filter field](image)                      | Displays only fields that match the text which has been entered.  
Example: if you enter **Pe**, only the **Permission** field will be displayed.  
**NOTE:** The API Designer only filters the settings fields and not the values contained within them. |
### Control

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 Sort ascending alphabetically</td>
<td>Sorts the settings alphabetically in ascending order within each category. If you enable sorting, the button will be highlighted with a colored frame.</td>
</tr>
<tr>
<td>🔄 Sort descending alphabetically</td>
<td>Sorts the settings alphabetically in descending order within each category. If you enable sorting, the button will be highlighted with a colored frame.</td>
</tr>
<tr>
<td>⚫ View grouping</td>
<td>Enables/disables the display of the various settings in categories. If you enable this option, the button will be highlighted with a colored frame.</td>
</tr>
<tr>
<td>🔄 Expand all</td>
<td>Expands all categories.</td>
</tr>
<tr>
<td>🔄 Collapse all</td>
<td>Collapses all categories.</td>
</tr>
<tr>
<td>📝 Highlights</td>
<td>Highlights setting that have had an extension added to them. Use 📝 to switch to the extension. A node that has had an extension added to it, is indicated in the definition tree view with the 📝 icon. If you enable this option, the button will be highlighted with a colored frame.</td>
</tr>
</tbody>
</table>

### Related topics

- Opening the node editor view on page 36
- Editing nodes on page 36
- Menu bar on page 16
- Editing database objects (definition tree view) on page 30

### Opening the node editor view

You can open the node editor view at any time.

**To open the node editor view**

- Click on View | Node editor on the menu bar.

### Related topics

- Editing nodes on page 36
- Node editor view on page 35

### Editing nodes

Node properties can be edited for any node in the node editor view.
To edit a node

1. Click on View | Navigation on the menu bar.
2. In the navigation, double-click on the object that you would like to edit.
3. Click on View | Node editor on the menu bar.
   The Node editor view opens.
4. In the Definition tree view, click on the node that you would like to edit.
   The fields and options shown in the node editor view will change depending on the selected node.
5. Make the changes in the Node editor view.
6. Click on Save on the toolbar.

Related topics
- Opening the node editor view on page 36
- Node editor view on page 35
- Editing database objects (definition tree view) on page 30
- Menu bar on page 16
- Navigation on page 21

Labeling changes

Use Change label to make it easier to view and assign changes in the database. Any database objects that yield a project are entered on a change label. Database Transporter is used for moving the project. You can create and edit change labels in different One Identity Manager tools.

Related topics
- Using change labels on page 37
- Creating a change label on page 38
- Change management on page 39
- Deleting change labels on page 40

Using change labels

In order to also be able to assign a change label to changes, you need to select a change label for further use.
To use a change label

- On the toolbar, select the required change label in the Change label list.
  Any changes stored in the project during this session will be assigned to the selected change label.

  TIP: If you are always using the same change label and do not want to select it every time you start the API Designer, click on Use the current change label as default in the toolbar.

Related topics

- Labeling changes on page 37
- Creating a change label on page 38
- Change management on page 39
- Deleting change labels on page 40

Creating a change label

You can create as many change labels as you wish and use them to track your changes.

To edit a change label

1. Click Manage change label on the toolbar.
2. Click Create a new change label in the Change label dialog.
3. Specify the properties of the change label in the right-hand column in the list.
   - Change label: Enter a name for the change label.
   - Description: (optional) Enter a description for the change label.
   - Locked: (Optional) Select whether you would like to block the change label from further use. If a change label is locked, no further changes can be booked to this label.
   - Comment: (optional) Enter a comment to monitor changes to the change label.
   - Status: (optional) Select a status from the list.
   - Status comments: (optional) Enter a comment in relation to the status.
   - Parent change label: (optional) Select a change label from the list to be the parent.
4. Click Save change label above the list.
5. Click OK.
Related topics

- Labeling changes on page 37
- Using change labels on page 37
- Change management on page 39
- Deleting change labels on page 40

Change management

You can edit existing change labels at any time.

To edit a change label

1. Click Manage change label on the toolbar.
2. In the Change label dialog box, click on the label to be edited.
3. Click Show / hide edit view .
4. Change the properties of the change label in the right-hand column in the list:
   - **Change label**: Enter a name for the change label.
   - **Description**: (optional) Enter a description for the change label.
   - **Locked**: (Optional) Select whether you would like to block the change label from further use. If a change label is locked, no further changes can be booked to this label.
   - **Comment**: (optional) Enter a comment to monitor changes to the change label.
   - **Status**: (optional) Select a status from the list.
   - **Status comments**: (optional) Enter a comment in relation to the status.
   - **Parent change label**: (optional) Select a change label from the list to be the parent.
5. Click Save change label above the list.
6. Click OK.

Related topics

- Labeling changes on page 37
- Using change labels on page 37
- Creating a change label on page 38
- Deleting change labels on page 40
Deleting change labels

You can delete existing change labels at any time.

To edit a change label

1. Click Manage change label on the toolbar.
2. In the Change label dialog box, click on the change label you would like to delete.
3. Click Delete the selected change label.
4. Confirm the prompt by selecting Yes in the dialog box.
5. Click Cancel.

Related topics

- Labeling changes on page 37
- Using change labels on page 37
- Creating a change label on page 38
- Change management on page 39

Find and replace

Open the search function over Edit | Find and replace (see Performing a search on page 43).

Use the Find and replace dialog box to search for (and replace) certain captions or items within your project. The following table gives an overview of the various features within the Find and replace dialog box.

Table 13: Controls for find and replace

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>Enter a search term.</td>
</tr>
<tr>
<td></td>
<td><strong>TIP</strong>: To reuse terms from previous searches, click on the arrow to the right of the field and select the required term.</td>
</tr>
<tr>
<td>Find as</td>
<td>In the Find as list, select whether you would like to search using simple text or by wildcards or regular expressions.</td>
</tr>
<tr>
<td>Replace with</td>
<td>(Optional) Enter the text which is to replace the searched text.</td>
</tr>
<tr>
<td>Search area</td>
<td>If you would like to further narrow the search, you can select which objects you would like to search for here:</td>
</tr>
<tr>
<td>Control</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Current document</td>
<td>Only the current document is included in the search.</td>
</tr>
<tr>
<td>Current document and its extensions</td>
<td>Only the current document and all of its extensions are included in the search.</td>
</tr>
<tr>
<td>Current document and its parent documents</td>
<td>Only the current document and all of its parent items are included in the search.</td>
</tr>
<tr>
<td>Below the selected object</td>
<td>Only the objects below the current object are included in the search.</td>
</tr>
<tr>
<td>All API designer objects</td>
<td>All objects are included in the search.</td>
</tr>
</tbody>
</table>

Search options Configure other search settings:

- **Case sensitive**: API Designer finds only occurrences of the text that match the case that you have entered into the Search field.
- **Whole word**: Only the whole word is included in the search. Example: If you enter *Person*, the search will only find “person” and not “persons”.
- **Match entire value**: API Designer finds only objects containing the exact values that match the text you have entered into the Search field.
- **Filter by type**: Select the objects to which your search will be limited.

Search Finds and opens the next occurrence of the term.

Find all Triggers a search that lists all occurrences of the text in the Search results area.

Search results Lists all occurrences of the term.

Replace Replaces the occurrence of the term that you have highlighted in Search results with the text that you have entered in the Replace field.

Replace all Replaces all occurrences of the term that you have selected in the Replace column under Search results with the text that you have entered in the Replace field.

**Wildcards**

Use wildcards to replace a single character, or a string of characters, using a single character when searching for strings. The most common wildcards are the question mark (?) - to symbolize an individual character - and the asterisk (*) - to symbolize any combination of characters.
To use wildcards in the **Find and replace** dialog box, enter the search term into the **Find** field and select the **Wildcards** option from the **Find as** list.

### Example
The following table gives some examples of wildcard searches:

<table>
<thead>
<tr>
<th>Sample term</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>P*&lt;i&gt;n&lt;/i&gt;</td>
<td>Finds &quot;person&quot;, &quot;position&quot;, &quot;plugin&quot; and so on</td>
</tr>
<tr>
<td>Per*</td>
<td>Finds &quot;person&quot;, &quot;personal&quot;, &quot;perfection&quot; and so on</td>
</tr>
<tr>
<td>AP?.json</td>
<td>Finds &quot;API.json&quot; but not &quot;APIProject.json&quot;</td>
</tr>
</tbody>
</table>

### Regular expressions

Regular expressions (also known as 'regex' or 'regexp') are similar to wildcards in that they also allow you to find strings. Regular expressions are more effective than wildcards, however.

To use regular expressions in the **Find and replace** dialog box, enter the search term into the **Search** field and select the **Regular expressions** option from the **Find as** list.

### Example
The following table gives some examples of regular expressions:

<table>
<thead>
<tr>
<th>Regular expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[a-z]</td>
<td>Any lowercase Latin letter</td>
</tr>
<tr>
<td>[A-Z]</td>
<td>Any uppercase Latin letter</td>
</tr>
<tr>
<td>\d</td>
<td>A digit</td>
</tr>
<tr>
<td>\D</td>
<td>A character that is not a digit</td>
</tr>
<tr>
<td>\w</td>
<td>A letter, a digit or an underscore</td>
</tr>
<tr>
<td>\W</td>
<td>A character that is neither a letter, a number nor an underscore</td>
</tr>
<tr>
<td>\s</td>
<td>Blank</td>
</tr>
<tr>
<td>\S</td>
<td>A character that is not a blank space</td>
</tr>
<tr>
<td>{n}</td>
<td>The preceding term must occur exactly ‘n’ number of times.</td>
</tr>
</tbody>
</table>
Regular expression | Description
--- | ---
Example: C{3} finds “CCC”
| Alternatives
Example: Regex|Regexp finds "Regex" or "Regexp"
? The preceding term is optional, it can occur once, but does not need to, i.e. the term either occurs once or not at all.

Related topics
- Performing a search on page 43
- Menu bar on page 16

Performing a search

You can perform a search of the whole project, or parts of it, at any time.

To run a search

1. Click Edit | Find and replace on the menu bar.
2. In the Find and replace dialog, enter a search term in the Find field.
3. In the Find as list, select whether you would like to search using simple text or by wildcards or regular expressions.
4. (Optional) In the Replace by field, enter the text to replace the text in the search.
5. In the Find scope list, select which objects you would like to include in the search.
6. (Optional) In the Find options area, click to configure further search settings.
7. Click Find or Find all.
   The search results are displayed in the Find results area
8. Click in the Find options area.
9. (Optional) Double-click a result in the results list.
   The relevant nodes are highlighted and displayed in the definition tree view.
10. (Optional) To replace a result, highlight it in the Find results area and click Replace.
11. (Optional) To replace several results, enable the checkbox next to the relevant results and click Replace all.
Related topics

- Find and replace on page 40

Multilingual captions

Open the Multilingual captions dialog box via Captions.

Use the Multilingual captions dialog box to add, edit or delete captions in multiple languages. You can use these texts later in your API or web application.

Before adding or editing multilingual captions, define objects in your project that output captions in your API. You create keys for these objects. You assign a text to the keys for each language. This means that the keys are translated into the different languages you wish to use.

You can create and edit keys and translations on the Captions tab in the Multilingual captions dialog box. The following table gives an overview of the various features within the Multilingual captions dialog box.

**Table 14: Controls**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td>Creates a new caption. For more information, see Creating multilingual captions on page 45.</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the caption selected in the results list. For more information, see Deleting a multilingual caption on page 47.</td>
</tr>
<tr>
<td>Save</td>
<td>Saves the changes. For more information, see Editing a multilingual caption on page 46.</td>
</tr>
<tr>
<td>Search mask</td>
<td>In this area, you can search for existing captions and edit them. For more information, see Searching for a multilingual caption on page 45.</td>
</tr>
<tr>
<td>Result list</td>
<td>Lists the results generated by the search. For more information, see Searching for a multilingual caption on page 45.</td>
</tr>
<tr>
<td>Editing</td>
<td>You can edit captions in this area. For more information, see Creating multilingual captions and Editing a multilingual caption on page 46.</td>
</tr>
</tbody>
</table>

Related topics

- Creating multilingual captions on page 45
- Searching for a multilingual caption on page 45
- Editing a multilingual caption on page 46
- Deleting a multilingual caption on page 47
- Menu bar on page 16
Creating multilingual captions

You can create multilingual captions at any time.

To create a multilingual caption
1. Click Captions on the menu bar.
2. Click Add on the toolbar in the Multilingual captions dialog box.
3. In the Edit area, enter a unique value in the Key field to be used to reference the text.
   
   **NOTE:** If you have not added a translation for a language, the API will use the caption that has been saved under Key.
4. Select the caption’s language in the Language list.
5. Enter the caption to be shown for the selected language in the Text field.
6. Repeat the last two steps for all the required languages.
7. Click Save on the toolbar in the Multilingual captions dialog box.
8. Click Apply.

Related topics
- Multilingual captions on page 44
- Searching for a multilingual caption on page 45
- Editing a multilingual caption on page 46
- Deleting a multilingual caption on page 47

Searching for a multilingual caption

You can search for multilingual captions at any time.

To search for a multilingual caption
1. Click Captions on the menu bar.
2. In the Multilingual captions dialog box, enter a term into the Search form (such as the name of a key or parts of the caption).
3. Use the options below the search bar to limit your search:
   - **Search key and value:** Searches for the term in the key and captions.
   - **Search for key only:** Only searches for the term in the key.
   - **Search for value only:** Only searches for the term in the captions.
   - **Search in all available languages:** Enable this option if you would like to
search for the term across all languages. If you disable this option, the search will only be performed in the language that is current shown in the Language field in the Edit area.

4. Click Search.

The search results are displayed in the Result list area.

   TIP: If one of the Search key and value or Search for key only options is set, the keys shown in the result list are labeled with an asterisk (*).

Related topics

- Multilingual captions on page 44
- Creating multilingual captions on page 45
- Editing a multilingual caption on page 46
- Deleting a multilingual caption on page 47

**Editing a multilingual caption**

You can edit existing multilingual captions at any time.

**To edit a multilingual caption**

1. Click Captions on the menu bar.
2. In the Multilingual captions dialog box, enter the search term in the Search form area.
3. Use the options below the search bar to limit your search:
   
   - Search key and value: Searches for the term in the key and captions.
   - Search for key only: Only searches for the term in the key.
   - Search for value only: Only searches for the term in the captions.
   - Search in all available languages: Enable this option if you would like to search for the term across all languages. If you disable this option, the search will only be performed in the language that is current shown in the Language field in the Edit area.
4. Click Search.
5. In the Result list area, click on the caption to be edited.

   TIP: If one of the Search key and value or Search for key only options is set, the keys shown in the result list are labeled with an asterisk (*).
6. In the Edit area, change the unique value in the Key field that is used to reference the text.
NOTE: If you have not added a translation for a language, the API will use the caption that has been saved under **Key**.

7. Select the caption’s language in the **Language** list.
8. Enter the caption to be shown for the selected language in the **Text** field.
9. Repeat the last two steps for any languages that you would like to change or add.
10. Click **Save** on the toolbar in the **Multilingual captions** dialog box.
11. Click **Apply**.

**Related topics**

- Multilingual captions on page 44
- Creating multilingual captions on page 45
- Searching for a multilingual caption on page 45
- Deleting a multilingual caption on page 47

**Deleting a multilingual caption**

You can delete existing multilingual captions at any time.

NOTE: You cannot delete multilingual captions that have been predefined by API Designer.

**To delete a multilingual caption**

1. Click **Captions** on the menu bar.
2. In the **Multilingual captions** dialog box, enter a term into the **Search form** (such as the name of a key or parts of the caption).
3. Use the options below the search bar to limit your search:
   - **Search key and value**: Searches for the term in the key and captions.
   - **Search for key only**: Only searches for the term in the key.
   - **Search for value only**: Only searches for the term in the captions.
   - **Search in all available languages**: Enable this option if you would like to search for the term across all languages. If you disable this option, the search will only be performed in the language that is current shown in the **Language** field in the **Edit** area.
4. Click **.**
5. In the **Result list** area, click on the caption to be deleted.

TIP: If one of the **Search key and value** or **Search for key only** options is set, the keys shown in the result list are labeled with an asterisk (*).
6. Click 🗑️ Delete on the toolbar in the Multilingual captions dialog box.
7. Confirm the prompt by selecting Yes in the dialog box.
8. Click Apply.

Related topics
- Multilingual captions on page 44
- Creating multilingual captions on page 45
- Editing a multilingual caption on page 46
- Searching for a multilingual caption on page 45

Editing database queries

Open the Edit database queries dialog box via Edit | Edit database queries (see Displaying database queries on page 49).

In the Edit database queries dialog box, you can display, create, edit, delete and test database queries.

Database queries enable secure communication between your web application and your One Identity Manager database. The actual SQL code is saved in SQL snippets, to which only certain parameters can be added later. The database queries are referenced later in the code only based on their name.

The following table gives an overview of the various features within the Edit database queries dialog box.

Table 15: Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>📄 Add</td>
<td>Creates a new database query. For more information, see Creating a database query on page 49.</td>
</tr>
<tr>
<td>🗑️ Delete</td>
<td>Deletes the selected database query. For more information, see Deleting a database query on page 52.</td>
</tr>
<tr>
<td>📝 Save</td>
<td>Saves changes to the selected database query. For more information, see Editing a database query on page 50.</td>
</tr>
<tr>
<td>📝 Save all</td>
<td>Saves all changes. For more information, see Editing a database query on page 50.</td>
</tr>
<tr>
<td>✖️ Discard object changes</td>
<td>Undoes all changes to the selected database query.</td>
</tr>
<tr>
<td>Identifier</td>
<td>Unique database query ID</td>
</tr>
</tbody>
</table>
### Control

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>A description of the database query that explains the query and its function</td>
</tr>
<tr>
<td>SQL expression</td>
<td>The actual query in SQL code</td>
</tr>
<tr>
<td>Dialog groups</td>
<td>Dialog groups that are permitted to use this database query (labeled ☑) or forbidden from using it (labeled ☒).</td>
</tr>
<tr>
<td>Test statement</td>
<td>Opens the <strong>Test statement</strong> dialog. You can test the database query here. For more information, see Testing a database query on page 51.</td>
</tr>
</tbody>
</table>

### Related topics

- Displaying database queries on page 49
- Creating a database query on page 49
- Editing a database query on page 50
- Testing a database query on page 51
- Deleting a database query on page 52

### Displaying database queries

You can edit all existing database queries at any time.

**To show all database queries**

- Click **Edit | Edit database queries** on the menu bar.

### Related topics

- Editing database queries on page 48
- Creating a database query on page 49
- Editing a database query on page 50
- Testing a database query on page 51
- Deleting a database query on page 52

### Creating a database query

You can create new database queries at any time.
To create a database query

1. Click Edit | Edit database queries on the menu bar.
2. Click Add in the Edit database queries dialog box.
3. Enter a unique name for the database query in the Identifier field.
4. (Optional) In the Description field, enter a description for the database query that describes the database query and its function.
5. Enter the query as an SQL code in the SQL expression field.
6. Double-click the dialog groups which may use the database query in the Dialog groups area.
   TIP: To select or deselect all dialog groups quickly, click Select all or Deselect all.
7. (Optional) Click Test statement to test the SQL statement with different values. For more information, see Testing a database query on page 51.
8. Click Save.
9. Click Close.

Related topics

- Editing database queries on page 48
- Displaying database queries on page 49
- Editing a database query on page 50
- Testing a database query on page 51
- Deleting a database query on page 52

Editing a database query

NOTE: You cannot edit any database queries that have been predefined by the API Designer.

You can edit existing database queries at any time.

To edit a database query

1. Click Edit | Edit database queries on the menu bar.
2. Click on the database query you would like to edit in the Edit database queries dialog box.
3. Enter a unique name for the database query in the Identifier field.
4. (Optional) In the Description field, enter a description for the database query that describes the database query and its function.
5. Enter the query as an SQL code in the **SQL expression** field.

6. In the **Dialog groups** area, double-click the dialog groups for which access permissions are to be changed.

   **TIP:** Click **Select all** or **Deselect all** to quickly select or deselect all dialog groups.

7. (Optional) Click **Test statement** to test the SQL statement with different values. For more information, see **Testing a database query** on page 51.

8. Click **Save**.

9. Click **Close**.

**Related topics**
- Editing database queries on page 48
- Displaying database queries on page 49
- Creating a database query on page 49
- Testing a database query on page 51
- Deleting a database query on page 52

**Testing a database query**

You can test database queries at any time.

_To test a database query_

1. Click **Edit | Edit database queries** on the menu bar.

2. In the **Edit database queries** dialog box, click on the database query you would like to test.

3. Click **Test statement**.

4. In the **Test statement** dialog box, enable the checkbox next to the SQL statement parameters that you would like to include in the test.

5. Enter the values for each parameter in the **Value** column.

6. Click **Test statement**.

7. Click **Close**.

8. Click **Close** in the **Edit database queries** dialog box.

**Related topics**
- Editing database queries on page 48
- Displaying database queries on page 49
Deleting a database query

NOTE: You cannot delete any database queries that have been predefined by the API Designer. The Delete button is disabled for such database queries.

You can delete existing database queries at any time.

To delete a query

1. Click Edit | Edit database queries on the menu bar.
2. Click on the database query you would like to delete in the Edit database queries dialog box.
3. Click Delete.
4. Confirm the prompt by selecting Yes in the dialog box.
5. Click Close in the Edit database queries dialog box.

Related topics

- Editing database queries on page 48
- Displaying database queries on page 49
- Creating a database query on page 49
- Editing a database query on page 50
- Testing a database query on page 51

Managing tabs

Use the Tabs dialog box to enable opened tabs, close tabs, or save changes that you have performed in tabs.

To manage tabs

1. Click on View | Tabs on the menu bar.
2. Click one or more tabs (while holding down the Ctrl key) in the Tabs dialog box.
3. Perform one of the following tasks:
- Click **Enable** to enable the tab.
- Click **Save** to save tab changes.
- Click **Close tabs** to close the tab.

4. Click **Close**.

**Related topics**
- **Menu bar** on page 16

**Managing layouts**

You can adjust, save and, where necessary, restore the layout (assign and show different menus and areas) at any time.

**To save your own layout**

1. Customize the layout of the API Designer to suit your needs.
2. Click on **View | Save layout** on the menu bar.

**To restore a saved layout**

- Click on **View | Save layout** on the menu bar.

**To restore the default layout**

- Click on **View | Restore standard layout** on the menu bar.

**To restore the default layout including the window size**

- Click on **View | Restore standard layout** on the menu bar.

**Related topics**
- **Menu bar** on page 16

**Displaying the change history (command list)**

Open the command list via **View | Command list** (see **Opening the command list** on page 54).

Use the **Command list** window to display any changes performed on an object and to either undo or redo these.
Executed commands are indicated with the ✔️ icon. This icon is not shown for commands that were undone.

The use of wizards allows numerous commands to be automatically implemented. They are displayed in the command list as composite commands. The individual commands are shown at a second level. You can only undo the composite commands.

The following table gives an overview of the various features available in the Command list window.

Table 16: Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🔄 Undo</td>
<td>Undoes the last implemented command in the list.</td>
</tr>
<tr>
<td>🔄 Redo</td>
<td>Redoes the last command in the list that was undone.</td>
</tr>
</tbody>
</table>

Related topics

- Opening the command list on page 54

Opening the command list

You can view a list of actions that have been taken at any time.

To open the command list

- Click on View | Command list on the menu bar.

Related topics

- Displaying the change history (command list) on page 53

Bookmarks

Open the Bookmarks window via View | Bookmarks (see Editing bookmarks on page 55).

To make navigating around the API Designer easier and to find objects quickly, you can set bookmarks to any node in the Definition tree view. View, manage and use these bookmarks in the Bookmarks window.

The following table provides an overview of the various features available within the Bookmarks window.
Table 17: Controls

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🗑 Delete all bookmarks</td>
<td>Deletes all bookmarks (see Deleting bookmarks on page 56).</td>
</tr>
<tr>
<td>⌨ Edit bookmark description</td>
<td>Opens a dialog box which allows you to change the description of the selected bookmark (see Editing bookmarks on page 55).</td>
</tr>
</tbody>
</table>
| 🗑 Remove module/component bookmark | Deletes all bookmarks for the module (or component) selected in the list (see Deleting bookmarks on page 56).  

**NOTE:** This feature is only available if you selected to group the bookmarks by module/component earlier (see Group by modules/components button). |
| 🗑 Delete current bookmark | Deletes the bookmark that you have selected in the list (see Deleting bookmarks on page 56). |
| 🗑 Group by modules/components | Groups the bookmarks by the API projects to which they are assigned. |

Related topics

- Editing bookmarks on page 55
- Setting bookmarks on page 56
- Deleting bookmarks on page 56

## Editing bookmarks

You can show all bookmarks, delete all bookmarks, or change the description of bookmarks at any time.

**To open the bookmark window**

- Click on View | Bookmarks on the menu bar.

**To open a bookmark**

1. Click on View | Bookmarks on the menu bar.
2. Double-click on a bookmark in the Bookmarks window.

**To edit the bookmark description**

1. Click on View | Bookmarks on the menu bar.
2. Click on a bookmark in the Bookmarks window.
3. Click ✂ Edit bookmark description.
4. Insert a description for the bookmark in the **Bookmark description** dialog box.
5. Click **OK**.

**Related topics**
- [Bookmarks](#) on page 54
- [Setting bookmarks](#) on page 56
- [Deleting bookmarks](#) on page 56

## Setting bookmarks

You can set a bookmark for database objects at any time. This allows you to things such as quickly access frequently used nodes.

**To set a bookmark**

1. Click on **View | Navigation** on the menu bar.
2. In the navigation, double-click on the database object that contains the node you require.
3. In the definition tree view, right-click on the node for which you would like to set a bookmark.
4. Click ![Set bookmark](...) in the context menu.
5. Enter a description for the bookmark in the **Bookmark description** dialog box.
6. Click **OK**.

**Related topics**
- [Bookmarks](#) on page 54
- [Editing bookmarks](#) on page 55
- [Deleting bookmarks](#) on page 56

## Deleting bookmarks

You can delete single bookmarks, all bookmarks within a module or all bookmarks within a project at any time.

**To delete a single bookmark**

1. Click on **View | Bookmarks** on the menu bar.
2. Click on the bookmark that you would like to delete in the **Bookmarks** window.
3. Click on \(\text{Delete current bookmark}\).
4. Confirm the prompt by selecting \textbf{Yes} in the dialog box.

\textbf{To delete all the bookmarks in a module/component}

1. Click on \textit{View | Bookmarks} on the menu bar.
2. Click on \textit{Group by modules/components} in the \textit{Bookmarks} window.
3. Click on the module whose bookmarks you would like to delete.
4. Click on \textit{Remove bookmarks from module / component}.
5. Confirm the prompt by selecting \textbf{Yes} in the dialog box.

\textbf{To remove all bookmarks}

1. Click on \textit{View | Bookmarks} on the menu bar.
2. Click on \textit{Delete all bookmarks} in the \textit{Bookmarks} window.
3. Confirm the prompt by selecting \textbf{Yes} in the dialog box.

\textbf{Related topics}

- \textit{Bookmarks} on page 54
- \textit{Editing bookmarks} on page 55
- \textit{Setting bookmarks} on page 56

\textbf{Compiling an API}

\textbf{TIP:} Also see the further configuration options for compilation directly on the API project (see \textit{Configuring the compilation} on page 74 and in the \textit{global settings}).

Before you can use an API which has been created with the API Designer, the API must be compiled using the API Designer compiler.

The API Designer compiler can be opened from the API Designer, as well as from the Database Compiler. For more detailed information on Database Compiler, see \textit{One Identity Manager Operational Guide}.

A web project must be compiled in the following cases:

- After changing a definition (API file or API project) in the API Designer.
- After changing certain system settings which require Database Compiler to be run.

The API Designer compiler creates a set of DLL files from the project's XML definition and saves them in the database.

\textbf{Methods of compiling}

There are two ways to compile a web project:
- Run a **release compilation** to release a specific version of the project for use.
  
  **NOTE:** Changes within the API Designer do not affect the API as long as there is no release compilation.

- Use a **debug compilation** during the development phase for testing and debugging purposes. Debug compilation creates additional code to support the API Designer debugger. This means the DLL files are somewhat larger.

An API loads the latest compiled DLL files on startup. If these DLL files are updated, the web project reloads the new DLL files; however, only new sessions run with the code from the newly loaded DLLs.

### Related topics

- Testing a compilation on page 58
- Starting a compilation on page 59
- Managing versions (compilation branches) on page 62
- Managing compilation errors and warnings (task window) on page 59
- Resolving errors and warnings on page 61
- Opening the compiling log on page 61
- Configuring the compilation on page 74
- Global settings on page 24

### Testing a compilation

The two types of compilation (**Debug** and **Release** compilation) can be tested in advance.

**To test the compilation**

1. Click on **View** | **Start page** on the menu bar.
2. Expand the **Compilation** area on the homepage.
3. In the **Compilation** area, perform one of the following actions:
   - To test a debug compilation, enable the **DEBUG** option.
   - To test a release compilation, enable the **RELEASE** option.
4. Click **Test compilation**.
   
   The **Compilation log** opens and shows compilation status and progress.

   The **Tasks** window opens. Use this to view and rectify any errors and warnings that arise during compilation. For more information, see Managing compilation errors and warnings (task window) on page 59.
Starting a compilation

You can start two types of compilation (Debug and Release). If compilation completes without any errors, then the compiled API is written to the database.

To start compilation

1. Click on View | Start page on the menu bar.
2. Expand the Compilation area on the homepage.
3. In the Compilation area, perform one of the following actions:
   - To start a debug compilation, enable the DEBUG option.
   - To start a release compilation, enable the RELEASE option.
4. Click Compile.
   The Compilation log opens and shows compilation status and progress.
   The Tasks window opens if errors arise. Use this to view and rectify any errors and warnings that arise during compilation. For more information, see Managing compilation errors and warnings (task window) on page 59.

Managing compilation errors and warnings (task window)

Open the task window via View | Tasks (see Opening the task window on page 60).
Use the Tasks window to view compilation errors and warnings and to rectify them where necessary (debug).
• **Compilation errors** prevent the web project from compiling properly and must therefore be eliminated. Development states cannot be released if they cannot be compiled.

• **Compilation warnings** relate to missing extensions, or to messages concerning accessibility. The compilation warnings of the relevant compiler are also taken into account. If compilation warnings are the only type of message generated, the development state will still be compiled successfully.

  | TIP: In the global settings, you can define which warnings are displayed as errors during compilation and which errors are ignored. For more information, see Global settings on page 24 |

If individual messages are displayed (icon) together, it means that the errors indicated occur at various locations throughout the web project. This can happen, for example, if an extension that is referenced by a number of nodes is missing.

The following table gives an overview of the various features within the **Tasks** window.

**Table 18: Controls**

<table>
<thead>
<tr>
<th>Control</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>Hides or displays compilation errors.</td>
</tr>
<tr>
<td>Warnings</td>
<td>Hides or displays compilation warnings.</td>
</tr>
<tr>
<td>Error details</td>
<td>Displays the error message. This button is only active if you have highlighted a listed error message. Click on this button to view a detailed description of the error. Click <strong>Send as mail</strong> to transfer the error message to an email and send.</td>
</tr>
</tbody>
</table>

**Related topics**

• Opening the task window on page 60

• Resolving errors and warnings on page 61

**Opening the task window**

You can open the task window at any time.

*To open the task window*

• Click on **View** | **Tasks** on the menu bar.
Related topics

- Resolving errors and warnings on page 61
- Managing compilation errors and warnings (task window) on page 59

Resolving errors and warnings

If errors occur when compiling your API, the task view can be used to see and resolve them.

TIP: In the global settings, you can define which warnings are displayed as errors during compilation and which errors are ignored. For more information, see Global settings on page 24

To open and edit objects with errors in the task window

1. Click on View | Tasks on the menu bar.
2. Click on either Errors or Warnings in the task window.
3. Perform one of the following tasks:
   - Double-click on the listed error/warning and then resolve the error/warning within the Node editor window itself.
     NOTE: The node editor window opens if the node contains the error. If the error occurs when the resulting code is compiled, you can double-click the error to display the Generated code (read only) view in the definition tree view.
   - OR -
   - Click on the error/warning in the list and then click Correct errors.

Related topics

- Opening the task window on page 60
- Managing compilation errors and warnings (task window) on page 59

Opening the compiling log

The compiling log displays compiling status and progress.

To open the compiling log

- Click on View | Compilation on the menu bar.
Managing versions (compilation branches)

To manage different versions of your compiled API and save them in the database, you use compilation branches. Compilation branches are managed in a separate area of the home page. Here you can select the compilation branch that you want to use, display the existing compilation branches, create compilation branches, edit compilation branches, and delete compilation branches.

Related topics
- Compiling an API on page 57
- Menu bar on page 16

Selecting and using compilation branches

To select and use compilation branches
1. Click on View | Start page on the menu bar.
2. On the home page, expand the Compilation branches view.
3. In the Compilation branches view, click on the compilation branch that you want to use in the Branch ID selection list.
   
   TIP: If you do not want to use a compilation branch, click on Use master status.

Related topics
- Managing versions (compilation branches) on page 62
- Creating compilation branches on page 63
- Editing compilation branches on page 63
- Deleting compilation branches on page 64
Creating compilation branches

To use a compilation branch

1. Click on View | Start page on the menu bar.
2. On the start page, expand the Compilation branches pane.
3. In the Compilation branches pane, click Manage compilation branches.
4. In the Manage compilation branches window, click Add. At the end of the list, a new entry Unknown is added.
5. In the list, click the Unknown entry.
6. In the Identifier field, enter a unique name for the compilation branch.
7. (Optional) In the Description field, enter a description for the compilation branch. This description may explain, for example, the intended purpose of the compilation branch.
8. Click Save.

Related topics

- Managing versions (compilation branches) on page 62
- Selecting and using compilation branches on page 62
- Editing compilation branches on page 63
- Deleting compilation branches on page 64

Editing compilation branches

To edit existing compilation branches

1. Click on View | Start page on the menu bar.
2. On the start page, expand the Compilation branches pane.
3. In the Compilation branches pane, click Manage compilation branches.
4. In the Manage compilation branches dialog, click the compilation branch that you want to edit in the list.
5. In the Identifier field, enter a unique name for the compilation branch.
6. (Optional) In the Description field, enter a description for the compilation branch. This description may explain, for example, the intended purpose of the compilation branch.
7. Click Save.
Related topics

- Managing versions (compilation branches) on page 62
- Selecting and using compilation branches on page 62
- Creating compilation branches on page 63
- Deleting compilation branches on page 64

Deleting compilation branches

To delete compilation branches

1. Click on View | Start page on the menu bar.
2. On the start page, expand the Compilation branches pane.
3. In the Compilation branches pane, click Manage compilation branches.
4. In the Manage compilation branches dialog, click the compilation branch that you want to delete from the list.
5. Click Delete.
6. In the Delete? dialog, confirm the prompt by selecting Yes.

Related topics

- Managing versions (compilation branches) on page 62
- Selecting and using compilation branches on page 62
- Creating compilation branches on page 63
- Editing compilation branches on page 63

Testing an API

You can test the functionality of your API locally on your PC at any time.

To test your API locally

1. Click on View | Start page on the menu bar.
2. Expand the Self-hosted API Server area on the homepage.
3. In the Startup options field, enter the options for starting the API.
   
   TIP: Alternatively, you can also select previous commands from the list.
4. Click Start.
The API is compiled and will then be available locally. The corresponding web address is displayed next to the **Status**. Click the address to open this directly in the browser.

**Related topics**
- Opening the homepage on page 23

## Linking C# projects to the API Designer

To edit an API as a C# project with external programs (for example, Visual Studio), you must perform 3 steps:

1. **Link** your API Designer project with a new or existing C# project.
2. **Edit** the C# project in an external program.
3. **Open** the C# project again in API Designer and save it in the API project.

### To link the API Designer project with a C# project

1. Click on View | Start page on the menu bar.
2. On the start page, expand the **C# development** pane.
3. In the **C# development** pane, perform one of the following actions:
   - To export and link the API Designer project as a new C# project, click ![Export C# project](image). Navigate to the folder where you want to export the content of your API Designer project as a C# project and click **OK**.
   - To link the API Designer project with an existing project, click ![Load C# project](image). The API Designer is linked with the C# project.

### To edit the C# project in an external program

1. Click on View | Start page on the menu bar.
2. On the start page, expand the **C# development** pane.
3. In the **C# development** pane, click **Project file: <path to C# project>**. The project opens in the program that the project file is linked with.

   **TIP:** To open the project folder directly, click ![Folder](image).

4. Edit the project in the external program. For example, you can create, change, or delete API files.
**To save changes back to the API Designer project**

1. Open the API Designer.
2. Click on **View | Start page** on the menu bar.
3. On the start page, expand the **C# development** pane.
4. In the **C# development** pane, click **Synchronize C# project with base**.
   You now see the objects that you have changed outside the API Designer.
5. In the list on the right next to the changed objects, select the actions that you want to perform for the API Designer project.

<table>
<thead>
<tr>
<th>Status message</th>
<th>Description</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>New C# file</td>
<td>A new API file was created in the C# project.</td>
<td>Add API file: Adds the C# file that you created in the external program to your API Designer project as an API file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delete C# file: Deletes the C# class in the C# project.</td>
</tr>
<tr>
<td>New API file</td>
<td>A new API file was created in the API Designer Sharp project.</td>
<td>Add C# file: Adds the API file that you created in the API Designer to your C# project as a C# file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Delete API file: Deletes the API file in the API Designer project.</td>
</tr>
<tr>
<td>API file is newer</td>
<td>An API file was modified in the API Designer project.</td>
<td>Update: Transfers the changes to the API file from the API Designer project into the C# project.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Revert changes: Reverses the changes that you made in your API Designer project in the API file.</td>
</tr>
<tr>
<td>C# file is newer</td>
<td>An API file was modified in the C# project.</td>
<td>Update: Transfers the changes to the API file from the C# project into the API Designer project.</td>
</tr>
</tbody>
</table>
### Status message  | Description  | Actions
---|---|---

|  |  | **Revert changes:** Reverses the changes that you made in your C# project in the API file. |

6. Select the check box next to the objects for which the selected actions should be performed.

**TIP:** To quickly select or deselect all objects, select the **Select all/deselect all** check box above the list.

7. Click ✔️ **Apply selected actions**.
API projects

An API project represents the actual API application itself. You can combine API files that you have created in one API project for a logical application. The API project includes the configuration. The API project includes the authentication on the database.

Related topics

- Creating API projects on page 68
- Editing API projects on page 69
- Deleting API projects on page 70
- Configuring authentication on page 71
- Configuring the compilation on page 74
- Assigning API files to an API project on page 80
- Importing API projects on page 75

Creating API projects

You can create new API projects at any time.

To create an API project

1. Click on View | Navigation on the menu bar.
2. Click API projects in the navigation.
3. Click Add | Add API project.
4. In the definition tree view, click on the topmost node.
5. Click on View | Node editor on the menu bar.
6. In the node editor view, configure the following settings.
<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General settings</strong></td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Enter a unique name for the node.</td>
</tr>
<tr>
<td>Technical abbreviation</td>
<td>Enter a short name for the API project. This name is used later for the assignment of the project.</td>
</tr>
<tr>
<td>Control-ID</td>
<td>Enter a unique ID for the node.</td>
</tr>
<tr>
<td>Model version</td>
<td>Shows the model version being used.</td>
</tr>
<tr>
<td><strong>Workflow control</strong></td>
<td></td>
</tr>
<tr>
<td>Allow extensions to this document</td>
<td>Enable this option to permit the creation of enhancements to this node.</td>
</tr>
<tr>
<td><strong>Advanced settings</strong></td>
<td></td>
</tr>
<tr>
<td>Required database modules</td>
<td>(Optional) Enter the database modules required for this API project.</td>
</tr>
</tbody>
</table>

7. Configure further API project settings (see Configuring authentication on page 71 and Configuring the compilation on page 74).

8. Click Save on the toolbar.

Related topics
- Editing API projects on page 69
- Configuring authentication on page 71
- Configuring the compilation on page 74
- Deleting API projects on page 70
- API projects on page 68
- Navigation on page 21

**Editing API projects**

You can edit any API projects you have created at any time.

NOTE: You cannot edit any API projects that have been predefined by the API Designer.
To edit an API project

1. Click on View | Navigation on the menu bar.
2. Click API projects in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. Click on View | Node editor on the menu bar.
5. Use the definition tree view and the node editor view to configure further API project settings (see Configuring authentication on page 71 and Configuring the compilation on page 74).
6. Click Save on the toolbar.

Related topics

- Creating API projects on page 68
- Configuring authentication on page 71
- Configuring the compilation on page 74
- Deleting API projects on page 70
- API projects on page 68
- Navigation on page 21

Deleting API projects

You can delete any API projects you have created at any time.

NOTE: You cannot delete any API projects that have been predefined by the API Designer.

To delete an API project

1. Click on View | Navigation on the menu bar.
2. Click API projects in the navigation.
3. In the tree structure, click on the API project to be deleted.
4. Click Delete.
5. Confirm the prompt by selecting Yes in the dialog box.
6. Click Reload data in the tree structure.
Configuring authentication

The **Authentication** node in the definition tree view is used to determine how clients will log in to the API.

There are three authentication options that you can configure:

- **Standard**: Only the authentication methods you list here are permitted. For this method you can also enable single sign-on.
- **Allow all manual modules**: All manual authentication modules are permitted, if these are activated for the selected product. For this method you can also enable single sign-on.
- **Fixed credentials**: This settings enables the user to log on to the API using fixed login data. To use this option, the login data must be stored in the `web.config` files on each API server.

**To set the permitted manual authentication options**

1. Click on **View | Navigation** on the menu bar.
2. Click **API projects** in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. In the definition tree view, click on the **Authentication** node.
5. Click on **View | Node editor** on the menu bar.
6. In the node editor view, perform the following actions:
   a. In the **Authentication type** selection list, select **Standard**.
   b. Enter a unique ID for the node into the **Control ID** field.
   c. (Optional) In the **Authentication properties** input field, enter the properties for the authentication. For more information about the authentication modules, see the *One Identity Manager Configuration Guide*.
   d. (Optional) If authorizations for users are controlled via a product, enable the **Product** option and select the required product in the selection list.
7. In the definition tree view, open the **Authentication** node.
8. Right-click the **Manual authentication modules** node.
9. Click **Authentication module** in the context menu.
   The **Authentication module** node is placed as a child to the node.
10. Click the **Authentication module** node.
11. In the node editing view, select the required authentication module from the **Name** selection list. For more information about the authentication modules, see the One Identity Manager Configuration Guide.
12. Enter a unique ID for the node into the **Control ID** field.
13. Repeat steps 8 to 12 until to add each authentication method you wish to use.
14. Click **Save** on the toolbar.

**To permit all manual authentication options**

1. Click on **View | Navigation** on the menu bar.
2. Click **API projects** in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. In the **definition tree view**, click on the **Authentication** node.
5. Click on **View | Node editor** on the menu bar.
6. In the **node editor view**, perform the following actions:
   a. In the **Authentication type** selection list, select **Allow all manual modules**.
   b. Enter a unique ID for the node into the **Control ID** field.
   c. (Optional) In the **Authentication properties** field, enter the authentication properties.
   d. (Optional) Enable the **Product** option and select the product from the list.
7. Click **Save** on the toolbar.

**To allow single sign-on**

1. Click on **View | Navigation** on the menu bar.
2. Click **API projects** in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. In the **definition tree view**, open the **Authentication** node.
5. Right click on the **Authentication modules for single sign-on** node.
   a. **NOTE:** This node is only available if you have manually set the authentication options yourself or if you have enabled all manual authentication options.
6. Click **Authentication module** in the context menu.
   The **Authentication module** node is placed as a child to the node.
7. Click the **Authentication module** node.
8. Click on View | Node editor on the menu bar.
9. In the node editing view, select the required authentication module from the Name selection list. For more information about the authentication modules, see the One Identity Manager Configuration Guide.
10. Click Save on the menu bar.

Log on with stored login data
To approve a login with stored login data, perform the following two steps:
1. Store the login data for the users for whom you want to allow access in the web.config file of each API server.
2. Configure the authentication in the API Designer for the API project.

To store login data in the API Server
1. Connect to your API Server.
2. Open the web.config file in a text editor.
   
   ![NOTE:](image) If the file is encrypted, you must decrypt it before editing.
3. In the <connectionStrings> section, add the following entry:
   
   ```xml
   <add name="sub:<NAME>" connectionString="Module=DialogUser;User=<USER>; (Password)=<PASSWORD>" />
   ```
   
   - `<NAME>` stands for the name/ID of the API project
   - `<USER>` stands for the login name of the user
   - `<PASSWORD>` stands for the user's password
4. Save your changes to the file.
5. (Optional) Encrypt the file.

To configure the login with the saved login data on the API project
1. Start the API Designer.
2. Click on View | Navigation on the menu bar.
3. Click API projects in the navigation.
4. In the tree structure, double-click on the API project to be edited.
5. In the definition tree view, click on the Authentication node.
6. Click on View | Node editor on the menu bar.
7. In the node editor view, perform the following actions:
   a. In the Authentication type selection list, select Fixed credentials.
   b. Enter a unique ID for the node into the Control ID field.
   c. (Optional) In the Authentication properties field, enter the authentication
properties.

d. (Optional) Enable the **Product** option and select the product from the list.

8. Click ✅ Save on the toolbar.

**Related topics**

- Creating API projects on page 68
- Editing API projects on page 69
- API projects on page 68

### Configuring the compilation

Use the **Compilation settings** node in the definition tree view to configure the settings for compiling the API.

If you want to use code from other DLLs in your API code (for example, API files from other projects), you need to create an assembly reference so that the compiler recognizes these DLLs.

**To create an assembly reference**

1. Click on **View | Navigation** on the menu bar.

2. Click 🔄 API projects in the navigation.

3. In the tree structure, double-click on the API project to be edited.

4. In the definition tree view, right-click on the **Compilation settings** node.

5. In the context menu, click **Assembly reference**.

   The **Assembly reference** node is added under this node.

6. In the definition tree view, click **Assembly reference**.

7. Click on **View | Node editor** on the menu bar.

8. In the node editor view, enter the name of the assembly you wish to reference into the **Referenced assembly** field.

9. Click ✅ Save on the toolbar.

**Related topics**

- Compiling an API on page 57
- Creating API projects on page 68
- Editing API projects on page 69
- API projects on page 68
- Configuring authentication on page 71
Assigning API files to an API project

So that the API files you have created can be used practically, they must be assigned to an API project.

To assign API files to an API project

1. Click on View | Navigation on the menu bar.
2. Click API projects in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. In the definition tree view, right-click on the topmost node.
5. In the context menu, click API file reference.
   The new API file reference node is added.
6. In the definition tree view, click API file reference.
7. Click on View | Node editor on the menu bar.
8. In the node editor view, enter a unique ID for the node into the Control ID field.
9. Select the required file in the Name list.
   TIP: To skip directly to the file definition, click Show definition object.
10. Click Save on the toolbar.

Related topics

- Creating API files on page 78
- Creating API projects on page 68
- Editing API projects on page 69
- Deleting API projects on page 70
- API projects on page 68

Importing API projects

You can import API projects into the API Designer. The API Designer independently detects whether the projects are API projects, and imports them as such.

To import an API project

1. Click Edit | Import object on the menu bar.
2. In the file browser, select the required API project and click on Open.
   The API project is imported.
Related topics

- API projects on page 68
- Menu bar on page 16
API files

You can use an API file to create a call for exchanging data with the server (or database) relating to a specific application scenario. An API file can belong to more than one project. In an API file, you can define an API method, for example.

**NOTE:** If you create a new API file to provide a new API method, you must name the API file the same as the class. Ensure you use the correct case.

You can use the API Designer to create the following API files.

**Entity files**

Entity files work with small parts of the object model in order to read data from the database or write data to the database. When you create an entity file, you only need to enter the table and column name and, if required, a filter condition (WHERE clause). Internal processing is handled by the API Server. The data schema for the input and output also has a specific format.

For examples for the definition of entity files, see the SDK under Sdk01_Basics\01-BasicQueryMethod.cs.

**C# files**

C# files are files for which you fully define the processing, input and output data in code. This type therefore offers the greatest flexibility.

For examples for the definition C# files, see the SDK under Sdk01_Basics\03-CustomMethod.cs.

**SQL files**

SQL files are files that provide data from a predefined SQL query via the API. Create the parameters of a query as SQL parameters.

For examples for the definition of SQL files, see the SDK under Sdk01_Basics\02-BasicSqlMethod.cs.
Related topics

- Creating API files on page 78
- Editing API files on page 78
- Deleting API files on page 79
- Assigning API files to an API project on page 80
- Importing API files on page 80

Creating API files

You can create new API files at any time.

| NOTE: If you create a new API file to provide a new API method, you must name the API file the same as the class. Ensure you use the correct case.

To create an API file

1. Click on View | Navigation on the menu bar.
2. In the navigation, click API files.
3. Click Add | Add API file.
4. In the new window, create the file definition.
5. On the toolbar, click Save.

Related topics

- API files on page 77
- Editing API files on page 78
- Deleting API files on page 79
- Navigation on page 21

Editing API files

You can edit the API files you have created at any time.

| NOTE: You cannot edit API files that have been predefined by the API Designer.

One Identity Manager 8.1.1 API Designer User and Development Guide
To edit an API file

1. Click on View | Navigation on the menu bar.
2. Click API files in the navigation.
3. In the tree structure, click on the API file that you want to edit.
4. Create the file definition in the new window.
5. Click Save on the toolbar.

Related topics

- API files on page 77
- Creating API files on page 78
- Deleting API files on page 79
- Navigation on page 21

Deleting API files

You can delete the API files you have created at any time.

NOTE: You cannot delete API files that have been predefined by API Designer.

To delete an API file

1. Click on View | Navigation on the menu bar.
2. Click API files in the navigation.
3. In the tree structure, click on the API file that you want to delete.
4. Click Delete.
5. Confirm the prompt by selecting Yes in the dialog box.
6. Click Reload data in the tree structure.

Related topics

- API files on page 77
- Creating API files on page 78
- Editing API files on page 78
- Navigation on page 21
Assigning API files to an API project

So that the API files you have created can be used practically, they must be assigned to an API project.

To assign API files to an API project

1. Click on View | Navigation on the menu bar.
2. Click API projects in the navigation.
3. In the tree structure, double-click on the API project to be edited.
4. In the definition tree view, right-click on the topmost node.
5. In the context menu, click API file reference.
   The new API file reference node is added.
6. In the definition tree view, click API file reference.
7. Click on View | Node editor on the menu bar.
8. In the node editor view, enter a unique ID for the node into the Control ID field.
9. Select the required file in the Name list.

   TIP: To skip directly to the file definition, click Show definition object.
10. Click Save on the toolbar.

Related topics

- Creating API files on page 78
- Creating API projects on page 68
- Editing API projects on page 69
- Deleting API projects on page 70
- API projects on page 68

Importing API files

You can import API files into the API Designer. The API Designer automatically detects that the files are API files and imports them as such.

To import an API file

1. Click Edit | Import object on the menu bar.
2. In the file browser, select the API file and click Open.
   The API file is imported.
Related topics

- API files on page 77
- Menu bar on page 16
API Server

The API Server provides the API that you create in the API Designer.

To open the API Server web interface

- In a web browser, open the web address (URL) of your API server.
  From here, you can:
  - configure the API Server
  - open the Swagger documentation for your API
  - open the Operations Support Web Portal
  - open all installed HTML5 applications
ImxClient command line program

The ImxClient command line tool can be used to execute all API Designer API functions from a command line without a graphical interface.

Related topics
- Starting the ImxClient command line program on page 83
- ImxClient command overview on page 83

Starting the ImxClient command line program

You can start the ImxClient command line tool at any time using any command line interface.

To start the ImxClient command line program
1. Open a command line interface (e.g. Windows Powershell).
2. In the command line program, go to the installation path for the API Designer.
3. Run the ImxClient.exe application.

Related topics
- ImxClient command line program on page 83

ImxClient command overview

The following chapters contain a list of all ImxClient commands that you can execute.
Related topics
- help on page 84
- compile-app on page 84
- repl on page 85
- branch on page 85
- connect on page 86
- compile-api on page 86
- fetch-files on page 88
- push-files on page 88
- get-apistate on page 89
- get-filestate on page 89
- setup-web on page 90
- run-apiserver on page 91
- check-translations on page 92

help
Displays a list of available commands.

Parameter
To view help for a specific command, add the command as a parameter.
Example: help fetch-files

Related topics
- ImxClient command line program on page 83
- ImxClient command overview on page 83

compile-app
Executes HTML5 package compilation.
This command performs the following steps:
1. Execution of the npm install command in the application folder
2. Execution of the npm run build command in the package folder
3. Create the output in subdirectory dist
   The output is stored as a ZIP file in the database.
Parameters

- /conn <name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /path <path to main folder>: Enter the path to the main folder. This folder contains the application to be compiled. This folder normally contains the package.json file of the application.
- -N: (Optional) Prevents saving to the database.
- -D: Executes the debug compilation.
- /branch <ID of the compilation branch>: (Optional) Saves the compiled result under the specified compilation branch. You must also specify the parameter -D.

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83

repl

Starts the ImxClient command line tool in REPL mode.
In this mode, the following actions are performed in an infinite loop:

- Read commands from stdin
- Forward commands to the relevant plug-in
- Output the results of processing to stdout

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83

branch

Manages compilation branches in the database

Parameter

- Run without parameters: If the command is called without specifying any parameters, this command outputs the total number of compilation branches and
their IDs from the database.

- `/id <ID of the compilation branch>`: (Optional) queries the field name and corresponding values of a compilation branch.
- `-d /id <ID of the compilation branch>`: (Optional) deletes a compilation branch.
- `branch -c /id <ID of the compilation branch>`: (Optional) creates a new compilation branch.
  To also specify a description of the compilation branch, use `/description <description>`.

**Related topics**
- ImxClient command line program on page 83
- ImxClient command overview on page 83

**connect**

Establishes a database connection.

If a connection to a database has already been established, this is closed and a new connection is then established.

**Parameter**

- `/conn <name of the database connection>`: Specify the database to which you want to connect.
- `/dialog <name of the dialog authentication>`: Enter the dialog authentication.
- `/factory <target system>`: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.

**Related topics**
- ImxClient command line program on page 83
- ImxClient command overview on page 83

**compile-api**

Compiles the API and saves the result to the database.
Parameter

- `/conn <name of the database connection>`: Specify the database to which you want to connect.

- `/dialog <name of the dialog authentication>`: Enter the dialog authentication.

- `/factory <name of target system>`: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server. Example: QBM.AppServer.Client

- `/solution <path to solution project>`: (Optional) Enter the path to the solution project file that you want to use. If you do not enter these parameters, a database project is used.

- `/mode <compile mode>`: (optional) Enter the mode for the compilation:
  - `normal`: complete compilation (default mode)
  - `nostore`: assemblies are not saved in the database
  - `nocompile`: creates only C# code. The compilation is not executed.
  - `nocodegen`: executes only the compilation. No C# code is generated.

- `-E`: (optional) activates extended checks.

- `-D`: (optional) the compilation is executed as debug compilation.

- `/csharpout <path to target folder for C# files>`: (optional) enter the path to the folder in which you want to save the C# files.

- `/copyapi <path to folder>`: (optional) enter the path to the folder to which you want to copy the imx-api.tgz file.

- `/apidll <path to an API-DLL>`: (optional) enter the path for an API-DLL file that you want to use. The `/solution` and `/branch` parameters are ignored if you use this parameter.

- `/branch <ID of the compilation branch>`: (optional) enter the compilation branch under which you want to save the project. You must also specify the parameter `-D`.

- `/nowarn <error1, error2, ...>`: (optional) specify which errors are to be ignored during compilation. Enter the codes for the warnings, separated by commas.

- `/warnaserror <error1, error2, ...>`: (optional) specify which warnings are displayed as errors during compilation. Enter the codes for the warnings, separated by commas.

Related topics

- `ImxClient command line program` on page 83
- `ImxClient command overview` on page 83
fetch-files

Loads all files of the "HTML Development" machine role from the database and saves them in a local file.

Parameters

- /conn <Name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /factory <target system>: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- /path <path to target folder>: (optional) enter the path to the folder in which you want to save the files.
- /targets <target1;target2;...>: (optional) specify which machine roles you want to use. If you do not use this parameter, the "HTML Development" machine role is used.

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83

push-files

Saves files that you have changed locally back to the database.

Parameters

- /conn <Name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /targets <target1;target2;...>: Specify which machine roles you want to use. If you do not use this parameter, the "HTML Development" machine role is used.
- /factory <target system>: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- /path <path to folder>: (optional) enter the path to the folder containing the files that you have changed and you now want to save to the database.
- /tag <uid>: (optional) specifies the UID of a change tag.
- /add <file1;file2;...>: (optional) specify the paths to all new files to be added to the database. Use relative paths.
- /del <file1;file2;...>: (optional) enter the paths to all files that you want to delete from the database. Use relative paths.
- -c: (optional) prevents the saving of changed files and saves only new files, and deletes files from the database.

**Related topics**
- ImxClient command line program on page 83
- ImxClient command overview on page 83

**get-apistate**

Queries the compilation status of the API in the database.

**Parameters**
- /conn <Name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /factory <target system>: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- /branch <ID of the compilation branch>: (optional) queries the date of the compilation status of the API under which this compilation branch was saved.
- /htmlapp <name of the HTML package>: (optional) supplies data for the specified HTML package.
- -D: (optional) supplies data for debug assemblies.
- -R: (optional) supplies data for release assemblies.

**Related topics**
- ImxClient command line program on page 83
- ImxClient command overview on page 83

**get-filestate**

Compares the local file structure with the file structure in the database.

Using the QBM | ImxClient | get-filestate | NewFilesExcludePatterns configuration parameter, you can define which files are excluded from the synchronization. This prevents excessive load during synchronization. The node_modules and imx-modules folders are excluded from the synchronization by default.
You can adjust this parameter in the Designer. Use the following formats when defining the rules:

https://docs.microsoft.com/en-us/dotnet/api/microsoft.extensions.filesystemglobbing.matcher

Use the | character to separate multiple entries.

**NOTE:** This configuration parameter is generally only used to exclude new files from the synchronization. Files that already exist in the database are not taken into account.

**Parameters**

- `/conn <Name of the database connection>`: Specify the database to which you want to connect.
- `/dialog <name of the dialog authentication>`: Enter the dialog authentication.
- `/factory <target system>`: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- `/targets <target1;target2;...>`: (optional) specify which machine roles you want to use. If you do not use this parameter, the "HTML Development" machine role is used.
- `/path <path to folder>`: (optional) enter the path to the folder containing the files that you want to synchronize.

**Related topics**

- [ImxClient command line program](#) on page 83
- [ImxClient command overview](#) on page 83

**setup-web**

Installs necessary files for the development of TypeScript clients.

**Parameter**

- `/root`: Enter the path to the main folder.
- `/conn <name of the database connection>`: Specify the database to which you want to connect.
- `/dialog <name of the dialog authentication>`: Enter the dialog authentication.
- `/factory <target system>`: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- `/path <path to the API project>`: (Optional) Specify the path to the folder for the API project.
/branch <ID of the compilation branch>: (Optional) Enter the ID of a compilation branch for which you want to install files.

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83

run-apiserver

Starts or stops a "self-hosted" API Server.
This command requires a database connection.

Parameters

- /conn <Name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /factory <target system>: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.
- -S: (Optional) Stops the API Server.
- /baseaddress <URL with port>: (optional) specify the base URL and the port of the HTML application.
- /baseurl <base URL>: (optional) enter the base URL of the HTML application.
- /branch <ID of the compilation branch>: (optional) specify the ID of a compilation branch for whose API you want to start the API Server.
- -D: (Optional) loads the debugging assemblies.
- -C: (Optional) Compiles the API using source data from the database.
- -T: (Optional) Retrieves the status of the currently running API Server.
- -B: (Optional) Locks the console.
- /compile {path to solution file}: Compiles the API using source data from a (local) solution project.
- /excludedMiddlewares {Middleware-Name1,Middleware-Name2,...}: (Optional) Specify which middleware services you do not want to offer. Enter multiple values separated by a comma.

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83
check-translations

Search for captions (multilingual captions) with missing translations in a particular folder and its subfolders.

Parameters

- /conn <Name of the database connection>: Specify the database to which you want to connect.
- /dialog <name of the dialog authentication>: Enter the dialog authentication.
- /path <Path to folder>: Enter the path to the folder that you want to check.
- /factory <target system>: (Optional) Enter the target system for the connection. Enter this parameter if you want to establish a connection to the application server.

Related topics

- ImxClient command line program on page 83
- ImxClient command overview on page 83
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 or other inquiries, visit https://www.oneidentity.com/company/contact-us.aspx or call +1-800-306-9329.

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
A

add
  API files 78
  API projects 68
change label 38
compiler branch 63
database queries 49
multilingual captions 45
API
  test 64
API Designer 7
  starting 11
API development
  basics 8
API file
  delete 79
API files 77
  add 78
  assign to a project 75, 80
  change 78
  create 78
  delete 79
  edit 78
  import 80
API projects 68
  add 68
  change 69
  create 68
  delete 70
  edit 69
  import 75
remove 70
API Server 82
assembly reference 74
authentication
  configure 71

B
basics 8
benefits 7
bookmark
  open 55
  show 55
bookmarks 54
  delete 56
  remove 56
  set 56
branches 62

C
C# methods 77
C# project 65
captions 44-45, 47
  change 46
  create 45
  delete 47
  edit 46
  find 45
  new 45
  remove 47
  search 45
change
  API methods 78
  API projects 69
captions 46
change label 39
compiler branch 63
database queries 50
global settings 26
multilingual captions 46
node 36
settings 26
change label 37
  add 38
  change 39
create 38
delete 40
edit 39
new 38
remove 40
use 37
CLI 83
close
tab 52
code
  view 34
colors 21
command line 83
command list 53
  display 54
open 54
commands 53, 83
compilation branches 62
compile 57, 59
compiler branch
  add 63
apply 62
change 63
create 63
delete 64
edit 63
new 63
remove 64
select 62
compiling 57
  configure 74
display 61
execute 59
open 61
starting 59
test 58
configurations
  authentication 71
compiling 74
context menu 31
create
  API files 78
  API projects 68
captions 45
change label 38
compiler branch 63
database queries 49
multilingual captions 45
D
database projects 12
database queries
  add 49
change 50
create 49
edit 50
new 49
remove 52
test 51
database query 48
delete 52
manage 48
deactivate compilation 29
debug 57
default layout 53
restore 53
definition tree 31
   open 31
definition tree view 27, 30-31
       open
definition tree view 31
delete
   API files 79
   API methods 79
   API projects 70
   bookmarks 56
captions 47
change label 40
compiler branch 64
database queries 52
multilingual captions 47
display
   compiling 61
definition tree 31
definition tree view 31
definitions 31
   object properties 28
task window 60
displays
   command list 54

E
   edit
       API files 78
       API projects 69
captions 46
change label 39
compiler branch 63
database queries 50
multilingual captions 46
   node 36
edit nodes 35
ten entity methods 77
ten examples 9
execute
   compiling 59
export 12, 14
export solution 14
extensions 33

F
   file-based 14
   file path 29
find 43
   captions 45
   multilingual captions 45
find and replace 40, 43
fix error 61
fix warnings 61

G
   generated code 34
global settings 24
   change 26
open 26
graphical user interface 15
GUI 15

ImxClient command line tool
start 83
interface 15

H
help 9
home page 23
homepage
open 23

L
language 44-47
languages 44-47
layout 53
restore 53
save 53

M
managing database queries 48
menu 16
menu bar 16
multilingual captions 44-47
add 45
change 46
create 45
delete 47
edit 46
find 45
new 45
remove 47
search 45

N	namespace 74
navigate 21
navigation 21
open 22
show 22
new
API files 78
API projects 68
captions 45
change label 38
compiler branch 63
database queries 49
multilingual captions 45
new API files 78
new API project 68
node
  change 36
  edit 36
node editor 35
  open 36
node editor view 35

O
object properties 27
  display 28
  open 28
open 31
  bookmark 55
  command list 54
  compiling 61
  generated code 34
  global settings 26
  homepage 23
  navigation 22
  node editor 36
  object properties 28
  settings 26
  solution 30
  tab 52
  task window 60
option 24
  own layout 53

P
Postman 82
  project types 12
  properties of an object 27

Q
quick start 10

R
redo 24, 53
release 57
reload object 29
remove
  bookmarks 56
  captions 47
  change label 40
  compiler branch 64
  database queries 52
  multilingual captions 47
repeat 24
replace 40, 43
resolution 24
right click 31
run
  compiling 59

S
save
  tab 52
save changes 14
save locally 12
save to disk 12
save type
  specify 12
SDK 9
search 40
  captions 45
  multilingual captions 45
settings 24
  change 26
  open 26
show
  bookmark 55
  navigation 22
single sign-on 71
Software Development Kit 9
solution 29
  open 30
solution projects 12, 14
specify
  save type 12
SQL files 77
SQL statement
  test 51
SSO 71
start 11
  command line tool 83
status bar 21
  meaning of the colors 21
Swagger 82

T
  save type 12
tab 52
  close 52
  manage 52
  open 52
  save 52
task window 59
  display 60
  open 60
tasks 59
test
  API 64
  compiling 58
  database queries 51
  SQL statement 51
text 46
  toolbar 19
type member 74

U
  UI 15
undo 24, 53
use
  change label 37
  user interface 15

V
  version management 62
view
  code 34
  generated code 34
Visual Studio 65