

iWay

iWay Integration Tools
Getting Started Guide
Version 8.0 and Higher

Active Technologies, EDA, EDA/SQL, FIDEL, FOCUS, Information Builders, the Information Builders logo, iWay, iWay Software, Parlay, PC/FOCUS, RStat, Table Talk, Web390, WebFOCUS, WebFOCUS Active Technologies, and WebFOCUS Magnify are registered trademarks, and DataMigrator and Hyperstage are trademarks of Information Builders, Inc.

Adobe, the Adobe logo, Acrobat, Adobe Reader, Flash, Adobe Flash Builder, Flex, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2020, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Contents

Preface	5
Documentation Conventions	5
Related Publications	6
Customer Support	6
Help Us to Serve You Better	6
User Feedback	9
Information Builders Consulting and Training	9
1. Introducing iWay Integration Tools Version 8	11
iWay Integration Tools Overview	11
Installing and Starting iWay Integration Tools.....	11
Setting the Workspace.....	12
Workbench Layout.....	13
Managing Perspectives.....	15
Setting the iWay Home Directory.....	17
Working With Application Projects	20
Understanding the Application Project Folder Structure.....	22
Understanding the Application Project Bundle.....	23
A. Use Case: Hosting RESTful APIs	27
Defining the API Structure Using a RAML File	27
Defining the Business Logic to be Executed for each API Endpoint and its Corresponding Action	33
Configuring the API Settings for the HTTP Communication Layer	35
Deploying the Application	37
Starting and Testing the Application Using a Browser	39

Preface

This document is intended for system integrators and application designers. It provides an introduction to iWay Integration Tools (iIT) and explains how to configure and use iIT in software integration projects.

How This Manual Is Organized

This manual includes the following chapters:

Chapter/Appendix	Contents
1	Introducing iWay Integration Tools Version 8
A	Use Case: Hosting RESTful APIs

Documentation Conventions

The following table describes the documentation conventions that are used in this manual.

Convention	Description
THIS TYPEFACE or this typeface	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable), a cross-reference, or an important term. It may also indicate a button, menu item, or dialog box option that you can click or select.
<u>underscore</u>	Indicates a default setting.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices. Type one of them, not the braces.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis (...).

Convention	Description
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

Visit our Technical Documentation Library at <http://documentation.informationbuilders.com>. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have any questions about this product?

Join the Focal Point community. Focal Point is our online developer center and more than a message board. It is an interactive network of more than 3,000 developers from almost every profession and industry, collaborating on solutions and sharing tips and techniques. Access Focal Point at <http://forums.informationbuilders.com/eve/forums>.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our website, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of <http://www.informationbuilders.com> also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

Call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 a.m. and 8:00 p.m. EST to address all your questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code number (xxxx.xx) when you call.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively, be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following tables list the environment information our consultants require.

Platform	
Operating System	
OS Version	
JVM Vendor	
JVM Version	

The following table lists the deployment information our consultants require.

Adapter Deployment	For example, JCA, Business Services Provider, iWay Service Manager
Container	For example, WebSphere
Version	
Enterprise Information System (EIS) - if any	
EIS Release Level	
EIS Service Pack	
EIS Platform	

The following table lists iWay-related information needed by our consultants.

iWay Adapter	
iWay Release Level	
iWay Patch	

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Did the problem arise through a service or event?	
Provide usage scenarios or summarize the application that produces the problem.	
When did the problem start?	
Can you reproduce this problem consistently?	
Describe the problem.	
Describe the steps to reproduce the problem.	
Specify the error message(s).	
Any change in the application environment: software configuration, EIS/database configuration, application, and so forth?	
Under what circumstance does the problem <i>not</i> occur?	

The following is a list of error/problem files that might be applicable.

- Input documents (XML instance, XML schema, non-XML documents)
- Transformation files
- Error screen shots
- Error output files
- Trace files
- Service Manager package to reproduce problem

- Custom functions and agents in use
- Diagnostic Zip
- Transaction log

For information on tracing, see the *iWay Service Manager User's Guide*.

User Feedback

In an effort to produce effective documentation, the Technical Content Management staff welcomes your opinions regarding this document. Please use the Reader Comments form at the end of this document to communicate your feedback to us or to suggest changes that will support improvements to our documentation. You can also contact us through our website, <http://documentation.informationbuilders.com/connections.asp>.

Thank you, in advance, for your comments.

Information Builders Consulting and Training

Interested in training? Information Builders Education Department offers a wide variety of training courses for this and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our website (<http://education.informationbuilders.com>) or call (800) 969-INFO to speak to an Education Representative.

Introducing iWay Integration Tools Version 8

This section provides an introduction to iWay Integration Tools (iIT) version 8 and a quick walkthrough of the environment.

In this chapter:

- [iWay Integration Tools Overview](#)
 - [Working With Application Projects](#)
-

iWay Integration Tools Overview

iWay Integration Tools (iIT) is an Eclipse-based design-time environment that allows you to develop and manage scalable integration applications within the iWay Service Manager (iSM) framework requiring little to no coding.

Installing and Starting iWay Integration Tools

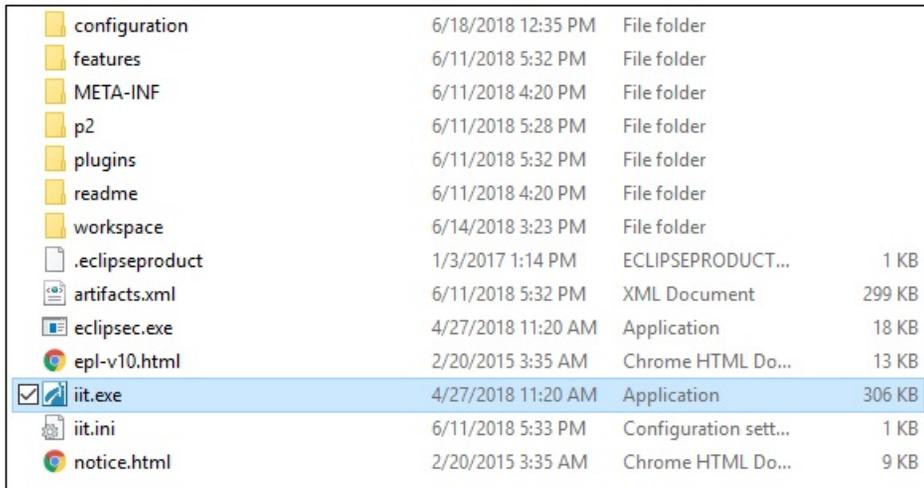
iWay Integration Tools (iIT) is packaged as a .zip file, for example:

`iIT-8.0.1-win32.win32.x86_64.zip`

To install iIT, simply extract this .zip file to a folder on your file system, for example:

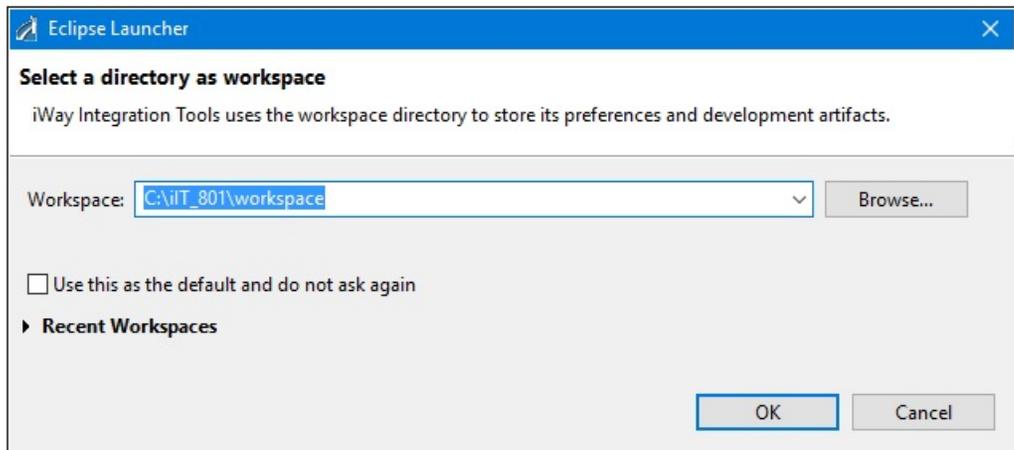
`C:\iIT_801`

To start iIT, double-click the *iit.exe* file under the folder where you extracted the .zip file, as shown in the following image.



Setting the Workspace

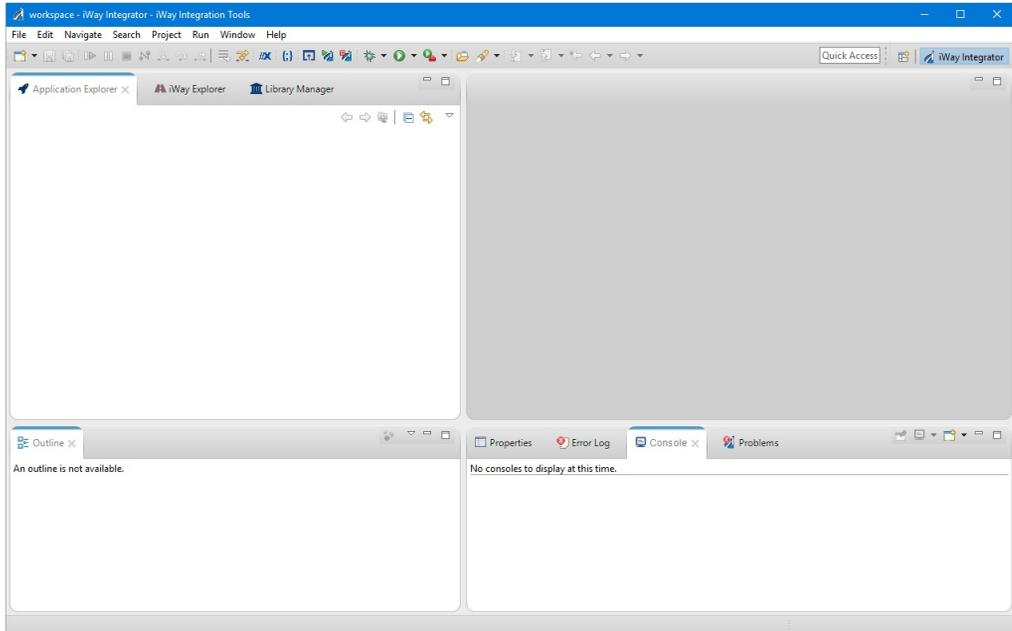
After starting iIT, the Eclipse Launcher dialog box opens, as shown in the following image, which prompts you to set a workspace location.



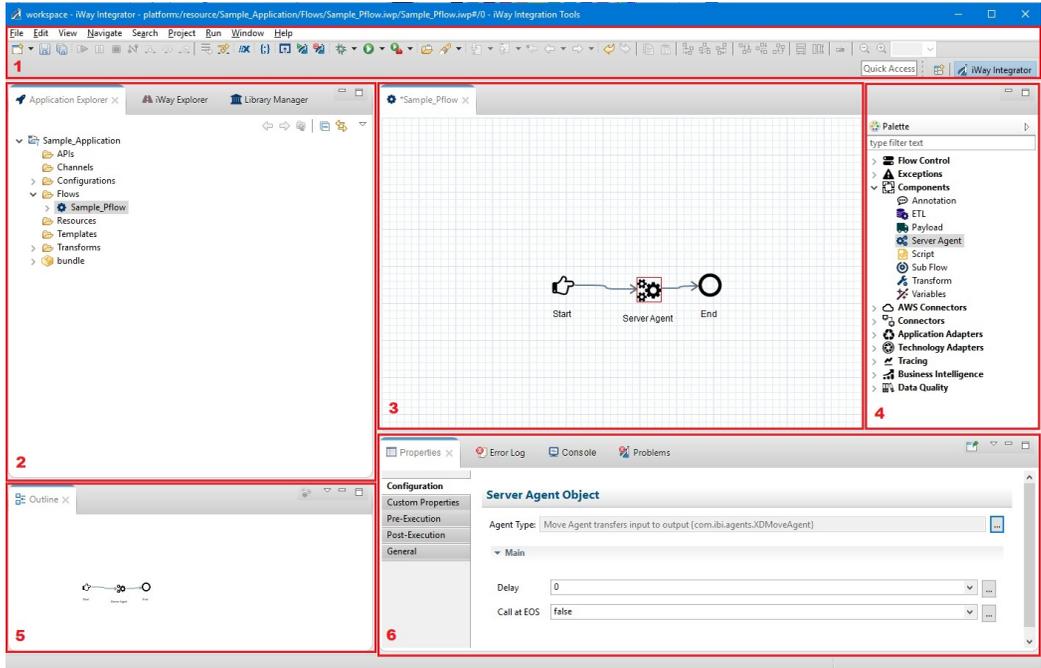
The workspace is a container for all of the application projects that you plan to create.

Workbench Layout

After you have defined a directory for your iIT workspace, the workbench opens, as shown in the following image.



When you start iIT for the first time, the workbench is empty. To better understand the layout of the workbench, the following image shows a sample application project that has been created, which includes a simple process flow.



The workbench is organized as follows:

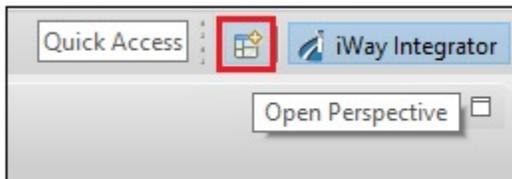
1. **Menu bar and toolbar.** Provides a variety of options for managing your projects, starting and stopping iWay Service Manager (iSM), testing and running your projects, and selecting additional tools.
2. **Navigator.** The navigator pane appears on the left and contains the following tabs:
 - Application Explorer.** Used to build your individual application projects. The Application Explorer uses a folder structure to organize all of your components and resources.
 - iWay Explorer.** Allows you to interact with deployed instances of iSM.
 - Library Manager.** Provides a centralized location of reusable components for channels and transforms.
3. **Workspace.** The workspace pane appears to the right of the navigator pane. This is the area in which you design and develop various iWay components (for example, channels, APIs, process flows, and transforms).

4. **Palette.** Appears to the right of the workspace area when a new process flow is created. The palette contains all of the objects and components that are required to construct a process flow.
5. **Outline.** The outline pane appears below the navigator pane. It displays a preview of the process flow as you create it.
6. **Properties.** The Properties window pane appears at the bottom of the workspace. This pane displays the attributes of the selected object in a process. You can change any attribute of the selected object from the Properties window.

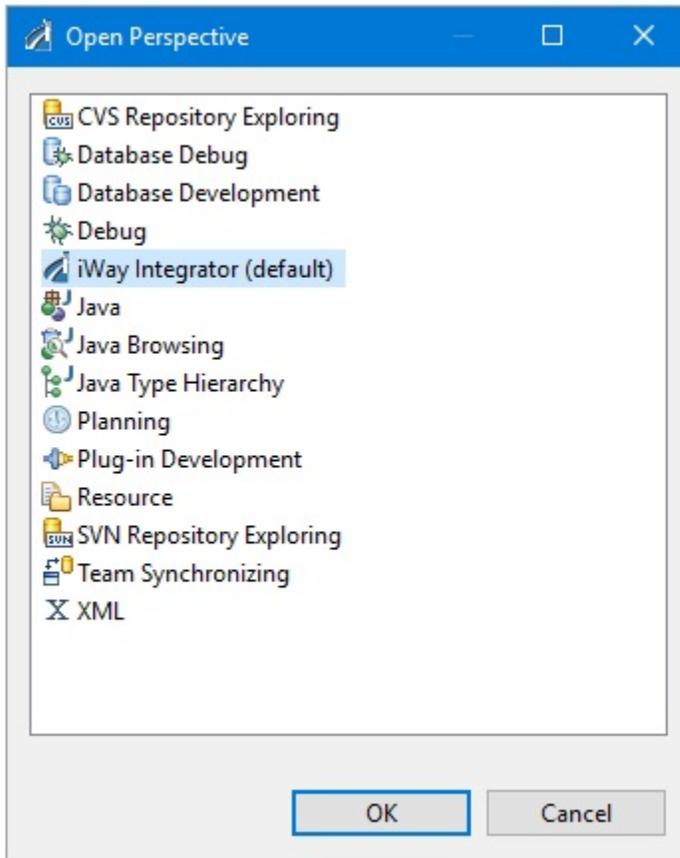
Managing Perspectives

The workbench is the name for the development environment that contains one or more perspectives. A perspective is a simple collection and layout of views and editors. Editors are usually found in the center of a perspective and are used for editing content. Views surround the editor and allow you to browse and modify the properties of resources.

To access available perspectives, click *Open Perspective* in the upper-right corner, as shown in the following image.

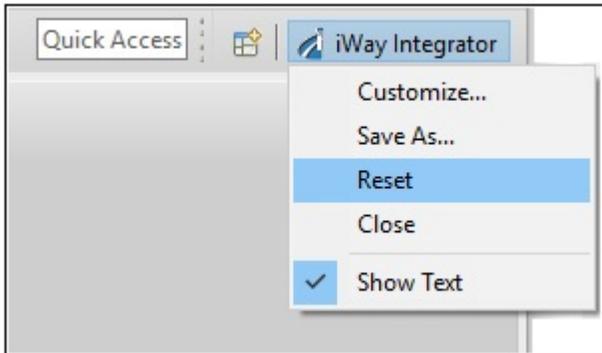


The Open Perspective dialog opens, as shown in the following image.

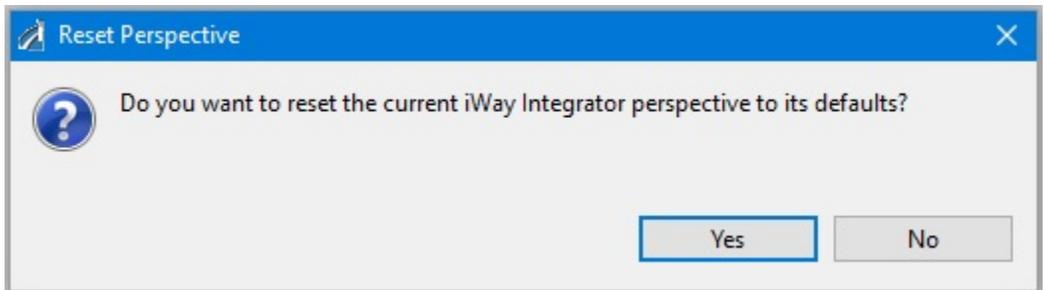


Notice that the iWay Integrator perspective is set by default. Click OK to exit from this dialog.

If your iWay Integrator perspective should change for any reason (for example, you accidentally close one of the views), you can easily restore the default settings by right-clicking *iWay Integrator* in the upper-right and selecting *Reset* from the context menu, as shown in the following image.



The Reset Perspective prompt is displayed, as shown in the following image.

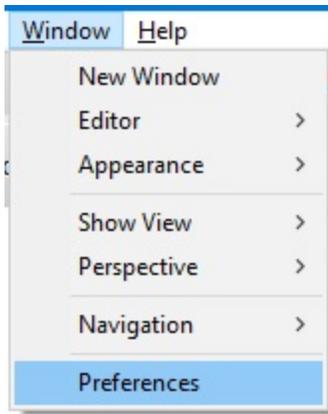


Click Yes to confirm.

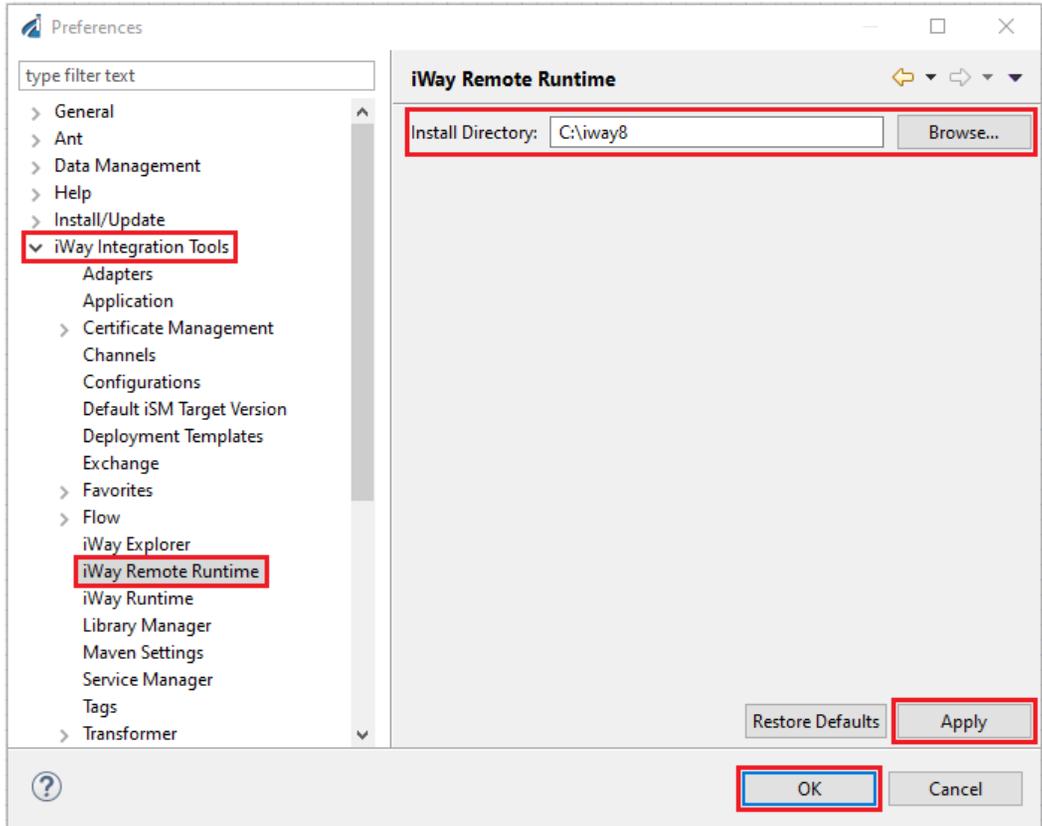
Setting the iWay Home Directory

Once the workbench is open, it is recommended that you first set your iWay home directory.

To do this, click *Window* from the menu bar and select *Preferences*, as shown in the following image.

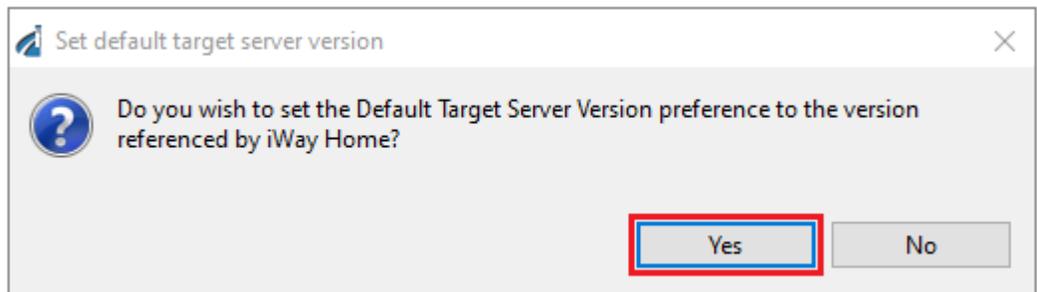


The Preferences dialog opens, as shown in the following image.



Expand the *iWay Integration Tools* preference in the left pane and then click *iWay Remote Runtime*. Click *Browse* to navigate to the location of your iWay 8 installation on your system. Click *Apply* and then click *OK*.

The Set default target server version dialog opens, as shown in the following image.

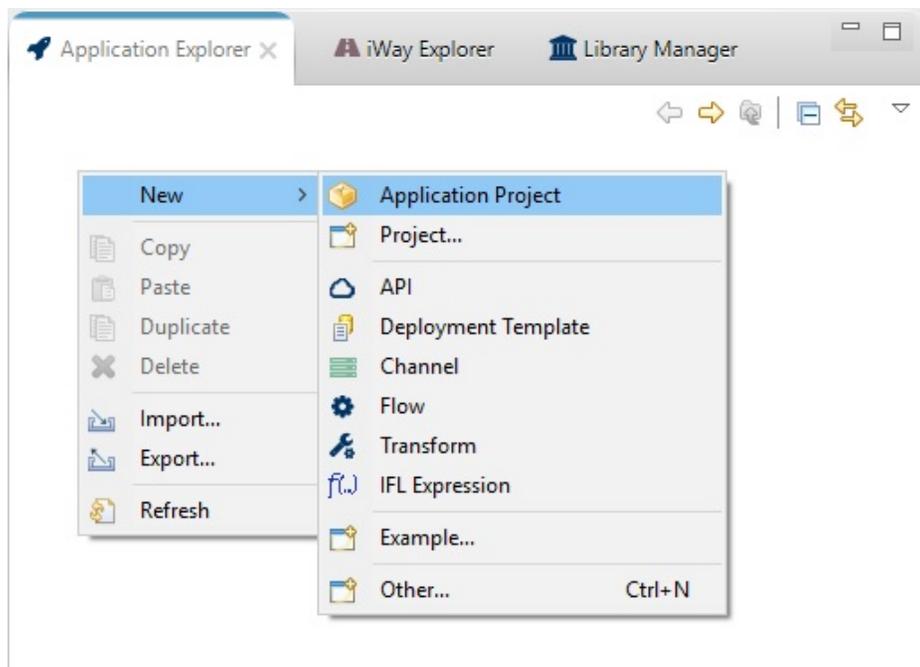


Click Yes, if you want the *Default Target Server Version* that is used by iIT to match the version of your iWay 8 installation (iWay Service Manager) that is installed in the iWay home directory you have referenced.

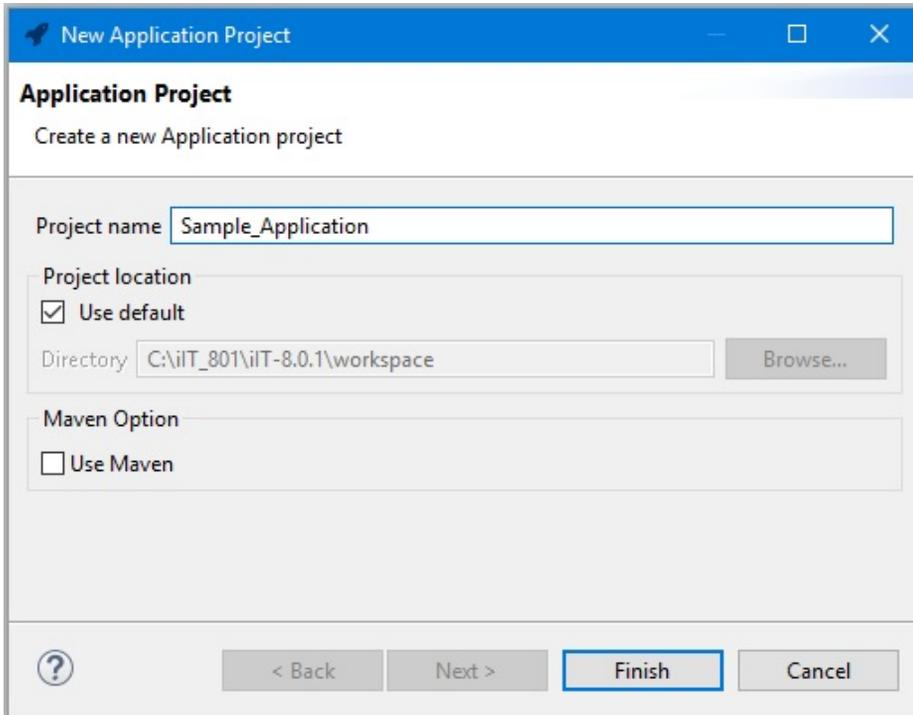
Working With Application Projects

The Application Explorer is displayed as a tab in iWay Integration Tools (iIT) and is the primary area where you create and manage all of your application projects. An application project contains all of the required components (for example, channels, process flows, and transforms) and artifacts that are used by an integration application.

To get started, simply right-click anywhere within the Application Explorer and select *Application Project* from the context menu, as shown in the following image.

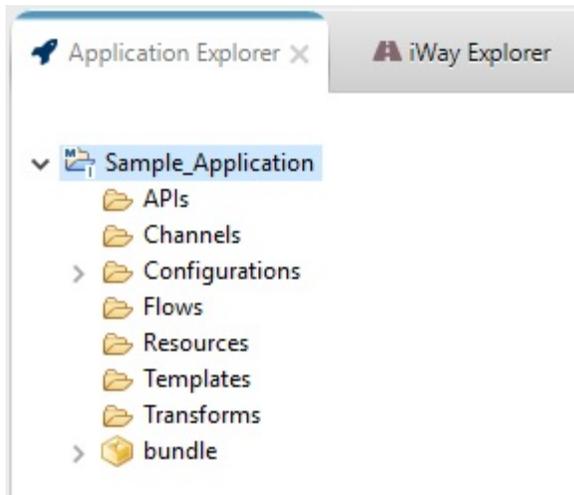


The New Application Project dialog opens, as shown in the following image.



Provide a name for your Application Project (for example, `Sample_Application`) and specify a project location. The default setting creates your application projects under the workspace location that you specified. If your application project is to be built and managed using Maven, then select the *Use Maven* checkbox. For the purposes of this example, leave this option unchecked. Click *Finish* to continue.

Your new application project is now listed in the Application Explorer tab, as shown in the following image.



Understanding the Application Project Folder Structure

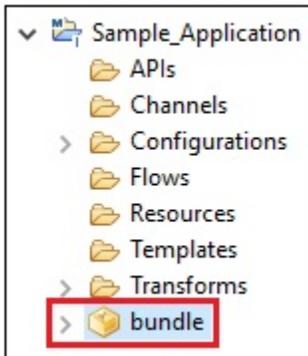
Each application project is organized into several folders, which are described in this section.

- ❑ **APIs.** The APIs folder is the default folder for storing REST-based APIs. These APIs are created using the API Editor.
- ❑ **Channels.** The Channels folder is the default folder for storing event based channels. Channels are created using the Channel Builder.
- ❑ **Configurations.** The Configurations folder is the default folder for managing configurations which store iWay Generics bindings. Users can create more than one configurations using the iWay Configurations Editor. An application project can have only one active configuration file. Users can select a configuration within the properties of the application project.
- ❑ **Flows.** The Flows folder is the default folder for storing Flows. Flows are orchestrated using the iWay Flow Editor, a drag drop interface for creating simple to complex point to point integrating solutions.
- ❑ **Resources.** The Resources folder is the default folder for storing integration dependent artifacts. By default any file based artifact will be included within the application bundle. These artifacts will be found within the resources directory of the deployed run-time.

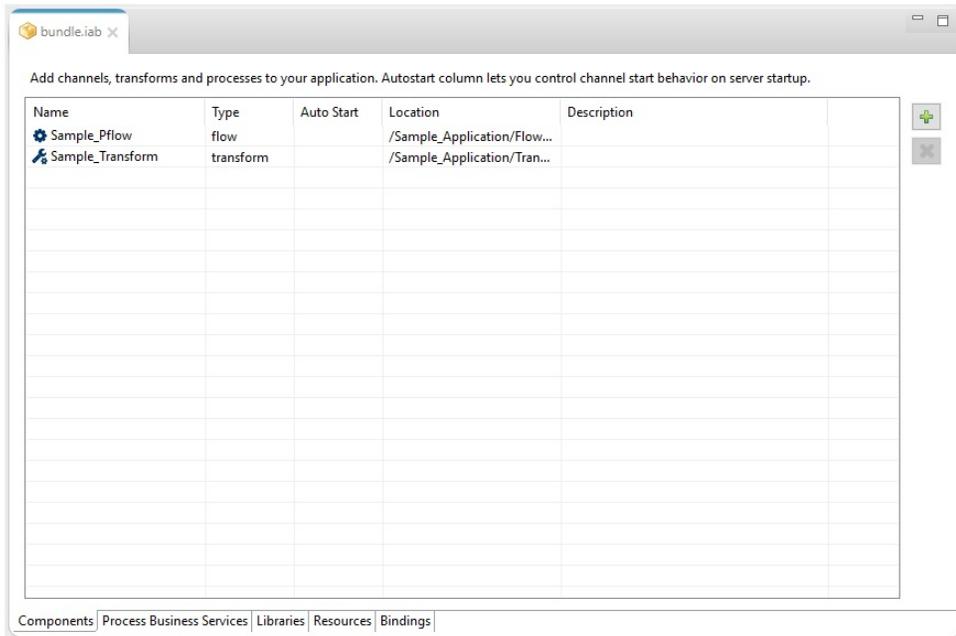
- ❑ **Templates.** The Templates folder is the default folder for storing iWay deployment templates. Deployment templates allow users to create a component which inject configuration into runtime creation. Often these templates contain variables whose values are specific for either development, test and or production environments. Templates can be created using the iWay Template Builder.
- ❑ **Transforms.** The Transforms folder is the default folder for storing iWay Transforms. Transformation is one of the most important foundations of an integration platform. The iWay Transformation Builder allows users to create simple to complex transformations using a graphical drag-drop interface. A wide variety of formats are supported such as JSON, XML, CSV, and many more.

Understanding the Application Project Bundle

The bundle is the core of an application project and is represented by a package icon below the Transforms folder, as shown in the following image.



The name of the bundle package can be renamed, but its meaning remains the same. The contents of the bundle is self-managed by the application project. Its contents can be viewed and changed through the application editor. To open the application editor, double-click on the bundle item within the application project. The editor displays, as shown in the following image.



As you work with your application project, the contents of the bundle will change accordingly. Most likely, items listed in the Components section will change when project artifacts, such as APIs, process flows, and channels are either added or removed from the application project. By default, any artifact added to the project is added to the bundle. Thus, the bundle represents all artifacts making up an application. The application editor is organized by tabs as follows:

- Components.** This tab allows you to see what is contained within the bundle. As mentioned above, any artifact added to the project automatically gets added to the bundle. However, using the components tab users can control what goes in and out of the bundle.
- Process Business Services.** When creating XML-based SOAP RESTful services using the iWay Explorer (Adapters), those services need to be injected into the runtime. This tab manages which SOAP-based REST services will be available to call during run-time.
- Libraries.** Often applications will be required to add one or more run-time dependencies on the classpath. Any file listed in this tab will be available on the classpath during run-time execution.

- ❑ **Resources.** The bundle contains a resources section where applications can store any file-based artifact. Often applications require sample files, configuration, or even images. By default the bundle includes the "resources" folder of the application project. Any file or directory added to this folder will be included in the bundle. Users have full control of what is included in the bundle by adding or removing artifacts.
- ❑ **Bindings.** Application Bindings allow developers the ability to inject variables into the run-time. It is assumed those variables will be used either by a flow or channel at run-time.

Use Case: Hosting RESTful APIs

This use case describes how to host RESTful APIs for consumption by external callers using iWay Integration Tools (iIT). This enables an application to expose business processes based on a given endpoint definition. The use case includes a simple RAML file (*mailbox.raml*) as an example, which is packaged in the *mailbox_raml_sample.zip* file.

In a real-world scenario you will be responsible for creating your own RAML file to describe your specific business endpoints, which you would like to expose. This can be done using a variety of tools. You also have the option of designing and hosting your RESTful APIs without a pre-existing RAML file. However, this is not recommended and not covered in this use case. This use case also assumes general understanding of RESTful services and concepts related to RESTful APIs.

Key Terms:

- ❑ **RESTful API Modeling Language (RAML).** A YAML-based language for describing RESTful APIs. It provides all of the information required to describe RESTful or practically RESTful APIs.
- ❑ **RESTful API.** A RESTful API is an application programming interface (API) that uses HTTP requests to GET, PUT, POST, and DELETE data.

This use case includes the following topics:

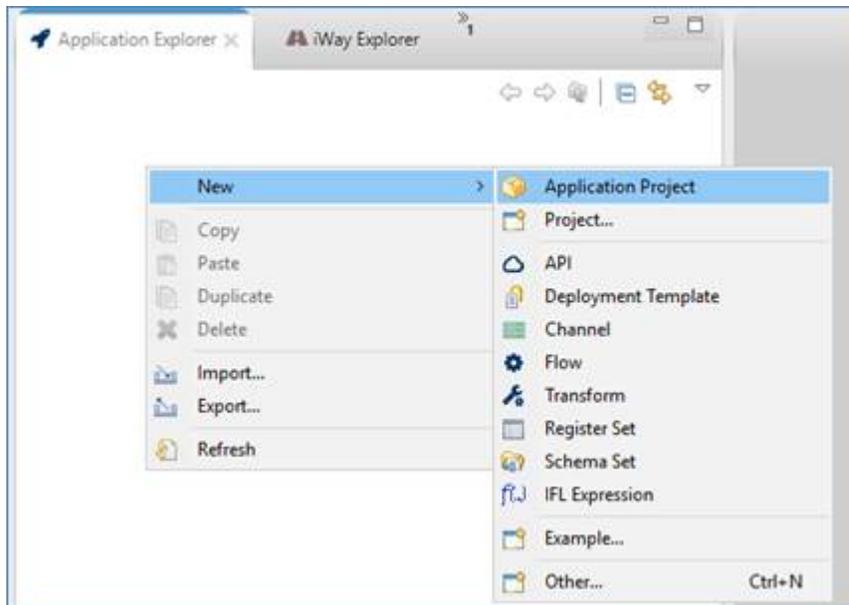
In this appendix:

- ❑ [Defining the API Structure Using a RAML File](#)
 - ❑ [Defining the Business Logic to be Executed for each API Endpoint and its Corresponding Action](#)
 - ❑ [Configuring the API Settings for the HTTP Communication Layer](#)
 - ❑ [Deploying the Application](#)
 - ❑ [Starting and Testing the Application Using a Browser](#)
-

Defining the API Structure Using a RAML File

1. Open iWay Integration Tools (iIT) and select the default workspace.

2. Right-click anywhere within the Application Explorer tab, select New from the context menu, and then click Application Project, as shown in the following image.



The New Application Project dialog opens, as shown in the following image.

New Application Project

Application Project
Create a new Application project

Project name

Project location

Use default

Directory

Maven Option

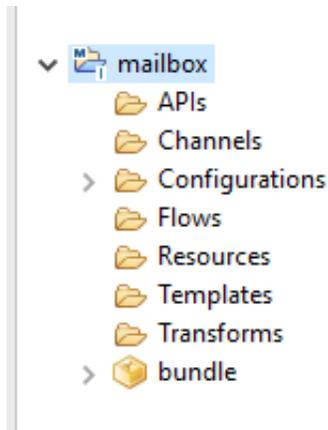
Use Maven

Target Server Version

[Install additional Target Server Version](#)

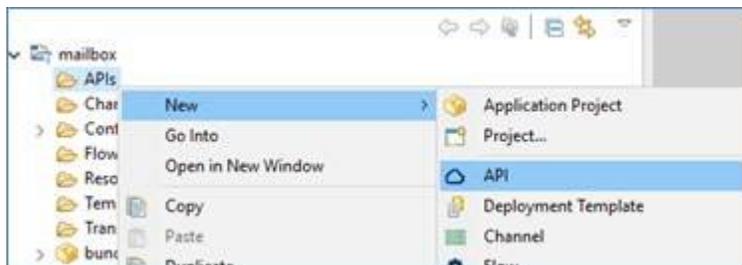
3. Provide a project name (for example, *mailbox*) and then click *Finish*.

A new application is created, containing the required project folder structure, as shown in the following image.



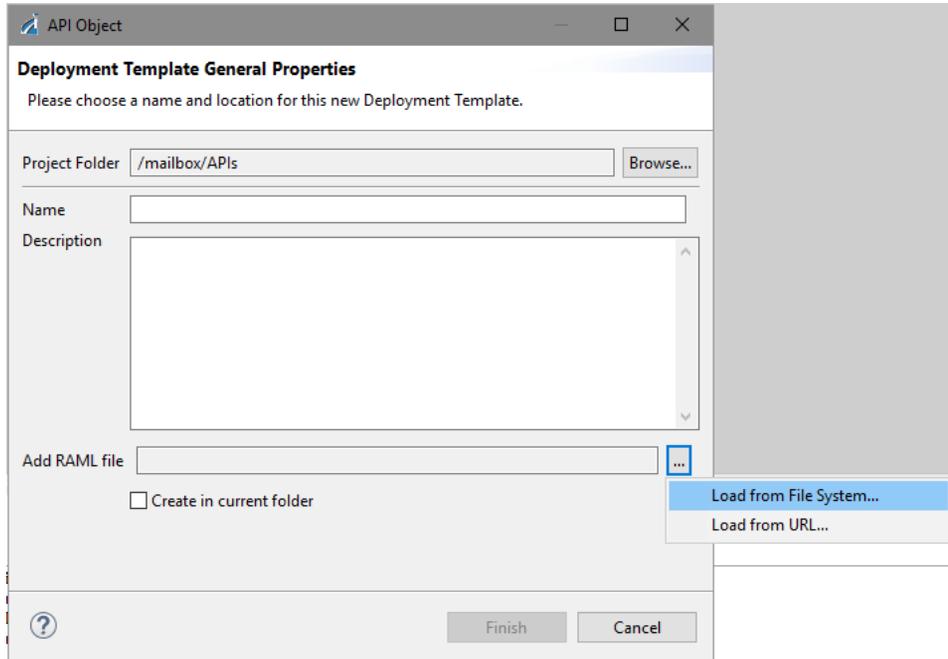
The *bundle* folder is the application package name, which you can rename if required. During deployment, you will be prompted to select the name of the deployed application.

4. Right-click the APIs folder, select New from the context menu, and then click API, as shown in the following image.



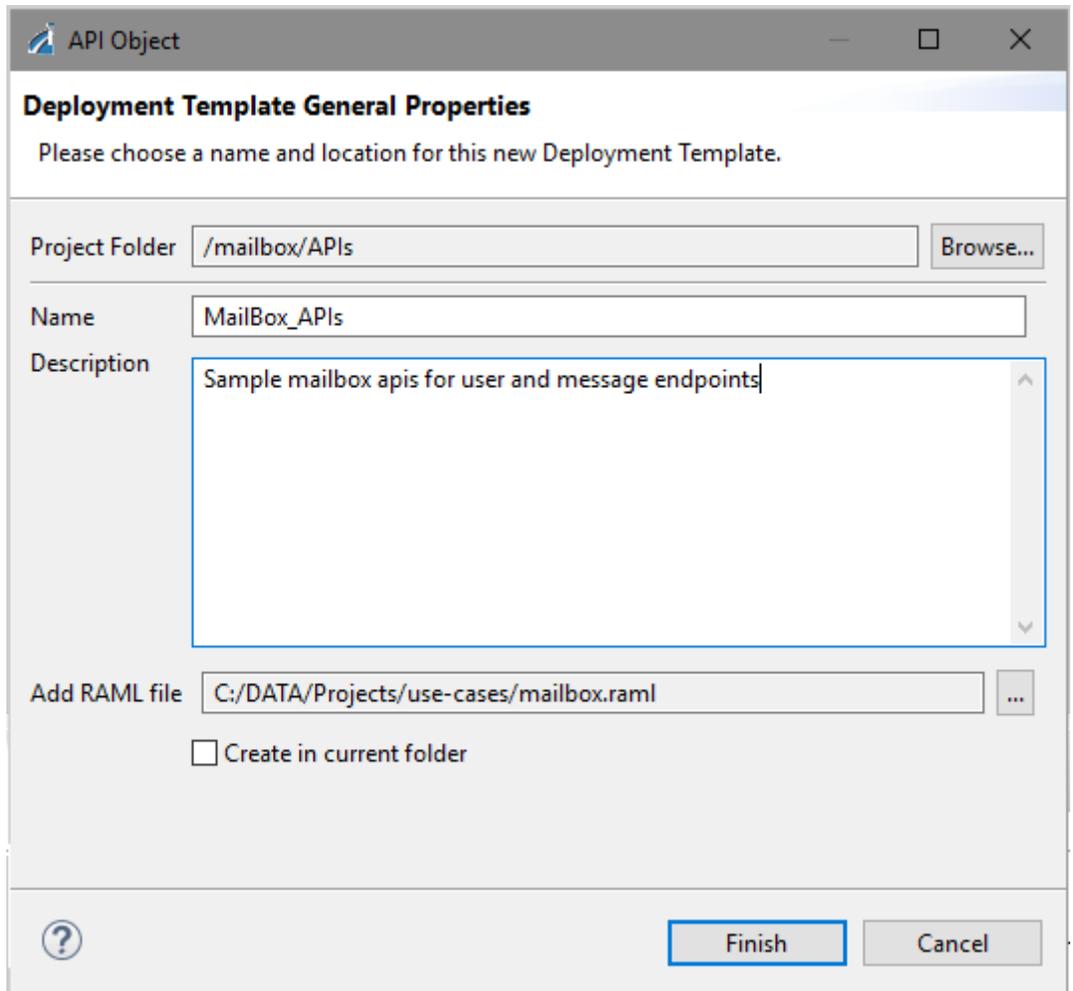
It is highly recommended that you create the API based on a RAML file definition. A sample RAML file (*mailbox.raml*) is provided with this use case. However, there are many samples that you can find and download online.

The API Object dialog opens showing the Deployment Template General Properties pane.



5. In the Add RAML file field, click the Browse (...) button and then select *Load from File System*.
6. Navigate to the location on your file system where the sample *mailbox.raml* file is located.

After selecting the *mailbox.ram* file, you are returned to the Deployment Template General Properties pane, as shown in the following image



7. Provide a name (for example, *MailBox_APis*) and an optional description.
8. Click Finish.

The RAML file is parsed and the structure will be presented for configuration purposes.

Defining the Business Logic to be Executed for each API Endpoint and its Corresponding Action

Each of the actions for any given endpoint has an Edit option located on the right pane. This enables you to assign business logic (a process flow) to be executed for a specific action on the given endpoint.

1. In the API Editor, select a GET action, and then on the right pane, click *Edit*, as shown in the following image.



The default process flow opens in a new tab, which will either have only a START and END object, or will also include a PAYLOAD object between the START and END objects. A PAYLOAD object is available if the RAML definition included a sample response document.

The following image shows a sample `/{userid}/profile` GET operation where the PAYLOAD object is preconfigured with information from the RAML file and provides a sample response.

The screenshot displays an IDE interface for defining API endpoints. The top pane shows a process flow diagram on a grid background. The flow starts with a 'Start' node (hand icon), followed by a 'Payload' node (truck icon), and ends with an 'End' node (circle icon). The 'Payload' node is highlighted with a red box. Below the diagram, the 'Payload Object' configuration panel is visible. It includes tabs for 'Configuration', 'Payload', 'Pre-Execution', 'Post-Execution', and 'General'. The 'Payload' tab is active, showing a 'Format' dropdown set to 'json' and a text area containing a JSON object:

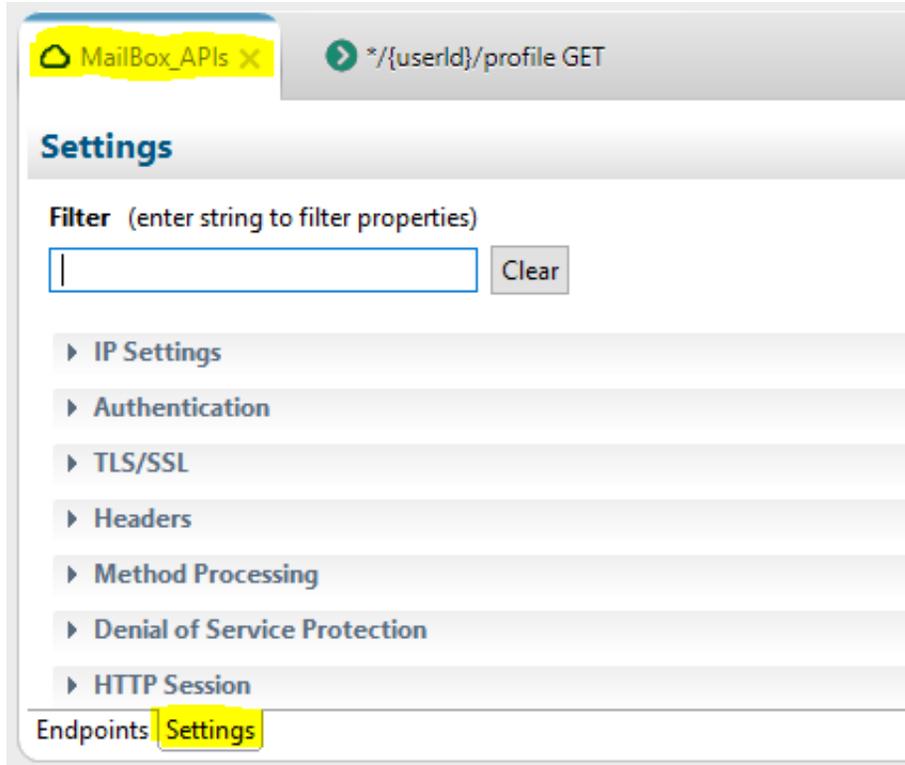
```
{ "id": "1234", "lastName": "L_name", "firstName": "f_name", "dob": "27/12/1954", "str_address": "1234 my street", "city": "mycity", "state": "mystate" }
```

2. If required, you can change the preconfigured values of the sample response for testing purposes and save the application project.

In a real-world scenario, the process flow for each action endpoint will be updated to host the business logic to render the response for the specific application. This can be a simple or very complex process flow.

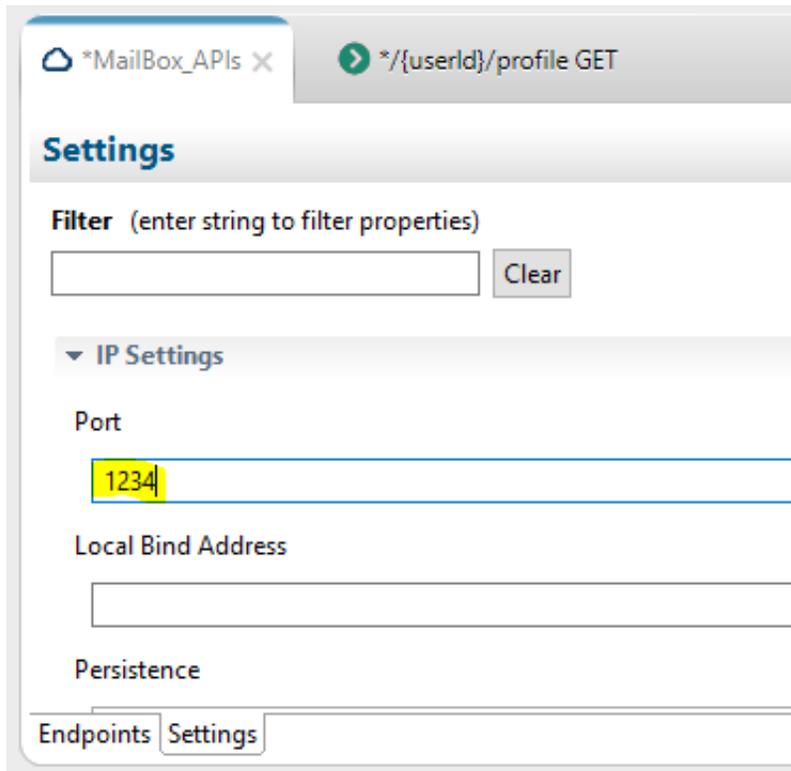
Configuring the API Settings for the HTTP Communication Layer

1. Click the *MailBox_APIs* tab, and then in the middle of the screen, click Settings, as shown in the following image.



This enables you to configure the HTTP properties for hosting the APIs. APIs consist of multiple endpoints where each API is hosted by a single HTTP listener. The application itself can have multiple APIs hosted on many HTTP listeners.

2. Expand the *IP Settings* section and provide a port number that is not currently being used by your system (for example, *1234*), as shown in the following image.



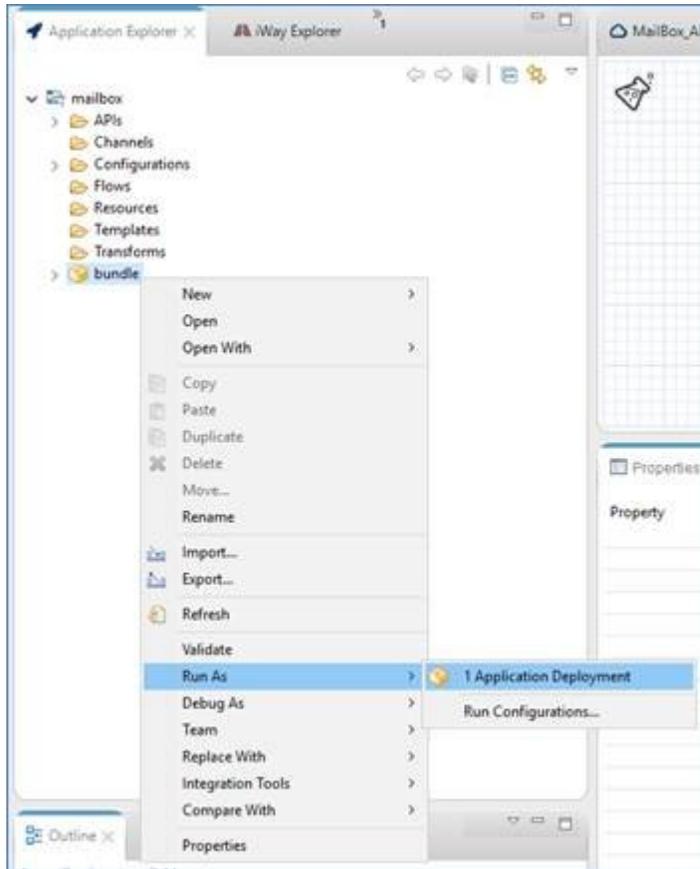
Note: You must provide a valid port number. For more information on any of the other parameters in this section, see the *iWay HTTP Solutions Development Guide*.

3. Save your work by clicking the multi-disk icon.

Your application is now ready for deployment.

Deploying the Application

1. Right-click the *bundle* folder (or the new name you provided for this folder), select *Run As* from the context menu, and then click *Application Deployment*, as shown in the following image.



The application deployment (Main) dialog opens, as shown in the following image.

The image shows a 'Main' dialog box for application deployment. It contains several sections:

- Application:** A text field containing '/mailbox/bundle.iab' and a 'Browse...' button.
- Template:** Two radio buttons. The first is selected and points to an empty text field with a 'Browse...' button. The second is unselected and points to a dropdown menu with a 'Refresh' button.
- Server Environment:** A 'URL:' dropdown menu with 'http://localhost:9000', a 'User Name:' text field with 'iway', and a 'Password:' text field with four dots.
- Deployment Options:** A 'Deployment Name:' text field with 'mail_apis', an empty 'Console Port:' text field, a 'Description:' text area with 'sample mail apis', and an 'Autostart Application:' checkbox which is unchecked.

At the bottom of the dialog, there are buttons for 'Revert', 'Apply', 'Run' (highlighted with a blue border), and 'Close'. A help icon (?) is located in the bottom-left corner.

Note: The first time you deploy your application, you are prompted to provide values for the deployment properties. For any subsequent deployments or redeployments, you are prompted to reconfirm the replacement of the previous deployment. If you wish to reconfigure the deployment properties, then select *Run Configurations*.

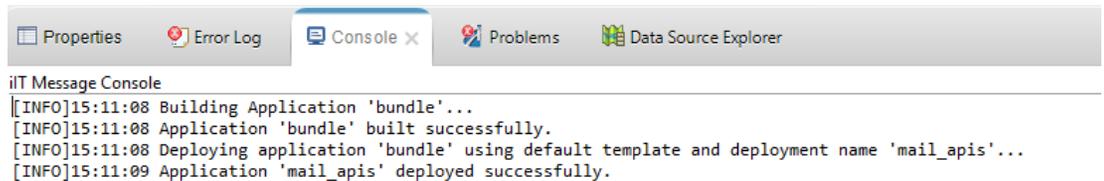
2. In the Server Environment section, provide the URL and credentials for the iWay Service Manager (iSM) instance where your application will be deployed.
3. In the Deployment Options section, provide the deployment name for your application, which will be the actual name of the deployed/running application, then provide an optional description.

You also have the Autostart Application option, which you can leave unchecked for this deployment instance.

If your application required a runtime template with server-based properties, you can also select the specific template to apply to the deployment. You will not be specifying a template for this deployment instance.

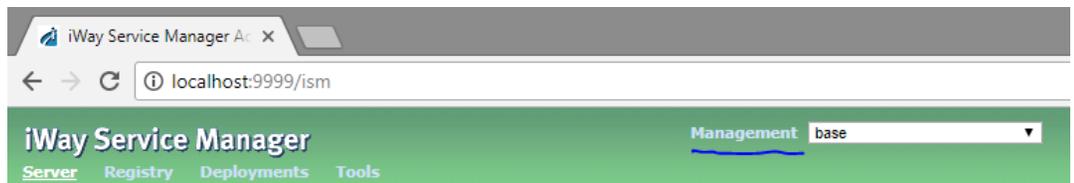
4. Click *Apply* and then click *Run*.

The application is deployed successfully and the Console tab should display messages similar to the ones shown in the following image.



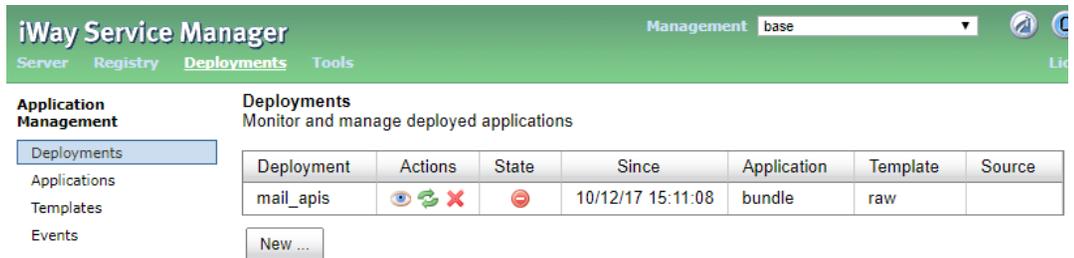
Starting and Testing the Application Using a Browser

1. Open the iSM Administration Console (the default is `http://localhost:9999` with admin/admin credentials).
2. Click the *Management* link in the upper-right corner, as shown in the following image.



3. Click *Deployments*.

Your application is listed in the Deployments pane, but is not yet started, as shown in the following image.



4. Start your application either from the Windows Services dialog, a command prompt, or from the iSM Administration Console by clicking on the red minus icon in the State column.

Once the deployed application has started, the icon in the State column will change to a green check mark, as shown in the following image.

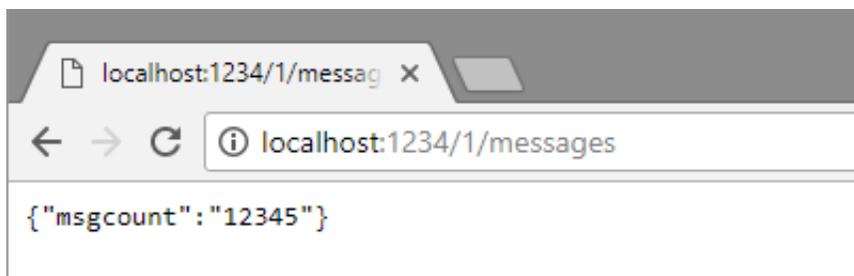
Deployment	Actions	State	Since	Application	Template	Source
mail_apis	  		10/12/17 15:11:08	bundle	raw	

You can use your browser to test the API call. Since no business logic (process flows) was modified, which are hosted by the end-points, you should receive the default response which was provided in the Payload objects as defined by the sample RAML file.

5. Open a new browser tab/window and enter the following URL:

<http://localhost:1234/1/profile>

You will receive a default JSON response, as shown in the following image.



You have now hosted your first set of APIs providing two endpoints for consumers. You can proceed to create your own RAML definitions, define business processes to be executed for the endpoints, and expose your APIs to internal and external consumers.



Feedback

Customer success is our top priority. Connect with us today!

Information Builders Technical Content Management team is comprised of many talented individuals who work together to design and deliver quality technical documentation products. Your feedback supports our ongoing efforts!

You can also preview new innovations to get an early look at new content products and services. Your participation helps us create great experiences for every customer.

To send us feedback or make a connection, contact Sarah Buccellato, Technical Editor, Technical Content Management at Sarah_Buccellato@ibi.com.

To request permission to repurpose copyrighted material, please contact Frances Gambino, Vice President, Technical Content Management at Frances_Gambino@ibi.com.



iWay

/ iWay Integration Tools Getting Started Guide

Version 8.0 and Higher

DN3502333.1020

Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898