

IBM Application Discovery and Delivery
Intelligence Extensions
6.1.1

User Guide



Note

Before using this information and the product it supports, read the information in [Chapter 15, “Notices,”](#) on page 293.

October 2022 edition

This edition applies to 6.1.1 of IBM® Application Discovery and Delivery Intelligence Extensions (ADDI Extensions) and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure that you use the correct edition for the level of ADDI Extensions.

For more information about IBM Application Discovery and Delivery Intelligence for IBM Z®, visit IBM ADDI site at: <https://www.ibm.com/products/app-discovery-and-delivery-intelligence>.

© **Copyright International Business Machines Corporation 2016, 2022.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Product overview.....	1
Chapter 2. Terminology.....	3
Chapter 3. Tutorials.....	5
Installing and setting up ADDI Extensions.....	5
Setting up LDAP connections for ADDI Extensions.....	18
Setting up and analyzing code coverage for the Manual Builds data provider.....	67
Creating a Manual Builds data provider as a data source for code coverage results.....	68
Defining a new application as an analysis workbook.....	71
Viewing the code coverage analysis results of the first build.....	74
Adding a new build for code coverage analysis.....	77
Performing the code coverage analysis.....	79
Setting up automated code coverage data collections.....	83
Adding a Manual Builds data provider to collect code coverage data automatically.....	83
Setting up a workbook to view code coverage results from automated data collection.....	89
Exercising setting up a Manual Builds data provider for code coverage analysis.....	91
Setting up a Manual Builds data provider.....	91
Setting up a workbook.....	95
Reviewing code coverage reports and dashboards.....	97
Generating sample data.....	106
End-to-end system performance root cause analysis.....	116
Analyzing the static analysis metrics from the Application Discovery data provider.....	129
Chapter 4. Installation and setup.....	139
Hardware and software requirements.....	139
Security considerations.....	140
Installing and setting up ADDI Extensions with ADDI installer.....	141
Creating the database.....	142
Uninstalling IBM ADDI Extensions.....	143
Chapter 5. Migrating from a previous release.....	145
Chapter 6. Administration.....	151
Starting up the server.....	151
Shutting down the server.....	151
Server configuration settings.....	151
Configuring ADDI Extensions Install Configuration Page.....	152
Authentication setup.....	154
Configuring the parameters in the dex.yaml file.....	155
Starting the Authentication Server (DEX).....	159
Configuring Authentication Server.....	159
Configuring User Groups.....	160
Performing ADDI Extensions setup.....	161
Updating server.xml for communication between ADDI Extensions server and Authentication Server (DEX).....	161
Running the adi-setup script.....	162
References to the adi-setup script.....	162
Backing up data.....	163
Installing a security certificate into Liberty.....	163

Accessing Business Rule Discovery repository through APIs.....	163
Generating sample data for evaluation.....	164
Generating OMEGAMON for CICS data.....	164
Generating static analysis sample data.....	165
Chapter 7. Preparing external data sources.....	167
Preparing code coverage results for COBOL and PL/I.....	167
Preparing code coverage results for batch applications.....	167
Preparing code coverage results for CICS transactions by using the debug tool backend and IDz client.....	171
Preparing code coverage results for COBOL and PL/I by using RTC.....	172
Preparing code coverage results for Java.....	178
Preparing code coverage results by using the RAD code coverage solutions.....	178
Preparing code coverage results by using the ADDI Extensions code coverage generator.....	185
Preparing code coverage results by using the ADDI Extensions code coverage on server generator.....	192
Chapter 8. Managing connections.....	197
Creating a connection.....	197
Editing the connection information.....	197
Deleting a connection.....	198
Testing a connection.....	198
Chapter 9. Managing data providers.....	199
Data providers.....	199
Managing Manual Builds data providers.....	200
Adding a Manual Builds data provider.....	200
Modifying a Manual Builds data provider.....	202
Managing OMEGAMON for CICS data providers.....	206
Adding an OMEGAMON for CICS data provider.....	206
Modifying an OMEGAMON for CICS data provider.....	206
Managing Rational Team Concert Builds data providers.....	207
Adding a Rational Team Concert Builds data provider.....	208
Modifying a Rational Team Concert Builds data provider.....	209
Managing System Management Facility data providers.....	211
Adding a System Management Facility data provider.....	211
Modifying a System Management Facility data provider.....	212
Managing Application Discovery and Business Rule Discovery data providers.....	213
Creating an Application Discovery data provider.....	213
Creating a Business Rule Discovery data provider.....	213
Editing the information of a data provider.....	214
Collecting the data to get updated data from a data provider.....	214
Viewing the data collection logs.....	214
Deleting a data provider.....	215
Chapter 10. Managing workbooks.....	217
Adding a workbook.....	217
Modifying a workbook.....	218
Pinning a workbook.....	219
Deleting a workbook.....	220
Chapter 11. Analyzing workbooks.....	221
Analyzing and displaying code coverage results.....	221
Code coverage reports and information.....	223
Analyzing code coverage trends.....	232
Comparing code coverage results with baseline builds.....	233
Analyzing pinned files.....	233

Filtering data in the Build Analysis view.....	234
Searching for files, packages, or directories within a build.....	235
Analyzing the Tests to Run.....	235
Analyzing code coverage of flowpoint level break-down.....	236
Analyzing and displaying operational data from IBM OMEGAMON for CICS.....	237
OMEGAMON for CICS related reports and information.....	238
Analyzing transaction data.....	242
Analyzing and displaying performance data from SMF and APA data.....	243
SMF and APA related reports and information.....	244
Analyzing the job performance data.....	245
Performing Business Rule Discovery.....	246
Business Rule Discovery workbook reports and information.....	246
Working with keywords.....	251
Managing a business term.....	255
Managing business rule packages.....	262
Analyzing and displaying project data from Application Discovery.....	266
Project data from Application Discovery reports and information.....	268
Analyzing static metrics for a workbook.....	271
Analyzing static metrics for a project.....	273
Analyzing shared resources.....	274
Analyzing project data.....	274
Analyzing the Artifact Composition.....	274
Comparing the projects within a workbook.....	279
Chapter 12. Troubleshooting.....	281
Chapter 13. Providing feedback.....	289
Chapter 14. Accessibility features.....	291
Chapter 15. Notices.....	293
Trademarks.....	294
Index.....	295

Chapter 1. Product overview

Learn about the basic knowledge about IBM Application Discovery and Delivery Intelligence Extensions (ADDI Extensions).

Throughout the DevOps lifecycle, a very large amount of data can be produced, including code, requirements, designs, test artifacts, and operational information about applications. IBM Application Discovery and Delivery Intelligence Extensions (ADDI Extensions) is a web-based tool to facilitate enterprise DevOps adoption and drive continuous improvement across the lifecycle of product development, delivery, and maintenance.

Key features

ADDI Extensions provides an insight dashboard with actionable recommendations to enterprise through the following key features:

- Analyzing and visualizing historical trend information on application usage, code coverage, complexity, quality, and performance metrics
- Providing near real-time application and process health checks
- Analyzing code coverage data and recommending tests to run to achieve better tests and code coverage
- Monitoring previously hidden performance and resource data to make performance issues exposed to development team
- Showing the application inventory and quality trends to reduce the development and maintenance costs
- Highlighting the artifacts that are related to the code quality concerned areas and providing the call graph of the corresponding transaction that shows the artifact complexity metrics of risk assessment
- Facilitating the inventory creation of business and IT artifacts for future usages such as code refactoring and API discovery in order to modernize applications
- Making information accessible to stakeholders through a simple and intuitive interface

Key benefits

With the capabilities of ADDI Extensions, users can analyze the runtime performance data and the testing details over time. ADDI Extensions pro-actively alerts users on early problem detection so that resources can be shifted to the areas that need attention. The following typical users can benefit from the functions of ADDI Extensions to improve their DevOps process.

By using the code coverage function, testers can:

- See the current and previous test coverage status in a single dashboard view.
- Determine the test coverage of a build and identify the modules that need more testing for each build.
- Develop a test list that contains the minimum number of test cases to run for a regression for the next build.

By using the code coverage function, test managers can:

- Receive recommendations for test case optimization.
- Identify testing coverage issues. With the help of test coverage trends to examine overall release quality, test managers can act sooner in the development cycle.

By leveraging OMEGAMON for CICS data or System Management Facility data, developers can:

- Be alerted about potential performance problems.
- Analyze transactions or jobs that have performance issues.
- See whether their code changes drive up response time that includes DB2 and file I/O percentage wait time with the performance information of transactions that they own and the associated trend data.

By monitoring the application inventory and quality trends, developers can:

- Visualize whether their codes are becoming easier to maintain.
- Identify unreachable codes to further simplify the code base.

By monitoring and comparing maintainability trends, development and test managers can:

- Ensure accountability of outsourced projects and in-house projects alike.
- Enable quantifiable improvement objectives to be established by organization.

By building an inventory of business and IT artifacts, enterprise architects can:

- Quickly discover the business terms and implementation terms from both business artifacts and IT artifacts.
- Have a dictionary that maps the business terms to their implementation terms to make the business stakeholders and IT stakeholders understand the same language.
- Quickly see the scope of the IT artifacts that are impacted by the business terms.

Chapter 2. Terminology

You can find the technical terms and abbreviations that are used in ADDI Extensions.

To view the glossaries for other IBM products, go to www.ibm.com/software/globalization/terminology.

Application Discovery (AD) project

A project in AD is a user (typically an admin role) defined grouping of program artifacts (source codes), dependencies (copybooks or includes), build scripts (JCL) and other data, on which the static analysis is performed and analysis metrics are gathered.

build

The process of creating an executable to identify the user activities that an application intends to address.

For z Systems®:

- Only changed programs are built. If changes are made only in a sub program, only the subprogram is compiled and linked in with the UI module or main program.
- A build can consist of only one module, depending on the size of the application and the changes that are going in.
- The test team tests the whole application, which includes all the programs that constitute an application.

For example, getting an insurance quote consists of:

- A UI program to accept user input and display the insurance quote or errors.
- A main program that validates the user input.
- A sub program that is called by the main program to get or calculate the insurance quote and return the value or error to the main program.

All these programs, modules, or files need to be built together to verify the insurance quote application tasks.

business rule

The business rule describes a business policy, business operation or business procedure. For business applications or enterprise applications, business rules are commonly expressed in programming language statements with one or more business terms and condition logic.

business rule package

An ADDI term that represents the information of a business rule. It groups business terms and snippets that are related and represents business logic or a business rule.

business term

A terminology that is commonly used in business operations. It provides a definition of the key business information. The list of business terms is usually different by industry, for example, there are business terms for financial, banking, and healthcare industry. For business applications or enterprise applications, business terms are usually implemented in the system. Hence, there is traceability between the business terms and the IT artifacts.

code coverage

A measure of how much of code in a program or file is being executed as part of a test or test case. The result can contain coverage for one or more programs or files, depending on how the test is structured.

CICSplex

A CICSplex is an environment in which multiple and interconnected CICS regions operate. A CICSplex lets CICS and associated applications use the full capabilities of Parallel Sysplex® architecture.

data provider

Data source. A workbook can have multiple data providers that ideally cover the DevOps entry points.

file / program

A software artifact that contains the logic to perform one or more of the application's activities.

Examples:

- Updating a database record
- Calculating the insurance quote
- Calculating mortgage rate
- Validating UI fields
- Querying the account balance

implementation name

A name or term that represents the business term in the IT artifacts. It is the term by which the business term is implemented in the source code of business applications or enterprise applications. For example, in COBOL applications the implementation name can be the data-name; in Java™ applications the implementation name can be the variable name or class name.

keyword

A term that has potential to become a business term or an implementation name. The term is discovered by ADDI Extensions from the AD static analysis or enterprise artifacts.

service class

IBM OMEGAMON® for CICS® allows users to define a service policy, that is, a set of categorized goals called service classes that define the targeted performance throughput and which transactions the goal applies to. And users can apply the service policy to the workload running in the environment. A default service policy and set of service classes is provided upon installation of IBM Tivoli® OMEGAMON XE for CICS. Users can customize the default service policy and service classes and also add additional service classes.

shared resources

Data sources that are used by multiple projects within the Application Discovery data provider. Data source types can be database table, IMS database, or data set.

snippet

A small part of source code or a small part of text from enterprise artifacts.

test / test case

A set of steps to verify a specific activity, a set of activities, or part of an activity that the application is intended to support.

For IBM Z:

- Tests are mostly manual tests.
- A test case might or might not contain a list of steps to go through.

Examples:

- Running a transaction.
- Submitting a batch job or jobs with different set of inputs.

workbook

ADDI Extensions uses the concept of a workbook to define the scope of artifacts on which to perform analysis. You can group the related artifacts from different data providers in a workbook. One workbook can be associated with one or more data providers.

Chapter 3. Tutorials

Tutorials are available to help you learn how to get started with ADDI Extensions for demonstration and evaluation purposes.

Installing and setting up ADDI Extensions

This tutorial guides you through the installation and setup of ADDI Extensions to be used for evaluation or demonstration purposes.

Note: The procedures in this tutorial are based on a Windows system.

System requirements

To set up the environment for the tutorial, you need a system with 32 GB RAM with Windows 10, x86-64 operating system. If you have a system with less memory, you need to stop the following IBM Application Discovery Services:

- IBM Application Discovery Analyze Service
- IBM Application Discovery Batch Service
- IBM Application Discovery File Service
- IBM Application Discovery Graph DB Service
- IBM Application Discovery Mainframe Projects Service
- IBM Application Discovery Manual Resolution Service
- IBM Application Discovery Search Service

Before you begin

Before the installation, you need to update the `hosts` file to understand the public URI that you use for the setup. For example, complete the following steps to update the `hosts` file.

1. Type Notepad in the search box next to the **Start** menu icon.
2. In the search results, right-click **Notepad**, and select **Run as administrator**.
3. From the Notepad, open the `hosts` file in the `C:\Windows\System32\drivers\etc` directory.
4. Add the following entry under the `localhost` section as shown in the example.

```
127.0.0.1    healthcare.example.com
```

```
# localhost name resolution is handled within DNS itself.
```

```
127.0.0.1    localhost
```

```
127.0.0.1    healthcare.example.com
```

```
#          ::1          localhost
```

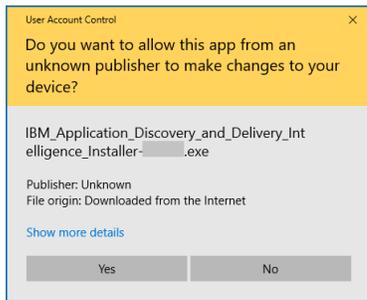
5. Save the `hosts` file.

Procedures

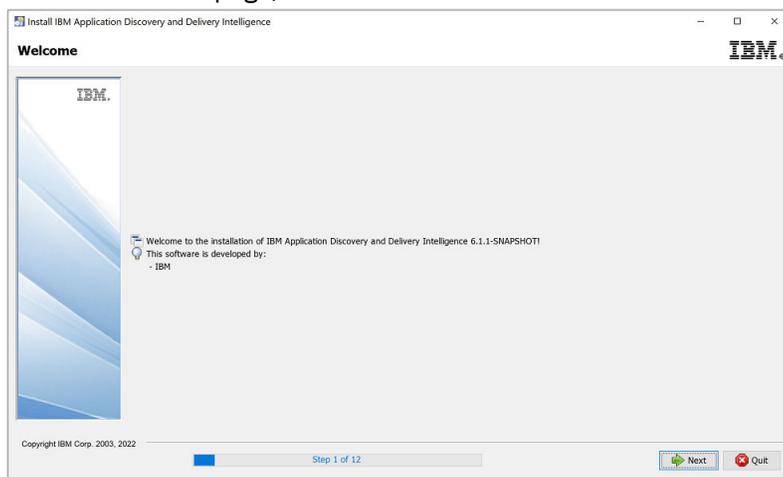
Complete the following steps to install and set up ADDI Extensions.

1. Download the ADDI installer.

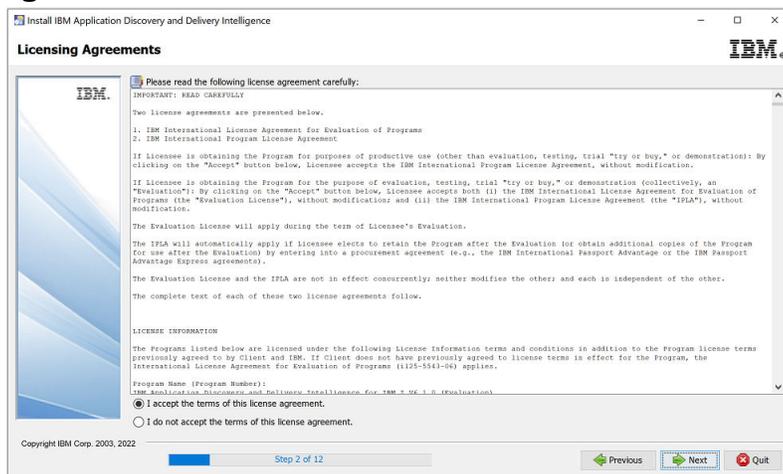
- Right-click the ADDI installer that you download and select **Run as administrator** to start the ADDI installer wizard.
- On the User Account Control dialog box, select **Yes** to allow the application make changes to your device.



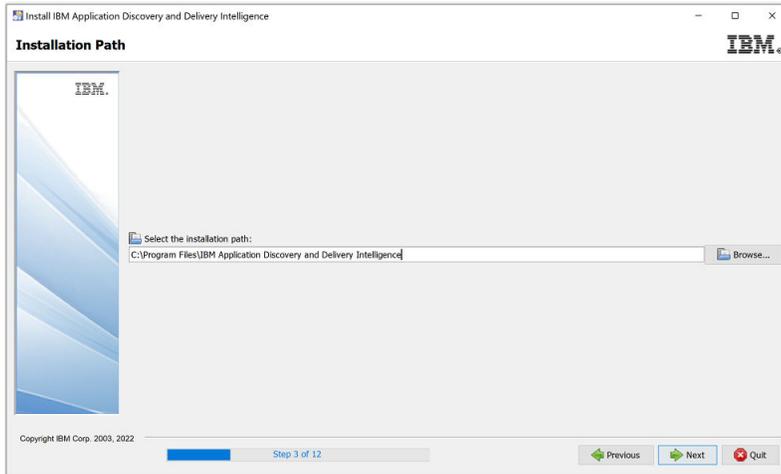
- Complete the following steps when the ADDI installation wizard appears.
 - On the **Welcome** page, click **Next**.



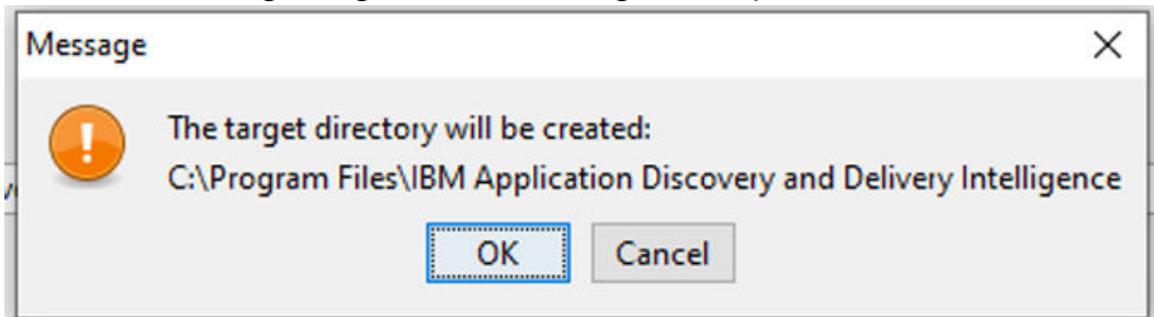
- Review the **Licensing Agreements** page. Then, select **I accept the terms of this license agreement** and click **Next**.



- On the **Installation Path** page, click **Next** to use the default path or click **Browse** to select a path to install IBM ADDI and click **Next**.

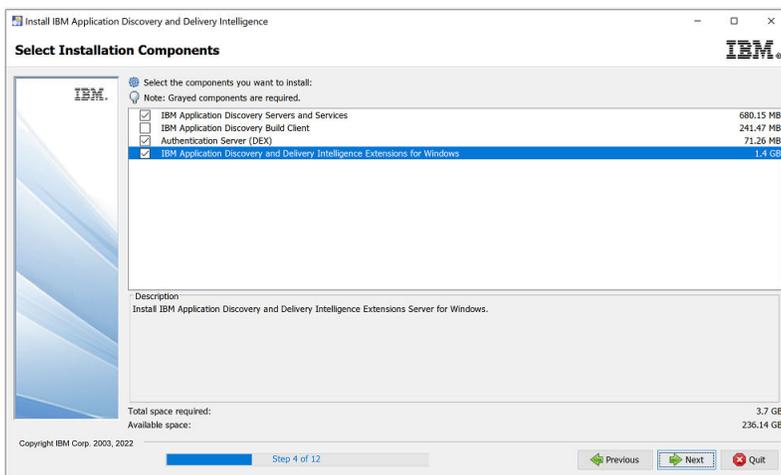


d. Click **OK** in the Message dialog box to create the target directory.

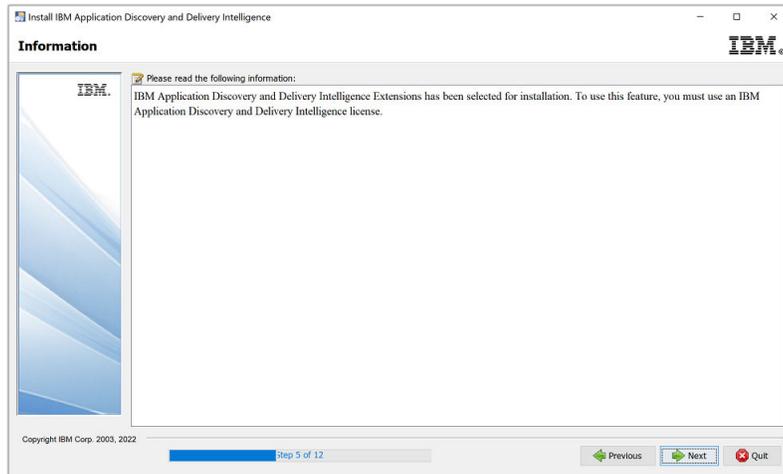


e. Check the following components to install and click **Next**.

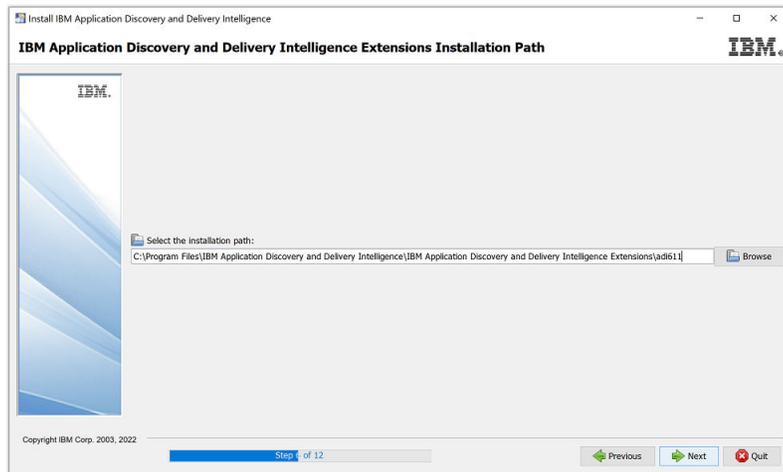
- IBM Application Discovery Servers and Services
- Authentication Server (DEX)
- IBM Application Discovery and Delivery Intelligence Extensions for *your system*. The following example shows the **Select Installation Components** page that is displayed on a Windows system.



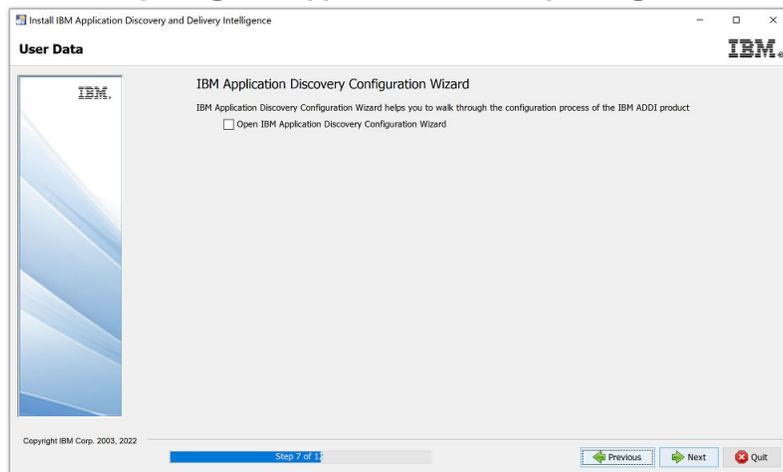
f. Click **Next** on the **Information** page.



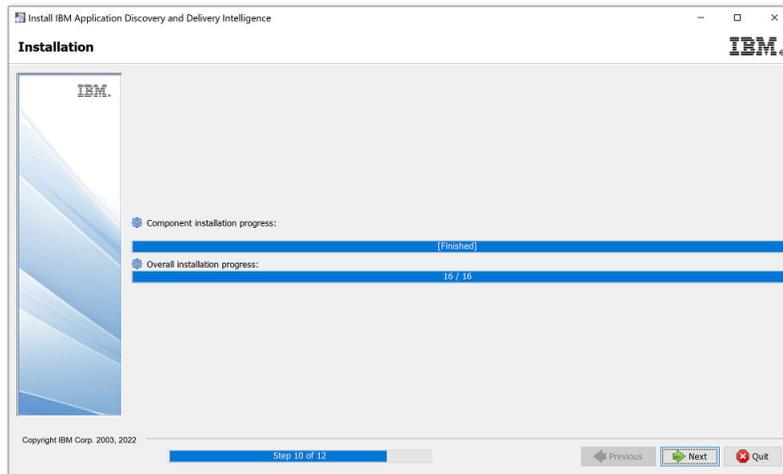
- g. On the **IBM Application Discovery and Delivery Intelligence Extensions Installation Path** page, click **Next** to use the default path or click **Browse** to select a path to install IBM ADDI Extensions and click **Next**.



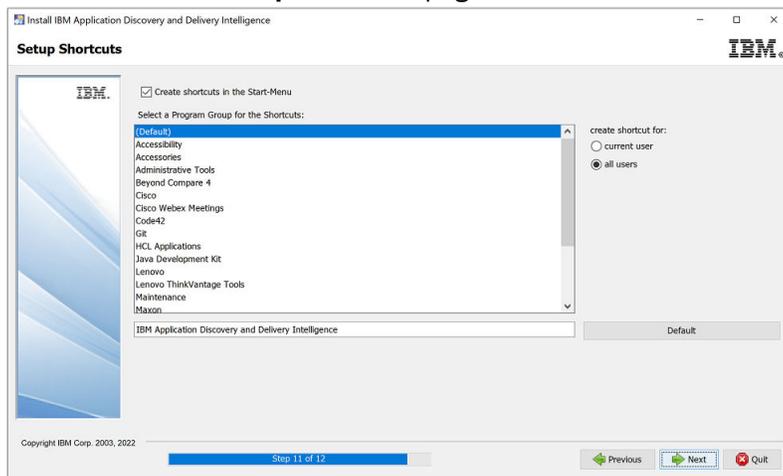
- h. Click **OK** in the Message dialog box to create the target directory.
i. Clear the **Opening IBM Application Discovery Configuration Wizard** checkbox.



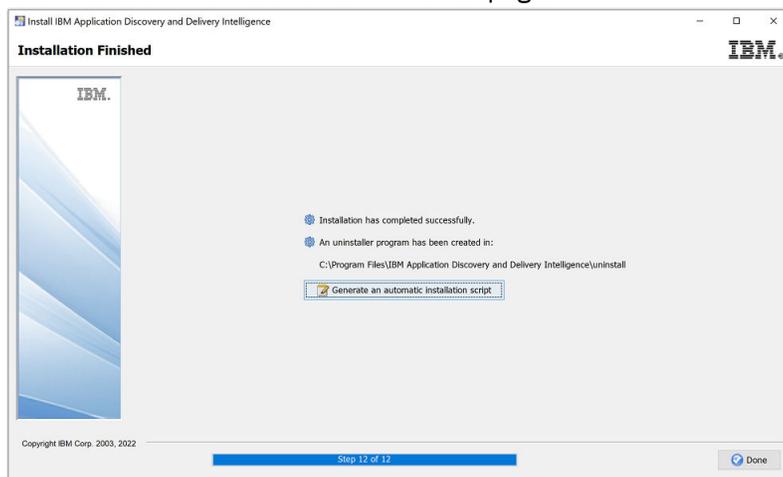
- j. Click **Next** on the **Installation** page when the installation progress is finished.



k. Click **Next** on the **Setup Shortcuts** page.



l. Click **Done** on the **Installation Finished** page.



5. Open the command prompt as an administrator and navigate to `c : \Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server` directory.

6. Run the following command to generate the bcrypt hash of an admin password:

```
adi-setup bcryptPassword -dex.password <password>
```

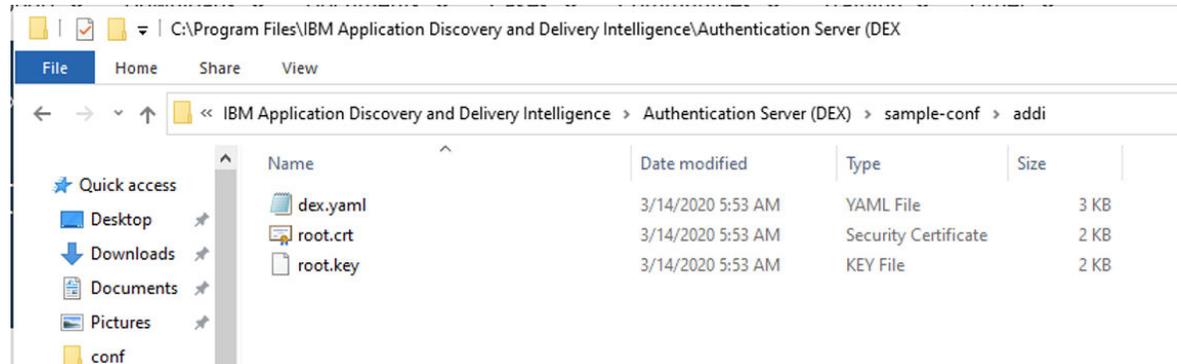
Note: In the following example, **adiadmin** is a password that you want to generate the bcrypt hash.

```

C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5107\server>
adi-setup bcryptPassword -dex.password adiadmin
Jun 25, 2020 11:56:55 AM com.ibm.dimez.tools.setup.common.AdiSetup showWelcome
INFO: CRIDA0398I ADI Setup started with operation "bcryptPassword".
Jun 25, 2020 11:56:56 AM com.ibm.dimez.tools.setup.common.AdiSetup bcryptPassword
INFO: CRIDA0654I The bcrypt hash of the password is: $2a$10$BrCnTGUMp.9Py7ezP6mOv.M/K6M80kqFzbzjwY4HS.YuQ5X4e4kNG
[BeanContext] = [2,0,0%]
[total] = [2,0,0%]

```

7. Save the generated bcrypt hash password somewhere. The password will be used when you set up the dex.yaml file.
8. Configure the Authentication Server (DEX) as described in the following steps:
 - a. Navigate to the C:\Program Files\IBM Application Discovery and Delivery Intelligence\Authentication Server (DEX)\sample-conf\addi directory.
 - b. Copy all three files in the directory: dex.yaml, root.crt, and root.key.



Note: All these files are provided for only evaluation purposes. For the production server, you must generate SSL keystore and security certificate for your server and configure your own dex.yaml file as described in the “Configuring the parameters in the dex.yaml file” on page 155 topic.

- c. Navigate to the c:\Program Files\IBM Application Discovery and Delivery Intelligence\Authentication Server (DEX)\conf\ directory and paste the copied files there.
- d. Run the text editor as the administrator and update the dex.yaml file with the following changes:

- i) Update the **issuer** to https://healthcare.example.com:7600/dex.

```
# The base path of dex and the external name of the OpenID Connect service.
issuer: https://healthcare.example.com:7600/dex
```

- ii) Update the **web** section with the following changes:

- Change the http property to https: healthcare.example.com:7600.
- Uncomment the TLSCert and TLSKey properties and update their paths as shown in the following sample.

web:

```
https: healthcare.example.com:7600
#if https is used provide path to certificate and key.
TLSCert: conf/root.crt
TLSKey: conf/root.key
```

- iii) Update the staticClients section with the following changes while removing the <<>> brackets.

- Update the id property to addi-liberty. Remove the extra leading space characters on this line.

- Replace the localhost within the redirectURIs property with healthcare.example.com.
- Update the name property to 'ADDI Liberty Server'.
- Update the secret to f1a75f8abc2ffc46e2c1b5f7b12c7b.
- Comment out lines from 77 through 81 by using # symbol in front of each of those lines.

```
staticClients:
- id: addi-liberty
  redirectURIs:
  - 'https://healthcare.example.com:9753/oidcclient/redirect/addi-liberty'
  name: 'ADDI Liberty Server'
  secret: f1a75f8abc2ffc46e2c1b5f7b12c7b
```

iv) Update the **StaticPasswords** section with the following changes while removing the <<>> brackets.

- Uncomment the email property and update it to "adiadmin@healthcare.example.com". Remove the extra leading space characters on this line.
- Uncomment the hash property and update it as the hash password that you saved in step 7 with double quotes.
- Uncomment the username property and update it to "adiadmin".
- Uncomment the userID property and update it to "adiadmin".

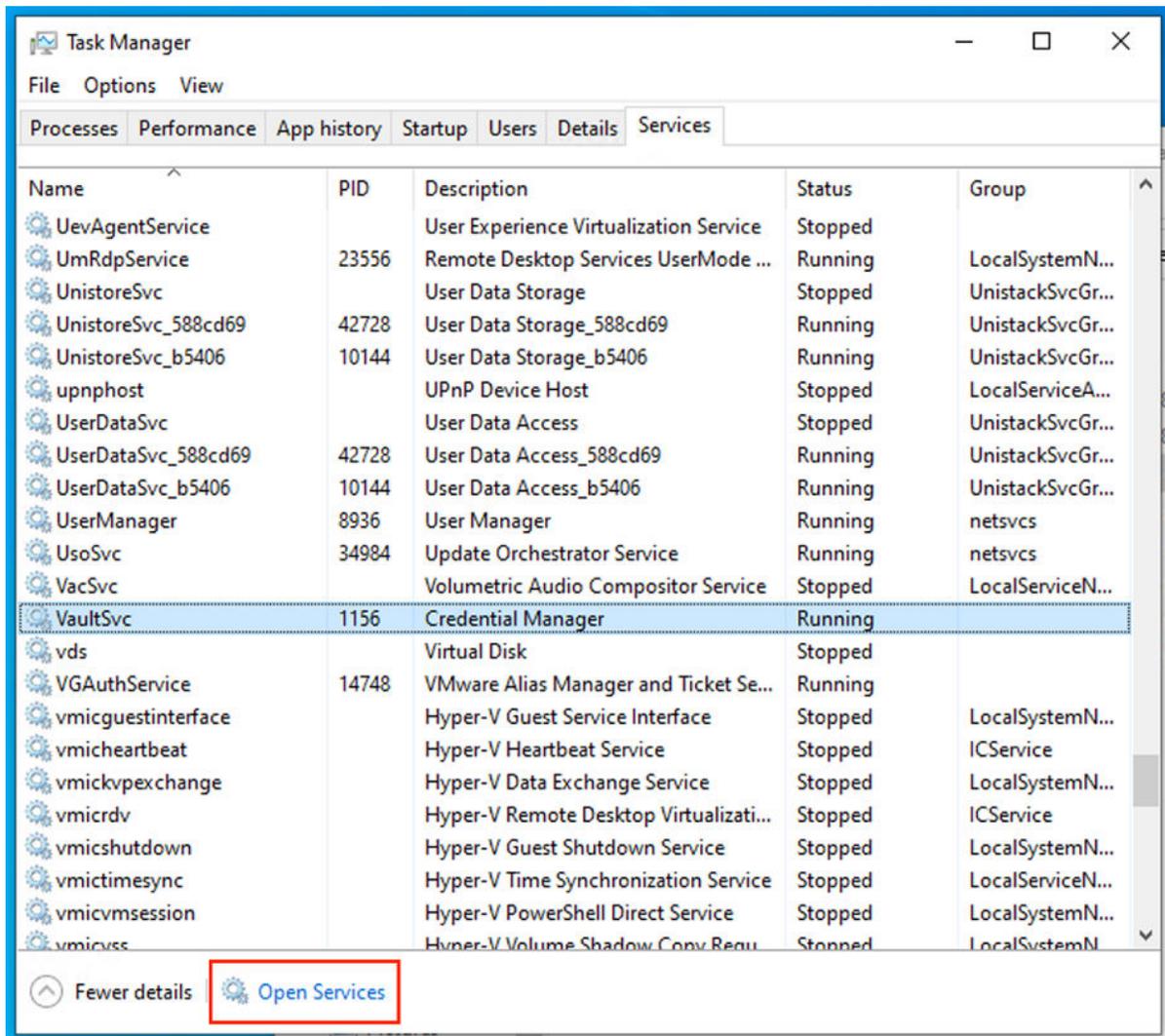
Note: Make sure that the hash, username, and userID properties are correctly indented to be in the same column under the email property. Remove any leading space characters to fix the indentation.

```
# A static list of passwords to login the end user. By identifying here, dex
# won't look in its underlying storage for passwords.
#
# If this option isn't chosen users may be added through the gRPC API.
staticPasswords:
- email: "adiadmin@healthcare.example.com"
  hash: "$2a$10$ztA5trILlRdTjVDg9LTe.OwVoYtfgIYaCxi9YQPCsrQmlWDIdMej2"
  username: "adiadmin"
  userID: "adiadmin"
```

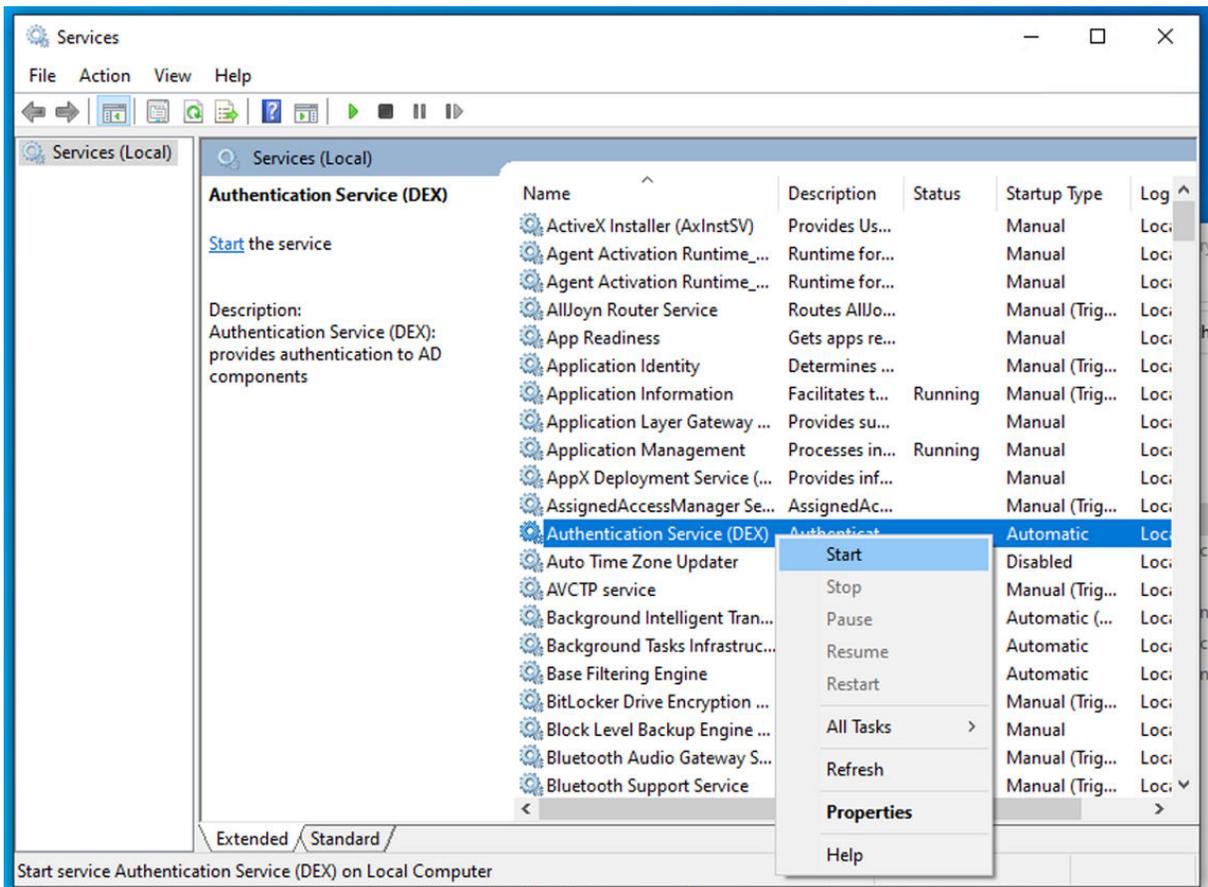
Note: The hash value comes from the value that you saved on step 7.

e. Save the dex.yaml file.

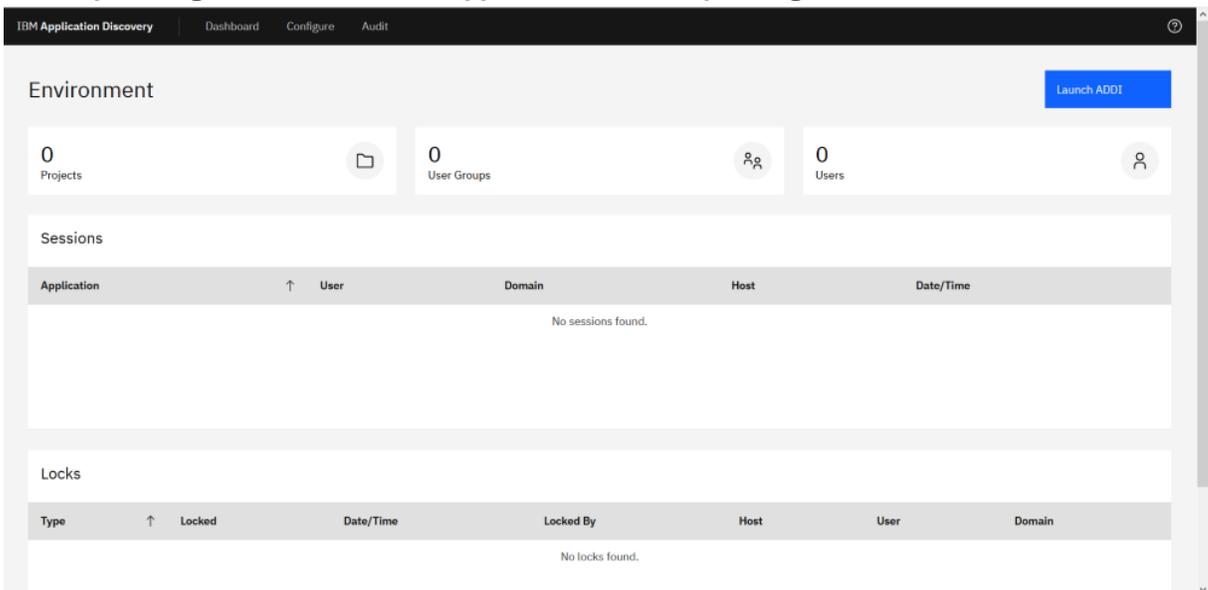
9. Press Ctrl + Alt + Del and choose **Task Manager** to open the Task Manager window.
10. Select the **Services** tab and click **Open Services** on the bottom of the **Task Manager** window.



11. Right-click **Authentication Service (DEX)** and select **Start** to start the service.

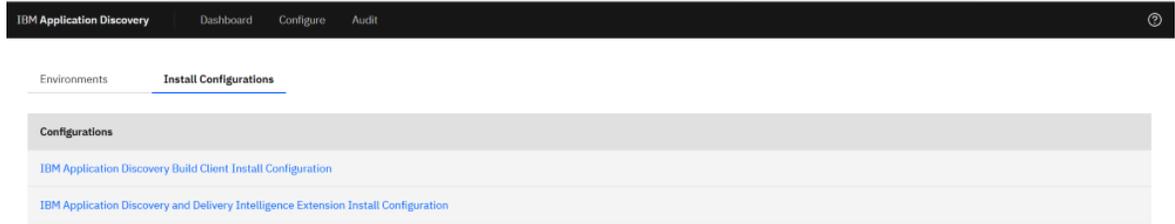


12. If the Authentication Service (DEX) fails to start and reports any errors, open the dex.log file from C:\Program Files\IBM Application Discovery and Delivery Intelligence\Authentication Server (DEX) and investigate it for any parsing or validation errors. If parsing or validation errors are found, open the dex.yaml file and fix the syntax errors on the reported lines and restart the Authentication Service (DEX).
13. To access **IBM AD Configuration Service**, go to **Start Menu > IBM Application Discovery and Delivery Intelligence > Launch IBM Application Discovery Configuration Service Admin**.



14. Complete the ADDI Extensions Install Configuration as described in the following steps:

- a. Select **Configure > Install Configurations > IBM Application Discovery and Delivery Intelligence Extensions Install Configuration.**



- b. Specify the Base URL on the **Web and Application Server** tab as shown in the following example.

`https://healthcare.example.com:9753`

[Install Configurations](#) /
IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server Databases Authentication Service User Groups

Base URL

`https://healthcare.example.com:9753`

Save

- c. Leave the default value on the **Databases** tab.

[Install Configurations](#) /

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server

Databases

Authentication Service

User Groups

Database server type

Derby ⓘ

DB2 ⓘ

Save

d. Specify the following information on the **Authentication Service** tab.

[Install Configurations](#) /

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server

Databases

Authentication Service

User Groups

Host

healthcare.example.com

Port

7600

HTTP protocol

HTTP

HTTP Secure (https)

Save

e. Specify the following information on the **User Groups** tab.

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server Databases Authentication Service **User Groups**

Admin Group List (optional)

agroup1

User Group List (optional)

group1

Save

- f. Click **SAVE** to create the ADDI Extensions configuration. A message is displayed to indicate the configuration was successfully created.

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server Databases Authentication Service **User Groups**

Admin Group List (optional)

agroup1

User Group List (optional)

group1

✓ Saved!

Save

15. On your Command Prompt window, navigate to c:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server directory.

Note: If you have closed the Command Prompt previously, you need to run the Command Prompt again as an administrator.

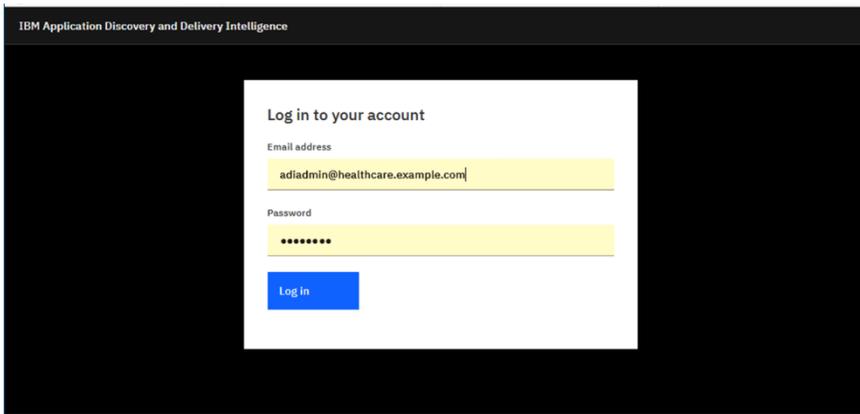
16. Run the command `adi-setup addiConfigurationServer`.

```
C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5107\server>
adi-setup addiConfigurationServer
Jun 25, 2020 12:11:47 PM com.ibm.dimez.tools.setup.common.AdiSetup showWelcome
INFO: CRIDA0398I ADI Setup started with operation "addiConfigurationServer".
[BeanContext] = [2,0,0%]
[UrlEncodingSerializer] = [1,0,0%]
[JsonSerializer] = [1,1,50%]
[JsonParser] = [1,1,50%]
[total] = [5,2,28%]
```

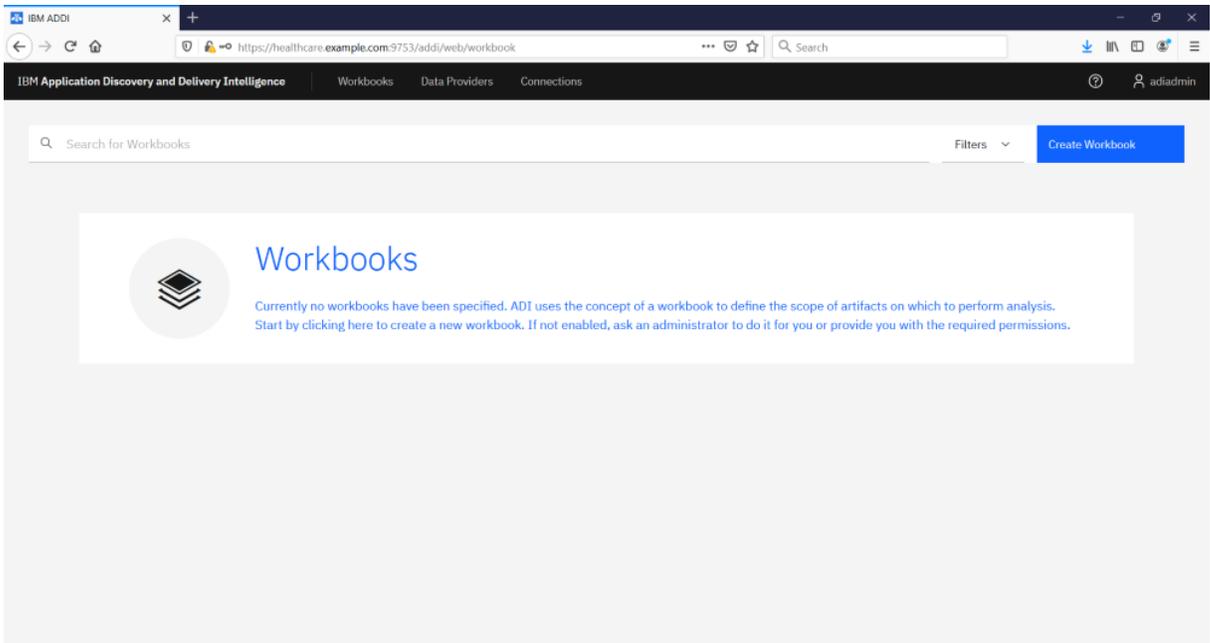
17. Run the `server.startup.bat` command and wait until the server is started successfully.

```
C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5107\server>
server.startup.bat
"Installing adi-elasticsearch service. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Appl
ication Delivery Intelligence\adi5107\server\..\elasticsearch\logs directory for results."
Installing service      : "adi-elasticsearch"
Using JAVA_HOME (64-bit): "C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery
Intelligence\adi5107\server\jre"
The service 'adi-elasticsearch' has been installed.
"Starting adi-elasticsearch service. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Applic
ation Delivery Intelligence\adi5107\server\..\elasticsearch\logs directory for results."
The service 'adi-elasticsearch' has been started
"Starting Derby Network Server. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application
Delivery Intelligence\adi5107\server\ directory for results."
Starting server adiServer.
Thu Jun 25 12:12:39 EDT 2020 : Security manager installed using the Basic server security policy.
Thu Jun 25 12:12:40 EDT 2020 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connections
 on port 1527
Server adiServer started.
```

18. Browse to <https://healthcare.example.com:9753/addi/web/workbook> to test your access to ADDI Extensions.
19. Log in to your account for ADDI Extensions with the following credentials.
 - Email address: `adiadmin@healthcare.example.com`
 - Password: `adiadmin`



After you log in, the **Workbooks** page is displayed, which is the home page of ADDI Extensions.



Setting up LDAP connections for ADDI Extensions

This tutorial guides you through the setup of ADDI Extensions with the LDAP connections through Microsoft Active Directory Domain Services (AD DS) for user management.

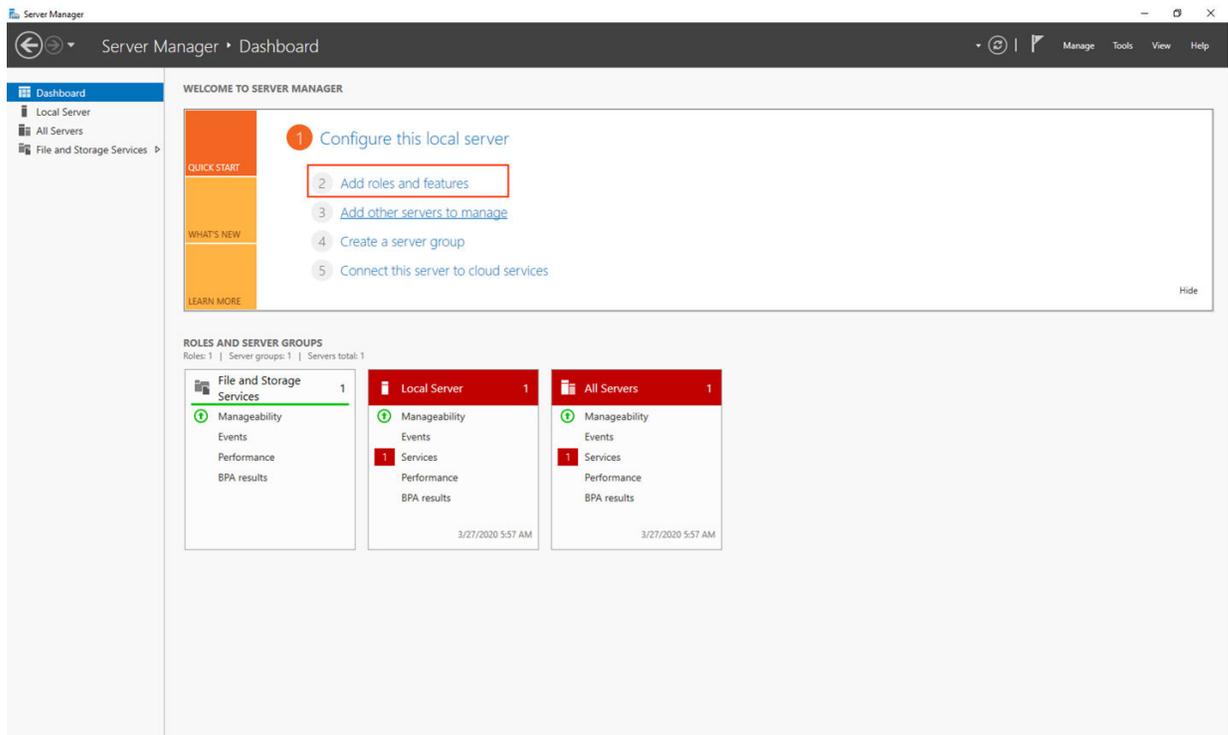
Prerequisite

In this tutorial, it is required to have a machine with the Windows Server 2012 or above for LDAP setup.

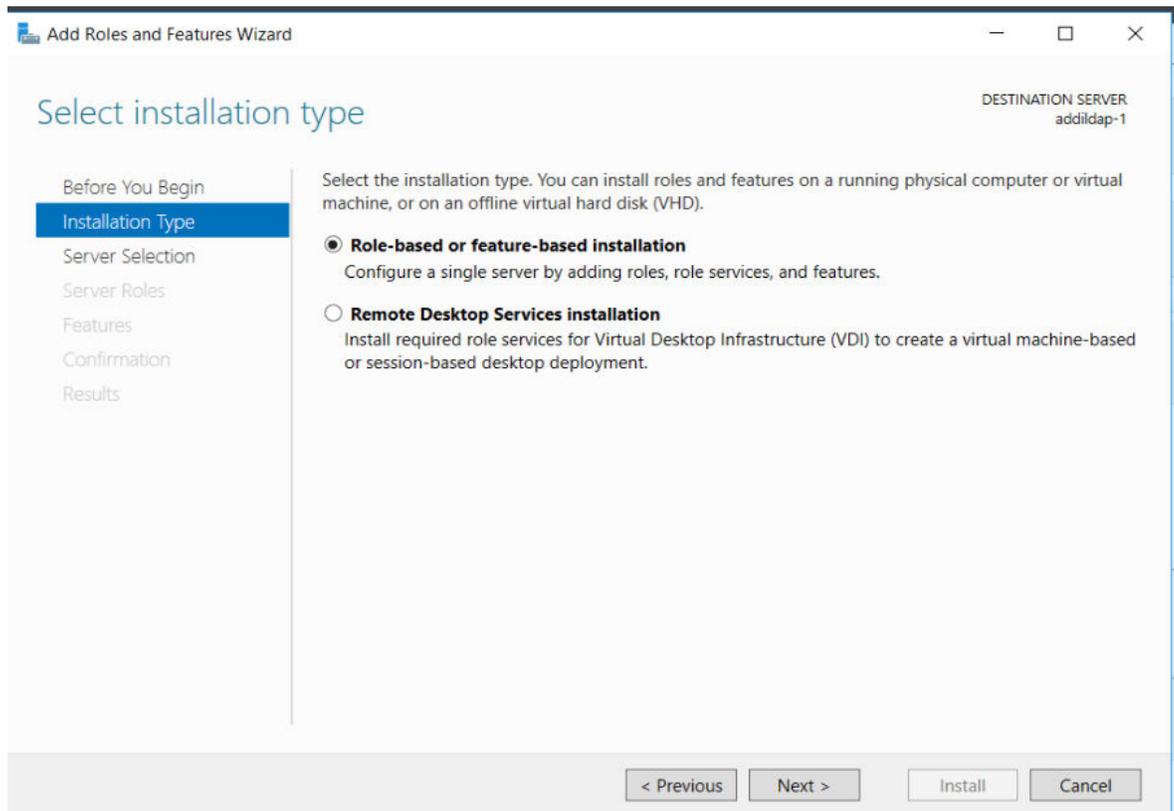
Installing and configuring AD DS

Complete the following steps to set up LDAP through AD DS.

1. Log in as the administrator on the Windows Server machine.
2. Click **Start > Server Manager**.
3. Select **Dashboard** on the left menu and select **2) Add roles and features**.

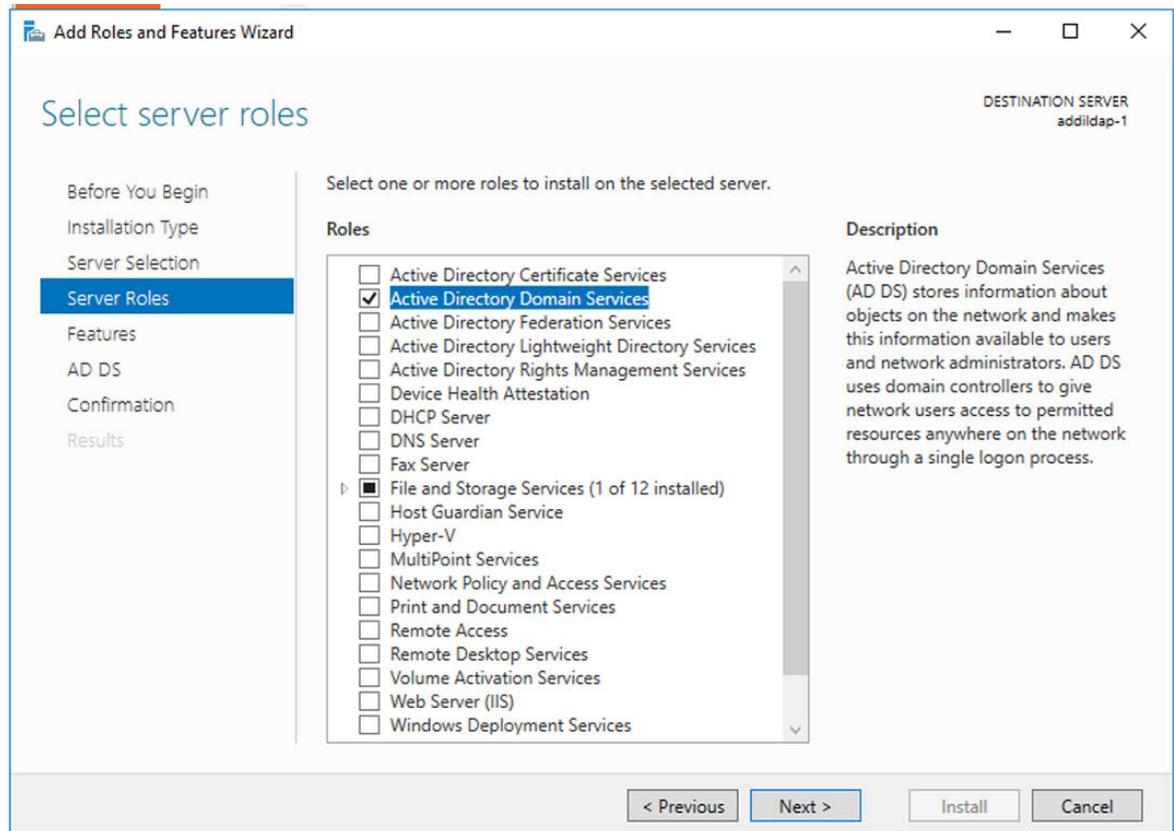


4. Perform the feature installation with the **Add Roles and Features Wizard**.
 - a. Click **Next** on the **Before You Begin** tab.
 - b. Select **Role-based or feature-based installation** on the **Installation Type** tab and click **Next**.

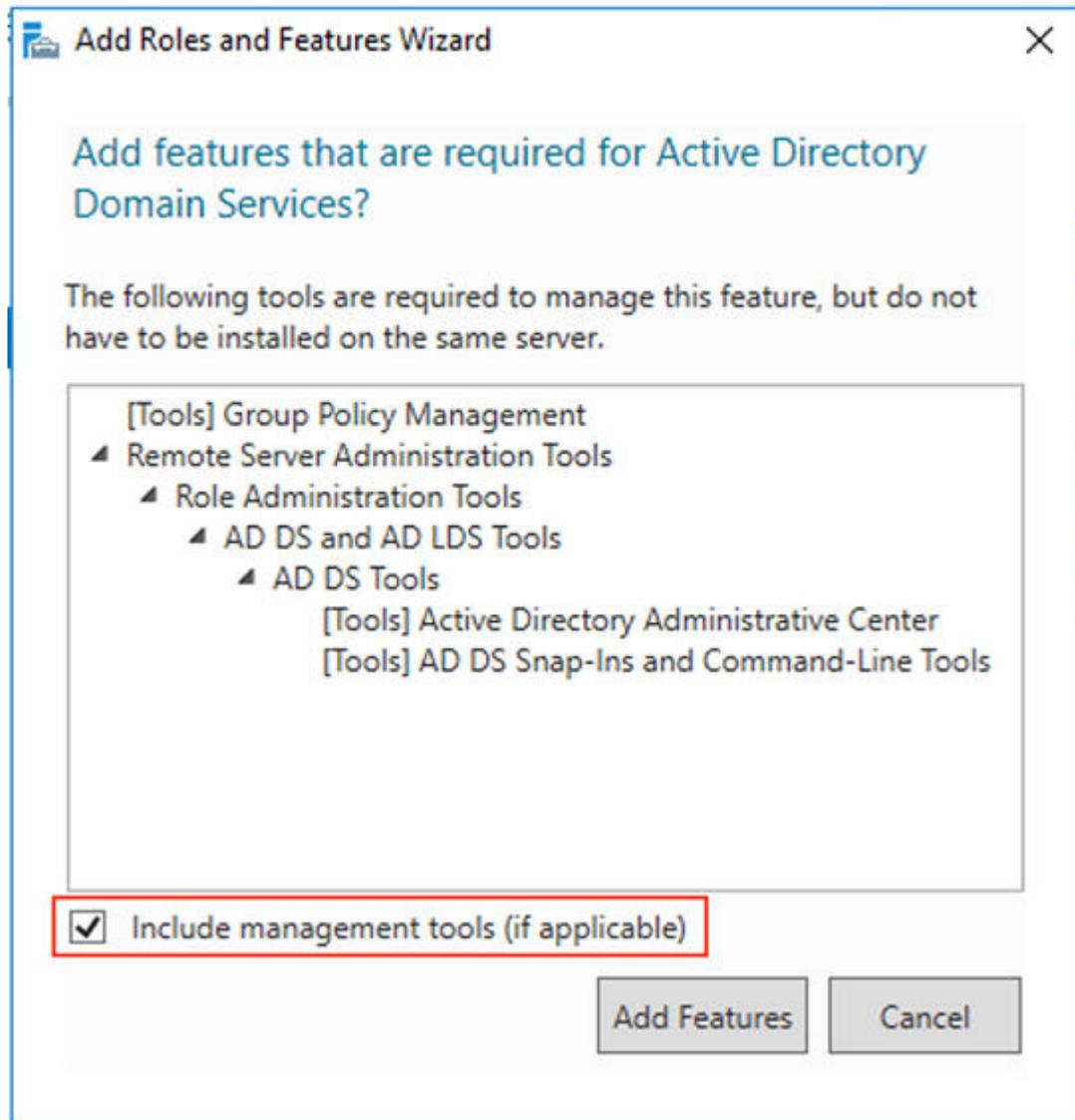


c. Click **Next** on the **Server Selection** tab to keep the default server information.

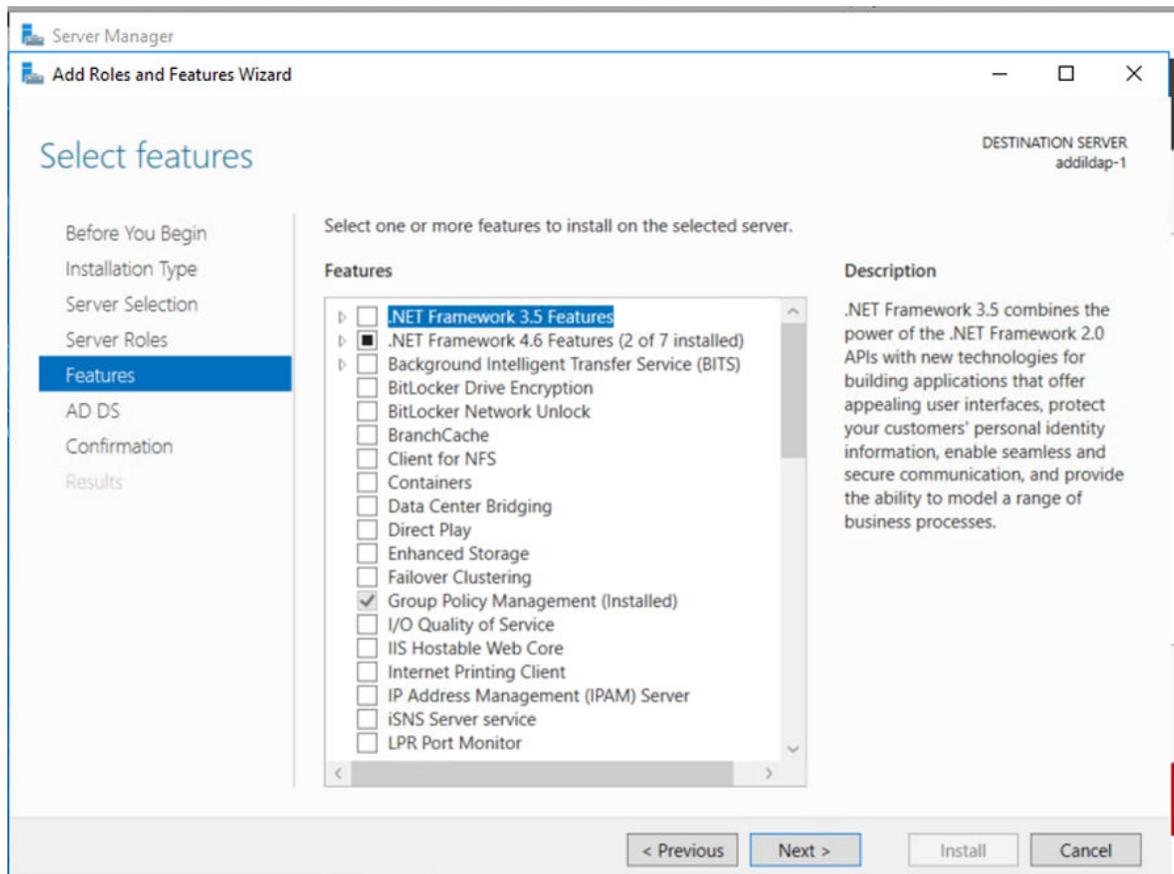
d. Select **Active Directory Domain Services** from the Roles checklist on the **Server Roles** tab.



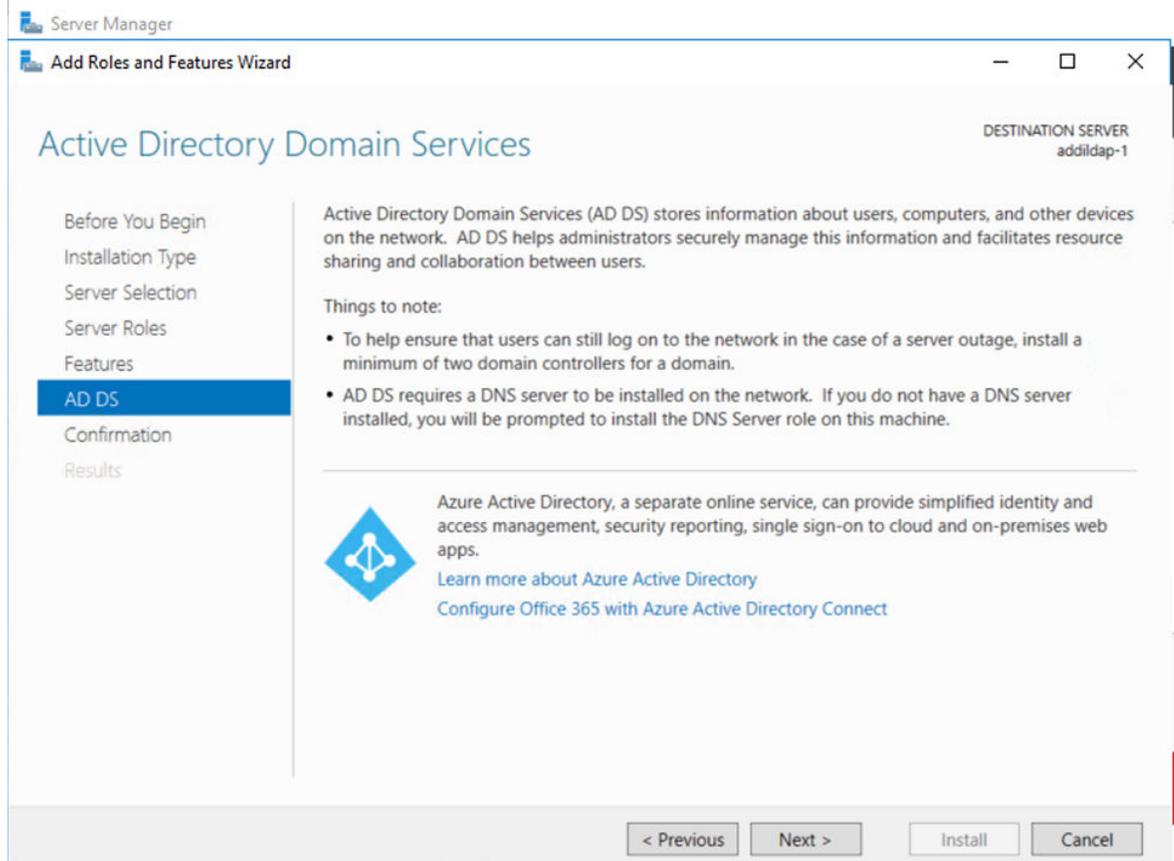
e. Select the **Include management tools (if applicable)** checkbox and click **Add Features** on the pop-up dialog box.



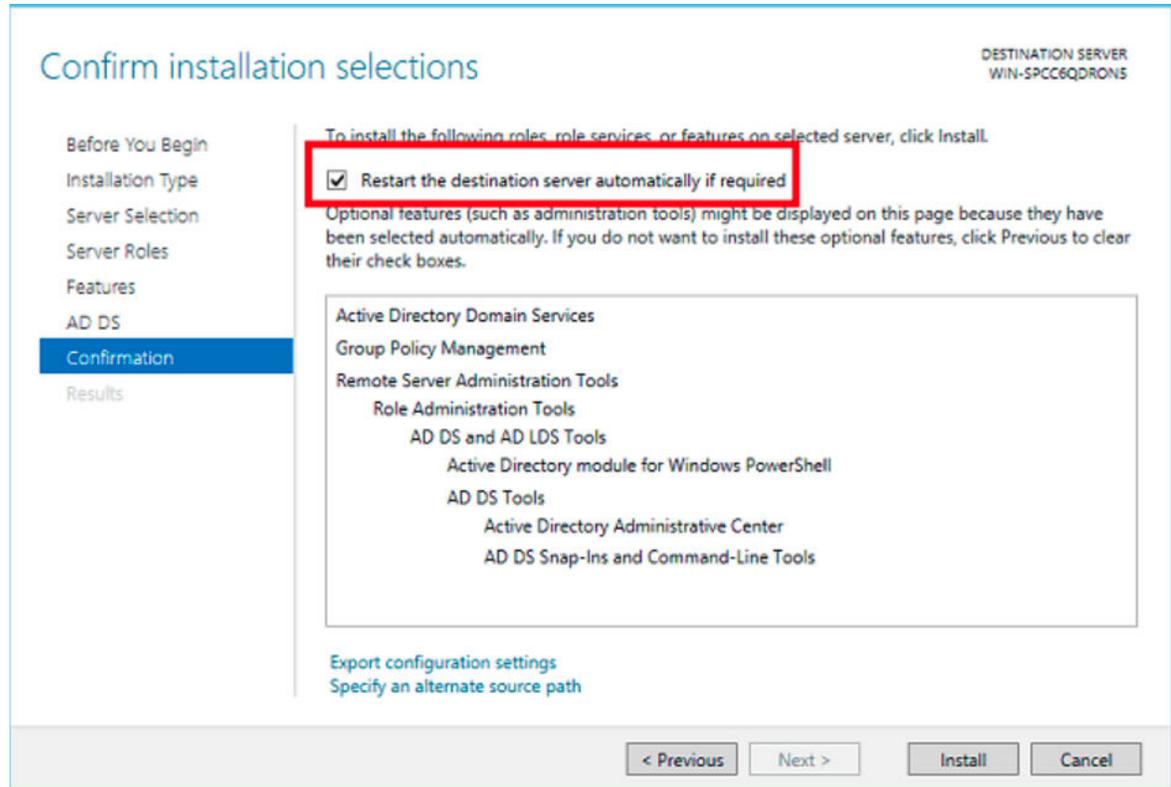
- f. Click **Next** on the **Server Roles** tab. Make sure that **Active Directory Domain Services** checkbox is selected.
- g. Click **Next** on the **Features** tab.



h. Click **Next** on the **AD DS** tab.

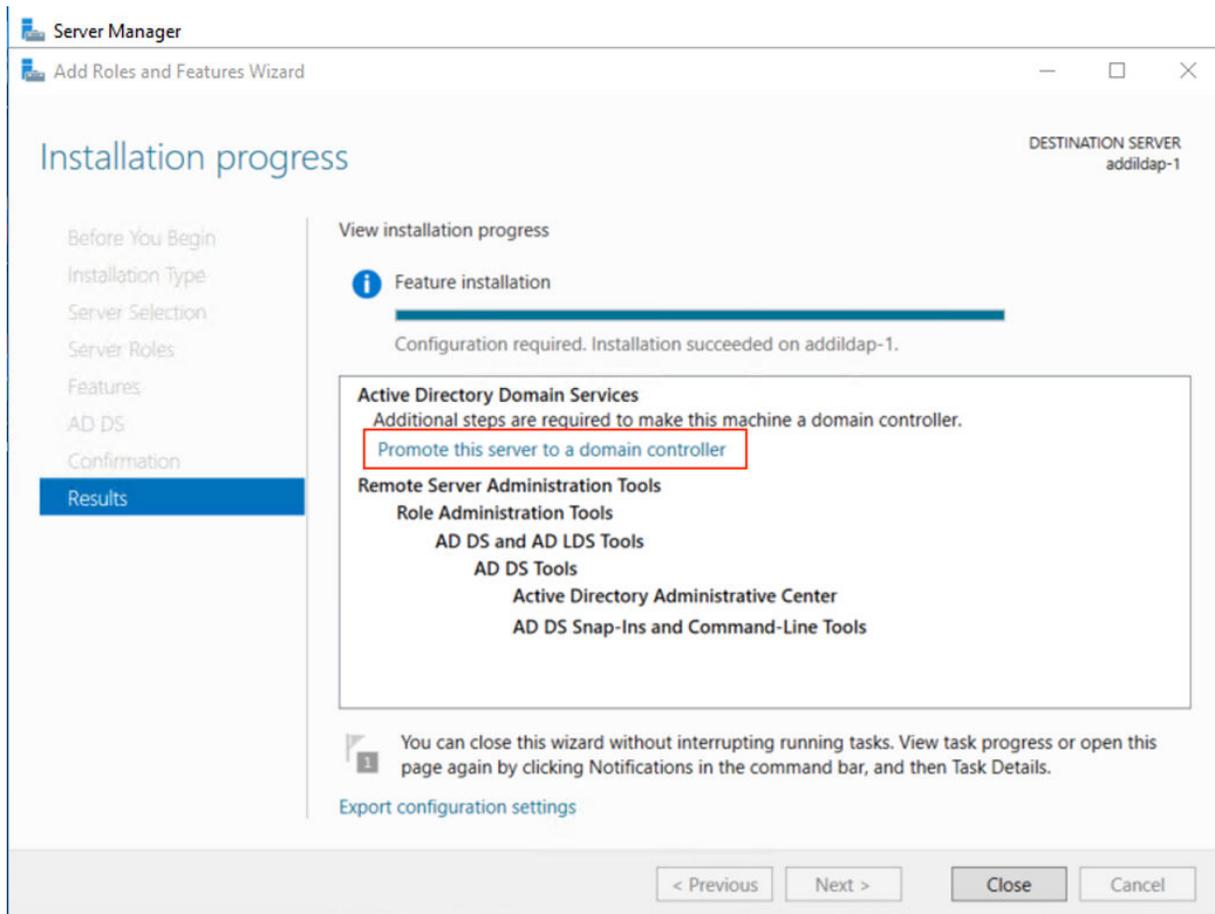


- i. Select the **Restart the destination server automatically if required** checkbox and click **Install** on the **Confirmation** tab.



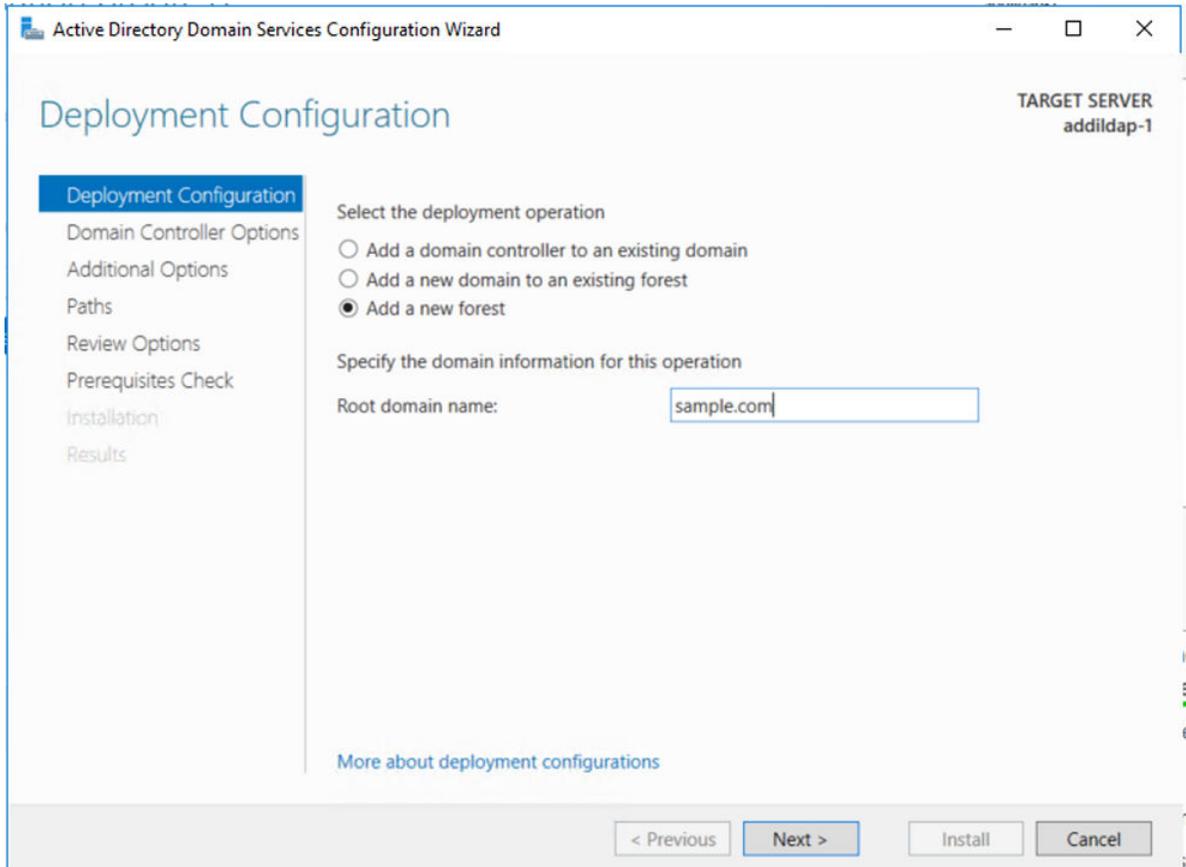
- j. Wait until the installation is done.

5. Click the **Promote this server to a domain controller** link on the **Results** tab.



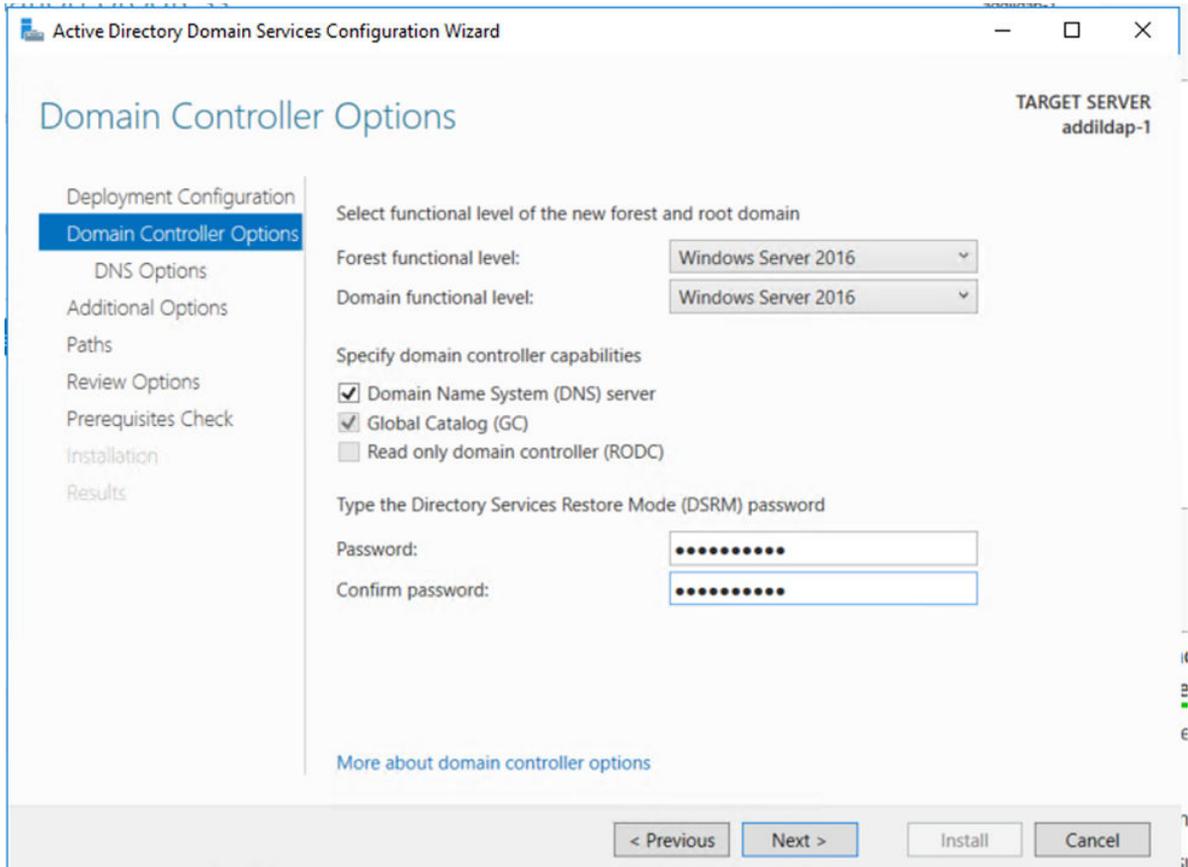
6. Complete the following steps to configure AD DS.

- a. Click **Add a new forest** and enter `sample.com` in the **Root domain name** field. Then, click **Next** on the **Deployment Configuration** tab.

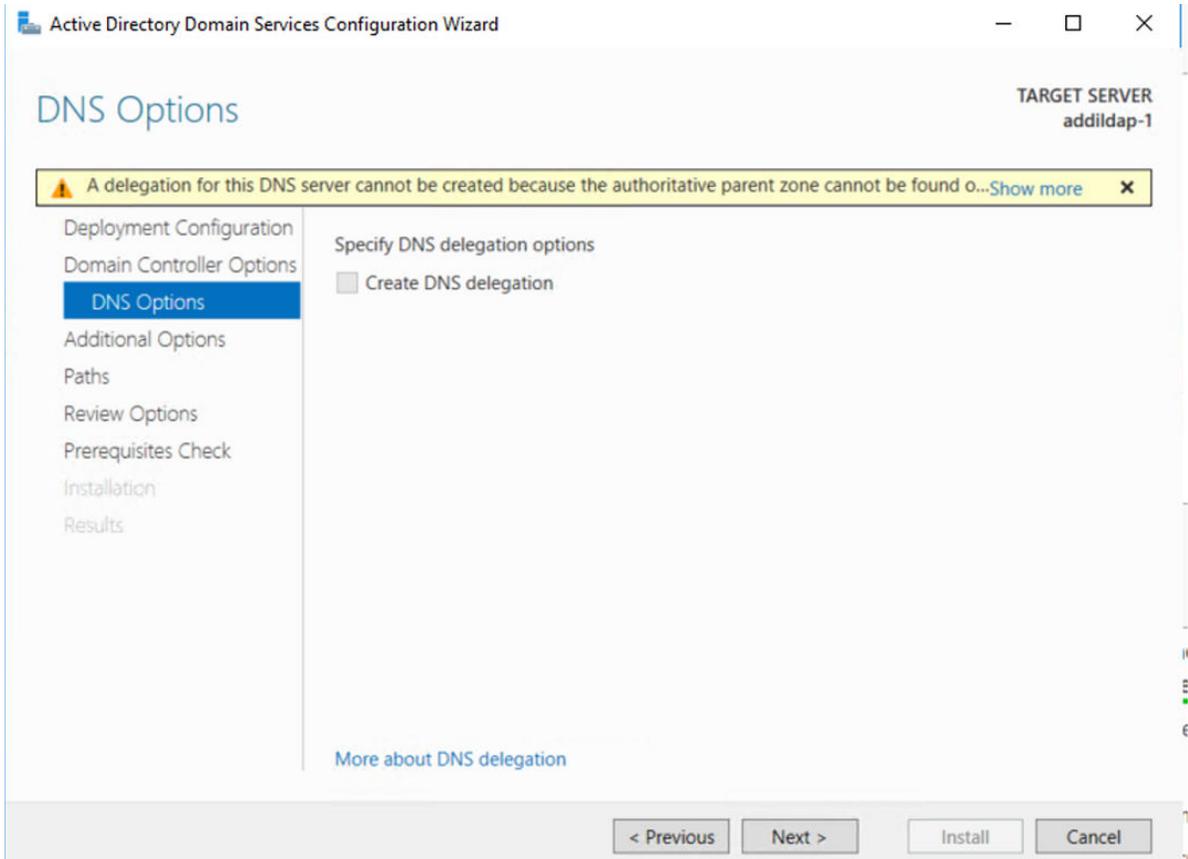


- b. Enter the password for directory service restore in the **Password** and **Confirm password** fields. Then, click **Next**.

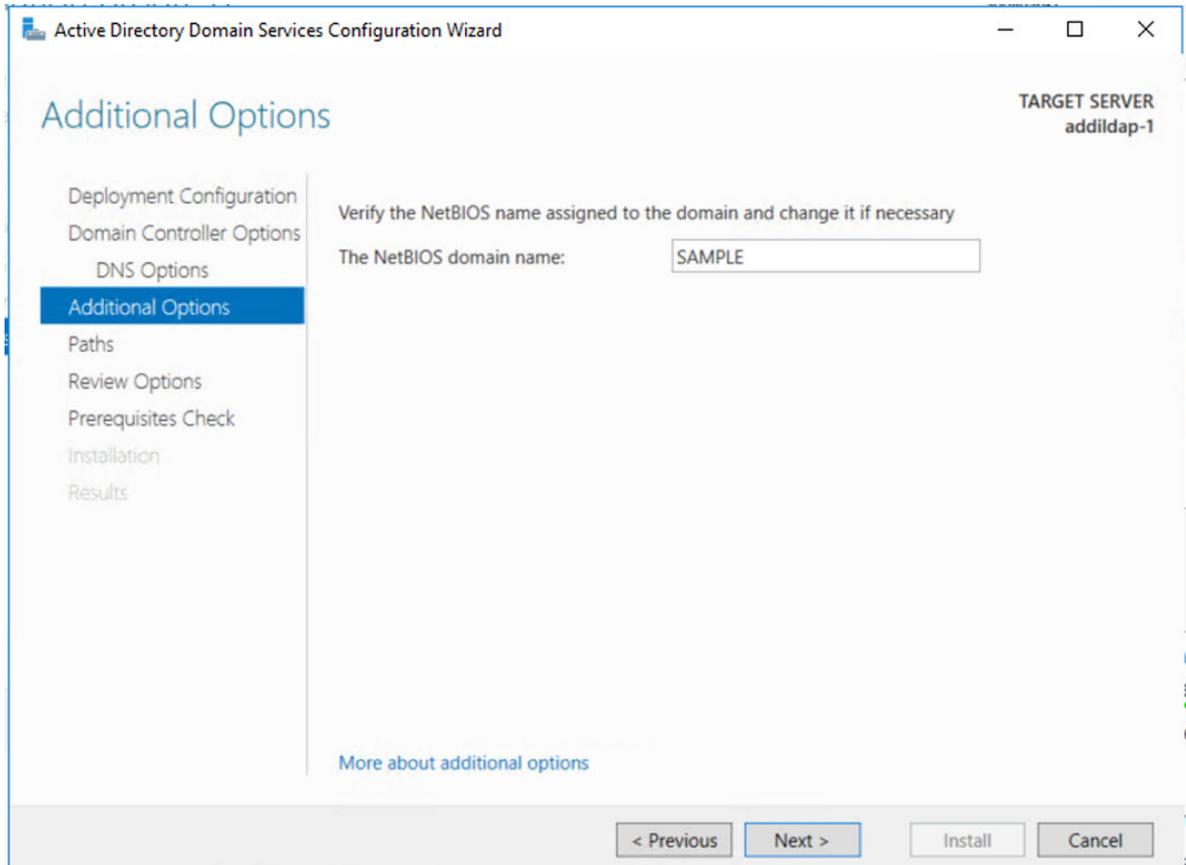
Note: Remember this password in case that you might need to restore your directory service.



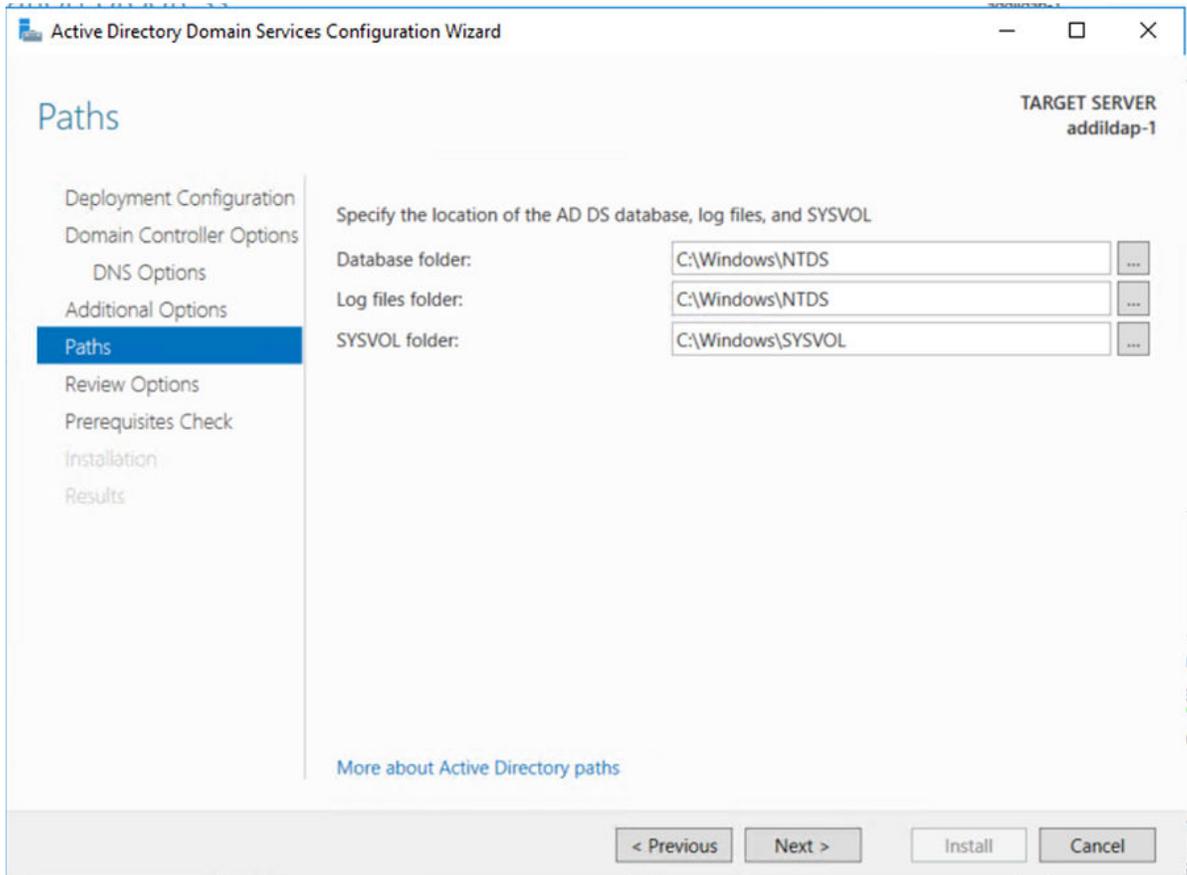
c. Click **Next** on the **DNS Options** tab.



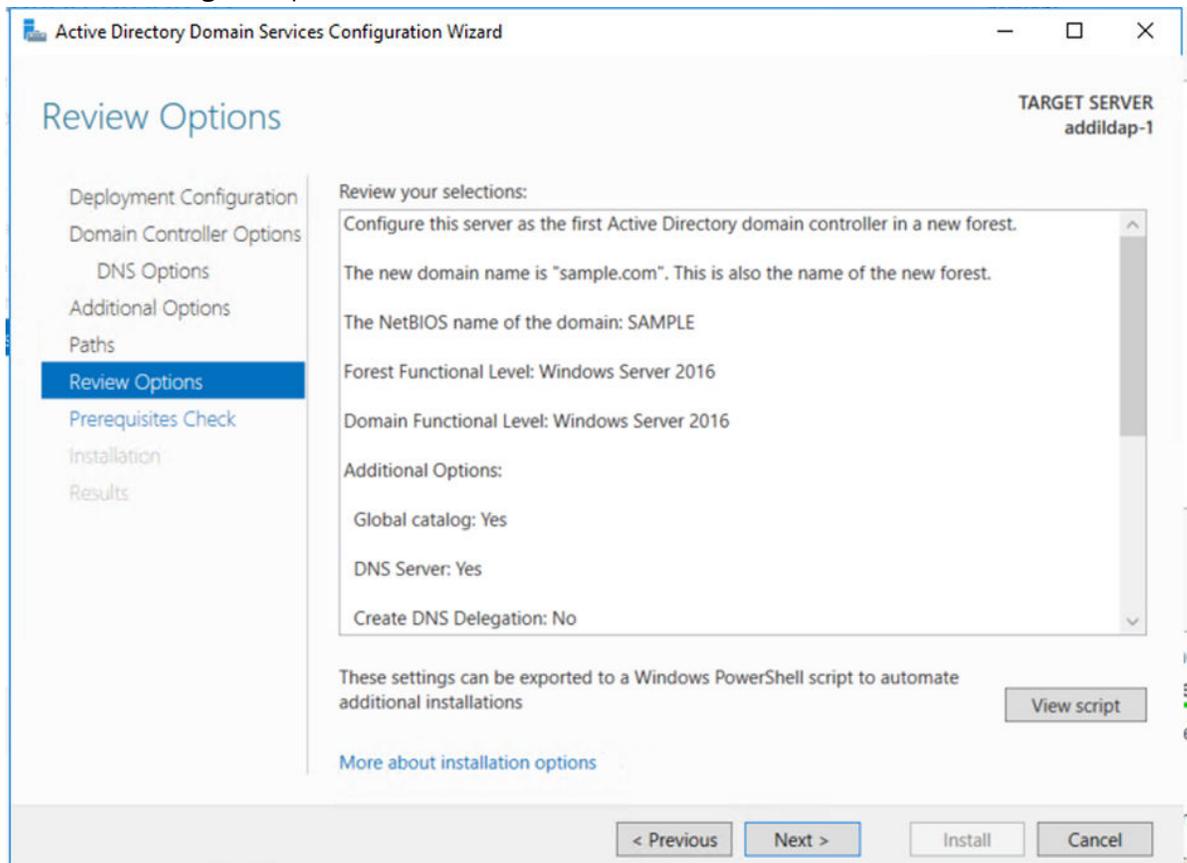
- d. Click **Next** on the **Additional Options** tab. The **NetBIOS domain name** field is automatically completed for you at this step.



- e. Click **Next** on the **Path** tab to keep the location of AD DS database, log files, and SYSVOL.



f. Review the configured options and click **Next** to confirm.



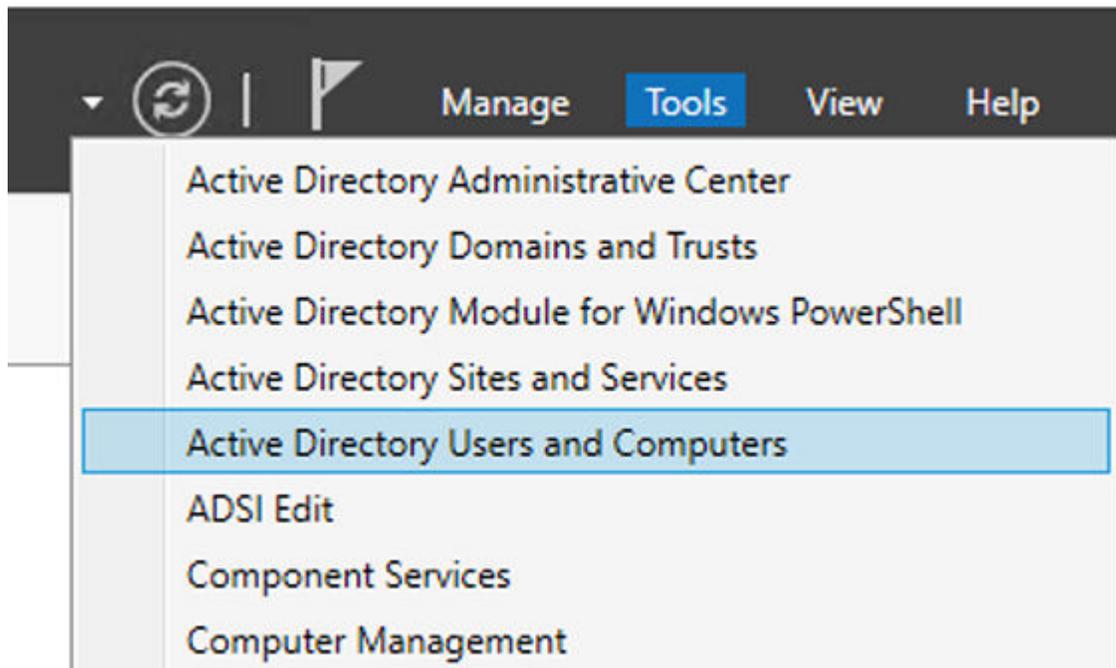
- g. Click **Install** on the **Prerequisites Check** page when all prerequisites checks are passed.
- h. Wait until the installation is done and the results page is displayed. Then, your server will be automatically restarted to finish the configuration.

After the server is restarted, you can log back into the server. You have now successfully installed and configured AD DS. Next you will set up active directory user groups and users for ADDI Extensions. These users and user groups will be used to access ADDI Extensions and show you how to setup user permissions.

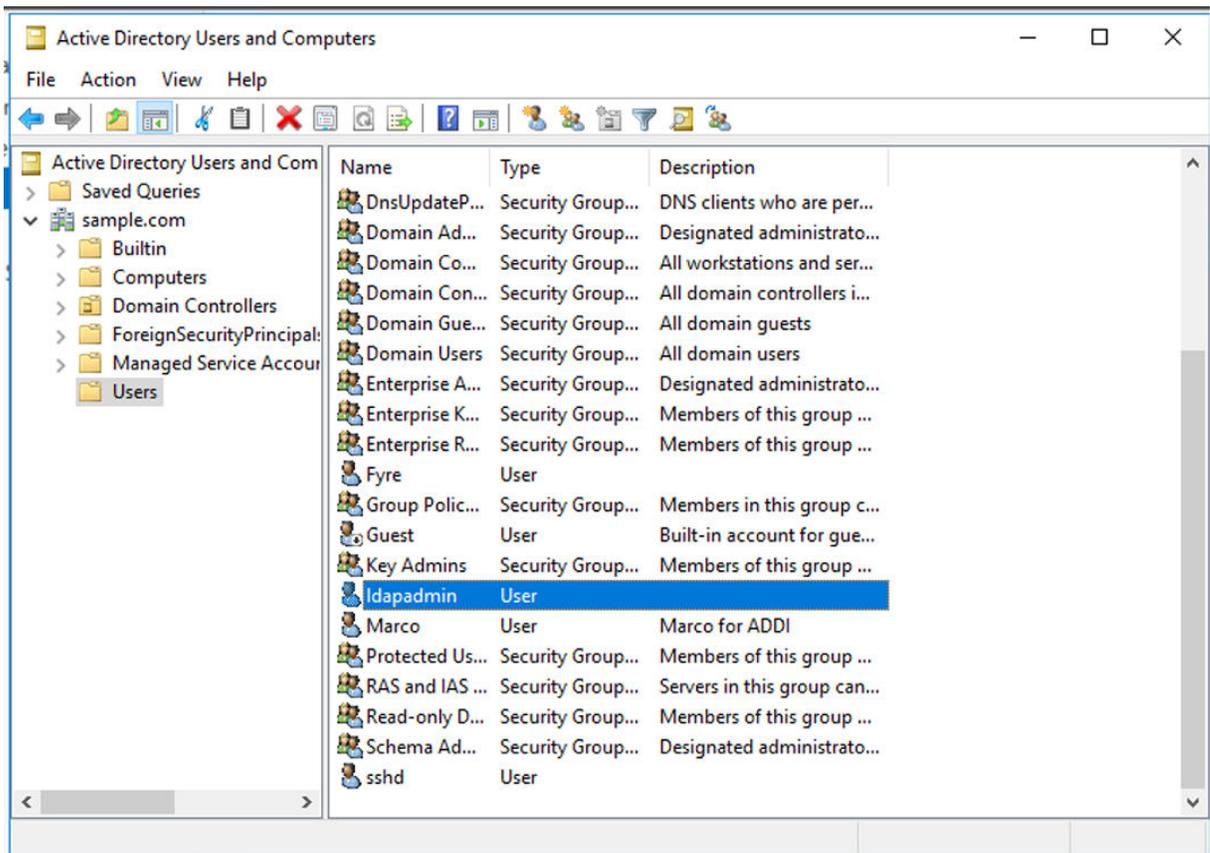
Setting up active directory user groups and users

Complete the following steps to setup active directory user groups and users.

1. In the Server Manager, select **Tools > Active Directory Users and Computers** from the upper right menu.



2. Select the name of the user who has Administrators role and will initialize the connection between active directory and ADDI Extensions. In this tutorial, **ldapadmin** user is the user for example. For the rest of this tutorial, this user will be used as the administrator for ADDI Extensions.



3. Double-click the user item to open the ldapadmin Properties window.
4. Enter the email address that you want to use for initialization in the **E-mail** field. It is ldapadmin@sample.com in this example.

The screenshot shows the 'Idapadmin Properties' dialog box with the 'General' tab selected. The user name 'ldapadmin' is visible in the top left. The 'E-mail' field contains 'ldapadmin@sample.com'. The 'OK' button is highlighted with a blue border.

Member Of	Dial-in	Environment	Sessions
Remote control	Remote Desktop Services Profile		COM+
General	Address	Account	Profile
	Telephones	Organization	

ldapadmin

First name: Initials:

Last name:

Display name:

Description:

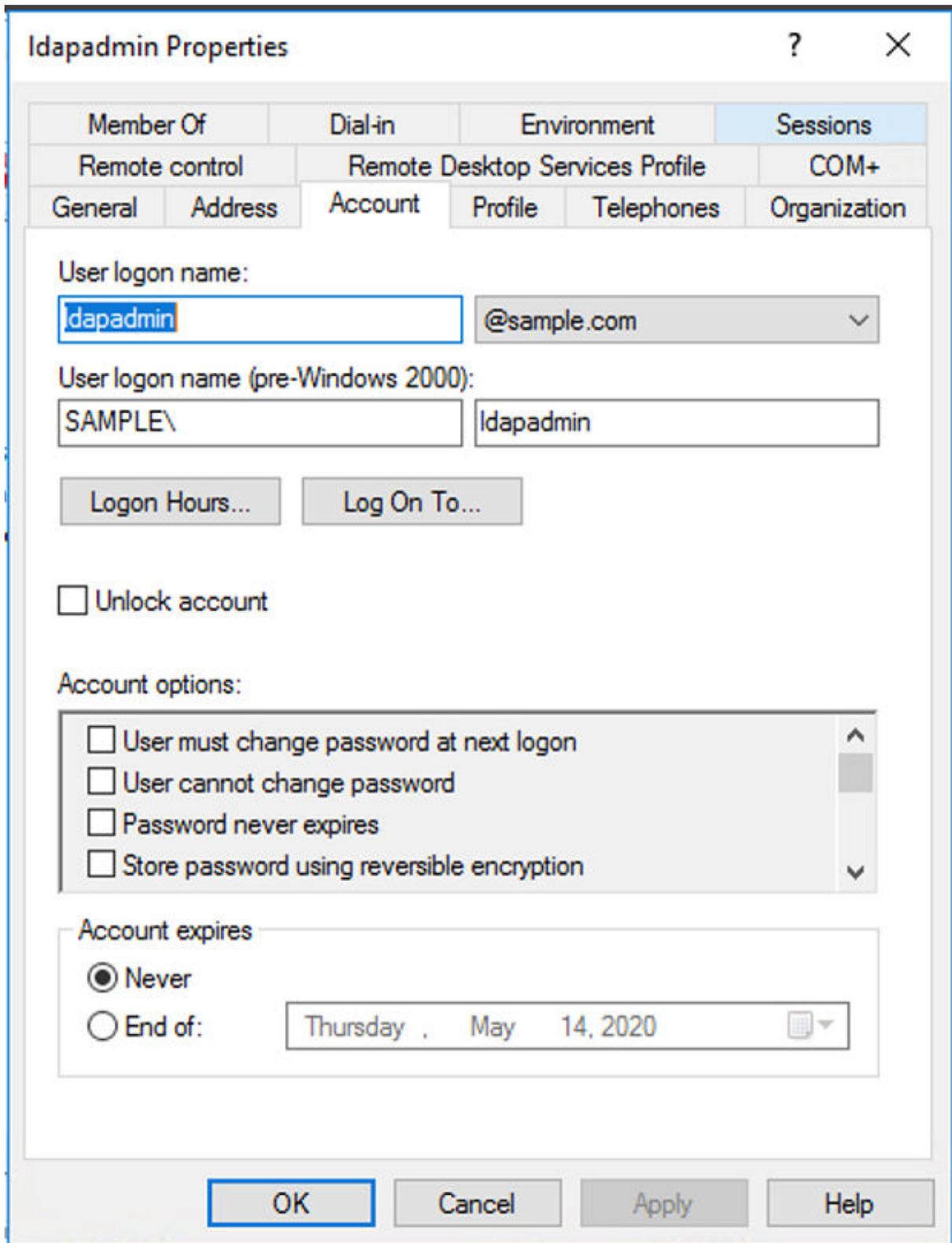
Office:

Telephone number:

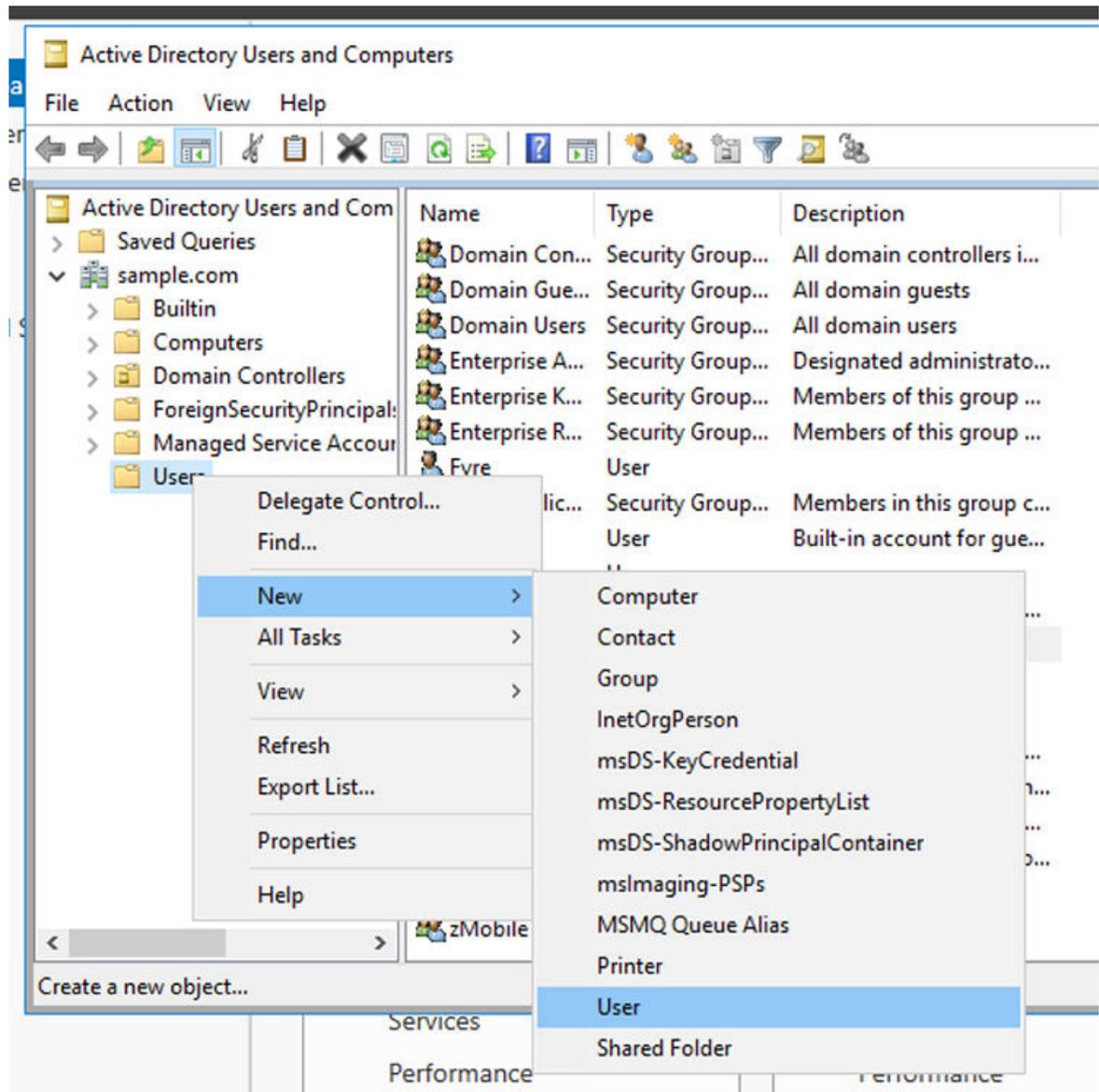
E-mail:

Web page:

5. Select the **Account** tab and enter the same e-mail account name as the user logon name. The email account name will be used as the username when you log in to ADDI Extensions. It is ldapadmin in this example.



6. Clear all the checkboxes in the Account options list and click **OK**.
7. Create the active directory users to access ADDI Extensions.
 - a. In the Active Directory Users and Computers window, right-click the **Users** folder.
 - b. Select **New > User**.



- c. Enter the following information in the New Object – User dialog box.
- i) First name: Marco
 - ii) User logon name: marco

Active Directory Users and Computers

New Object - User

Create in: sample.com/Users

First name: Initials:

Last name:

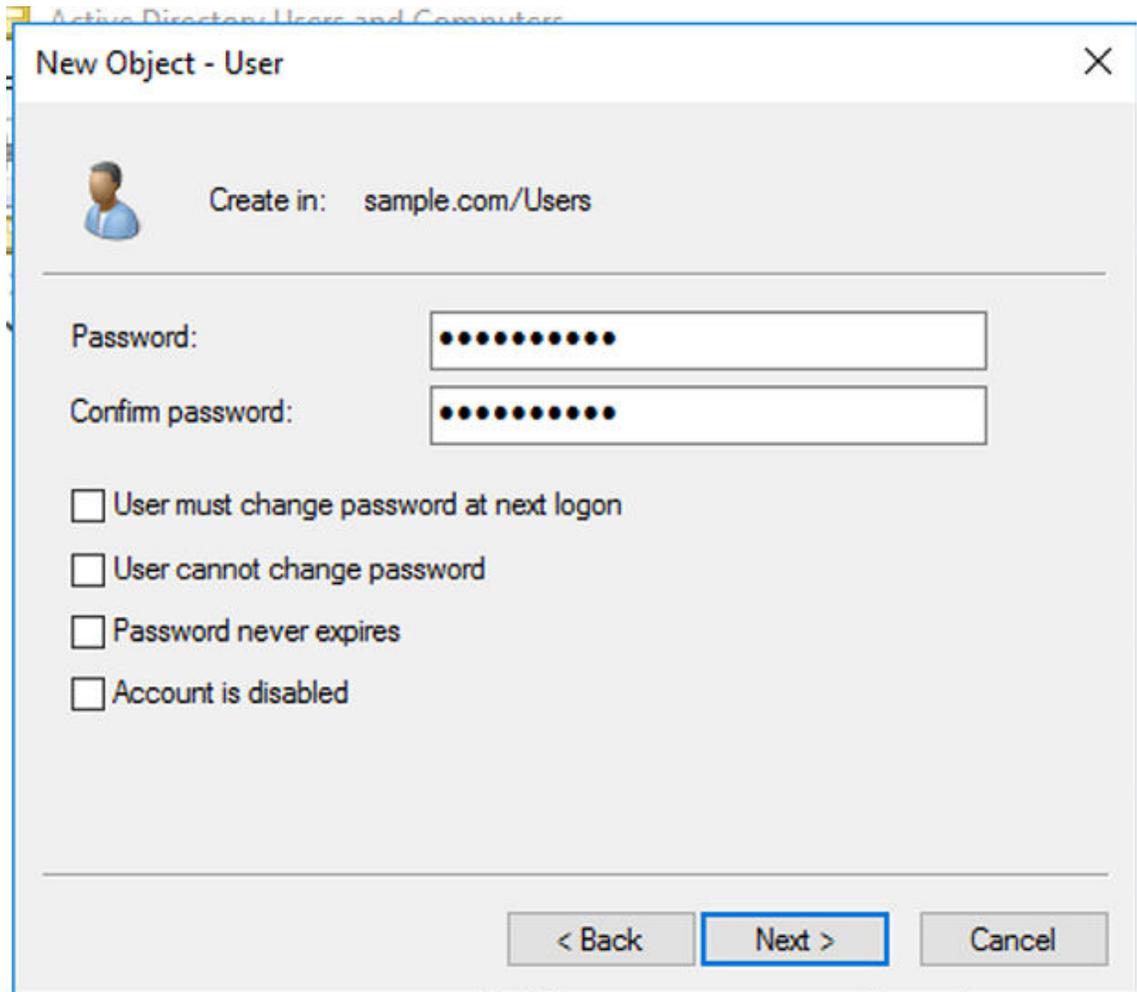
Full name:

User logon name:

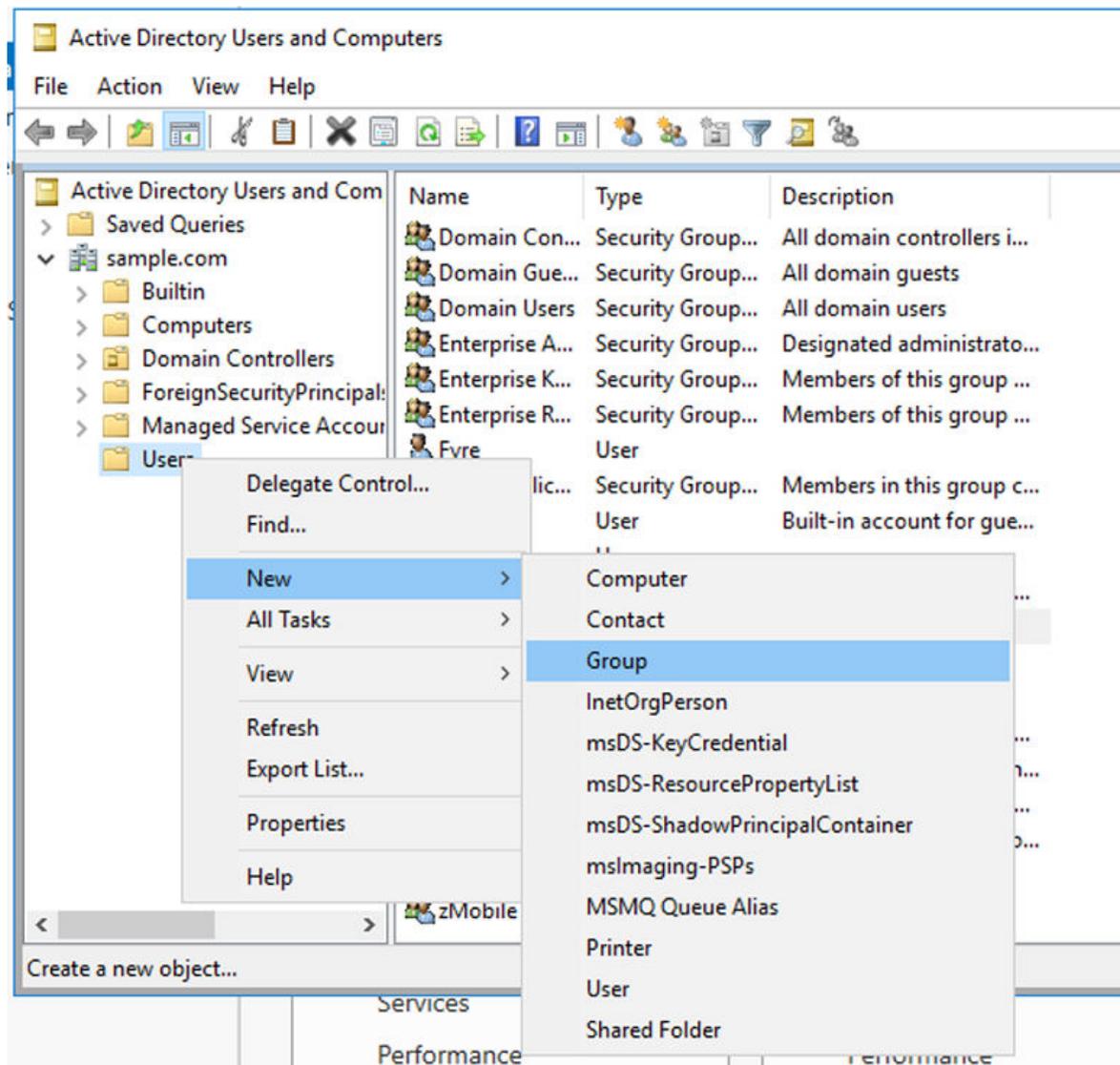
User logon name (pre-Windows 2000):

< Back **Next >** Cancel

- iii) Click **Next**.
- iv) Enter the password to be used for this user in the **Password** and **Confirm Password** fields. Note that you can use the user name as password to make it easy to remember.

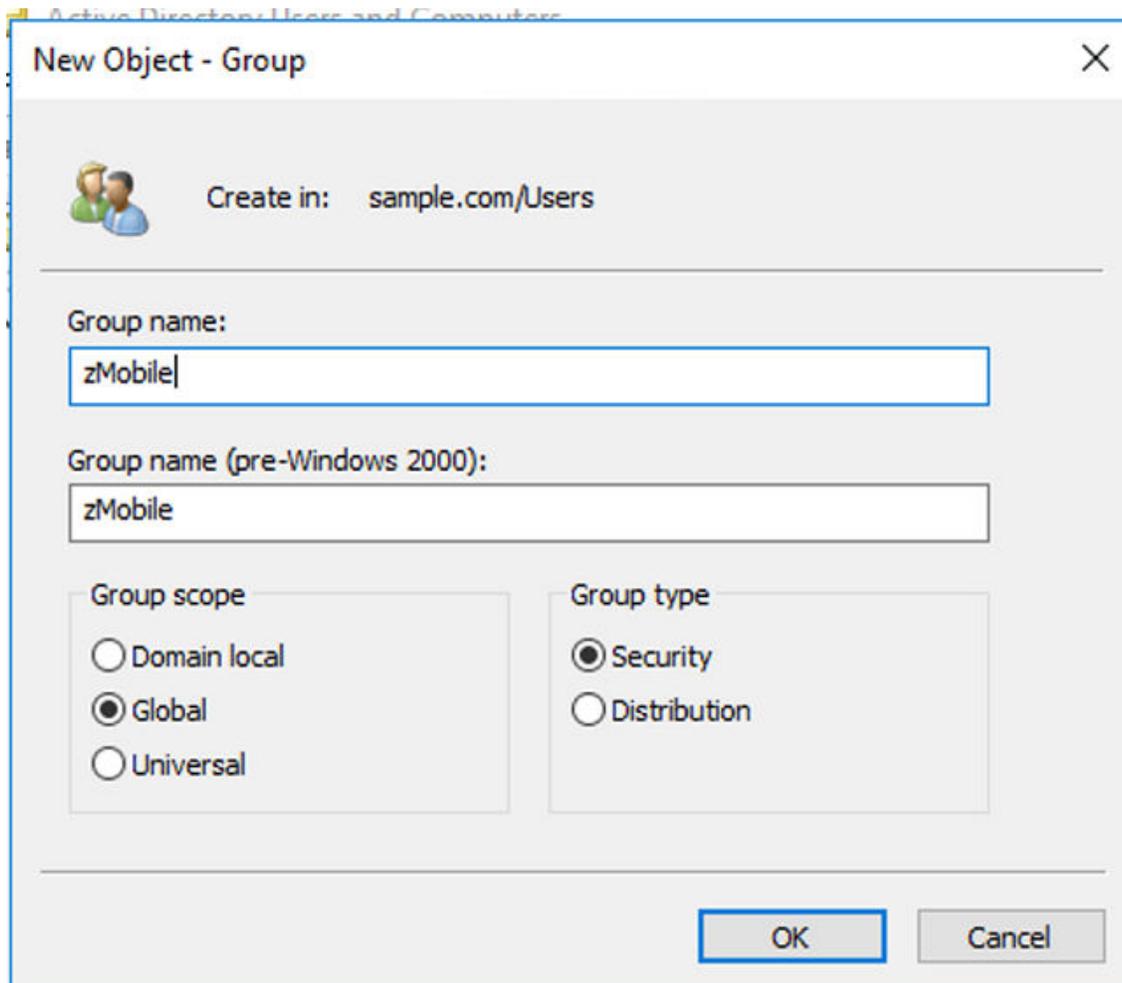


- v) Clear the **User must change password at next logon** checkbox.
 - vi) Click **Next**.
 - vii) Click **Finish** to confirm the user creation.
 - d. Repeat step a through c to create **Jane** and **Tammy** as users.
8. Create user groups to manage workbook permissions in ADDI Extensions.
- a. In the Active Directory Users and Computers window, right-click the **Users** folder.
 - b. Select **New > Group**.

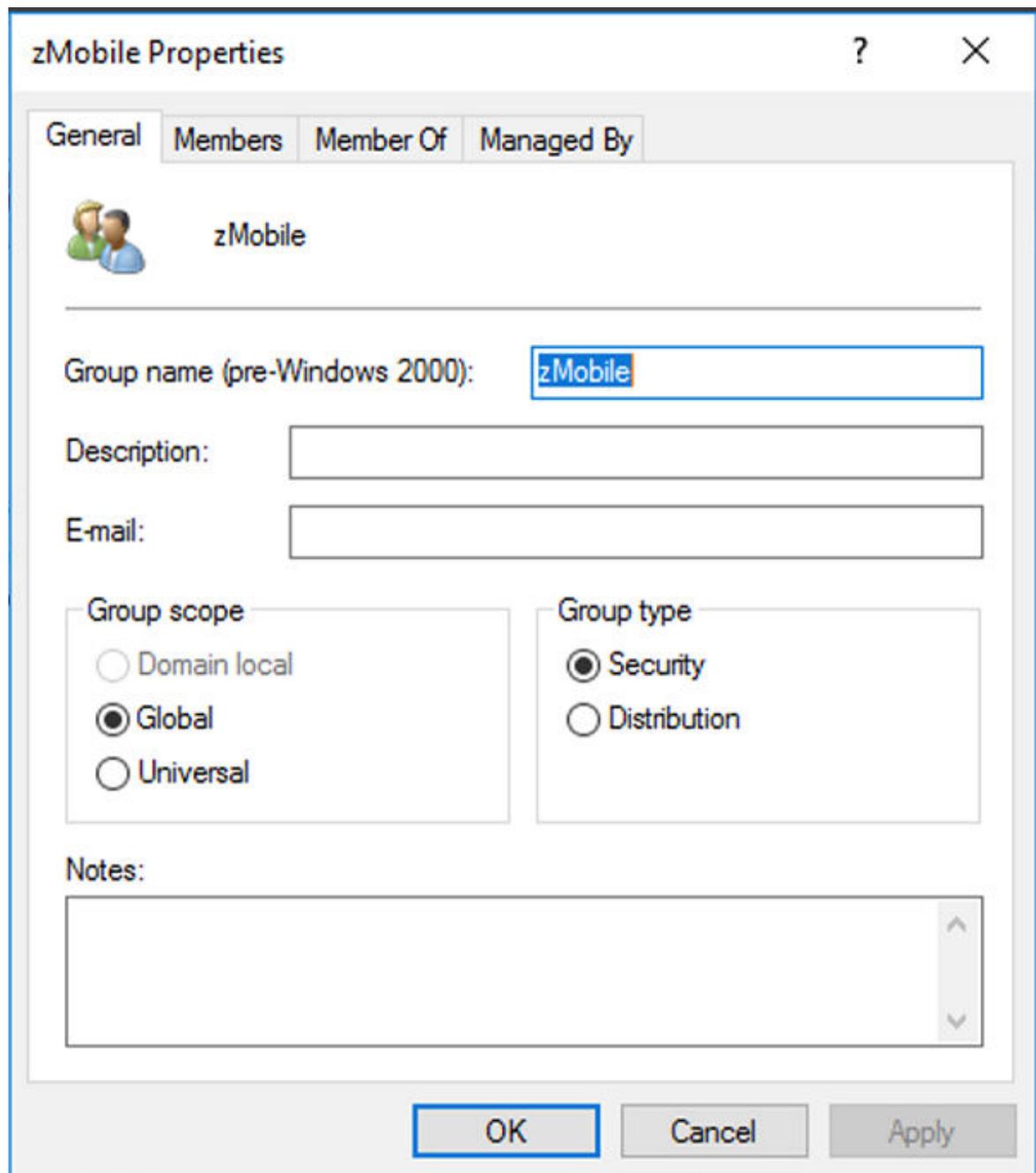


c. Specify the following information in the New Object – Group dialog box.

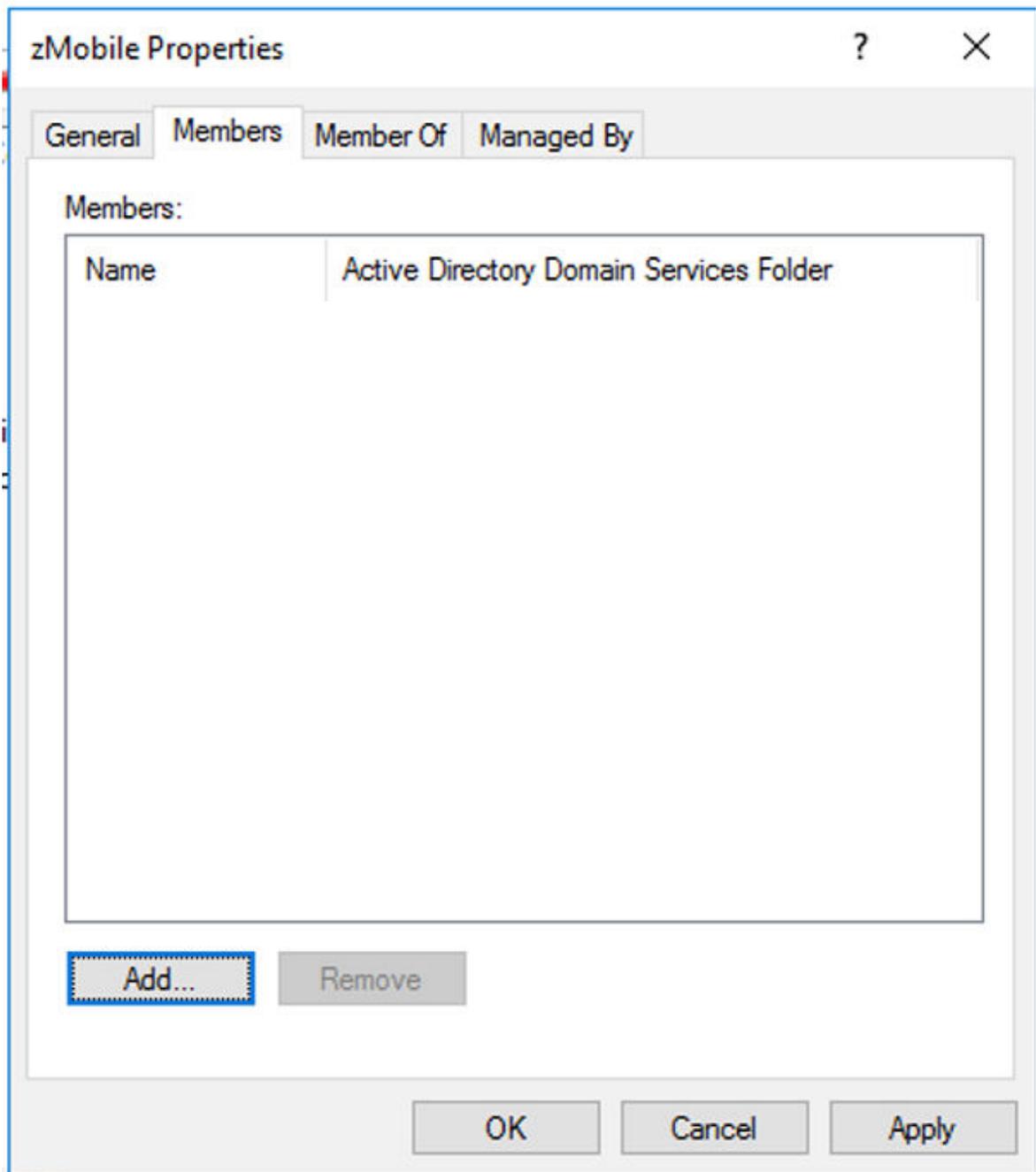
- i) Group name: zMobile
- ii) Group scope: Ensure that the **Global** is selected.
- iii) Click **OK** to create a group.



- d. Repeat step a through c to create **Addi Administrators** and **HRM App** groups.
9. Assign users to the **zMobile** group.
 - a. Double-click the **zMobile** group to open the zMobile Properties window.

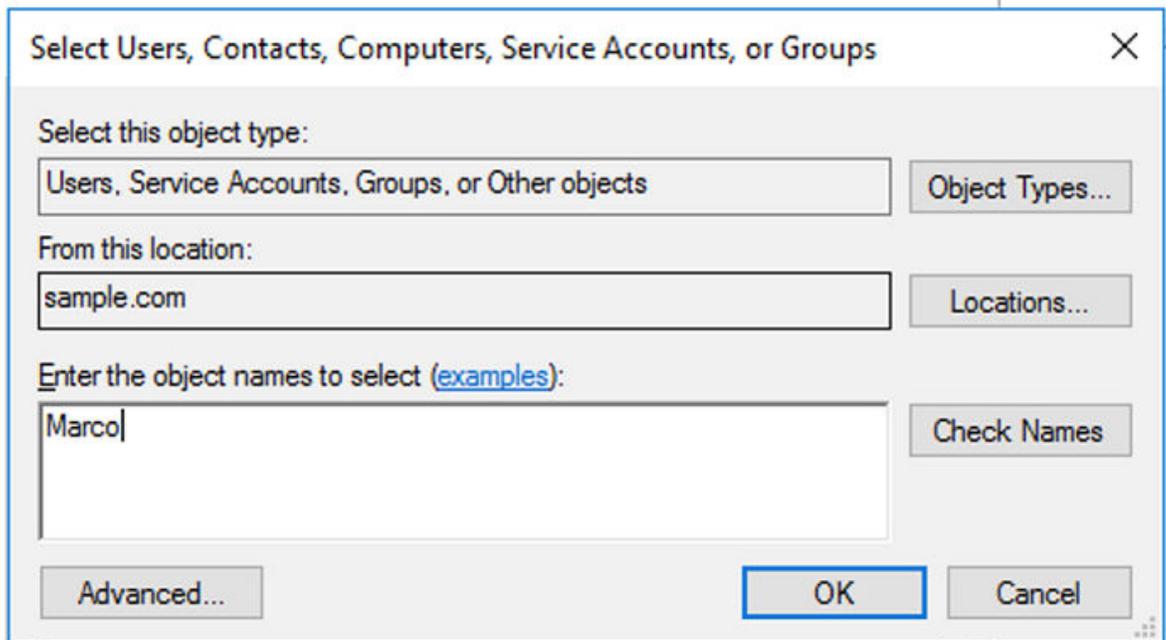


b. Select the **Members** tab.

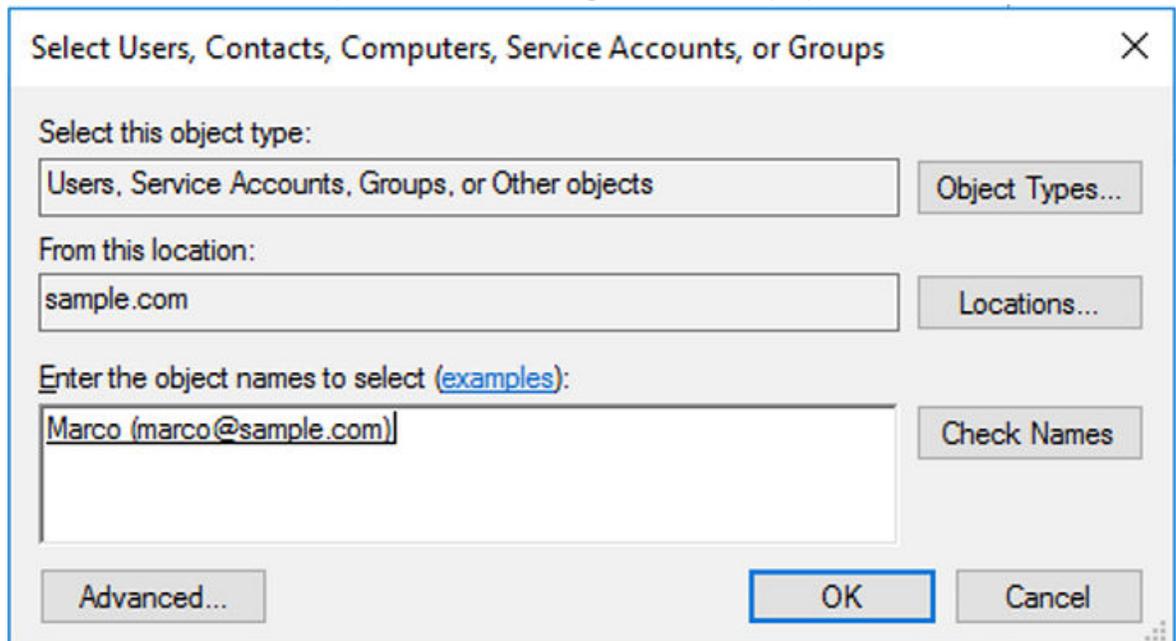


c. Click the **Add** button.

d. Type Marco in the **Enter the object name to select** field.

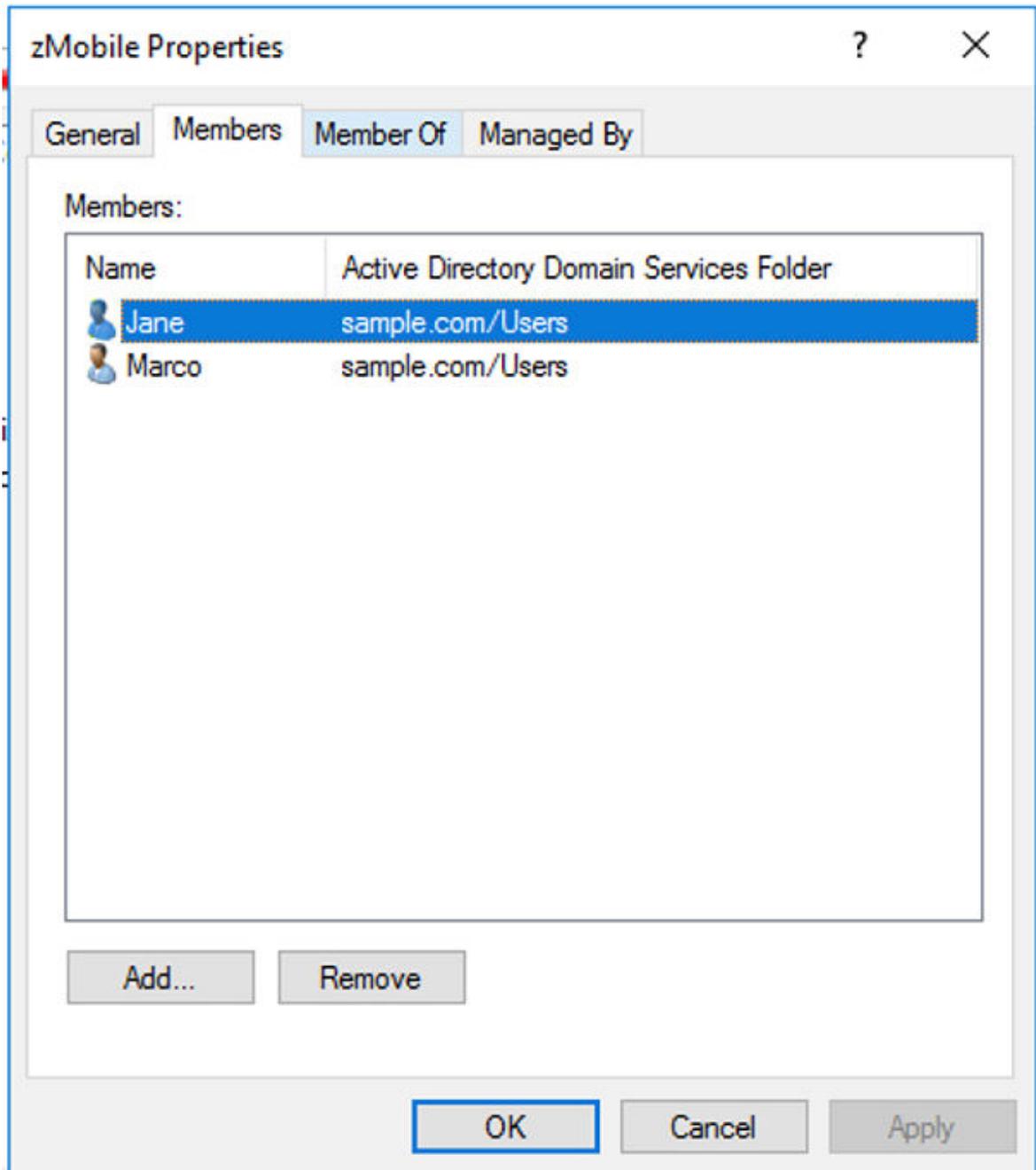


e. Click **Check Names**. The object name with the logon name is displayed.

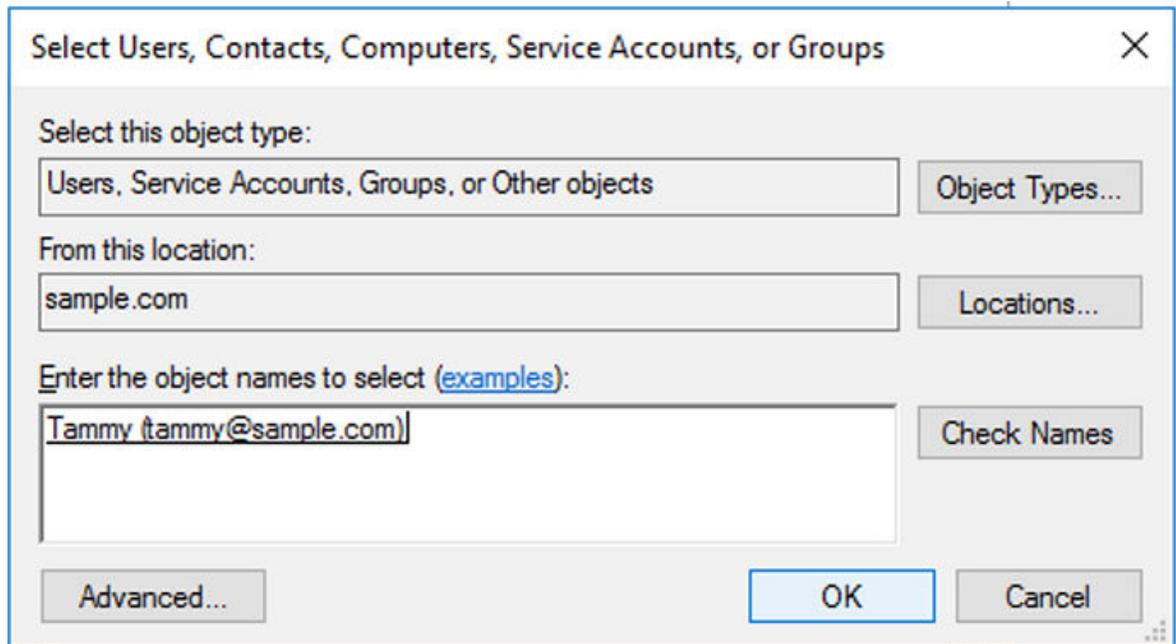


f. Click **OK** to finish.

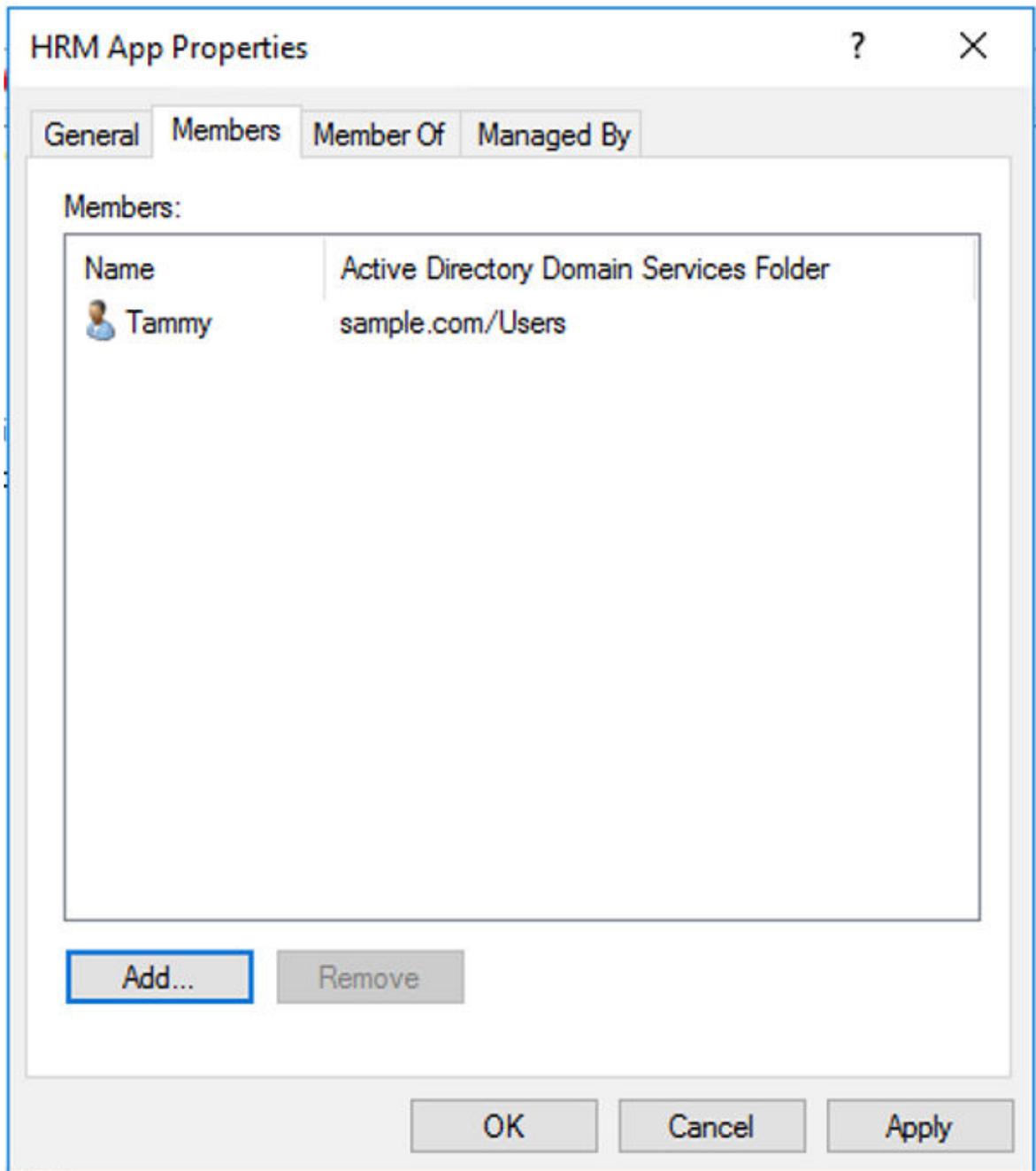
g. Repeat step c through f to add **Jane** to the **zMobile** group.



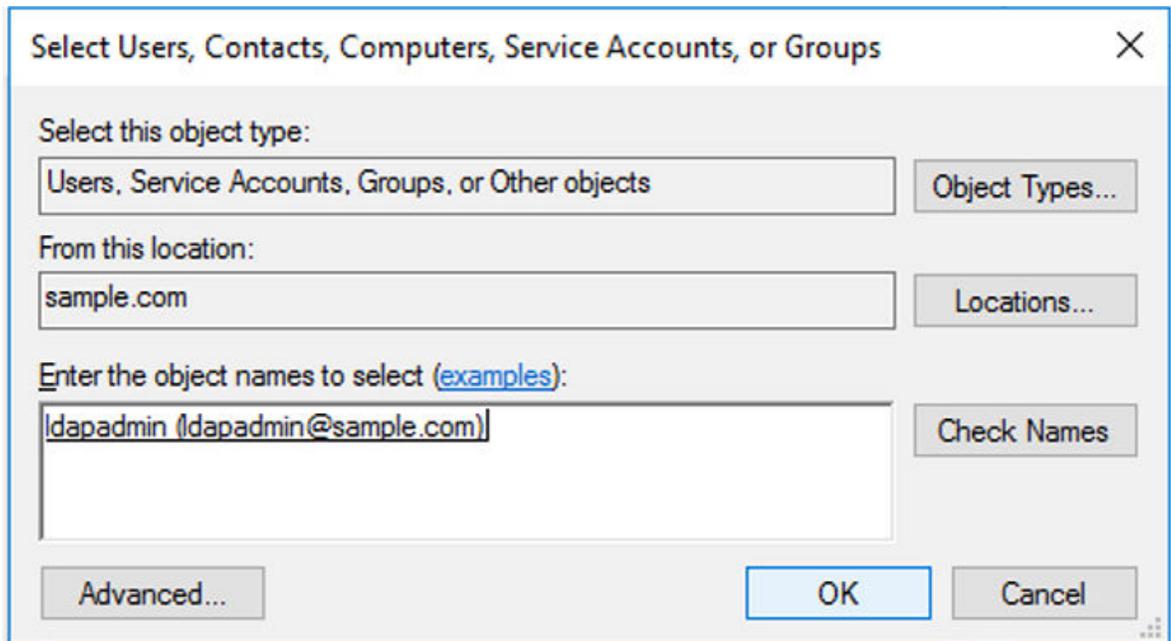
- h. Click **OK** to close the zMobile Properties window.
- 10. Assign **Tammy** to the **HRM App** group.
 - a. Double-click **HRM App** group to open the HRM App Properties window.
 - b. Select the **Members** tab.
 - c. Click the **Add** button.
 - d. Type tammy in the **Enter the object name to select** field.
 - e. Click **Check Names**. The object name with the logon name is displayed.



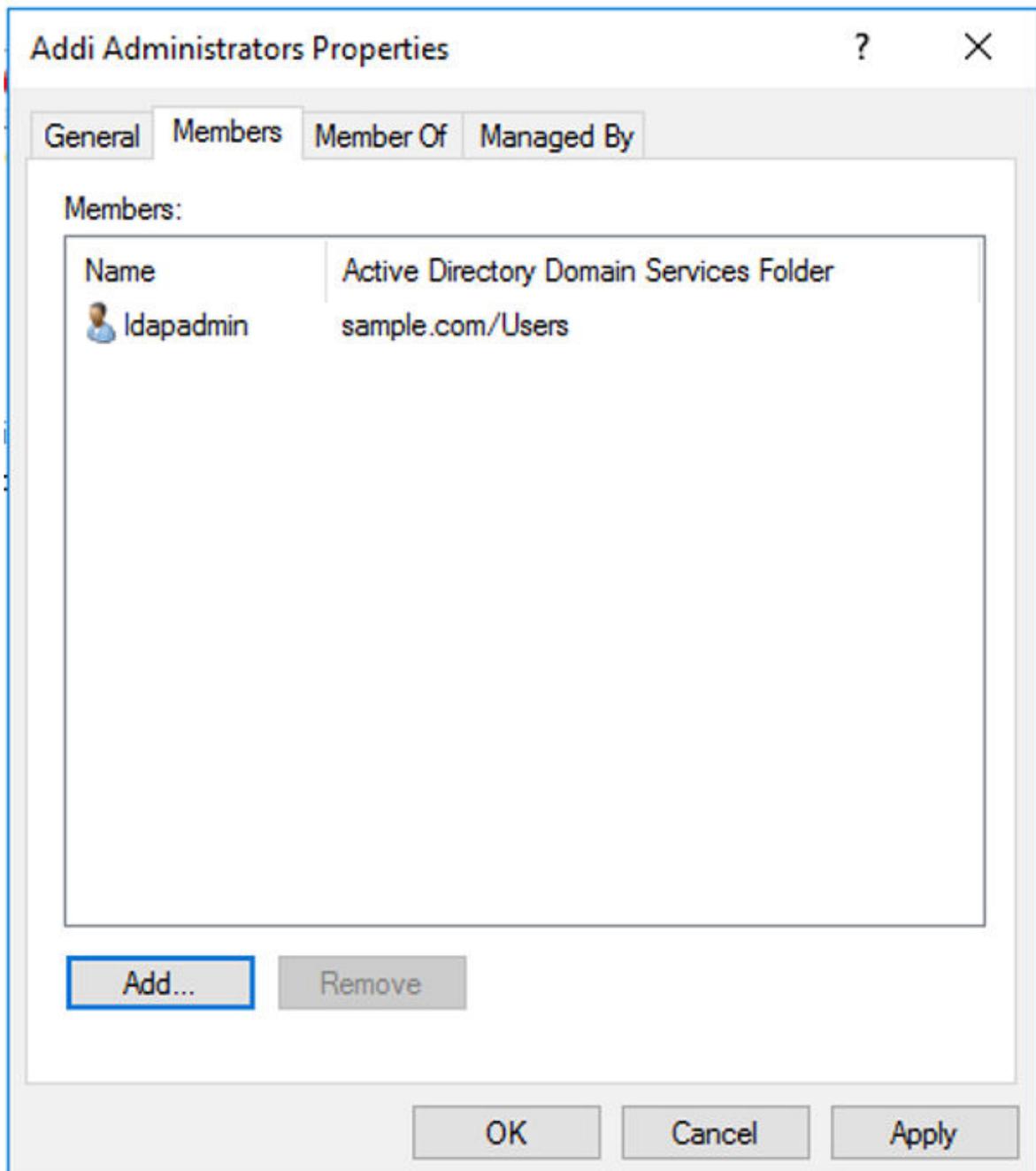
f. Click **OK** to finish.



- g. Click **OK** to close the HRM App Properties window.
11. Assign administrator user to the **Addi Administrators** group.
 - a. Double-click the **Addi Administrators** group to open the Addi Administrators Properties window.
 - b. Select the **Members** tab.
 - c. Click the **Add** button.
 - d. Type the name of user who you would like to be the ADDI administrator in the **Enter the object name to select** field. In this example, it is `ldadmin` (the local administrator who is also the active directory administrator).
 - e. Click **Check Names**. The object name with the logon name is displayed.



f. Click **OK** to finish.



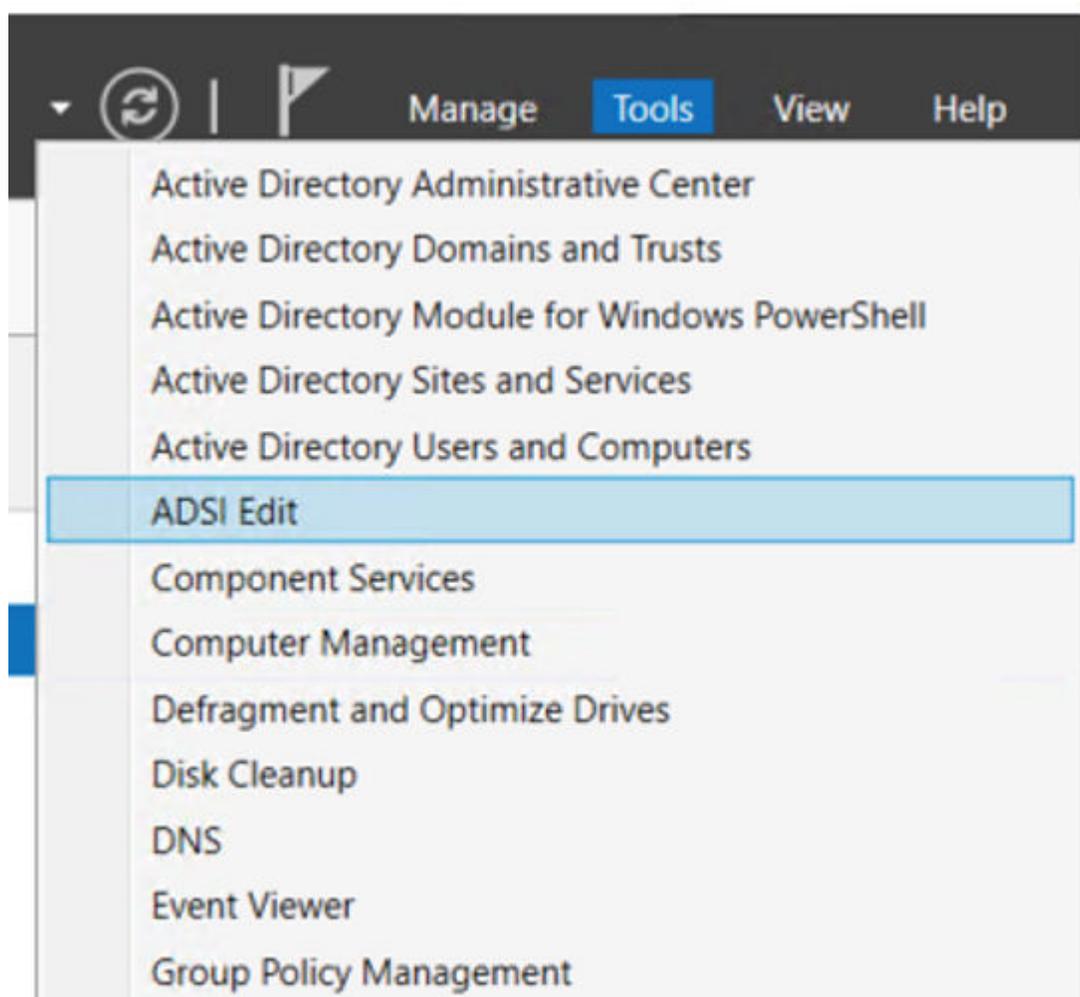
g.

h. Click **OK** to close the Addi Administrators Properties window.

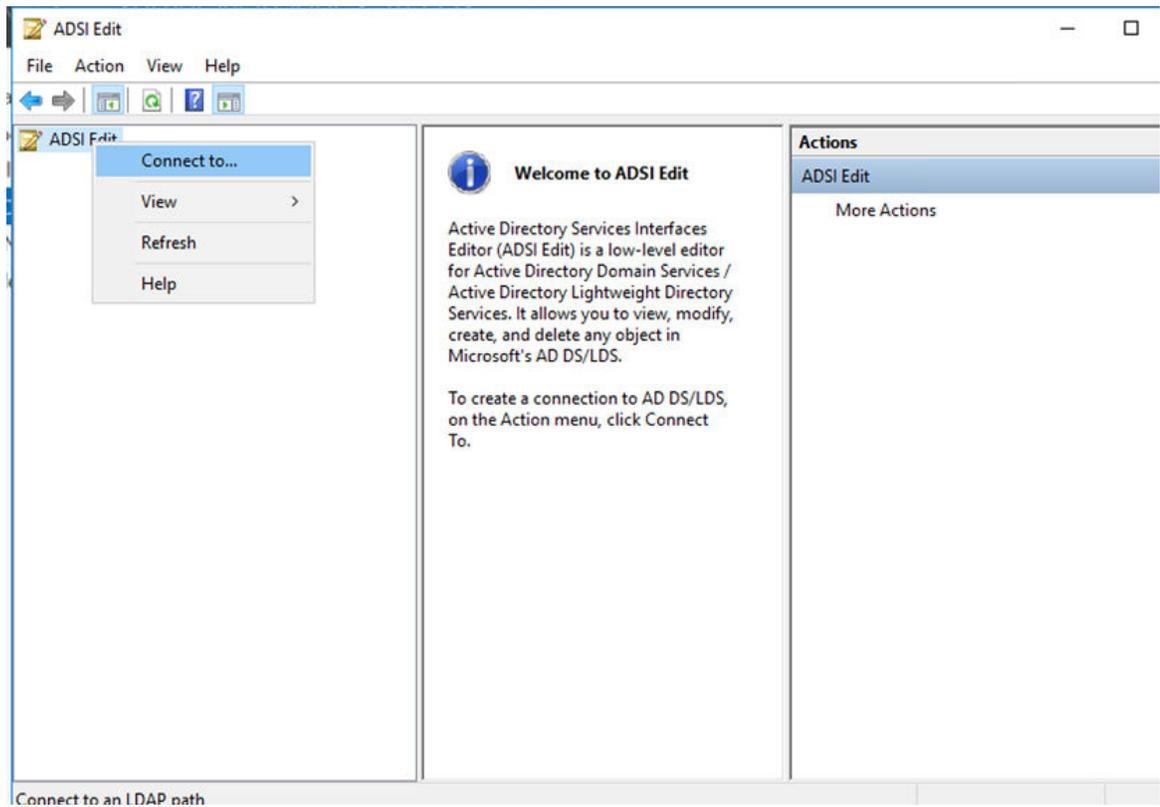
12. Close the Active Directory Users and Computers window.

13. Get the Distinguished Name of active directory administrator user to use for ADDI Extensions authentication setup.

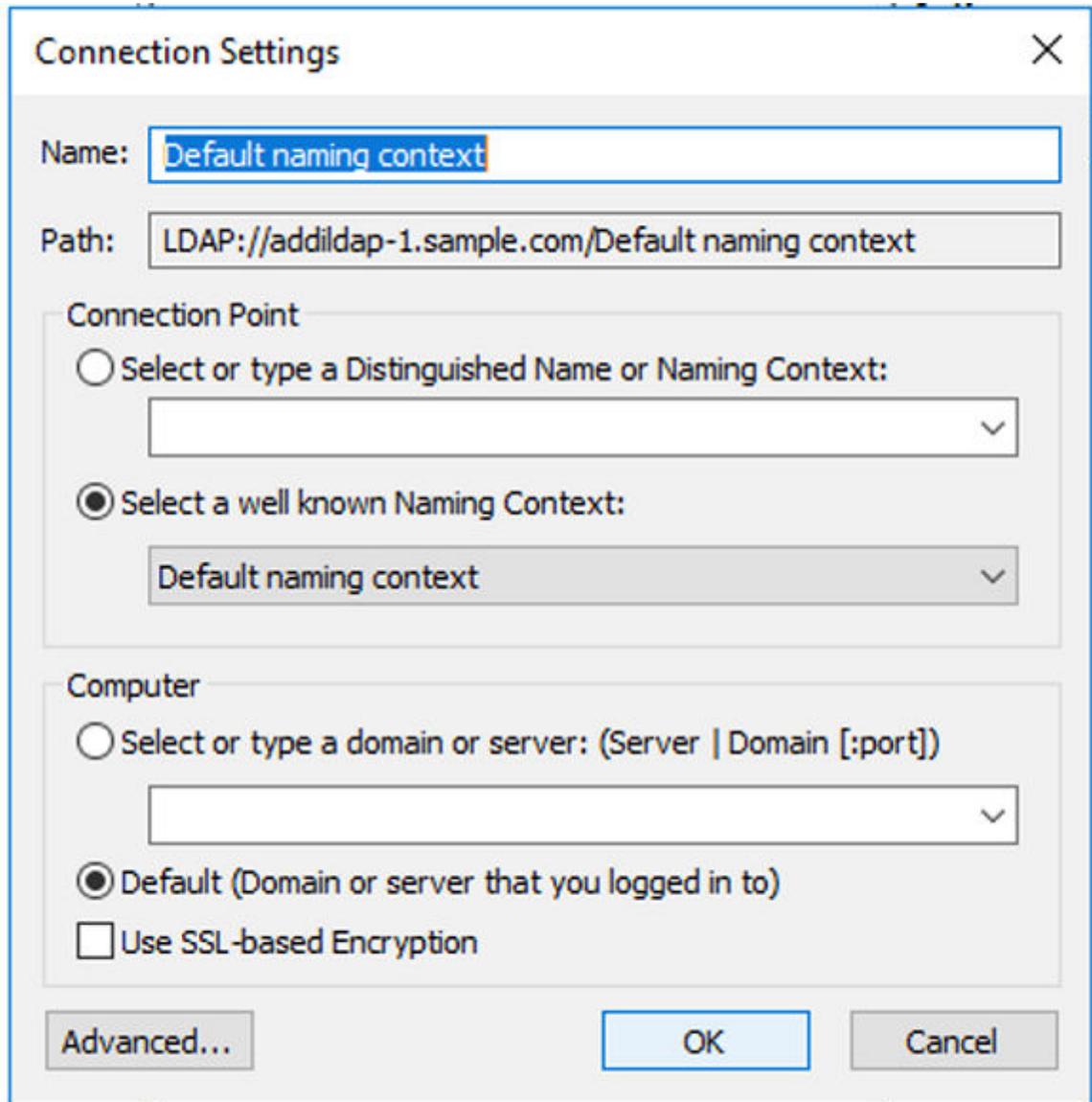
a. In the Server Manager, select **Tools > ADSI Edit** from the upper right menu.



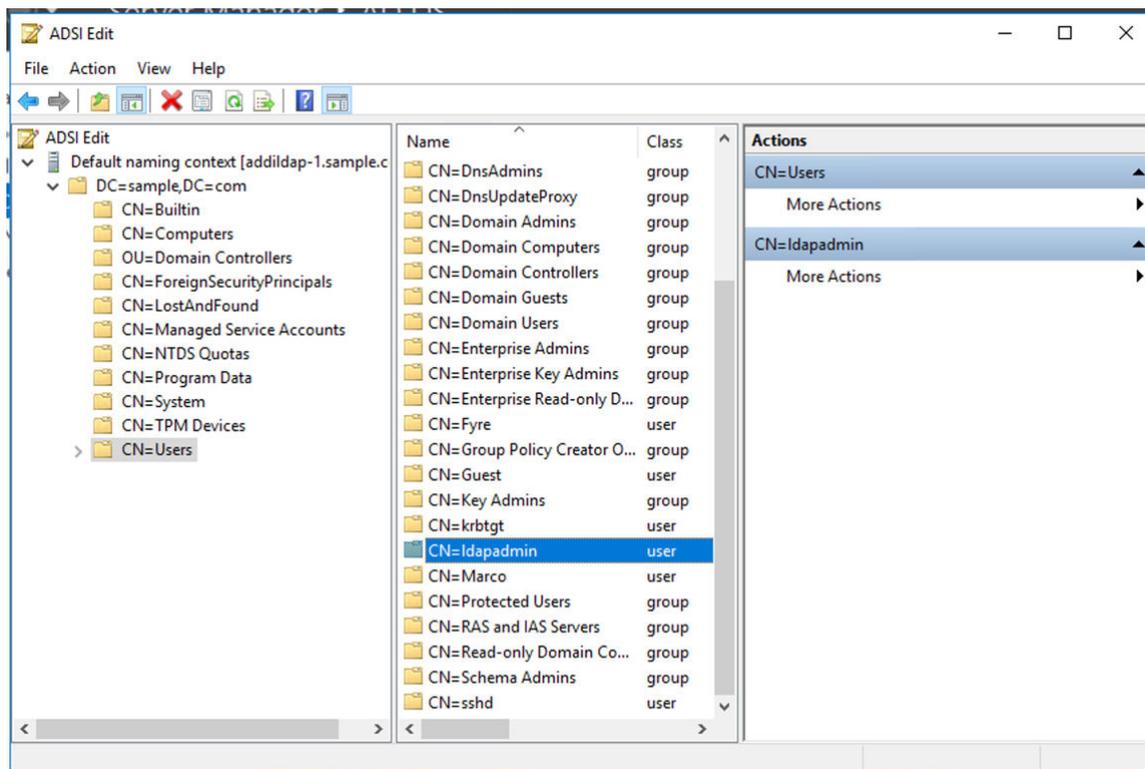
b. Right-click the **ADSI Edit** item and select **Connect to**.



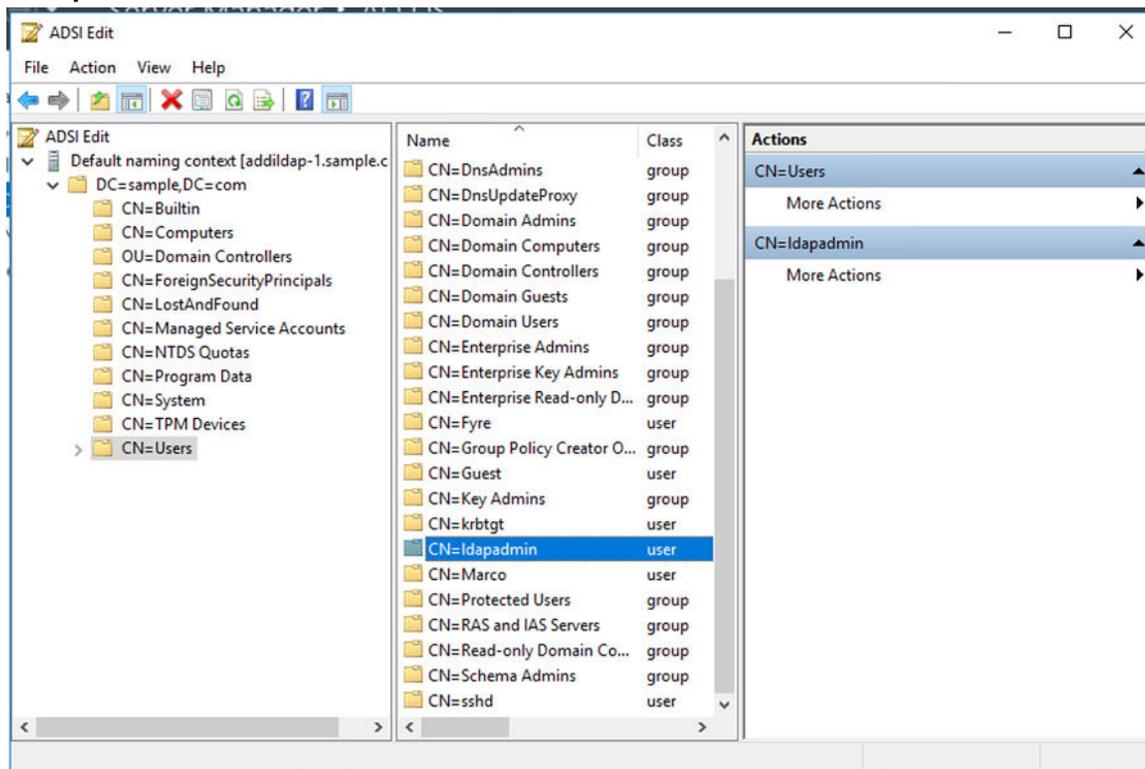
c. Keep the default settings and click **OK** in the Connection Settings window.



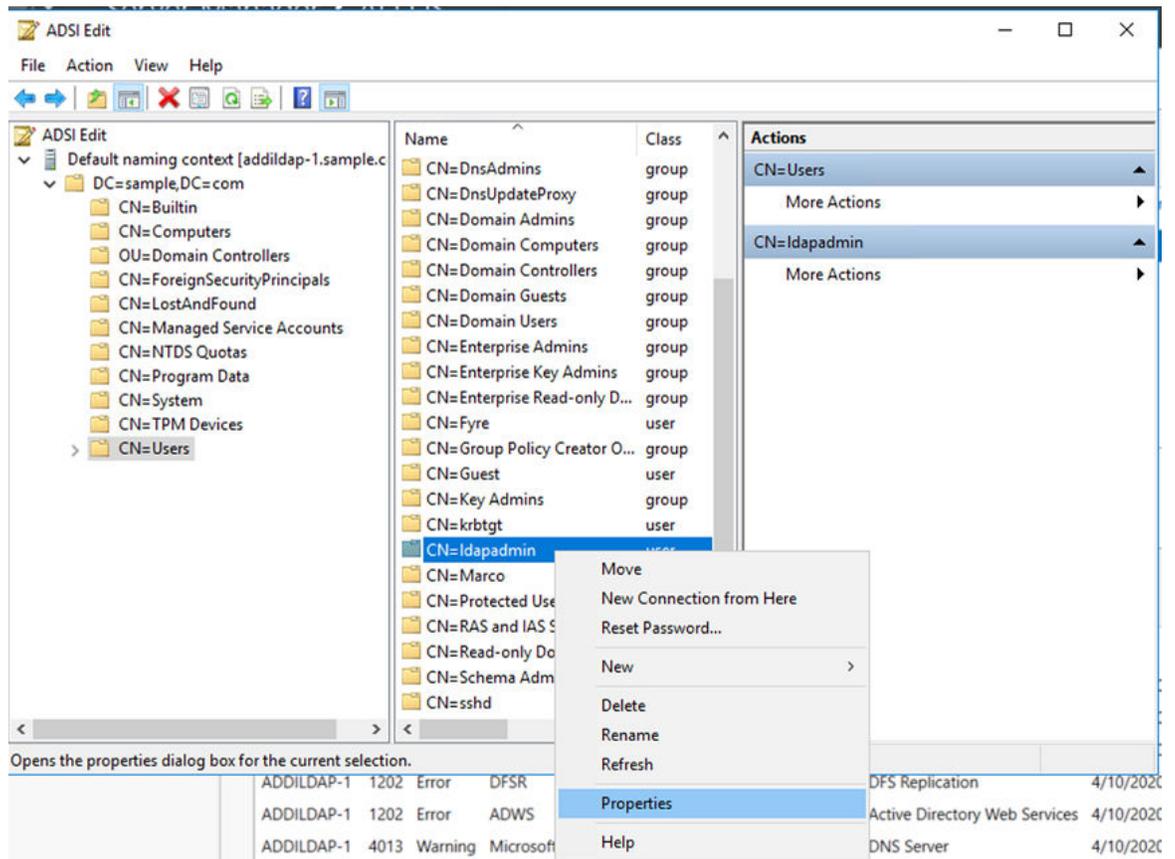
- d. The default object tree for your sample.com domain is populated.
- e. Select the **CN=Users** object. The objects within the **CN=Users** are displayed in the middle pane.



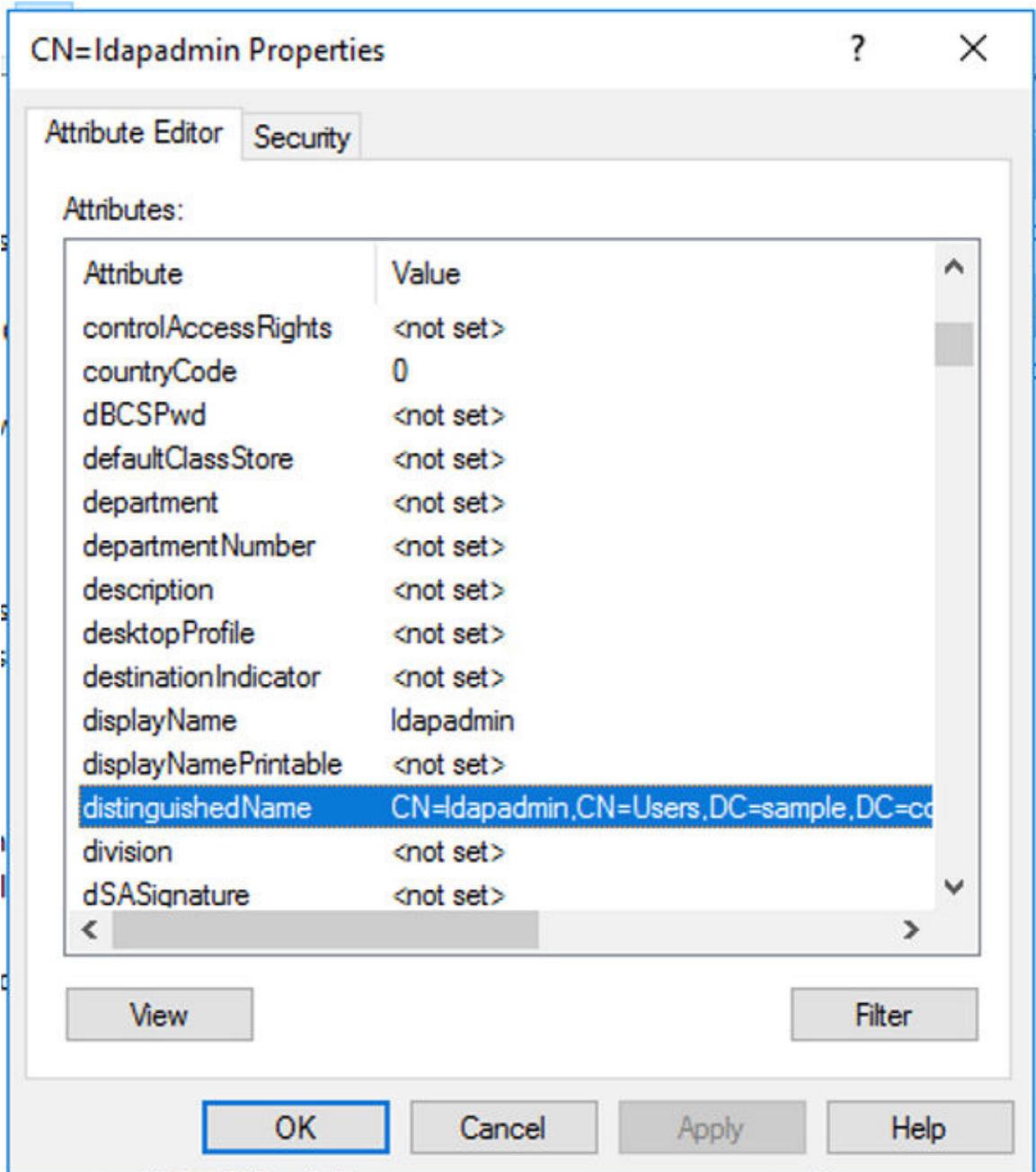
f. Select the active directory administrator user that you set up in the previous steps. In this case, it is **Ldapadmin**.



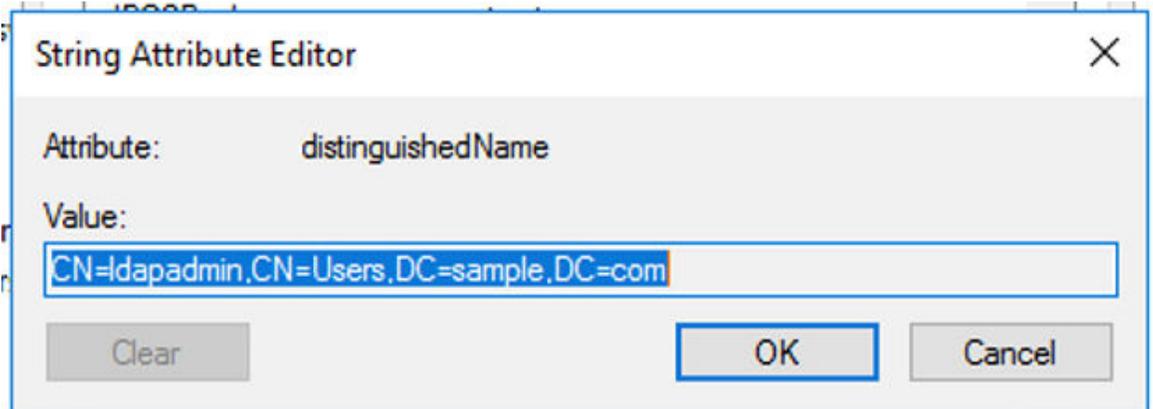
g. Right-click and select **Properties**.



h. Find the **distinguishedName** attribute and double-click.



- i. Copy the **distinguishedName** and paste on a note for future use.



- j. Click **OK** to close the String Attribute Editor dialog box.

k. Close the ADSI Edit window.

You have finished the setup of active directory users to access ADDI Extensions through LDAP. Next step is to install and setup ADDI Extensions.

Installing and setting up ADDI Extensions

Before the installation, you need to update the hosts file to understand the public URI that you use for the setup. For example, complete the following steps to update the hosts file.

1. Type Notepad in the search box next to the **Start** menu icon.
2. In the search results, right-click **Notepad**, and select **Run as administrator**.
3. From the Notepad, open the hosts file in the C : \Windows\System32\drivers\etc directory.
4. Add the following entry under the localhost section.

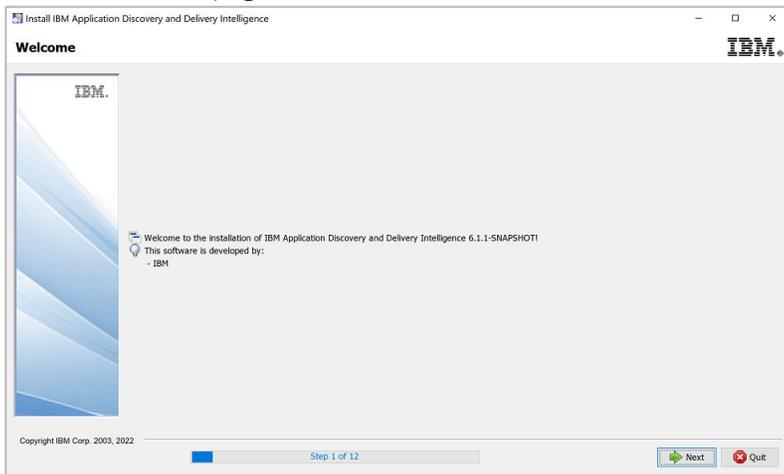
```
127.0.0.1    sample.com
```

5. Save the hosts file.

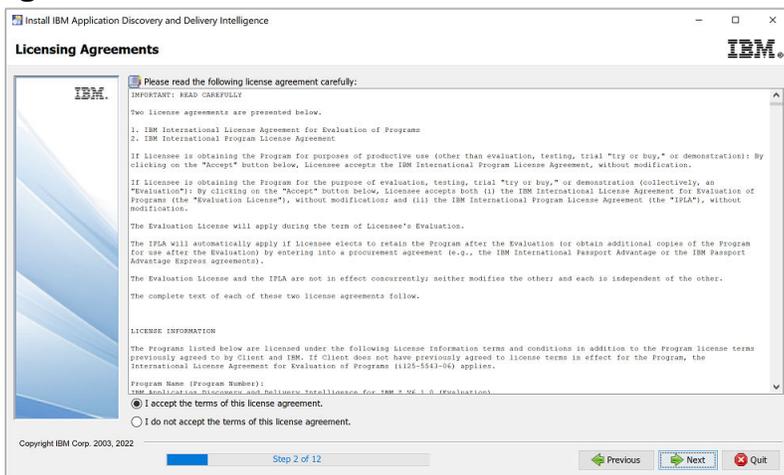
Complete the following steps to install and set up ADDI Extensions.

1. Download the ADDI installer and run the ADDI installer wizard as an administrator.
2. Follow the instructions in the wizard to install ADDI.

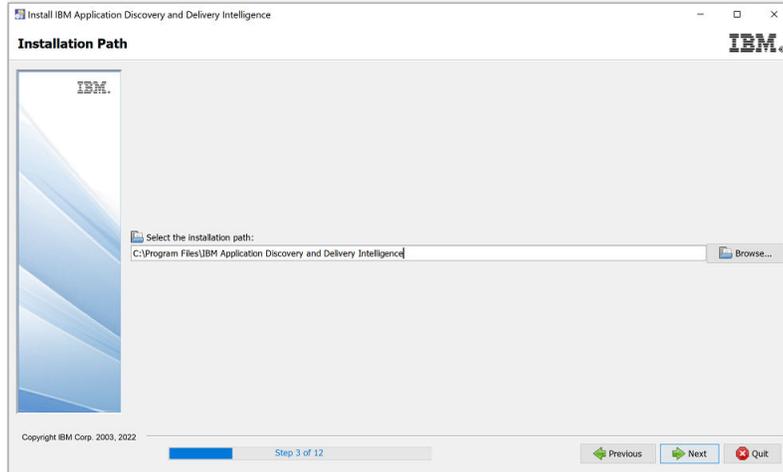
a. On the **Welcome** page, click **Next**.



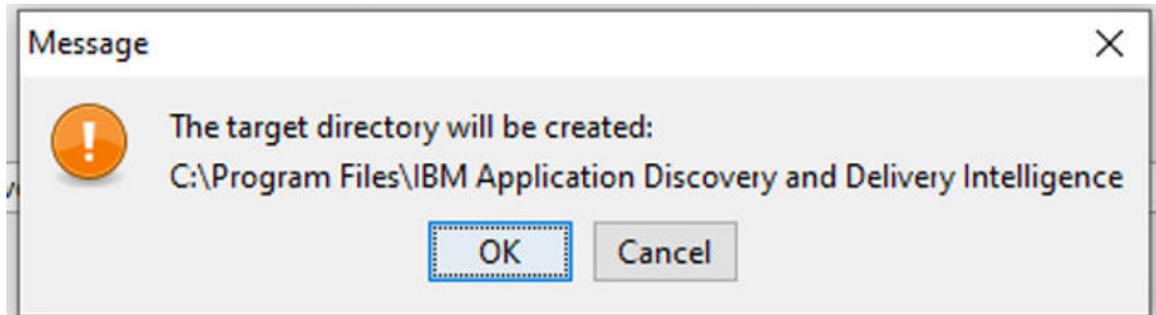
b. Review the **Licensing Agreements** page. Then, select **I accept the terms of this license agreement** and click **Next**.



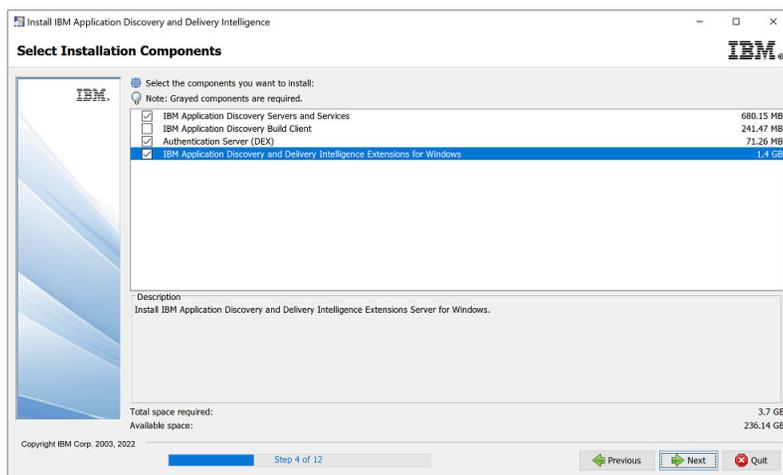
- c. On the **Installation Path** page, click **Next** to use the default path or click **Browse** to select a path to install IBM ADDI and click **Next**.



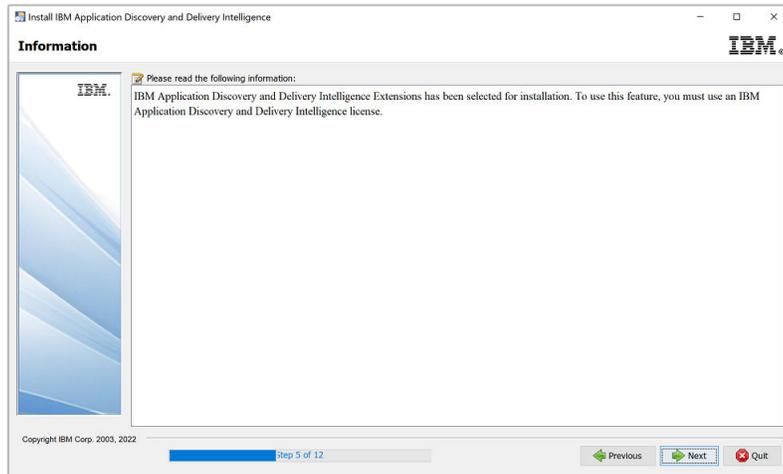
- d. Click **OK** in the Message dialog box to create the target directory.



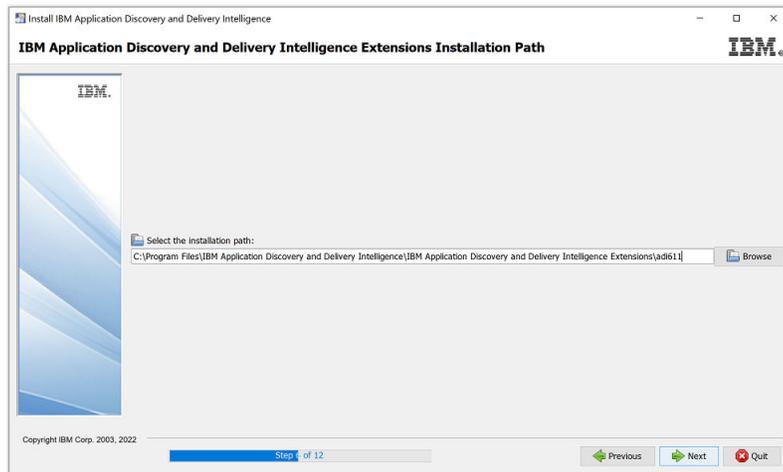
- e. Check the following components to install and click **Next**.
- IBM Application Discovery Servers and Services
 - Authentication Server (DEX)
 - IBM Application Discovery and Delivery Intelligence Extensions for *your system*. The following example shows the **Select Installation Components** page that is displayed on a Windows system.



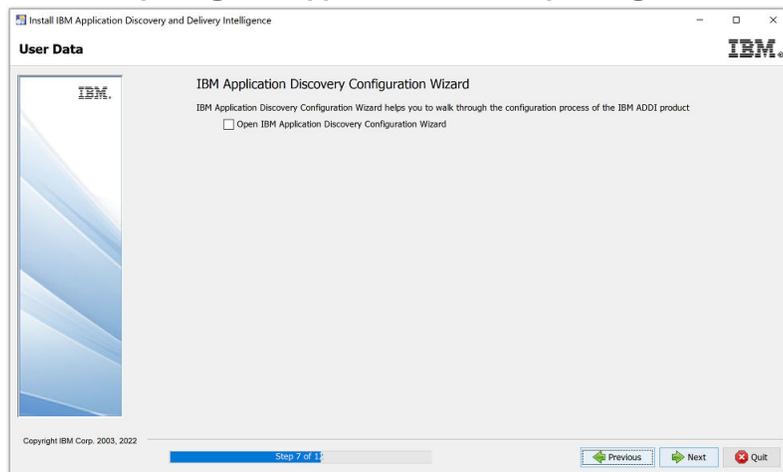
- f. Click **Next** on the **Information** page.



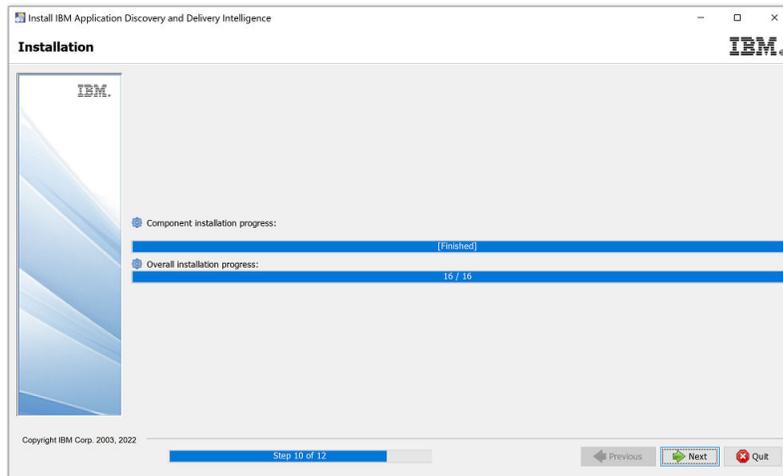
- g. On the **IBM Application Discovery and Delivery Intelligence Extensions Installation Path** page, click **Next** to use the default path or click **Browse** to select a path to install IBM ADDI Extensions and click **Next**.



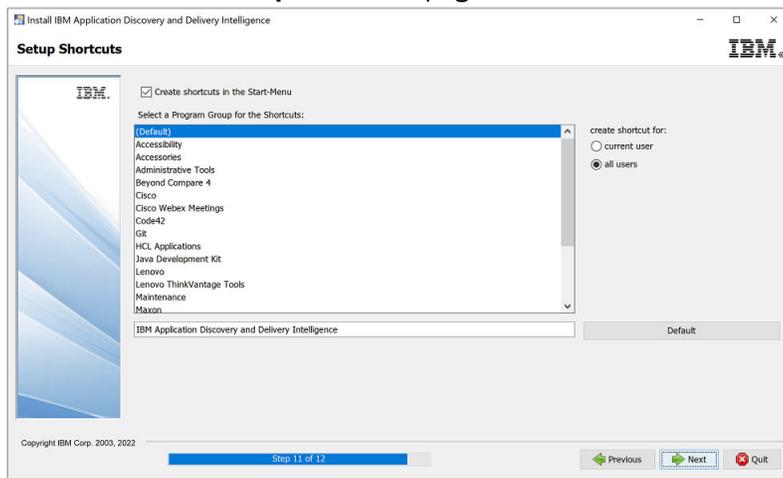
- h. Click **OK** in the Message dialog box to create the target directory.
 i. Clear the **Opening IBM Application Discovery Configuration Wizard** checkbox.



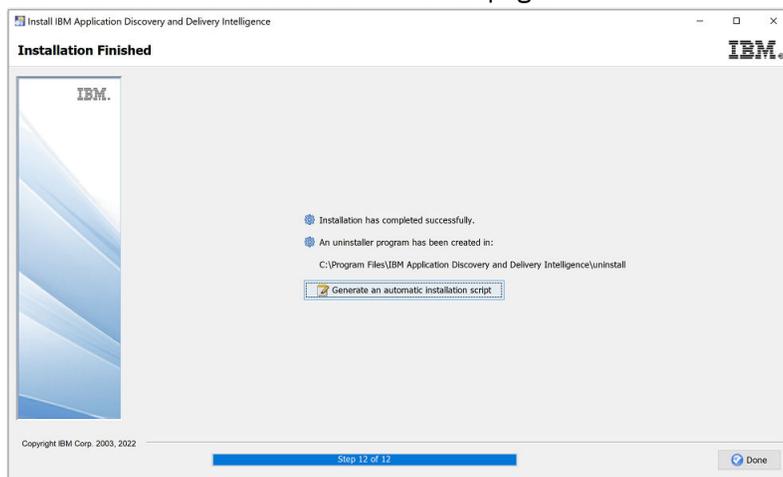
- j. Click **Next** on the **Installation** page when the installation progress is finished.



k. Click **Next** on the **Setup Shortcuts** page.



l. Click **Done** on the **Installation Finished** page.



3. Open the command prompt as an administrator and navigate to `c : \Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server` directory.

4. Run the following command to generate the bcrypt hash of an admin password:

```
adi-setup bcryptPassword -dex.password <password>
```

Note: In the following example, **adiadmin** is a password that you want to generate the bcrypt hash.

```

C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5106\serve
r>adi-setup bcryptPassword -dex.password adiaadmin
Mar 19, 2020 2:46:11 AM com.ibm.dimez.tools.setup.common.AdiSetup showWelcome
INFO: CRIDA0398I ADI Setup started with operation "bcryptPassword".
Mar 19, 2020 2:46:12 AM com.ibm.dimez.tools.setup.common.AdiSetup bcryptPassword
INFO: CRIDA0654I The bcrypt hash of the password is: $2a$10$ztA5trILtRdTjVd9LTe.OwVoYtfgIVaCxi9YQPCsrQmWIDMej2
[BeanContext] = [2,0,0%]
[total] = [2,0,0%]

```

5. Save the generated bcrypt hash password somewhere. The password will be used when you set up the dex.yaml file.
6. Configure the Authentication Server (DEX) as described in the following steps:
 - a. Navigate to the C:\Program Files\IBM Application Discovery and Delivery Intelligence\Authentication Server (DEX)\sample-conf\addi directory.
 - b. Copy all three files in the directory: dex.yaml, root.crt, and root.key.

Note: All these files are provided for only evaluation purposes. For the production server, you must generate SSL keystore and security certificate for your server and configure your own dex.yaml file as described in the [“Configuring the parameters in the dex.yaml file”](#) on page 155 topic.

- c. Navigate to the c:\Program Files\IBM Application Discovery and Delivery Intelligence\Authentication Server (DEX)\conf\ directory and paste the copied files there.
- d. Run the text editor as the administrator and update the dex.yaml file with the following changes:
 - i) Update the **issuer** to https://sample.com:7600/dex.

```

# The base path of dex and the external name of the OpenID Connect service.
issuer: https://sample.com:7600/dex

```

- ii) Update the **web** section with the following changes:

- Change the http property to https: sample.com:7600.
- Uncomment the TLSCert and TLSKey properties and update their paths as shown in the following sample.

web:

```

https: sample.com:7600
#if https is used provide path to certificate and key.
TLSCert: conf/root.crt
TLSKey: conf/root.key

```

- iii) Make sure that the **connectors** section is uncommented and update the section with the following changes:

- a) Change the **host** to localhost:389.
- b) Change the **insecureNoSSL** value to true.
- c) Change the **bindDN** to the distinguished name of Active Directory administrator that you have copied to the note in the previous steps.
- d) Change the **bindPW** to the password of Active Directory administrator.

```

connectors:
- type: ldap
  name: OpenLDAP
  id: ldap
  config:
    host: localhost:389

    # No TLS for this setup.
    insecureNoSSL: true

    # This would normally be a read-only user.
    bindDN: CN=ldapadmin,CN=Users,DC=sample,DC=com
    bindPW: ██████████

```

- e) Update the **baseDN** in the **userSearch** section to CN=Users,DC=sample,DC=COM.

```

userSearch:
  baseDN: CN=Users,DC=sample,DC=COM
  filter: "(objectClass=person)"
  username: userPrincipalName
  # "DN" (case sensitive) is a special attribute name. It indicates that
  # this value should be taken from the entity's DN not an attribute on
  # the entity.
  idAttr: DN
  emailAttr: userPrincipalName
  nameAttr: cn

```

- f) Update the **baseDN** in the **groupSearch** section to CN=Users,DC=sample,DC=COM.

```

groupSearch:
  baseDN: CN=Users,DC=sample,DC=COM
  filter: "(objectClass=group)"

  # A user is a member of a group when their DN matches
  # the value of a "member" attribute on the group entity.
  userAttr: DN
  groupAttr: member

  # The group name should be the "cn" value.
  nameAttr: cn

```

- iv) Update the staticClients section with the following changes while removing the <<>> brackets.

- Update the id property to addi-liberty. Remove the extra leading space characters on this line.
- Replace the localhost within the redirectURIs property with sample.com.
- Update the name property to 'ADDI Liberty Server'.
- Update the secret to f1a75f8abc2ffcbd46e2c1b5f7b12c7b.
- Comment out lines from 77 through 81 by using # symbol in front of each of those lines.

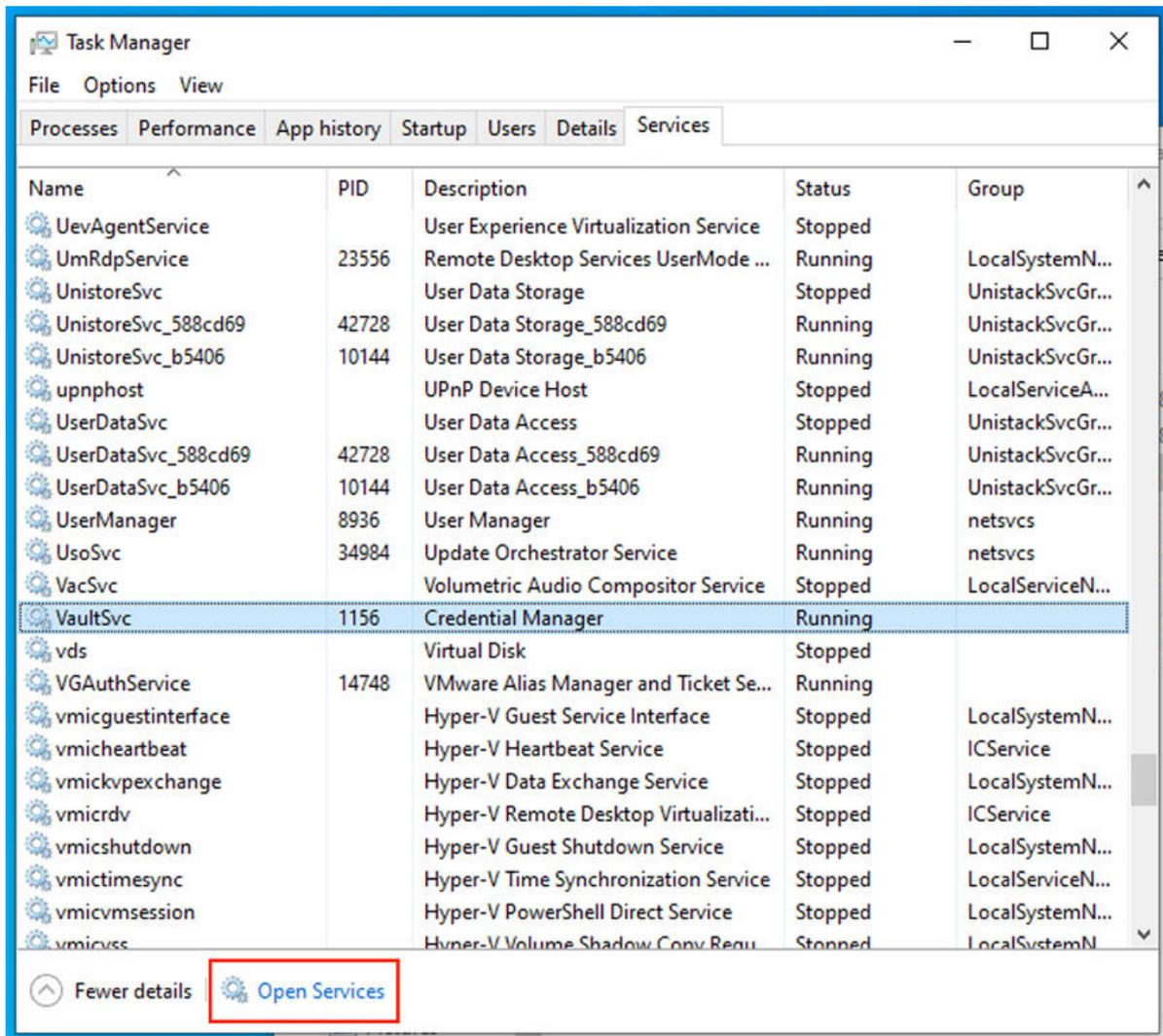
```
staticClients:
- id: addi-liberty
  redirectURIs:
  - 'https://sample.com:9753/oidcclient/redirect/addi-liberty'
  name: 'ADDI Liberty Server'
  secret: f1a75f8abc2ffcbbd46e2c1b5f7b12c7b
```

- v) Update the **StaticPasswords** section with the following changes while removing the <<>> brackets.
- Uncomment the email property and update it to "adiadmin@sample.com". Remove the extra leading space characters on this line.
 - Uncomment the hash property and update it as the hash password that you saved in step 5 with double quotes.
 - Uncomment the username property and update it to "adiadmin".

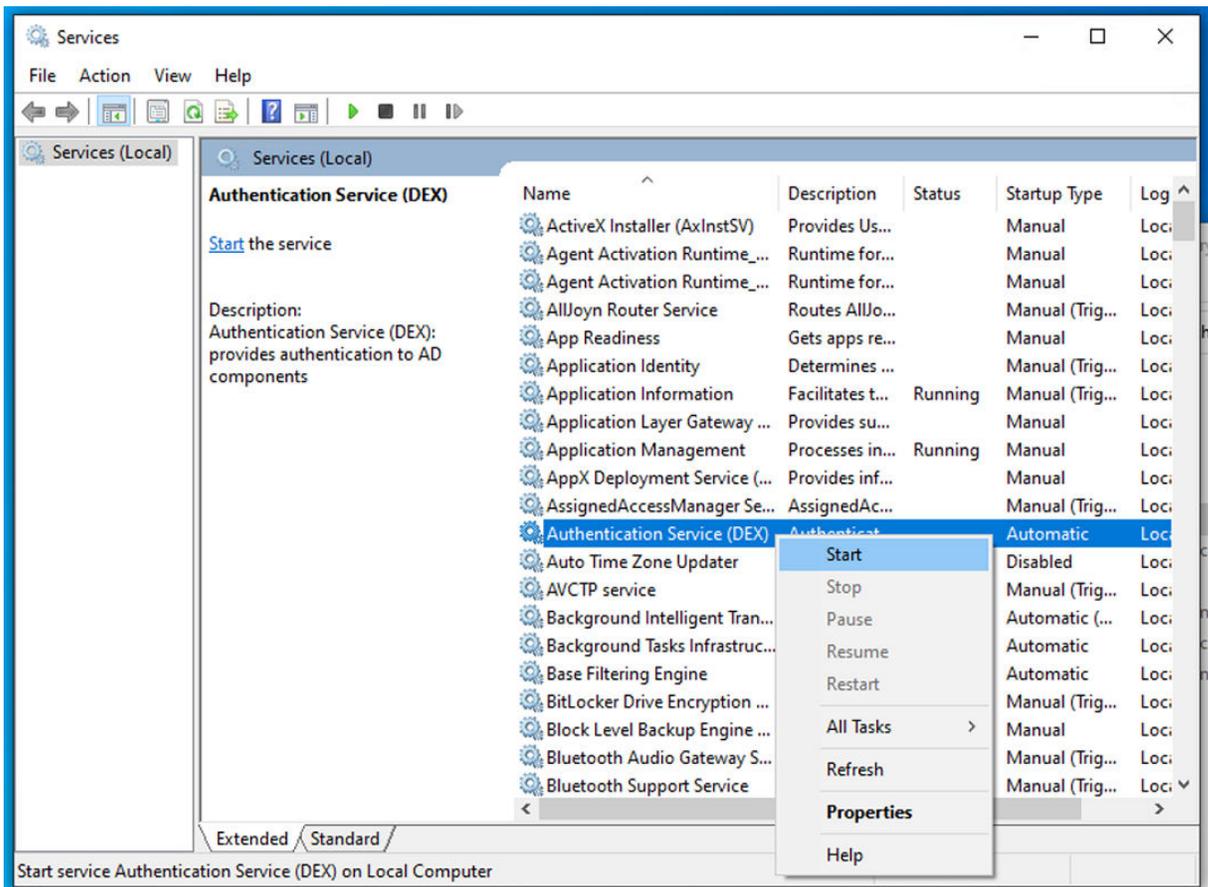
```
# A static list of passwords to login the end user. By identifying here, dex
# won't look in its underlying storage for passwords.
#
# If this option isn't chosen users may be added through the gRPC API.
staticPasswords:
- email: adiadmin@sample.com
  hash: "$2a$10$pKqTbFTBeu520Gerxqyks0Vifxq9vXLR4yEvd9jts.zPNohFAsiyi"
  username: "adiadmin"
```

Note: The hash value comes from the value that you saved on step 5.

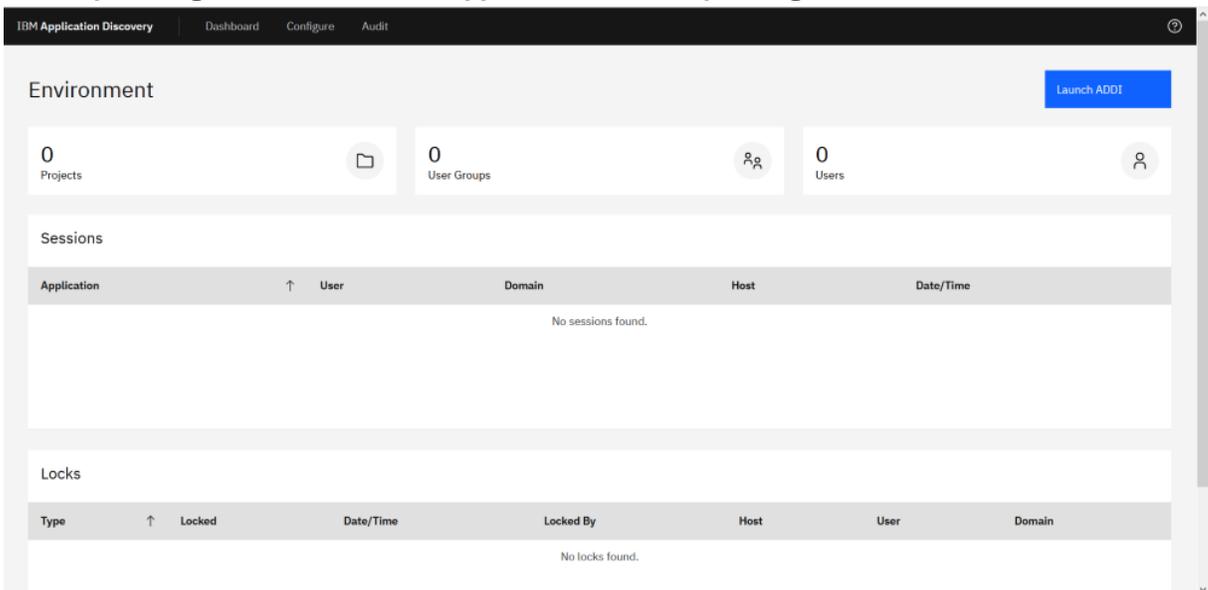
- e. Save the dex.yaml file.
7. Press Ctrl + Alt + Del and choose **Task Manager** to open the Task Manager window.
 8. Select the **Services** tab and click **Open Services** on the bottom of the **Task Manager** window.



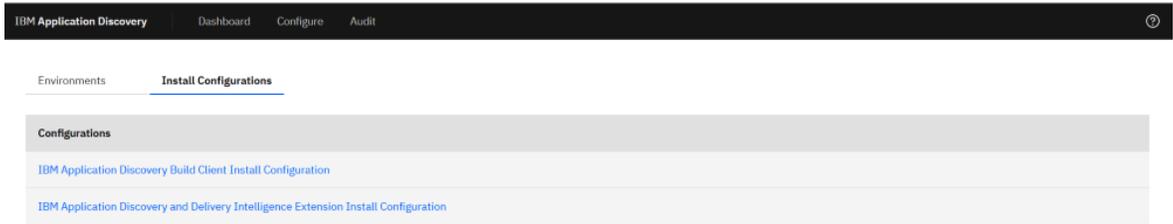
9. Right-click the **Authentication Service (DEX)** and select **Start** to start the service.



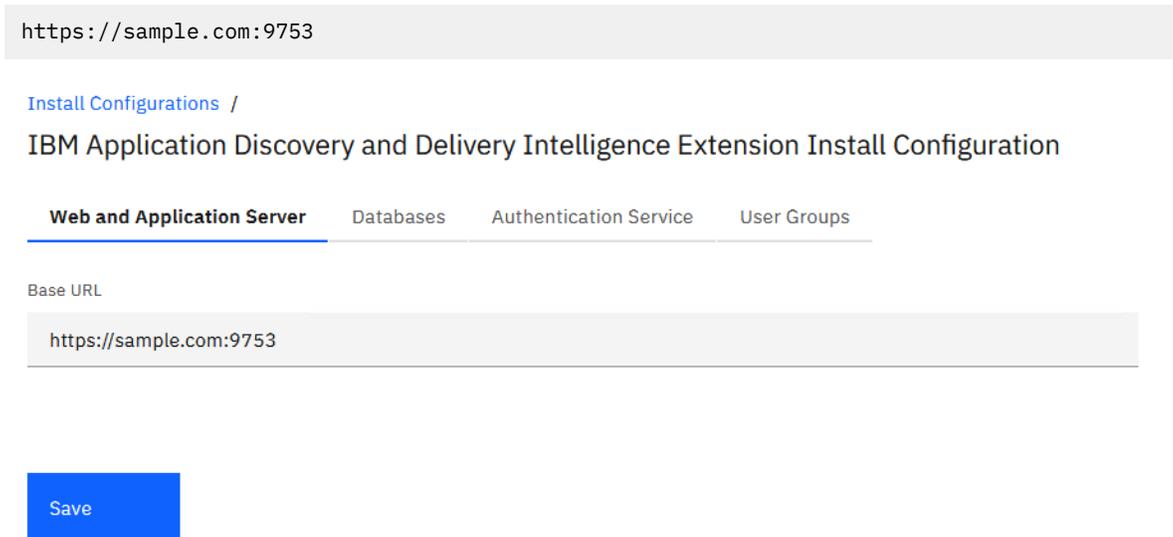
- To access **IBM AD Configuration Service**, go to **Start Menu > IBM Application Discovery and Delivery Intelligence > Launch IBM Application Discovery Configuration Service Admin**.



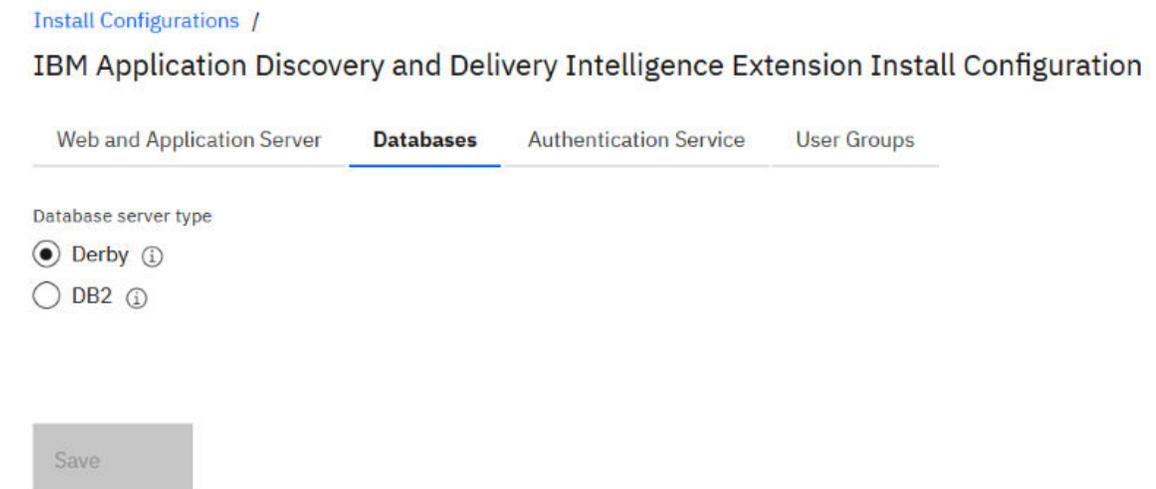
- Complete the ADDI Extensions Install Configuration as described in the following steps:
 - Select **Configure > Install Configurations > IBM Application Discovery and Delivery Intelligence Extensions Install Configuration**.



b. Specify the Base URL on the **Web and Application Server** tab as shown in the following example.



c. Leave the default value on the **Databases** tab.



d. Specify the following information on the **Authentication Service** tab.

- **Host:** sample.com
- **Port:** 7600
- **HTTP protocol:** Select **HTTP Secure (https)**.

[Install Configurations](#) /

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server Databases **Authentication Service** User Groups

Host

sample.com

Port

7600

HTTP protocol

HTTP

HTTP Secure (https)

Save

e. Specify the following information on the **User Groups** tab.

- **Admin Group List:** ADDI Administrators
- **User Group List:** zMobile, HRM App

[Install Configurations](#) /

IBM Application Discovery and Delivery Intelligence Extension Install Configuration

Web and Application Server Databases Authentication Service **User Groups**

Admin Group List (optional)

ADDI Administrators

User Group List (optional)

zMobile, HRM App

Save

f. Click **SAVE** to create the ADDI Extensions configuration. A message is displayed to indicate the configuration was successfully created.

12. On your Command Prompt window, navigate to c:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server directory.

Note: If you have closed the Command Prompt previously, you need to run the Command Prompt again as an administrator.

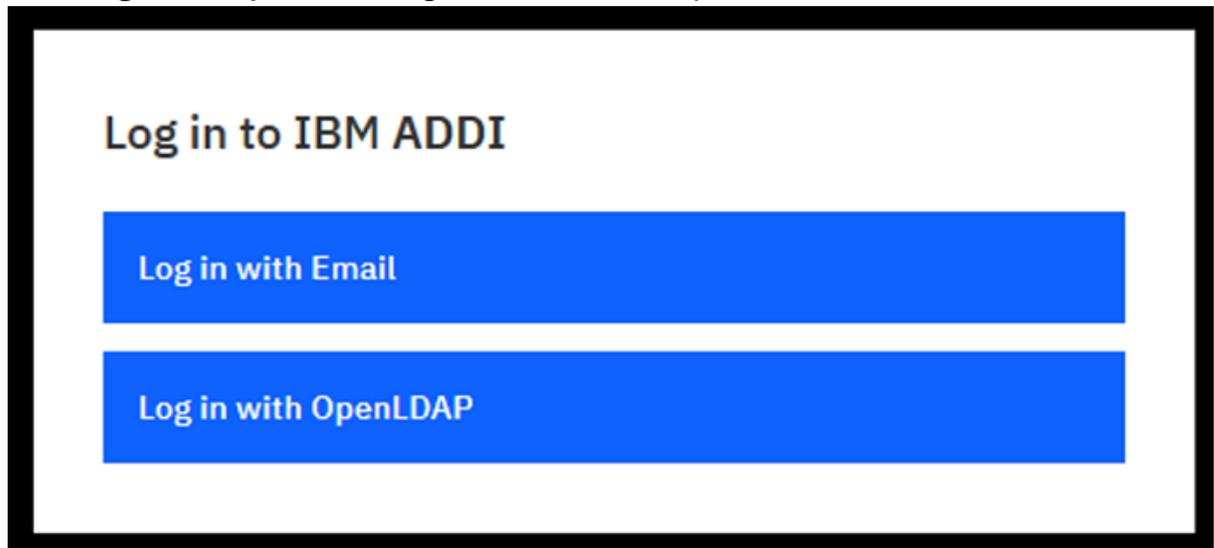
13. Run the command `adi-setup addiConfigurationServer`.

```
C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5107\server>
adi-setup addiConfigurationServer
Jun 25, 2020 12:11:47 PM com.ibm.dimez.tools.setup.common.AdiSetup showWelcome
INFO: CRIDA0398I ADI Setup started with operation "addiConfigurationServer".
[BeanContext] = [2,0,0%]
[UrlEncodingSerializer] = [1,0,0%]
[JsonSerializer] = [1,1,50%]
[JsonParser] = [1,1,50%]
[total] = [5,2,28%]
```

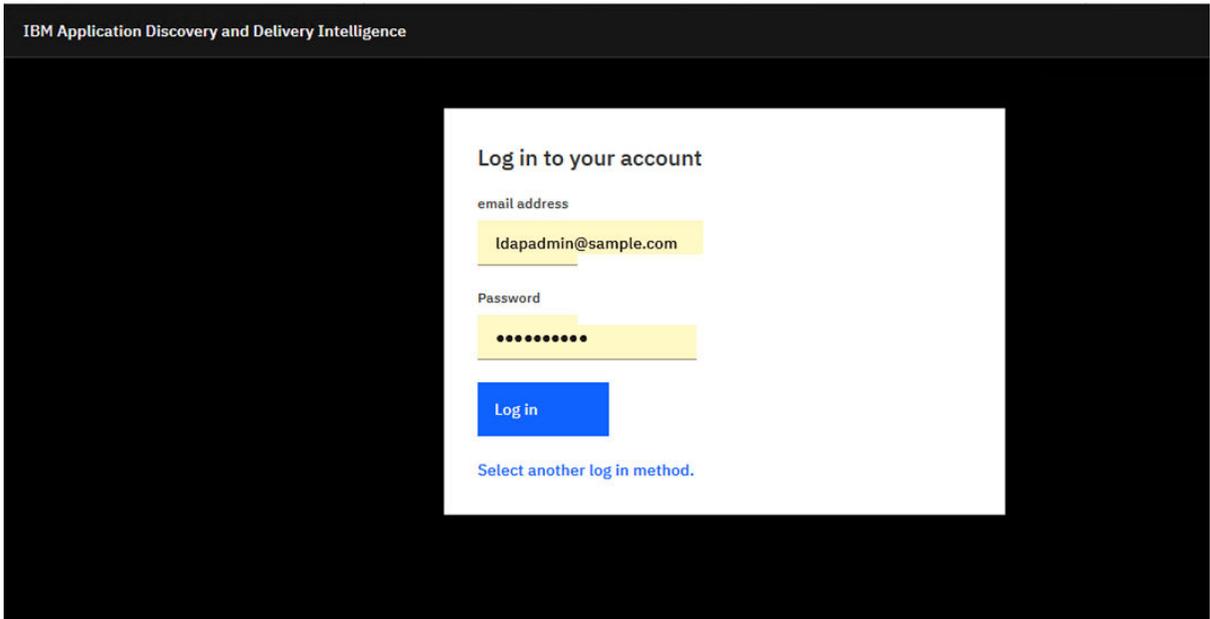
14. Run the `server.startup.bat` command and wait until the server is started successfully.

```
C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\adi5107\server>
server.startup.bat
"Installing adi-elasticsearch service. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Appl
ication Delivery Intelligence\adi5107\server\..\elasticsearch\logs directory for results."
Installing service      : "adi-elasticsearch"
Using JAVA_HOME (64-bit): "C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery
Intelligence\adi5107\server\jre"
The service 'adi-elasticsearch' has been installed.
"Starting adi-elasticsearch service. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Applic
ation Delivery Intelligence\adi5107\server\..\elasticsearch\logs directory for results."
The service 'adi-elasticsearch' has been started
"Starting Derby Network Server. Check C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application
Delivery Intelligence\adi5107\server\ directory for results."
Starting server adiServer.
Thu Jun 25 12:12:39 EDT 2020 : Security manager installed using the Basic server security policy.
Thu Jun 25 12:12:40 EDT 2020 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connections
on port 1527
Server adiServer started.
```

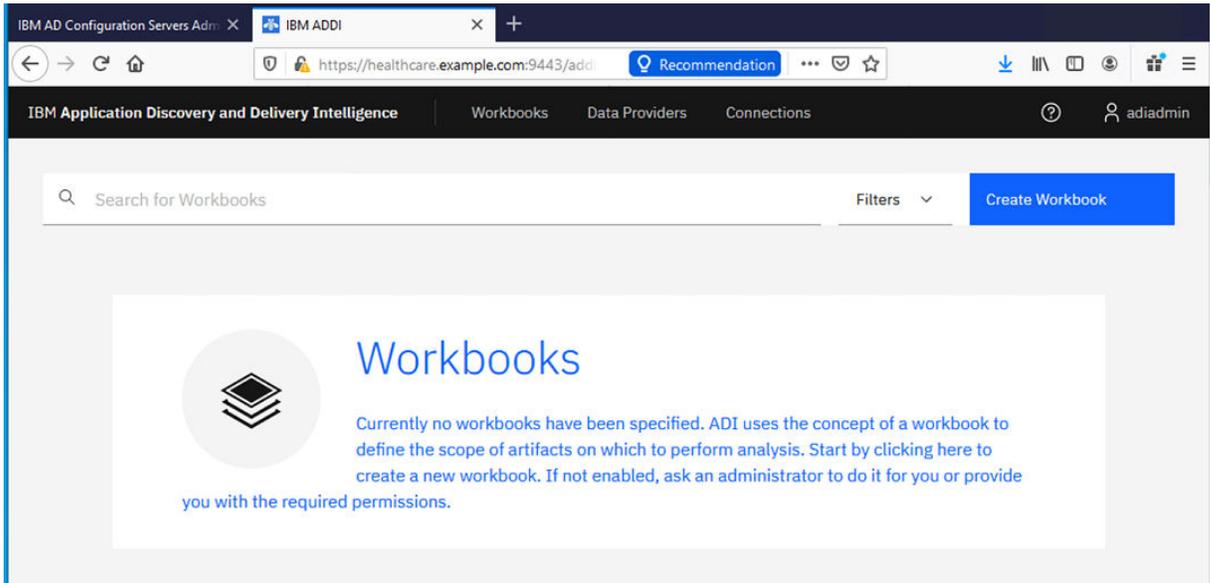
15. Browse to <https://sample.com:9753/addi/web/workbook> to test your access to ADDI Extensions.
16. Select **Log in with OpenLDAP** to login with active directory users.



17. Log in to ADDI Extensions by using a user that is a member of the **Addi Administrators** group.



18. The ADDI Extensions home page appears.



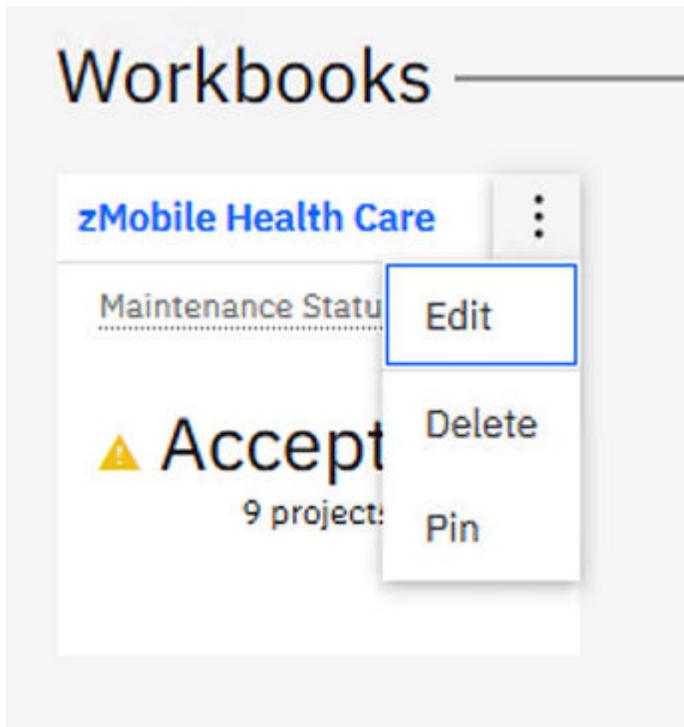
Now, you have finished the installation and configuration of ADDI to enable the login through Active Directory. Next, follow the steps in the [“Generating sample data”](#) on page 106 to generate the demonstration data and create a workbook accordingly for data analysis.

Note: You will log in by using the OpenLDAP method instead of the static user. Use **ADDI administrators** user that you have set up in the previous steps to perform the tutorial instead of the **adiadmin** user.

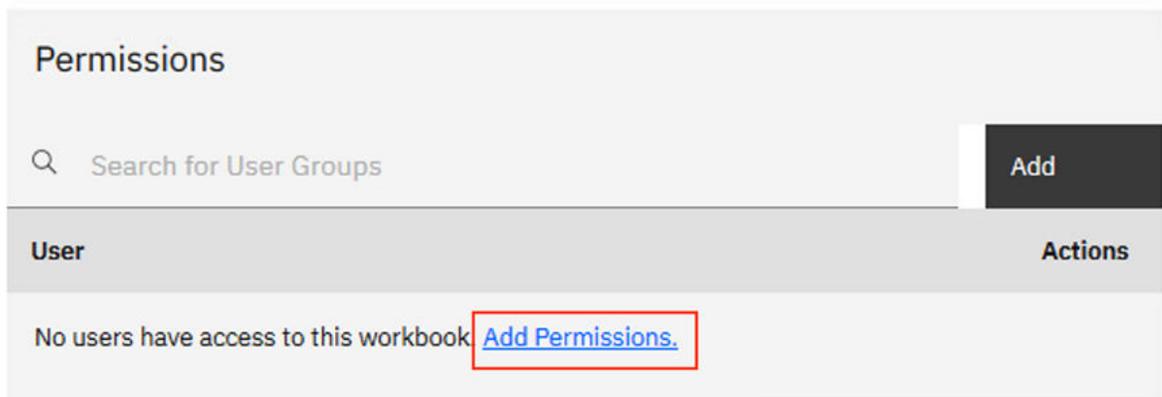
Setting up user permissions

After you complete the tutorial for [“Generating sample data”](#) on page 106, you need to complete the following steps to set up user permissions to allow only zMobile users to have access to this workbook.

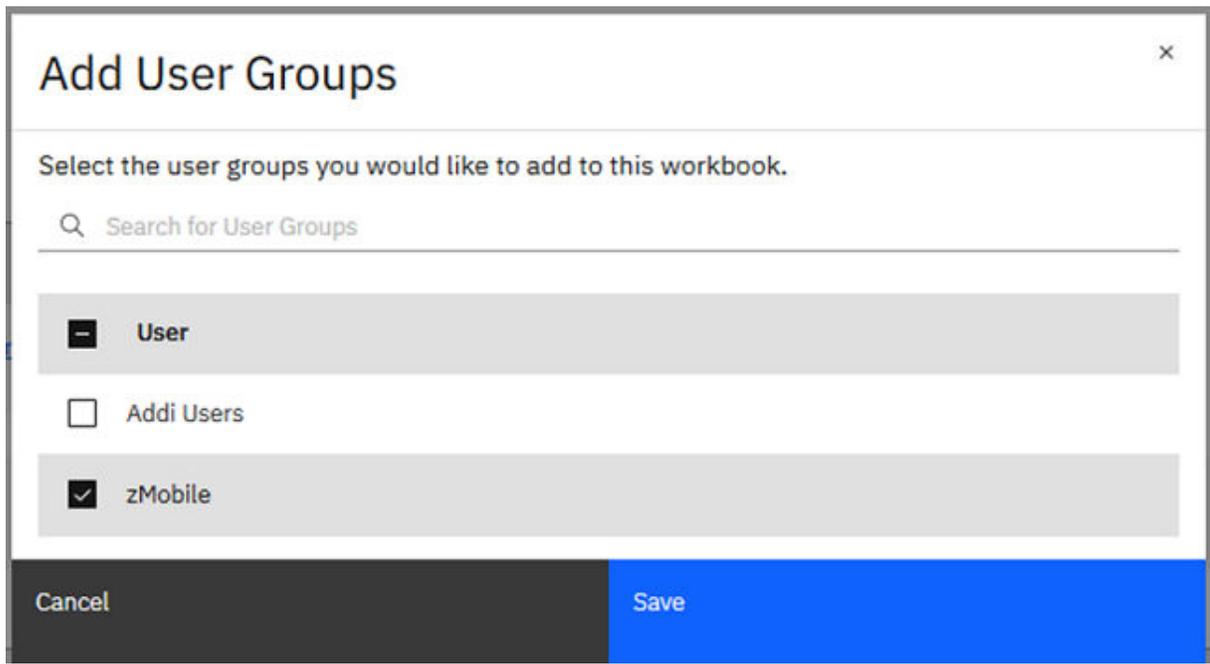
1. On the ADDI Extensions home page, click the overflow menu (vertical ellipsis) icon on the zMibile Health Care workbook card.
2. Select **Edit** to edit the workbook information.



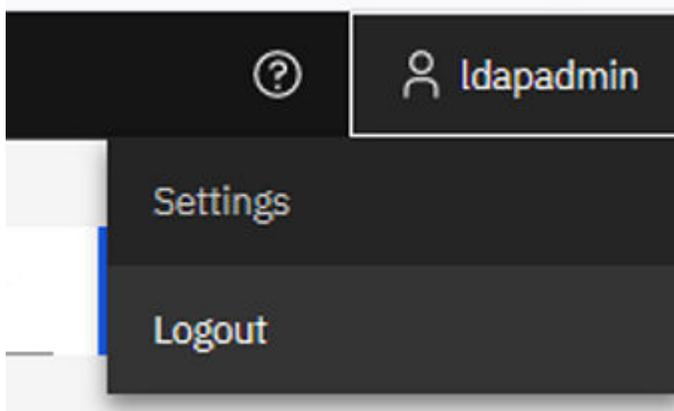
3. Click the **Add Permissions** link in the **Permissions** section.



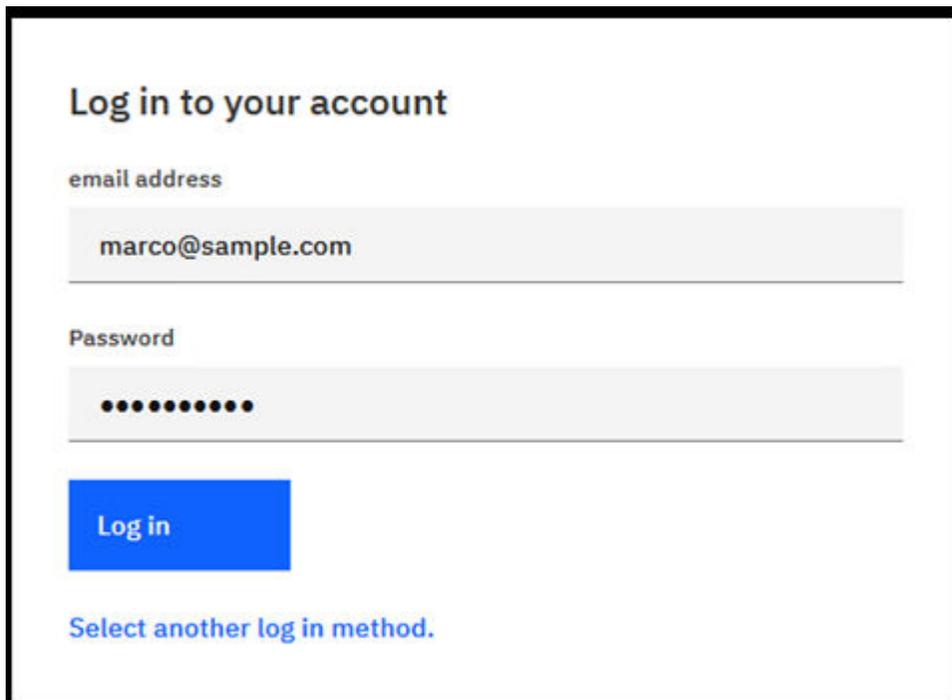
4. Select the **zMobile** checkbox in the Add User Groups window and click **Save** to save the updates of user groups.



5. Scroll down and click **Save** to update the workbook.
6. Click the profile icon on the upper right corner of the window and select **Logout**.



7. Select **Log in with OpenLDAP** to log back in.
8. Log in as Marco with the email address `marco@sample.com` and the password that you set for Marco.



Log in to your account

email address

marco@sample.com

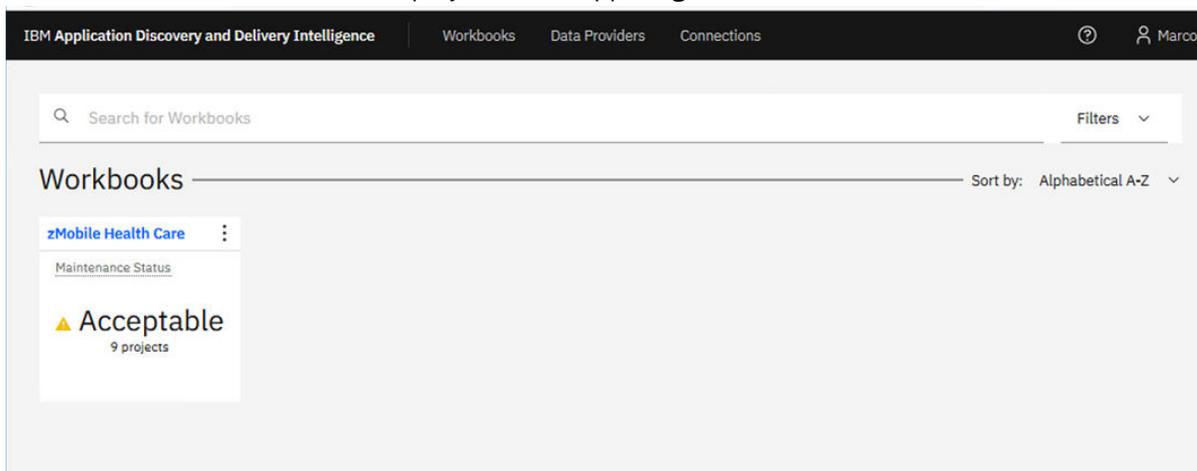
Password

●●●●●●●●

Log in

[Select another log in method.](#)

You can now see that Marco has access to this workbook. Since Marco is not an ADDI administrator, no **Create Workbook** button is displayed on the upper right corner of the window.



IBM Application Discovery and Delivery Intelligence | Workbooks | Data Providers | Connections | Marco

Search for Workbooks | Filters

Workbooks | Sort by: Alphabetical A-Z

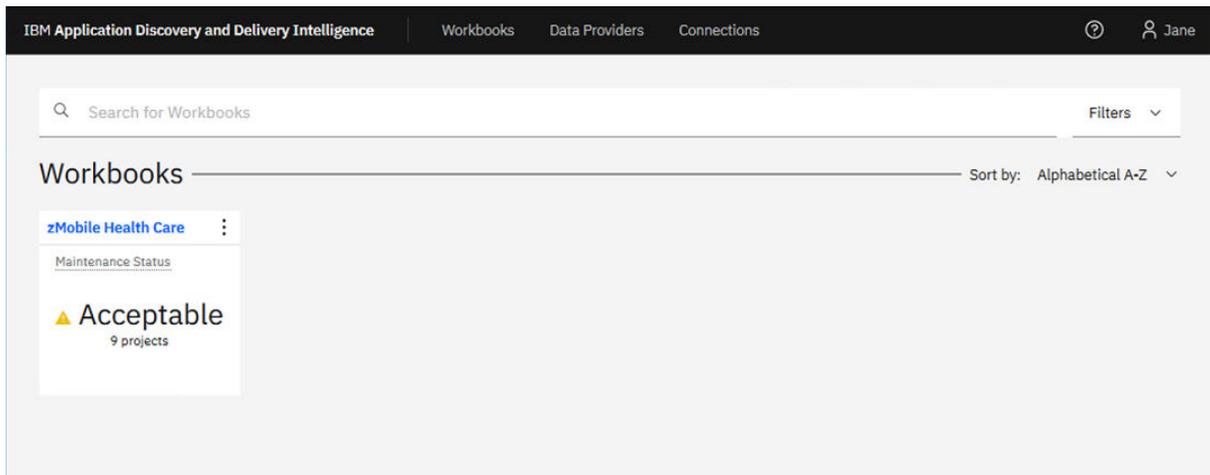
zMobile Health Care

Maintenance Status

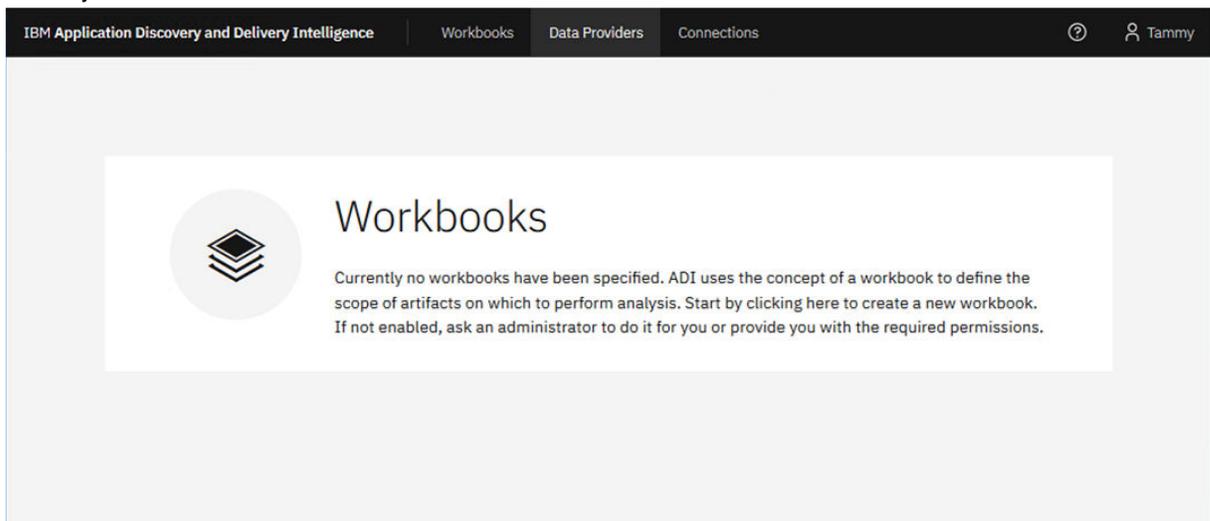
Acceptable

9 projects

9. Click the profile icon on the upper right corner of the window and select **Logout**.
10. Select **Log in with OpenLDAP** to log back in.
11. Log in as Jane with the email address jane@sample.com and the password that you set for Jane.



12. Click the profile icon on the upper right corner of the window and select **Logout**.
13. Select **Log in with OpenLDAP** to log back in.
14. Log in as Tammy with the email address `tammy@sample.com` and the password that you set for Tammy.



This time, you cannot see the zMobile Health Care workbook since Tammy is not a member of zMobile group.

You have now explored how ADDI Extensions manages user authentication and permissions through LDAP. You can now be able to set up your production environment by using your organization active directory.

Setting up and analyzing code coverage for the Manual Builds data provider

In this tutorial, you will learn how to create a Manual Builds data provider as a data source for code coverage results of SAM application. Then, you will use ADDI Extensions to perform a code coverage analysis for the new SAM Application.

Prerequisite

Before you begin this tutorial, you need to complete the following task.

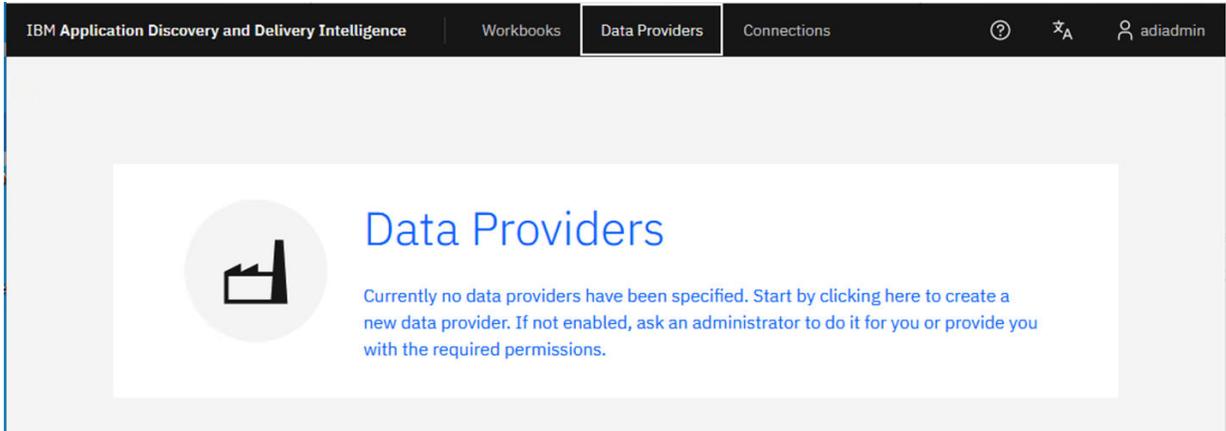
- [“Installing and setting up ADDI Extensions” on page 5](#)

Creating a Manual Builds data provider as a data source for code coverage results

A data provider is a source of data being analyzed. To add a new application for code coverage analysis, you need to start with adding a data provider. In this lab, you will create a Manual Builds data provider.

Complete the following steps to create a Manual Builds data provider to store code coverage results.

1. Navigate your browser to `https://healthcare.example.com:9753/addi/web/projects/HealthCare4All`.
2. Log in with `AdiAdmin` as the user ID and `AdiAdmin` as the password.
3. Select the **Data Providers** tab on the header to go to the **Data Providers** page. If you didn't create any data providers before, a message is displayed to indicate that no data providers have been created.



4. Click the **Data Providers** header on the message to create your first data provider or click the **Create Data Provider** button if this is not your first data provider. The **Create Data Provider** page is displayed.

[All Data Providers](#) / [Create Data Provider](#)

Create Data Provider

Name *

Description

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.

 Application Discovery	 Business Rule Discovery	 Manual Builds	 Rational Team Concert Builds
 System Management Facility			

Cancel Create

5. Enter the following information in the **Create Data Provider** page.

- **Name:** SAM Manual Builds
- **Description:** Manual builds for SAM application.

[All Data Providers](#) / Create Data Provider

Create Data Provider

Name *

SAM Manual Builds

Description

Manual builds for SAM application.

6. Select the **Manual Builds** data provider type.

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.

 Application Discovery	 Business Rule Discovery	 Manual Builds <input checked="" type="checkbox"/>	 Rational Team Concert Builds
 System Management Facility			

7. Complete the First Build section with the following information.

- **Build Name:** 2015Nov10Build
- **Date of Build:** 11/10/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
- **Description:** November 10, 2015 build for SAM application.
- **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > SAM App > November 10 2015 directory. Select all the files that include samt1.zip, samt2.zip, samt3.zip, samt4.zip, samt5.zip, samt6.zip, and samt7.zip at once.
- **Enable headless collection support:** Make sure that the checkbox is not selected.
- Select **All Days** for the number of days to keep data in the data warehouse.

First Build

Build Name *

2015Nov10Build

Date Of Build *

11/10/2015, 12:00 AM

Build Description

November 10, 2015 build for SAM application.

Code Coverage Files

Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

Browse

samt1.zip ×
samt2.zip ×
samt3.zip ×
samt4.zip ×
samt5.zip ×
samt6.zip ×
samt7.zip ×

Enable headless collection support

How many days' data do you want to keep in the data warehouse?

All Days

Cancel

Create

8. Click **Create** to create this data provider. The **Data Providers** page is displayed with the **SAM Manual Builds** data provider added.

Data Providers

Search by name or description

Filters

Create Data Provider

Name	Description	Type	Last Collection Date	Actions
SAM Manual Builds	Manual builds for SAM application.	Manual Builds	Unknown	

9. Optional: If you want to edit the Manual Builds data provider that you created, complete the following steps to update the data provider details.
 - a. Click the overflow menu (vertical ellipsis) icon on the **SAM Manual Builds** data provider row. An options menu opens.

SAM Manual Builds

Manual builds for SAM application.

Manual Builds

9/12/20, 3:10 PM

- Edit
- Delete
- Add Build
- View Builds
- Download

- b. Select **Edit** to edit the Manual Builds data provider information. The **Edit Data Provider** page is displayed.
- c. On the **Edit Data Provider** page, you can update the details of the Manual Builds data provider.

Edit Data Provider

Name *

SAM Manual Builds

Description

Manual builds for SAM application.

Selected Data Provider Type

Manual Builds

Enable headless collection support

How many days' data do you want to keep in the data warehouse?

All Days

Cancel Save

d. Click **Save** to save your changes. You can also click **Cancel** to quite editing.

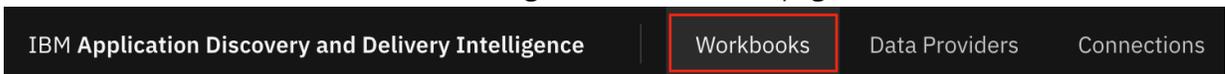
As the ADDI Extensions administrator Alvin, you have now successfully created the **SAM Manual Builds** data provider. Next you will create a workbook to represent a new application that the team can use for code coverage analysis.

Defining a new application as an analysis workbook

In ADDI Extensions, a workbook is used for grouping the same or disparate artifacts for analysis. In this lab, the workbook is used to represent an application. You will define a new application as a workbook and then associate the workbook with the **SAM Manual Builds** data provider that stores the code coverage results of that application.

Complete the following steps to define a new application as a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.



2. Click **Create Workbook**. The **Create Workbook** page is displayed.

Create Workbook

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

Name *

Description

Permissions

Search for User Groups

User	Actions
No users have access to this workbook. Add Permissions.	

Data Providers

3. Complete the **Create Workbook** page with the following information.

- **Name:** SAM Application
- **Description:** This is an analysis workbook for SAM application.

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

Name *

Description

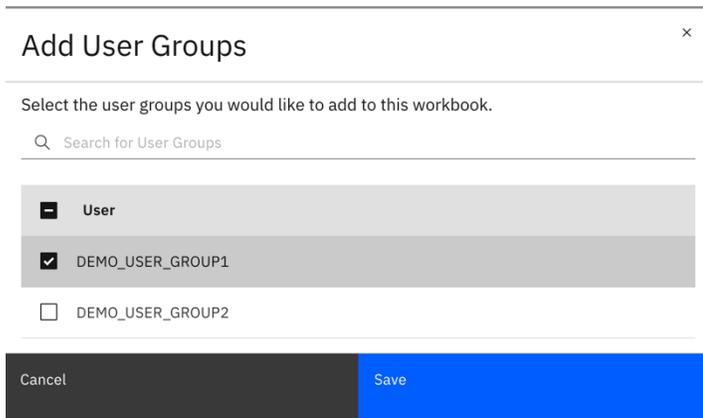
- **Permissions:** Add the user group to the workbook.
 - a. Select **Add** under the **Permissions** section. The Add User Groups dialog box opens.

Permissions

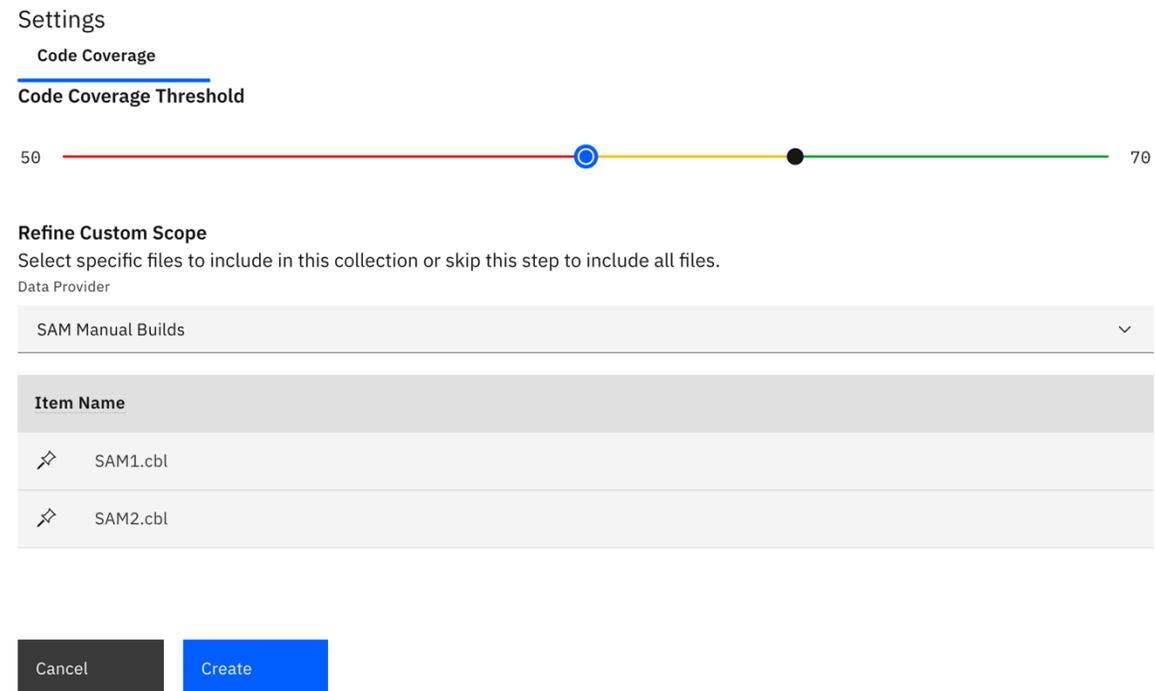
Search for User Groups

User	Actions
No users have access to this workbook. Add Permissions.	

- b. Select the **DEMO_USER_GROUP1** checkbox.
- c. Click **Save** to add the user group to the workbook.



- **Providers:** Select **SAM Manual Builds**. The **Code Coverage Data Provider Settings** section is displayed.

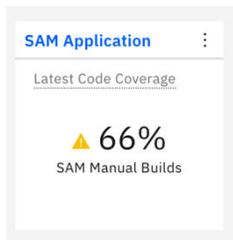


- **Code Coverage Threshold:** Move the left slider to 50% and the right slider to 70%.
- **Refine Custom Scope:** You will see the source files that are loaded to the **SAM Manual Builds** data provider. Those source files, SAM1.cb1 and SAM2.cb1 in this case, are tested as part of the code coverage results. For this tutorial, you will not define the custom scope for the SAM Application.

Notes:

- By default, if you do not select any files, ADDI Extensions includes all files for analysis.
- If you want to define your own scope for this analysis workbook, select the Pin icon () in front of the item name to include the file in the scope.

4. Click **Create** to add this analysis workbook. The Workbooks page is displayed with the **SAM Application** workbook that you just created.



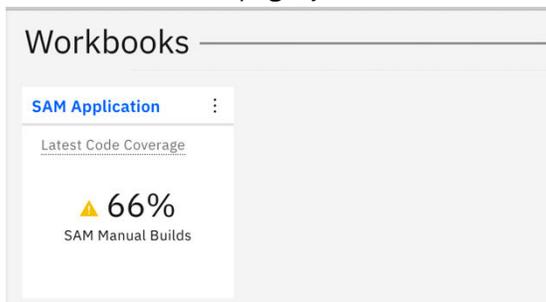
Now you have created the analysis workbook. You can notice that the code coverage results for the first build of SAM Application is displayed on the workbook that you created.

Viewing the code coverage analysis results of the first build

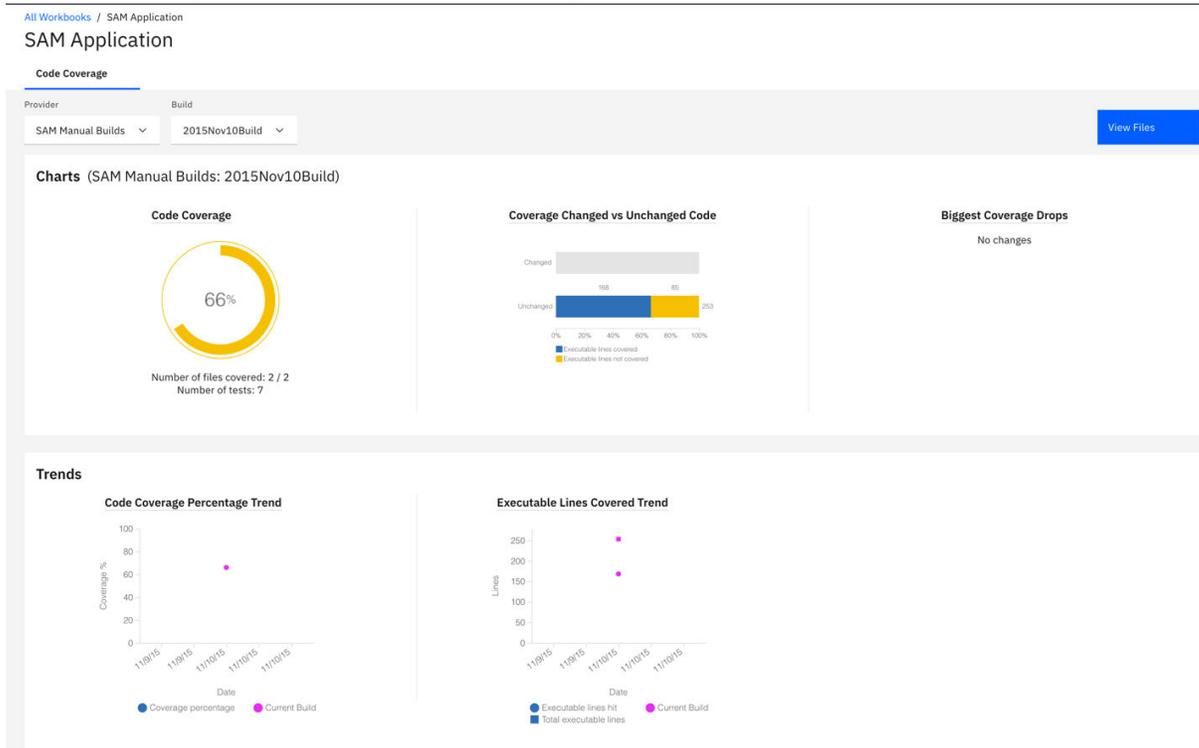
In this lab, you can view the analysis of code coverage results of the first build.

Complete the following steps to view the analysis of code coverage results for **2015Nov10Build** from the Workbooks page.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. On the **Workbooks** page, you can see that **SAM Application** workbook has only 66% code coverage.



3. Click the name of **SAM Application** workbook to view the detailed code coverage reports. The **SAM Application** page is displayed with code coverage summary charts.



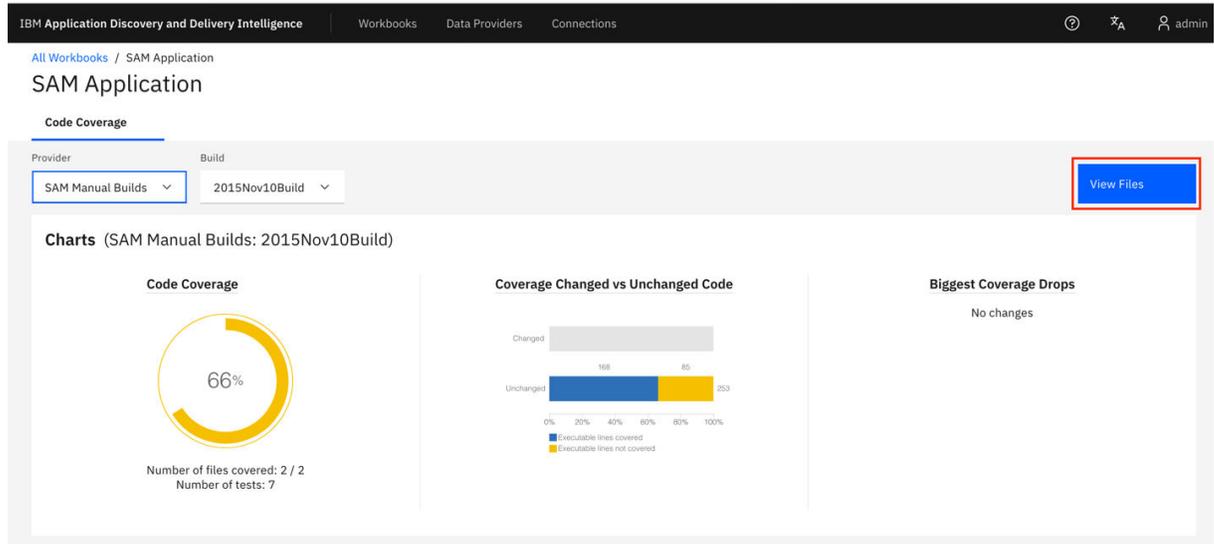
On the **SAM Application** page, you can see the analysis of **2015Nov10Build** in the **Charts** section. The first chart is the latest code coverage result of **SAM Manual Builds** that shows 66%. Next to

the **Code Coverage** report are the comparison of code coverage for the changed and unchanged executable lines of code. You can see that among the total of 253 executable lines of code in this build, 168 lines are tested and 85 lines are not covered by the tests. The **Biggest Coverage Drops** report shows no data since there is only one build for SAM application.

The **Trends** section shows the historical trends from all builds. However, since there is only one build, you can only see one data point on the charts.

For each report, you can hover over on the name of the report to see the report information.

4. Click the **View Files** button on the upper right corner of the header to view detailed analysis.



5. Observe the **Build Analysis** page that is displayed. On the header section, you can see the summary for the entire build. The current build is in the yellow state with 66% code coverage that shows only 168 executable lines of code are covered by the tests. Next to the code coverage percentage is the percentage of code coverage changes comparing to the previous build. Since this is the first build, there is no percentage of code coverage changes yet. It is for the same reason that the number of modified files and new files are zero. For **2015Nov10Build**, you can see that 7 tests are tested against this build. ADDI Extensions analyzes that you can run only two tests to get the same code coverage results.



On the table section, you can see the code coverage analysis details of all the files within the build. SAM2 . cb1 has a red warning icon because SAM2 . cb1 has only 40% code coverage, which is lower than 50% threshold that you set earlier in the lab of [“Defining a new application as an analysis workbook”](#) on page 71.

Item Name	Warnings	Code Coverage	Executable Lines		Tests	
			Covered	Total	Historical	Minimal
> SAM1.cb1		78%	137	176	7	2
> SAM2.cb1	⚠	40%	31	77	3	1

6. Click the name of SAM2 . cb1 to view the list of Historical Tests to Run and Minimal Tests to Run. ADDI Extensions recommends that for SAM2 . cb1 you can just run SAMT3 test to get 40% code coverage instead of running all the historical tests.

SAM2.cb1		40%	31	77	3	1
Historical Tests to Run: 3		Minimal Tests to Run: 1				
SAMT1		SAMT3				
SAMT3						
SAMT4						

From the detailed code coverage analysis, you can see that only 31 executable lines are tested among the 77 executable lines. SAMT1, SAMT3, and SAMT4 are three test cases ran against SAM2 . cb1. ADDI Extensions suggested SAMT3 test case as minimal tests to run. This means that your test cases are not effective. SAMT3 covers the same executable lines that are covered in SAMT1, SAMT3, and SAMT4 together.

- Click the name of SAM2 . cb1 again to hide the information section.
- Select the **Expand** icon (>) in front of the SAM2 . cb1 file to view the code coverage analysis result of the flowpoints within SAM2 . cb1. You can see that **100-VALIDAT-TRAN-ACTION** and **200-PROCESS-TRAN** are not properly tested. These two flowpoints cause insufficient code coverage for SAM2 . cb1.

SAM2.cb1		40%	31	77	3	1
000-MAIN	75%	6	8	3	1	
100-VALIDATE-TRAN	28%	7	25	3	1	
200-PROCESS-TRAN	0%	0	24	0	0	
300-PROCESS-CPU-CRUNCH	100%	2	2	3	1	
310-CRUNCH-LOOP	89%	8	9	3	1	
110-VALIDATE-TRAN-ACTION	88%	7	8	3	1	

You can also notice that all three tests are testing exactly the same set of flowpoints, which means that the test cases are not all effective. It is suggested that you should pay attention to these test cases in the future tests.

- Click the **Expand** icon in front of the SAM2 . cb1 file again to close the flowpoint analysis section.
- Click the **Tests** tab on the header area.

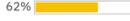
All Workbooks / SAM Application / Build Analysis

Build Analysis

Files Pinned Files **Tests** File Trends

Provider: SAM Manual Builds Build: 2015Nov10Build Filters: Select a Filter

On the **Tests** tab, the view shows all the tests which are executed against the entire build and their code coverage details. You can notice the code coverage details of SAMT1, SAMT3 and SAMT4 which you observed in the previous steps. You can see that SAMT1, SAMT3 and SAMT4 have almost exactly the same detailed information, such as number of lines tested, flowpoints tested, and files tested. But SAMT1 has a slightly higher number of lines tested. As a result, SAMT3 and SAMT4 are candidates for retirement.

Item Name	Code Coverage	Executable Lines		Flowpoints		Files		Elapsed Time
		Covered	Total	Covered	Total	Tested	Missing	
SAMT1	66% 	168	253	23	25	2	0	4 sec.
SAMT2	55% 	97	176	13	18	1	0	4 sec.
SAMT3	65% 	165	253	23	25	2	0	4 sec.
SAMT4	65% 	165	253	23	25	2	0	4 sec.
SAMT5	68% 	120	176	16	18	1	0	4 sec.
SAMT6	62% 	109	176	15	18	1	0	4 sec.
SAMT7	60% 	106	176	15	18	1	0	4 sec.

Now you understand the quality issues of the **2015Nov10Build**. You can improve the test quality by increasing the number of line being tested for SAM2 .cb1 for the future builds. Now you have seen how to set up an analysis workbook to analyze code coverage results. Next, you will add a new build to the **SAM Manual Builds** data provider.

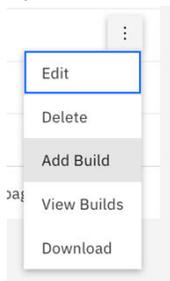
Adding a new build for code coverage analysis

In this lab, you will add a new build with code coverage results to the existing **SAM Manual Builds** data provider.

Assume that you are informed that the new build is available for testing. After you perform the test, you want to add a new build to the **SAM Manual Build** data provider and upload the code coverage for analysis.

To add a new build for code coverage analysis, complete the following steps:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click the overflow menu (vertical ellipsis) icon on the **SAM Manual Builds** data provider row. An options menu opens.



3. Select **Add Builds** from the options menu. The **Add Build** dialog box opens.
4. Complete the **Add Build** dialog box with the following information.
 - **Build Name:** 2015Nov11Build
 - **Date of Build:** 11/11/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
 - **Description:** November 11, 2015 build for SAM application.
 - **Code Coverage Files:** Click **Browse** and then navigate to `<Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > SAM App > November 11 2015` directory. Select multiple files that include `samt1.zip`, `samt2.zip`, `samt3.zip`, `samt4.zip`, `samt5.zip`, `samt6.zip`, and `samt7.zip` at once.

Add Build ×

Build Name *

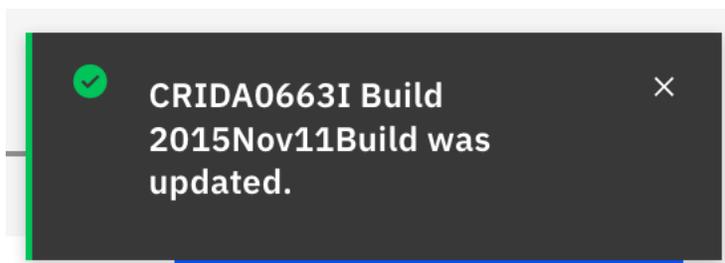
Date Of Build * 📅

Build Description

Code Coverage Files
Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

- samt1.zip ×
- samt2.zip ×
- samt3.zip ×
- samt4.zip ×
- samt5.zip ×
- samt6.zip ×
- samt7.zip ×

5. Click **Add** to add the **2015Nov11Build**.



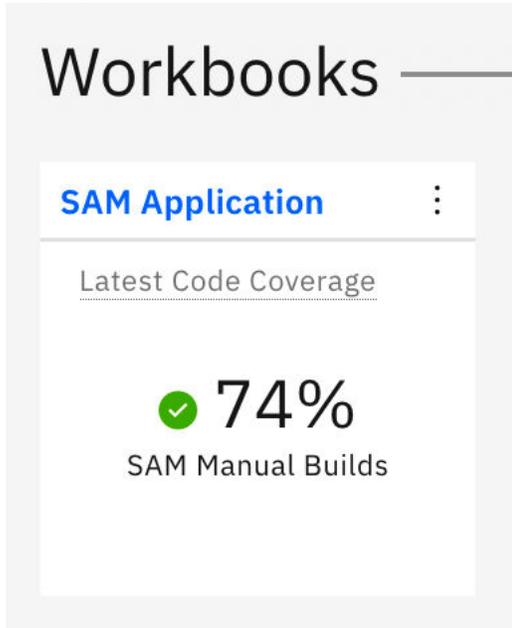
You have now added a new build to **SAM Manual Build**. You are ready to view the new results.

Performing the code coverage analysis

In this lab, you will review the code coverage analysis results of the new build that you added in the previous lab. You can pin the critical files that you need your team to pay attention to when they perform regression tests.

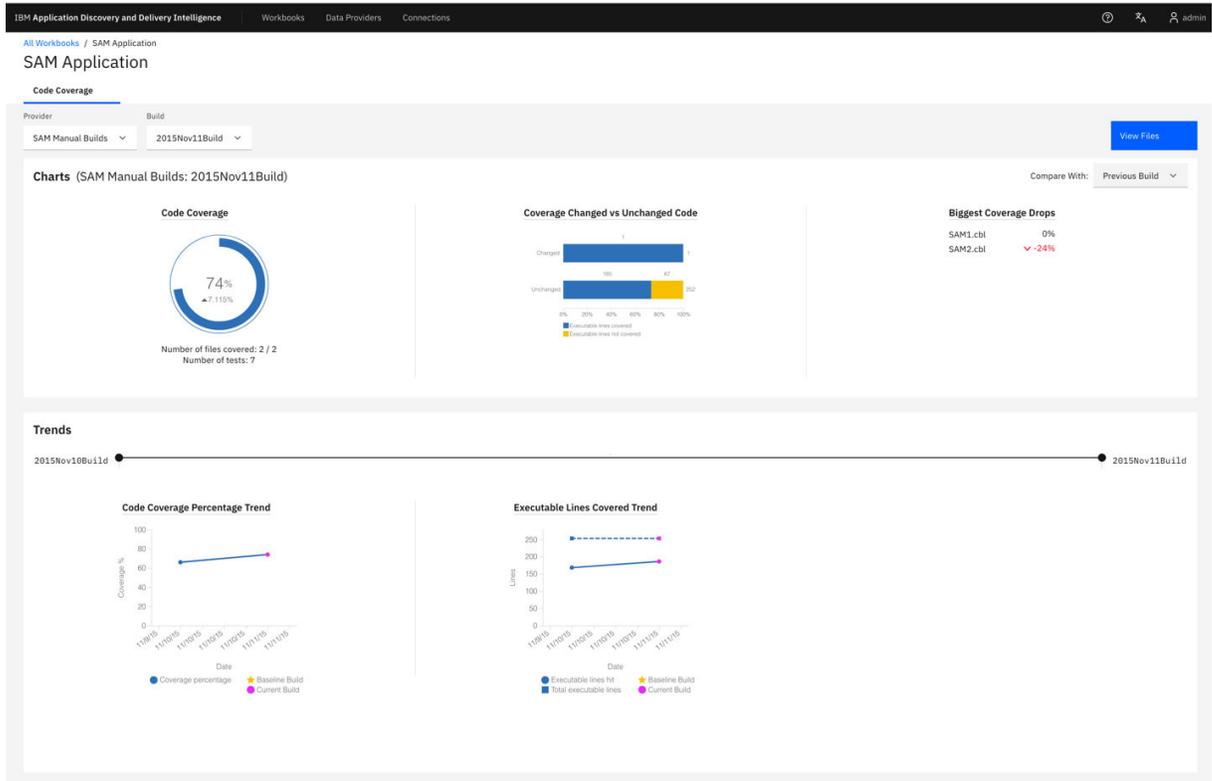
Complete the following steps to perform an analysis on the code coverage results of the new build.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page. The **Workbooks** page is displayed.



You can immediately see that the code coverage of the SAM Application is now improved to the acceptable area at 74%, which is a good sign.

2. Click the header name of **SAM Application** card to view the detailed code coverage report. The SAM Application page is displayed with summary charts.



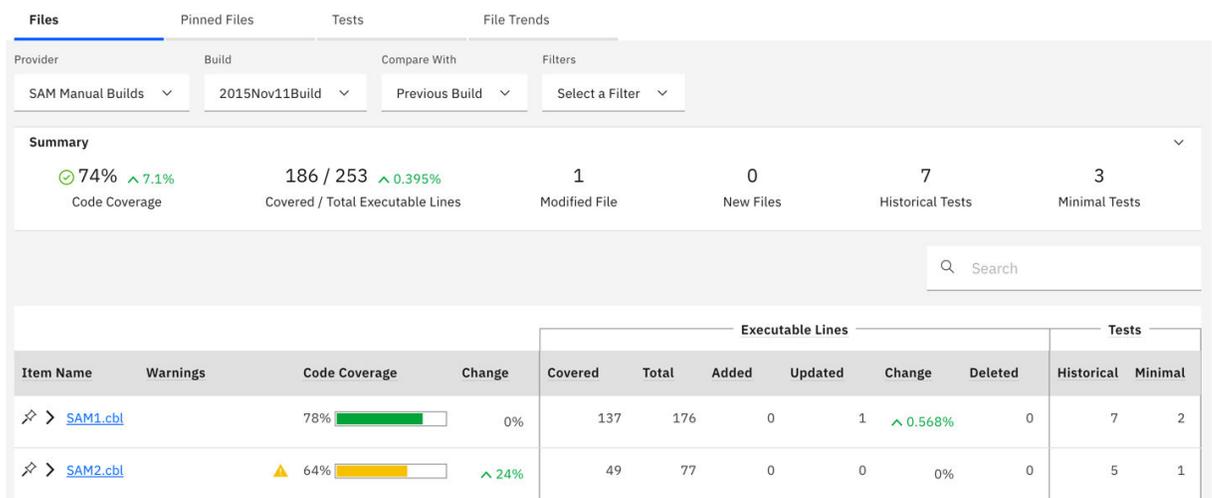
You can see that code coverage of SAM Application is now increased to 74%, up by 7.115%. The Code Coverage Percentage Trend and the Executable Lines Covered Trend are now updated with the data for Nov 10 and Nov 11 builds.

Note: From the two builds, the total executable lines of two builds are the same but the executable lines hit are increased from 168 to 186. This is the reason that the overall code coverage for SAM Application is increased.

3. Click **View Files** on the upper right to view detailed code coverage analysis of SAM Application. The **Build Analysis** page is displayed. You can immediately see that there is no file with a red warning.

[All Workbooks / SAM Application / Build Analysis](#)

Build Analysis



4. Click the **Expand** icon (>) in front of SAM2 . cbl to view the code coverage analysis for SAM2 . cbl at the flowpoint level in the expanded section.

SAM2.cbl	64% ^ 24%	49	77	0	0	0%	0	5	1
000-MAIN	88% ^ 13%	7	8	0	0	0%	0	5	1
100-VALIDATE-TRAN	60% ^ 32%	15	25	0	0	0%	0	5	1
200-PROCESS-TRAN	33% ^ 33%	8	24	0	0	0%	0	5	1
300-PROCESS-CPU-CRUNCH	100% 0%	2	2	0	0	0%	0	5	1
310-CRUNCH-LOOP	89% 0%	8	9	0	0	0%	0	5	1
110-VALIDATE-TRAN-ACTION	100% ^ 12%	8	8	0	0	0%	0	5	1

[Show All 6 Flowpoints](#)

- Click **Show All 6 Flowpoints** to navigate to the **Flowpoints Analysis** view. On the **Flowpoints Analysis** view, you can see the code coverage analysis for each flowpoint. For example, Main is tested 88%, which is increased 13% from the previous run. You can also notice the two flowpoints that had issues previously. Although the code coverage percentages are increased, they are still in the poor and insufficient code coverage threshold.

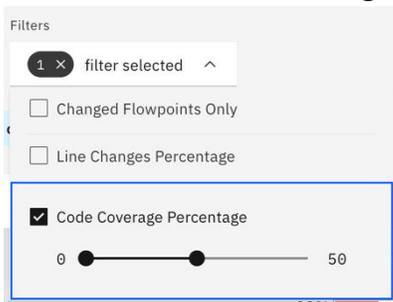
All Workbooks / SAM Application / Build Analysis / Flowpoint Analysis

Flowpoint Analysis

Flowpoint Name	Code Coverage	Change	Executable Lines					Tests		
			Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
000-MAIN	88% ^ 13%		7	8	0	0	0%	0	5	1
100-VALIDATE-TRAN	60% ^ 32%		15	25	0	0	0%	0	5	1
110-VALIDATE-TRAN-ACTION	100% ^ 12%		8	8	0	0	0%	0	5	1
200-PROCESS-TRAN	33% ^ 33%		8	24	0	0	0%	0	5	1
300-PROCESS-CPU-CRUNCH	100% 0%		2	2	0	0	0%	0	5	1
310-CRUNCH-LOOP	89% 0%		8	9	0	0	0%	0	5	1

Items per page: 50 | 1-6 of 6 items | 1 of 1 pages

- Click the **Select a Filter** drop-down box on the top and select the **Code Coverage Percentage** checkbox. The slider bars show the threshold between 0 - 50% appears. This is by default the value of the insufficient code coverage threshold which has been set when you set up the workbook.



- In the **Flowpoints Analysis** view, it now shows only the flowpoint where the code coverage is lower than 50%. You can move the right slider to 70%. The view is refreshed to show the flowpoints where the code coverage is lower than 70%.

Flowpoint Analysis

The screenshot shows a filter dialog box with the following options:

- Changed Flowpoints Only
- Line Changes Percentage
- Code Coverage Percentage

The table below shows the analysis results for two flowpoints:

Flowpoint Name	Code Coverage	Change	Executable Lines						Tests	
			Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
100-VALIDATE-TRAN	60%	↑ 32%	15	25	0	0	0%	0	5	1
200-PROCESS-TRAN	33%	↑ 33%	8	24	0	0	0%	0	5	1

- Click the **Close** icon (X) in the filter dialog message box to clear the filter. All flowpoints are displayed.
- Select **Build Analysis** on the navigation tree to go to the **Build Analysis** view.

[All Workbooks](#) / [SAM Application](#) / [Build Analysis](#) / Flowpoint Analysis

Flowpoint Analysis

- Click the **Pin** icon in front of SAM2 . cb1 name. You decide to pin the SAM2 . cb1 file since this is the file that you would like your team and yourself to pay attention to for the next round of testing for this build.

Item Name	Warnings	Code Coverage	Change	Executable Lines						Tests	
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
> SAM1.cb1		78%	0%	137	176	0	1	↑ 0.568%	0	7	2
> SAM2.cb1		64%	↑ 24%	49	77	0	0	0%	0	5	1

- Click the **Pinned Files** tab to view analysis of pinned file.

Build Analysis

The screenshot shows the Build Analysis view with the following summary information:

- Code Coverage: 74% (↑ 7.1%)
- Covered / Total Executable Lines: 186 / 253 (↑ 0.395%)
- Modified File: 1 / 0
- New Files: 0 / 0
- Historical Tests: 7 / 5
- Minimal Tests: 3 / 1

The table below shows the analysis results for the pinned file SAM2.cb1:

Item Name	Warnings	Code Coverage	Change	Executable Lines						Tests	
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
> SAM2.cb1		64%	↑ 24%	49	77	0	0	0%	0	5	1

The updated view shows only the code coverage information of the files that are pinned. The header section shows the code coverage summary of the pinned files along with the code coverage summary for the entire build.

- Under the **Summary** section, click on the right side number of the **Minimal Tests**.

3 | 1
Minimal Tests

The list of the minimal tests file that ADDI Extensions recommends to run for the pinned files appears. This is the list of minimal tests that ADDI Extensions analyzed based on code coverage results. They can yield the same code coverage percentage as if you run all tests. In the real life scenario, when you have many test files but have limited resources to perform the manual tests, ADDI Extensions can help you to reduce time to run all these tests.

Item Name	Code Coverage	Executable Lines		Flowpoints		Files		Elapsed Time
		Covered	Total	Covered	Total	Tested	Missing	
SAMT1	74%	186	253	24	25	2	0	10 sec.

You have now explored how to upload the code coverage results to ADDI Extensions for analysis and how to perform the code coverage analysis based on the code coverage results that you uploaded.

Setting up automated code coverage data collections

This tutorial guides you through how to set up automated code coverage data collections and load the code coverage results for analysis.

ADDI Extensions supports various automation scenarios for code coverage data collection as below.

- Automated code coverage data collection utilizing the headless code coverage collector daemon program of RDz (IBM Rational® Developer for z Systems, 9.5). ADDI Extensions ships the headless collector and allows debuggers and code coverage engines to send code coverage data directly over the network so that ADDI Extensions can collect the code coverage data automatically.
- Automated code coverage data collection from RTC (IBM Rational Team Concert 6.0 or higher) Build results. ADDI Extensions supports connecting to RTC servers (6.0 or higher) and accessing RTC build results that have code coverage data attached. This capability allows you to download these coverage files and add them automatically to ADDI Extensions.

In this tutorial, you can explore one of the automation scenarios that ADDI Extensions supports. That is, the automated code coverage data collection utilizing the headless code coverage daemon program of RDz 9.5. You will add a new application in order to perform the code coverage analysis. This task is similar to [“Setting up and analyzing code coverage for the Manual Builds data provider”](#) on page 67. However, in this tutorial, when you create the data provider, you enable the headless collector for ADDI Extensions to automatically collect the code coverage data.

Prerequisites

Before you begin this tutorial, you should complete the following activities.

- [“Installing and setting up ADDI Extensions”](#) on page 5

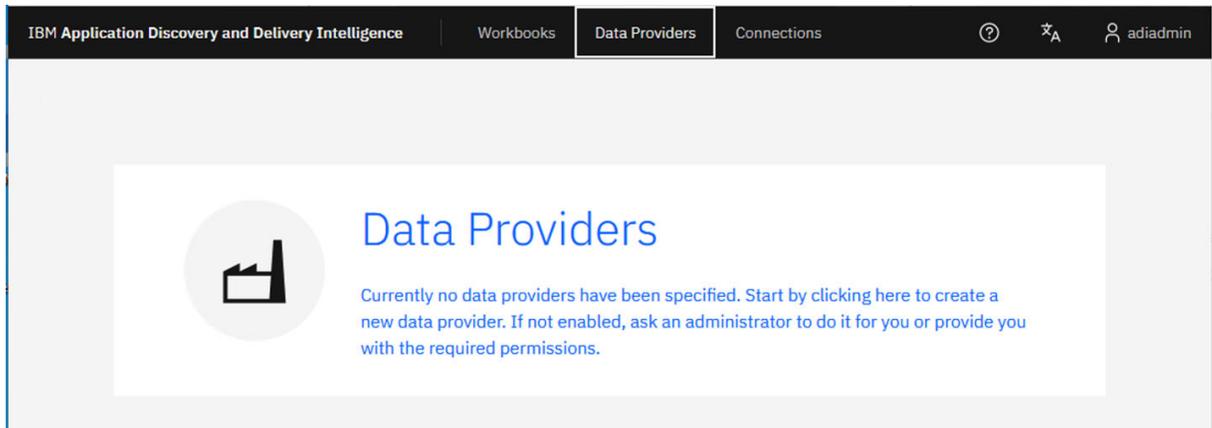
Adding a Manual Builds data provider to collect code coverage data automatically

In this section, assume that you want to start monitoring code coverage for a new application called Health Care application. You want the code coverage data to be automatically collected by ADDI Extensions. So, you create a new data provider to collect the data for the Health Care application.

Complete the following steps to add a Manual Builds data provider to collect code coverage data automatically.

1. Navigate your browser to <https://healthcare.example.com:9753/addi/web/projects/HealthCare4All>.
2. Log in with AdiAdmin as the user ID and AdiAdmin as the password.

3. Select the **Data Providers** tab on the header to go to the **Data Providers** page. If you didn't create any data providers before, a message is displayed to indicate that no data providers have been created.



4. Click the **Data Providers** header on the message to create your first data provider or click the **Create Data Provider** button if this is not your first data provider. The **Create Data Provider** page is displayed.

[All Data Providers](#) / [Create Data Provider](#)

Create Data Provider

Name *

Description

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.

Application Discovery	Business Rule Discovery	Manual Builds	Rational Team Concert Builds
System Management Facility			

Cancel Create

5. Enter the following information in the **Create Data Provider** page.

- **Name:** Health Care Manual Builds
- **Description:** Manual build with automated data collection for Health Care application.

Name *

Health Care Manual Builds

Description

Manual build with automated data collection for Health Care application.

6. Select the **Manual Builds** data provider type.

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.

The screenshot shows a dialog box titled 'Select a Data Provider Type'. It contains five buttons arranged in two rows. The first row has 'Application Discovery', 'Business Rule Discovery', 'Manual Builds', and 'Rational Team Concert Builds'. The second row has 'System Management Facility'. The 'Manual Builds' button is highlighted with a blue border and has a small checkmark icon in its top right corner.

7. Complete the First Build section with the following information.

- **Build Name:** 2016Feb2Build
- **Date of Build:** 2/2/2016, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
- **Description:** February 2, 2016 build for Health Care application.
- **Code Coverage Files:** You will not load any code coverage files at this moment.
- **Enable headless collection support:** Select the checkbox.
- **Collection Trigger:** Select **Manual**.

Note: In the real-life scenarios, instead of manually triggering the data collection, you would set the interval time for ADDI Extensions to collect data automatically.

This **Collection Trigger** section is to set how the code coverage data is to be collected. It can be on demand by selecting **Manual** or on schedule by selecting **Automatic** and specifying the collection interval in hours. For the objective of demonstration in this tutorial, you do on-demand collection in order to see the code coverage data being loaded immediately.

- Select **All Days** for the number of days to keep data in the data warehouse. You can choose among **All Days** to keep the data forever, **90 Days**, **120 Days**, **180 Days**, and **360 Days**.

First Build

Build Name *

2016Feb2Build

Date Of Build *

2/2/2016, 12:00 AM.

Build Description

February 2, 2016 build for Health Care application.

Code Coverage Files

Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

Browse

Enable headless collection support

Collection Trigger

Manual

How many days' data do you want to keep in the data warehouse?

All Days

Cancel

Create

- Click **Create** to create this data provider. The **Data Providers** page is displayed with the **Health Care Manual Builds** data provider added.

The screenshot shows the 'Data Providers' page with a search bar, filters, and a table. The table has columns for Name, Description, Type, Last Collection Date, and Actions. One row is visible with the name 'Health Care Manual Builds' and a description 'Manual build with automated data collection for Health Care application.' The type is 'Manual Builds' and the last collection date is 'Unknown'. There is a vertical ellipsis icon in the Actions column. Below the table, there are controls for 'Items per page' (set to 10), '1-1 of 1 items', and '1 of 1 pages' with navigation arrows.

Name	Description	Type	Last Collection Date	Actions
Health Care Manual Builds	Manual build with automated data collection for Health Care application.	Manual Builds	Unknown	⋮

Now you have successfully created the Manual Builds data provider to collect data automatically. Next you will load the code coverage results to the location where ADDI Extensions can collect data automatically. Then you trigger the **Health Care Manual Builds** data provider to collect the code coverage results.

- Click the overflow menu (vertical ellipsis) icon on the **Health Care Manual Builds** data provider row. An options menu opens.

Name	Description	Type	Last Collection Date	Actions
Health Care Manual Builds	Manual build with automated data collection for Health Care application.	Manual Builds	Unknown	<ul style="list-style-type: none"> Edit Delete Start Data Collection Copy Startup Key Add Build View Builds Download

Items per page: 10 1-1 of 1 items 1

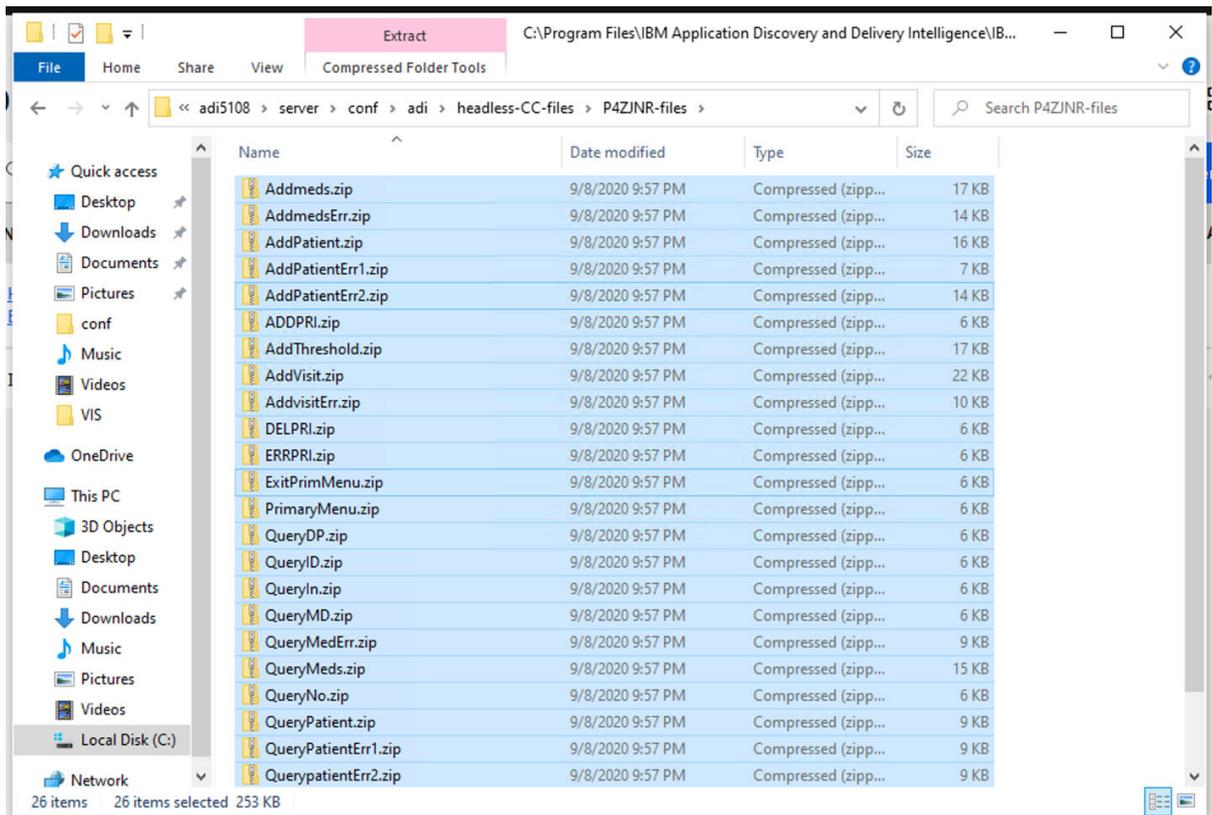
10. Select **Copy Startup Key** from the options menu to copy the startup key to your clipboard.
11. Launch your Notepad program and paste the startup key to the Notepad. You will see the text that is similar to the following sample. Remember the provider ID. In the following sample, the providerid is P4ZJNR.

```

*Untitled - Notepad
File Edit Format View Help
TEST(,,TCPIP&127.0.0.1%8005:*)
ENVAR("EQA_STARTUP_KEY=CC,,providerid=P4ZJNR,testid=<fill in your test-id>")

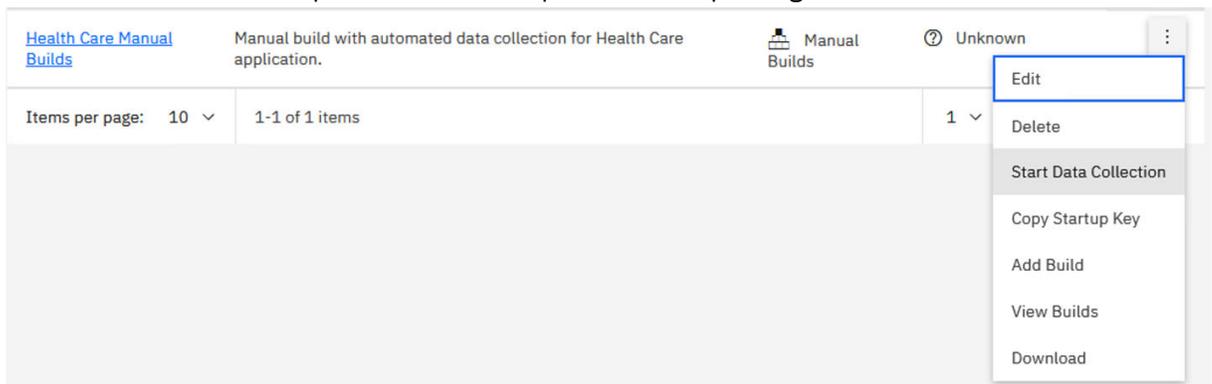
```

12. Browse to the sample data folder in the <addi_installed_directory> IBM Application Discovery and Delivery Intelligence Extensions > adi611 > examples> cobol-coverage > zMobile Health App > Build1 directory. Select all the zip files under Build1 folder and copy them to the clipboard.
13. Browse to the ADDI Extensions location and navigate to <addi_installed_directory> IBM Application Discovery and Delivery Intelligence Extensions > adi611 > server > conf > adi > headless-CC-filesdirectory.
14. Open the folder that has the same name as the providerid. From this example it is the P4ZJNR-files folder. Paste all the zip files that you copied to this folder.



In your typical scenario, you want the code coverage results to be loaded into this location automatically, which depends on the programming languages and tools you are using to generate code coverage data. For more information, see [Chapter 7, “Preparing external data sources,”](#) on page 167.

- Go back to the **Data Providers** page and click the overflow menu (vertical ellipsis) icon on the **Health Care Manual Builds** data provider row. The options menu opens again.



- Select **Start Data Collection** to trigger ADDI Extensions to collect the code coverage data. You can see that in the **Last Collection Date** column, the data collection progress is started. When the data collection is done, you will see the success status and the collection date that is set to current date and time.

Last Collection Date

 Collection started

Last Collection Date

 9/14/20, 9:24 AM

- Click the overflow menu (vertical ellipsis) icon on the **Health Care Manual Builds** data provider card. Then, select **View Builds** from the options menu that opens. You can now see that the code coverage percentage of this build is 66%.

Name	Date of Build	Code Coverage	Actions
☆ 2016Feb2Build	Feb 2, 2016, 12:00:00 AM	66%	 

Items per page: 10 | 1-1 of 1 items | 1 of 1 pages

You have now successfully loaded the code coverage results to the **Health Care Manual Builds** data provider. Next, you will create a workbook and associate it to this data provider to view the results of code coverage analysis.

Setting up a workbook to view code coverage results from automated data collection

In this lab, you will create a workbook and associate it with the **Health Care Manual Builds** data provider to view the code coverage results that are automatically collected.

Complete the following steps to set up the workbook.

- Select the **Workbooks** tab on the header to go to the **Workbooks** page.



- Click **Create Workbook**. The **Create Workbook** page is displayed.

[All Workbooks](#) / [Create Workbook](#)

Create Workbook

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

Name *

Description

Permissions

Search for User Groups

User	Actions
No users have access to this workbook. Add Permissions.	

Data Providers

3. Complete the **Create Workbook** page with the following information.

- **Name:** zMobile Health Care
- **Description:** This is an analysis workbook for the core Health Care application..

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

Name *

zMobile Health Care

Description

This is an analysis workbook for the core Health Care application.

- **Permissions:** Do not set up any user permissions.
- **Providers:** Select **Health Care Manual Builds**. The **Code Coverage Data Provider Settings** section is displayed.

Data Providers

Search by name or description Filters ▾

<input checked="" type="checkbox"/>	Name	Description	Type	Last Collection Date
<input checked="" type="checkbox"/>	Health Care Manual Builds	Manual build with automated data collection for Health Care application.	Manual Builds	9/14/20, 9:24 AM

Items per page: 10 ▾ 1-1 of 1 items 1 ▾ of 1 pages

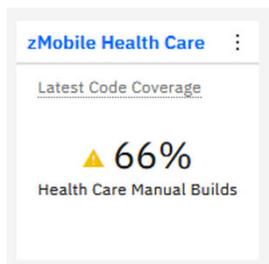
Settings

Code Coverage

Code Coverage Threshold



- **Code Coverage Threshold:** Move the left slider to 60% and the right slider to 80%.
 - **Refine Custom Scope:** Leave it as default, which means that all the files are in the analysis scope.
4. Click **Create** to create this **zMobile Health Care** workbook. The **Workbooks** page is displayed with the **zMobile Health Care** that you just created.



The **zMobile Health Care** workbook is successfully created and associated with the **Health Care Manual Builds** data provider. You are ready to view analysis results as described in [“Viewing the code coverage analysis results of the first build”](#) on page 74.

You have explored how to add a new Manual Builds data provider to collect the code coverage data automatically. This scenario reduces the manual steps that you have to complete for ADDI Extensions to

perform code coverage data analysis. As mentioned in the beginning of “[Setting up automated code coverage data collections](#)” on page 83, ADDI Extensions also includes the capability of automated code coverage data collection from an RTC build. Due to the lack of RTC hosted environment, you are not able to explore this scenario in this tutorial for now. If you are interested in the ADDI Extensions capability of automated data collection from an RTC build and already have an RTC server hosted with code coverage data, you can find more RTC specific information in the following sections: [Chapter 8, “Managing connections,”](#) on page 197, [Chapter 9, “Managing data providers,”](#) on page 199, and [Chapter 10, “Managing workbooks,”](#) on page 217.

Exercising setting up a Manual Builds data provider for code coverage analysis

This tutorial guides you how to set up a Manual Builds data provider for code coverage analysis of Perfect Calculator application and how to review the code coverage reports and dashboards.

Before you begin this tutorial, you need to complete the following task.

- “[Installing and setting up ADDI Extensions](#)” on page 5

Setting up a Manual Builds data provider

This section guides you how to set up a Manual Builds data provider for code coverage analysis of Perfect Calculator application.

Complete the following steps to set up a Manual Builds data provider for Perfect Calculator application.

1. Navigate your browser to <https://healthcare.example.com:9753/addi/web/projects/HealthCare4A11>.
2. Log in as the ADDI Extensions administrator Alvin with AdiAdmin as the user ID and AdiAdmin as the password. The ADDI Extensions home page is displayed.
3. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
4. Click **Create Data Provider**. The **Create Data Provider** page is displayed.

[All Data Providers](#) / [Create Data Provider](#)

Create Data Provider

Name *

Description

Description

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.

Application Discovery

Business Rule Discovery

Manual Builds

Rational Team Concert Builds

System Management Facility

Cancel Create

5. Enter the following information in the **Create Data Provider** page.

- **Name:** Perfect Calculator Manual Builds
- **Description:** Manual builds for Perfect Calculator application.

Name *

Health Care Manual Builds

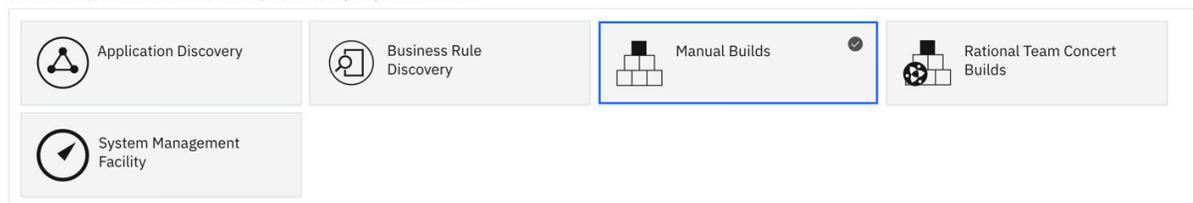
Description

Manual build with automated data collection for Health Care application.

6. Select the **Manual Builds** data provider type.

Select a Data Provider Type

Select a data provider kind to define a new provider and specify data to collect.



7. Complete the First Build section with the following information.

- **Build Name:** 2015Nov10Build
- **Date of Build:** 11/10/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
- **Description:** November 10, 2015 build for Perfect Calculator application.
- **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > Perfcalc App > November 10 2015 directory. Select all the zip files in the November 10 2015 folder, that is, Perfcalct1.zip, Perfcalct2.zip, Perfcalct3.zip, and Perfcalct4.zip.
- **Enable headless collection support:** Make sure that the checkbox is not selected.
- Select **All Days** for the number of days to keep data in the data warehouse.

First Build

Build Name *

2015Nov10Build

Date Of Build *

11/10/2015, 12:00 AM.

Build Description

November 10, 2015 build for Perfect Calculator application.

Code Coverage Files

Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

Browse

Perfcalct1.zip

×

Perfcalct2.zip

×

Perfcalct3.zip

×

Perfcalct4.zip

×

Enable headless collection support

How many days' data do you want to keep in the data warehouse?

All Days

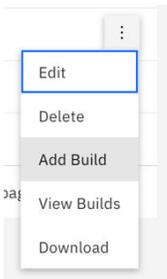
Cancel

Create

- Click **Create** to create this data provider. The **Data Providers** page is displayed with the **Perfect Calculator Manual Builds** data provider added.

Name	Description	Type	Last Collection Date	Actions
Perfect Calculator Manual Builds	Manual builds for Perfect Calculator application	Manual Builds	Unknown	

- Click the overflow menu (vertical ellipsis) icon on the **Perfect Calculator Manual Builds** data provider row. An options menu opens.



- Select **Add Builds** from the options menu to add builds. The **Add Build** dialog box opens.

- Complete the **Add Build** dialog box with the following information.

- **Build Name:** 2015Nov11Build
- **Date of Build:** 11/11/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
- **Description:** November 11, 2015 build for Perfect Calculator application.

- **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > Perfcalc App > November 11 2015 directory. Select all the zip files in the November 11 2015 folder, that is, Perfcalct1.zip, Perfcalct2.zip, Perfcalct3.zip, and Perfcalct4.zip.
12. Click **Add** to add this build. The status message is displayed to indicate that the build is added.
 13. Repeat step 9 - step 12 to add another build with the following information.
 - a. **Build Name:** 2015Nov12Build
 - b. **Date of Build:** 11/12/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
 - c. **Description:** November 12, 2015 build for Perfect Calculator application.
 - d. **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > Perfcalc App > November 12 2015 directory. Select all the zip files in the November 12 2015 folder, that is, Perfcalct1.zip, Perfcalct2.zip, Perfcalct3.zip, and Perfcalct4.zip.
 14. Repeat step 9 - step 12 to add another build with the following information.
 - a. **Build Name:** 2015Nov13Build
 - b. **Date of Build:** 11/13/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
 - c. **Description:** November 13, 2015 build for Perfect Calculator application.
 - d. **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > Perfcalc App > November 13 2015 directory. Select all the zip files in the November 13 2015 folder, that is, Perfcalct1.zip, Perfcalct2.zip, Perfcalct3.zip, and Perfcalct4.zip.
 15. Repeat step 9 - step 12 to add another build with the following information.
 - a. **Build Name:** 2015Nov17Build
 - b. **Date of Build:** 11/17/2015, 12:00 AM. You can select the date from the drop-down calendar to complete the **Date of Build** field.
 - c. **Description:** November 17, 2015 build for Perfect Calculator application.
 - d. **Code Coverage Files:** Click **Browse** and then navigate to <Installed ADDI Extensions folder> > adi611 > examples> cobol-coverage > Perfcalc App > November 17 2015 directory. Select all the zip files in the November 17 2015 folder, that is, Perfcalct1.zip, Perfcalct2.zip, Perfcalct3.zip, and Perfcalct4.zip.
 16. Click the overflow menu (vertical ellipsis) icon on the **Perfect Calculator Manual Builds** data provider row. An options menu opens.
 17. Select **View Builds** from the options menu. You can see 5 builds on the **Builds** page.

Perfect Calculator Manual Builds

Builds			
Q Search		Show all builds ▾	↻ Add
Name	Date of Build	Code Coverage	Actions
▾ ☆ 2015Nov17Build	Nov 17, 2015, 12:00:00 AM	75%	✎ 🗑
▾ ☆ 2015Nov13Build	Nov 13, 2015, 12:00:00 AM	86%	✎ 🗑
▾ ☆ 2015Nov12Build	Nov 12, 2015, 12:00:00 AM	100%	✎ 🗑
▾ ☆ 2015Nov11Build	Nov 11, 2015, 12:00:00 AM	100%	✎ 🗑
▾ ☆ 2015Nov10Build	Nov 10, 2015, 12:00:00 AM	84%	✎ 🗑

Items per page: 10 ▾ 1-5 of 5 items 1 ▾ of 1 pages ◀ ▶

Cancel Save baseline builds

You have now finished setting up of the **Perfect Calculator Manual Builds** data provider. Next you will set up a workbook for the **Perfect Calculator** application.

Setting up a workbook

This lab guides you how to set up workbook for code coverage analysis of Perfect Calculator application. Complete the following steps to set up a workbook for Perfect Calculator application.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.



2. Click **Create Workbook**. The **Create Workbook** page is displayed.

All Workbooks / Create Workbook
Create Workbook

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

Name *

Description

Permissions

Q Search for User Groups Add

User	Actions
No users have access to this workbook. Add Permissions.	

Data Providers

3. Complete the **Create Workbook** page with the following information.

- **Name:** Perfect Calculator
- **Description:** This is an analysis workbook for Perfect Calculator application.

Define a Workbook that groups all data for one or more providers and defines which team members have access to this data.

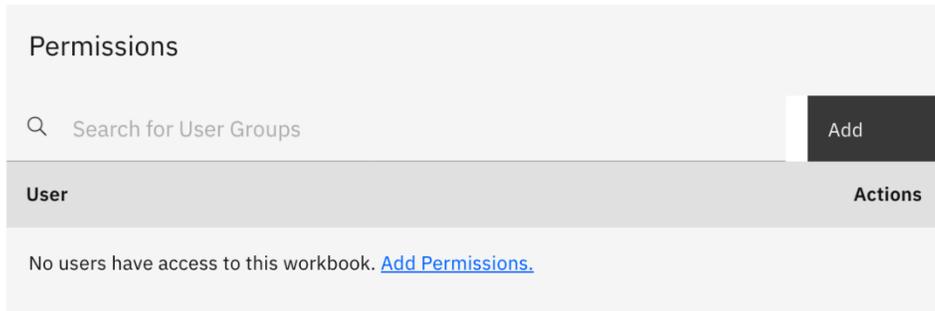
Name *

Perfect Calculator

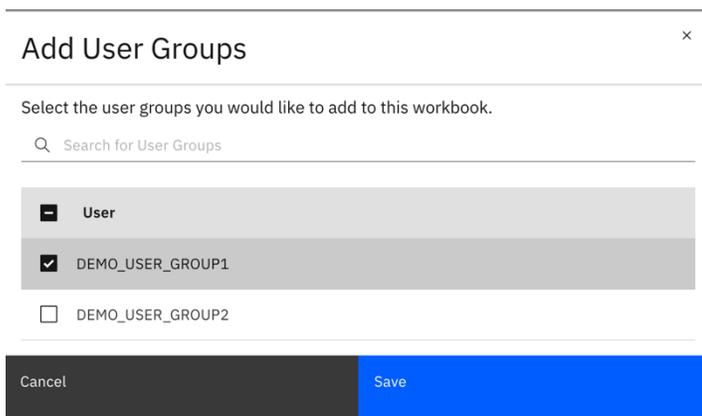
Description

This is an analysis workbook for Perfect Calculator application.

- **Permissions:** Add the user group to the workbook.
 - a. Select **Add** under the **Permissions** section. The Add User Groups dialog box opens.



- b. Select the **DEMO_USER_GROUP1** checkbox.
- c. Click **Save** to add the user group to the workbook.



- **Providers:** Select **Perfect Calculator Manual Builds**. The **Code Coverage Data Provider Settings** section is displayed.
 - **Code Coverage Threshold:** Move the left slider to 50% and the right slider to 70%.

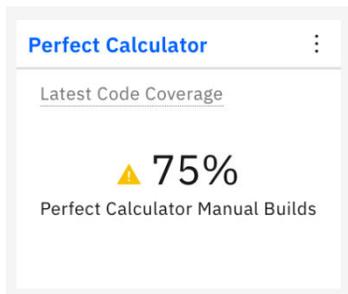
Settings

Code Coverage

Code Coverage Threshold



4. Click **Create** to add this analysis workbook. The Workbooks page is displayed with the **Perfect Calculator** workbook that you just created.



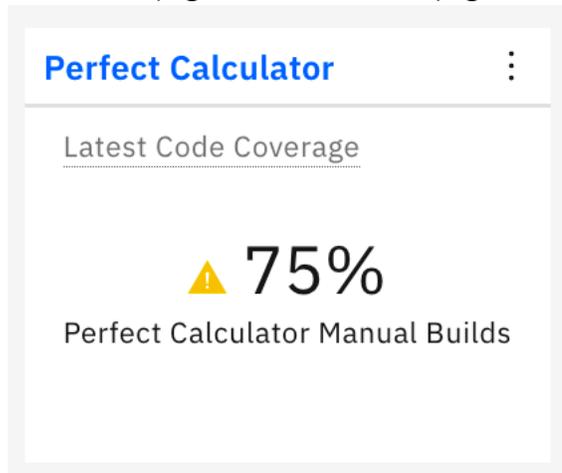
You have completed the setup of the **Perfect Calculator** workbook. You are now ready to analyze code coverage data for Perfect Calculator application.

Reviewing code coverage reports and dashboards

This lab guides you how to review code coverage reports and dashboards.

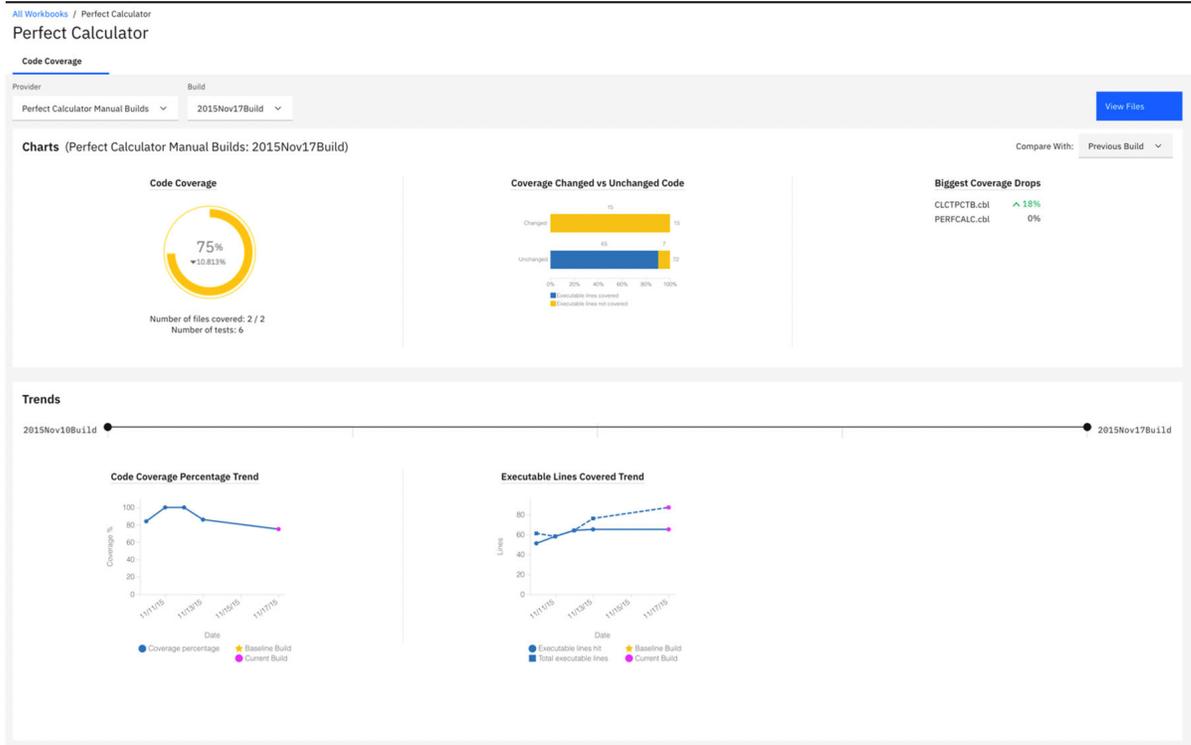
Complete the following steps to examine the code coverage reports and dashboards.

1. If you are not yet on the ADDI Extensions home page, launch your browser and go to <https://healthcare.example.com:9753/addi/web/projects/HealthCare4All>.
2. Log in to ADDI Extensions with `AdiAdmin` as the user ID and `AdiAdmin` as the password. The **Workbooks** page which is the homepage for ADDI Extensions is displayed.



You can notice that, **Perfect Calculator** has a 75% code coverage and the status is in warning state. This is because when you were setting up the **Perfect Calculator** workbook, you defined the code coverage threshold for warning state between 50% - 80%.

3. Click the name of **Perfect Calculator** card to deep dive to the code coverage reports of **Perfect Calculator** workbook. The summary charts for **Perfect Calculator** loads.



The reports show the code coverage analysis for the latest build of **Perfect Calculator Manual Builds**. On the top, it shows that the build is **2015Nov17Build**. The first report shows that the **Perfect Calculator** has a 75% Code Coverage which is in the warning area. The code coverage is dropped from the previous build by 10.813%. From the next report, you can see that 15 executable lines of code are modified on this build but none of the lines is tested. There seems to be some issues here. The cause could be that the modified code has not been tested properly.

Under the **Trends** section, the **Code Coverage Percentage Trend** is starting to go down from Nov 12. On the next report, the number of total executable lines in the next report is going up from 64 to 76 on 11/13/2015 and then to 87 on 11/17/2015. However, the executable lines hit remains quite stable. The issue seems to happen since the previous build.

4. Click **View Files** to drill down further.



5. On the top row of the **Build Analysis** view that appears, you can see the summary of the entire build: code coverage percentage, number of modified files, number of new files, number of historical tests files ran against this build, and number of minimal tests to run. You can see the same information as in the **Workbook** view that the code coverage drops by almost 11%.

Build Analysis

Files		Pinned Files		Tests		File Trends					
Provider	Build	Compare With	Filters								
Perfect Calculator Manual Builds	2015Nov17Build	Previous Build	Select a Filter								
Summary											
▲ 75% ▼ -10.8%	65 / 87 ▲ 17.2%	1	0	6	3						
Code Coverage	Covered / Total Executable Lines	Modified File	New Files	Historical Tests	Minimal Tests						
Search											
Item Name	Warnings	Code Coverage	Change	Executable Lines						Tests	
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
CLCTPCTB.cbl	▲	46%	▼ -18%	18	39	11	4	▲ 38.5%	0	3	1
PERFCALC.cbl		98%	0%	47	48	0	0	0%	0	6	3

You can notice that CLCTPCTB.cbl status is in red warning and the percentage of code coverage change goes down by 18%. From the total of 39 executable lines, there are 11 executable lines added and 4 lines updated to the build. You suspect that these added and updated executable lines that didn't get tested causes the code coverage of the file to go down by 18%.

Note: The "Change" of executable lines is the percentage of executable lines that are changed with in a build, which is calculated from the number of total executable lines within a build and the number of lines that are added and updated within a build. The executable lines represent the number of executable lines for a source file or program. The executable line is defined as the line of code that the compiler marks as executable. For COBOL, the executable line may not directly correspond to the exact source line, as COBOL is not debugged by using source but rather the expanded source.

6. Click the **Expand** icon in front of CLCTPCTB.cbl name to expand the flowpoints analysis section.

Item Name	Warnings	Code Coverage	Change	Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
CLCTPCTB.cbl	▲	46%	▼ -18%	18	39	11	4	▲ 38.5%	0	3	1
FT-100-CALCULATE-TOP-VALUES		100%	0%	1	1	0	0	0%	0	3	1
100-CALCULATE-TOP-VALUES		100%	0%	3	3	0	0	0%	0	3	1
200-CALCULATE-LOW-SALES		100%	0%	3	3	0	0	0%	0	3	1
P-OUTPUT-AZUE0000		0%	0%	0	5	0	0	0%	0	0	0
COMBINE-AND-WRITE		0%	0%	0	16	11	4	▲ 94%	0	0	0
Show All 5 Flowpoints											

7. Click **Show All 5 Flowpoints** on the bottom of the expanded section to see the code coverage details of all flowpoints within the file. The **Flowpoints Analysis** view is displayed.

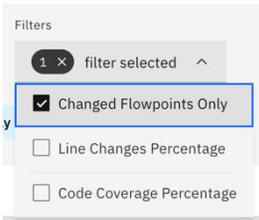
Flowpoint Analysis

Flowpoint Name	Code Coverage	Change	Executable Lines							Tests	
			Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal	
100-CALCULATE-TOP-VALUES	100% <div style="width:100%; height:10px; background-color:green;"></div>	0%	3	3	0	0	0%	0	3	1	
200-CALCULATE-LOW-SALES	100% <div style="width:100%; height:10px; background-color:green;"></div>	0%	3	3	0	0	0%	0	3	1	
COMBINE-AND-WRITE	0% <div style="width:0%; height:10px; background-color:green;"></div>	0%	0	16	11	4	^ 94%	0	0	0	
FT-100-CALCULATE-TOP-VALUES	100% <div style="width:100%; height:10px; background-color:green;"></div>	0%	1	1	0	0	0%	0	3	1	
P-OUTPUT-AZUE0000	0% <div style="width:0%; height:10px; background-color:green;"></div>	0%	0	5	0	0	0%	0	0	0	

Items per page: 50 | 1-5 of 5 items | 1 of 1 pages

Similar to the **Buils Analysis** view, this view shows the code coverage details for all flowpoints within a file. You can see that the code coverage of **COMBINE-AND-WRITE** is showing 0% and 11 executable lines from the total of 16 executable lines are added and 4 executable lines are updated. This confirms your suspicion that the executable lines that are modified within the CLCTPCTB.cbl are not tested.

8. Select the **Filters** drop-down list on the top menu and select the **Changed Flowpoints Only** checkbox to filter the view to show only the flowpoints that are modified.



9. The view updates with only **COMBINE-AND-WRITE** flowpoint which confirms that this is the only flowpoint that is modified and it has not yet been tested.

Flowpoint Analysis

Flowpoint Name	Code Coverage	Change	Executable Lines							Tests	
			Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal	
COMBINE-AND-WRITE	0% <div style="width:0%; height:10px; background-color:green;"></div>	0%	0	16	11	4	^ 94%	0	0	0	

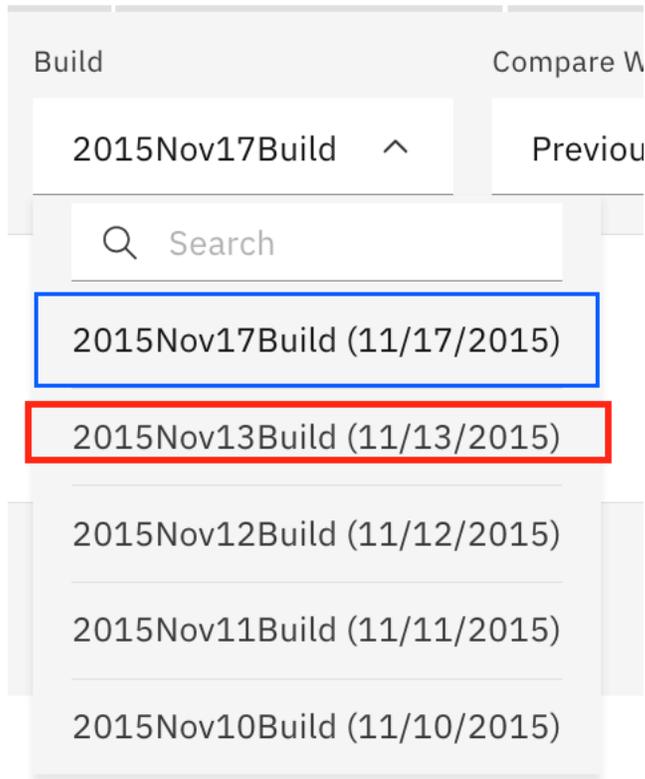
Items per page: 50 | 1-1 of 1 items | 1 of 1 pages

10. Select **Build Analysis** on the navigation tree to go to the **Build Analysis** view.

All Workbooks / SAM Application / **Build Analysis** / Flowpoint Analysis

Flowpoint Analysis

11. Click the **Build** drop-down list on the top menu and select **2015Nov13Build** to go to the build on Nov 13, 2015. You suspect that the code coverage issue started from this build.



On the page for **2015Nov13Build**, you can see that the overall code coverage is dropped by 14.5% comparing with the previous build. For **2015Nov13Build**, the CLCTPCTB.cb1 status is also in warning status for this build and the percentage of code coverage change goes down by 36% comparing with the previous build. 10 executable lines are added to be build. Again, you suspect that these 10 added lines probably were not tested well.

Summary										
86%	-14.5%	65 / 76	15.8%	2	0	6	3			
Code Coverage		Covered / Total Executable Lines		Modified Files	New Files	Historical Tests	Minimal Tests			

Item Name	Warnings	Code Coverage	Change	Executable Lines					Tests		
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
CLCTPCTB.cb1		64%	-36%	18	28	10	0	35.7%	0	3	1
PERFCALC.cb1		98%	-2%	47	48	2	0	4.2%	0	6	3

12. Click the **Build** drop-down list again and select **2015Nov12Build** to go to the build on Nov 12, 2015.

Files Pinned Files Tests File Trends

Provider Perfect Calculator Manual Builds Build 2015Nov12Build Compare With Previous Build Filters Select a Filter

Summary

100% 0% Code Coverage 64 / 64 9.4% Covered / Total Executable Lines 2 Modified Files 0 New Files 6 Historical Tests 3 Minimal Tests

Search

Item Name	Warnings	Code Coverage	Change	Executable Lines					Tests		
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
> CLCTPCTB.cbl		100%	0%	18	18	4	0	22.2%	0	3	1
> PERFCALC.cbl		100%	0%	46	46	2	0	4.3%	0	6	3

On the page for **2015Nov12Build**, the code coverage for overall build indicates 100% code coverage. This build has been fully tested. The issues seem to be arisen starting in **2015Nov13Build**. So you want to compare the builds to get a better understanding of what has changed.

- Click the **File Trends** tab to go to the **File Trends** page to analyze multiple builds at the same time. The **File Trends** page is displayed. By default, **2015Nov12Build** is preselected.

All Workbooks / Perfect Calculator / Build Analysis

Build Analysis

Files Pinned Files Tests File Trends

Select between two and five builds for comparison

Build Name	Build Date	Tag
<input type="checkbox"/> 2015Nov17Build	Nov 17, 2015, 12:00:00 AM	
<input type="checkbox"/> 2015Nov13Build	Nov 13, 2015, 12:00:00 AM	
<input checked="" type="checkbox"/> 2015Nov12Build	Nov 12, 2015, 12:00:00 AM	
<input type="checkbox"/> 2015Nov11Build	Nov 11, 2015, 12:00:00 AM	
<input type="checkbox"/> 2015Nov10Build	Nov 10, 2015, 12:00:00 AM	

Cancel Compare

- Select **2015Nov13Build** and **2015Nov17Build**.

Select between two and five builds for comparison

Build Name	Build Date
<input checked="" type="checkbox"/> 2015Nov17Build	Nov 17, 2015, 12:00:00 AM
<input checked="" type="checkbox"/> 2015Nov13Build	Nov 13, 2015, 12:00:00 AM
<input checked="" type="checkbox"/> 2015Nov12Build	Nov 12, 2015, 12:00:00 AM
<input type="checkbox"/> 2015Nov11Build	Nov 11, 2015, 12:00:00 AM
<input type="checkbox"/> 2015Nov10Build	Nov 10, 2015, 12:00:00 AM

Cancel Compare

- Click **Compare** to compare the three builds. The page of code coverage details for three builds is displayed.

Build Analysis

Files	Pinned Files	Tests	File Trends				
Provider: Perfect Calculator Manual Builds		Filters: Select a Filter					
Summary							
2015Nov17Build (11/17/2015)	75% ▼ -10.8% Code Coverage	65 / 87 ▲ 17.2% Covered / Total Executable Lines	1 Modified File	0 New Files	6 Historical Tests	3 Minimal Tests	
2015Nov13Build (11/13/2015)	86% ▼ -14.5% Code Coverage	65 / 76 ▲ 15.8% Covered / Total Executable Lines	2 Modified Files	0 New Files	6 Historical Tests	3 Minimal Tests	
2015Nov12Build (11/12/2015)	100% ○ 0% Code Coverage	64 / 64 ▲ 9.4% Covered / Total Executable Lines	2 Modified Files	0 New Files	6 Historical Tests	3 Minimal Tests	

Item Name	Code Coverage	Executable Lines		Tests	
		Covered	Total	Historical	Minimal
> CLCTPCTB.cb1	46%	18	39	3	1
> CLCTPCTB.cb1	64%	18	28	3	1
> CLCTPCTB.cb1	100%	18	18	3	1
> PERFCALC.cb1	98%	47	48	6	3
> PERFCALC.cb1	98%	47	48	6	3
> PERFCALC.cb1	100%	46	46	6	3

On this page, you can compare the details of three builds in one view. This view helps confirm that your observation earlier that file **CLCTPCTB.cb1** has brought the code coverage percentage of the entire build down from 100% to 86% and 75%. The cause could be that the executable lines that are added to the **CLCTPCTB.cb1** in the past 2 builds have not been properly tested. You can also see that the code coverage percentage of **PERFCALC.cb1** also drops 2% on 2015Nov13Build. There are 2 execution lines added but one of them seems not tested.

- Click the **Expand** icon (>) in front of **CLCTPCTB.cb1** from all three builds to show the flowpoints analysis.

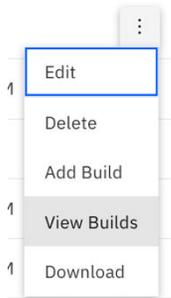
Item Name	Code Coverage	Executable Lines		Tests	
		Covered	Total	Historical	Minimal
▼ CLCTPCTB.cb1	46%	18	39	3	1
FT-100-CALCULATE-TOP-VALUES	100%	1	1	3	1
100-CALCULATE-TOP-VALUES	100%	3	3	3	1
200-CALCULATE-LOW-SALES	100%	3	3	3	1
P-OUTPUT-AZUE0000	0%	0	5	0	0
COMBINE-AND-WRITE	0%	0	16	0	0
Show All 5 Flowpoints					
▼ CLCTPCTB.cb1	64%	18	28	3	1
FT-100-CALCULATE-TOP-VALUES	100%	1	1	3	1
100-CALCULATE-TOP-VALUES	100%	3	3	3	1
200-CALCULATE-LOW-SALES	100%	3	3	3	1
P-OUTPUT-AZUE0000	0%	0	5	0	0
P-INPUT-AZUE0000	0%	0	5	0	0
Show All 5 Flowpoints					
▼ CLCTPCTB.cb1	100%	18	18	3	1
FT-100-CALCULATE-TOP-VALUES	100%	1	1	3	1
100-CALCULATE-TOP-VALUES	100%	3	3	3	1
200-CALCULATE-LOW-SALES	100%	3	3	3	1
Show All 3 Flowpoints					

You can see that for **2015Nov13Build**, P-INPUT-AZUE0000 and P-OUTPUT-AZUE0000 are newly added to the file but they have not been tested. P-INPUT-AZUE0000 is removed from **2015Nov17Build** and COMBINE-AND-WRITE is added to the build. Again the new flowpoint has not been tested either. These 2 flowpoints are areas that require testing.

You are satisfied with your investigation on the code coverage issue for **Perfect Calculator** application. You want to communicate the results to your team to improve the tests to cover P-OUT-AZUE0000 and COMBINE-AND-WRITE flowpoints.

Next, you learn that **2015Nov12Build** is one of the milestone builds and it is fully tested. You want to make sure that in the future your team can use this build as baseline for comparing the code coverage results of the next builds.

17. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
18. On the **Perfect Calculator Manual Builds** data provider, click the overflow menu (vertical ellipsis) icon and then select **View Builds**. The **Builds** page is displayed.



19. Select the **Star** icon in front of the **2015Nov12Build**. The **Star** icon in front of a build name is the indication of a baseline build.

Builds

Search Show all builds ▼ ↻ Add

Name	Date of Build	Code Coverage	Actions
☆ 2015Nov17Build	Nov 17, 2015, 12:00:00 AM	75%	✎ 🗑️
☆ 2015Nov13Build	Nov 13, 2015, 12:00:00 AM	86%	✎ 🗑️
★ 2015Nov12Build	Nov 12, 2015, 12:00:00 AM	100%	✎ 🗑️
☆ 2015Nov11Build	Nov 11, 2015, 12:00:00 AM	100%	✎ 🗑️
☆ 2015Nov10Build	Nov 10, 2015, 12:00:00 AM	84%	✎ 🗑️

Items per page: 10 ▼ 1-5 of 5 items 1 ▼ of 1 pages ⏪ ⏩

Cancel Save baseline builds

20. Click **Save baseline builds** to save your changes.

Baseline Builds Saved ✕

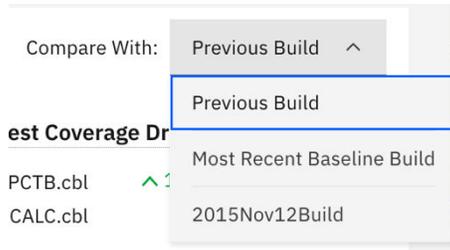
Your new baselines builds selection has been saved successfully. If you removed baselines then all associated metrics were deleted. Calculation of any new baseline metrics will be triggered automatically on the next data collection cycle and you will be notified when the new data is ready to use. Do you want to start data collection right away instead?

Cancel Start Data Collection

21. Click **Start Data Collection** on the window for ADDI Extensions to perform the data collection of the latest build and generate the code coverage reports against the new baseline build. When the data collection is completed, you will notice that the Last Collection date is updated to the current time.

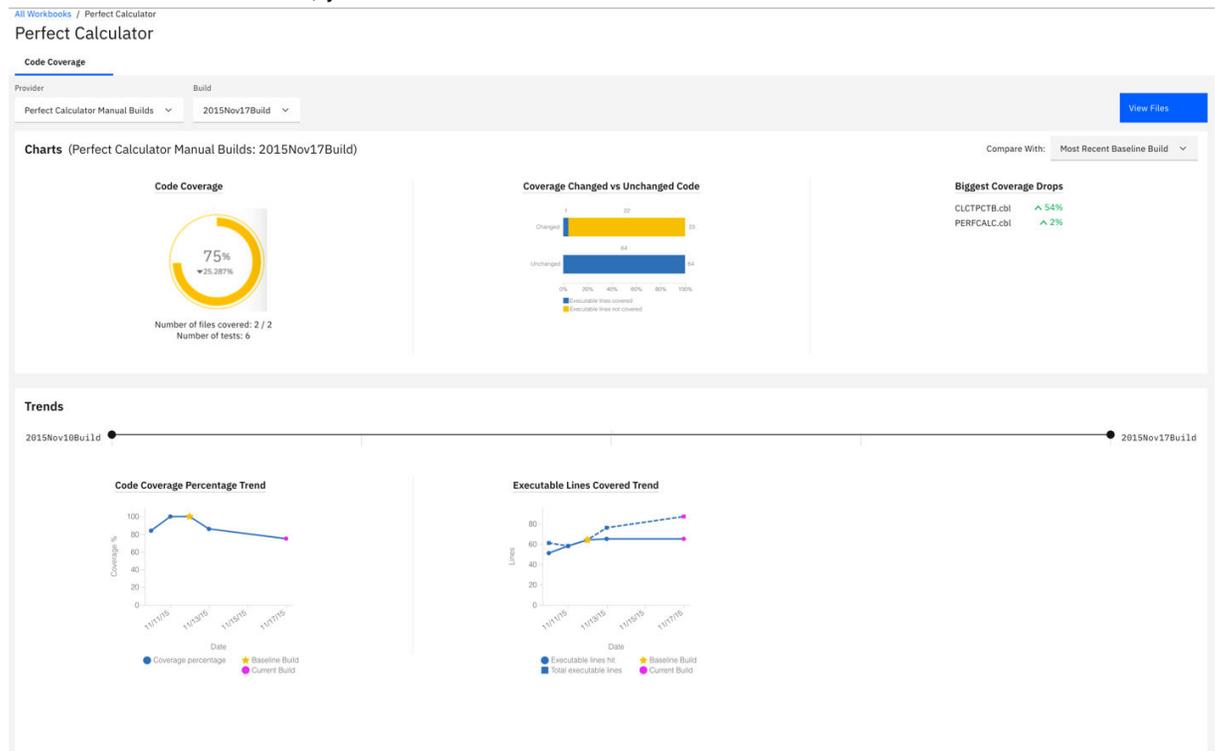
Perfect Calculator Manual Builds Manual builds for Perfect Calculator application 🏠 Manual Builds 🟢 9/13/20, 3:54 AM ⋮

- Select the **Workbooks** tab on the header to go to the **Workbooks** page.
- Select **Perfect Calculator** workbook to view the code coverage reports of **Perfect Calculator**.
- On the header of the **Charts** section, select **Compare With** drop-down menu. By default, the code coverage information, such as the percentage of code coverage changes and the percentage of executable line changes, is calculated against the previous build. After you define one or more baseline builds, ADDI Extensions also calculates the information against the baseline builds that you define.

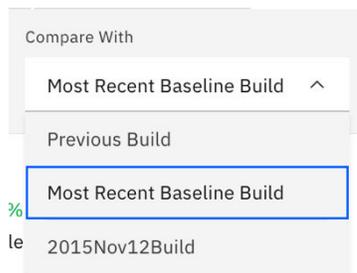


- Select **Most Recent Baseline Build**. With the reports that load, you can see the updated information, such as the percentage of code coverage changes is updated from 10.81% to 25.287% and the code coverage of CLCTPCTB . cbl is dropped by 54%.

Under the **Trends** section, you can see the baseline build is marked on the trends line as well.



- Select **View Files** button on the upper right to go to the **Build Analysis** view.
- Select **Most Recent Baseline Build** from the **Compare With** drop-down menu on the header area.



The view is updated to display the code coverage results comparing with the most recent baseline build.

All Workbooks / Perfect Calculator / Build Analysis

Build Analysis

Files Pinned Files Tests File Trends

Provider: Perfect Calculator Manual Builds Build: 2015Nov17Build Compare With: Most Recent Baseline Build Filters: Select a Filter

Summary

75% ▼ -25.3% Code Coverage 65 / 87 ▲ 26.4% Covered / Total Executable Lines 2 Modified Files 0 New Files 6 Historical Tests 3 Minimal Tests

Search

Item Name	Warnings	Code Coverage	Change	Executable Lines						Tests	
				Covered	Total	Added	Updated	Change	Deleted	Historical	Minimal
CLCTPCTB.cbl		46% ▼ <div style="width: 46%; background-color: red; height: 10px; display: inline-block;"></div>	▼ -54%	18	39	21	0	▲ 53.8%	0	3	1
PERFCALC.cbl		98% ▲ <div style="width: 98%; background-color: green; height: 10px; display: inline-block;"></div>	▼ -2%	47	48	2	0	▲ 4.2%	0	6	3

Now you are satisfied with your investigation. You suggest your team using ADDI Extensions to always check the code coverage reports and check against the baseline build to ensure that the application is fully tested.

Now you have explored how you can use the test analysis results of the latest build to investigate the testing issues of your build. Typically, you can use this information to perform the following analysis.

- Determine if the coverage has improved from the previous builds.
- Determine if the changed files were indeed covered during the testing.
- Compare the results of the current build with one or more older builds to see differences between coverage results of these builds.
- Determine how much the set of files have actually changed from build to build.
- Review the trends of changes and coverage over time to determine the quality of the tests performed.
- Set the baseline builds for code coverage analysis against major builds or milestone builds.

Generating sample data

In this tutorial, you play the role of ADDI Extensions admin, Alvin, to generate the sample data for the static analysis tutorial by using the Sample Data Generation function.

The Sample Data Generation function of ADDI Extensions is useful when you do not have access to the Application Discovery instance. You can use the sample data that ADDI Extensions generates to see how the data is analyzed. After the data is generated, you can create and set up a workbook to view the analysis results.

Before you begin

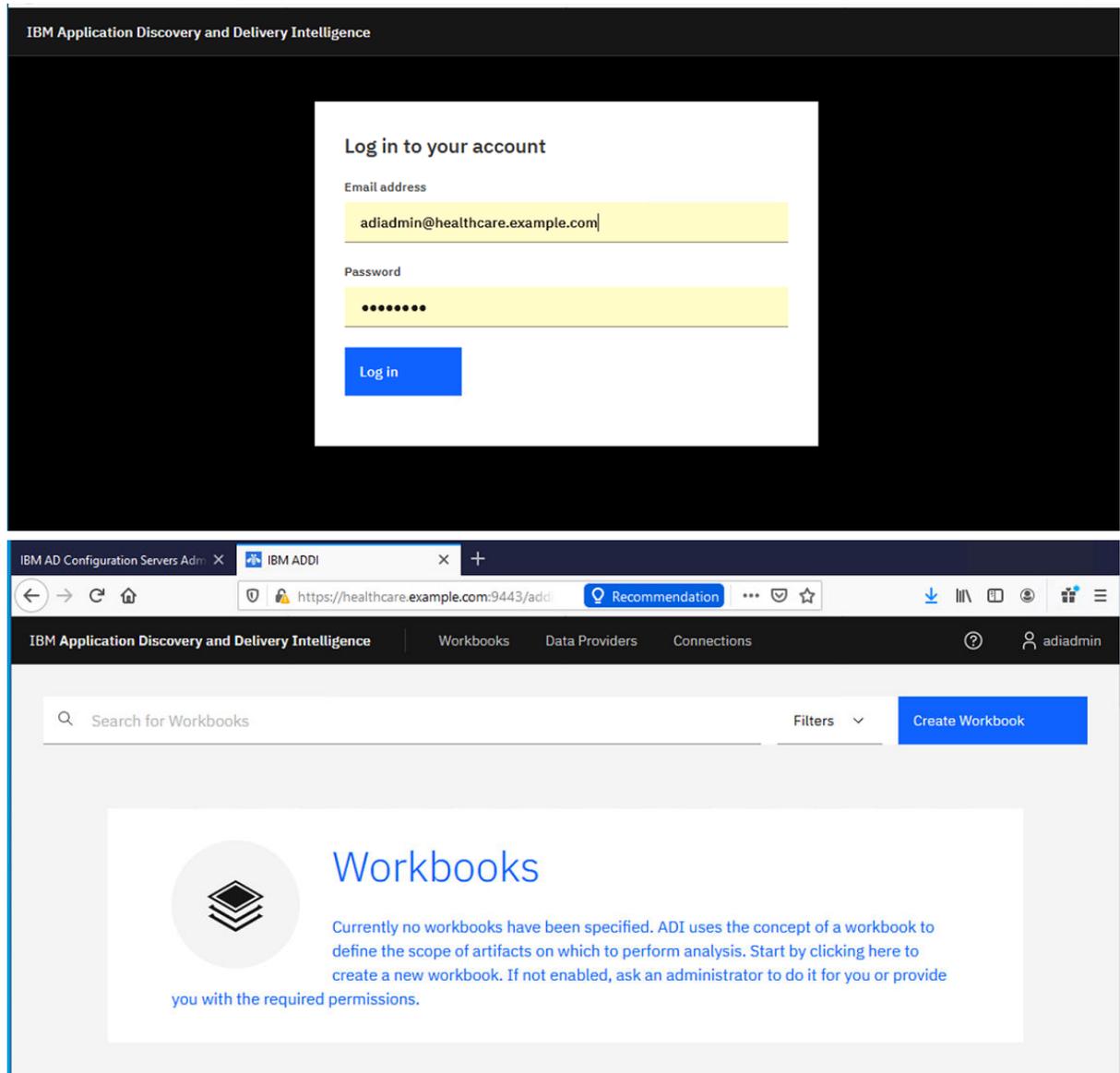
Before you begin this tutorial, perform the activities as described in [“Installing and setting up ADDI Extensions”](#) on page 5.

Procedures

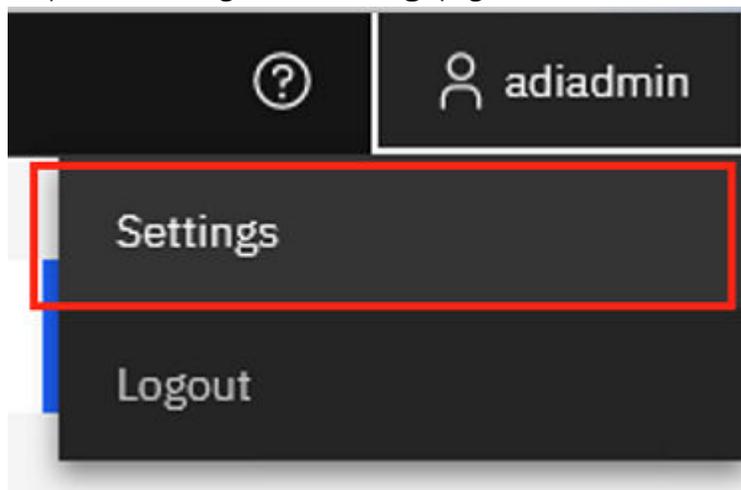
Complete the following steps to generate sample data for Application Discovery.

1. Open your Firefox browser and go to ADDI Extensions home page at <https://healthcare.example.com:9753/addi/web/workbook>.
2. Log in with the following credentials. After you log in, the **Workbooks** page is displayed.

- Email address: adiadmin@healthcare.example.com
- Password: adiadmin



3. Click the **User Profile** icon on the upper right of the **Workbooks** page and select **Settings** from the drop-down list to go to the **Settings** page.



4. Select **OMEGAMON for CICS** from Data Providers drop-down list on the Sample Data Generation form.

Settings

Sample Data Generation

Select a data provider kind and specify data generation settings for the sample.

Data Providers

OMEGAMON for CICS ▼

Data Generation Settings

Plex

CICSPLX2::CICSgen

Days

7 ▲▼

Goal Response Time (μs)

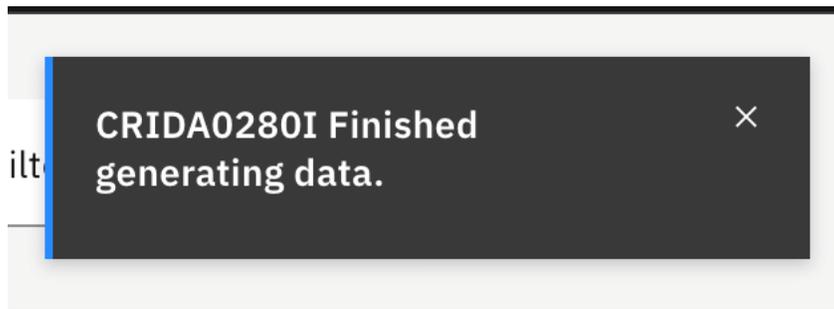
1000000 ▲▼

- Execute Warehouse RUNSTATS after data generation

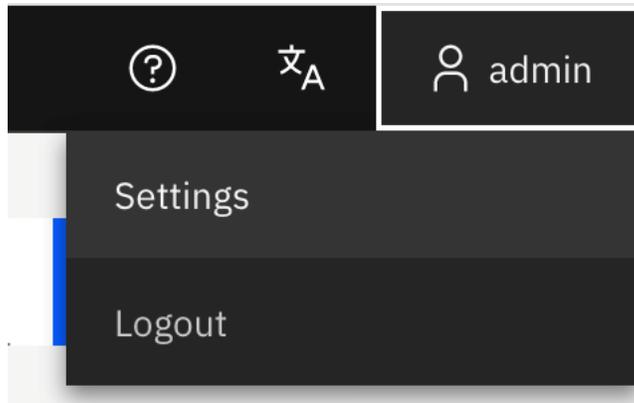
Generate

Cancel

5. Leave all the data in the **Sample Data Generation** form as default and click **Generate**.
6. Close the message dialog box, which shows that the data generation is finished.



7. Click the **User Profile** icon on the upper right of the **Workbooks** page and select **Settings** from the drop-down list to go to the **Settings** page.



8. Leave all the data in the **Sample Data Generation** form as default and make sure that Application Discovery is selected in the Data Provider drop-down list. Then, click **Generate**.

Settings

Sample Data Generation

Select a data provider kind and specify data generation settings for the sample.

Data Providers

Application Discovery

Data Generation Settings

Data Points

7

Time Interval (hours)

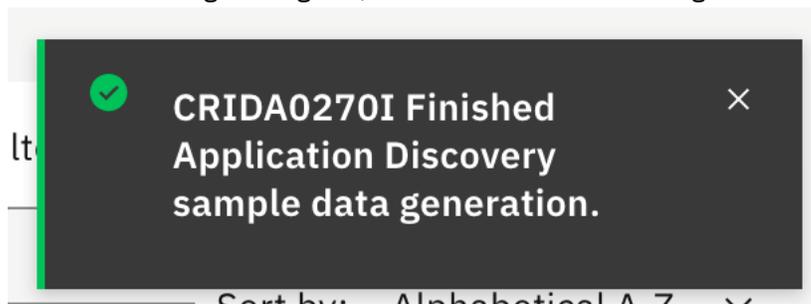
24

Execute Warehouse RUNSTATS after data generation

Generate

Cancel

- Close the message dialog box, which shows that the data generation is finished.



- Select the **Data Providers** tab. You can see that the **Demo Application Discovery provider** and **Demo OMEGAMON for CICS provider** are created.

Data Providers

Search by name or description

Filters

Create Data Provider

Name	Description	Type	Last Collection Date	Actions
Demo Application Discovery provider	FOR DEMONSTRATION PURPOSES ONLY	Application Discovery	10/20/20, 2:52 PM	
Demo OMEGAMON for CICS provider	FOR DEMONSTRATION PURPOSES ONLY	OMEGAMON for CICS	10/20/20, 2:50 PM	

- Select the **Connections** tab. You can see that the **Generated Application Discovery connection** and **Generate OMEGAMON for CICS connection** are created.

IBM Application Discovery and Delivery Intelligence | Workbooks | Data Providers | **Connections** | ? | admin

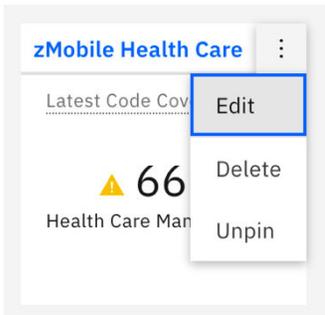
Connections

Search for Connections | Filters | Create Connection

Connection Name	Description	Connection Type	Connection URL	Actions
Generated Application Discovery connection	FOR DEMONSTRATION PURPOSES ONLY	IBM Application Discovery	http://sample.com	
Generated OMEGAMON for CICS connection	FOR DEMONSTRATION PURPOSES ONLY	IBM OMEGAMON for CICS	http://sample.com	

Now, the connections and data providers of Application Discovery and OMEGAMON for CICS are generated for the demonstration. Typically, the next step is to generate a workbook and associate both data providers with the workbook to analyze both data provider together. For this exercise, you will use the existing workbook which you have created in the “Setting up automated code coverage data collections” on page 83 tutorial.

12. Select the **Workbooks** tab to go to **Workbooks** page.
13. Click the overflow menu (vertical ellipsis) icon on the upper right corner of the **zMobile Health Care** card. An option list appears.



14. Select **Edit** from the option list to view the analysis workbook information. The **Edit Workbook** page appears.
15. Scroll down to the **Providers** section. You can see that **Health Care Manual Builds** checkbox is selected.

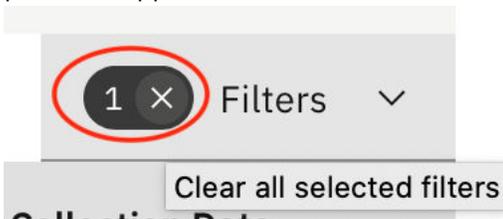
Data Providers

Search by name or description | 1 x Filters

<input checked="" type="checkbox"/> Name	Description	Type	Last Collection Date
<input checked="" type="checkbox"/> Health Care Manual Builds	Manual build with automated data collection for Health Care application.	Manual Builds	10/20/20, 2:45 PM

Items per page: 20 | 1-1 of 1 items | 1 of 1 pages

16. In the **Providers** section, click **X** next to the number 1 to clear all the selected filter. A list of all data providers appears.



17. Select **Demo Application Discovery provider** and **Demo OMEGAMON for CICS provider** as additional data providers.

Data Providers			
<input type="text" value="Search by name or description"/>			Filters ▼
<input type="checkbox"/> Name	↑ Description	Type	Last Collection Date
<input type="checkbox"/> CC provider		Manual Builds	✓ 10/20/20, 5:50 AM
<input checked="" type="checkbox"/> Demo Application Discovery provider	FOR DEMONSTRATION PURPOSES ONLY	Application Discovery	🕒 10/20/20, 2:52 PM
<input checked="" type="checkbox"/> Demo OMEGAMON for CICS provider	FOR DEMONSTRATION PURPOSES ONLY	OMEGAMON for CICS	🕒 10/20/20, 2:50 PM
<input checked="" type="checkbox"/> Health Care Manual Builds	Manual build with automated data collection for Health Care application.	Manual Builds	✓ 10/20/20, 2:45 PM

The **OMEGAMON for CICS Settings** form and **Static Analysis Settings** form appear as tabs next to the **Code Coverage Settings** from.

18. Select the **Static Analysis** tab and fill in the following information in the **Static Analysis Settings** form.

Settings

Static Analysis Code Coverage OMEGAMON for CICS

Select Project(s) * 🔍 Search for Pro...

Select all

BBO4_maint ✓	BBO4_preprod ✓	BBOI_maint ✓
BBOI_preprod ✓	genericGraph ✓	graphSample ✓
sharedSample1 ✓	sharedSample2 ✓	zMobile ✓

Source Changes Detection
Interval (days)

7 ⬇️

Threshold Settings ⌵

Maintainability Index

50 —————●—————●————— 85

Unreachable Code

10% —————●—————●————— 20%

Cyclomatic Complexity

10 —————●—————●————— 50

- **Select Project(s)**: Select the **Select all** checkbox.
- **Source Changes Detection Interval (days)**: Use the default value, which is 7.
- **Threshold Settings**: Move the lower slide bar of Maintainability Index to 50. Leave the rest as the default values.

19. Select **OMEGAMON for CICS** tab and fill in the following information in the **OMEGAMON for CICS settings** form.

Settings

Static Analysis

Code Coverage

OMEGAMON for CICS

Select the service classes to analyze from the list below. Then choose the timeframe to consider for the analysis dashboard's metrics.

Service Classes *

Search

BTRANS <input checked="" type="checkbox"/>	CTRANS	DTRANS
ETRANS	FTRANS	HTRANS

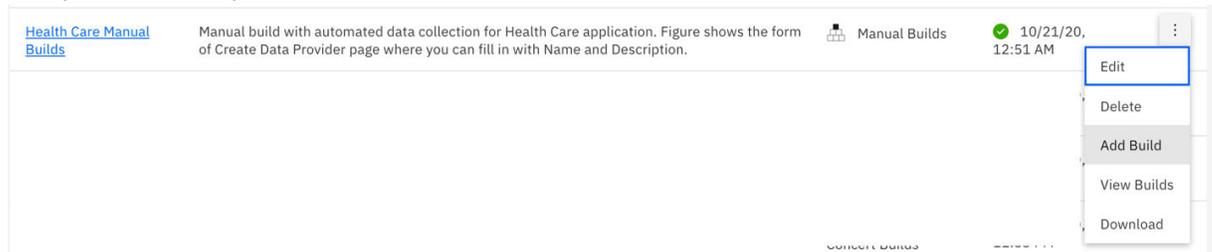
Timeframe to view Dashboard's Metrics

Last Day

- **Service Classes:** Select **HTRANS**.
 - **Timeframe to view Dashboard's Metrics:** Leave as default value, that is, **Last Day**.
20. Click **Save** to save the changes to this workbook. The **Workbooks** page appears.

Now you have associated the **Demo Application Discovery** provider and **Demo OMEGAMON** provider with zMobile Health Care analysis workbook. You are ready to perform analysis. However, for the demonstration objective, you want to add additional code coverage results to the **Health Care Manual Builds** data provider.

21. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
22. Click the overflow menu (vertical ellipsis) icon on the **Health Care Manual Builds** data provider row. An options menu opens.



23. Select **Add Builds** from the options menu. The Add Build dialog box opens.
24. Complete the Add Build dialog box with the following information.

- **Build Name** and **Date of Build:** Set them to 2 weeks before the day you generate the sample data. For example, if you generate the sample data on October 21, 2020, you will update the **Build Name** to 2020ct7Build and the **Date of Build** to 10/07/2020 as shown in the following screenshot.
- **Code Coverage Files:** Browse to *Installed ADDI Extensions folder* > *adi611* > *examples* > *cobol-coverage* > *zMobile Health App* > *Build2* folder. Then, select all the zip files under Build2 folder.

Add Build ×

Build Name *

Date Of Build *

Build Description

October 7, 2020 build for Health Care application

Code Coverage Files

Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

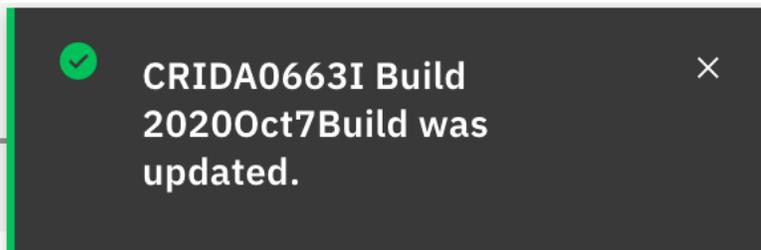
Browse

Addmeds.zip	×
AddmedsErr.zip	×
AddPatient.zip	×
AddPatientErr1.zip	×
AddPatientErr2.zip	×
ADDPRI.zip	×

Cancel

Add

25. Click **Add** to add the new build. A message box appears when the build update is done.



26. Close the message box.

27. Click the overflow menu (vertical ellipsis) icon on the **Health Care Manual Builds** data provider row. An options menu opens.

28. Select **Add Builds** from the options menu to add another build. The Add Build dialog box opens.

29. Complete the Add Build dialog box with the following information.

- **Build Name** and **Date of Build**: Set them to one week before the date that you generate the sample data. For example, if you generate the sample data on October 21, 2020, you will update the **Build Name** to 2020Oct14Build and the **Date of Build** to 10/14/2020 as shown in the following screenshot.
- **Code Coverage Files**: Browse to *Installed ADDI Extensions folder > adi611 > examples > cobol-coverage > zMobile Health App > Build3* folder. Then, select all the zip files under Build3 folder.

Add Build

×

Build Name *

Date Of Build *

Build Description

October 14, 2020 build for Health Care application.

Code Coverage Files

Browse and select one or more code coverage zip files created by testing this build. You can add more files later.

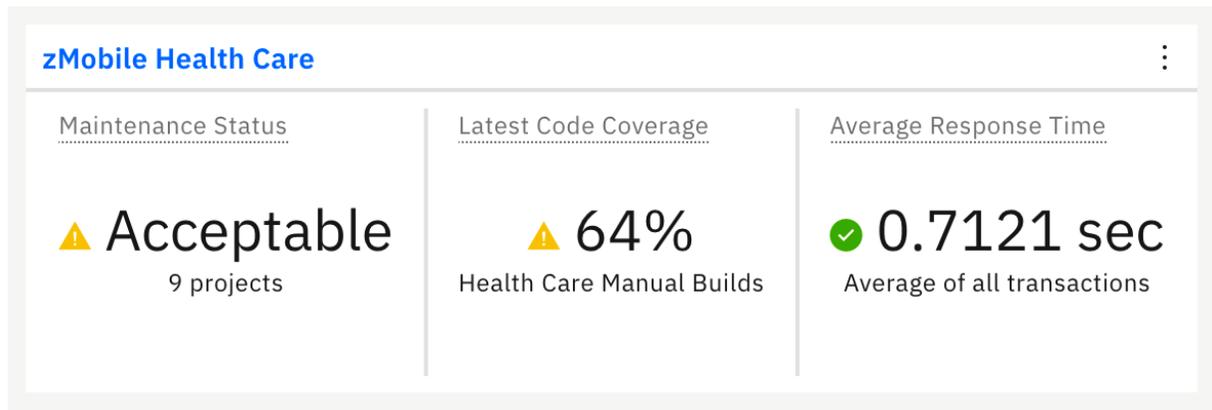
Browse

Addmeds.zip	×
AddmedsErr.zip	×
AddPatient.zip	×
AddPatientErr1.zip	×
AddPatientErr2.zip	×
AddPRI.zip	×

Cancel

Add

30. Click **Add** to add the new build. A message box appears when the build update is done.
31. Close the message box.
32. Select the **Workbooks** tab to go to the **Workbooks** page. You can now see that the percentage of code coverage for zMobile Health Care is updated.



33. Now you have updated the information on **Health Care Manual Builds** data provider. You are now ready to perform the end-to-end analysis of performance issues. In the next tutorial, you play the role as a development lead to analyze the potential cause of performance issues for this zMobile Health Care application.

End-to-end system performance root cause analysis

This tutorial guides you through the end-to-end scenario about how to perform root cause analysis by analyzing data from multiple data providers. In this tutorial, you will perform root cause analysis of a performance defect that is found in the zMobile Health Care application

Prerequisite

Before you begin this tutorial, you need to perform the following activities.

- [“Installing and setting up ADDI Extensions” on page 5](#)
- [“Setting up automated code coverage data collections” on page 83](#)
- [“Generating sample data” on page 106](#)

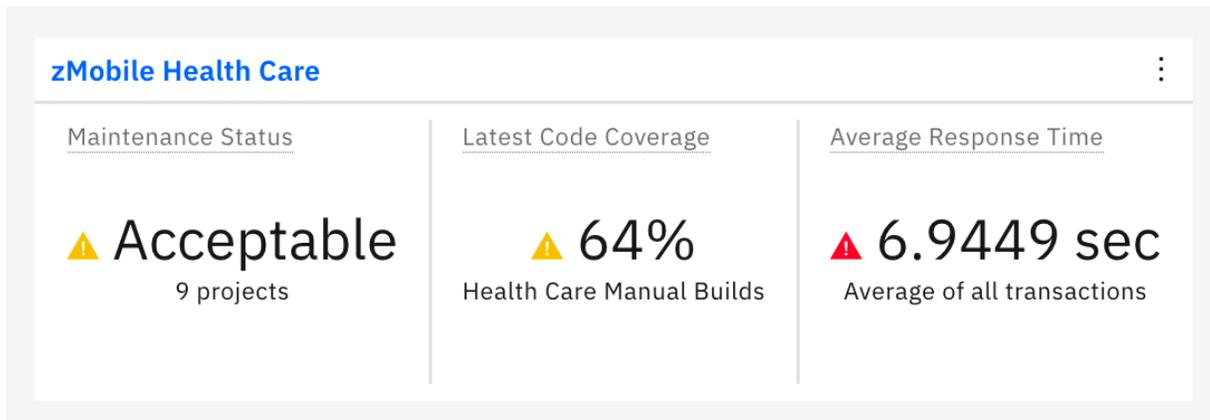
Note: Since the data in this tutorial is the generated sample data, you might not see the exact same data on your reports.

Trish is a test lead who is testing the response time of the Inquire Patient feature of the zMobile Health Care application. During the testing, Trish found that in the latest development build, the response time of the Inquire Patient feature increased to 2 minutes from less than a second in the previous build. She reported it to you, the development lead. And you investigate the issues by consulting dashboards and reports in ADDI Extensions.

As a development lead, you need to complete the following steps to perform the root cause analysis for the performance issue:

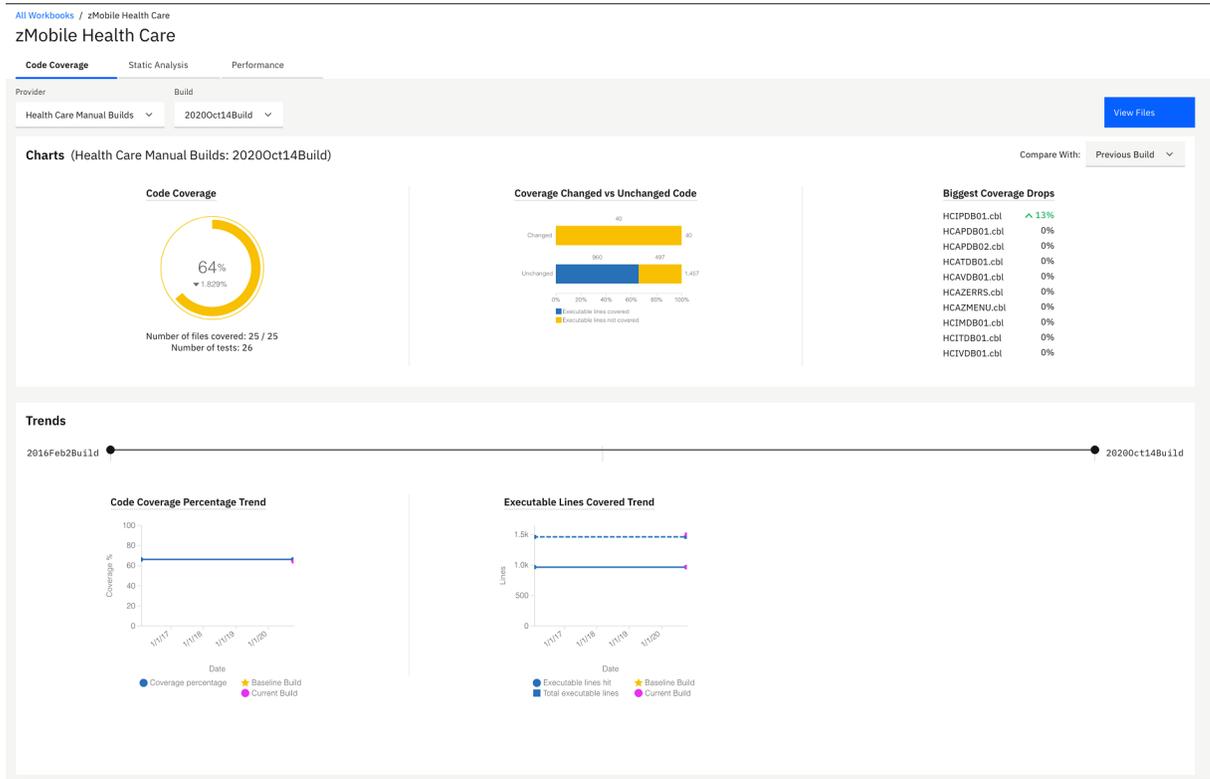
1. Navigate your browser to ADDI Extensions home page <https://healthcare.example.com:9753/addi/web/workbook>.
2. Log in with the following credentials. After you log in, the **Workbooks** page is displayed.
 - **Email address:** `adiadmin@healthcare.example.com`
 - **Password:** `adiadmin`

You can see the dashboard that summarizes the overall status of zMobile Health Care application. You can notice the problem that Trish reported right away. The average response time of zMobile Health application is almost 7 seconds. It is higher than the service level agreement for your company. And you can notice that the code coverage percentage is also in the warning area.



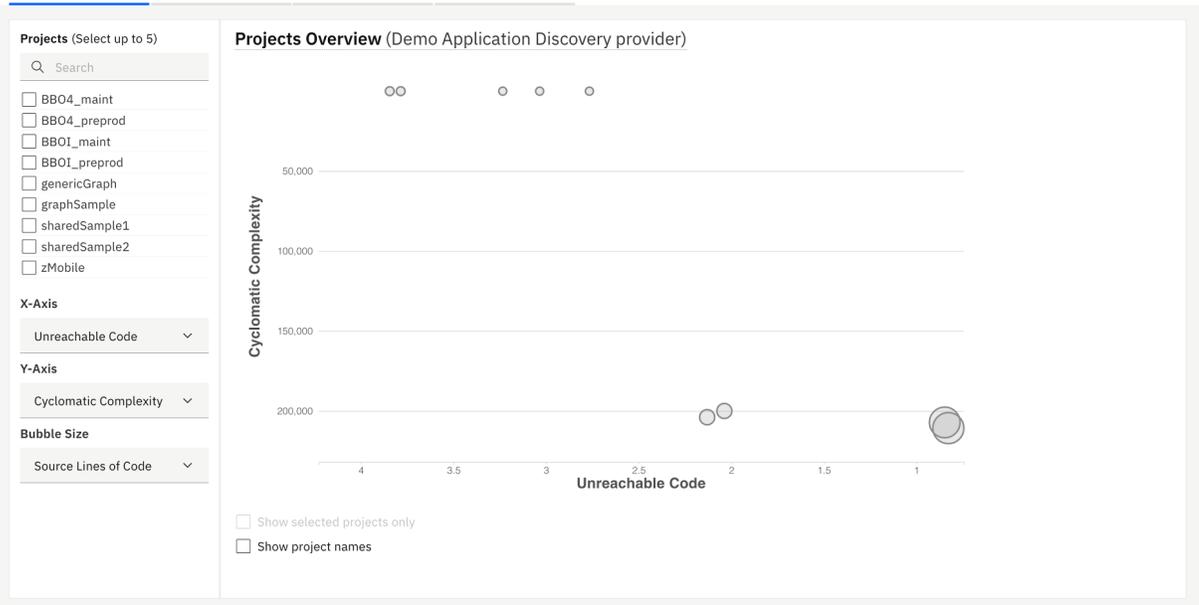
- Click the name of **zMobile Health Care** to view detailed analysis. The zMobile Health Care page loads with summary charts of the contributing providers.
- View the reports on the workbook summary view of zMobile Health Care page. The reports are organized into 3 tabs: **Code Coverage**, **Static Analysis**, and **Performance**. All the reports are analyzed based on three data providers that are associated with this analysis workbook:
 - A Manual Builds data provider that collects the code coverage results.
 - An Application Discovery data provider that collects static analysis data about transactions, programs and project-level metrics in the zMobile Health Care application.
 - An OMEGAMON for CICS data provider that monitors performance of the zMobile Health Care application CICS transactions.

With reports from multiple providers in one workbook, you can correlate different data from different sources.

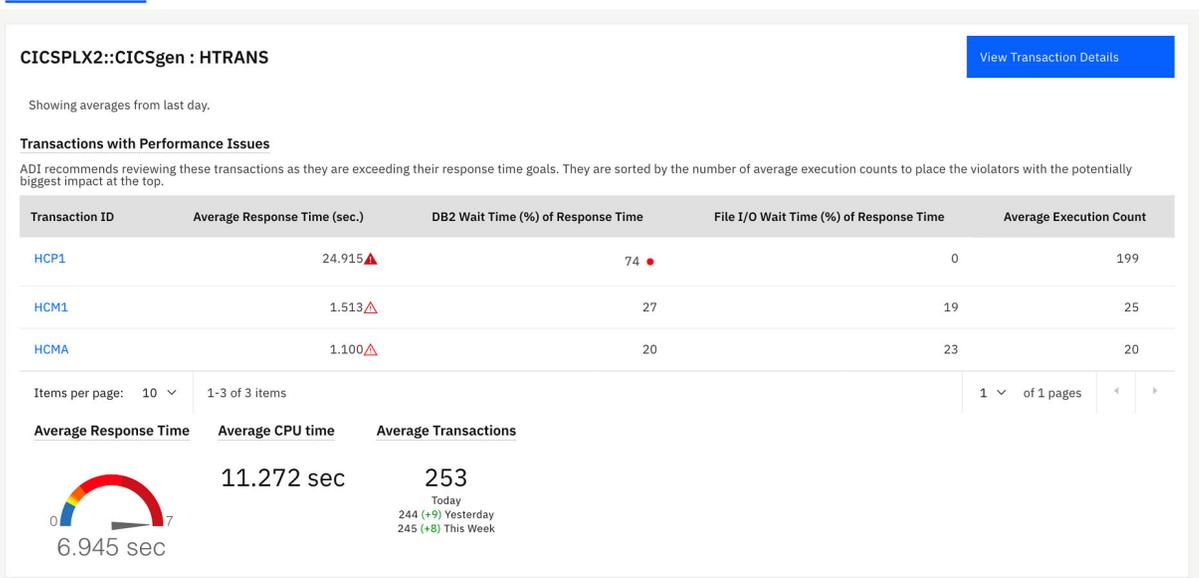


- Select **Static Analysis** tab and **Performance** tab to navigate to static analysis reports and performance reports respectively.

zMobile Health Care

Code Coverage **Static Analysis** Performance**Overview** Metrics Structure Trends

zMobile Health Care

Code Coverage Static Analysis **Performance****Omegamon**

On the **Performance** tab, you can see the detailed analysis of program and transactions within the zMobile project that displays along with the performance reports of HTRANS.

HTRANS is a service class in OMEGAMON for CICS data provider. Service class is a concept that OMEGAMON for CICS uses to allow grouping related transactions together. If no service class is defined, OMEGAMON for CICS would define a default service class.

In the **Demo OMEGAMON provider: HTRANS** section, you can investigate the performance issue that Trish reported. Besides the average response time that is in the red zone, you notice that on the Transactions with Performance and Reliability Issues table, **HCP1** transaction that is run by the *Inquire Patient* feature is flagged as exceeding the goal response time threshold by more than

400%. Average execution count of HCP1 is also much higher than the other transactions. Then you think that this transaction probably causes the application to slow down.

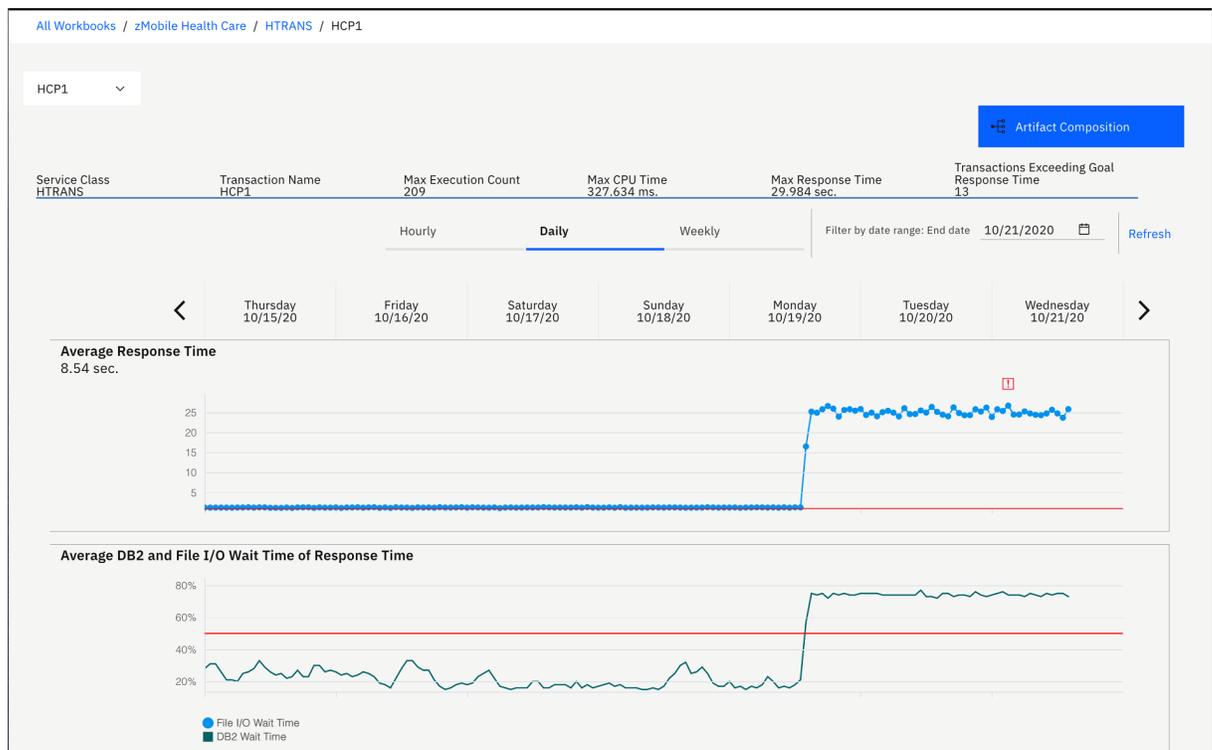
Note: Goal response time threshold is the setting in OMEGAMON for CICS to indicate the severity of response time that exceeds the standard service level agreement that the company requires. The different warning icons on the Transactions with Performance and Reliability Issues table indicate different levels of severity. For example, the transparent red warning icon indicates that the response time exceeds threshold by 400%, and the opaque red warning icon indicates that the response time exceeds threshold by more than 400%.

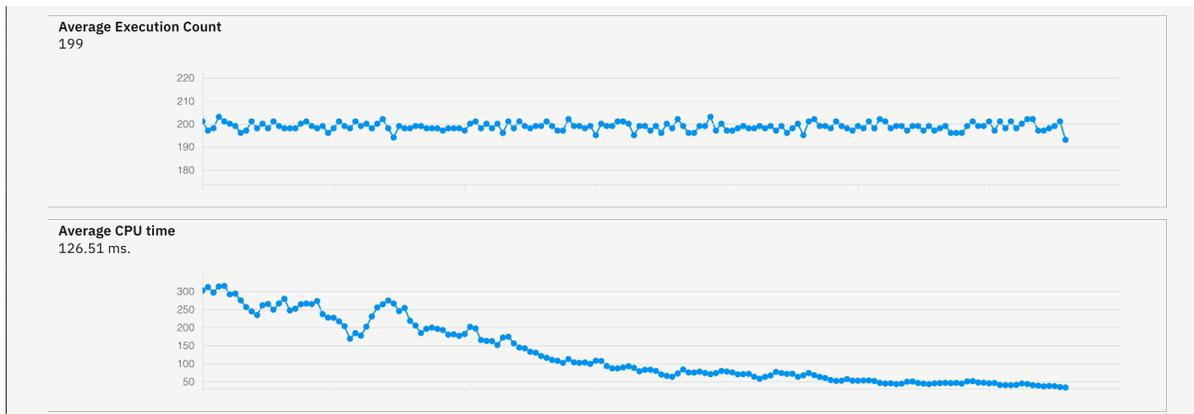
6. Click the **HCP1** transaction on the Transactions with Performance and Reliability Issues table to view the detailed transaction analysis. You now see the history of the average response time comparing to Average DB2 and File I/O wait time of response time, average CPU time, and execution count.

On the header of the report, the maximum number of the execution count, CPU time, and response time for a given timeframe display on this view.

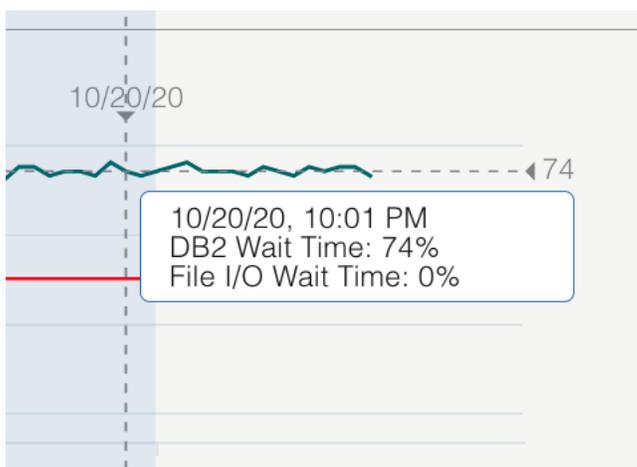
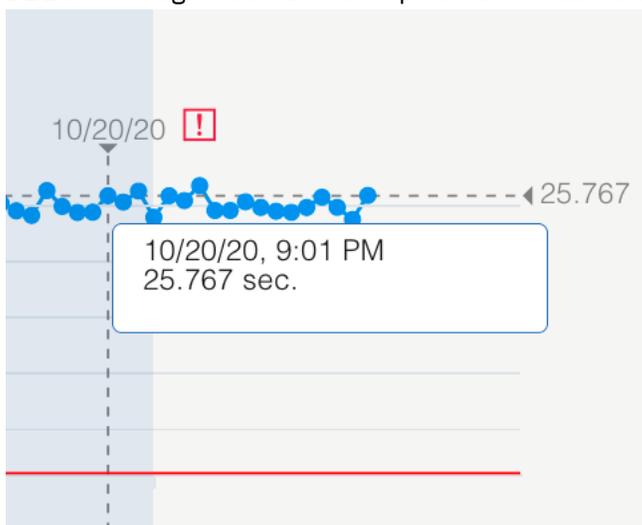
From the chart, you notice that the average response time was constantly rising high for the past 3 days from 2 seconds to 26 seconds. The same pattern occurs to DB2 wait time. DB2 wait time went up from about 20 - 30% to over 70% which is higher than the suggested level 50% that is indicated by the red line on the Average DB2 and File I/O Wait Time of Response Time.

You notice that the execution is always at about almost 200 which means this transaction has been executed frequently. You suspect that there probably are some changes made to the program within this transaction 3 days ago.



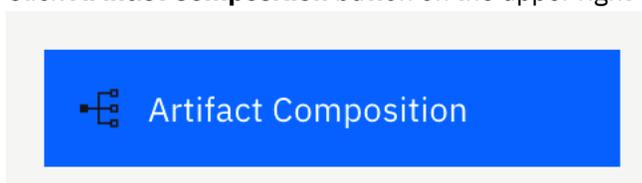


7. Hover over the chart on the area where the response time exceeds the threshold and the average DB2 time is high. You can see response time value and percent DB2 wait time value.



8. Click **OK** to do further analysis.

9. Click **Artifact Composition** button on the upper right corner to go to the **Artifact Composition** view.



- Check if the Maintainability Index and Number of Source Line of Code have been updated between the change interval time you have defined.

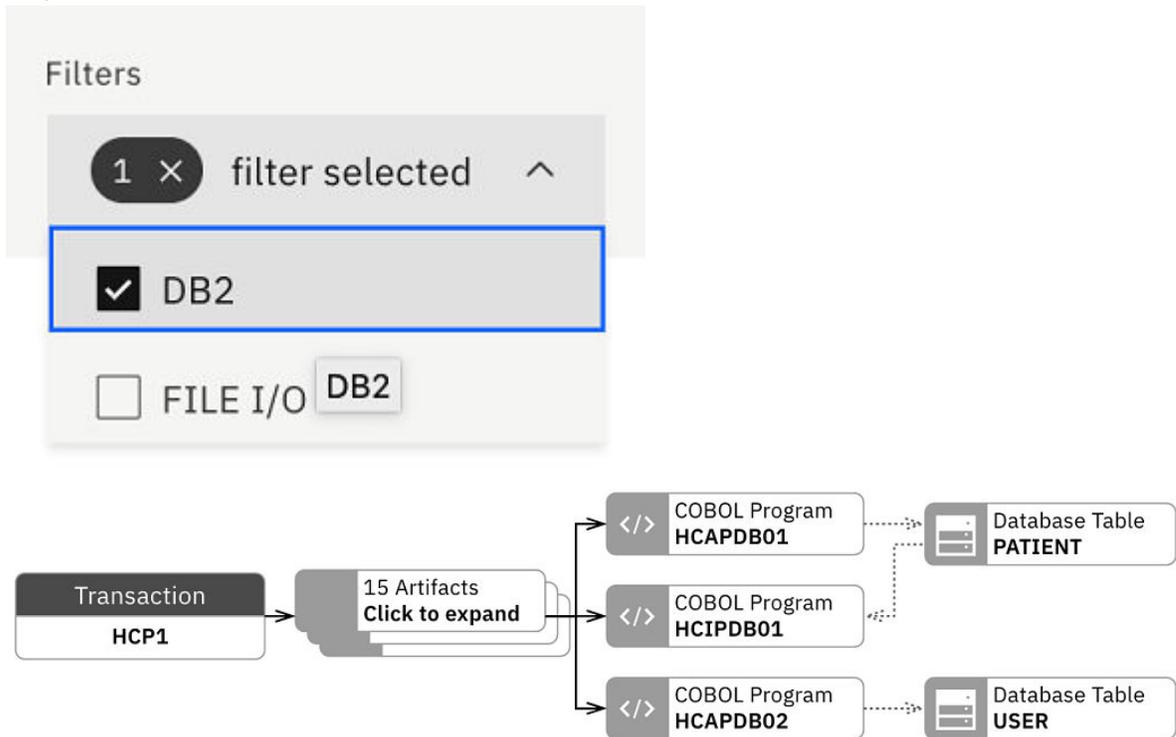
The red warning icon indicates that the artifact is risky to modify. The Risk Areas dialog box on top of the chart explains the meaning of blue and red icons with the threshold setting for key metrics.

11. You notice the warning icons on top of the artifacts. The blue warning icon indicates that there could be some changes to the artifacts. Because there are changes to artifact metrics during the interval time that you have set when you set up an analysis collection in the previous tutorial. In this case, it is 7 days which is between February 28th and March 7, 2018. For ADDI Extensions, the changed artifact is indicated by one of the following method.

- Check if there is the LASTUPDATED metric that is collected from IBM Application Discovery.
- Check if the Maintainability Index and Number of Source Line of Code have been updated between the change interval time you have defined.

The red warning icon indicates that the artifact is risky to modify. The Risk Areas dialog box on top of the chart explains the meaning of blue and red icons with the threshold setting for key metrics.

12. Click **X** on top right of the Risk Areas dialog to close the dialog box. The Risk Areas dialog disappears along with the warning icons.
13. Click the **Filters** drop-down list and select **DB2**. The Artifact Composition graph is updated to show only artifacts that are connected to DB2.



14. Click the **Risk Areas** icon () on the top menu. The **Risk Areas** dialogue displays with the warning icons on the Artifact Composition graph.

You can see the blue warning icon on **HCIPDB01**. This means that there could be some changes made to **HCIPDB01** in the past week.

15. Close the **Risk Areas** dialogue and click **HCIPDB01** box. The key metrics collected from Application Discovery for this artifact are displayed. You can see that the Maintainability Index is in red area but it is not your concern at this point.

You can notice that the code coverage is 82%. When you hover over the question mark on the Test Code Coverage metric for **HCIPDB01**, you see that the date of the last test code coverage data is a

week from today, which is before the date when the program was changed last time. This means that it has not been properly tested.

COBOL Program - HCIPDB01 ✕

Metrics [View Trends](#)

Active Statements	86
Unreachable Code	2.22
Delivered Bugs	0.4
• Maintainability Index	40.55
Total Statements	87
Cyclomatic Complexity	10
Source Lines of Code	136

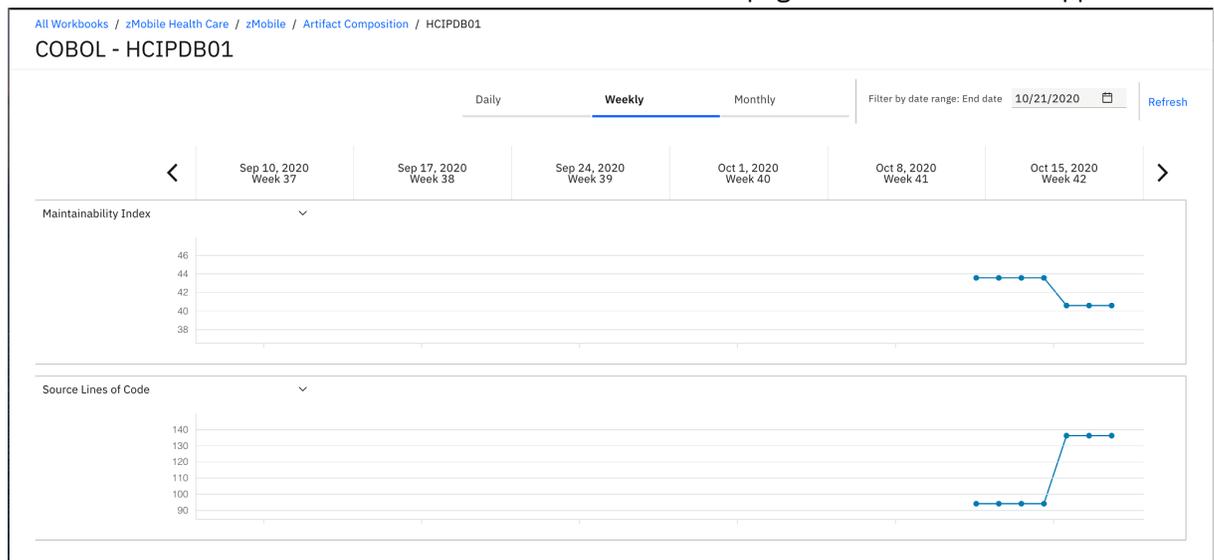
Program References

Incoming	1
Outgoing	2

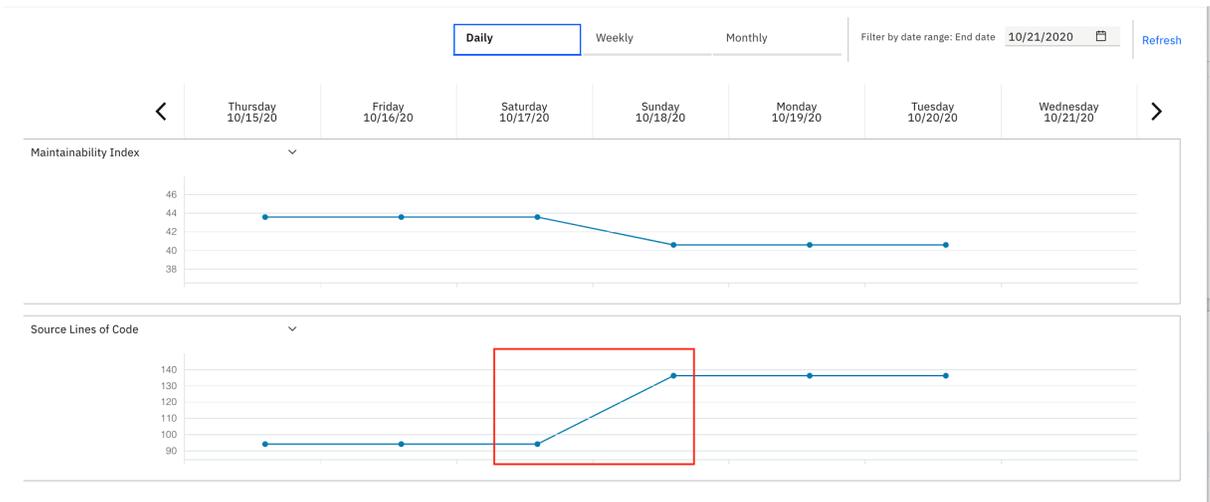
Test Code Coverage

Health Care Manual Builds	82
---------------------------	-----------

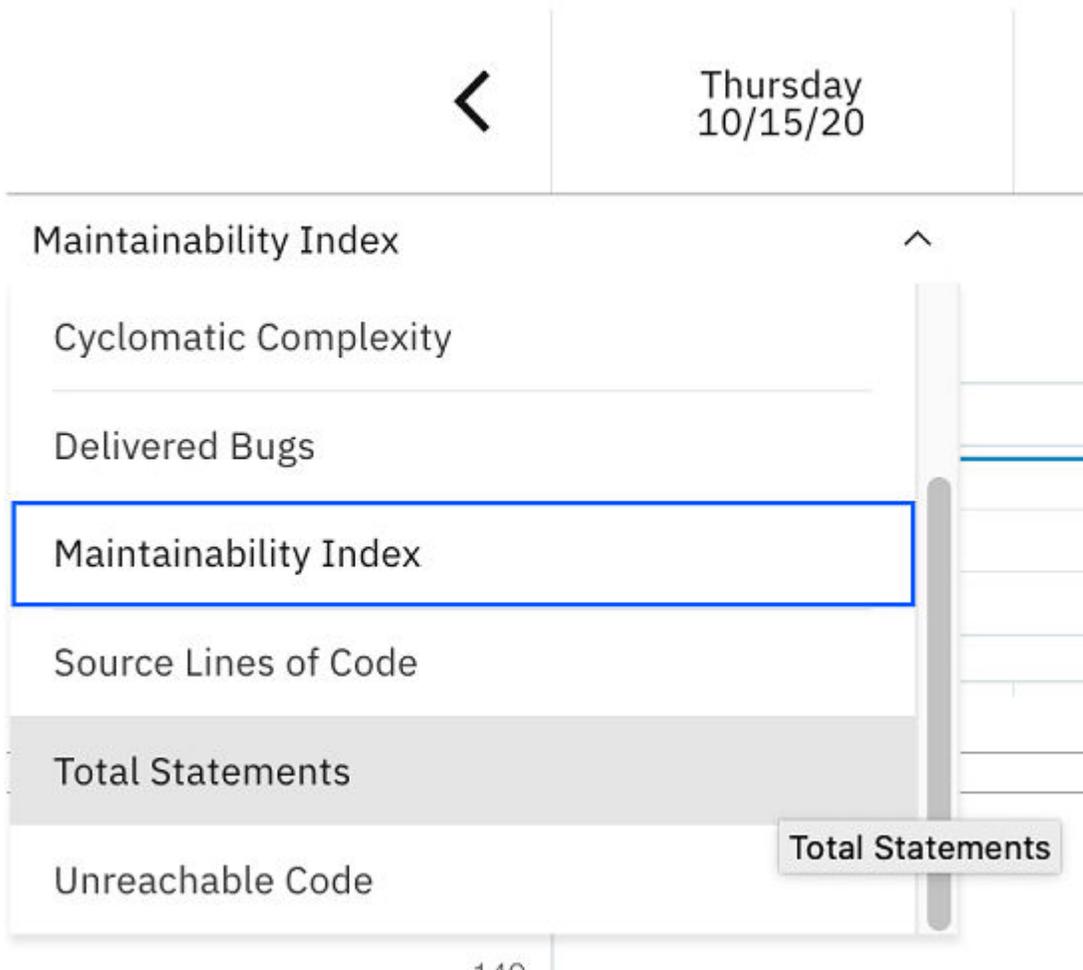
16. Click **View Trends** link next to the Metrics header. The **Trends** page for HCIPDB01 . cb1 appears.



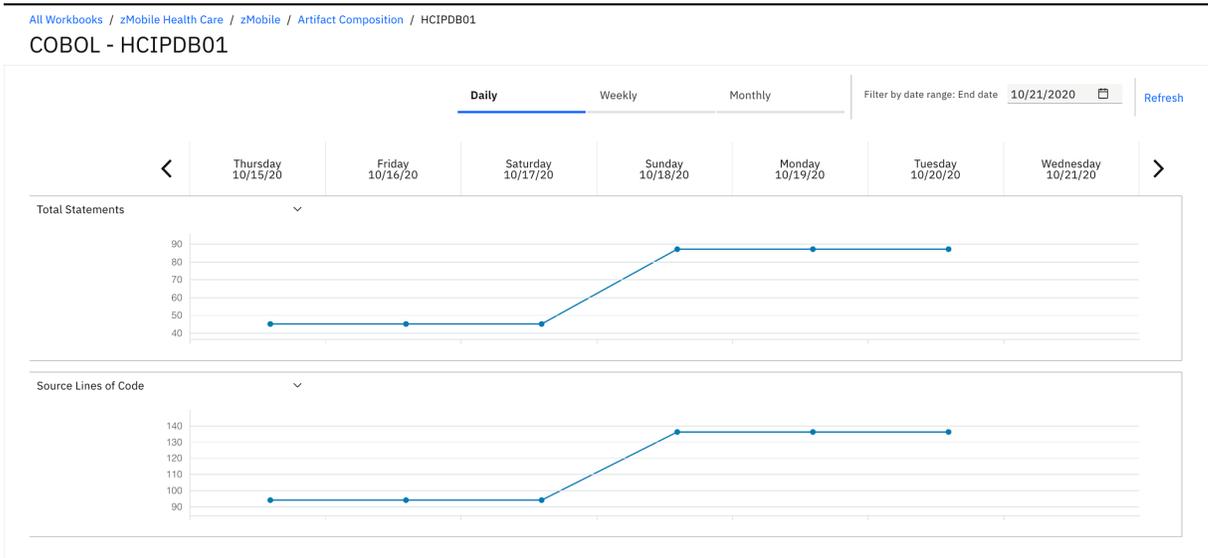
17. Click **Daily** to change the view from **Weekly** view to **Daily** view. Observe the **Trends** page that shows the trend lines of Maintainability Index and Source Lines of Code. You notice that Source Lines of Code went up about 4 days ago.



18. Select the drop-down list of **Maintainability Index** and select **Total Statements**. You notice that both metrics have the same trending. About 40 statements are added to HCIPDB01 . cb1.



Now you as the development lead conclude that the **HCIPDB01** should be the area of concern. You discuss with Jane, the developer. You two investigate further into the code and find that an extra SQL Join statement contributes to the excessive wait time. Before making modifications, you and Jane go back to ADDI Extensions.



19. Select **Artifact Composition** link on the breadcrumb to go back to the Artifact Composition table.

All Workbooks / zMobile Health Care / zMobile / **Artifact Composition** / HCIPDB01

COBOL - HCIPDB01

20. Click the **Table** () icon on the top header to update the view to table view.

All Workbooks / zMobile Health Care / zMobile / Artifact Composition

Artifact Composition

Project: zMobile Region: CSR1 Transaction: HCP1 Date: 10/20/20, 2:17 PM

ADI recommends reviewing the top-ranked artifacts first as they are the potentially riskiest to modify. They are ranked based on factors such as thresholds comparison, outlier analysis status, and static metrics values. [Learn more](#)

Outlier Status	Artifact Type	Program Name	Maintainability Index	Unreachable Code	Cyclomatic Complexity	Number Of Incoming References	Number Of Outgoing References
Outlier	COBOL	HCAZERRS	5.721 ▲	38.696% ▲	24 △	5	0
	COBOL	HCP1PL01	33.892 ▲	7.447%	11 △	0	2
	COBOL	HCIPDB01	40.553 ▲	2.222%	10 △	1	2
	COBOL	HCAPDB01	43.75 ▲	2.222%	8	1	2
	COBOL	HCP1BA01	45.718 ▲	2.5%	8	1	6
	COBOL	HCAPDB02	44.853 ▲	2.326%	7	1	2
	COBOL	HCP1BI01	47.16 ▲	2.778%	6	1	4

Items per page: 20 1-7 of 7 items 1 of 1 pages

The Artifact Composition table shows the list of all program artifacts within a transaction. The table is sorted based on the analysis of high risk to modify. The rank of risk to modify is calculated based on combination of the following criteria:

- Whether the artifact is in risk areas: Risk areas are based on the thresholds setting.
- Whether the artifact is an outlier: Outlier artifacts are analyzed based on the outlier detection algorithm leveraging Mahalanobis distance of each artifact to the center of the distribution. Artifacts with large Mahalanobis distances mean that they are outlier or abnormal because they are very far away from majorities.
- Metrics values sorting: Given the same risk area and outlier status, artifacts are finally sorted by the values of Maintainability Index, Unreachable Code, and Cyclomatic Complexity.

21. On the Artifact Composition table, you and Jane notice that **HCIPDB01** is at medium risk to modify. Another program **HCAZERRS** is at high risk. Hence making another changes would not make much impact to the zMobile Health Care application.

Outlier Status	Artifact Type	Program Name	Maintainability Index	Unreachable Code	Cyclomatic Complexity	Number Of Incoming References	Number Of Outgoing References
Outlier	COBOL	HCAZERRS	5.721 ▲	38.696% ▲	24 ▲	5	0
	COBOL	HCP1PL01	33.892 ▲	7.447% ▲	11 ▲	0	2
	COBOL	HCIPDB01	40.553 ▲	2.222% ▲	10 ▲	1	2

You can click on the **Risk Areas** icon () to update the threshold settings and see the effect of the changes. For example, move the lower slider of Maintainability Index threshold to 40%. You now see that the warning icons of the programs with Maintainability Index higher than 40% are changed from red warning icons to transparent yellow warning icons.

All Workbooks / zMobile Health Care / zMobile / Artifact Composition

Artifact Composition

Project: zMobile | Region: CSR1 | Transaction: HCP1 | Date: 10/20/20, 2:17 PM

ADI recommends reviewing the top-ranked artifacts first as they are the potentially riskiest to modify. They are ranked based on factors such as threshold metrics values. [Learn more](#)

Outlier Status	Artifact Type	Program Name	Maintainability Index	Unreachable Code	Cyclomatic Complexity	Number Of Incoming
Outlier	COBOL	HCAZERRS	5.721 ▲	38.696% ▲	24 ▲	
	COBOL	HCP1PL01	33.892 ▲	7.447% ▲	11 ▲	
	COBOL	HCIPDB01	40.553 ▲	2.222% ▲	10 ▲	
	COBOL	HCAPDB01	43.75 ▲	2.222% ▲	8 ▲	
	COBOL	HCP1BA01	45.718 ▲	2.5% ▲	8 ▲	
	COBOL	HCAPDB02	44.853 ▲	2.326% ▲	7 ▲	
	COBOL	HCP1BI01	47.16 ▲	2.778% ▲	6 ▲	

Risk Areas X

The combination of red ranges on the threshold setting slider below defines a risk area.

You can create different risk areas by sliding thumbnails for highlighting metric values and ranking artifacts by the risk to modify.

Maintainability Index Threshold
 40

 85

Unreachable Code Threshold
 10%

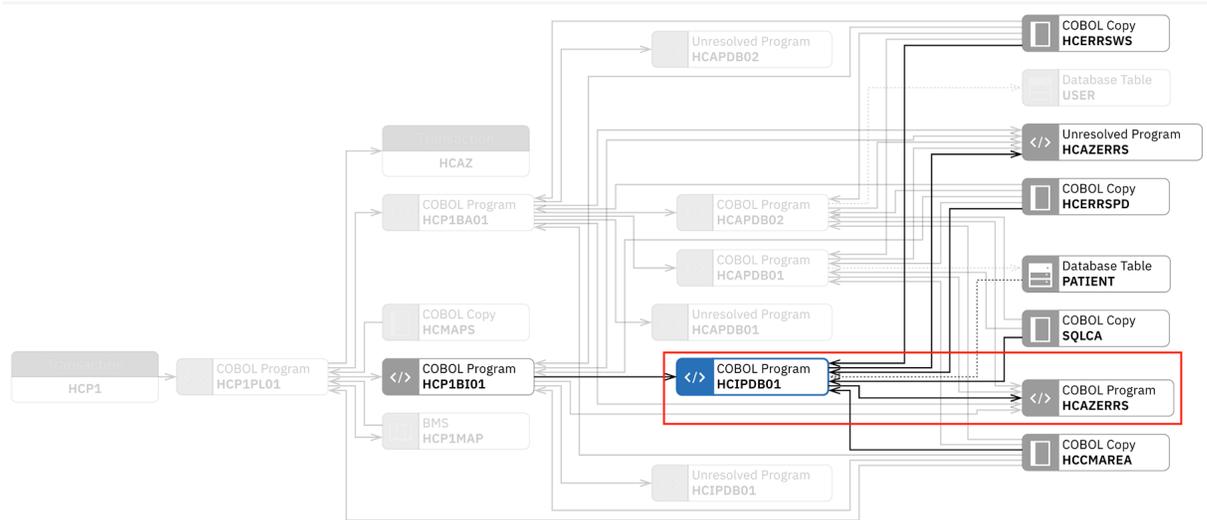
 20%

Cyclomatic Complexity Threshold
 10

 50

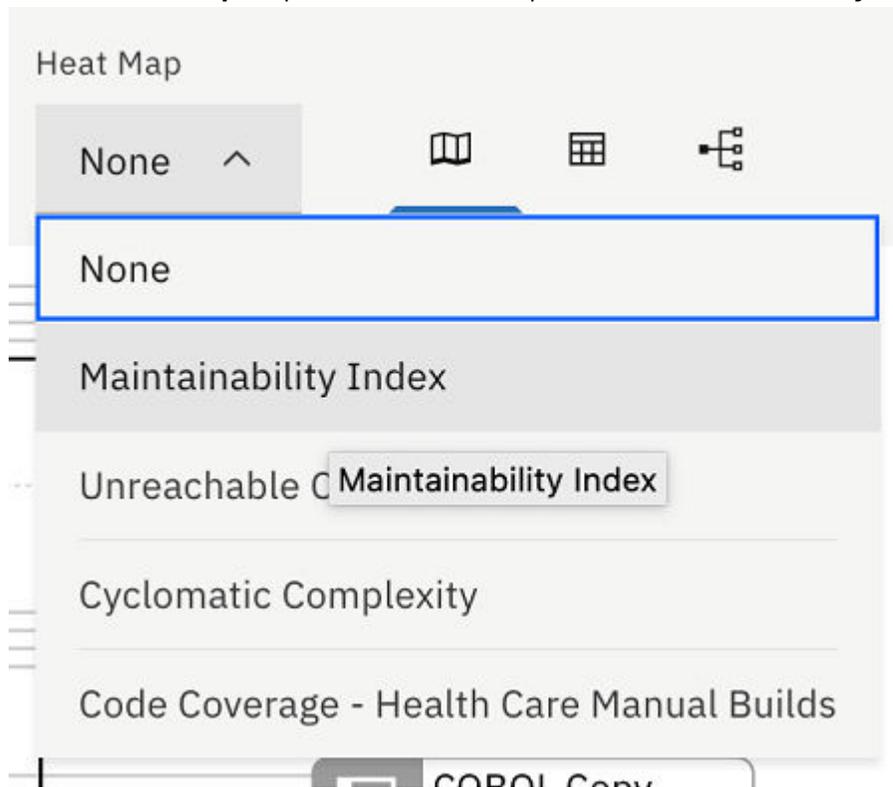
Items per page: 20 | 1-7 of 7 items | 1 of 1 pages

22. Click the **Graph View** () icon on the top right menu to go back to the Artifact Composition graph view.
23. Click on the **HCIPDB01** box on Artifact Composition graph view to only shows the relationship of artifacts related to HCP1BA01.
24. Click **Close (X)** icon to close the information dialog box to get the full view of the Artifact Composition graph.



You and Jane notice that **HCIPDB01** has dependency with **HCAZERRS** which has high risk to modify. These two programs have a very strong impact. Jane needs to test both module very well to make sure that after fixing the SQL problems, her changes to **HCIPDB01** would not cause any regressions to **HCAZERRS**.

25. Click the **Heat Map** drop-down list on the top and select **Maintainability Index**.



The nodes in the graph are displayed with color. The color indicates the threshold status of metric that you have selected. For example, **HCIPDB01** and **HCAZERRS** both are in red, which indicates that their maintainability index is in an unacceptable value. You need to pay close attention when you modify the code.

26. Click the **Close (X)** icon to close the HCIPDB01 information dialog box to get full view of the graph.

Analyzing the static analysis metrics from the Application Discovery data provider

This tutorial guides you through the static analysis of projects within the portfolio by analyzing project data from the Application Discovery data provider.

Before you begin

Before you begin this tutorial, perform the following activities.

1. [“Installing and setting up ADDI Extensions” on page 5](#)
2. [“Generating sample data” on page 106](#)

Note: Since the data in this tutorial is the generated sample data, you might not see the exact same data on your reports.

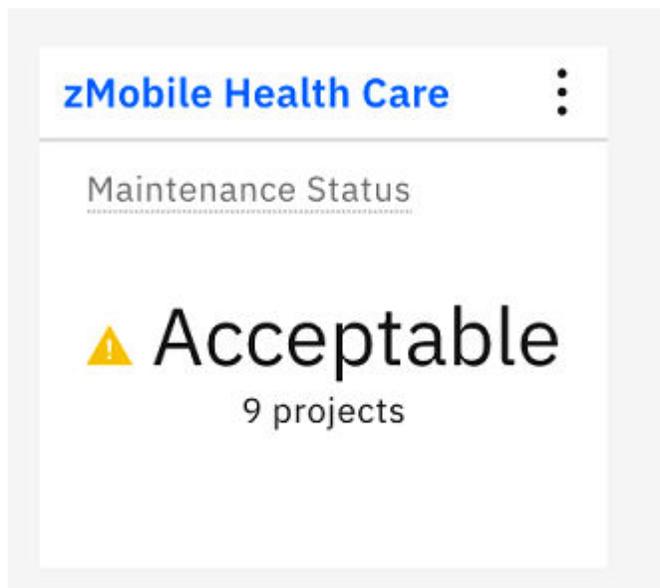
Procedures

In this tutorial, you play the role of Marco, the development lead of the **zMobile Health Care** application, to monitor and analyze various project metrics within your portfolio. The **zMobile Health Care** application is the core application of your company. You and Scott, the enterprise architect, want to ensure that the **zMobile Health Care** application has a low maintainability because you expect that a number of changes are coming soon. To have a better understanding of the code quality of **zMobile Health Care** application, you need to gather the static analysis metrics data from the Application Discovery provider and use ADDI Extensions as dashboards to monitor the code quality of all the projects within the application.

As Marco, you need to complete the following steps to analyze **zMobile Health Care** application:

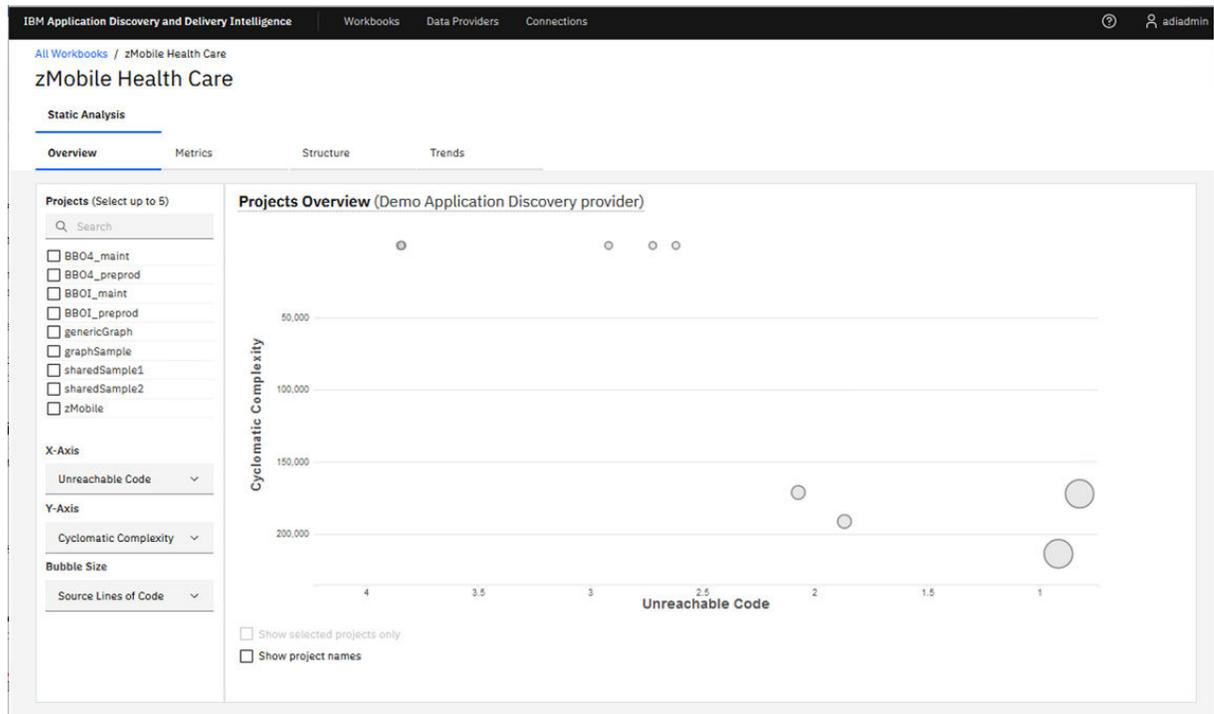
1. Open your Firefox browser and navigate to <https://healthcare.example.com:9753/addi/web/workbook>.
2. Log in with the following credentials. The **Workbooks** page is displayed with the dashboards that summarize the overall status of **zMobile Health Care** application.
 - Email address: adiadmin@healthcare.example.com
 - Password: `adiadmin`

You can see that the average maintenance status of all the projects within the **zMobile Health Care** application is at an acceptable level. But you want to review the individual project to see whether any of the projects need your attention.



- Click **zMobile Health Care** to view the detailed analysis. The **zMobile Health Care** page is displayed with the Static Analysis data. The information is organized into four subtabs: **Overview**, **Metrics**, **Structure**, and **Trends**.
 - The **Overview** tab is displayed with the bubble chart as the summary of the overall status for all projects within the scope of a workbook. You can select an individual bubble on the bubble chart to browse the detailed reports of that project.
 - The **Metrics** tab is displayed with the radar chart of different metrics that you are interested in. You can compare up to five projects along with the metrics.
 - The **Structure** tab is displayed with the information about the shared resources for selected projects.
 - The **Trends** tab is displayed with the historical trends of different metrics that you are interested in. You can compare up to five projects for each of the historical trends chart.

In the **Project Overview** area on the **Overview** tab, you can see a group of bubbles on the upper left of the chart. The bubbles represent the projects that have a higher value of *Unreachable Code* than the rest of the projects. *Unreachable Code* is the percentage of statements within subroutines that cannot be reached by the control flow out of the total number of statements.

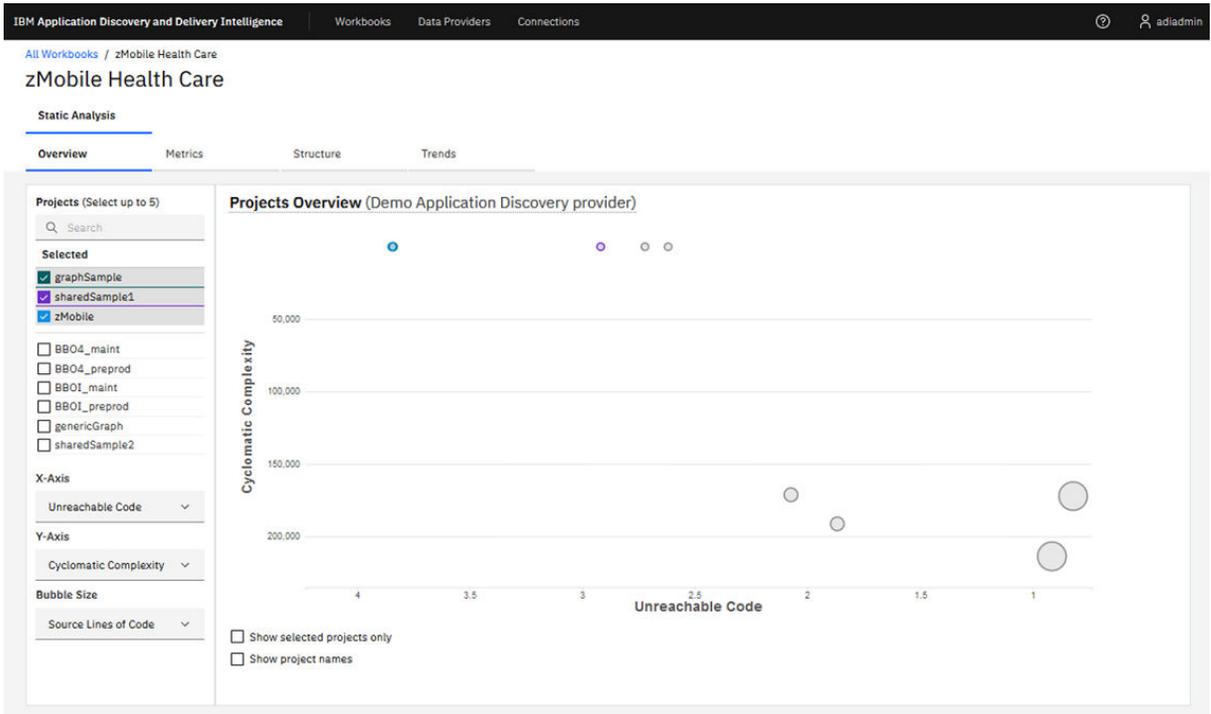


Note: You can see only eight bubbles because the **zMobile** and **graphSample** projects overlap.

The bubbles on the lower right of the chart represent the projects with a higher *Cyclomatic Complexity*. These projects seem to be larger and more complex but with a lower *Unreachable Code*.

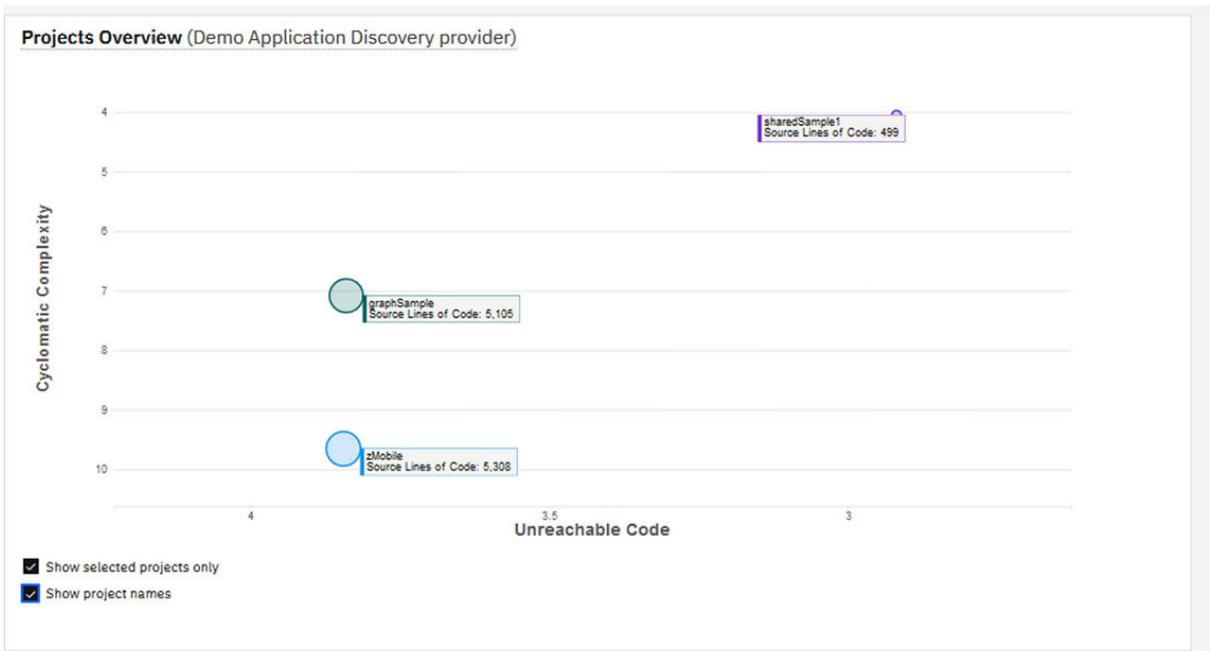
For this tutorial, you need to investigate the projects with a higher *Unreachable Code* because your organization's goal is to have the *Unreachable Code* less than 2% in the entire application.

- Select the **graphSample**, **sharedSample2**, and **zMobile** check boxes for projects on the left pane. The bubbles that represent the selected projects are highlighted in color.



Note: You cannot see the **graphSample** project because it overlaps with the **zMobile** project. You can see it in the next step.

5. Select the **Show selected projects only** and **Show projects name** check boxes. The bubbles for those projects that are not selected disappear and the project names are displayed next to the bubbles.



6. Click the bubble that represents the **zMobile** project. More static analysis reports for the project are displayed at the bottom.

zMobile Project Trends

[View Project Details](#)

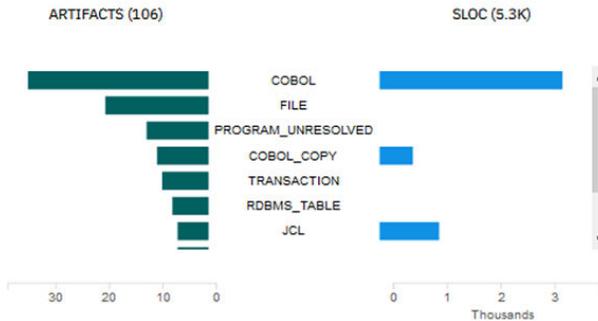
Unreachable Code Trend



Total and Active Statements Trend



Artifact and Lines of Code Breakdown



Maintainability Index vs Delivered Bugs



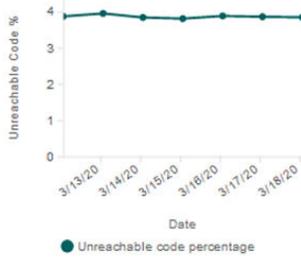
In this example, when you review the reports, you notice that both total statements and active statements are added to the **Total and Active Statements Trend** chart on 16 March. However, the unreachable code percentage remains the same on the **Unreachable Code Trend** chart, which means that no new unreachable code is added to the **zMobile** project for the past week.

7. Scroll up to the Project Overview bubble chart and select the bubble that represents the **graphSample** project.
8. Scroll down to the more detailed reports for the **graphSample** project. The **Unreachable Code Trend** chart remains stable but both total statements and active statements are going down, which means that the development team removes the active statements code from their programs but has not yet cleaned up the unreachable code.

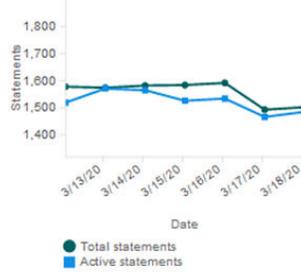
graphSample Project Trends

[View Project Details](#)

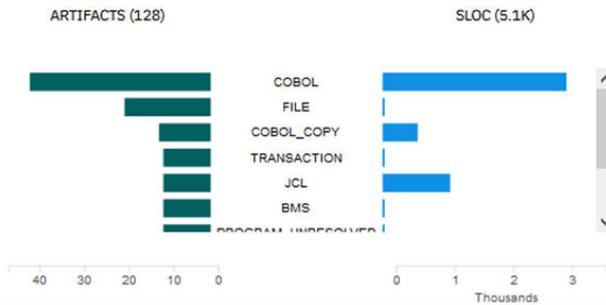
Unreachable Code Trend



Total and Active Statements Trend



Artifact and Lines of Code Breakdown



Maintainability Index vs Delivered Bugs



9. Scroll up and select the **Metrics** tab to compare the project-level metrics of those selected projects.

All Workbooks / zMobile Health Care

zMobile Health Care

Static Analysis

Overview **Metrics** Structure Trends

Projects (Select up to 5)

Search

Selected

- graphSample
- sharedSample1
- zMobile
- BBO4_maint
- BBO4_preprod
- BBO1_maint
- BBO1_preprod
- genericGraph
- sharedSample2

Axis 1

Unreachable Code

Axis 2

Cyclomatic Complexity

Axis 3

Source Lines of Code

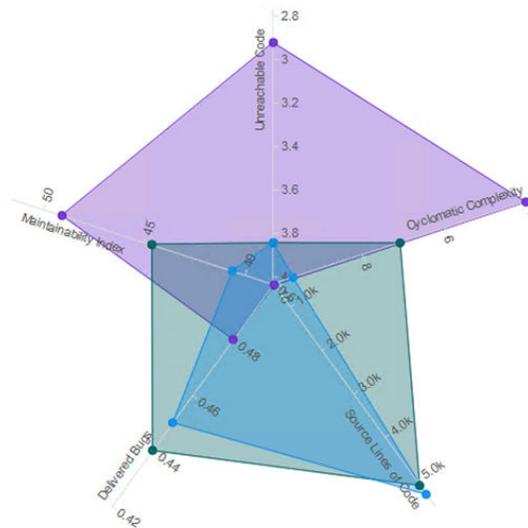
Axis 4

Delivered Bugs

Axis 5

Maintainability Index

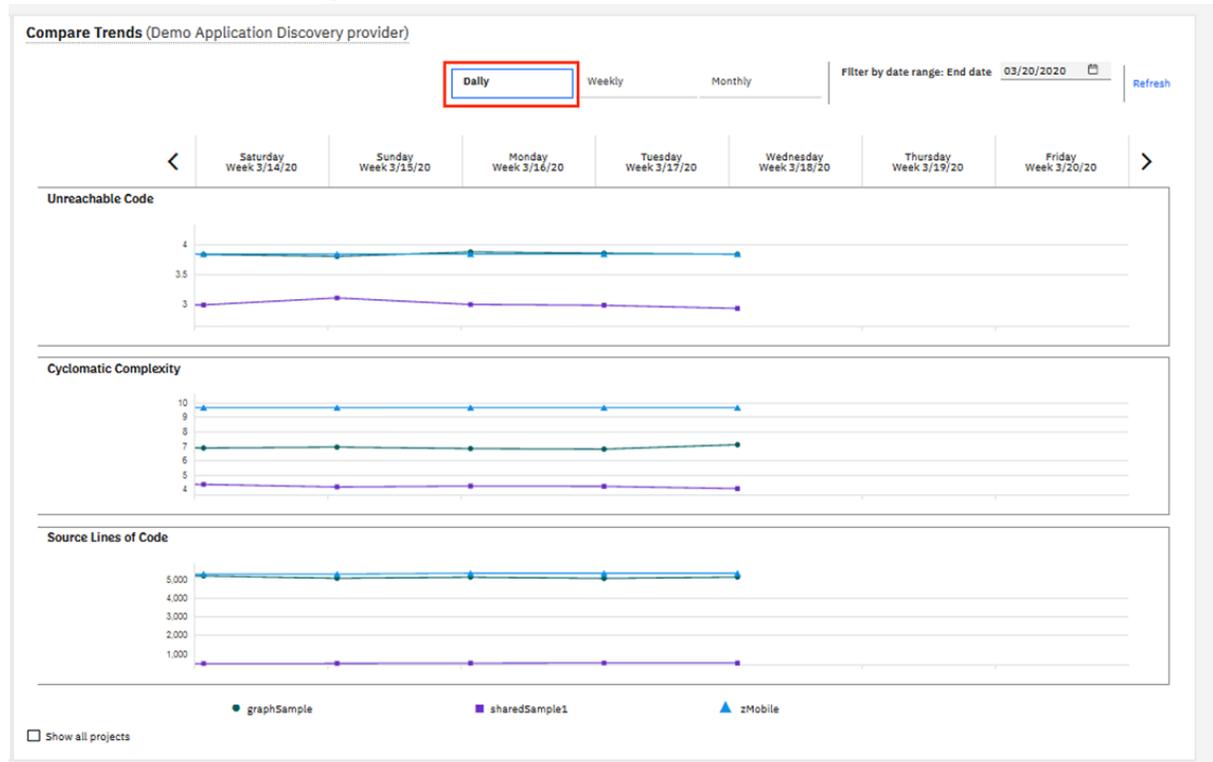
Compare Project-Level Metrics (Demo Application Discovery provider)



On the **Metrics** tab, you can see the radar chart with five axes that represent **Unreachable Code**, **Cyclomatic Complexity**, **Source Lines of Code**, **Delivered Bugs**, and **Maintainability Index**.

The radar chart shows that, for all three projects, the **Maintainability Index** is less than 50, which is lower than the organization's goal. You need to investigate on all these three projects to improve the **Maintainability Index**. The **SharedSample1** project seems to be better than the other two projects in all dimensions. However, if you look at the **Source Lines of Code**, the size of this project is much smaller than the other two projects.

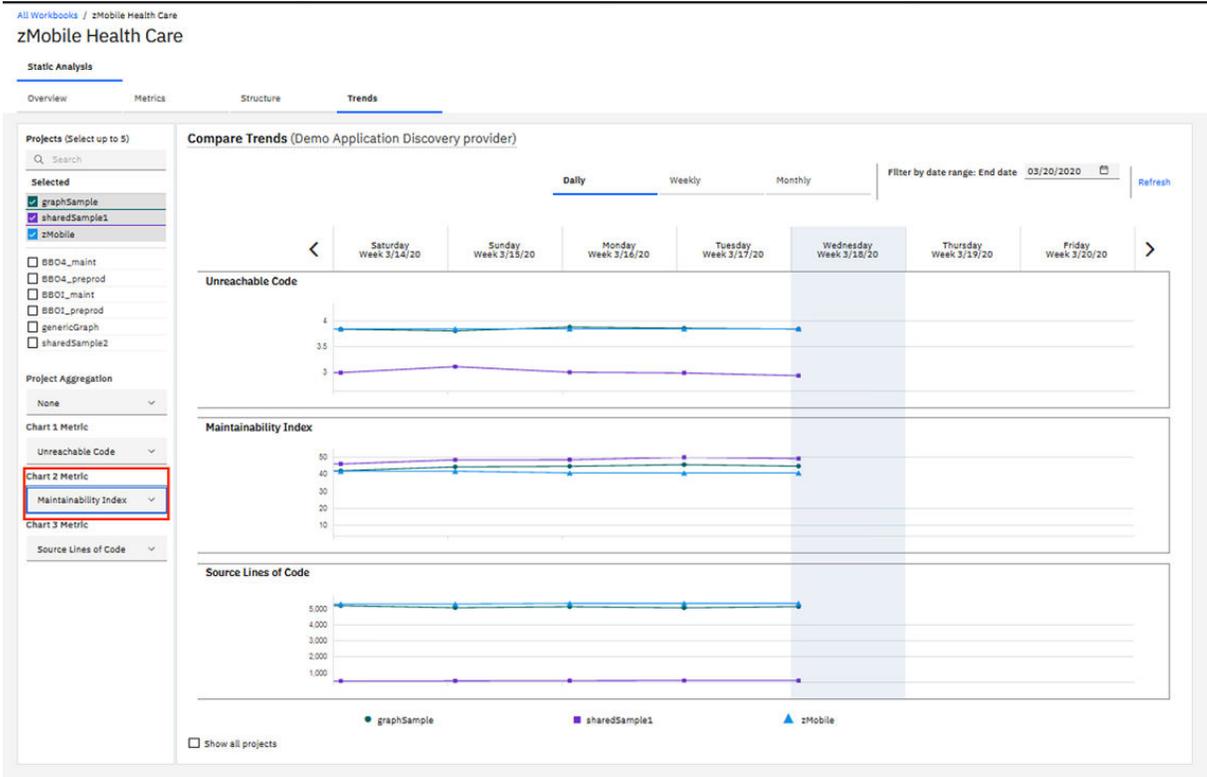
10. Select the **Trends** tab to view the historical data of project-level metrics. By default, the trends for **Unreachable Code**, **Cyclomatic Complexity**, and **Source Lines of Code** are displayed.
11. Select **Daily** to display the graph in a daily view.



You can see that all trend metrics (**Unreachable Code**, **Cyclomatic Complexity**, and **Source Lines of Code**) for all selected projects do not have many changes except that the trends of **Unreachable Code** and **Cyclomatic Complexity** for the **sharedSample1** project are decreasing. This is a good sign for the **sharedSample1** project.

From the left pane, you can choose different metrics to be displayed and select up to five projects to compare trends for different projects.

12. Select **Maintainability Index** from the **Chart 2** Metric drop-down list. You can notice that the **Maintainability Index** of all three projects is under 50, which is lower than the organization's goal. The **graphSample** project has the lowest **Maintainability Index** but the trend is slightly increasing. The **Maintainability Index** of the **sharedSample1** project is also going upwards.



To increase the **Maintainability Index** of all three projects to be over 50, first you probably need to remove the unreachable code from those projects.

Before you remove the unreachable code, you need to use ADDI Extensions to check the data sources that are shared among those projects to make sure that they are consistent when you change the source code.

13. Select the **Structure** tab. The information of data sources that are used by multiple projects within the scope of the workbook is displayed. ADDI Extensions refers those shared data sources as Shared Resources.

zMobile Health Care

Static Analysis

Overview

Metrics

Structure

Trends

Projects (Select up to 5)

Selected

- graphSample
- sharedSample1
- zMobile

- BBO4_maint
- BBO4_preprod
- BBO1_maint
- BBO1_preprod
- genericGraph
- sharedSample2

Resource Type Filters

- Database Table
- Dataset
- IMS Database

Shared Resources (Demo Application Discovery provider)

Find Shared Resources

Name	Type	Number of Projects ↓
<input type="checkbox"/> DB2.V10R1M0.SDSNLOAD	DATASET	3
<input type="checkbox"/> PATIENT	RDBMS_TABLE	3
<input type="checkbox"/> INS_RATES	RDBMS_TABLE	2
<input type="checkbox"/> USER93.ZMOBILE.COPYLIB	DATASET	2

Items per page: 10 | 1-4 of 4 items | 1 of 1 pages

 Show selected resources only

14. Select all the shared resources from the Shared Resources list. You can notice that DB2.V10R1M0.SDSNLOAD and PATIENT are shared by the three projects. If you want to update the codes from one of the three projects that impact PATIENT database table, you need to consider the impact to the source codes of the other two projects.

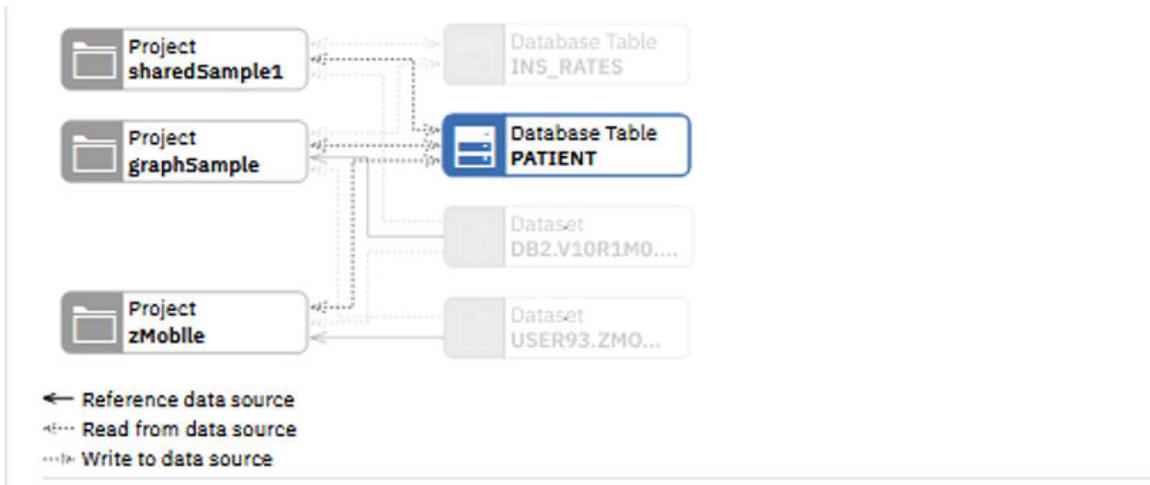
Find Shared Resources

Name	Type	Number of Projects ↓
<input checked="" type="checkbox"/> DB2.V10R1M0.SDSNLOAD	DATASET	3
<input checked="" type="checkbox"/> PATIENT	RDBMS_TABLE	3
<input checked="" type="checkbox"/> INS_RATES	RDBMS_TABLE	2
<input checked="" type="checkbox"/> USER93.ZMOBILE.COPYLIB	DATASET	2

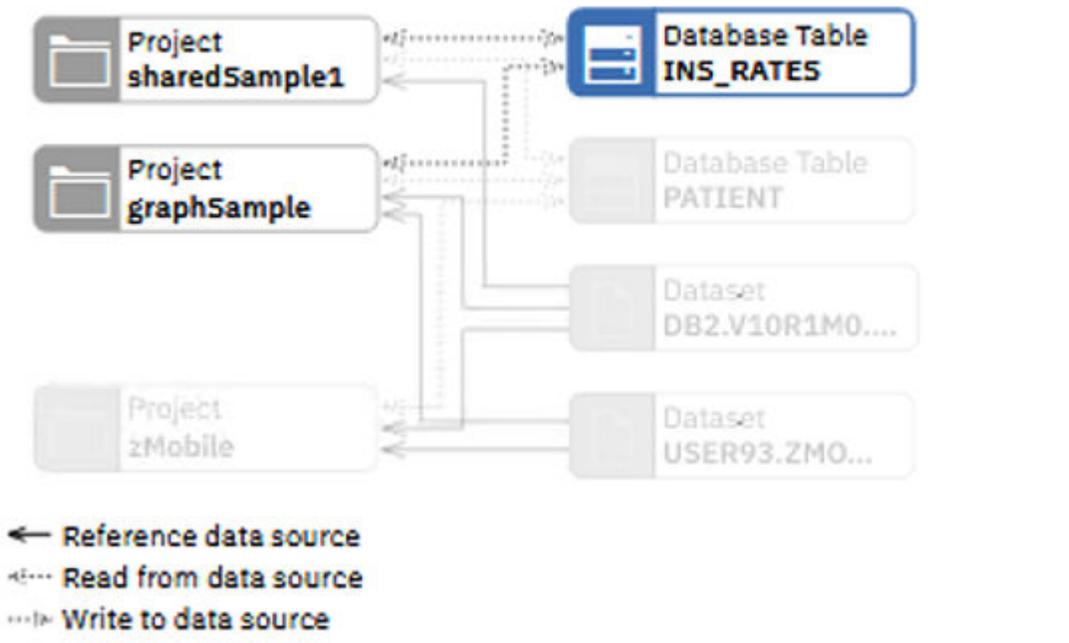
Items per page: 10 | 1-4 of 4 items | 1 of 1 pages

 Show selected resources only

15. Select the PATIENT node from the network graph on the right. You can see that PATIENT are shared by all three projects.



16. Select the INS_RATES node from the network graph on the right. You can see that INS_RATES are shared by the **sharedSample1** and **graphSample** projects.



Now you have explored how you can use ADDI Extensions to monitor the quality of source code by analyzing the static analysis metrics from IBM Application Discovery. Through the shared resources, you can understand the dependencies between different Application Discovery projects better, which helps to ensure that the changes you make for one project are consistent with those of all the dependent projects.

Chapter 4. Installation and setup

You can use ADDI installer to install and set up ADDI Extensions component.

Hardware and software requirements

Before installation, verify that your hardware and software meets the minimum requirements for installing and using ADDI Extensions.

A 64-bit operating system (Windows or Linux®) and a minimum of 8 GB server memory provide the best environment for running the ADDI Extensions component without other IBM Application Discovery (AD) components.

Notes:

- On the Linux platform, ADDI Extensions should be installed and configured by using a **non-root** user.
- When you install ADDI Extensions with additional AD components, you need to combine the minimum hardware and software requirements from both ADDI Extensions and the additional AD components. For specific details about other AD components, see [AD Installation Prerequisites](#).

A complete list of system requirements is as follows.

System requirements for ADDI Extensions server

Operating systems

• Linux

- Red Hat® Enterprise Linux (RHEL) Server 8, IBM Z
- Red Hat Enterprise Linux (RHEL) Server 8, x86-64
- Red Hat Enterprise Linux (RHEL) Server 7, IBM Z
- Red Hat Enterprise Linux (RHEL) Server 7, x86-64
- SUSE Linux Enterprise Server (SLES) 12, IBM Z
- SUSE Linux Enterprise Server (SLES) 12, x86-64
- Ubuntu 16.04 LTS, IBM Z
- Ubuntu 16.04 LTS, x86-64
- Ubuntu 18.04 LTS, IBM Z
- Ubuntu 18.04 LTS, x86-64

• Windows

- Windows 10 Enterprise, x86-64
- Windows 10 Pro, x86-64
- Windows Server 2012 R2 Datacenter Edition, x86-64
- Windows Server 2012 R2 Standard Edition, x86-64
- Windows Server 2016 Datacenter Edition, x86-64
- Windows Server 2016 Standard Edition, x86-64
- Windows Server 2019 Datacenter Edition, x86-64
- Windows Server 2019 Standard Edition, x86-64

Application servers

- Embedded WebSphere® Application Server (Liberty Profile 8.5.5)

Databases

- Embedded Apache Derby 10.10 (limited to 10 end users)
- DB2® Standard Edition 11.5.5.x

Important: Based on your operating system, it is recommended to download and use one of the following images from the [Db2® download page](#)

- Windows 64 bit, x86, on AMD64 and INTEL EM64T
- Linux 64 bit, x86-64 on AMD64 and INTEL EM64T
- Linux 64 bit, System Z, System z9® or zSeries

System requirements for ADDI Extensions client

Browsers

- Mozilla Firefox ESR 65.0 and later versions
- Google Chrome 70 and later versions

Business Rule Discovery data provider

- IBM Application Discovery for IBM Z V5.1 and later versions.

Security considerations

You can take actions to ensure your secure installation, customize security settings, and set up user access controls. You can also ensure that you know about any security limitations that you might encounter with this application.

Enabling security during the installation process

ADDI Extensions supports only HTTPS connections. All communications with IBM Application Discovery for IBM Z (IBM AD) components are HTTPS connections. You should always use Secure Sockets Layer (SSL) HTTPS connections to browse to the application from the browser. SSL encrypts all data that is passed over an HTTPS connection. During installation, ADDI Extensions security is enabled by default.

User Authentication

For more information, see [“Authentication setup” on page 154](#).

Ports, protocols, and services

The default port for ADDI Extensions is 9443. However, on the production environment where you deploy ADDI Extensions with other IBM Application Discovery components, you need to change the port number to avoid conflicts. Because IBM AD Web Services also uses port 9443. Also, the following ports cannot be used since they are occupied by other components:

- Port 9444 is used by Elasticsearch Front Server.
- Port 7600 is used by Authentication Service (DEX).

Before you assign a port to ADDI Extensions, make sure the port doesn't conflict with any other services running on the server. For more information, see [TCP port requirements](#).

Customizing your security settings

After the installation, you need to configure a certificate authority that is signed or self-signed certificate for your application server. You need to send instructions to your end users about how to import this certificate to avoid warnings of errors in their browsers.

- For more information, see [“Installing a security certificate into Liberty” on page 163](#) and [Enabling SSL communication in Liberty](#).

Note: SSL keystore and certificates are included in the Authentication Server (DEX) but for only evaluation purposes. You need to replace the pre-packaged SSL keystore and certificate with your own SSL keystore and certificate for your production environment.

Privacy policy considerations

This software offering does not use cookies or other technologies to collect personally identifiable information.

Installing and setting up ADDI Extensions with ADDI installer

IBM Application Discovery and Delivery Intelligence for IBM Z (IBM ADDI) provides an ADDI installer with which you can install IBM Application Discovery for IBM Z (IBM AD) components and ADDI Extensions component.

Note: On the Linux platform, ADDI Extensions should be installed and configured by using a **non-root** user.

Complete the following steps to install and set up ADDI Extensions with ADDI installer:

1. Download the ADDI installer and run the ADDI installer wizard.
2. Follow the instructions in the wizard to install ADDI.
 - a. On the **Welcome** page, click **Next**.
 - b. Review the **Licensing Agreements** page. Then, select **I accept the terms of this license agreement** and click **Next**.
 - c. On the **Installation Path** page, specify the installation path, and then click **Next**. The default installation path is C:\Program Files\IBM Application Discovery and Delivery Intelligence.

If the installation path that you specify does not exist, the target directory is created. Confirm the path, and click **OK** in the Message dialog box.

- d. On the **Select Installation Components** page, select the components that you want to install, and then click **Next**. The components that are not applicable for the current system cannot be selected.

Note: For ADDI Extensions, you need to install at least the following components:

- IBM Application Discovery Servers and Services
- Authentication Server (DEX)
- IBM Application Discovery and Delivery Intelligence Extensions for *your system (Windows, Linux or zLinux)*

To perform Static Analysis or Business Rule Discovery, you also need to install additional corresponding AD components, such as IBM Application Discovery Batch Server and IBM AD Web Services. For more information about the relationship among different components, see [IBM AD High-Level Architecture Overview](#).

Note: Make sure that the IBM Application Discovery Batch Server is installed on the same machine as the ADDI Extensions component.

- e. Click **Next** on the **Information** page.
 - f. On the **IBM Application Discovery and Delivery Intelligence Extensions Installation Path** page, specify the installation path, and then click **Next**.

If the installation path that you specify does not exist, the target directory will be created. Confirm the path, and click **OK** in the Message dialog box.

Note: The default installation path is C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611.

- g. On the **User Data** page, clear the **Opening IBM Application Discovery Configuration Wizard** checkbox.
 - h. Click **Next** on the **Installation** page when the installation progress is finished.
 - i. On the **Setup Shortcuts** page, select the shortcuts that you want to create, and then click **Next**.
 - j. Click **Done** on the **Installation Finished** page.
3. Create DB2 database as described in [“Creating the database” on page 142](#).
 4. Configure the `dex.yaml` file as described in [“Configuring the parameters in the dex.yaml file” on page 155](#).

Notes:

- Make sure the fully qualified hostname that is configured for Dex Authentication Service issuer url is same as what is configured for ADDI Extensions server base url.
 - When you configure **addi-liberty staticClients** in the `dex.yaml` file, make sure the **-id** attribute has no spaces at the beginning of the line. For more information, see [“Configuring the parameters in the dex.yaml file” on page 155](#).
5. Start Authentication Server (DEX) as described in [“Starting the Authentication Server \(DEX\)” on page 159](#).
 6. Specify the configuration properties as described in [“Configuring ADDI Extensions Install Configuration Page” on page 152](#).
 7. Perform adi-setup as described in [“Running the adi-setup script” on page 162](#).
 8. Update the ADDI Extensions `server.xml` as described in [“Updating server.xml for communication between ADDI Extensions server and Authentication Server \(DEX\)” on page 161](#).
 9. Start ADDI Extensions server as described in [“Starting up the server” on page 151](#).

Creating the database

You need to create a database before you set up the application.

If you are running against a database other than the default Derby database that is included in the installation, you need to create a database before you set up the application.

Review [the prerequisites](#) and make sure that your DB2 database meets the requirements before you continue.

Before you create the DB2 databases for ADDI Extensions, you must set the `DB2_COMPATIBILITY_VECTOR` registry variable with the following commands. Otherwise, ADDI Extensions will not be able to perform paged queries correctly and will show error messages in various UI pages.

```
db2set DB2_COMPATIBILITY_VECTOR=MYS
db2stop
db2start
```

Then, you can create the required databases by using the following DB2 commands:

```
db2 create database DW using codeset UTF-8 territory en PAGESIZE 16384
```

Optional: Data Warehouse (DW) that is used for raw data storage can be large. If you plan to perform massive DB2 update transactions on the DW, it is recommended to increase the default sizes of DW transaction logs. You can use the following commands to increase the primary and secondary log sizes:

```
db2 get db cfg for DW
db2 update db cfg for DW using LOGPRIMARY 64
db2 update db cfg for DW using LOGSECOND 192
```

Note: The total number of primary logs plus secondary logs can not exceed 256. It is also recommended to set a larger secondary log size since they are constantly cleared by DB2.

For more information about DB2 installation, see [IBM DB2 10.5 for Linux, Unix and Windows documentation](#).

Uninstalling IBM ADDI Extensions

Learn how to uninstall ADDI Extensions.

Prerequisite: Stop the ADDI Extensions server as described in [“Shutting down the server”](#) on page 151.

To uninstall ADDI Extensions, go to the installation path and delete the subfolder where the ADDI Extensions installation files are stored. For example, if you used the default installation path to install ADDI Extensions on a Windows system, you just need to delete the `adi<version_number>` folder under the `C:\Program Files\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions` directory to uninstall ADDI Extensions.

Note: The actual installation path for your installed ADDI Extensions depends on what you specified during the installation. For more information, see [“Installing and setting up ADDI Extensions with ADDI installer”](#) on page 141.

Chapter 5. Migrating from a previous release

Find the instructions about how to upgrade and migrate data from a previous release based on your migration scenario.



Attention: The Elasticsearch data migration for V6.0.2 is not supported. Your saved code snippets for the Business Rule Discovery workbook will be impacted and unavailable.

Before the migration, select one of the following migration scenarios based on your needs and then check the detailed instructions in the corresponding section.

- [“Migrating from a previous release of ADI” on page 145](#)
- [“Migrating from a previous release of ADDI Extensions” on page 148](#)

Migrating from a previous release of ADI

If you use a previous release of ADI and want to migrate all of its data (except for the user and user groups information) to the ADDI Extensions 6.1.1, you must follow the migration steps in this section instead of the other installation scenarios. You can run `adi-setup` application to automate the steps for migrating ADI data.

Notes:

1. The following instructions assume that you had either installed a previous version in either of the following locations. If you installed ADI in different locations, you can substitute the path when necessary.

Linux: `/opt/ibm/adi/`
Windows: `C:\IBM\adi\`
2. The following instructions use `adi_previous` as an installation directory name to differentiate the previous version from the new version.
3. ADDI Extensions no longer uses Jazz® Team Server (JTS). Some repositories that are stored in the JTS data warehouse, such as user and user group information, will not be migrated.
4. The database migration from a previous releases is supported for only production DB2 databases. Evaluation and demonstration data that used Derby is not supported for migration.
5. Those workbooks that use the sample demonstration data that is generated in the previous releases cannot be migrated.

Preparation before migration

Delete those workbooks that used the demonstration provider and the demonstration connection that are generated by the Sample Demo Data Generation feature, as well as the sample demonstration provider and the sample demonstration connection in the previous releases before you start the data migration.

Complete the following steps to migrate from a previous release of ADI by using the `adi-setup` application.

1. Stop the current release of ADI by using the following commands.

Linux:

```
cd /opt/ibm/adi/server
./server.shutdown
```

Windows:

```
cd C:\IBM\adi\server
server.shutdown
```

2. Optional: Back up your data warehouse as described in [“Backing up data” on page 163](#).

3. Rename the installation directory to make sure that you can identify where the previous version is installed.

Linux:

```
cd /opt/ibm
mv adi adi_previous
```

Windows:

```
cd C:\IBM
ren adi adi_previous
```

4. Download the NEW release of ADDI and install ADDI Extensions component into the directory you want. By default, the new ADDI Extensions component is installed in either of the following locations:

Linux: /opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/*adi611*

Windows: C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions*adi611*

5. Use a command window to navigate to the new installation directory.

Linux:

```
cd "/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server"
```

Windows:

```
cd "C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server"
```

Note: For Windows, you must open the command window as the administrator. Right-click the **Command Window** icon and select **Run as administrator** option. The header of the command window shows as "Administrator".

6. Run the following command and wait for the command to complete:

Linux:

```
./adi-setup migration -fromDir /opt/ibm/adi_previous
```

Windows:

```
adi-setup migration -fromDir C:\IBM\adi_previous
```

When the migration command completes, all of your *adi_previous* server data are copied over successfully to *adi611* server and ready for the new server to migrate them.

Note: Since the underlying technology for ADDI Extensions is changed, the *adi_previous* server configuration files are not applied to *adi611* server.

7. The *adi_previous* server configuration files are backed up for your reference under *adi611* server. Verify that the following configuration files are backed at the given locations:

Linux:

```
/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/conf/adi-legacy/server.xml
/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/conf/adi-legacy/teamserver.properties
/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/conf/adi-legacy/tdb
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server\conf\adi-legacy\server.xml
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server\conf\adi-legacy\teamserver.properties
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server\conf\adi-legacy\tdb
```

8. Configure the Authentication Service (DEX) as described in [“Authentication setup” on page 154](#). With the change in the underlying technology, a few new components are now included in ADDI Extensions, such as the Authentication Service (DEX). This means that the existing user accounts have been deprecated and the authentication and authorization are now supported through Authentication Service (DEX).
9. Configure IBM Application Discovery Configuration Service as described in [“Configuring ADDI Extensions Install Configuration Page” on page 152](#).

Notes:

- If you need to set up the *adi611* server similar to your *adi_previous* server then you can refer the properties, such as server hostname, port number, and database information, from the backed-up configuration files (*server.xml* or *teamserver.properties*) and use the same information when you set up *adi611* server.
 - If the data warehouse database is used from a previously migrated release of ADI, the default tablespace from that database will be used regardless of the specified setting of the Database Tablespace Folder. Because the Database Tablespace Folder field is mandatory, you can specify anything in case of the default tablespace from the migrated database. If a non-default tablespace was used, then enter that tablespace location in this Database Tablespace Folder field.
10. Set up the ADDI Extensions server as described in [“Running the adi-setup script” on page 162](#).
 11. After the ADDI Extensions setup is complete, you can start the *adi611* server as described in [“Starting up the server” on page 151](#). When you start the *adi611* server, it executes additional migration operations automatically. An administrator can review the successful progress of these operations in the log files that are generated on the server machine.
 12. Observe the progress of the migration from the log file that is in either of the following locations:

Linux:

```
IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/logs/console.log
```

Windows:

```
c:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server\logs\console.log
```

For example, you can observe it by opening it in a text editor and frequently reloading it or using the tail command on Linux.

13. Wait until the following line shows in the log file:

```
ADI RDF migration succeeded without error.
```

14. Also, review previous lines that are shown for status messages about the successful migration of ADDI Extensions rest server connections, data providers, and application settings. If there are errors, review the error message to understand the nature of the problem. You can also start over by redoing the steps that are described in this guide with a new fresh installation. Contact technical support to get help with any migration error messages.
15. When the migration is shown as successful in the log, you can notify your users that the new server is available for use.
16. Open the browser and navigate to ADDI Extensions home page at `https://<hostname>:<port>/addi/web`. Then you are automatically redirected to DEX authentication server and the **Log in to**

your account page is displayed. You can now log in as any user that is a member of the User Groups configured with ADDI Extensions or the static admin user (if this is an evaluation copy).

Note: The home page URL has been updated for ADDI Extensions.

Migrating from a previous release of ADDI Extensions

If you use a previous release of ADDI Extensions and want to migrate all of its data to ADDI Extensions 6.1.1, you must follow the migration steps in this section instead of the other installation scenarios.

Notes:

1. The following instructions assume that you had either installed a previous release in either of the following locations. If you installed ADDI Extensions in different locations, you can substitute the path when necessary.

Linux: /opt/IBM Application Discovery and Delivery Intelligence/IBM Application Delivery Intelligence/

Windows: C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\

2. The database migration from a previous releases is supported for only production DB2 databases. Evaluation and demonstration data that used Derby is not supported for migration.
3. Those workbooks that use the sample demonstration data that is generated in the previous releases cannot be migrated.

Preparation before migration

Delete those workbooks that used the demonstration provider and the demonstration connection that are generated by the Sample Demo Data Generation feature, as well as the sample demonstration provider and the sample demonstration connection in the previous releases before you start the data migration.

Complete the following steps to migrate ADDI Extensions.

1. Stop the current release of ADDI Extensions as described in the [“Shutting down the server” on page 151](#) topic.
2. Stop the IBM Application Discovery Configuration Service, the Authentication Server (DEX), and all other ADDI services, including but not limited to the following ones:
 - IBM Application Discovery Analyze Service
 - IBM Application Discovery Batch Service
 - IBM Application Discovery File Service
 - IBM Application Discovery Graph DB Service
 - IBM Application Discovery Mainframe Projects Service
 - IBM Application Discovery Manual Resolution Service
 - IBM Application Discovery Search Service
3. Back up the current installed ADDI Extensions from the following installed directory to `adi_previous` directory.

Linux:

```
/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Delivery Intelligence/
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Delivery Intelligence\
```

4. Back up the DEX configuration files from the following directory.

Linux:

```
/opt/IBM Application Discovery and Delivery Intelligence/Authentication Server (Dex)/conf/
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\Authentication Server (Dex)\conf\
```

5. Optional: Back up your data warehouse as described in [“Backing up data” on page 163](#).

Uninstall IBM Application Discovery and Delivery Intelligence and IBM Application Discovery Configuration Service.

Note: Make sure that the following path is deleted and all the IBM Application Discovery services are no longer available.

Linux:

```
/opt/IBM Application Discovery and Delivery Intelligence/
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\
```

6. Download the NEW release of ADDI and install the ADDI Extensions component into the directory you want. By default, the new ADDI Extensions component is installed in either of the following locations:

Linux:

```
/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611
```

7. Copy the DEX configuration files from the previous release that you back up and paste into the following directory.

Linux:

```
cd "/opt/IBM Application Discovery and Delivery Intelligence/ Authentication Server (Dex)/conf/"
```

Windows:

```
cd "C:\IBM\IBM Application Discovery and Delivery Intelligence\ Authentication Server (Dex)\conf\"
```

8. Configure IBM Application Discovery Configuration Service as described in [“Configuring ADDI Extensions Install Configuration Page” on page 152](#) by using the same configuration values as the previously installed ADDI Extensions, such as server hostname, port number, and database information.

9. Set up the ADDI Extensions server as described in [“Running the adi-setup script” on page 162](#).
10. Use a command window to navigate to the new installation directory.

Linux:

```
cd "/opt/IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server"
```

Windows:

```
cd "C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server"
```

Note: For Windows, you must open the command window as the administrator. Right-click the **Command Window** icon and select **Run as administrator** option. The header of the command window shows as "Administrator".

11. Run the following command and wait for the command to complete:

```
./adi-setup migration -fromDir backup_directory_for_previous_ADDI_Extensions
```

Note: You will need to replace *backup_directory_for_previous_ADDI_Extensions* with the directory where you back up the previous version of ADDI Extensions (*adi_previous* directory).

When the migration command completes, all of your data are copied over successfully to *adi611* server and ready for the new server to migrate them.

12. After the ADDI Extensions migration is complete, you can start the *adi611* server as described in [“Starting up the server” on page 151](#). When you start the *adi611* server, it executes additional migration operations automatically. An administrator can review the successful progress of these operations in the log files that are generated on the server machine.
13. Observe the progress of the migration from the log file that is in either of the following locations:

Linux:

```
IBM Application Discovery and Delivery Intelligence/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/logs/console.log
```

Windows:

```
C:\IBM\IBM Application Discovery and Delivery Intelligence\IBM Application Discovery and Delivery Intelligence Extensions\adi611\server\logs\console.log
```

For example, you can observe it by opening it in a text editor and frequently reloading it or using the tail command on Linux.

14. Wait until the following line shows in the log file:

```
ADI RDF migration succeeded without error.
```

15. Review the previous lines that are shown for status messages about the successful migration of ADDI Extensions rest server connections, data providers, and application settings. If there are errors, review the error message to understand the nature of the problem. You can also start over by redoing the steps that are described in this guide with a new fresh installation. Contact technical support to get help with any migration error messages.
16. When the migration is shown as successful in the log, you can notify your users that the new server is available for use.
17. Open the browser and navigate to ADDI Extensions home page at <https://<hostname>:<port>/addi/web>. Then you are automatically redirected to DEX authentication server and the **Log in to your account** page is displayed. You can now log in as any user that is a member of the User Groups configured with ADDI Extensions or the static admin user (if this is an evaluation copy).

Chapter 6. Administration

This section describes the administration information and administrator's tasks for ADDI Extensions.

Starting up the server

Run the `server.startup` script to start the server.

To start the server, complete the following steps:

1. Open a terminal/command prompt and navigate to the `<Installed ADDI Extensions folder>/adi611/server` directory by running the following command.

```
cd <Installed ADDI Extensions folder>/adi611/server
```

2. In the current directory, start the server by running the following command.

```
server.startup
```

Shutting down the server

Run the `server.shutdown` script to shut down the server.

To shutdown the server, complete the following steps:

1. Open a terminal/command prompt and navigate to the `<ADDI_Extensions_installation_path>/adi<version_number>/server` directory by running the following command.

```
cd <ADDI_Extensions_installation_path>/adi<version_number>/server
```

2. In the current directory, shut down the server by running the following command.

```
server.shutdown
```

Server configuration settings

Administrator can update the startup script to configure the ADDI server behaviors when the server is starting up.

The following table describes the variables that define the behavior of those startup scripts.

Table 1. Variables defined in startup scripts

Script File Name	Property Name / Parameter Option	Default Value	Description
<code><addi_installed_directory>/server/server.startup</code>	-Xmx	4G	Determines the maximum memory that is allocated to ADDI Extensions. Note: When you set up ADDI Extensions on a machine with inadequate resources for evaluation, you can set the maximum memory allocation to 2G.
	-Xms	4G	Determines the initial memory that is allocated to ADDI Extensions. Note: When you set up ADDI Extensions on a machine with inadequate resources for evaluation, you can set the maximum memory allocation to 2G.
	-Xmn	1G	Determines the minimum memory that is allocated to ADDI Extensions.
<code><addi_installed_directory>/elasticsearch/config/jvm.options</code>	-Xmx	2G	Determines the maximum memory that is allocated to Elasticsearch.
	-Xms	2G	Determines the initial memory that is allocated to Elasticsearch.

Configuring ADDI Extensions Install Configuration Page

Learn how to specify the configuration properties in the IBM ADDI Extensions Install Configuration page.

Complete the following steps to configure the IBM ADDI Extensions Install Configuration page.

1. Launch the **IBM Application Discovery Configuration Service Admin**.
 - On a Windows system, select **Start Menu > IBM Application Discovery and Delivery Intelligence > Launch IBM Application Discovery Configuration Service Admin**.

- On a Linux system, use a terminal to navigate to the `<addi_installed_directory>/IBM Application Discovery Configuration Service` directory and run the following commands to launch the **IBM Application Discovery Configuration Service Admin**.

```
startServer.sh
startWebServerUI.sh
```

2. Select **Configure > Install Configurations > IBM Application Discovery and Delivery Intelligence Extensions Install Configuration**. The **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page is displayed.
3. Specify the following properties in the **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page.

<i>Table 2. Properties in the IBM Application Discovery and Delivery Intelligence Install Configuration page</i>		
Tab	Name	Description
Web and Application Server	Base URL	<p>The base URL of the ADDI Extensions server without any path information. For example, if you want to reach ADDI Extensions with a url such as <code>https://addi.mycompany.com:9753/addi/web</code> in the future, enter the url <code>https://addi.mycompany.com:9753</code>.</p> <p>Make sure that the ADDI Extensions server machine is configured with this name and the fully qualified name is configured in your DNS so that it can be reached inside your organization like that. For more information, see Planning URLs.</p> <p>The default port for ADDI Extensions is 9443. However, on the production environment where you deploy ADDI Extensions with other IBM Application Discovery components, you need to change the port number to avoid conflicts. Because IBM AD Web Services also uses port 9443. Also, the following ports cannot be used since they are occupied by other components:</p> <ul style="list-style-type: none"> • Port 9444 is used by Elasticsearch Front Server. • Port 7600 is used by Authentication Service (DEX). <p>Before you assign a port to ADDI Extensions, make sure that the port doesn't conflict with any other services running on the server. For more information, see TCP port requirements.</p>
Databases	Database server type	You need to create a Data Warehouse (DW) database before configuring ADDI Extensions. For an evaluation setup, you can use the Derby by default. The Derby database will be created automatically if it does not exist. If you are setting up ADDI Extensions with DB2 for production, select DB2 from the list and provide the details.
	Host	This is the IP or fully qualified network name of your DB2 database server.
	Port	This is the port where DB2 can be reached on the DB2 server. The default port for DB2 is 50000.
	Database Name	The name of the database. It is recommended to use the name Data Warehouse (DW) for the databases to be created.

Table 2. Properties in the IBM Application Discovery and Delivery Intelligence Install Configuration page (continued)		
Tab	Name	Description
	Tablespace Folder	A folder that has been created on the DB2 server machine in which table spaces can be created during setup. The DB2 Admin user must have full write access to this folder. Note: If the data warehouse database is used from a previously migrated release of ADDI Extensions, the default tablespace from that database will be used regardless of the specified setting of the Tablespace Folder. Because the Tablespace Folder field is mandatory, you can specify anything in case of the default tablespace from the migrated database. If a non-default tablespace was used, then enter that tablespace location in this Tablespace Folder field.
	Username	The username of the DB2 Admin user that was created on the DB2 server machine.
	Password	The password for DB2 Admin user. Note: This password also needs to be specified when you run the <code>adi-setup</code> application.
Authentication Services	Host	This is the IP or fully qualified network name of Authentication Server (DEX).
	Port	The port that is used for the Authentication Server.
	HTTP protocol	The protocol configured for this Authentication Server. Select HTTP or HTTPS.
User Groups	Admin Group List	Enter a comma-separated list of user groups that will have <i>Administrator privileges</i> on the configured ADDI Extensions server.
	User Group List	Enter a comma-separated list of user groups that will have <i>User privileges</i> on the configured ADDI Extensions server. This list of user groups will be made available for each ADDI Extensions workbook such that the admin can select a sublist to provide them access to this workbook.

Authentication setup

Check the configuration steps that are needed to start and run the Authentication Service (DEX).

Authentication Server (DEX) is used for the authentication and authorization of ADDI Extensions. The *Dex Authentication Service* is an identity service that uses OpenID Connect and supports OAuth2 protocol to support clients that use SSO authentication within ADDI Extensions. With the credentials provided by the user, it interrogates a Secure Storage through the LDAP protocol. The **Secure Storage** can be an **Active Directory** or any other entity that stores users and groups and can communicate through **LDAP**. The authentication service is mandatory to be installed and configured for ADDI Extensions.

When you first access ADDI Extensions through the browser, you are automatically redirected to the Dex login page where you can enter your email address and password to log in to ADDI Extensions. The credentials reach the Authentication Service (DEX) and it checks, using the LDAP protocol, whether the credentials of the user are bounded to an account in **Secure Storage**. The **Secure Storage** can be an **Active Directory** or any other entity that stores users and groups and can communicate through LDAP.

When the authentication finishes, an authorization process is started to determine whether the currently authenticated user is a member of a user group that has access to ADDI Extensions.

When you are authenticated and authorized, you can access the workbooks that you are authorized to, and start the analysis.

Note: The Authentication Service is based on DEX and provides an authentication solution, which connects through LDAP to **Secure Storage**. For more information about DEX, see [Authentication through LDAP](#).

Configuring the parameters in the dex.yaml file

On the machine where Authentication Server is installed, navigate to `<addi_installed_directory>/Authentication Server (DEX)/sample-conf/` and copy the `dex.yaml` file under this directory to the `<addi_installed_directory>/Authentication Server (DEX)/conf/` directory. Open the `dex.yaml` file by using a text editor, and enter the values that you want for the following properties as described in detail.

Note: The parameters are represented in `.yaml` as strings terminated by a trailing colon. Values are represented by a string following the colon, which is separated by a space. Example:

```
my_parameter: my_value
```

1. Set the **issuer** parameter as follows.

- a. If the communication to and from DEX is done through **https**, the **issuer** parameter has the following format:

Note: This step implies the use of certificates. If you want to set the communication to be secured, make sure that a certificate authority issues a signed certificate (`.crt`) and a private key for the certificate (`.key`).

```
https://<machine name where DEX is installed>.<machine domain>:<port>/dex
```

Example:

```
issuer: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

- b. If the communication to and from DEX is done through **http**, the **issuer** parameter has the following format:

```
http://<machine name where DEX is installed>.<machine domain>:<port>/dex
```

Example:

```
issuer: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
```

2. The next section can be configured as follows:

- a. If the communication to and from DEX is done through **https**, generate the TLS certificates for Authentication Server (DEX) and add the paths for the certificate (`.crt`) and the key (`.key`) files in the `TLSCert` and `TLSKey` fields. The default port is 7600.

```
storage:
  type: sqlite3
  config:
    file: dex.db
frontend:
  theme: addi
web:
  https: 0.0.0.0:7600
  TLSCert: C:\certs\dex.crt
  TLSKey: C:\certs\dex.key
```

- b. If the communication to and from DEX is done through **http**, comment the `TLSCert` and `TLSKey` fields.

```
storage:
  type: sqlite3
```

```

config:
  file: dex.db
frontend:
  theme: addi
web:
  http: 0.0.0.0:7600
  #TLSCert:
  #TLSKey:

```

3. The **skipApprovalScreen** parameter can be set to true or false. The true value offers the possibility to skip the "Grant access screen" after the user logs in.

```

oauth2:
  skipApprovalScreen: true

```

4. The **connectors** section can be used to configure LDAP as follows:

- a. Set the **type** parameter to **ldap** and provide a name and id for this connector.
- b. Under the **config** section:

Set the **host** parameter, including the default port 389 or 636. The **host** parameter has the following format:

```
host: << IP:PORT >>
```

Example:

```

config:
  host: WIN-NSSMI7A1KJQ.ferdinand2.com:636

```

- c. The **insecureNoSSL** parameter can be set as follows:

- If the **host** parameter was set to use the default port 389, set the **insecureNoSSL** to true.

```
insecureNoSSL: true
```

- If the **host** parameter was set to use the default port 636, set the **InsecureNoSSL** to false.

```
insecureNoSSL: false
```

- d. Set the **bindDN** parameter by adding the account that has the rights for the LDAP bind action.

Note: To add the account that has rights for LDAP bind action, run `adsiedit.msc` on the Active Directory machine and load the current domain. Right click *CN=Users* and *CN=Administrator*, select **Properties** and search for **distinguishedName** attribute. For more information, see [ADSI Edit \(adsiedit.msc\)](#).

```

# This would normally be a read-only user.
bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com

```

- e. Set the **bindPW** parameter by adding the account's password that has the rights for the LDAP bind action.

```
bindPW: password
```

- f. Do not modify the value of the **usernamePrompt** parameter.

```
usernamePrompt: email address
```

- g. Under the **userSearch** section, modify the values to map the username and password that are entered by a user to an LDAP entry.

Note: The **baseDN** parameter contains the base distinguished name of all **User Accounts**.

```

userSearch:
  baseDN: dc=ferdinand2,dc=com
  filter: "(objectClass=person)"
  username: userPrincipalName
  # "DN" (case sensitive) is a special attribute name. It indicates that

```

```
# this value should be taken from the entity's DN not an attribute on
# the entity.
idAttr: DN
emailAttr: userPrincipalName
nameAttr: cn
```

h. Under the **groupSearch** section, modify the values to query for groups given a user entry.

Note: The **baseDN** parameter contains the base distinguished name of the groups in LDAP registry.

```
groupSearch:
  baseDN: cn=Users,dc=alpaca,dc=com
  filter: "(objectClass=group)"
  # A user is a member of a group when their DN matches
  # the value of a "member" attribute on the group entity.
  userAttr: DN
  groupAttr: member
  # The group name should be the "cn" value.
  nameAttr: cn
```

i. The **staticClients** section should be configured to add the ADDI Liberty client as follows:

```
staticClients:
  - id: addi-liberty
    redirectURLs:
      - 'https://<addi_hostname>:<addi_port>/oidcclient/redirect/addi-liberty'
    name: 'ADDI Liberty Server'
    secret: password
```

Where:

- **id** is the generic name that is given for the **IBM ADDI Liberty Client**.
- **redirectURLs** takes as value the **fully qualified IP** and a generic **port** that is used for callback to Authentication Service (DEX).
- **name** takes the value **ADDI Liberty Server**.
- **secret** is a secret that is shared among application.

Configuration Examples:

- When the communication to and from DEX is done through **https**, the `dex.yaml` file is configured as follows:

```
issuer: https://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
storage:
  type: sqlite3
  config:
    file: dex.db

frontend:
  theme: addi

web:
  https: 0.0.0.0:7600
  TLSCert: C:\certs\dex.crt
  TLSKey: C:\certs\dex.key

oauth2:
  skipApprovalScreen: true

connectors:
  - type: ldap
    name: ADLDAP
    id: ldap
    config:
      host: WIN-NSSMI7A1KJQ.ferdinand2.com:636

  # No TLS for this setup.
  insecureNoSSL: false

  # This would normally be a read-only user.
  bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com
  bindPW: Admin15_

  usernamePrompt: email address
```

```

userSearch:
  baseDN: dc=ferdinand2,dc=com
  filter: "(objectClass=person)"
  username: userPrincipalName
  # "DN" (case sensitive) is a special attribute name. It indicates that
  # this value should be taken from the entity's DN not an attribute on
  # the entity.
  idAttr: DN
  emailAttr: userPrincipalName
  nameAttr: cn

groupSearch:
  baseDN: dc=ferdinand2,dc=com
  filter: "(objectClass=group)"
  # A user is a member of a group when their DN matches
  # the value of a "member" attribute on the group entity.
  userAttr: DN
  groupAttr: member

  # The group name should be the "cn" value.
  nameAttr: cn

staticClients:
- id: addi-liberty
  redirectURIs:
  - 'https://WIN-NSSMI7A1KJQ.ferdinand2.com:9443/oidcclient/redirect/addi-liberty'
  name: 'ADDI Liberty Server'
  secret: password

```

- When the communication to and from DEX is done through **http**, the `dex.yaml` file is configured as follows:

```

issuer: http://WIN-ASK7V692EKB.ferdinand2.com:7600/dex
storage:
  type: sqlite3
  config:
    file: dex.db

frontend:
  theme: addi

web:
  https: 0.0.0.0:7600
  TLSCert:
  TLSKey:

oauth2:
  skipApprovalScreen: true

connectors:
- type: ldap
  name: ADLDAP
  id: ldap
  config:
    host: WIN-NSSMI7A1KJQ.ferdinand2.com:389

  # No TLS for this setup.
  insecureNoSSL: true

  # This would normally be a read-only user.
  bindDN: CN=Administrator,CN=Users,DC=ferdinand2,DC=com
  bindPW: Admin15_

  usernamePrompt: email address

userSearch:
  baseDN: dc=ferdinand2,dc=com
  filter: "(objectClass=person)"
  username: userPrincipalName
  # "DN" (case sensitive) is a special attribute name. It indicates that
  # this value should be taken from the entity's DN not an attribute on
  # the entity.
  idAttr: DN
  emailAttr: userPrincipalName
  nameAttr: cn

groupSearch:
  baseDN: dc=ferdinand2,dc=com

```

```

filter: "(objectClass=group)"
# A user is a member of a group when their DN matches
# the value of a "member" attribute on the group entity.
userAttr: DN
groupAttr: member

# The group name should be the "cn" value.
nameAttr: cn

staticClients:
- id: addi-liberty
  redirectURIs:
  - 'https://WIN-NSSMI7A1KJQ.ferdinand2.com:9443/oidcclient/redirect/addi-liberty'
  name: 'ADDI Liberty Server'
  secret: password

```

5. If you want to use ADDI Extensions for only evaluation or demonstration purposes, you might also populate **staticPasswords** section to create the admin-like authority as follows:

- a. Generate the bcrypt hash of a password by using `adi-setup bcryptPassword -dex .password <password>` and save this bcrypt hash to be used in the following step.
- b. Uncomment the **staticPasswords** section and populate as follows:

```

staticPasswords:
- email: <<user email>> (Enter an user email)
  hash: <<bcrypt hash of user password>> (Enter the bcrypt hash from step a)
  username: <<user name>> (Enter an user name)
  userID: <<user id>> (Enter an unique user id)

```

Example of the **staticPasswords** section:

```

staticPasswords:
- email: "eg@na.na"
  hash: "$2a$10$uCf8IzfxFSFmSZY1jhDqk0r.AH.B/mNUshRSwT.pGu33h2sN45seu"
  username: "eg"
  userID: "eg"

```

Starting the Authentication Server (DEX)

The Authentication Server (DEX) is required to communicate with ADDI Extensions through https protocol for security purpose. Before you start the Authentication Server (DEX), replace the pre-packaged SSL keystore and certificate with your own SSL keystore and certificate on your production environment.

Complete the following steps to start Authentication Server (DEX).

1. Click **Start**, and select **Run**.
2. Type `services.msc` and start the Authentication Server (DEX).
3. If the service does not start, check the `dex.log` file under `<addi_installed_directory>/Authentication Server/` folder.

Configuring Authentication Server

After the Authentication Server (DEX) is up and running, launch the **IBM Application Discovery Configuration Service Admin** and configure **IBM ADDI Extensions Install Configuration Page** to make Authentication Server (DEX) available for ADDI Extensions.

Complete the following steps to configure Authentication Server in the **IBM ADDI Extensions Install Configuration Page**.

1. Launch the **IBM Application Discovery Configuration Service Admin**.
 - On a Windows system, select **Start Menu > IBM Application Discovery and Delivery Intelligence > Launch IBM Application Discovery Configuration Service Admin**.

- On a Linux system, use a terminal to navigate to the `<addi_installed_directory>/IBM Application Discovery Configuration Service` directory and run the following commands to launch the **IBM Application Discovery Configuration Service Admin**.

```
startServer.sh
startWebServerUI.sh
```

2. Select **Configure > Install Configurations > IBM Application Discovery and Delivery Intelligence Extensions Install Configuration**. The **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page is displayed.
3. Select the **Authentication Service** tab in the **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page.
4. Under the **Authentication Service** tab, add the following information:

Host

Enter the fully qualified hostname of the machine on which Authentication Server (DEX) is running.

Port

Enter the generic port on which the Authentication Server (DEX) is running.

Protocol

Select the protocol (HTTP / HTTPS) that Authentication Server (DEX) is using.

5. Click **Save** to save the parameters.

Configuring User Groups

You can manage users who have access to ADDI Extensions by configuring User Groups in the IBM ADDI Extensions Install Configuration page.

Complete the following steps to configure User Groups in the IBM ADDI Extensions Install Configuration page.

1. Launch the **IBM Application Discovery Configuration Service Admin**.
 - On a Windows system, select **Start Menu > IBM Application Discovery and Delivery Intelligence > Launch IBM Application Discovery Configuration Service Admin**.
 - On a Linux system, use a terminal to navigate to the `<addi_installed_directory>/IBM Application Discovery Configuration Service` directory and run the following commands to launch the **IBM Application Discovery Configuration Service Admin**.

```
startServer.sh
startWebServerUI.sh
```

2. Select **Configure > Install Configurations > IBM Application Discovery and Delivery Intelligence Extensions Install Configuration**. The **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page is displayed.
3. Select **User Groups** tab in **IBM Application Discovery and Delivery Intelligence Extensions Install Configuration** page.
4. Under the **User Groups** tab, enter the following information:

Admin Group List

Enter a comma-separated list of user groups that will have *Administrator privileges* on the configured ADDI Extensions server.

User Group List

Enter a comma-separated list of user groups that will have *User privileges* on the configured ADDI Extensions server. This list of user groups will be made available for each ADDI Extensions workbook so that the admin can select a sublist to provide them access to this workbook.

5. Click **Save** to save the configured values.

Performing ADDI Extensions setup

When you complete the configuration in the IBM ADDI Extensions Install Configuration page, you can follow the steps as described in “[Running the adi-setup script](#)” on page 162 to perform ADDI Extensions setup and allow ADDI Extensions to connect to Authentication Service (DEX) for user authentication and authorization.

After ADDI Extensions setup is complete, open the browser and try to access the ADDI Extensions home page. You will be automatically redirected to DEX authentication server and the **Log in to your account** page is displayed.

You can enter the following information:

Email address

Expects the account's email address that is defined in **Secure Storage**.

Password

Expects the account's password that is defined in **Secure Storage**.

After you click **Log in**, the credentials reach the **Authentication Service (DEX)** and it checks, by using the configured **LDAP protocol**, whether the credentials of the user are bounded to an account in **Secure Storage**.

When the authentication finishes, an authorization process is started to determine whether the currently authenticated user is member of a **User Groups** that has been configured with ADDI Extensions.

After you are authenticated and authorized, you can access the workbooks that you are authorized to and start the analysis.

Updating server.xml for communication between ADDI Extensions server and Authentication Server (DEX)

By default ADDI Extensions communicates with Authentication Server (DEX) through https protocol. But since IBM AD doesn't package any SSL certificates out of the box and the sample `dex.yaml` template file configures the communication of Authentication Server (DEX) through http protocol, you need to configure your ADDI Extensions instance to communicate with DEX through the protocol that you want.

Complete the following steps to update the `server.xml` file to use https protocol to communicate between ADDI Extensions server and Authentication Server (DEX):

1. Open the `server.xml` file in `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer/configDropins/overrides` directory.
2. Find the **openidConnectClient** element in the `server.xml` file.
3. Update the **clientSecret** to be the same value as **secret** that is defined under **addi-liberty staticClients** within the `dex.yaml` file.
4. Save the `server.xml` file.

Complete the following steps to update the `server.xml` file to use http protocol to communicate between ADDI Extensions server and Authentication Server (DEX):

1. Open the `server.xml` file in `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer` directory.
2. Find the **openidConnectClient** element in the `server.xml` file.
3. Remove the **discoveryEndpointUrl** property.
4. Open the `server.xml` file in `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer/configDropins/overrides` directory.

Note: Make sure that the DEX service is running.

5. Find the **openidConnectClient** element in the `server.xml` file.
6. Update the **clientSecret** to be the same value as **secret** that is defined under **addi-liberty staticClients** within the `dex.yaml` file.
7. Within the **openidConnectClient** element in the `server.xml` file, get the url that is defined in the **discoveryEndpointUrl** property.
8. Open the defined url within a browser. You will need the information that is displayed on this page for the next steps.

Note: The url should be similar to `http://<fully_qualified_hostname>:<dex_port>/dex/.well-known/openid-configuration`.

9. Make the following updates to the **openidConnectClient** element in the `server.xml` file.
 - Remove the **discoveryEndpointUrl** property.
 - Add the property **httpsRequired="false"**
 - Add the property **authorizationEndpointUrl** and set its value to what is defined under discovery information in the browser page as **authorization_endpoint**.
 - Add the **tokenEndpointUrl** property and set its value to what is defined under discovery information in the browser page as **token_endpoint**.
 - Add the **jwkEndpointUrl** property and set its value to what is defined under discovery information in the browser page as **jwt_issuer**.
10. Save the `server.xml` file.

Running the adi-setup script

To set up ADDI Extensions with data from a configuration server, you need to run the `adi-setup` script. Complete the following steps to run the `adi-setup` script.

1. Open command window and navigate to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server` directory.
2. Run the `adi-setup` script by using the following options:

```
adi-setup addiConfigurationServer -- Run automated setup using an ADDI Configuration
Server
  [-url https://server.org:443] -- Url to the ADDI Configuration Server. If not
provided it will assume http://localhost:8080.
  [-db.password password] -- password of the data warehouse admin. If you skip this
option the user password will not be used in database connection.
```

References to the adi-setup script

This page provides the documentation of all the options available for `adi-setup` script.

You can also view this documentation in the ADDI Extensions server by following the following instructions:

1. Navigate to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server`
2. Execute `adi-setup help` command to display the `adi-setup` documentation as listed in the following section.

```
adi-setup help -- Get this information.
adi-setup addiConfigurationServer -- Run automated setup using an ADDI Configuration
Server
  [-url https://server.org:443] -- Url to the ADDI Configuration Server. If not provided
it will assume http://localhost:8080.
  [-db.password password] -- password of the data warehouse admin. If you skip this
option the user password will not be used in database connection.
adi-setup migration -- Run a migration from a previous build.
  -fromDir /opt/ibm/adi504 -- An absolute path to the location of the previous
installation. The directory you specify needs to have a server child directory.
  [-verbose true] -- Switches on verbose output for all the operations performed.
adi-setup bcryptPassword -- Generates the bcrypt hash of the password as expected by the
```

```
Dex configuration file.
-dex.password password
  adi-setup encodePassword -- Encodes a string into the format expected by the ADI
teamserver.properties files as well as data collection application properties file.
-db.password password
```

Backing up data

You can back up your data with ADDI Extensions.

Complete the following steps to back up data.

1. Stop the ADDI Extensions server by following the steps in [“Shutting down the server” on page 151](#).
2. Navigate to the directory location of the database that you want to back up and copy the database to a secure location.
3. Right click on ADDI Extensions database and select **Backup**.
4. For the **Media type**, select **File system**, and point to the location for saving the backup.
5. Click **Finish**.
6. Start the ADDI Extensions server by following the steps in [“Starting up the server” on page 151](#).

Installing a security certificate into Liberty

Install a certificate into the WebSphere Application Server (WAS) Liberty server to establish a secure connection.

Complete the following steps to install a security certificate into Liberty server.

1. Configure the WebSphere Liberty Security Certificate in the `server.xml` file that is located under the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer` directory. For more information, see [Enabling SSL communication in Liberty](#).
2. By default, WAS Liberty is configured to look for and read the security certificates from the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer/resources/security` directory. Copy the new `cert.p12` file into this directory.
3. Open the `server.xml` file and update the entries in the following line to point to the new KeyStore file.

```
<keyStore id="defaultKeyStore" location="cert.p12" type="PKCS12" password="<export
password>" />
```

4. Save the changes.
5. (Optional) Encode the password by using WAS Liberty's **securityUtility** command. For more information, see [securityUtility command](#).
6. Restart the ADDI Extensions server.
7. Access the ADDI Extensions application from Chrome and you can see that the connection is secure.

Accessing Business Rule Discovery repository through APIs

ADDI provides Business Rule Discovery (BRD) APIs that allow users to access Business Rule Discovery repository through REST services. You can use the following REST services to read, write, and manage BRD data directly or connect to other third-party tools without using ADDI web UI.

The BRD REST APIs are documented within ADDI build by Swagger UI. After installing ADDI Extensions, you can find the documentation of those APIs by browsing to `https://<addi_hostname>:<addi_port>/addi/brd/swagger-ui.html`, for example, `https://localhost:9443/addi/brd/swagger-ui.html`.

You can find the available APIs in the following list.

- `bt-folder-storage-resource`: Services to add, update, delete, and query the contents of business term hierarchy folders.
- `business-rule-package-storage-resource`: services to add, update business rule packages and their information (for example, associated business terms and snippets), delete or get business rule packages and their information.
- `business-term-storage-resource`: services to add, updated, deleted or get business terms and their information.
- `implementation-name-storage-resources`: services to add, update, delete or get implementation names
- `metric-storage-resources`: services to get keywords paging information
- `project-storage-resource`: services to manage information for IBM Application Discovery (AD) projects.
- `relationship-storage-resource`: services to manage relationship of business terms information
- `snippet-storage-resource`: services to add, update, delete or get snippets and their information
- `tag-storage-resource`: service to manage tags.
- `workbook-storage-resources`: services to manage workbook setting information.

Using APIs out of online document

The BRD APIs can be used out of Swagger UI, such as in another rest client tool or make rest call through program. To use APIs out of Swagger UI, you need username, password and CSRF token along with the REST service URL and the API definition. CSRF token is a must-have header (X-XSRF-TOKEN) if you are making a **POST/PUT/DELETE** request.

To get your CSRF token, you must make a GET call to `https://<addi_hostname>:<addi_port>/addi/brd/api/csrf`

You can find this REST service in `project-storage-resource`.

Using XML instead of JSON

By default APIs return result in JSON. If you need XML as response, you need to set request header "Accept" with value "application/xml".

Generating sample data for evaluation

For ADDI Extensions, you can generate sample OMEGAMON for CICS data in order to evaluate the ADDI Extensions functionality.

Generating OMEGAMON for CICS data

You can learn how to generate OMEGAMON for CICS data as sample data to evaluate the functions of ADDI Extensions.

Complete the following steps to generate OMEGAMON for CICS sample data.

1. Navigate your browser to ADDI Extensions main page. Go to `https://<addi_hostname>:<addi_port>/addi/web/workbook`.
2. Log on with ADDI Extensions administrator user ID and password. The ADDI Extensions home page appears.
3. Click the **Profile** icon on the upper right corner and select **Settings** from the drop-down list to go to the **Settings** page.
4. Under **Sample Data Generation** section, select **OMEGAMON for CICS** from Data Providers drop-down list.
5. Fill in the following information in **Data Generation Settings**. For more information about terms, see Chapter 2, "Terminology," on page 3

Note: For evaluation purpose, you can leave all the default values in the **Data Generation Settings** form.

- **Plex:** Name of a CICS plex that you want to generate data.
 - **Days:** Number of days to create the data.
 - **Goal Response Time:** Goal response time that you want to set as threshold in microseconds.
 - **Execute Warehouse RUNSTATS after data generation:** Select this checkbox.
6. Click **Generate** to generate OMEGAMON for CICS data as sample. Upon finish, you will see the message showing the data generation has completed. The OMEGAMON for CICS data provider with sample data appears in the list of data providers on the **Providers** page. You can then go to the **Workbooks** page to create an analysis workbook by using the generated OMEGAMON for CICS data provider.

Generating static analysis sample data

You can generate static analysis sample data to evaluate the functions of ADDI Extensions.

Complete the following steps to generate sample data of Application Discovery data provider for static analysis.

1. Navigate your browser to ADDI Extensions main page. Go to `https://<addi_hostname>:<addi_port>/addi/web/workbook`.
2. Log on with ADDI Extensions administrator user ID and password. The ADDI Extensions home page appears.
3. Click the **Profile** icon on the upper right corner and select **Settings** from the drop-down list to go to the **Settings** page.
4. Under **Sample Data Generation** section, select **Application Discovery** from Data Providers drop-down list.
5. Fill in the following information in **Data Generation Settings** form. See [Chapter 2, “Terminology,” on page 3](#) for detailed terminology information.

Note: For evaluation purposes, you can leave all the default values in the **Data Generation Settings** form except data points. If you're using Derby as the database for evaluation, you might not want to increase the number of data points to be generated.

Data Points

Number of data points you want to generate. One data point represents one data set that ADDI Extensions collects from AD.

Time Interval

Time interval in hours that you want to simulate the data collection.

Execute Warehouse RUNSTATS after data generation

Select this checkbox.

6. Click the **Generate** button to generate sample data. Upon completion, you will see the message that shows the data generation has completed. The Application Discovery data provider with sample data will appear in the list of data providers on the **Data Providers** page. You can go to the **Workbook** page to create a workbook by using the generated Application Discovery data provider.

Chapter 7. Preparing external data sources

You can learn how to prepare code coverage results for batch applications, CICS, COBOL, PL/I and Java.

Preparing code coverage results for COBOL and PL/I

You can learn how to prepare code coverage results for batch applications, CICS, COBOL, and PL/I.

There are four scenarios for ADDI Extensions headless code coverage collection:

1. RDz v901x - no headless CC - client UI required.
Integrated Debugger (DIRECT) and Debug Tool (TCPIP) are both supported.
2. RDz v91x, v95x - headless on client (Windows, Linux)
Integrated Debugger (DIRECT) and Debug Tool (TCPIP) are both supported.
Headless on z/OS® Debug Tool (TCPIP) only (not relevant for ADDI Extensions).
3. IBM Debug for z Systems v14
Headless on client (Windows, Linux) in Debug Tool Compatibility Mode (TCPIP)
No headless in Standard Mode (DIRECT with UI running)
Headless on z/OS, Debug Tool Compatibility Mode (TCPIP) only (not relevant for ADDI Extensions)
4. IBM Developer for z Systems v14, IBM Developer for z Systems v14 Enterprise Edition
Headless on client (Windows, Linux) in Debug Tool Compatibility Mode (TCPIP)
No headless in Standard Mode (DIRECT with UI running)
Headless on z/OS, Debug Tool Compatibility Mode (TCPIP) only (not relevant for ADDI Extensions)

Preparing code coverage results for batch applications

If you use IBM Developer for z Systems (IDz), you can learn how to generate the code coverage results for batch applications either by using the IDz client or by using the IDz headless mode.

Generating code coverage results by using the IDz client

You can learn how to generate code coverage results by using the IDz client.

Assumptions

Verify the assumptions before you prepare code coverage results by using IDz client.

- The tester or developer uses the IDz client to run the test.
- The program or a set of programs, such as LIST, MAP, XREF, and SOURCE, for which the code coverage results are being gathered should be compiled for debugging.
- The tester or developer starts the IDz with a new or existing workspace, defines a connection to the remote system if necessary, and connects to the remote system.
- The JCL for testing the batch application resides in a set of data sets.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results for batch applications by using IDz client.

1. Navigate to a JCL data set, for example, VENKATU . INS . QUOTE . JCL.
2. Open a batch JCL member file to test the Insurance quote batch application, example: INSQTE1 . jcl.

3. Add a set of statements in the JCL RUN or GO step to execute the batch program in the code coverage mode, for example:

```
//CEE0PTS DO *  
TEST(,,DBM%VENKATU)  
  ENVAR("EQA_STARTUP_KEY=CC,,testid=INSQTE1")  
/*
```

Note:

- a. The above set of statements are additional directives to run or execute the batch program in code coverage mode.
 - b. VENKATU is the user id, which is optional.
 - c. EQA_STARTUP_KEY=CC indicates that the code coverage mode and the testid parameter, INSQTE1, are to pass in the name of the test case or a unique ID to be associated with the generated code coverage results.
4. Save the JCL by right clicking in the Editor or the JCL member and click **Submit**.
 5. Click any of the buttons on the dialog that pops up after the submission:
 - Click **Locate Job** to go to the node where the job status can be viewed.
 - Click **Notify** to update the status bar when the job finishes execution.
 - Click **OK** to close the dialog.

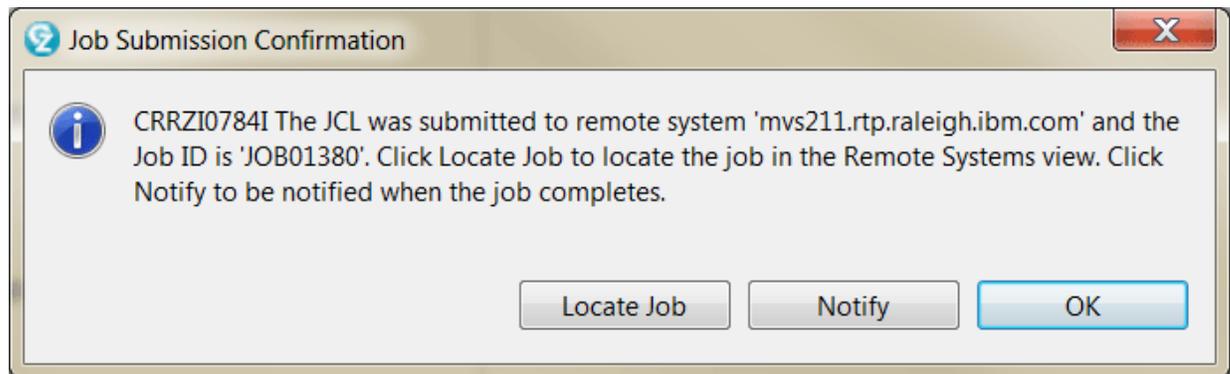


Figure 1. Dialog showing the successful submission of the job

6. On successful execution, view the code coverage results from the current run in the Code Coverage Report.

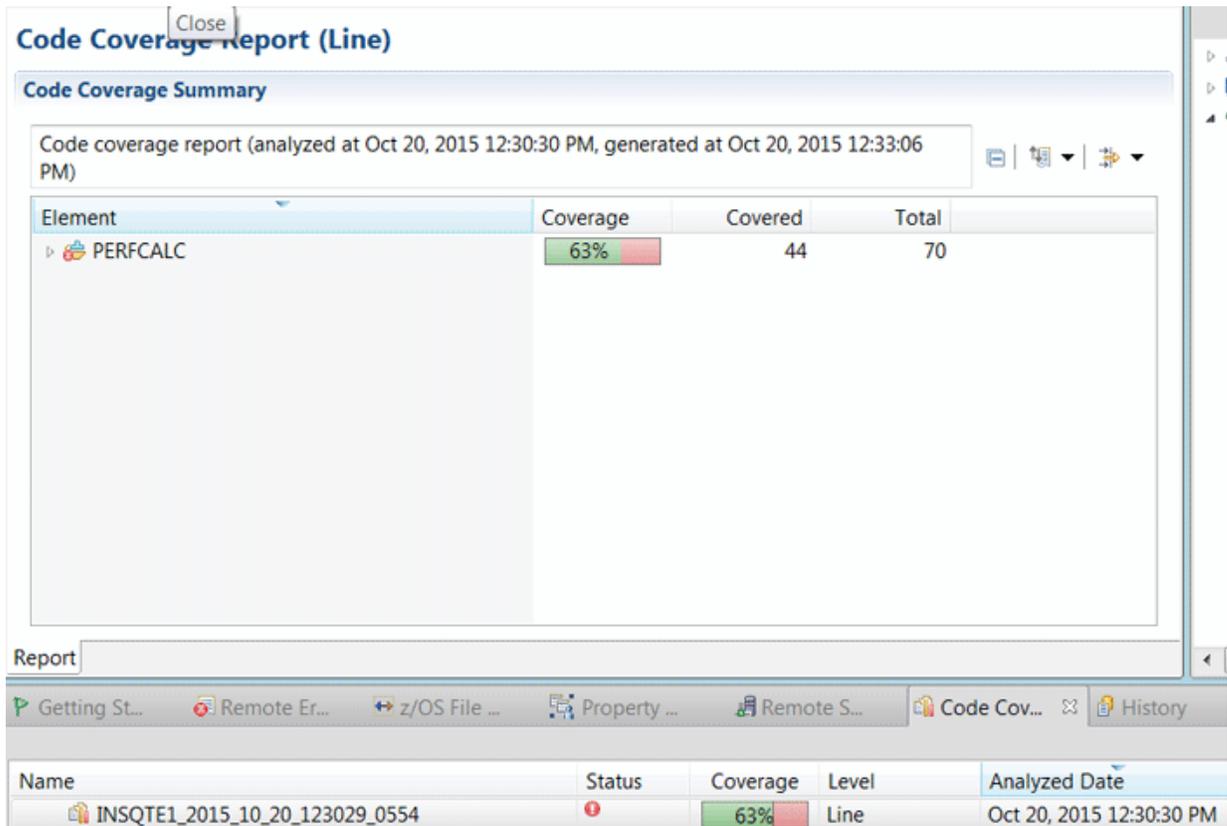


Figure 2. Sample showing the Code coverage result displayed in Code Coverage results view

7. Right click the code coverage result and select **Export**.

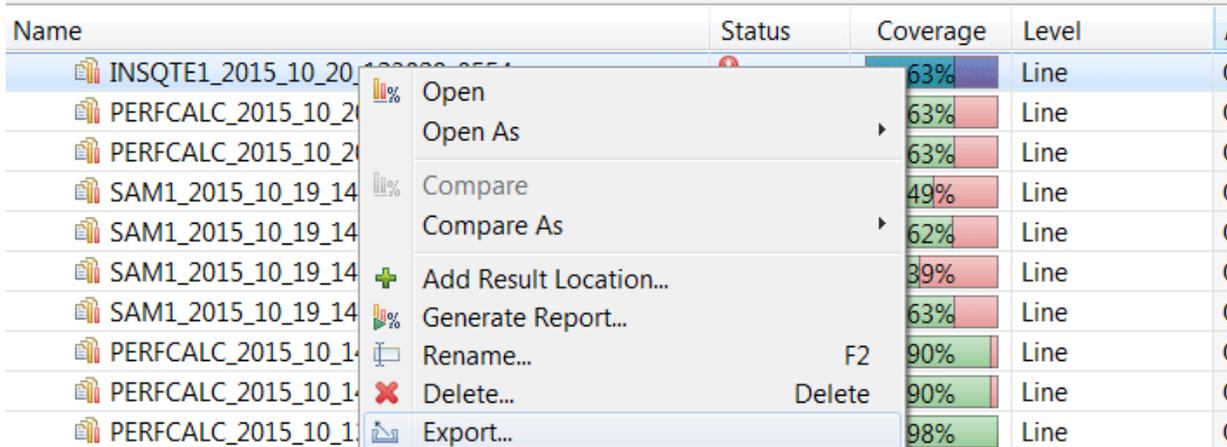


Figure 3. Sample showing the Export menu in Code coverage results view

8. In the dialog that pops up, browse to the location on the file system to where the result is exported to, and specify a name for the output file. This will create a compressed file with the code coverage results from the test run.

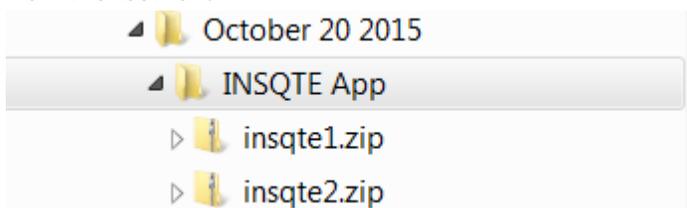


Figure 4. Sample showing file system organization of code coverage results

- Repeat steps 1 - 8 for running the other JCL scripts that are intended to test the same build of the same application and export the code coverage results to the same location. .

Generating code coverage results by using the headless collector running on ADDI Extensions server

You can learn how to generate code coverage results by using the headless collector running on ADDI Extensions server. Verify the assumptions before you perform this activity.

Assumption

Verify the assumption before you prepare code coverage results by using the headless code coverage collector.

The ADDI Extensions server is started and running. As a part of the ADDI Extensions startup, the headless code coverage collector daemon is started.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results by using the headless code coverage collector.

- Create the data provider by following the instructions in [“Adding a Manual Builds data provider to collect data automatically”](#) on page 201.
- On the **Data Providers** page, click the **Options Menu** icon on the data provider card you created. The options menu appears.
- Select **Copy Startup Key** from the options menu to copy the startup key to clipboard.
- To see the startup key example, select **Edit** from the options menu. On the **Edit Data Provider** page that appears, you can see a **Startup key example** section similar to the following snippet.

Startup key example

```
TEST(,,,TCPIP&insurance.example.com%8005:*)
ENVAR("EQA_STARTUP_KEY=CC,,,providerid=BGMT7L,testid=<fill in your test-id>")
```

- Provide the startup key example on your screen to the test team to insert it into the JCL. In this case, take the one with providerid=BGMT7L for example.

```
//CEEOPTS DD *
TEST(,,,TCPIP&<adi server tcpip address>%8005:*)
ENVAR("EQA_STARTUP_KEY=CC,,,providerid=BGMT7L,testid=<your test-id>")
/*
```

Notes:

- The providerid is unique and case-sensitive. Make sure that you copy exactly the same providerid on your screen.
- The testid that you specify for a test is also case-sensitive and needs to be unique within a provider to distinguish it from other tests. If you specify a test into two locations (for example, because the test is split into two parts), enter the testid with same spelling and casing. To specify a testid that has either blanks or commas, you must enclose it in paired single quotation marks, for example, testid='a test id'.

The IP address is the IP address of the ADDI Extensions server machine where the code coverage collector daemon is running. It can be the IP address or the DNS name of the server. Submitting this JCL results in the code coverage for the test case with test case id HDLS6 to be generated to the ADDI Extensions server for automatic collection. The providerid parameter ensures that the results are grouped to the appropriate data provider. Code coverage files are saved to the directory **adi-server location \adi\conf\headless-CC-files\providerid-file\program_datetime_id.cczip**. When you refresh the data provider, these code coverage files are deleted.

Preparing code coverage results for CICS transactions by using the debug tool backend and IDz client

By using the debug tool backend and IDz client, you can prepare code coverage results for CICS transactions.

Assumptions

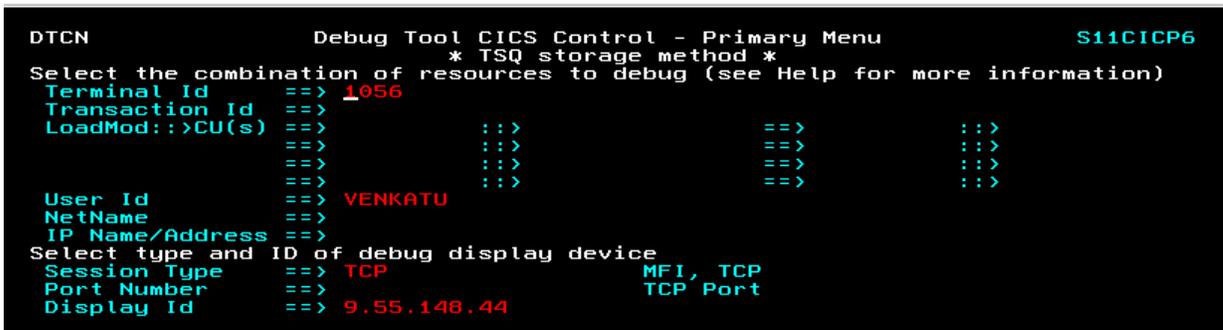
Verify the assumptions before you prepare code coverage results for CICS transactions by using the debug tool backend and IDz client.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results for CICS transactions.

Complete the following steps to generate code coverage results for CICS transactions by using Debug Tool backend and IDz client.

1. Create a DTCN profile to collect the code coverage results for a transaction.

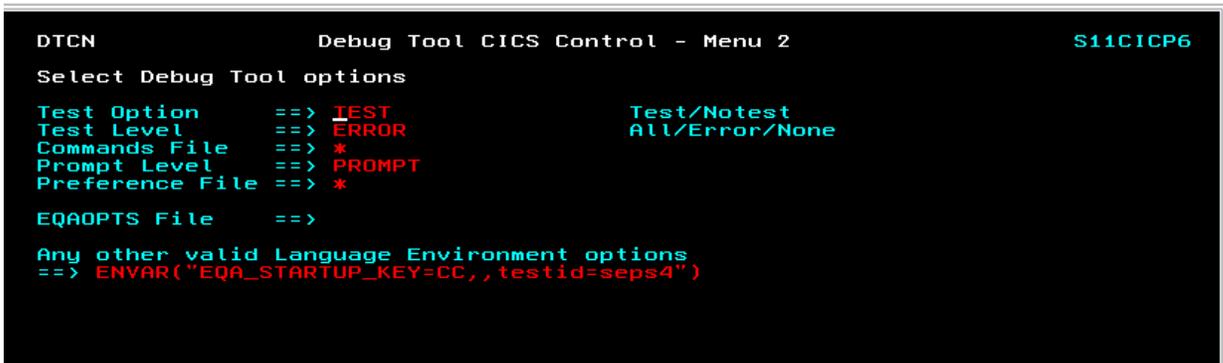


```
DTCN                               Debug Tool CICS Control - Primary Menu                               S11CICP6
                                * TSQ storage method *
Select the combination of resources to debug (see Help for more information)
Terminal Id    ==> 1056
Transaction Id ==>
LoadMod::>CU(s) ==>           :>>           ==>           :>>
                ==>           :>>           ==>           :>>
                ==>           :>>           ==>           :>>
                ==>           :>>           ==>           :>>
                ==>           :>>           ==>           :>>
User Id       ==> VENKATU
NetName      ==>
IP Name/Address ==>
Select type and ID of debug display device
Session Type ==> TCP           MFI, TCP
Port Number  ==>           TCP Port
Display Id   ==> 9.55.148.44
```

Figure 5. Screen showing the DTCN profile creation primary menu

2. Specify User Id, Session Type to be TCP, and IP Address pointing to the IDz instance where the code coverage results would be displayed. Specify the port on which the debug daemon on the client runs. The default port is 8001.

Press **PF9** and enter the following ENVAR("EQA_STARTUP_KEY=CC,,testid=seps4") as shown in Figure 2 below.



```
DTCN                               Debug Tool CICS Control - Menu 2                               S11CICP6
Select Debug Tool options
Test Option    ==> TEST           Test/Notest
Test Level     ==> ERROR          All/Error/None
Commands File  ==> *
Prompt Level   ==> PROMPT
Preference File ==> *
EQAOPTS File   ==>
Any other valid Language Environment options
==> ENVAR("EQA_STARTUP_KEY=CC,,testid=seps4")
```

Figure 6. Screen showing DTCN secondary menu to specify the Code coverage invocation environment variable

The ENVAR is the directive to capture code coverage and the testid parameter is for passing in the test case id. Press **PF3** and then **PF4** to save the profile.

On the IDz client, click Window > Open Perspective > Debug to switch to the debug perspective and check the daemon which is listening on the specified port as shown in Figure 3 below. The default port is 8001.

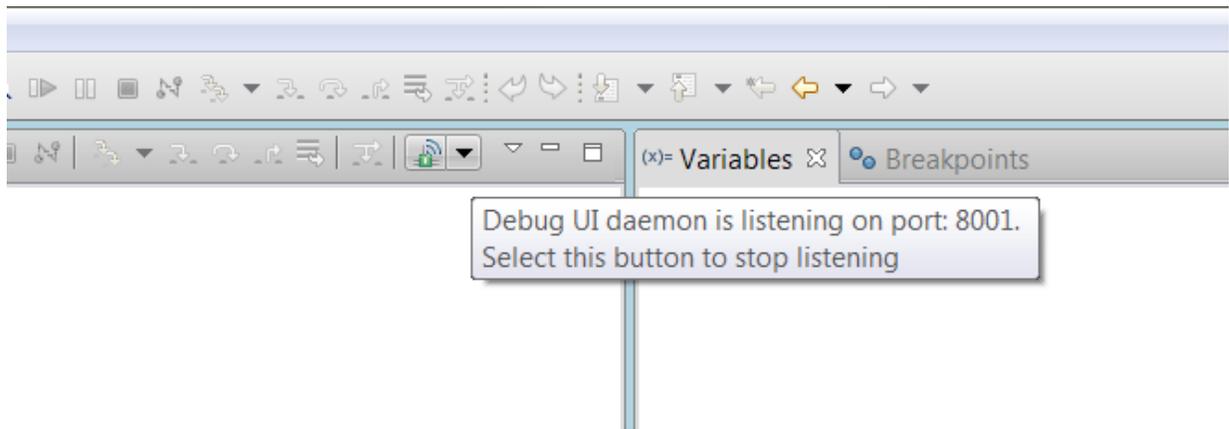


Figure 7. Screen shot showing the verification of debug daemon running

3. Run the transaction, for example, SEPS.
4. Get the code coverage results in the IDz code coverage view.

Name	Status	Coverage	Level	Analyzed Date	Addition
Compiled Code Coverage Workspace Results					
SEPS100_2015_10_23_144214_0503	❌	46%	Line	Oct 23, 2015 2:42:23 PM	
SEPS100_2015_10_23_143901_0295	❌	57%	Line	Oct 23, 2015 2:39:15 PM	
SEPS100_2015_10_23_142307_0419	❌	22%	Line	Oct 23, 2015 2:23:15 PM	
SEPS100_2015_10_23_141709_0763	❌	53%	Line	Oct 23, 2015 2:17:21 PM	
SEPS100_2015_10_23_140340_0168	❌	48%	Line	Oct 23, 2015 2:03:50 PM	

Figure 8. Screen showing the Code Coverage result in the Code Coverage results view

5. Right-click each of the results and export the results into a compressed file.

Preparing code coverage results for COBOL and PL/I by using RTC

You can generate code coverage results for COBOL and PL/I by using automated builds provided by IBM Rational Team Concert Enterprise Extensions (RTC EE). Also, you can connect ADDI Extensions to this RTC EE server to collect the code coverage data automatically.

Take the following procedures to set up RTC builds to generate the coverage data. For details about how to configure ADDI Extensions to connect to RTC and retrieve the data, see [Adding a Rational Team Concert builds provider](#).

Assumptions

Verify the assumptions before you prepare code coverage for COBOL and PL/I by using RTC.

- The development team uses RTC EE for z Systems Build component to build their COBOL and PL/I artifacts and run either automated or manual tests as part of that build process.
- A set of build definitions are created for the COBOL and PL/I applications being managed by RTC EE. For more information, see [Setting up Enterprise Extension builds](#).
- RDz provided *Host Utilities (GI13-2864)* that offers the headless Code coverage support, which is installed and configured on the z/OS machine. For more information, see Chapter 6 in *IBM Rational Developer for z Systems: Configuration Guide*.
- The test team uses the RDz provided [headless Code Coverage](#) setup to automatically generate code coverage results files when you run a build.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage for COBOL and PL/I by using RTC.

Extend the existing build definition to submit a set of JCL that triggers a manual or automated test after a successful build.

1. Use the Eclipse RTC client and select the project for which you would want to run the tests and capture code coverage results as part of the build process.
2. Make sure that you installed all the binaries required by the build engine, that is, the correct version of Ant or Maven that your build scripts would be using.
3. In other RAD online help pages, there are steps for installing additional binaries on the build engine such as the Eclipse Birt libraries. This is not required for use with ADDI Extensions, as ADDI Extensions provides its own presentation of the code coverage results and does not require the Birt-based processing of the data described there.

On your Eclipse RTC rich client.

Create an Ant post build script

1. Start your Eclipse RTC rich client and select the project or application for which you want to run the tests and gather code coverage results as part of the Build process. If no projects are loaded, follow the standard RTC flow to load the project(s).
2. Refer to the following postbuild.xml sample to create an Ant script. You can start with the provided sample and modify as appropriate.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--=====
      This Ant script calls the code coverage headless script
      using the properties defined in the build definition.
      =====-->
<project name="Code Coverage" default="all"
xmlns:jazz="antlib:com.ibm.team.enterprise.anttasks"
xmlns:ac="antlib:net.sf.antcontrib" >
  <taskdef resource="net/sf/antcontrib/antcontrib.properties"/>

  <description>
    This Ant file is used to run the Code Review and Code Coverage
    applications. It publishes the results to the RTC repository.
  </description>

  <taskdef classname="com.ibm.team.build.ant.task.ArtifactFilePublisherTask"
name="artifactFilePublisher"/>
  <taskdef name="junitLogPublisher"
classname="com.ibm.team.build.ant.task.JUnitLogPublisherTask" />
  <taskdef name="nunitLogPublisher"
classname="com.ibm.team.build.ant.task.NUnitLogPublisherTask" />

  <target name="main">
    <property name="temp.dir" value="${team.enterprise.scm.fetchDestination}/tmp"/>
    <mkdir dir="${temp.dir}"/>
    <property name="rexx.dir" value="${jcl.dir}"/>

    <!-- Convert and add execute permission -->
    <exec executable="iconv" output="${rexx.dir}/getJobRC.rexx"><arg line="-f UTF-8
-t IBM-1047 ${rexx.dir}/getJobRC.rexx"/></exec>
    <chmod perm="755" file="${rexx.dir}/getJobRC.rexx"/>

    <!-- Convert and add execute permission -->
    <exec executable="iconv" output="${jcl.dir}/renameZipFile.sh">
      <arg line="-f UTF-8 -t IBM-1047 ${jcl.dir}/renameZipFile.sh"/>
    </exec>
    <fixcrlf srcdir="${jcl.dir}" includes="renameZipFile.sh" cr="remove" />
    <chmod perm="755" file="${jcl.dir}/renameZipFile.sh" verbose="true"/>

    <ac:for list="${jcl.list}" param="jobname" >

      <sequential>

        <echo message="*** Job to be executed @${jobname}" />
        <property name="job" value="@${jobname}" />
```

```

        <echo message="job property = ${job}" />
        <antcall target="startCodeCoverageDaemon" />
        <!-- Wait until the debug daemons are available for running code
coverage sessions -->
        <waitfor maxwait="3" maxwaitunit="minute" checkevery="5000">
            <and>
                <socket server="${IP}" port="${CC.PORT_1}"/>
            </and>
        </waitfor>

        <echo message="Submit ${job} job" />
        <submitJCL src="&quot;/${resourceTeamPrefix}.JCL(${job})&quot;"/>

        <!-- Stop the code coverage debug daemon that is listening -->

        <exec executable="${RDZ_UTIL_BIN}/ccstop" dir="$
{team.enterprise.scm.fetchDestination}" spawn="true">
            <env key="JAVA_HOME" value="${JAVA_HOME}"/>
            <arg line = "${CC.PORT_1}"/>
        </exec>

        <waitfor maxwait="3" maxwaitunit="minute" checkevery="5000">
            <and>
                <available file="${team.enterprise.scm.fetchDestination}/cc/
results_${buildLabel}/cc_done" />
            </and>
        </waitfor>

        <delete file="${team.enterprise.scm.fetchDestination}/cc/results_$
{buildLabel}/cc_done" />

        <exec executable="${jcl.dir}/renameZipFile.sh" dir="${jcl.dir}">
            <arg line = "${team.enterprise.scm.fetchDestination}/cc/results_$
{buildLabel}"/>
            <arg line = "HospitalApplication-CC-${buildLabel}-${job}.cczip"/>
        </exec>

        <!-- Publish Code Coverage Results -->

        <antcall target = "publishResults"/>

        <var name="job" unset="true"/>

    </sequential>

</ac:for>

</target>

<!--=====
Start the daemon listener for code coverage. This task is run as
a daemon because the end of it will never be reached (gets stuck in the
exec task)
=====-->

<target name="startCodeCoverageDaemon">

    <exec executable="${RDZ_UTIL_BIN}/ccstop" dir="$
{team.enterprise.scm.fetchDestination}" spawn="true">
        <env key="JAVA_HOME" value="${JAVA_HOME}"/>
        <arg line = "${CC.PORT_1}"/>
    </exec>

    <waitfor maxwait="3" maxwaitunit="minute" checkevery="5000">
        <and>
            <available file="${team.enterprise.scm.fetchDestination}/cc/results_$
{buildLabel}/cc_done" />
        </and>
    </waitfor>

    <delete file="${team.enterprise.scm.fetchDestination}/cc/results_${buildLabel}/
cc_done" />

    <property environment="env"/>

    <mkdir dir = "${team.enterprise.scm.fetchDestination}/cc/results_${buildLabel}"/>

```

```

        <mkdir dir = "${team.enterprise.scm.fetchDestination}/cc_1/workspace"/>

        <!-- Invoke the headless code coverage using the ccstart command, the required
        parms are passed in via the arg line -->

        <exec executable="${RDZ_UTIL_BIN}/ccstart" dir="$
{team.enterprise.scm.fetchDestination}" spawn="true">
            <env key="JAVA_HOME" value="${JAVA_HOME}"/>
            <arg line = "-port=${CC.PORT_1}"/>
            <arg line = "-zipresult"/>
            <arg line = "-output=${team.enterprise.scm.fetchDestination}/cc/results_
{buildLabel}"/>
            <arg line = "-prevresultpath=PREV"/>
            <arg line = "-reportformat=HTML"/>
            <arg line = "-data ${team.enterprise.scm.fetchDestination}/cc_1/workspace"/>
            <arg line = "-savesource"/>
        </exec>

        <!-- Nothing else in this task can be reached! This task gets stuck in the exec
        because the program is a daemon-->

        <touch file="${team.enterprise.scm.fetchDestination}/cc/results_{buildLabel}/
cc_done" />

    </target>

    <!-- Submits the JCL -->
    <macrodef name="submitJCL">
        <attribute name="src"/>
        <attribute name="sleepInterval" default="5"/>
        <attribute name="ignoreRC" default="0"/>
        <sequential>
            <antcall target="doJCLSubmission">
                <param name="src" value="@{src}"/>
                <param name="sleepInterval" value="@{sleepInterval}"/>
                <param name="ignoreRC" value="@{ignoreRC}"/>
            </antcall>
        </sequential>
    </macrodef>

    <target name="doJCLSubmission">
        <property name="output.file" value="${temp.dir}/getJobRC.out"/>
        <property name="result.file" value="${temp.dir}/getJobRC.result"/>
        <delete file="${result.file}"/>
        <exec executable="submit" outputproperty="job.id"><arg line="-j ${src}"/></exec>
        <echo message="Submitted JCL from ${src}. Job ID: ${job.id}"/>
        <exec executable="${rexx.dir}/getJobRC.rexx" output="${output.file}"
append="false">
            <arg value="${job.id}"/>
            <arg value="${result.file}"/>
            <arg value="${sleepInterval}"/>
        </exec>
        <echo message="*** REXX output ***"/>
        <exec executable="cat"><arg value="${output.file}"/></exec>
        <echo message="*** End REXX output ***"/>
        <exec executable="cat" outputproperty="job.maxRC">
            <arg value="${result.file}"/>
        </exec>
        <fail message="Mac RC from ${src} was: ${job.maxRC}">
            <condition>
                <and>
                    <not><equals arg1="${job.maxRC}" arg2="0"/></not>
                    <not><equals arg1="${job.maxRC}" arg2="${ignoreRC}"/></not>
                    <not>
                        <and>
                            <isset property="shouldIgnoreWarnings"/>
                            <or>
                                <equals arg1="${job.maxRC}" arg2="1"/>
                                <equals arg1="${job.maxRC}" arg2="4"/>
                            </or>
                        </and>
                    </not>
                </and>
            </condition>
        </fail>
    </target>

    <!--=====
    Execute an additional artifactFilePublisher command to upload the coverage result zip
    file to that it appears in the downloadable artifacts section of a build result, i.e.
    the Downloads section of a build result editor in the RTC rich client. The extension of
    the file should be .cczip and .zip so that ADI easily can find and download the file when
    communicating with the RTC server. =====>

```

```

<target name="publishResults">
    <artifactFilePublisher repositoryAddress="{repositoryAddress}"
        userId="{userId}" passwordFile = "{passwordFile}"
        buildResultUUID="{buildResultUUID}"
        filePath="{team.enterprise.scm.fetchDestination}/cc/results_${
        buildLabel}/HospitalApplication-CC-{buildLabel}-{job}.cczip"
        label="{job} Code Coverage Results for batch COBOL source for
        SaylesDemo - Hospital Application"
        verbose="YES">
    </artifactFilePublisher>

    <delete dir="{team.enterprise.scm.fetchDestination}/cc/results_${
        buildLabel}" />

</target>
<target name="all" depends="main"/>
</project>

```

- The `{jcl.dir}` and `{jcl.list}` parameters in the previous ANT script define the location of the set of JCL to be submitted and the list of JCL to be submitted. This information is conveyed to the build definition by using the Properties tab of the Build definition. See the following example.

Build Definition

ID: Hospital Application build Project or Team Area: SaylesDemo

Name	Value	Description
base_dir	/usr/lpp/rdzutil/cr	
CC.PORT_1	17351	
cr.pds	MERRILL.HOSPITAL.COBOL	
DB2Subsys	DSNB	
debug	TEST_ADATA.EXIT(ADEXIT(ELAXMGUX))	
IP	mvs255.rtp.raleigh.ibm.com	
JAVA_HOME	/var/java1701_64/j7.1_64	
jcl.dir	\${team.enterprise.scm.fetchDestination}/HospitalDemo	location of source project
jcl.list	TRTMNT1,TRTMNT2,TRTMNTJ	List of jcl members to be executed (** DO NOT INCLUDE THE 'jcl' exte...

- Then specify the ANT script as a "post build script" as part of the RTC EE build definition. Select the associated build definition for the project from the Work Items perspective, open the build definition, navigate to the z/OS Dependency Build tab, and add the location of the created ANT script as shown below:

Build Definition ▾

ID: Hospital Application build Project or Team Area:

General | Dependency Options | Request Options

Build File and Targets

Generate a build file from these language definitions

REXX exec (Scrum Demo Project)	Add...
Text Files (Scrum Demo Project)	Remove
Plan Bind (Scrum Demo Project)	Move Up
COBOL compilation (DB2) and link-edit and Bind (Scrum Demo Project)	Move Down

Use an existing build file

Specify a custom build file and the targets to be invoked. Properties can be referenced using \${propertyName}.

Build file:*

Build targets:

Trust build outputs

Delete obsolete outputs

Publish build map links

Reuse ISPF Session

Frequency of updates to the number of buildable files pre-processed in the build report contribution summary:

BPXWDYN Options:

Pre-build script:

Post-build script:

Always run post-build script

5. Now you can run your build. When you finish, verify in IDz Eclipse client by opening the build result editor that you have coverage results available in a Coverage tab. Check the Downloads tab that for a coverage .cczip file available for downloads.
6. You are now ready to consume these build results with ADDI Extensions.
 - a. For information about setting up an ADDI Extensions data provider that connects to your RTC server and build definition, see [“Adding a Rational Team Concert Builds data provider”](#) on page 208.
 - b. For how to collect the coverage from the latest build available in ADDI Extensions, see [“Collecting the builds and code coverage data”](#) on page 209.

Preparing code coverage results for Java

You can learn how to prepare code coverage results for Java by using IBM Rational Application Developer for WebSphere Software (RAD) v9.5.0.1, v9.6, or v9.6.1 or newer.

Preparing code coverage results by using the RAD code coverage solutions

You can choose to generate code coverage results either by using IBM Rational Application Developer (RAD) client or IBM Rational Team Concert (RTC) with RAD Quality Extensions.

Preparing code coverage results for Java by using the RAD client

You can generate code coverage results for Java by using the IBM Rational Application Developer for WebSphere Software (RAD) version 9.5.0.1, 9.6, or 9.6.1.

Assumptions

Verify the assumptions before you prepare code coverage results by using RAD client.

- The developer has written JUnit tests for his Java classes.
- The tester or developer uses the RAD environment to run individual JUnit tests manually.
- The tester has the optional Code Coverage capability that is installed in their RAD environment. And the tester uses the latest version 9.5.0.1, 9.6, or 9.6.1 that provides the Enhance Code Coverage capability.
- The tester uses only the Enhanced Code Coverage Test Runner to receive JUnit test-specific coverage results and exports each result as its own results file for ADDI Extensions.
- The tester runs all tests for each of new build of his code base again.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results for Java by using RAD client.

In RAD:

1. Review the details for "Determining Code Coverage" in either the RAD 9.5 Online Help or the [RAD 9.6.1 Online Help](#). Familiarize yourself with the options available to generate code coverage results. A main difference of RAD 9.6 and 9.6.1 is the ability to include the source code files into the exported code coverage files, which will enable many additional source code change-based analytical capabilities in ADDI Extensions. For example, only with the sources included can ADDI Extensions compute metrics of line changes and changed lines covered.
2. Set up a Java project in RAD with the code base of the current build under test.
3. Follow the steps described in "Configuring the JUnit test runner for enhanced code coverage" in the RAD 9.5 Online Help and the [RAD 9.6.1 Online Help](#) respectively to enable the enhanced coverage runner for a specific test.
 - Switch on Code Coverage in the Project Properties dialog. For RAD 9.6 or newer version, make sure you select the "Save source" check box as it will enable change-based metrics in ADDI Extensions.
 - Create a new Run Configuration for each test so that you can run each test individually or an entire Test Suite. Make sure you select the "JUnit 4 with Enhanced Code Coverage" JUnit runner in each of your configurations.
 - You can decide on the granularity of the test that ADDI Extensions will recognize by defining the Run Configuration to compose of just one JUnit test class or even one test method; or even a whole set of test classes by grouping them into a JUnit test suite and defining a Run Configuration for that suite.
4. Execute the Run Configuration created in Step 3.
5. The code coverage results for the executed configuration will appear at the top of the Code Coverage Results view in RAD. See "Accessing the code coverage results view" in the RAD 9.5 online help.
6. Right-click the new result and select **Export...** from the context menu. Also, see "Import and export results" in the RAD 9.5 online help for more information.

In RAD 9.6 or newer version, you can simply specify the location folder to export the cczip file that ADDI Extensions can read.

In RAD 9.5.0.1, you need to export two files and zip them up yourself.

- a. In the dialog, select **Coverage Data** and **Project Baseline** as the artifacts to export.
 - b. Specify a destination folder where to export.
 - c. Click **Finish**.
 - d. Navigate to the destination folder you specified in Windows Explorer. Then you see exported files.
 - e. Select the two files and create a zip file. Do not select the parent folder as the zip file must not contain any folders.
7. Rename the zip file to the name of your test.
- This step is important as the name of the zip file will determine the name of the test as it appears in ADDI Extensions.
 - If you ran this test before for an earlier build of your code base you must make sure that the name is exactly the same as before so that ADDI Extensions can map the test and compare its current results against its earlier results.
 - For example, if you ran a JUnit test called `com.example.app1.ExampleTest` then you want to name the zip file `com.example.app1.ExampleTest.zip` and upload it with that name to ADDI Extensions. When you create a new build of your code base and rerun the test to find regression, you want to again export the coverage results after the test completed, export the files and zip them and then call the zip file exactly the same as before so that ADDI Extensions can compare the result of the first with the second test.

In ADDI Extensions:

1. Log on to ADDI Extensions.
2. If you did not do so already, create a new Data Provider for your Java project. See [Adding a manual build provider to collect data manually for details](#).
3. If you or another tester did not do so already, create a new Build that reflects your current code base under test. Note, that when multiple testers submit their results against the same build the assumption ADDI Extensions makes is that they all used the exact same code base. See [Adding a build to a manual build provider when collecting data manually for details](#).
4. Open the build information page and add the exported, zipped, and renamed code coverage files created in Steps 6 to 10. See [Updating an existing build for details](#).
5. Review the results for this build in the Workbooks pages to confirm that the tests were correctly imported and that the coverage percentage is the same as seen in the RAD Coverage Results view. See [Code Coverage Reports and Information for details](#).

Preparing code coverage results for Java by using RTC with RAD Quality Extensions

You can generate code coverage results for Java by using automated builds that are provided by IBM Rational Team Concert (RTC) and connect ADDI Extensions to this RTC server to collect the code coverage data automatically.

Assumptions

Verify the assumptions before you prepare code coverage results by using RTC with RAD Quality Extensions.

- The development team uses RTC's Build component to build their Java project and run JUnit tests as part of that build.
- The development team uses the RAD Code Coverage Extension for RTC to automatically generate code coverage results files when running a build.

- The automated creation of code coverage files is limited to a coarse-grained definition of ADDI Extensions test cases. In fact, the most common case is that the execution of all JUnit tests during a build results into one code coverage file, which is mapped to only one test in ADDI Extensions, that is, all JUnit tests executed map to one ADDI Extensions test. To achieve a lower level of granularity, you can break your build into smaller groups of test execution. Each execution would generate a code coverage file that covers the JUnit test executed in that unit to be mapped to one ADDI Extensions test case. And each group results into a separate ADDI Extensions test.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results by using RTC with RAD Quality Extensions.

On your RTC build engine machine:

1. Ensure that RAD Code Coverage Extensions are installed on the RTC build engine. For more information, see *Installing the Build System Toolkit* in the Jazz online help and *Installing code quality extension for continuous integration for Rational Team Concert builds* feature using *Installation Manager* in the RAD 9.5 online help.
2. Also make sure that you installed all the binaries that are required by the build engine, for example, the correct version of Ant or Maven that your build scripts would be using.
3. In other RAD online help pages (referenced below), there are steps for installing additional binaries on the build engine such as the Eclipse Birt libraries. This is not required for use with ADDI Extensions, as ADDI Extensions provides its own presentation of the code coverage results and does not require the Birt-based processing of the data described there.

On your Eclipse RTC rich client.

Alternative 1: create an ANT build script

1. Start your Eclipse RTC rich client and follow the steps described in "Configuring a build definition in Rational Team Concert" in the RAD 9.5 online help for creating a new build definition that use Ant on the build engine.
2. Follow Steps 1 to 3 and 5 of "Creating an Ant build script to generate code coverage statistics in remote build environments" in the RAD 9.5 online help to create an Ant script that compiles and tests your code.
 - It is recommended that you start with the sample Ant script provided in "Sample Ant script (build.xml) files for remote build environments and update/refine the script" for your needs.
 - Step 5 described in "Publishing results to the Rational Team Concert server" in the RAD 9.5 online help uses the `filePublisher` command to publish the coverage result and baseline files to the RTC server so that special RTC client extensions for Eclipse that are shipping with RAD and Rational Software Architect can show the results in a special Eclipse UI of the rich client. For ADDI Extensions, you need to type a second command described in the next step.
3. In the Ant script, you need to execute an additional `artifactFilePublisher` command to upload the coverage result compressed file to that it appears in the downloadable artifacts section of a build result, that is, the **Downloads** section of a build result editor in the RTC rich client. It is strongly recommended that you rename the file before uploading for two reasons:
 - a. The name of the zip file determines the name of the test in ADDI Extensions under which the coverage results will be listed.
 - b. The extension of the file should be `.cczip` and `.zip` so that ADDI Extensions can find and download the file when communicating with the RTC server. For more details, see ["Adding a Rational Team Concert Builds data provider" on page 208](#).
4. Example for renaming and uploading the `.cczip` file:

```
<move file="${resultsDir}\Coveragedata-${buildLabel}.zip"
      tofile="${resultsDir}\Coveragedata-${buildLabel}.cczip" />

<artifactFilePublisher buildResultUUID="${buildResultUUID}"
      repositoryAddress="${repositoryAddress}" userId="${userId}" />
```

```
passwordFile="${PasswordFile}" verbose="true"
filePath="${resultsDir}\Coveragedata-${buildLabel}.cczip"
label="Code coverage data file for ADI." />
```

5. Now you can run your build. When you finish, verify in the RSA or RAD Eclipse client by opening the build result editor that you have coverage results available in a Coverage tab. Check the Downloads tab for a coverage .cczip file available for downloads.
6. You are now ready to consume these build results with ADDI Extensions.
 - a. For setting up an ADDI Extensions data provider that connects to your RTC server and build definition, see [“Adding a Rational Team Concert Builds data provider”](#) on page 208
 - b. For how to collect the coverage from the latest build available in ADDI Extensions, see [“Collecting the builds and code coverage data”](#) on page 209

Alternative 2: create one or more Maven build scripts

As an alternative to Ant, you can also use Maven to run your JUnit tests and collect code coverage data. The advantage of Maven is that, it organizes JUnit tests in the same Eclipse projects as the code under test and you could create an individual code coverage result easily for each project. Therefore, you can have a more fine-grained representation of tests in ADDI Extensions, that is, one test for each Maven project.

1. To learn more about getting started with building with Maven, start from this topic on jazz.net: [A Jazz-based Maven build](#). You can learn the main idea for working with RTC from Maven to call the same Ant commands using Maven's ability to execute Ant commands.
2. When you create your RTC Build Definition for Maven, the key thing is that you provide all Ant properties to Maven as JVM arguments following a simple schema as in this example (just an example and not all the variables you might need):

```
-DbuildLabel=${buildLabel} -DrepositoryAddress=${repositoryAddress} -DbuildResultUUID=${buildResultUUID} ....
```

Tip: As the RTC Build Definition editor only provides a single line editor, it is better if you maintain the list of variable declarations in a text editor of your choice and then paste it into RTC when ready.

3. It is recommended that you add in each Maven project's POM file a profile that is only executed when running on a build engine to generate coverage data. In this way, you can still run Maven builds in your IDE without the need to generate code coverage data and update RTC build requests. For example, you can add a profile as follows.

```
<profiles>
  <profile>
    <id>build-engine</id>
    <activation>
      <property>
        <name>buildLabel</name>
      </property>
    </activation>
    <build>
      <plugins>
```

This profile checks for the existence of the buildLabel variable, which is defined by the RTC build engine. Therefore, it would only be executed when you run on a build engine and not during development when you run in your Eclipse IDE.

4. The following example now is one possible end-to-end way of defining the Maven profile. You can define this profile in each of your POM file or if you use a hierarchical POM file in your base pom file.
 - The example uses the following variable that should be declared in the JVM variable section of the Maven RTC Build Definition. For more details of these variables listed, see last bullet point.

```
-DbuildLabel=${buildLabel} -DBuild_System_HOME=${Build_System_HOME} -DrepositoryAddress=${repositoryAddress} -DUserId=${UserId} -DPasswordFile=${PasswordFile} -DbuildResultUUID=${buildResultUUID} -Dllc-engine-plugin=${llc-engine-plugin} -Declipse-jdt-core-plugin=${eclipse-jdt-core-plugin} -Declipse-equinox-common-plugin=${eclipse-equinox-common-plugin} -Dllc-common-plugin=${llc-common-plugin} -Dllc-jvmti-library=${llc-jvmti-library}
```

- You can see the comments inside the following sample example. They map to the steps of the Ant sample script shown in the RAD Online help that was referenced previously.

```

<profile>
  <id>build-engine</id>
  <activation>
    <property>
      <name>buildLabel</name>
    </property>
  </activation>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-antrun-plugin</artifactId>
        <version>1.8</version>
        <executions>
          <execution>
            <id>code-analysis</id>
            <phase>compile</phase>
            <configuration>
              <target>
                <!-- Analyze compiled src code:
Generating .probescript and .baseline -->
                <taskdef name="code-coverage-app-analyzer"

                <classpath>
                  <fileset dir="${Build_System_HOME}/

codecoverage">
                  <include name="${llc-engine-plugin}" />
                  <include name="plugins/${eclipse-jdt-

core-plugin}" />
                  <include name="plugins/${eclipse-

equinox-common-plugin}" />
                  <include name="plugins/${llc-common-

plugin}" />
                </fileset>
              </classpath>
            </taskdef>
            <code-coverage-app-analyzer
              projectDir="${project.basedir}/target"
              probescript="${project.basedir}/$
{project.groupId}.${project.artifactId}.probescript"
              baseline="${project.basedir}/$
{project.groupId}.${project.artifactId}.baseline" />
            <taskdef name="startBuildActivity"

            <classpath>
              <fileset dir="${Build_System_HOME}/

buildtoolkit">
              <include name="*.jar" />
              </fileset>
            </classpath>
          </taskdef>
          <startBuildActivity buildResultUUID="$
{buildResultUUID}"
            repositoryAddress="${repositoryAddress}"
            passwordFile="${PasswordFile}" verbose="true"
            autoComplete="true"
            label="Building and testing $
{project.groupId}.${project.artifactId}." />
          </target>
        </configuration>
        <goals>
          <goal>run</goal>
        </goals>
      </execution>
      <execution>
        <id>publish-cc-file</id>
        <phase>package</phase>
        <configuration>
          <target>
            <zip
              destfile="${project.basedir}/$
{project.groupId}.${project.artifactId}.zip"
              basedir="${project.basedir}"

```

```

        includes="${project.groupId}.$
{project.artifactId}.coveragedata"
        encoding="UTF-8" />
contributions -->
        <!-- Publish the code coverage results as RTC
        <taskdef name="filePublisher"
        classname="com.ibm.team.build.ant.task.FilePublisherTask">
        <classpath>
        <fileset dir="${Build_System_HOME}/
buildtoolkit">
        <include name="*.jar" />
        </fileset>
        </classpath>
        </taskdef>
        <filePublisher buildResultUUID="${buildResultUUID}"
        repositoryAddress="${repositoryAddress}"
        userId="${UserId}"
        passwordFile="${PasswordFile}"
        contributionTypeId="com.ibm.rational.llc.build.coverage"
        verbose="true"
        filePath="${project.basedir}/$
{project.groupId}.${project.artifactId}.zip"
        label="Coveragedata File for $
{project.groupId}.${project.artifactId}"
        failOnError="false" />
        <zip
        destfile="${project.basedir}/$
{project.groupId}.${project.artifactId}-baseline.zip"
        basedir="${project.basedir}" includes="$
{project.groupId}.${project.artifactId}.baseline"
        encoding="UTF-8" />
        <filePublisher buildResultUUID="${buildResultUUID}"
        repositoryAddress="${repositoryAddress}"
        userId="${UserId}"
        passwordFile="${PasswordFile}"
        contributionTypeId="com.ibm.rational.llc.build.baseline"
        verbose="true"
        filePath="${project.basedir}/$
{project.groupId}.${project.artifactId}-baseline.zip"
        label="Baseline file for ${project.groupId}.$
{project.artifactId}"
        failOnError="false" />
        <!-- Publish the code coverage results to the
        <move
        file="${project.basedir}/${project.groupId}.$
{project.artifactId}.zip"
        tofile="${project.basedir}/${project.groupId}.$
{project.artifactId}.cczip" />
        <taskdef name="artifactFilePublisher"
        classname="com.ibm.team.build.ant.task.ArtifactFilePublisherTask">
        <classpath>
        <fileset dir="${Build_System_HOME}/
buildtoolkit">
        <include name="*.jar" />
        </fileset>
        </classpath>
        </taskdef>
        <artifactFilePublisher buildResultUUID="$
{buildResultUUID}"
        repositoryAddress="${repositoryAddress}"
        userId="${UserId}"
        passwordFile="${PasswordFile}" verbose="true"
        filePath="${project.basedir}/$
{project.groupId}.${project.artifactId}.cczip"
        label="Code coverage data for $
{project.groupId}.${project.artifactId}" />
        </target>
    </configuration>
    <goals>
    <goal>run</goal>
    </goals>
    </execution>
    </executions>
</plugin>
<plugin>
    <groupId>org.apache.maven.plugins</groupId>
    <artifactId>maven-surefire-plugin</artifactId>

```

```

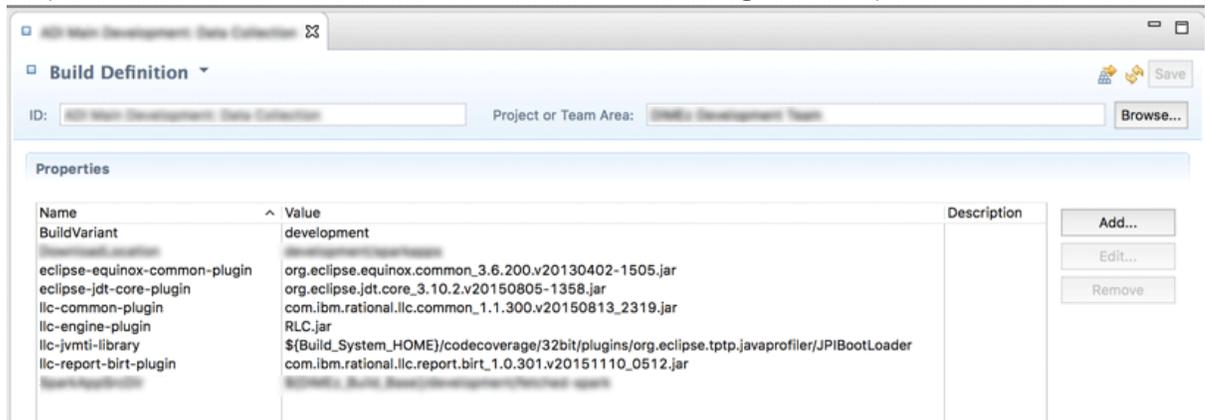
        <version>2.9</version>
        <!-- Execute tests with code coverage: .coveragedata -->
        <configuration>
            <argLine>-Dcoverage.out.file=${project.basedir}/$
{project.groupId}.${project.artifactId}.coveragedata
            -Xbootclasspath/a:${Build_System_HOME}/codecoverage/${llc-
engine-plugin}
                -agentpath:${llc-jvmti-
library}=JPIAgent:server=standalone,file=;ProbekitAgent:ext-pk-
BCILibraryName=BCIEngProbe,ext-pk-probascript=${project.basedir}/${project.groupId}.${
project.artifactId}.probascript</argLine>
        </configuration>
    </plugin>
</plugins>
</build>
</profile>

```

- You can see in the previous example that the code coverage file is uploaded as a .cczip with the same name as the Maven artifact name defined in this project. This ensures that ADDI Extensions will load these results as a test with that name. If you prefer a different name, ensure that the .cczip file is named accordingly.
- If you are running multiple POM files of this build, you need to have a parent POM that is executed by the Build Definition, which then defines each other POM as a module running each project's build and tests and upload the results for each project using the artifact name.
- The last part of this sample script that defines the argLine for running a JUnit test is slightly more complicated as in the Ant example. Because the Ant configuration of the RAD code coverage toolkit defines those in macros that are not available to Maven. Therefore, you must define these variable either as part of the Build Definition or part of the Maven POM file. The values to be used for your current version of the RAD coverage toolkit is available in this file on the build engine:

```
<rtc-build-install-dir>/buildsystem/codecoverage/CodeCoverageProperties.xml
```

You would copy each value from that file and define the variables listed in the example. For defining them as part of your RTC Build Definition, you would enter the variables and values in the Properties tab of the build definitions as shown in the following screen capture.



- Now you can run your build. When you finish, verify in RSA or RAD Eclipse client by opening the build result editor that you have coverage results available in a Coverage tab. Check the Downloads tab for a coverage .cczip file available for downloads.
- You are now ready to consume these build results with ADDI Extensions.
 - a. For setting up an ADDI Extensions data provider that connects to your RTC server and build definition, see [“Adding a Rational Team Concert Builds data provider”](#) on page 208
 - b. For how to collect the coverage from the latest build available in ADDI Extensions, see [“Collecting the builds and code coverage data”](#) on page 209

Preparing code coverage results by using the ADDI Extensions code coverage generator

You can learn how to use the code coverage generator that ships with ADDI Extensions to prepare code coverage results for Java.

The ADDI Extensions code coverage generator provides the following enhanced capabilities over the IBM Rational Application Developer (RAD) 9.5 generator.

1. It includes the source code into the code coverage results, which allows ADDI Extensions to analyze and relate code changes to code coverage results.
2. It generates test result on the JUnit test method level, which gives ADDI Extensions a much finer granularity of the results for analysis and results presentation.

Preparing code coverage results by using Apache Ant build scripts

By using an Apache Ant build script template that you can adopt for your own Java projects, you can execute the code coverage generator that is shipped with ADDI Extensions to prepare code coverage results.

Assumptions

Verify the assumptions before you prepare code coverage results for Java by using Apache Ant script.

- Apache Ant, JUnit jar, and Hamcrest jar are available as part of the prerequisites for running code coverage results in headless mode by using Ant build scripts.
- The code coverage generator works on Windows and Linux platforms. And the Java Runtime Environment is limited to IBM 64-bit Java 7 or 8 Runtime only.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results for Java by using Apache Ant script.

Generating code coverage results

Complete the following steps to generate code coverage results:

1. Go to your installed ADDI Extensions folder and navigate to `install-dir/adi/java-coverage` directory.
2. Copy all the content in the `build-junit-coverage` folder to your build or development machine.
3. Run `generatePropertyFileForADI.bat` for the Windows platform and `generatePropertyFileForADI.sh` for the Linux platform to generate `CodeCoverageProperties64.properties` file. You can find this properties file in the copy of the `build-junit-coverage` directory you created in Step 2.
4. Specify properties for `build.properties` file, such as the location of Jars to run JUnit test, directories that contain source code, compiled class files, testcases to run, and storage location of code coverage results and so on. For a complete list of all the properties, see [“Reference for build.properties and build.xml”](#) on page 186.
5. Run `build.xml` in command line. For example, run `cd` to change the directory in which `build.xml` is located and execute:

```
ant
```

The generation of code coverage results initiates.

6. Find the code coverage results in the specified result directory. The file with the `cczip` extension includes all complete code coverage results and source as shown in the following example.

```
CCtest_2016xxxx_xxxxxx.cczip
```

Unzip the zip file to find the following files:

- `/src/` - `src` directory contains all source files

- ccdata - ccdata file contains code coverage statistics data

Generating code coverage results for the Example application

ADDI Extensions includes a simple Java program with JUnit tests that you can use to quickly generate code coverage results for learning and experimentation. You can then upload those results to ADDI Extensions to review them in the application.

1. Copy the entire /adi/java-coverage/build-junit-coverage directory to a new location.
2. In the copied directory, find and unzip the sample "VehicleExample.zip".
3. Open the file build.properties in an editor and modify it to configure the correct path for homeDir and resultsDir corresponding to the current location (that is, where you copied the directory in Step 1).
4. Execute **generatePropertyFileForADI** to set other location specific variables.
5. Run **ant** to execute the build.xml which compiles the project and executes various JUnit tests.
6. Find the result files in the resultsDir location that you specified in Step 3.

Reference for build.properties and build.xml

Find the complete reference of all the properties that you can modify in build.properties.

Reference of properties in build.properties

Attribute	Description	Required
analyzeArchive	Specify whether to analyze internal archive file (jar, war, zip) within the target project directory. If this value is true, all Java classes inside archive file will be analyzed. Note: This property applies to only the build.properties file in the JavaProjectAnalyzer directory.	No. The default value is true. To avoid analyzing classes inside internal archive, set this value to False. Note: For classes inside internal archive, the source might not be available.
binDir	Specify the paths or directories where to store the compiled java .class files that you would like to analyze code coverage statistics. For multiple paths/directories: <ul style="list-style-type: none"> • Split by ";" for the Windows platform • Split by ":" for the Linux platform 	Yes
ccZipName	Specify the full name of the zip file containing the resulting code coverage data. The ".cczip" extension will be added automatically to the name. If you don't set ccZipName, it defaults to the value of resultName.	No

Table 3. Reference of properties in build.properties (continued)

Attribute	Description	Required
excludefilter	Specify the full package name and class that you would like to exclude in collecting code coverage data. You can use regular expressions to specify the classes. You can also specify the path which contains the file that listed multiple package and class name that would like to exclude. For more information, see Granular filter support for code coverage enabled from command line and an Ant script .	No. Default is empty. For multiple filters, each separated by a space. For example: excludefilter=com.ibm.vehicles.tests.TestCar com.ibm.vehicles.tests.TestCarImproved
exporterType	Specify the export format for code coverage data. You can specify multiple exporter types by using a comma to separate: <ul style="list-style-type: none"> • CCRESULT produces the compressed format with the .cczip extension. • CCSONARQUBE produces the SonarQube format with the .xml extension. • If exporter type is not specified, it will be set to CCRESULT by default. 	No. Default is CCRESULT.
haltonfailure	To stop the build process if a test fails set on (errors are considered failures as well).	No. The default value is off.
hamcrestJar	Specify the path of Hamcrest jar.	Yes
homeDir	Specify the location or directory where the java project is located.	Yes
includefilter	Specify the full package name and class that you would like to include in collecting code coverage data. You can use regular expressions to specify the classes. You can also specify the path which contains the file that listed multiple package and class name that would like to include. For more information, see Granular filter support for code coverage enabled from command line and an Ant script .	No. Default is empty. For multiple filters, each separated by a space. For example: includefilter=com.ibm.vehicles.tests.TestCar com.ibm.vehicles.tests.TestCarImproved

Table 3. Reference of properties in build.properties (continued)

Attribute	Description	Required
inputDir	Specify the paths or directories where to store the java source files that you would like to analyze code coverage statistics. For multiple paths/directories: <ul style="list-style-type: none"> • Split by ";" for the Windows platform • Split by ":" for the Linux platform 	Yes
junitJar	Specify the path of JUnit jar.	Yes
junitPath	Specify the classpath for the JUnit classes. For multiple paths/directories: <ul style="list-style-type: none"> • Split by ";" for Windows platform • Split by ":" for Linux platform 	Yes
jvmargs	Specify extra JVM arguments, split by a space.	No. This is optional. It can be used for specify extra memory. For example: jvmargs=-Xmx200m
methodLevelCapture	Specify whether to capture JUnit method level or not.	No. The default value is true. If it's not set yet or set to true, it captures the JUnit method level. If it is set to false, it doesn't capture JUnit method level (captures only JUnit class level).
projectAnalysisResultZip	Specify a zip file path to write out the project analysis result. The result zip will contain: <ul style="list-style-type: none"> • .metadata • .probescript source (if available) <p>Note: This property applies to only the build.properties file in the JavaProjectAnalyzer directory.</p>	No. This value is empty by default. If projectAnalysisResultZip is empty or is not a valid file path, no output zip file will be generated.
resultsDir	Specify the path or directory where you would like to store all the result files.	Yes
resultName	Specify the name to use for result files.	Yes

Table 3. Reference of properties in build.properties (continued)

Attribute	Description	Required
serverURL	<p>Specify the host name and port of the running application server with the javacc WAR deployed. The project analysis result zip will be uploaded to the server's current code coverage result location.</p> <p>For more information about starting the application server, see “Generating Java code coverage by using the ADDI Extensions code coverage on server generator” on page 192.</p> <p>An ServerURL example: <code>http://localhost:9080</code>.</p> <p>Note: This property applies to only the build.properties file in the JavaProjectAnalyzer directory.</p>	<p>No. This value is empty by default.</p> <p>If either serverURL or projectAnalysisResultZip is empty or not valid, no project analysis result artifacts will be uploaded to server.</p>
serverAppNames	<p>Specify a list of applications to configure WAS/Liberty server to collect code coverage exclusively for these applications.</p> <p>Note: This property applies to only the build.properties file in the JavaProjectAnalyzer directory.</p>	<p>No. This value is empty by default.</p> <p>If serverURL or projectAnalysisResultZip is empty or not valid, no project analysis result artifacts will be uploaded to server and serverAppNames will have no effects.</p>
testCase	<p>Specify the test case/class that you would like to run and get code coverage statistics results split by a space between test cases/classes for multiple test cases/classes.</p>	<p>Yes</p>

Reference for publish-result task in the build.xml

Use the **publish-result** task in the build.xml file in the JavaProjectAnalyzer directory to publish a metadata artifacts zip to a running application server with the **javacc** WAR deployed. The content of the zip will be sent to server's current code coverage result location for further analysis. For more information, see [Generating Java code coverage by using the ADDI Extensions code coverage on server generator](#).

Note: If the javacc service is not available on the specify server, no operations will be performed.

Table 4. Reference for publish-result task in the build.xml

Attribute	Description	Required
zipLocation	Specify the file path to the target zip artifact for upload. If zipLocation is empty or the zip file does not exist. No operations will be performed.	Yes.
serverURL	Specify the domain of a server with javacc service available to upload metadata artifact zip. An serverURL Example: http://localhost:9080.	Yes.
serverAppNames	Specify the names for analyzed server applications.	No.

Next steps

After you generate the code coverage results, you can find the next steps to perform more operations.

After you generate the code coverage results, create a manual data provider in ADDI Extensions with a first build. Then add the czip file of code coverage results that you generated to that build to upload those results.

For more information about how to create a manual data provider, see [Adding a manual build provider to collect data manually](#) and its peer sections.

Preparing code coverage results by using Rational Team Concert Build and Apache Ant

You can use the ADDI Extensions Apache Ant build script template as part of your Ant or Maven build scripts that you execute from a build automation system. You can prepare code coverage results by using IBM Rational Team Concert Build (RTC Build) and Apache Ant.

You can use the Apache Ant build script template that is described in the previous topic as a starting point for the Apache Ant script for your build automation. You can also call this ADDI Extensions Apache Ant build script from a Maven or other build technology that supports calling Apache Ant scripts. Another option is to reverse engineer the script and implement it with another technology that you would like to use. It requires that you execute the JUnit Runner that is provided with ADDI Extensions and used in the Apache Ant build script example to execute all tests and make sure the JUnit Runner is not from a third-party. The following example assumes that you are using IBM Rational Team Concert with Apache Ant.

Assumptions

Verify the assumptions before you prepare code coverage results for Java by using RTC Build and Ant.

- The development team uses RTC Build version 6.0 or later to build their application.
- The development team uses Ant to execute its RTC builds.
- The RTC build engine is set up and configured already. For details, see the [Rational Team Concert Online Help](#). Make sure that you have write access to the build engine server machine and the RTC Build installation directory.
- ADDI Extensions is installed as described in [Installation and setup](#). You are granted with read access to the ADDI Extensions server machine and the ADDI Extensions installation directory.
- RTC Build and ADDI Extensions could be installed on the same server machine or two different ones. Make sure that you are granted with permissions to copy some files from ADDI Extensions to the

RTC installation directories. For example, you need to have permissions to execute SSH copy (scp) operations, to transfer files via FTP, or to copy files through a USB stick.

Procedures

After you verify the assumptions, you can complete steps to prepare code coverage results for Java by using RTC Build and Ant.

Complete the following steps to prepare the code coverage results by using RTC Build and Apache Ant.

1. On the ADDI Extensions server machine, find the ADDI Extensions installation directory such as `/opt/ibm/adi/`.
2. On the RTC server machine, find the RTC Build installation directory. For example, `/opt/ibm/rtcbuild/`.
3. Copy the entire directory `/adi/java-coverage/` from the ADDI Extensions installation directory to RTC Build installation directory under the subdirectory `rtcbuild/buildsystem`.
 - The resulting directory would look like this: `/opt/ibm/rtcbuild/buildsystem/java-coverage`. Inside that directory, you find another directory that is called `plugins` and other files. For details about the files included, see the previous topic [Generating code coverage results](#).
4. On the RTC Build engine, run `cd` to change directory to `/buildsystem/java-coverage`.
5. Execute `generatePropertyFileForADI.bat` on Windows and `generatePropertyFileForADI.sh` on Linux to generate a new file called `CodeCoverageProperties64.xml`.
6. In RTC, create a Build Engine and Build Definition for your build.
7. Create your RTC Build Ant build script.
 - If you are not familiar with RTC Build, you can check Tutorial in the Team Concert Online Help for an introduction. For more information about the Ant build script file itself, see [Getting Started with Ant scripts and RTC Build Ant Tasks](#).
 - If you do not have an Ant build script, copy the files `/buildsystem/java-coverage/build.xml` and `build.properties` to your development project. These files need to be delivered to the RTC source control management system so that they can be found when you build your application.
 - If you already have an Ant build script for your RTC Build, integrate the contents of the `/buildsystem/java-coverage/build.xml` and `build.properties`.
8. Complete the `build.xml` to build your project and then use the `gen-result` target to run all JUnit tests and generate the code coverage files.
9. Check [Reference of properties in build.properties](#) for more details about properties to configure.
10. Configure the `resultsDir` property that defines the path where the resulting code coverage result file (`.cczip`) is written when JUnit tests get executed. This result file needs to be uploaded to RTC server so that they appear in the **Downloads** tab of the build results record, where ADDI Extensions downloads them from. To achieve this, augment your `build.xml` file with additional Jazz Ant task calls that will upload the code coverage files to the RTC build result record.
 - The [Rational Team Concert Online Help](#) provides a reference for the Ant tasks available.
 - The task to use for uploading the code coverage files is called `artifactFilePublisher`.
 - An example for such a call in your `build.xml` Ant script would be:

```
<artifactFilePublisher buildResultUUID="${buildResultUUID}"
  repositoryAddress="${repositoryAddress}" userId="${UserId}"
  passwordFile="${PasswordFile}" verbose="true"
  filePath="${resultsDir}/${resultFileName}"
  label="Code coverage data for my project." />
```

- A good place for this task execution in the `build.xml` is at the end of the `gen-result` target, after the line `<antcall target="delete-result" />`.
- You need to include the definition of those RTC tasks at the beginning of your `build.xml` file. For example, run a statement such as `<import file="${buildToolkitLocation}"/`

BuildToolkitTaskDefs.xml" />. The \${} you would set for your RTC Build Engine record is *Build Toolkit Install/buildsystem/buildtoolkit*.

11. Execute your RTC builds. The result should be build records that include generated cczip file with the code coverage results available in the **Download** section.

Next steps

After you generate the code coverage results, you can find the next steps to perform more operations.

In ADDI Extensions, create a connection and an automated data provider that connects to your RTC Build Engine and Build Definition to collect the coverage result cczip files automatically. For more details, see [Managing connections, providers, and applications](#).

Preparing code coverage results by using the ADDI Extensions code coverage on server generator

You can learn how to use the ADDI Extensions code coverage on server generator to generate Java code coverage and project metadata.

Generating Java code coverage by using the ADDI Extensions code coverage on server generator

You can use the ADDI Extensions code coverage on server generator to capture Java code coverage on IBM WebSphere Application Server Traditional or Liberty at an individual test level. The generated code coverage results can be downloaded from the application server as a cczip file and then uploaded to ADDI Extensions for further analysis.

Assumptions

Verify the assumptions before you generate Java code coverage by using the ADDI Extensions code coverage on server generator.

Make sure that the ADDI Extensions code coverage on server generator is supported on:

- IBM WebSphere Application Server Traditional v8.5.5.x or v9.0.0.x running Java EE 7.
- IBM WebSphere Liberty 17.0.0.x server running Java EE 7. Make sure the Liberty server has the **jaxrs-2.0** feature or the other features containing the **jaxrs-2.0** feature, such as the **webProfile-7.0** feature.
- Windows and Linux 64-bit platforms
- IBM 64-bit Java 7 or 8 Runtime Environment

Procedures

After you verify the assumptions, you can use the ADDI Extensions code coverage on server generator to generate Java code coverage.

Complete the following steps to prepare Java code coverage by using the ADDI Extensions code coverage on server generator.

1. Under the `install-dir/adi/java-coverage` directory, copy the folder named **websphere-coverage** to the machine with the IBM WebSphere server installed.
2. Change to the `/adi/java-coverage/webshpere-coverage/scripts` directory. If you are using IBM WebSphere Application Server Liberty, run **createCCOptionsLiberty.bat** on Windows platform or run **createCCOptionsLiberty.sh** on Linux platform. If you are using IBM WebSphere Application Server Traditional, run the **createCCOptionsWAS.py** jython script using **wsadmin** on a running server. The createCCOptions scripts generate the required JVM arguments for Java code coverage and/or start the application server with the required Java code coverage parameters. For more information about how to run the scripts and the parameters that are required, see the README file in the `scripts` directory.
3. Restart the server with the new JVM arguments from the previous step.

4. Publish the user application you want to collect code coverage for.
5. Publish the **javacc** WAR file in the `websphere-coverage/webapp` directory to the application server to deploy the REST services that enable users to collect code coverage, save the coverage for specific set of tests, and download the code coverage result.

Note: By default, the code coverage collector on server does not target any specific application to for collecting coverage information. Interacting with applications on server will not trigger code coverage.

To enable code coverage for a specific application, you need to complete the following steps.

- a. Analyze the application through [“Generating project metadata by using the ADDI Extensions code coverage generator”](#) on page 194. The generated output is a zip file.
- b. Call `uploadProjectMetadata` with parameter `zipFileContent` to upload the result to server if it has not yet been uploaded. After you upload the application analysis result, server is now configured to collect code coverage for the analyzed project.

Now you can deploy the application to the server to perform tests with code coverage. If the application is already running, you need to restart the application to obtain complete and accurate code coverage result. To restart the application, you can either restart manually, or alternatively, call `restartApplication` with `appID`.

6. To illustrate how to call the Java code coverage REST API, a sample Code Coverage Control Panel is available on the context root of `/javacc` when `javacc` WAR is deployed. Use the action on that Control Panel to invoke the REST APIs.

You can invoke the REST APIs directly as indicated below or use similarly named buttons in the sample Code Coverage Control Panel to invoke the REST APIs.

- a. Before running the test, use `getCoverageCollectionStatus` to confirm that the code coverage collection is running.
- b. Run tests against the user application that is deployed on the application server and call `saveAndResetCoverage` to assign a test ID to the tests.
- c. Repeat running tests and calling `saveAndResetCoverage` to assign test ID to the individual tests.
- d. Optional: The coverage results for all the tests run will be stored in a default result location with timestamp. You can optionally organize your tests into different sets by calling `setResultLocation` before running the tests. Calling it with a location will result in using that result location. Calling it without providing a location will result in using a new default result location with the current timestamp.
- e. Call `uploadProjectMetadata` to upload metadata and source to the current result location for the user application that you want to collect code coverage for. For more information about how to generate the project metadata, see [“Generating project metadata by using the ADDI Extensions code coverage generator”](#) on page 194.
- f. Call `downloadCoverageResult` with no parameters to download the code coverage result zip (cczip) for the current result location or provide the specific result location to download.
- g. Optional: Use the following APIs to control code coverage on the server:

- `pauseCollection`: Pause code coverage collection.
- `resumeCollection`: Resume paused code coverage collection.
- `resetCoverage`: Reset the current code coverage statistics.

Note: Invoking `resetCoverage` will not affect the saved code coverage result.

- `saveAndSetCoverage`: Save a copy of current code coverage result, and then reset the current code coverage statistics.

7. After you finish the tests, you can call `downloadCodeCoverage` to download a complete collection of code coverage results. Optionally, you can invoke `clearCoedCoverage` to clear all the saved code coverage results on the server.

Note: Invoking `clearCoedCoverage` will permanently delete all the existing code coverage results on the server. Make sure you have downloaded the code coverage results before invoking this API.

To learn the REST APIs for collecting Java code coverage on IBM WebSphere servers, see the documentation of [Java Code Coverage API](#).

For more information about how to invoke the APIs, see [“Java code coverage REST API” on page 194](#).

Next steps

After you generate the code coverage results, you can find the next steps to perform more operations.

The downloaded code coverage result zip contains the source and metadata for the user application and the code coverage statistics for that application that is covered by specific tests.

After you generate the code coverage results, create a Manual Builds data provider in ADDI Extensions. Then add the cczip file of code coverage results that you generated to that build to upload those results.

For more information about how to create a Manual Builds data provider, see [“Adding a Manual Builds data provider to collect data manually” on page 200](#) and its peer sections.

Java code coverage REST API

For more information about the REST APIs for collecting Java code coverage on IBM WebSphere servers, see the documentation of [Java Code Coverage API](#). You can also display the documentation for Java Code Coverage API on the application server by clicking the **Java code coverage REST API documentation** link on the Code Coverage Control Panel after you publish the javacc WAR on the application server.

Alternatively, to explore the Java Code Coverage REST API live on the server, you can add the application discovery capability to the server and then go to the API Explorer or the API documentation URL on the server.

- On the Liberty server, complete the following steps to explore the Java Code Coverage REST API:
 1. Add the **apiDiscovery-1.0** feature to the server.
 2. Go to *server context root/api/explorer/*, for example, <http://localhost:9080/api/explorer/>.
 3. Click **Java Code Coverage API** to explore the API.
 4. To see the API doc, go to *server context root/api/docs/*, for example, <http://localhost:9080/api/docs/>.
- On WebSphere Application Server, complete the following steps to explore the Java Code Coverage REST API:
 1. Open the WebSphere Application Server Administrative Console.
 2. Select **Enable API discovery service** in **Web Container Settings**.
 3. Go to *server context root/ibm/api/explorer/*, for example, <http://localhost:9080/ibm/api/explorer/>.
 4. Click **Java Code Coverage API** to explore the API.
 5. To see the API documentation, go to *server context root/ibm/api/docs/*, for example, <http://localhost:9080/ibm/api/docs/>.

Generating project metadata by using the ADDI Extensions code coverage generator

Assumption

Verify the assumption before you generate project metadata by using the ADDI Extensions code coverage generator.

Apache Ant is available as part of the prerequisites for running the Java project analyzer.

Procedures

After you verify the assumption, you can complete steps to generate project metadata by using the ADDI Extensions code coverage generator.

Complete the following steps to generate project metadata.

1. Under the `install-dir/adi/java-coverage` directory, copy the `build-junit-coverage` folder to the build or development machine.
2. Run `generatePropertyFileForADI` script under `install-dir/adi/java-coverage/build-junit-coverage` to generate `CodeCoverageProperties64.properties`.
3. Navigate to the `install-dir/adi/java-coverage/build-junit-coverage/JavaProjectAnalyzer` directory and update `build.properties` with the appropriate values specific to your project.
4. Run Ant in the command line to run the `build.xml` file to generate metadata and to upload the metadata and source to the application server with the `javacc` WAR deployed. This metadata and source will be included in the `cczip` that can be downloaded from the application server together with the code coverage results.
5. If the application server is not running when you run the `build.xml` to generate metadata, you can upload metadata at a later time when the server is running by using the **publish-result** task in the `build.xml`.

For more information about how to set up build properties to generate project metadata and upload to the application server, see [“Reference for build.properties and build.xml” on page 186](#).

Chapter 8. Managing connections

Learn about how to create, edit, and delete a connection and how to test if a connection can connect to the server.

Creating a connection

For a data provider that stores data on the server, before adding a data provider to the system, you must make sure that you created at least one connection for the data provider. To create a connection, you need to be an ADDI Extensions administrator.

Complete the following steps to create a connection.

1. Select the **Connections** tab on the header to go to the **Connections** page.

Note: You can add a connection when you create the data provider. If you want to add a connection as part of creating a data provider, go to step 3 to provide connection information in the **Create Connection** form.

2. Click **Create Connection** on the upper right corner of the **Connections** page. The **Create Connection** form appears.
3. Provide the following information in the **Create Connection** form.

- **Connection Name:** Provide a unique name of the connection.
- **Description:** Provide a short description of the connection.
- **Connection Type:** Select IBM Application Discovery (default value).
- **Connection URL:** Provide the URL address of the location you want to connect to.

Note: The connection to static analysis data and the Business Rule Discovery (BRD) data are on different ports. You need to create different connections to connect to static analysis data or BRD data separately.

Examples of URL address:

- IBM Rational Team Concert - `https://hostname:9753/jazz`
- IBM OMEGAMON for CICS - `http://hostname:15200`
- Static Analysis- `https://ad_hostname:9091`
- Business Rule Discovery - `http://ad_hostname:9443`
- **Username and Password:** Provide username and password of the person who has access to the server you want to connect to. If it is available, you can use the username and password of function users.

4. Click **Test** to test the connection.
5. If the connection is a success, click **Create** to create the connection.

Editing the connection information

You can edit the information of a connection after you create at least one connection. To edit the connection information, you need to be an ADDI Extensions administrator.

Complete the following steps to edit the connection information:

1. Select the **Connections** tab on the header to go to the **Connections** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the connection that you want to edit.
3. Select **Edit** from the overflow menu.

4. Edit the information such as name, description, connection URL, username, and password or all the information. If you update the connection URL, you are required to retest the connection before you save the changes.
5. Click **Save** to update the changes or click **Cancel** to quit editing.

Deleting a connection

You can delete a connection from the system. If you delete the connection, you cannot undo the deletion or restore the connection. You cannot delete a connection that is being used by data providers. Before you delete a connection, make sure that you deleted the data providers that use the connection. To delete a connection, you need to be an ADDI Extensions administrator.

Complete the following steps to delete a connection.

1. Select the **Connections** tab on the header to go to the **Connections** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the connection that you want to delete.
3. Select **Delete** from the overflow menu. The dialog box to confirm the connection deletion opens.
4. Click **Delete** to confirm the deletion or **Cancel** to cancel the deletion.

Testing a connection

You can test if a connection can connect to the server successfully.

Complete the following steps to test the connection:

1. Select the **Connections** tab on the header to go to the **Connections** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the connection that you want to test.
3. Select **Test** from the overflow menu. A pop-up message appears with the test result.

Chapter 9. Managing data providers

The data provider is the data source for analysis. You need to specify the data provider which ADDI Extensions collects the data from.

The following types of data sources are available:

- Manual Builds data providers
- Rational Team Concert Builds data providers
- OMEGAMON for CICS data providers
- System Management Facility data providers
- Application Discovery data providers
- Business Rule Discovery data providers

Note: Only the ADDI Extensions administrator can add and modify the data providers.

Data providers

To get started with ADDI Extensions, first you need to define the data providers to collect data from the data sources for analysis. Currently, ADDI Extensions offers several types of data providers, that is, Manual Builds data providers, OMEGAMON for CICS data providers, Rational Team Concert Builds data providers, System Management Facility data providers, Application Discovery data providers, and Business Rule Discovery data providers.

Manual Builds data providers

You can define a Manual Builds data provider that is defined by a baseline of the code that you run coverage tests against. You can run many tests against that build and upload coverage result compressed files for a build in which each compressed file represents the results of exactly one test. When you change your code or your tests, you must create a new build for that data provider by using the **Add Build** dialog. The assumption is that all test that ran against a particular build used the exact same version of all the code files tested. For more information, see [“Preparing code coverage results for batch applications” on page 167](#).

OMEGAMON for CICS data providers

An OMEGAMON for CICS data provider connects to an IBM OMEGAMON for CICS instance based on one or more connections defined to collect the operations-related data, including response time, CPU time, and usage frequency for CICS transactions running in a CICSplex®. As part of the data provider creation, you can choose one or more transaction service classes for which you want ADDI Extensions to gather and present the data. For more information on IBM OMEGAMON for CICS, see [IBM Tivoli OMEGAMON XE for CICS on z/OS product page](#).

Rational Team Concert Builds data providers

An Rational Team Concert Builds data provider connects to an RTC instance based on one or more connections defined, queries the builds, and gathers any code coverage data included in the RTC builds for the selected project area and build definition. As part of the data provider creation, you can specify a build tag or more to match one or more tags for the RTC builds to identify the set of builds that you want ADDI Extensions to query and inspect for code coverage data.

System Management Facility data providers

You can upload the System Management Facility files or the Application Performance Analyzer files to ADDI Extensions for performance analysis such as CPU time, Execute Channel Program time, Service units, and Elapsed time.

Application Discovery data providers

An Application Discovery data provider connects to an IBM Application Discovery (AD) server instance and collects the inventory, code complexity, and code quality metrics for the AD projects. As part of the data provider creation, you can select the set of project or projects for which you want to display the metrics and analysis. For more information, see [IBM Application Discovery and Delivery Intelligence Marketplace](#).

Business Rule Discovery data providers

An Business Rule Discovery data provider collects data from two sources to perform analysis of business rules: IBM Application Discovery (AD) server and Enterprise Artifacts location to be scanned. While ADDI Extensions scans the artifacts within a provider, it indexes all the content within artifacts for future analysis. By default, the scan takes place every 24 hours but it can be invoked by users at any time.

Managing Manual Builds data providers

You can add Manual Builds data providers and modify the Manual Builds data provider information on the Data Providers page. You can specify the Manual Builds data provider to collect data manually or automatically.

Adding a Manual Builds data provider

For Manual Builds data providers, you can create a Manual Builds data provider to collect data manually or automatically.

Adding a Manual Builds data provider to collect data manually

You can create a Manual Builds data provider on the Data Providers page to manually collect the data from the data sources for analysis.

Complete the following steps to add a Manual Builds data provider for manual data collection.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right corner of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the name and description of the data provider in the **Name** and **Description** fields.
4. Under the **Select a Data Provider Type** section, select **Manual Builds**.
5. Provide the following information in the **First Build**, **Code Coverage Files**, and **Data Provider Details** forms.
 - **First Build** form. Provide the information of the first build you have for code coverage analysis. If you have more than one build, the later builds can be added to the data provider when you modify the Manual Builds data provider.
 - **Build Name**: Provide a unique name for the build.
 - **Date of Build**: Specify the date and time when the build is created.
 - **Description**: Provide a short description of the build.
 - **Code Coverage Files** form. Click the **Add** icon to browse to the code coverage result files. Select one or more files of code coverage results for this build. The list of the selected code coverage files appears along with the name of users who upload the file and the description. You can add a description to each of the file. This is an optional step. You can skip this step if you do not have code coverage results for this build. For more information about how to prepare the code coverage

results, see [“Preparing code coverage results for batch applications” on page 167](#) and [“Preparing code coverage results for CICS transactions by using the debug tool backend and IDz client” on page 171](#).

6. Make sure that the **Enable headless collection support** checkbox is unchecked.
7. Select the number of days you want to keep data in the data warehouse. By default, the value is set to **All Data**. All the data is kept in the data warehouse. You can set the value to 90 days, 180 days, or 360 days. The data that is older than the selected value will be cleaned out.
8. Click **Create** to create the Manual Builds data provider.

Adding a Manual Builds data provider to collect data automatically

You can create a Manual Builds data provider on the Data Providers page to automatically collect the data from the data sources for analysis.

Note: Before you can add a Manual Builds data provider to collect data automatically, you need to make sure that the code coverage results are stored in the headless-cc-file directory.

Complete the following steps to add a Manual Builds data provider for automated data collection:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right corner of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the name and description of the data provider in the **Name** and **Description** fields.
4. Under the **Select a Data Provider Type** section, select **Manual Builds**.
5. Provide the following information in the **First Build**, **Code Coverage Files**, and **Data Provider Details** forms.
 - **First Build** form. Provide the information of the first build you have for code coverage analysis. If you have more than one build, the later builds can be added to the data provider when you modify the Manual Builds data provider.
 - **Build Name:** Provide a unique name for the build.
 - **Date of Build:** Specify the date and time when the build is created.
 - **Description:** Provide a short description of the build.
 - **Code Coverage Files** form. For automated data collection, no zip files to be added here. To learn about the procedures to set up the automated data collection, see [“Generating code coverage results by using the headless collector running on ADDI Extensions server” on page 170](#).
6. Select the **Enable headless collection support** checkbox to enable headless collection support. The **Collection Trigger** drop-down menu appears.
 - Select **Manual** to allow users to manually trigger the data collection.
 - Select **Automatic** and fill in the collection interval in hours to schedule the interval time for data collection.
7. Select the number of days you want to keep data in the data warehouse. By default, the value is set to **All Data**. All the data is kept in the data warehouse. You can set the value to 90 days, 180 days, or 360 days. The data that is older than the selected value will be cleaned out.
8. Click **Create** to create the Manual Builds data provider.

There will be no code coverage data loaded to the build after the Manual Builds data provider is created.

- If you have put the code coverage data into the headless-cc-file directory as described in [Chapter 7, “Preparing external data sources,” on page 167](#), you need to refresh the data provider in order to collect data from the headless-cc-file directory. For more information, see [“Collecting the code coverage data” on page 203](#).
- If you have not put the code coverage data collection in the headless-cc-file directory, follow the steps described [Chapter 7, “Preparing external data sources,” on page 167](#) to put the code coverage data collection in the headless-cc-file directory.

Modifying a Manual Builds data provider

After you add a Manual Builds data provider, you can modify the Manual Builds data provider on **Data Provider** page. ADDI Extensions prevents the provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

Editing the Manual Builds data provider information

You can edit name, description, or both of the data provider.

Complete the following steps to edit the data provider information:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to edit.
3. Select **Edit** from the overflow menu. The **Edit Data Provider** page appears.
4. Modify the data provider details, such as name, description, connection and so on.
5. Enable or disable headless collection support to switch between automatic data collection and manual data collection by selecting or clearing the headless collection support check box.
6. Modify number of days you want to keep data in the data warehouse.
7. Click **Save** to update the changes or click **Cancel** to quit editing.

Adding a build to a Manual Builds data provider when collecting data manually

When a new build with code coverage results is available, you can add a build to the existing Manual Builds data provider which collects data manually.

Complete the following steps to add a build to a Manual Builds data provider which collects data manually.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to add a build to.
3. Select **Add Build** from the overflow menu. The **Add Build** page appears.
4. Provide the following build information on the **Add Build** page.
 - **Build Name:** Provide a unique name for the build.
 - **Date of Build:** Specify the date and time when the build is created.
 - **Description:** Provide a short description of the build.
 - **Code Coverage Files:** Click **Add** icon to browse and select one or more zip files of code coverage results created by testing this build. The list of the selected code coverage files appear along with the **Filename** and **Description**. You can add a description to each of the file.
5. Click **Add** to associate the new build with the data provider.

Note: If two users upload files at the same time to the same build, the upload is strictly sequential. You can adjust this timeouts in `/Users/everardo/Downloads/adi/server/conf/adi/teamserver.properties` and the value is `coverageFileStorageCheckoutTimeout`.

Adding a build to a Manual Builds data provider when collecting data automatically

When a new build with code coverage results is available, you can add a build to the existing Manual Builds data provider which collects data automatically.

Complete the following steps to add a build to a data provider which collects data automatically.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.

2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to add a build to.
3. Select **Add Build** from the overflow menu. The **Add Build** page appears.
4. Provide the following build information on the **Add Build** page.
 - **Build Name:** Provide a unique name for the build.
 - **Date of Build:** Specify the date and time when the build is created.
 - **Description:** Provide a short description of the build.
 - **Code Coverage Files:** No code coverage zip files to be added here. You need to make sure that the code coverage results for a new build are stored in the headless-CC-file directory for ADDI Extensions to collect data automatically. See [“Generating code coverage results by using the headless collector running on ADDI Extensions server” on page 170](#) for more information.
5. Click **Add** to associate the new build with the data provider.

There will be no code coverage data loaded to the build after adding a build to an existing Manual Builds data provider. Next step, you need to refresh the data provider in order to collect data from the headless-cc-file directory. For more information, see [“Collecting the builds and code coverage data” on page 209](#).

Collecting the code coverage data

When the code coverage results are available in headless-cc-file folder of ADDI Extensions, you can manually refresh the data collection.

For more information how the code coverage results are prepared for automatically data collection, see [“Generating code coverage results by using the headless collector running on ADDI Extensions server” on page 170](#).

Complete the following steps to refresh the code coverage results for collect code coverage data from headless-cc-file folder.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to collect data.
3. Select **Start Data Collection** from the overflow menu to start the data collection.

Updating an existing build

You can update an existing build by modifying the build information.

Complete the following steps to update an existing build associated with the data provider:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to update the builds.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all the builds that are associated with the data provider appears.
4. On the row which displays a build that you want to modify, click the **Edit** icon.
5. Modify the build information. You can perform the following activities to modify build information.
 - Modify build name, date of build, and description.
 - Add extra code coverage files to the build.
 - Click the **Add** icon to browse to the code coverage files.
 - Select one or more code coverage results to associate the files with the build. When you add existing code coverage files, the files replace the files that you added previously.

- The list of the selected code coverage files appear along with the name of users who upload the file and description. You can add description next to each of the file.
 - Update comments for code coverage files.
 - Delete a code coverage file from the build.
 - Move the mouse pointer over the code coverage file you want to delete. The **Delete** icon appears on the right.
 - Select the **Delete** icon to delete the code coverage file from the build.
6. Click **Save** to update the changes.

Defining baseline builds for a Manual Builds data provider

From all the builds that are associated with a Manual Builds data provider, you can define one or more builds as baseline builds. The baseline builds can be used as major builds for code coverage comparison.

Complete the following steps to define builds as baseline builds.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to define baseline builds.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all the builds that are associated with the data provider appears.
4. On the row, which displays a build that you want to define baseline builds, click the **Star** icon in front of the build name.

Note: You can select up to 10 baseline builds.

5. Click **Save baselines builds**. The dialog box to confirm the save operation opens.
6. Click **Start Data Collection** for ADDI Extensions to perform the data collection of the Manual Builds data provider and the calculation of any baseline metrics.

Note: Depending on the size of your data, the data collection and the calculation might take time. ADDI Extensions will send you a notification when data is ready to use.

7. Optional: Click **Close** to wait for the calculation of any baseline metrics to be triggered for the next data collection of this Manual Builds data provider.

Deleting a build from the Manual Builds data provider

On the Data Providers page, you can remove a build from the Manual Builds data provider.

Complete the following steps to delete a build from its associated Manual Builds data provider.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to delete a build.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all the builds that are associated with the data provider appears.
4. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
5. Click **Data Providers** to go to the **Data Providers** page.
6. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to delete a build from.
7. Select **View Builds** from the options menu that appears. The Builds page that lists all builds that are associated with the Manual Builds data provider card appears.
8. On the row which displays a build that you want to delete, click the **Remove** icon. The dialog box to confirm that the build deletion opens.
9. Click **OK** to confirm the deletion or **Cancel** to cancel the deletion.

Downloading the code coverage data associated with build

You can download the code coverage data that is associated with build to a .csv file or .json file to be used by external analytics tools.

Complete the following steps to download the code coverage data to a .csv file or .JSON file.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to download the build data.
3. Select **Download Builds** from the overflow menu. The **Download** page appears.
4. Specify the following information to scope the code coverage results to be downloaded.
 - **Date Range:** Specify the start date and end date of the code coverage results to be downloaded.
 - **Included Code Coverage Data:** Specify the information to be downloaded by selecting the following options.
 - **Build Definitions:** Select this option to download code coverage data from all build definitions.
 - **Build Results:** Select this option to include build results with the code coverage data.
 - **Test Results:** Select this option to include test results with the code coverage data.
 - **File Test Results:** Select this option to include file test results with the code coverage data.
 - **Export Details:** Specify the file name and file format to be downloaded.
 - **File Name:** Fill in the name of the file in which you want data to be downloaded.
 - **File Download Type:** Specify the type of file to be downloaded. 2 types of files, .csv and .json are available. By default the data will be downloaded in .csv file.
5. Click **Download** to download or **Cancel** to cancel the download action. When you click **Download**, the download box appears.
6. Select the **Save File** option and click **OK**.
7. Select **Save** to download the file.

Note: The downloaded file is a zip file that contains multiple files of the file type that you select. You can find the following list of files in the zip file.

Information selected to download	File name included
Build Definition	BuildDefinitions.csv
Build Results	BuildResults.csv
Test Results	TestResults.csv
	TestResultsFilesTested.csv
	TestResultsHistoricalFilesTested.csv
	TestResultsMinimalFileTests.csv
File Test Results	FileTestResults.csv
	FileTestResultsFilesTested.csv
	FileTestResultsHistoricalFilesTested.csv

Managing OMEGAMON for CICS data providers

You can add and modify OMEGAMON for CICS data providers on the Data Providers page.

Adding an OMEGAMON for CICS data provider

You can create an OMEGAMON for CICS data provider on the Data Providers page to collect data from the data sources for analysis. Before you can add an OMEGAMON for CICS data provider, you must create at least one connection that connects the data provider to OMEGAMON for CICS server.

Complete the following steps to add an OMEGAMON for CICS data provider for data collection.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right corner of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the name and description of the data provider in the **Name** and **Description** fields.
4. Under the **Select a Data Provider Type** section, select **OMEGAMON for CICS**.
5. Select the **Connection** to connect to OMEGAMON for CICS server where the data is stored. If the Connection to OMEGAMON for CICS server has not been created, you can create the connection by clicking the **Add** icon next to the Connection header. For more information about how to create a connection, see [“Creating a connection”](#) on page 197.

Note: You can also edit connection information or delete connection from here by clicking the **Options Menu** icon on the connection card that you want to edit, delete or test. Then select **Edit**, **Delete** or **Test**.

6. Click **Connect** to fetch data from OMEGAMON for CICS server. Provider and Plexes data appear.
7. Select **Provider** and **Plexes** from which you want the data provider to collect the data. You can search for the Plexes by typing part of the Plexes name in the search box on the right corner of **Plexes** section.

Note: A plex can be used by only one OMEGAMON for CICS data provider.

8. Provide the following information.
 - **Data collection:** Specify whether you want data to be collected manually or automatically. If you choose to select data automatically, specify the interval time of data to be collected in hours.
 - Specify number of days you want to keep the data in the data warehouse. By default, the value is set to **All Data**. All the data is kept in the data warehouse. You can set the value to 90 days, 180 days, or 360 days. The data that is older than the selected value will be cleaned out.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings”](#) on page 151.

9. Click **Add** to create the OMEGAMON for CICS data provider.

Modifying an OMEGAMON for CICS data provider

After you add an OMEGAMON for CICS data provider, you can modify the OMEGAMON for CICS data provider information, view the data collection logs and download the OMEGAMON for CICS data. ADDI Extensions prevents the data provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

Editing the OMEGAMON for CICS data provider information

You can edit name, description, or both of the OMEGAMON for CICS data provider. Only the data provider details such as name, description, and data collection and number of days to keep data in the data warehouse can be edited. ADDI Extensions prevents the provider to be updated by multiple users. The page rejects all but one of many parallel updates by multiple users.

Complete the following steps to edit the OMEGAMON for CICS data provider information.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to edit.
3. Select **Edit** from the overflow menu. The **Edit Data Provider** page appears.
4. Modify the name, description, data collection, or all information of the data provider details.
Note: The Connection and Provider Plex for the data provider cannot be modified.
5. Click **Save** to update the changes or click **Cancel** to quit editing.

Viewing OMEGAMON for CICS data collection logs

You can view the status of OMEGAMON for CICS data collection.

Complete the following steps to view OMEGAMON for CICS data collection.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to view the data collection logs.
3. Select **View Logs** from the overflow menu. The list of the date and time data appears. The logs are group by the hours when they are collected.
4. Select the date and time you want to view the data collection logs. Logs of the data collection status appear.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings” on page 151](#).

Downloading the OMEGAMON for CICS data

You can download the OMEGAMON for CICS data to a .csv file or .json file to be used by external analytics tools.

Complete the following steps to download the OMEGAMON for CICS data to a .csv file or .json file.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to download the OMEGAMON for CICS data.
3. Select **Download** from the overflow menu. The **Download** page appears.
4. Specify the following information to scope the data to be downloaded.
 - **Date Range:** Specify the start date and end date of the code coverage results to be exported.
 - **Included Data:** Select the projects to be packaged in the download.
 - **Export Details:**
 - **File Name:** Specify the name of the file to download to.
 - **File Format:** Specify the type of file to be downloaded. CSV and JSON are 2 types of files available. By default the data will be downloaded in .csv file.
5. Click **Download** to download or **Cancel** to cancel the export option. When you click **Download**, the download box appears.
6. Select the **Save File** option and click **OK**.
7. Click **Save** to download the file.

Note: The downloaded file contains multiple files of the file type that you select.

Managing Rational Team Concert Builds data providers

You can add and modify Rational Team Concert Builds data providers on Data Providers page. By default, only the users with data provider administrator can add and edit Rational Team Concert Builds data

provider. The users with a workbook owner role can add, edit, refresh, and delete builds data. Any of the role permissions can view and export builds data.

Adding a Rational Team Concert Builds data provider

Go to the Data Providers page and create a Rational Team Concert Builds data provider to collect the data from the Rational Team Concert for analysis.

Assumptions

You have setup your Rational Team Concert build engine and build definition as documented in [Preparing code coverage results for Java using Rational Team Concert with RAD Quality Extensions](#) or [“Preparing code coverage results for COBOL and PL/I by using RTC”](#) on page 172.

Procedures

Take the following steps to add a Rational Team Concert Builds data provider.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right corner of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the name and description of the data provider in the **Name** and **Description** fields.
4. Under the **Select a Data Provider Type** section, select **Rational Team Concert Builds**.
5. Select the **Connection** to connect to RTC server where the data is stored. If the connection to RTC server has not yet created, you can click **Create Connection** on the upper right corner of the connection header to add a connection. For more information, see [“Creating a connection”](#) on page 197.
6. Select **Project Area** you want the data provider to collect the data from the drop-down list.
7. **Build Definitions** associated with selected Project Area are displayed. Select a **Build Definition** you want to perform code coverage analysis. You can search for the build definition by typing part of the build definition name in the **Search** box on the top of **Build Definitions** list.
8. Optional: Specify **Build Tags** to filter only the builds you want to analyze code coverage. In other words, only builds that have one of the tags listed assigned will be considered.

By default all files available for Download in an RTC build result that end with `.cczip` will be downloaded and analyzed by ADDI Extensions. If your build results do not use `.cczip` extensions or you want to define a custom filter for which files to download and use then you select the checkbox of **Use alternative regular file name expression to override**. When you select the box, a text field appears that allows to specify a regular expression that follows [the pattern language defined](#).

9. Select one of the following options for **Collection Trigger**.
 - Select **Manual** to allow users to manually trigger the data collection.
 - Select **Automatic** and fill in the collection interval in hours to schedule the interval time for data collection.
10. Specify number of days you want to keep the data in the data warehouse. By default, the value is set to **All Data**. All the data is kept in the data warehouse. You can set the value to 90 days, 180 days, or 360 days. The data that is older than the selected value will be cleaned out.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings”](#) on page 151.

11. Click **Create** to create the Rational Team Concert Builds data provider.
12. Collect data as described in [“Collecting the builds and code coverage data”](#) on page 209.

Note: Once the project area and build definition is selected, you cannot modify them.

Modifying a Rational Team Concert Builds data provider

After you add a Rational Team Concert Builds data provider, you can modify the Rational Team Concert Builds data provider, edit the data provider information, refresh the build data, delete a build from an existing data provider, and download code coverage data associated with build. ADDI Extensions prevents the data provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

Editing the Rational Team Concert Builds data provider information

You can edit the Rational Team Concert Builds data provider information such as name, description, or update RTC settings including build tags and alternative regular file name expression.

Complete the following steps to edit the Rational Team Concert Builds data provider information:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to edit.
3. Select **Edit** from the overflow menu. The **Edit Data Provider** page appears.
4. Modify one or all of the following RTC setting information.

Note: The project area and build definition cannot be modified once they are selected.

- Build Tags
 - Alternative regular file name expression to override
5. Modify the name, description, collection trigger, number of days you want to keep data in the data warehouse or all information of the data provider.

Note: The Connection for the data provider cannot be modified.

6. Click **Save** to update the changes or click **Cancel** to quit editing.

Collecting the builds and code coverage data

You can refresh the data provider to manually retrieve more build data.

The builds and code coverage data that is associated with the builds can be refreshed manually from the RTC server.

Complete the following steps to refresh the builds and code coverage data.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to collect data.
3. Select **Start Data Collection** from the overflow menu to start the data collection.

Viewing builds data of the Rational Team Concert Builds data provider

You can view the builds data and status of their data collection of the Rational Team Concert Builds data provider.

Complete the following steps to view builds data.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to view the builds data.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all builds that are associated with the data provider appears.
4. Select the **Expansion** icon on the left side to view the log information that indicates the status of data collection.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings” on page 151](#).

Defining baseline builds for a Rational Team Concert Builds data provider

From all the builds that are associated with a Rational Team Concert Builds data provider, you can define one or more builds as baseline builds. The baseline builds can be used as major builds for code coverage comparison.

Complete the following steps to define builds as baseline builds.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to define baseline builds.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all builds that are associated with the data provider appears.
4. On the row, which displays a build that you want to define baseline builds, click the **Star** icon in front of the build name.

Note: You can select up to 10 baseline builds.

5. Click **Save baselines builds**. The dialog box to confirm the save operation opens.
6. Click **Start Data Collection** for ADDI Extensions to perform the data collection of the Rational Team Concert Builds data provider and the calculation of any baseline metrics.

Note: Depending on the size of your data, the data collection and the calculation might take time. ADDI Extensions will send you a notification when data is ready to use.

7. Optional: Click **Close** to wait for the calculation of any baseline metrics to be triggered for the next data collection of this Rational Team Concert Builds data provider.

Deleting a build from the Rational Team Concert Builds data provider

For an existing Rational Team Concert Builds provider, you can remove a build from the Rational Team Concert Builds data provider.

Complete the following steps to delete a build from its associated data provider.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to delete a build.
3. Select **View Builds** from the overflow menu. The **Builds** page with a list of all builds that are associated with the data provider appears.
4. On the row that displays a build that you want to delete, click the **Remove** icon. The dialog box to confirm that the build deletion opens.
5. Click **OK** to confirm the deletion or **Cancel** to cancel the deletion.

Downloading the code coverage data associated with build

You can download the code coverage data that is associated with build to a .csv or .json file to be used by external analytics tools.

Complete the following steps to download the code coverage data to a .csv file or .json file.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to download builds data.
3. Select **Download Builds** from the overflow menu. The **Download** page appears.
4. Specify the following information to scope the code coverage results to be downloaded.
 - **Date Range:** Specify the start date and end date of the code coverage results to be downloaded.

- **Included Data:** Specify the information to be downloaded by selecting the following options.
 - **Build Definitions:** Select this option to download code coverage data from all build definitions.
 - **Build Results:** Select this option to include build results with the code coverage data.
 - **Test Results:** Select this option to include test results with the code coverage data.
 - **File Test Results:** Select this option to include file test results with the code coverage data.
 - **Export Details:** Specify the name and the file format of the file to download.
 - **File Name:** Specify the name of the file to download to.
 - **File Format:** Specify the type of file to be downloaded. CSV and JSON are two types of files available. By default the data will be downloaded in .csv file.
5. Click **Download** to export or **Cancel** to cancel the export option. When you click **Download**, the download box appears.
 6. Select the **Save File** option and click **OK**.
 7. Click **Save** to download the file.

Note: The downloaded file is a zip file that contains multiple files of the file type that you select. You can find the following list of files in the zip file.

<i>Table 6. The list of files in the downloaded zip file</i>	
Information selected to download	File name included
Build Definition	BuildDefinitions.csv
Build Results	BuildResults.csv
Test Results	TestResults.csv
	TestResultsFilesTested.csv
	TestResultsHistoricalFilesTested.csv
	TestResultsMinimalFileTests.csv
File Test Results	FileTestResults.csv
	FileTestResultsFilesTested.csv
	FileTestResultsHistoricalFilesTested.csv

Managing System Management Facility data providers

You can manage System Management Facility data providers on the **Data Providers** page.

Adding a System Management Facility data provider

You can create a System Management Facility data provider on the **Data Providers** page to upload System Management Facility and Application Performance Analyzer files for analysis.

Complete the following steps to add a System Management Facility data provider for performance analysis.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right corner of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the name and description of the data provider in the **Name** and **Description** fields.
4. Under the **Select a Data Provider Type** section, select **System Management Facility**.
5. Upload the System Management Facility and Application Performance Analyzer files.
 - a. **System Management Facility (SMF) files:**

- Click the **Browse** button to browse to the System Management Facility files.
- Select one or more files to upload. You can click the **Download a sample file** link to view the format of SMF file to be uploaded.

Note: For SMF, ADDI Extensions only uses SMF record type 30 for analysis.

b. Application Performance Analyzer (APA) files:

- Click the **Browse** button to browse to the Application Performance Analyzer files.
- Select one or more files to upload.

6. Click **Create** to create the System Management Facility data provider.

Modifying a System Management Facility data provider

After you add a System Management Facility data provider, you can modify the System Management Facility data provider information and view the data collection logs. ADDI Extensions prevents the data provider from updating by multiple users. The page rejects all but one of many parallel updates by multiple users.

Editing the System Management Facility data provider information

For a System Management Facility data provider, you can edit the System Management Facility data provider information.

Complete the following steps to edit the System Management Facility data provider information.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to edit.
3. Select **Edit** from the overflow menu. The **Edit Data Provider** page appears.
4. Update the data provider setting information such as the name or the description.
5. Click **Save** to update the data provider information or click **Cancel** to quit editing.

Importing the SMF and APA data

After you add a System Management Facility data provider, you can import the System Management Facility (SMF) and Application Performance Analyzer (APA) data. ADDI Extensions prevents the provider to be uploaded by multiple users. The page rejects all but one of many parallel uploaded by multiple users.

Complete the following steps to import the SMF and APA data:

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to import the SMF and APA data.
3. Select **Import Data** from the overflow menu. The **Import Data** page appears.
4. Upload the SMF and APA files.

Note: When you upload a new set of files, the new files will be appended to the existing files. For the same name files, the new files will replace the existing files.

- **System Management Facility (SMF) files:**

- Click the **Browse** button to browse to the System Management Facility files.
- Select one or more files to import. You can click the **Download a sample file** link to view the format of SMF file to be imported.

Note: For SMF, ADDI Extensions only uses SMF record type 30 for analysis.

- **Application Performance Analyzer (APA) files:**

- Click the **Browse** button to browse to the Application Performance Analyzer files.

- Select one or more files to import.
5. Click **Import** to import the files or click **Cancel** to quit the import.

Managing Application Discovery and Business Rule Discovery data providers

Creating an Application Discovery data provider

You can create an Application Discovery data provider on the **Data Providers** page to collect the data from the Application Discovery server for analysis. To create the data provider, you need to be an ADDI Extensions administrator.

Complete the following steps to create an Application Discovery data provider for data collection.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the **Name** and **Description** of data provider.
4. Under the **Select a Data Provider Type** section, select **Application Discovery**.
5. In the **Select a Connection** section that appears, select the connection to Application Discovery server where the data is stored. You can search for the connection by typing part of the connection name in the search box under the **Select a Connection** section header. If the connection to AD server has not been created, you can create the connection by clicking the **Create Connection** button next to the search box for connections. For more information, see [“Creating a connection” on page 197](#).

Note: You can also edit, delete, or test the connection. Under the **Actions** column of the connections, click the overflow menu (vertical ellipsis) icon of the connection that you want to edit, delete or test. Then, select **Edit**, **Delete** or **Test** from the overflow menu.

6. To specify whether you want data to be collected manually or automatically, select **Manual** or **Automatic** from the **Collection Trigger** list. If you choose to select data automatically, specify the interval time of data to be collected in hours.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings” on page 151](#).

7. Click **Create** to create the Application Discovery data provider.
8. Collect data as described in [“Collecting the data to get updated data from a data provider” on page 214](#).

Creating a Business Rule Discovery data provider

You can create a Business Rule Discovery data provider on the **Data Providers** page to collect data from the Application Discovery server for analysis. To create the data provider, you need to be an ADDI Extensions administrator.

Complete the following steps to create a Business Rule Discovery data provider for data collection.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Click **Create Data Provider** on the upper right of the **Data Providers** page.
3. On the **Create Data Provider** page that is displayed, provide the **Name** and **Description** of data provider.
4. Under **Select a Data Provider Type**, select **Business Rule Discovery**.
5. In the **Select a Connection** section that appears, select the connection to Application Discovery server where the data is stored. You can search for the connection by typing part of the connection name in the search box under the **Select a Connection** section header. If the connection to AD server has

not been created, you can create the connection by clicking the **Create Connection** button next to the search box for connections. For more information, see [“Creating a connection” on page 197](#).

Note: You can also edit, delete, or test the connection. Under the **Actions** column of the connections, click the overflow menu (vertical ellipsis) icon of the connection that you want to edit, delete or test. Then, select **Edit**, **Delete** or **Test** from the overflow menu.

6. Optional: Add all the artifacts that you want to perform the business rule discovery to ADDI Extensions location by either one of the following methods.
 - Manually copy all the artifacts to the directory that is specified as the **Artifacts Location** under the **Business Rule Discovery Data Provider Settings** section.
 - Upload a .zip file that contains all the artifacts by clicking the **Add File** button and select a .zip file that you want to upload.
7. To specify whether you want data to be collected manually or automatically, select **Manual** or **Automatic** from the **Collection Trigger** list. If you choose to select data automatically, specify the interval time of data to be collected in hours.
8. Click **Create** to create the Business Rule Discovery data provider.

Editing the information of a data provider

Complete the following steps to edit the information of a data provider.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to edit.
3. Select **Edit** from the overflow menu. The **Edit Data Provider** page appears.
4. Modify the data provider information such as Name, Description, and data collection settings.
5. Click **Save** to update the changes or **Cancel** to quit editing.

Collecting the data to get updated data from a data provider

You can refresh the data provider to manually retrieve additional or updated data from IBM AD server. To perform the refresh operation, you need to be an ADDI Extensions administrator.

Complete the following steps to refresh the data from Application Discovery server.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to refresh data.
3. Select **Start Data Collection** from the overflow menu to start the data collection.

Viewing the data collection logs

You can view the status of data collection.

Complete the following steps to view the data collection logs.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Select the name of the data provider that you want to view the data collection log. A list of the date and time when the data is collected appears along with the data collection status.

Note: If the data collection fails due to memory issues, the administrator can set the memory parameters to be used during data collection. For more information, see [“Server configuration settings” on page 151](#).

Deleting a data provider

When you delete a data provider, you delete all the collected data and analytics results for the workbooks that are associated with this data provider. To delete a data provider, you need to be an ADDI Extensions administrator.

Complete the following steps to delete a data provider.

1. Select the **Data Providers** tab on the header to go to the **Data Providers** page.
2. Under the **Actions** column, click the overflow menu (vertical ellipsis) icon of the data provider that you want to delete.
3. Select **Delete** from the overflow menu. The dialog box to confirm the data provider deletion opens.
4. Click **Delete** to confirm the deletion or **Cancel** to cancel the deletion.

Chapter 10. Managing workbooks

ADDI Extensions uses the concept of a workbook to define the scope of artifacts on which to perform analysis. You can group related artifacts from different data providers in a workbook. To perform data analysis, you can associate data providers with a workbook in the Workbooks page. To edit information of the workbook or delete a workbook, the users need to be an owner of that workbook. As members of the workbook, users can view only the reports and dashboards of that analysis collection.

Adding a workbook

Before you create a workbook to the system, you must make sure that you created at least one appropriate data provider for the workbook.

Complete the following steps to add a workbook.

1. Click **Create Workbook** on the upper right corner of the **Workbooks** page, which is the main page of ADDI Extensions.
2. Provide the following information on the **Create Workbook** page.
 - **Name:** The workbook name.
 - **Description:** A brief description of the workbook.
 - **Permissions:** User groups who have access to this workbook. Click the **Add** button to add user groups.
 - **Data Providers:** The list of all data providers in the system appears. Select one or more data providers you want to associate with the workbook. The **Settings** section for one or more data providers that you selected appears.
 - **Select Project(s):** Select one or more projects that you want to perform analysis.
 - **Settings:** Select the tab for each corresponding data provider and provide the following information for each of the data provider added.

Note: For a System Management Facility data provider, no additional settings is required.

– **Manual Builds Data Provider and Rational Team Concert Builds Data Provider**

- **Code Coverage Threshold:** Set the code coverage threshold for the code coverage analysis by using the slide bar. Slide the left and right bars to indicate the lower and upper threshold level.
- **Refine Custom Scope:** Select files to include in the analysis workbook. The custom scope helps you to focus only on the files that you are interested in. By default all the files within a build will be selected.

– **OMEGAMON for CICS Data Provider**

- Select one or more service classes you want to perform analysis.
- **Timeframe to view Dashboard's Metrics:** Set the timeframe you want the analysis report to display on the dashboard. There are 5 choices: **Last Month**, **Last Week**, **Last Day**, **Last Hour**, and **Most Recent**. ADDI Extensions uses the selected timeframe to analyze and display the information on all the data analysis views associated with OMEGAMON for CICS data providers. For example, if you select **Last Week**, ADDI Extensions takes all the data collected during past week to calculate the information that displays on the OMEGAMON for CICS analysis views. The default timeframe is **Last Day**.

– **Application Discovery Data Provider**

- **Source Changes Detection Interval (days):** Enter the number of days that you want ADDI Extensions to check and notify if any of the artifacts within the data warehouse have been modified.
- **Threshold Settings:** Expand to set the threshold values for the following metrics:

- **Maintainability Index:** Set the maintainability index threshold for the maintainability index analysis by using the slide bar. Slide the left and right bars to indicate the acceptable, poor, and unacceptable level. For more information of maintainability index, see [“Project data from Application Discovery reports and information”](#) on page 268.
- **Unreachable Code:** Set the unreachable code threshold for the unreachable code analysis by using the slide bar. Slide the left and right bars to indicate the acceptable, poor, and unacceptable level. For more information of Unreachable Code metric, see [“Project data from Application Discovery reports and information”](#) on page 268.
- **Cyclomatic Complexity:** Set the Cyclomatic Complexity threshold for the complexity analysis by using the slide bar. Slide the right and left bars to indicate the acceptable, poor, and unacceptable level. For more information of Cyclomatic Complexity metric, see [“Project data from Application Discovery reports and information”](#) on page 268.
- **Visualization Options:** Configure the charts that are displayed on the Static Analysis dashboard.
 - a. Under the **Projects Overview** section, select a metric you want to be displayed as "X-Axis", "Y-Axis" or "Bubble" size from metric drop-down lists. By default, the bubble chart is displayed as:
 - Unreachable code as X-Axis
 - Cyclomatic complexity as Y-Axis
 - Source Lines of Code as Bubble size
 - b. Under the **Project-level Metrics**, select a metric you want to display for each axis on the radar chart from metric drop-down lists. The radar chart can display three metrics to five metrics. By default, the radar chart is displayed as follows:
 - Unreachable Code as Axis 1
 - Cyclometric complexity as Axis 2
 - Source Lines of Code as Axis 3
 - Delivered Bugs as Axis 4
 - Maintainability Index as Axis 5
- **Select Resource Types for Shared Resource Analysis:** : Select shared resource for analysis from the list.
- **Business Rule Discovery Data Provider**
 - Select an Application Discovery project that you want to perform analysis.

Note: You cannot change the selected project after the workbook is created.
 - Set the sliders to configure the weights to be used for discovering keywords from the Application Discovery server. By default, all the weights are set in the middle. You can slide a slider to the left if you want to lower the weight of a specific condition.
 - Update the maximum number of keywords to request from Application Discovery server. This is the number of unique keywords that Application Discovery server returns to ADDI Extensions based on the weights configuration.

3. Click **Create** to add the workbook or **Cancel** to cancel adding the workbook.

Modifying a workbook

You can edit the workbook information, update user groups who have access to the workbook, and update the data provider settings of the workbook.

Complete the following steps to modify a workbook:

1. On the **Workbooks** page, click the overflow menu (vertical ellipsis) icon on the header of a workbook that you want to edit information and select **Edit** from the overflow menu.
2. Edit the workbook information. You can perform one or more of the following activities.

- Modify the name, description, or both of the workbook from the **Name** and **Description** field.
 - Edit the permissions of the users who can access the workbook. You can add or remove user groups.
 - a. Select one or more names of the workbook owners from the left box of contributors list.
 - b. Select **Owner** role from the right box.
 - c. Click > icon to add owners to the contributor list.
 - Remove the owners of the workbook.
 - a. Select one or more names of the workbook owners from the right box of contributors list.
 - b. Click < icon to remove owners from the contributor list.
 - Add the members of the workbook.
 - a. Select one or more names of the workbook members from the left box of contributors list.
 - b. Select **Member** role from the right box.
 - c. Click > icon to add members to the contributor list.
 - Update the data providers.
 - Select or deselect the data providers from the data provider list.

Note: You cannot change the Business Rule Discovery data provider that has already been selected.
 - Update the data provider settings for each of the data provider.
 - **Manual Builds data provider and Rational Team Concert Builds data provider**
 - **Code Coverage Threshold:** Slide the right and left bars to change the threshold level.
 - **Refine Custom Scope:** Select or unselect files to update files you want to include in the workbook.
 - **OMEGAMON for CICS data provider**
 - Select or deselect the service classes from the list.
 - **Timeframe to view Dashboard's Metrics:** Select the timeframe from the list.
 - **Application Discovery data provider**
 - Select or deselect the projects from the list.
 - **Source Changes Detection Interval (days):** Update number of days you want ADDI Extensions to check and notify if any of the artifacts within the data warehouse have been modified.
 - **Maintainability Index:** Slide the right and left bars to change the threshold level.
 - **Unreachable Code:** Slide the right and left bars to change the threshold level.
 - **Cyclomatic Complexity:** Slide the right and left bars to change the threshold level.
 - **Business Rule Discovery data provider**
 - Slide any of the sliders to update the keyword discovery weights.
 - Update the maximum number of keywords to requests from Application Discovery server.

Note: You cannot change the selected Application Discovery project and the Business Rule Discovery data provider after the workbook is created.
3. Click **Save** to save changes or click **Cancel** to quit editing.

Pinning a workbook

When you work with multiple workbooks, you can pin some of the workbooks that you are interested in or often work with.

Complete the following steps to pin a workbook:

1. On the **Workbooks** page, click the overflow menu (vertical ellipsis) icon on the header of a workbook that you want to pin.
2. Select **Pin** from the overflow menu. The Workbooks page reloads with the pinned workbook under the **Pinned Workbook** section.

Deleting a workbook

You can delete a workbook from the system. If you remove a workbook, you cannot undo or restore the action. When you delete a workbook, you do not delete the data providers that are associated with that workbook.

Complete the following steps to delete a workbook.

1. On the Workbook page, click the overflow menu (vertical ellipsis) icon on the header of the workbook that you want to delete.
2. Select **Delete** from the overflow menu. A confirmation dialog appears.
3. Click **Delete** to delete the workbook or **Cancel** to cancel the deletion.

Chapter 11. Analyzing workbooks

For each workbook, it shows its latest data analysis results from their associated data provider. For example, for the workbooks associated with Application Discovery data provider, the view displays the summary report of maintenance status of all projects within the workbook. You can click the name of a workbook to navigate to the detailed analysis dashboards and reports for all data providers that are associated with the workbook.

Analyzing and displaying code coverage results

Code coverage results are analyzed and displayed in different views. You can navigate through different views of code coverage analysis results for a workbook.

Before you start this task, you need to add at least one workbook with one Manual Builds data provider or Rational Team Concert Builds data provider to the system. For more information, see [“Adding a workbook”](#) on page 217.

Navigating through different views of code coverage results

Code coverage results can be analyzed and displayed in these views: **Workbooks** view, **Summary** view of a workbook, **Build Analysis** view, and **Flowpoints Analysis** view.

Note: A workbook is a logical grouping of programs or files. You can use a workbook to organize programs or files in the way that is meaningful to you. For more information about workbooks, see [Chapter 2, “Terminology,”](#) on page 3.

- **Workbooks** view

The **Workbooks** view is the home page of ADDI Extensions.

The Code Coverage on this view displays the overall coverage status for each workbook. The overall status of a workbook is represented as the latest aggregated code coverage of that workbook. The aggregated code coverage is the code coverage percentage of all the latest builds within that workbook for which code coverage results have been provided. For more information, see [“Code coverage reports and information”](#) on page 223.

- **Summary** view of a workbook

From the **Workbooks** view, select the name of a workbook to browse the summary view of that workbook.

The code coverage analysis reports are displayed in the **Code Coverage** tab of the workbook summary view. By default, the reports show the code coverage analysis of the latest build of a Manual Builds data provider or a Manual Builds data provider.

You can use the dropdown list on top of the tab to switch the analysis reports to a different Manual Builds data provider or Manual Builds data provider and a different build.

On the **Code Coverage** tab, the reports are organized into two sections: **Charts** and **Trends**. For each section, you can find the following reports.

- **Charts:**

- Code Coverage
- Coverage Changed vs Unchanged Code
- Biggest Coverage Drops

- **Trends:**

- Code Coverage Percentage Trend
- Executable Lines Covered Trend

For more information, see [“Code coverage reports and information” on page 223](#).

On top of the **Trends** section, you can see the dual slider to define the scope of builds that are displayed on the reports. Use the left slider to set the start build and the right slider to set the end build.

- **Build Analysis** view

On the **Code Coverage** tab in the summary view of a workbook, click **View Files** on the upper right corner of the tab to open the **Build Analysis** view.

By default, the **Build Analysis** view displays code coverage information of the latest build and the details of the files within the build. The **Build Analysis** view is composed of three areas, that is, **Header**, **Summary**, and **Content**.

The **Header** area at the most top of the view shows the data provider and build information and the menu to filter data that is displayed. You can use the dropdown icon (v) to navigate to a different data provider or build. You can use the filter menu to filter that data that is being displayed in this view. For more information about how to filter data that is displayed in the view, see [“Filtering data in the Build Analysis view” on page 234](#).

To view the code coverage results by source files, pinned files, test files, or file trends, you can select the menu on the upper right of the **Header** area.

The **Summary** area displays the summary of the entire build that includes the following information.

- Code Coverage: The percentage of code coverage and the percentage of code coverage change.
- Covered / Total Executable Lines: The number of executable lines that covered by the tests versus the total number of executable lines and the percentage of number of executable lines covered by the tests that changed.
- Modified Files: The number of modified files.
- New Files: The number of new files
- Historical Tests: The number of historical tests that are run.
- Minimal Tests: The number of minimal tests to run.

The **Content** area displays code coverage details of the items within that build. You can examine the following information. To sort the items within the content area, you can click any one of the column headers.

- Item Name: The name of item within a build.
- Warnings: Warnings for code coverage status of the item.
- Code Coverage: The percentage of code coverage.
- Change: The percentage of code coverage change.
- Covered: The number of executable lines that are covered.
- Total: The number of total executable lines.
- Added: The number of executable lines that are added.
- Updated: The number of executable lines that are updated.
- Change: The change percentage of executable lines that are added and updated.
- Deleted: The number of executable lines that are deleted.
- Historical: The number of historical tests that are run.
- Minimal: The number of minimal tests to run.

For each file within the build, you can expand the row to see the code coverage detail information of flowpoints within that file. You can click on the file name or flowpoint to see the list of historical Test to Run and the list of Minimal Tests to Run.

For detailed information about each code coverage reports or code coverage, see [“Code coverage reports and information” on page 223](#)

- **Flowpoints Analysis** view

From the **Build Analysis** view, expand the file to see the flowpoints within the file and select **Show All Flowpoints**.

The **Flowpoints Analysis** view is composed of two areas, that is, **Header** and **Content**.

- The **Header** area displays the name of the file and the menu to filter data that is displayed. You can use the dropdown icon (v) next to the file name to navigate to different file within the builds. You can also use the Filter options provided on the header area to filter information that is displayed in the view. For more information, see [“Analyzing code coverage of flowpoint level break-down”](#) on page 236.
- The **Content** area displays code coverage details of the flowpoints within that file. You can examine the following information.
 - Item Name: The name of item within a file.
 - Code Coverage: The percentage of code coverage.
 - Change: The percentage of code coverage change.
 - Covered: The number of executable lines that are covered.
 - Total Lines: The number of total executable lines.
 - Added: The number of executable lines that are added.
 - Updated: The number of executable lines that are updated.
 - Change: The change percentage of executable lines that are added and updated.
 - Deleted: The number of executable lines that are deleted.
 - Historical: The number of historical tests that are run.
 - Minimal: The number of minimal tests to run.

For each flowpoint within the file, you can click on the flowpoint to see the list of historical Test to Run and the list of Minimal Tests to Run. For detail of each code coverage reports or code coverage information, see [“Code coverage reports and information”](#) on page 223.

Code coverage reports and information

You can find the details of reports and information that is displayed for the code coverage analysis views.

Code coverage reports

<i>Table 7. Code coverage reports</i>		
Reports Name	Description	Displayed Location
Aggregated code coverage	Code coverage percentage of the analysis workbook is calculated by percentage of executable lines that are tested for all latest builds from all build providers within the analysis workbook.	Workbooks view

Table 7. Code coverage reports (continued)

Reports Name	Description	Displayed Location
Code coverage	<p>Two metrics are provided in the report.</p> <ul style="list-style-type: none"> Code coverage percentage of the build is calculated by percentage of executable lines that are tested for all the files or program within the most recent build. Percentage of code coverage change that is calculated by comparing with the previous build or the selected baseline displays along with up and down trending icon to indicate the direction of the changes. 	Code Coverage tab of a workbook summary view
Code Coverage Percentage Trend	Line charts that show the code coverage percentage trend of all builds.	Code Coverage tab of a workbook summary view
Executable Lines Covered Trend	Line charts that show the number of executable lines hit vs total executable lines of a build over time.	Code Coverage tab of a workbook summary view
Coverage Changed vs Unchanged Code	Stacked bar charts that display the comparison of code coverage between executable lines that are changed and executable lines unchanged in the current build. The stack for each bar presents the percentage executable lines that are hit vs not hit by the tests.	Code Coverage tab of a workbook summary view
Biggest Coverage Drops	Bar charts that display the top five files that have the biggest amount of code coverage drop that is compared to the last build. Any files that the code coverage drops is over 10% and the code coverage is crossed the threshold boundaries are considered as large amount of code coverage drop.	Code Coverage tab of a workbook summary view
Files with Lowest Coverage	Bar charts that display the top five source files that have the lowest code coverage.	Summary view of a workbook

Information that is displayed in Build Analysis Views

Table 8. Information that is displayed in Build Analysis views

Tab	Displayed Location	Information	Description
Files tab	Summary for Entire Build	Warning	<p>The code coverage status of the build. There are two statuses: red and yellow.</p> <ul style="list-style-type: none"> • Yellow indicates the poor code coverage percentage. • Red indicates the insufficient code coverage percentage. <p>The threshold of red and yellow statuses are set during the workbook creation. For more information, see Chapter 10, “Managing workbooks,” on page 217.</p>
		Code Coverage	<p>The code coverage percentage of the build is calculated by percentage of executable lines that are tested and total executable lines of all the source files within a build.</p>
		Covered / Total Executable Lines	<p>The number of executable lines that are exercised by the test and number of total executable lines within the entire build.</p>

Table 8. Information that is displayed in Build Analysis views (continued)

Tab	Displayed Location	Information	Description
		Change%	<p>The percentage of code coverage change is calculated by comparing the different between code coverage percentage of the current build and the previous build or the selected baseline build. There is up and down icon in front of the percentage of code coverage change to indicate the direction of the change.</p> <p>Note: You can select baseline build to compare from the Compare With drop-down menu on th top.</p>
		Modified Files	The number of the files that are modified within the build comparing to the previous build.
		New Files	The number of the files that are added to the build comparing to the previous build.
		Historical Tests	The number of all the test files that are run against the builds that are analyzed from all the previous builds that are stored in the data warehouse as well as the current build.
		Minimal Tests	The number of minimal test files testing the current build that achieve the highest code coverage percentage.
	Table	Item Name	The name of the item within the build. The item can be source file, program, package, flowpoints, and method.

Table 8. Information that is displayed in Build Analysis views (continued)

Tab	Displayed Location	Information	Description
		Warnings	<p>The code coverage status of the build. There are two statuses: red and yellow.</p> <ul style="list-style-type: none"> • Yellow indicates the poor code coverage percentage. • Red indicates the insufficient code coverage percentage. <p>The threshold of red and yellow statuses are set during the workbook creation. For more information, see Chapter 10, “Managing workbooks,” on page 217.</p>
		Code Coverage	<p>The code coverage percentage of the item displayed. For more information, see Chapter 2, “Terminology,” on page 3.</p>
		Change %	<p>The difference in code coverage of the item comparing to the previous build or the selected baseline build as a percentage.</p>
		Covered (Executable Lines)	<p>The number of the executable lines for an item displayed that are exercised by the tests.</p>
		Total (Executable Lines)	<p>The total number of executable lines for an item displayed. Executable line is defined as the line of code that the compiler marks as executable. For COBOL, the executable line might not directly correspond to the exact source line, as COBOL is not debugged by using source but rather the expanded source.</p>

Table 8. Information that is displayed in Build Analysis views (continued)

Tab	Displayed Location	Information	Description
		Added (Executable Lines)	The number of executable lines that are added to the item comparing to the previous build.
		Updated (Executable Lines)	The number of executable lines that are modified within the item comparing to the previous build.
		Change %	The percentage of executable lines that are added and modified within the item comparing to the previous build.
		Deleted (Executable Lines)	The number of executable lines that are deleted from the item comparing to the previous build.
		Historical (Tests)	The number of tests that test the item analyzed from all the previous builds as well as the current build.
		Minimal (Tests)	The number of minimal list of tests that test the item to achieve the highest code coverage percentage.
	Detailed section of the item (Click the item name to view)	Historical Tests To Run	The list of tests that test the source file or program analyzed from all the previous builds as well as the current build.
		Minimal Tests to Run	Recommendation of the list of minimal set of tests to run to achieve the highest code coverage percentage. The list is sorted by the test files with the higher test coverage.

Table 8. Information that is displayed in Build Analysis views (continued)

Tab	Displayed Location	Information	Description
Pinned Files tab			This tab displays the code coverage reports and information the same as in the Files tab but it narrows down the scope to only the files that are pinned.
Tests tab	Table	Item Name	Name of the test. For the definition of test, see Terminology Overview .
		Code Coverage	The code coverage percentage of the test that is calculated by the number of executable lines the test exercises and the total number of the executable lines of the source files the test exercises.
		Change %	The difference in code coverage of test comparing to the previous build or the selected baseline build as percentage.
		Covered (Executable Lines)	The number of executable lines the test exercises per total number of the executable lines of the source files the test exercises.
		Total (Executable Lines)	Total number of executable lines in the source files the test exercises.
		Flowpoints Covered	The number of flowpoints that the test exercises.
		Total Flowpoints	The total number of flowpoints in the sources files that the test exercises.
		File Tested	The number of source files that are exercised by the test for the current build being analyzed.

Table 8. Information that is displayed in Build Analysis views (continued)

Tab	Displayed Location	Information	Description
		Files Missing	The number of files that the test did not exercise in the test of the current build being analyzed but they were exercised in the previous tests of the previous builds.
		Elapsed Time	The time in seconds taken to exercise the test.
	Detailed section of the item (Click the item name to view)	Files Tested by Current® Build	The list of source files exercised by the test for the current build being analyzed.
		Historical Files Missing	The list of files that the test did not exercise in the test of the current build being analyzed but they were exercised in the previous tests of the previous builds.
File Trends tab		Name	The name of the source file or program within the builds.
		Build Name	The name of the build or build ID for automated build.
		Language	Programming language of the source file or program being analyzed.
		<other columns>	Same as the columns listed above for View by Source Files .

Information Display in Flowpoints Analysis View

Table 9. Information that is display in Flowpoints Analysis views

View	Information	Description
Summary	Code Coverage	The code coverage percentage of a source file. See Terminology Overview for Code Coverage.
	Coverage Change	It shows the difference in coverage of a source file in the current build compared to the same source file in the previous build as a percentage.

Table 9. Information that is display in Flowpoints Analysis views (continued)

View	Information	Description
	Executable Lines Changed	The number of executable lines that are updated in a source file from the previous build.
	Executable Lines Added	The number of executable lines that are added to a source file from the previous build.
	Executable Lines Deleted	The number of executable lines that are deleted from a source file from the previous build.
Table	Item Name	The name of a function or method within a source file.
	Code Coverage	The code coverage percentage of a function or method.
	Change %	It shows the difference in coverage of a function or method in the current build that is compared to the same function or method in the previous build or the selected baseline build as a percentage. For Java, on the package level you see the rollup coverage changes percentage for all methods under that package. Note: You can select baseline build to compare from the Compare With drop-down menu on th top.
	Covered Lines	The number of the executable lines for a function or method displayed that are exercised by the tests.
	Total Exec. Lines	Total number of executable lines for a function or method.
	Added	The number of executable lines that are added to the function or method comparing to the previous build.
	Updated	The number of executable lines that are updated in the function or method comparing to the previous build.
	Change %	The percentage of executable lines that is added or updated in the function or method comparing to the previous build.

Table 9. Information that is display in Flowpoints Analysis views (continued)

View	Information	Description
	Deleted	The number of executable lines that are deleted from the function or method comparing to the previous build.
	Historical	The number of test cases that test a function or method analyzed from all the previous builds as well as the current build.
	Minimal	The number of minimal list of tests that achieve the highest code coverage percentage.
Detailed section of the item (Click the item name to view)	Historical Tests to Run	The list of test cases that test a function or method analyzed from all the previous builds as well as the current build.
	Minimal Tests to Run	For each function or method, recommend the list of minimal set of test cases to be run to achieve the highest code coverage percentage. The list is sorted by the test files with the higher test coverage.

Analyzing code coverage trends

You can compare multiple builds to analyze code coverage trends. To analyze code coverage trends of multiple builds, you need to select two or more builds with code coverage data.

Complete the following steps to analyze code coverage trends.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of a workbook of the builds that you want to compare.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select a Manual Builds data provider or Manual Builds data provider that is associate with the builds that you want to analyze. The code coverage reports of selected data provider appear.
5. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information of the latest build and all the files within the build are displayed in the **Build Analysis** view.
6. Select the **File Trends** tab from the top of this view. The **File Trends** page appears.
7. Select the check boxes of two or more builds from the list to analyze. The maximum number of builds that are allowed to be selected is 5.
8. Click **Compare** to show the code coverage trends of the builds. The page that displays the information of all selected builds appears.
9. Optional: For Java code coverage, you can click the **Expand** icon to expand the packages and view the detailed package comparison.

For more information that is displayed in the **File Trends** view, see [“Code coverage reports and information”](#) on page 223.

Comparing code coverage results with baseline builds

ADDI Extensions uses code coverage information of baseline builds to analyze code coverage information, such as the percentage of code coverage changes or percentage of code coverage drop. By default, these information is calculated against the previous build. You can select any baseline builds to be used for comparison instead of the previous build.

Note: You can define one or more baseline builds within a Manual Builds data provider or Rational Team Concert Builds data provider. For more information, see [“Defining baseline builds for a Manual Builds data provider”](#) on page 204 and [“Defining baseline builds for a Rational Team Concert Builds data provider”](#) on page 210.

Complete the following steps to select baseline builds on the workbook summary view for comparison.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to perform the analysis on the builds.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. On the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that are associated with the build that you want to analyze. The latest build information is displayed in the **Build Analysis** view.

Note: By default, the code coverage information on the workbook summary view, such as the percentage of code coverage changes and percentage of code coverage drop, is calculated from the previous build.

5. To compare code coverage results with baseline builds, select **Compare With** drop-down menu from the menu options on the top.
6. Select a build that you want to use as a baseline for analysis. The reports on the **Code Coverage** tab is recalculated against the selected baseline build.

Analyzing pinned files

You can learn how to compare pinned files within a build to analyze code coverage for set of files.

Complete the following steps to pin files and perform analysis on the pinned files:

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook of the builds that you want to analyze.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.
5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. Click the **Pin** icon in front of the item name to pin a set of items that you want to perform the code coverage analysis on.
8. Select the **Pinned Files** tab on the top to view code coverage analysis results of pinned files.
9. On the header area, you can find the code coverage analysis summary for the pinned files next to the code coverage analysis summary for the entire build. You can see the following information.

- Code Coverage
- Modified Files
- New Files
- Historical Tests
- Minimal Tests

For more information, see [“Code coverage reports and information”](#) on page 223.

10. Click the **Minimal Tests** value for the pinned files on the header area. A list of minimal tests to run for the pinned files appears in the table on the content area. You can see the following information for each test file:

- Item Name
- Code Coverage
- Covered Lines
- Total Exec. Lines
- Flowpoints Covered
- Total Flowpoints
- Files Tested
- Files Missing
- Elapsed Time

For more information, see [“Code coverage reports and information”](#) on page 223.

11. Select the **Files** tab to go back to the list of all files within a build.
12. Optional: Click the **Pin** icon in front of the pinned items to unpin them.

Filtering data in the Build Analysis view

In this topic, you can learn how to filter information in the **Build Analysis** view.

Complete the following steps to filter the information:

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the workbook name that you want to view analysis information.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.
5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. On the top menu, click the **Filters** drop down menu.
8. In the drop down menu, select one or more of the following options to filter the information based on your needs.
 - **Show Only Files:** Select this option to show the flat list of all the files within the build.
 - **Changed Files Only:** Select this option to show only the files in the build that are modified.
 - **Hide 0% Coverage Items:** Select this option to hide items in the build that have zero percent code coverage.

- **Line Changes Percentage:** Select this option and set the slider bars below this option to show the items in the build that the percentage of added and updated executable lines is between the slider bars values.
- **Code Coverage Percentage:** Select this option and set the slider bar below this option. You can see the items in the build that have the code coverage between the slider bars values you set.

To clear a filter, you can uncheck the filter option or close the filtering message box that is displayed on top of the table

Searching for files, packages, or directories within a build

You can learn how to search for files, packages or directories within a build from the **Build Analysis** view to view code coverage results.

Complete the following steps to search for files, packages, or directories within a build.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.
5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. Enter the name or part of the name of the files, packages, or directories you want to search for in the Search box on the top of the table header. You can see the items that match the name or part of the name you are searching for in the table with the code coverage analysis status.

Analyzing the Tests to Run

You can learn how to use ADDI Extensions for suggestions about which tests to run based on the analysis of code coverage results. With the list of **Historical Tests to Run** or **Minimal Tests to Run**, ADDI Extensions helps you plan the testing of your build.

Before you begin this activity, you need to add at least one workbook and one data provider with code coverage results.

For more details about the information that is displayed in the code coverage analysis views, see [Code Coverage Reports and Information](#).

Analyzing the tests to run based-on the Historical Tests to Run

Complete the following steps to analyze tests to run by using the list of **Historical Tests to Run**.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.

5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. For each of the file, check the number of historical tests to run on the **Historical** column.
8. Select the item name to view the list of Historical Tests to Run.

Note: To find the item that you want to view the list of Historical Tests to run, you can use the search box on top of the table header to search for the item or you can click on the column header to sort the items based on the value in that column.
9. View the **Historical Tests to Run** column on the left. This column suggests the list of tests that are exercising a file from all the builds stored in data warehouse. You can use this list to construct the testing of your current build.
10. Optional: Select **Tests** view menu from the top menu to get the list of test files.

Analyzing the tests to run based on the Minimal Tests to Run

Complete the following steps to analyze the tests to run by using the list of Minimal Test to Run.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.
5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. For each of the item, check the number of minimal tests to run on the **Minimal** column.
8. Select the name of the item to view the list of Minimal Tests to Run.

Note: To find the item that you want to view the list of Minimal Tests to run, you can use the search box on top of the table header to search for the item or you can click on the column header to sort the items based on the value in that column.
9. View the **Minimal Tests to Run** column on the right. This column suggests the list of minimal set of tests that yields the maximum code coverage. You can use this list to construct the testing when perform regression testing to reduce amount of test effort.
10. Optional: Select **Tests** view menu from the top menu to get the list of test files.

Analyzing code coverage of flowpoint level break-down

The flowpoint level break-down refers to the function or method within a file. ADDI Extensions provides the detailed code coverage analysis of the flowpoint level break-down for a file. In this topic you can learn how to analyze code coverage data of flowpoint level break-down.

Complete the following steps to view the code coverage analysis of flowpoint level break-down. Before you perform this activity, you need to add at least one workbook and one data provider with code coverage results.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.

2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Code Coverage** tab if you are not yet on the tab.
4. From the **Provider** drop-down menu on top of the **Code Coverage** tab, select the Manual Builds data provider or Rational Team Concert Builds data provider that is associated with the build you want to analyze. The latest build information is displayed in the **Build Analysis** view.
5. Optional: Select the build that you want to analyze from the **Build** drop-down menu on top of the **Code Coverage** tab.
6. Select **View Files** on the upper right corner of the **Code Coverage** tab. The code coverage information for all files within the selected build is displayed in the **Build Analysis** view.
7. Expand the file or program name that you want to view the flowpoint analysis information and then some flowpoints are displayed.
8. Select **Show All Flowpoints** on the bottom to view all flowpoints. The **Flowpoints Analysis** view appears. The list of functions or methods within the file and the code coverage information is displayed in the **File Analysis** view with detail information. For more details about the information displayed in the **Flowpoints Analysis** view, see [“Code coverage reports and information”](#) on page 223.
9. Click **Filters** drop-down menu from the menu options on top.
10. Select one or more of the following options from the **Filters** drop-down menu to filter the information based on your needs.
 - **Changed Flowpoints Only:** Select this option to show the flowpoints in the files that are modified.
 - **Line Changes Percentage:** Select this option and then set the slider bars. You can see the items in the files that have the percentage of executable lines that are added and updated between the values you set in the slider bars.
 - **Code Coverage Percentage:** Select this option and set the slider bars. You can see the items in the files that the code coverage is between the values you set in the slider bars.
11. Clear the filter options or close the filtering message box that is displayed on top of the table.

Analyzing and displaying operational data from IBM OMEGAMON for CICS

Operational data from IBM OMEGAMON for CICS is analyzed and displayed in different views. You can navigate through different views of operational data analysis results for a workbook.

Before you start this task, you need to add at least one connection with one OMEGAMON for CICS data provider to the system. For more information, see [“Creating a connection”](#) on page 197.

Navigating through different views of OMEGAMON for CICS data analysis

Operational data from IBM OMEGAMON for CICS for a workbook can be analyzed and displayed in these views: **Workbooks** view, **Summary** view for a workbook, and **Transaction Analysis** view.

- **Workbooks** view

The **Workbooks** view is the home page of ADDI Extensions. The page is displayed after you log in ADDI Extensions. You can go to the Workbooks view by selecting Workbooks menu from the menu panel.

The **Workbooks** view is the view that shows all the statuses of all the workbooks in the system. Each workbook displays its latest data analysis results from their associated data provider. For the workbooks associated with OMEGAMON for CICS data provider, an average response time report displays on the **Workbooks** view. For more information, see [“OMEGAMON for CICS related reports and information”](#) on page 238

- **Summary** view of a workbook

From the **Workbooks** view, select the name of any workbook that is associated with OMEGAMON for CICS data provider to browse the **Summary** view for a workbook.

The OMEGAMON for CICS related reports are displayed on the **OMEGAMON** tab under the **Performance** tab of the summary view. The reports on the **OMEGAMON** tab displays all the service classes within that workbook. Each service class card displays the operational data reports related to the analysis of the latest operational data that is collected from the associated OMEGAMON for CICS data provider. The following five reports are displayed.

- Average transactions
- Average response time
- Average CPU time
- Transaction with Performance and Reliability Issues

Note: The average value of the reports are calculated by using the timeframe you select during the workbook creation. For example, if you select, Last Month timeframe, the average response time report is calculated by using the response time of all transactions within the past month from the present time.

For more information, see [“OMEGAMON for CICS related reports and information”](#) on page 238.

- **Transaction Analysis** view

From the **Summary** view of a workbook, you have two different ways to get to **Transaction Analysis** view report:

- Select the name of a service classe to view all transactions under the service class. Then select the name of the transaction you want to view the analysis results.
- Click on the transaction name under **Transaction with Performance and Reliability Issues** report.

The **Transaction Analysis** view is composed of **Header** and **Content** areas.

The **Header** area displays the name of service class the transaction belongs to, transaction name, maximum number of execution counts, maximum CPU time, maximum response time, and number of transactions exceeding goal response time. You can use the dropdown (v) icon, which locates next to the name of the transaction, to view transaction analysis for different transactions. You can also type part of the transaction name on the **Select Transaction** text box to search for a transaction.

The **Content** area display the history trends of average response time, average DB2 and File I/O wait time of response time, average execution count, and average CPU time of that transaction. The charts can be view hourly, daily, or weekly. The default display timeline can be set when you create an OMEGAMON for CICS workbook. For more information, see [“Adding a workbook”](#) on page 217.

- For more information, see [“OMEGAMON for CICS related reports and information”](#) on page 238.

OMEGAMON for CICS related reports and information

You can find the details of reports and information that is displayed for OMEGAMON for CICS data analysis.

For OMEGAMON for CICS data providers, two types of timeframe are available, including data collection interval timeframe and dashboard display timeframe.

Data collection interval timeframe refers to the timeframe that the collection of data from OMEGAMON for CICS server runs every 5 minutes.

Dashboard display timeframe refers to the default timeframe that the data displays on the dashboard. You can set the dashboard display timeframe during the workbook creation.

Table 10. OMEGAMON for CICS data analysis reports

Reports Name	Description	Displayed Location
Average of Average response time and its indicator status	<p>The average of average response time is calculated by averaging the average response time of all CICS transactions within all service classes in the scope of a workbook. The average response time data is collected within the dashboard display timeframe. The indicators show the level where the average response time exceeding the goal response time that is set in the OMEGAMON for CICS server.</p> <p>Red indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 400%</p> <p>Orange indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 200 - 400%</p> <p>Yellow indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 100 - 200%</p> <p>Blue indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time less than 100%</p>	Workbooks view
Average transactions	Average transactions viewlet shows the average number of CICS transactions for today comparing with that of yesterday and the average number of CICS transaction for the entire week. The average transactions are calculated by taking number of transactions for each data collection (every 5 minutes) within a specific timeframe (for example, today, yesterday or a week) and averaging them.	Performance tab of the workbook summary view

Table 10. OMEGAMON for CICS data analysis reports (continued)

Reports Name	Description	Displayed Location
Average response time	<p>Average response time that is calculated by averaging the average response time of all CICS transactions within a service class. The average response time data is collected within the dashboard display timeframe. The indicator shows the level where the average response time exceeds the goal response time that is set in the OMEGAMON for CICS server..</p> <p>Red indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 400%</p> <p>Orange indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 200 - 400%</p> <p>Yellow indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 100 - 200%</p> <p>Blue indicates that the average response time for all CICS transactions within the service class exceeds the goal response time less than 100%</p>	Performance tab of the workbook summary view
Average CPU time	Average CPU time that is calculated by averaging the CPU time of all CICS transactions in the workbook that are collected within the dashboard display timeframe.	Performance tab of the workbook summary view
Transactions with Performance and Reliability Issues	Transactions with Performance and Reliability Issues displays the table view of all transactions that exceeding the goal response time threshold that is set in the OMEGAMON for CICS server. The report is sorted based on the number of average execution counts to emphasize on the transactions with the potentially biggest impact on the top.	Performance tab of the workbook summary view

Table 10. OMEGAMON for CICS data analysis reports (continued)

Reports Name	Description	Displayed Location
Average response time trends	Average response time trends show history of average response time of a CICS transaction over the dashboard display setting timeframe (for example, hourly, daily, or weekly).	Transaction Analysis view
Average DB2 and File I/O Wait Time of Response Time trends	<p>This chart displays two trend lines: average DB2 wait time of response time trends and average file I/O wait time of response time trends.</p> <p>Average DB2 wait time of response time trends shows history of percent DB2 wait time of the response time of a CICS transaction over the dashboard display setting timeframe (for example, hourly, daily or weekly).</p> <p>Average file I/O wait time of response time trends shows history of percent file I/O wait time of the response time of a CICS transaction over the dashboard display setting timeframe (for example, hourly, daily or weekly).</p>	Transaction Analysis view
Average execution count trends	Average execution count trends show history of average execution count of a CICS transaction over the dashboard display setting timeframe (for example, hourly, daily, or weekly).	Transaction Analysis view
Average CPU time trends	Average CPU time trends show history of average CPU time of a CICS transaction over the dashboard display setting timeframe (e.g. hourly, daily or weekly).	Transaction Analysis view

Table 11. OMEGAMON for CICS data analysis information

View	Information	Description
Transaction Analysis view	Service Class	The name of service class the transaction belongs to.
	Transaction Name	The name of the transaction that is analyzed.

Table 11. OMEGAMON for CICS data analysis information (continued)

View	Information	Description
	Max Execution Count	Maximum number of execution counts of the transaction that is analyzed over the period of the dashboard display timeframe.
	Max CPU Time	Maximum number of CPU times in second of the transaction that is analyzed over the period of the dashboard display timeframe.
	Max Response Time	Maximum number of response times in minute of the transaction that is analyzed over the period of the dashboard display timeframe.
	Transaction Exceeding Goal Response Time	Number of transactions that the response time is going over the goal response time set in IBM OMEGAMON for CICS server.
	Average Response time	Average response time in seconds for a transaction.
	Average execution count	Average number of execution count for a transactions.
	Average CPU time	Average CPU time in milliseconds for a transaction.

Analyzing transaction data

ADDI Extensions provides the detailed CICS transaction analysis within a service class.

Complete the following steps to view the response time, CPU time or execution count analysis of CICS transaction. Before you begin this activity, you need to add at least one workbook with associated OMEGAMON for CICS data provider.

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Performance** tab if you are not yet on the tab.
4. On the **Performance** tab, select the **OMEGAMON** tab if you are not yet on the tab.
5. View the reports on the **OMEGAMON** tab. The latest analysis information such as number of average transactions, average CPU time, and average response time are displayed.
6. Go to the **Transaction Analysis** view by using one of the following steps.
 - a. Select the transaction name that you want to view the transaction analysis information from either **Top Average Execution Count** report or **Transaction Exceeding Response Time Threshold** report.
 - b. Select the name of the service class that you want to view analysis information. The list of all CICS transactions within that service class displays. Then, select the transaction name that you want to view the transaction analysis information.

7. The **Transaction Analysis** view displays with the following information calculating from the latest data that is collected within the default timeframe that is set when you create the collection. For more information, see [“OMEGAMON for CICS related reports and information”](#) on page 238.
 - Maximum execution count
 - Maximum CPU time
 - Maximum response time
 - Number of transactions exceeding goal response time
 - Average response time and average response time trends
 - Average DB2 and File I/O Wait Time of Response Time trends
 - Average Execution count and average execution count trends
 - Average CPU time and average CPU time trends
8. Hover over the trend line to view data on that trend line.
9. Observe the warning icon on top of the average response time trends. The icon indicates the time where the maximum response time occurs.
10. Observe the DB2 and File I/O wait time trends to see the area where the DB2 or File I/O wait time is above the red line which is the acceptable value. Currently it is set to 50%.
11. Select different timeframe, for example, Hourly, Daily or Weekly, on top of the trend chart to zoom in or zoom out information.

Analyzing and displaying performance data from SMF and APA data

The operational data from System Management Facility (SMF) and the Application Performance Analyzer (APA) data are analyzed and displayed in different views. You can navigate through different views of performance data analysis results for a workbook.

Note: For SMF, ADDI Extensions only uses SMF record type 30 for analysis.

Before you start this task, you need to add at least one System Management Facility data provider to the system. For more information, see [“Adding a workbook”](#) on page 217.

Navigating through different views of SMF and APA data analysis results

The performance data from System Management Facility (SMF) and Application Performance Analyzer (APA) data are analyzed and displayed in several views: Workbooks view, Summary view for a workbook, and Job Analysis view.

Workbooks view

The Workbooks view is the home page of ADDI Extensions. The page will be displayed after you logging in or you can get to the Workbooks view by selecting the Workbooks menu from the main menu panel.

The Workbooks view is the view that shows the status of all the workbooks in the system. Each workbook displays its latest data analysis results from their associated data provider. For the workbooks that are associated with SMF data provider, a job with the highest CPU time is displayed on the Workbooks view. For more information, see [“SMF and APA related reports and information”](#) on page 244.

Summary view for a workbook

From the Workbooks view, select the name of any workbook that is associated with the SMF data provider to navigate to the Summary view of a workbook. The SMF related reports is displayed on the **System Management Facility (SMF)** tab under **Performance** tab of the summary view.

The Summary view of SMF data provider displays performance data of all the jobs in a workbook. You can click on the column header of the table to sort the data on that column in an ascending or descending order to view the jobs with performance issues. You can find the following information that displays on this view:

- CPU time
- Execute Channel Program time count
- Service Units count
- Elapsed time

For more information, see [“SMF and APA related reports and information”](#) on page 244.

Job analysis view

From the Summary view of a workbook, select the job name under the **Job with Performance Issues** report. The Job analysis view is composed of the graph area and report area. By default, it displays the trending graphs of the selected job. On the trending graph, when there is an APA report available, the APA report icon is displayed on the graph. You can select the **APA report** icon to display APA report on the bottom of the graphs. For more information, see [“SMF and APA related reports and information”](#) on page 244.

SMF and APA related reports and information

You can find the details of reports and information that is displayed for System Management Facility (SMF) and Application Performance Analyzer (APA) data analysis.

Report Name	Description	Displayed Location
Job with highest CPU time	The name of job which has the highest CPU time within the scope of a workbook and its CPU time in seconds.	Workbook view
Jobs with Performance Issues	The table displaying the list of all the jobs in the scope of a workbook with their performance information e.g. CPU time, executable channel time, service units and elapsed time count. By default it sorts by highest CPU time to lowest CPU time.	Workbook summary view
CPU Time trends	The trends line showing CPU time for a selected job.	Job analysis view
Executable Channel Program Time trends	The trends line showing the count of Execute Channel Program time for a selected job.	Job analysis view
Total Services Unit trends	The trends line showing the amount of total service units consumed by a selected job.	Job analysis view
CPU Services Unit trends	The trends line showing the amount of CPU service units consumed by a selected job.	Job analysis view
I/O Services Unit trends	The trends line showing the amount of I/O service units consumed by a selected job.	Job analysis view
MSO Services Unit trends	The trends line showing the amount of main storage service units consumed by a selected job.	Job analysis view

Table 12. System Management Facility (SMF) and Application Performance Analyzer (APA) Reports (continued)

Report Name	Description	Displayed Location
SRB Services Unit trends	The trends line showing the amount of Service Request Block (SRB) service units consumed by a selected job.	Job analysis view
Application Performance Analyzer report	The table displaying the list of all programs within a selected job with their performance information e.g. CPU time, executable channel time count, service units count and elapsed time.	

Table 13. System Management Facility (SMF) and Application Performance Analyzer (APA) Information

View	Information	Description
Workbook summary view	Job	The name of job within a workbook.
	CPU time	The amount of time CPU was used for processing a job.
	Executable Program Channel (EXPC) time count	The count of call for low level-device access by a job.
	Service Units (SRVU) count	The amount of service unit consumed by a job.
	Elapsed time	The elapsed time processing a job.
Job analysis view	Program	The name of program within a job.
	CPU time	The percentage of CPU time used for processing a program.
	Measurements	The APA measurement count for a program.
	Description	The amount of service unit consumed by a program.

Analyzing the job performance data

You can analyze the job performance data from System Management Facility (SMF) and Application Performance Analyzer (APA).

Note: Before you start analyze the job performance data, you need to add at least one workbook with the associated System Management Facility data provider.

Complete the following steps to view the job performance:

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADDI Extensions. If you are not yet on the **Workbooks** view, click the **Workbooks** tab on the header to go to the **Workbooks** view.
2. Select the name of the workbook that you want to view analysis information.

3. On the summary view of that workbook, select the **Performance** tab if you are not yet on the tab.
4. On the **Performance** tab, select the **System Management Facility (SMF)** tab if you are not yet on the tab.
5. View the **Job with Performance Issues** report on the **System Management Facility (SMF)** tab. You can find the following information.
 - CPU time in second
 - Execute Channel Program (EXCP) time count
 - Service Units count
 - Elapsed time in second
6. Optional : Search for jobs by the job name in the search box on the top of the table.
7. Select the job name to go to the analysis of the selected job.
8. View the two trends lines to compare the performance data of the selected job. By default, the view displays the trends of CPU time and Execute Channel Program Time.
9. Optional: Click the label of trend line and select the different data to display such as CPU Service Units, I/O Service Units, MSO Service Units, and SRB Service Units.
10. Optional : Select the **APA report** icon on the trends line if it is available on the trend line to view the APA report that corresponds to the date you selected on the trend line.
11. View the following information of the APA report:
 - CPU time in seconds
 - Execute Channel Program (EXCP) time count
 - Service Units count
 - Elapsed time in second

Performing Business Rule Discovery

For a workbook that is associated with a Business Rule Discovery data provider, you can use this workbook to discover candidate business terms and track in which application artifacts or enterprise artifacts these business terms appear. You can also use the workbook to manage status of business terms in order to collaborate with your peers.

Business Rule Discovery workbook reports and information

You can find the details of reports and information that are displayed for the Business Rule Discovery workbook.

<i>Table 14. Business Rule Discovery workbook reports and information</i>		
Displayed Location	Report/Information	Description
Workbooks view	Business Terms	The number of all business terms within a workbook. If you create a workbook by importing the business terms, ADDI Extensions calculates the number from counting all the imported terms.
	Business Rule Packages	The number of all business rule packages that are defined within a workbook.

Table 14. Business Rule Discovery workbook reports and information (continued)

Displayed Location	Report/Information	Description
Keywords Exploration tab > Summary header	Keywords associated with Business Terms	The bar chart shows the number of keywords that either are promoted as business term or associated with one or more business terms.
	Keywords from Source Code (AD)	The number of all the keywords that are discovered from source code by analyzing the keyword usages based on the weights configuration in workbook setting through Application Discovery.
	Keywords from Enterprise Artifacts	The number of all the keywords that are discovered from enterprise artifacts.
	Associated Snippets	The number of all the snippets that are defined and associated with business terms within a workbook.
	Business Terms	The number of all the business terms that are defined within a workbook. If you create a workbook by importing the business terms, ADDI Extensions calculates the number from counting all the imported terms as well as those terms that are created from the discovery by users.

Table 14. Business Rule Discovery workbook reports and information (continued)

Displayed Location	Report/Information	Description
Keywords Exploration tab > Application Discovery tab	Keywords	The name of discovered keywords.
	Usage In Conditions	The number of keywords occurrences in condition statements such as IF statement within source files in scope for a workbook.
	Computations	The number of keywords occurrences in computation statements such as COMPUTE within source files in scope for a workbook.
	Map	The number of keywords occurrences that are used for screen display statement.
	Usage in File	The number of keywords occurrences in read or write file statements within source files in scope for a workbook.
	DB	The number of keywords occurrences in read or write to database within source files in scope for a workbook.
	MQ	The number of keywords occurrences in MQ statements.
	Programs Involved	The number of keywords occurrences that program call within source files in scope for a workbook.
	Usage in Enterprise Artifacts	The number of keywords occurrences in enterprise artifacts in scope for a workbook .
	Business Terms Associated	The number of business terms that are associated with the keyword. The association can be through the promotion of keyword as a business term or the association with a business term.

Table 14. Business Rule Discovery workbook reports and information (continued)

Displayed Location	Report/Information	Description
Keywords Exploration tab > Enterprise Artifacts tab	Usage in Artifacts	The number of keywords occurrences in enterprise artifacts in scope for a workbook.
	Number of Artifacts	The number of enterprise artifacts containing the keyword.
	Business Terms Associated	The number of business terms that are associated with the keyword. The association can be through the promotion of keyword as a business term or the association with a business term.
	Associated in Application Discovery	Check mark displayed when the keyword is also discovered by the analysis of Application Discovery through the usages in source file.
Business Terms tab > Summary header	Business Term Summary	The percentage of approved business terms form total number of business terms that are defined.
	Total	The number of all business terms that are defined within a workbook. If you create a workbook by importing the business terms, ADDI Extensions calculates the number from counting all the imported terms as well as those created from the discovery by users.
	With Artifacts Relationship	The number of business terms that have relationship to one or more artifacts in the enterprise artifacts that is associated with the workbook.
	Without Artifacts Relationship	The number and list of business terms that have no relationship to artifacts in the enterprise artifacts that are associated with the workbook.
	With Associated Snippets	The number of business terms that have snippets associated with.
Business Terms tab > Business Terms table	Business Terms	The name of a business term.
	Definition	The description of a business term.

Table 14. Business Rule Discovery workbook reports and information (continued)

Displayed Location	Report/Information	Description
	No. of Artifacts Relationship	The number of artifacts that have relationship with a business term.
	No. of Snippets Associated	The number of snippets that are associated with a business term.
	Implementation Names	The list of implementation names that are identified for a business term.
	Status	The status of a business term. Five statuses are available: New, In Progress, Waiting For Approval, Approved, and Rejected.
Business Rule Packages tab > Summary header	Business Rule Packages	The percentage of approved business rule packages form total number of business rule packages that are defined.
	Total	The number of all business rule packages that are defined within a workbook.
	With Business Terms associated	The number of business rule packages that have business terms associated with.
	With Snippets associated	The number of business rule packages that have snippets associated with.
Business Rule Packages tab > Business Rule Packages table	Business Rule Packages	The name or short description or identifier of a business rule package.
	Description	The description of a business rule package.
	Tags	The list of tags that are assigned to a business rule package.
	Associated Business Terms	The list of business terms that are associated with a business rule package.
	No. of Snippets Associated	The number of snippets that are associated with a business rule package.
	Status	The status of a business rule package. Five statuses are available: New, In Progress, Waiting For Approval, Approved, and Rejected.

Working with keywords

A keyword is a term that ADDI Extensions discovers and recommends as a potential business term or implementation name. ADDI Extensions analyzes the usages of keywords in the source code or the enterprise artifacts to discover keywords.

The analysis of source code is done through Application Discovery (AD) by analyzing the usages of all possible data names or variables within the source files in the scope for a workbook. AD takes the following types of usages into the analysis:

- Usage in included files
- Usage in conditions
- Usage in computations
- Usage in map
- Usage in file input/output
- Usage in database input/output
- Usage in MQ

The number of usages for each type is calculated with the usage weights that are set in the workbook to decide the ranking of the discovered keywords.

The keyword analysis of enterprise artifacts is performed when ADDI Extensions scans the files in the Enterprise Artifact data provider. ADDI Extensions automatically discovers the keywords and groups them based on the likelihood that the keywords can be the same term. For more information, see [“Enterprise artifacts keywords discovery process”](#) on page 253.

Exploring keywords

Complete the following steps to explore keywords:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Click the name of the workbook that contains business terms to explore keywords.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Select the **Keywords Exploration** tab. A list of keywords appears on the left with the usages information.

Note: The keywords found in Application Discovery are listed under the **Application Discovery** tab and the keywords found in enterprise artifacts are listed under the **Enterprise Artifacts** tab. Select the tab name to switch between two tabs.

5. View the information of keyword usages to help decide if a discovered keyword is a business term or an implementation name of a business term. Optionally, you can perform the following actions.
 - Promoting a discovered keyword to a business term
 - a. Click the overflow menu (vertical ellipsis) icon on the right of the discovered keyword.
 - b. Select **Promote to Business Term** from the overflow menu. The discovered keyword is created as a business term.
 - Associating a discovered keyword to the implementation name list of one or more business terms
 - a. Click the overflow menu (vertical ellipsis) icon on the right of a discovered keyword.
 - b. Select **Associate to Business Term** from the overflow menu.
 - c. Select one or more business terms that you want to associate the keyword with. The discovered keyword becomes the implementation name of the selected business term.
 - Viewing the affinities keywords.
 - a. Click the overflow menu (vertical ellipsis) icon on the right of the discovered keyword.

- b. Select **View Affinities** from the overflow menu. The **Affinities** page appears with the list of keywords that are used with the selected keyword and the usages information.

Note:

- The same keyword might be defined multiple times in the application. Keyword usages are grouped and displayed by variable definition location: included files and programs.
- To expedite the discovery process, you can search for a specific keyword that you know or scope the list of keywords that you want to explore based on where they are defined. For more information, see [“Searching and scoping the list of keywords”](#) on page 252.

6. Select the name of keyword to view detailed usages of a keyword.

Note: The usages that are displayed on this page can either be the direct usages or indirect usages. The indirect usage is the usage which refers to variable that keyword is part of. For example the record in which that keyword is one attribute of that record.

7. Expand and select an item on the left panel to view the usage statement of a keyword. The usage statement of a keyword is highlighted on the middle panel.

Notes:

- The keywords found in Application Discovery are listed under **Application Discovery** tab and the keywords found in enterprise artifacts are listed under **Enterprise Artifacts** tab. Select the tab name to switch between two tabs.
- For the Application Discovery keywords, the same keyword might be defined multiple times in the application. Keyword usages are grouped and displayed by variable definition location: included files and programs.

After reviewing the details of keyword usages, you may decide to perform one of the following actions.

- [“Defining snippets and associating them with business terms”](#) on page 253
- [“Viewing affinity keywords and their mutual statement”](#) on page 254

Searching and scoping the list of keywords

Large applications or projects can have thousands of keywords, which makes the keyword exploration activities exhaustive and time-consuming. To help expedite the process, you can search for a specific keyword that you know or scope the list of keywords that you want to explore based on where they are defined.

Perform the following actions to search or scope the list of keywords to explore.

1. If not yet selected, select the **Keywords Exploration** tab. A list of keywords appears on the left with the usages information.

Note: The keywords that are found in Application Discovery are listed under the **Application Discovery** tab and the keywords that are found in Enterprise Artifacts are listed under **Enterprise Artifacts** tab. Select the tab names to switch between two tabs.

2. To search for a specific keyword, enter the keyword name or part of the keyword name in the **Search** box on the top of the keywords list. ADDI Extensions returns all the keywords that have part of the search string that you enter. For example, the string `record` returns all the keywords that contain the `record` string, such as `record`, `ws-records`, `record-id`, `record-name`, `recordA`.

Note: For Application Discovery keywords, you can use the following wildcard characters to search for the keywords. Wildcard search is not accepted for the Enterprise Artifacts keywords.

- `"*` matches none or more non-space characters and the other text is exactly match the search string. For example, `record*` returns all the list of keyword that begin with `record` such as `record`, `record-id`, `record-name`, `record-type`, and `recordA`.
- `"_"` matches one or more non-space character and the other text has part of the search string. The behavior of `"_"` is similar to that of adding `"*"` before and after the search string. For example, `record_` is same as `*record_*`, which returns all the list of keywords that contain the search string

and are followed by at least one character such as `ws-record-id`, `in-record-id`, `record-id`, `record-name`, `record-type`, and `recordA`.

- `"_ "` with spacebar matches exactly one non-space character. For example, `record_(blank)` returns the keywords that are started with `record` and followed by one character (`recordA`) or `record__ (blank)` returns the keywords that are started with `record` and followed by two characters (`recordAB`).

3. To filter the list of keywords that are displayed or filter the search scope:

- a. Select the overflow menu (vertical ellipsis) icon next to the **Search** box on the top of the keywords list.
- b. Select one of the options in the dialogue that opens.
 - Select **Defined in Copybook** to search for the keywords that are defined in a specified copybook.
 - Select **Defined in Program** to search for the keywords that are defined in a specified program.
 - Select **Defined in Record** to search for the keywords that are defined in a specified record.
- c. Enter the name of a copybook, a program, or a record next to the option that you select to scope the search results.

Note: You can use wildcard characters as what you do while searching for the keywords.
- d. Click **Apply** to apply the filter or click **Clear** to clear the filter selection.

Note: When you view the keyword details page, define snippets, and associate them with business terms from the filtered results, the filter is saved for the next time you review the implementation of the business terms.

Enterprise artifacts keywords discovery process

Automated keywords discovery is one of the key cognitive features of ADDI Extensions that help identify potential important terms within both business artifacts and IT artifacts. In the business artifacts, the important terms are potential business terms while in the IT artifacts the important terms are mainly implementation names of these business terms. The process to identify these terms is usually labor intensive and time-consuming. ADDI Extensions provides the automation of this process by using Natural Language Processing (NLP) method. The output of the automated keywords discovery is the list of keywords and their potential implementation names that are sorted by the confidence score. The confidence score is a likelihood score that indicates whether a keyword is an important term or not. The higher score means that the term has higher potential to be a business term. Likelihood score is calculated from the training of machine learning based classification model which considers two main attributes:

- **Keyword features:** The important properties of the term which help indicate that whether a term is a keyword or not. The properties could be, for example, the appearances of the terms in the artifacts, term length or location of its first appearance. These features can be either programming language dependent or programming language independent.
- **Known business terms:** When there is an imported list of business terms or a list of business terms in a workbook, these terms providing a higher weight to the confidence scores of discovering keywords.

Defining snippets and associating them with business terms

A snippet is the statements within artifacts which contain one or more business terms. You can highlight one or more lines of statements within an artifact and save as a snippet. You can associate a snippet with business terms or business rule packages in that artifact when you explore the keyword details.

Complete the following steps to take an artifact snippet:

1. Go to the **Keyword** details page by exploring keywords as described in [“Exploring keywords” on page 251](#).
2. On the Keyword details page, expand and select the keyword usage from the left panel. ADDI Extensions highlights the statement usage of a keyword on the middle panel.

Note: For one keyword usage occurrence, ADDI Extensions only allow one snippet to be defined and associated with business terms. You can associate this statement with one or more business term as a snippet or define a new snippet to associate with one or more business term as described in the following steps.

3. Optional: Highlight one or more line within the artifact and select **Redefine snippet** from the pop-up menu to define a new snippet.
4. Optional: Provide a short description of the snippet on the right panel under the **Snippet Description** field.
5. Perform either one or both of the following actions:
 - Check one or more boxes in front of the business terms to associate with the snippet from the **Business Terms** tab on the right pane.
 - Check one or more boxes in front of the business rule packages to associate with the snippet from the **Business Rule Packages** tab on the right pane.

Note: When you perform both actions at the same time, the snippet is associated with selected business terms and business rule packages independently.

6. If not yet selected, select one or more business terms that you want to associate with the snippet.
7. Click **Save** to save the snippet and its associations or click **Cancel** to cancel the changes you made.
8. Optional: Click **Restore to the original state** to discard the new snippet that you redefined, to display the original usage statement, and to remove the association to the selected business term.

Viewing affinity keywords and their mutual statement

Affinity keywords are the keywords that are used together in one or more statements in the source files. A mutual statement is the statement that contains affinity keywords in the scope for analysis.

For example, the following statement is mutual statement of A, B, and C, which are affinity keywords because they are used together in the same condition statement.

```
IF (A > 3) and (B > 3) and (C = 5)
THEN do something
ELSE do something else
```

ADDI Extensions analyzes the affinity keywords based on the usage of the following criteria:

- Usage in condition statements
- Usage in computation statements
- Usage in assignment statements
- Usage in program

ADDI Extensions helps you discover additional business terms by viewing affinity keywords and their usage information along with the mutual statement.

Complete the following steps to view affinity keywords and their mutual statements:

1. Go to the **Keyword** details page by exploring keywords as described in [“Exploring keywords”](#) on page 251.
2. On the **Keyword** details page, select the **View Affinity** button on the right of header area. The **Affinity Keywords** page appears. You can see a list of affinity keywords and usage information are listed in the table.
3. You can perform one of the following activities.
 - To promote a keyword to a business term, complete the following steps.
 - a. Click the overflow menu (vertical ellipsis) icon under the **Actions** column of a keyword that you want to promote to a business term.
 - b. Select **Promote to Business Term** from the overflow menu. The keyword is created as a business term.

- To associate a keyword to a business term, complete the following steps:
 - a. Click the overflow menu (vertical ellipsis) icon under the **Actions** column of a keyword that you want to associate to a business term.
 - b. Select **Associate to Business Term** from the overflow menu.
 - c. Select a business term that you want to associate the keyword with. The keyword becomes the implementation name of the selected business term.
- To view the mutual statements of one or more keywords, complete the following steps.
 - a. Select one or more keywords that you want to view mutual statements.
 - b. View the header area to see if any mutual usages of the selected keywords are available.
 - c. If mutual usages are available, select the **Show Mutual Statements** button to view mutual statements of all selected keywords.

Note: If no mutual usages are available, the **Show Mutual Statements** button is disabled.
 - d.
- 4. Select to view mutual statements from the list on the left pane. You can associate the statement as a snippet to business terms or redefine new snippets before you associate them to a business term from this page. For more information, see [“Defining snippets and associating them with business terms” on page 253.](#)

Managing a business term

Business term is commonly used in business operations. It provides a definition of the key business information. Learn about how you can manage the business terms.

Adding a business term

You can add a business term from the **Business Terms** tab of a workbook.

Complete the following steps to add a business term from the **Business Terms** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Click the name of the workbook that contains the business term to add.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms in the workbook appears in the table.
5. Click the overflow menu (vertical ellipsis) icon on the right of the header section.
6. Select **Add Business Term** from the overflow menu that appears. The Add Business Terms dialog box opens.
7. In the Add Business Terms dialog box, provide the following information:
 - **Business Term:** Provide a name for the business term.

Note: The name with the plus (+) sign can be used but it will not work well when you search for the business term.
 - **Definition:** Provide the definition of the business term.
 - **Location:** (Optional) Select a folder of the business term. If no folder is selected, the business term is located in the root folder.
 - **Status:** Select one of the choices from the drop-down list to specify the status of the business term. Five statuses are available for business terms: New, In Progress, Waiting For Approval, Approved, and Rejected. By default, the "New" status is selected when you create a business term.
8. Click **Add** to add the business term or **Cancel** to cancel the operation.

Managing a business term folder

You can create a folder to organize business terms that are related. After you create a folder, you can edit and delete the folder as needed. You can also move business terms and folders to a new folder location.

Adding a business term folder

Complete the following steps to add a business term folder from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of workbook that contains the business term folder to add.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms and their folders in the workbook appears in the table.
5. Click the overflow menu (vertical ellipsis) icon on the right of the Business Term header section.
6. Select **Add Folder** from the overflow menu that appears. The Add Business Term Folder dialog box opens.
7. Enter a unique name in the **Business Term Folder** field.
8. Select the folder location in the **Location** section. By default, the root folder is selected.
9. Click **Save** to add the folder or click **Cancel** to cancel the operation.

Editing a business term folder

Complete the following steps to edit a business term folder from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of workbook that contains the business term folder to edit.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms and their folders in the workbook appears in the table.
5. Move the mouse pointer over the business term folder to edit.
6. Click the overflow menu (vertical ellipsis) icon on the right of the business term folder.
7. Select **Edit** from the overflow menu that appears. The Edit Business Term Folder dialog box opens.
8. Enter the name of the business term folder.
9. Click **Save** to update the business folder or click **Cancel** to cancel the changes.

Deleting a business term folder

Complete the following steps to delete a business term folder from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of workbook that contains the business term folder to delete.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms and their folders in the workbook appears in the table.
5. Move the mouse pointer over the business term folder to delete.
6. Click the overflow menu (vertical ellipsis) icon on the right of the business term folder.
7. Select **Delete** from the overflow menu that appears. The Delete Business Term Folder dialog box opens.
8. Click **Delete** to delete the business term folder or click **Cancel** to cancel the deletion.

Note: When you delete a business term folder, all the folders and business terms within that folder will be deleted.

Deleting multiple business term folders

Complete the following steps to delete more than one business term folder from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of workbook that contains the business term folders to delete.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms and their folders in the workbook appears in the table.
5. Select the check boxes in front of the folders and sub-folders that you want to delete.
6. Click **Trash** icon on the Business Term header section. The Delete Business Term Folder dialog box opens.
7. Click **Delete** to delete the business term folders or click **Cancel** to cancel the deletion.

Note: When you delete a business term folder, all the folders and business terms within that folder will be deleted.

Moving business terms and business term folders

Complete the following steps to move the location of one or more business terms and business folders from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of workbook that contains the business terms and business term folders that you want to move.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Terms** tab. A list of all defined business terms and their folders in the workbook appears in the table.
5. Select the check boxes in front of the folders and sub-folders that you want to move.
6. Click **Move** button on the Business Term header section. The Move Business Terms and Folders dialog box opens.
7. Select a folder as the new location for the business terms and folders. You can use the search box to search for folders.
8. Click **Move** button to move the selected business terms and folders or click **Cancel** to cancel the move.

Editing a business term

You can edit a business term directly from the **Business Terms** tab of a workbook or from the list of Business Terms in **Keywords Exploration** tab of a workbook.

Complete the following steps to edit a business term from the **Business Terms** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business term that you want to edit.
3. Optional: Select **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Select the **Business Terms** tab. A list of all business terms in the workbook appears in the table.
5. Click the **Edit** icon on the right under the **Action** column of the business term that you want to edit. The **Edit Business Term** page appears.
6. Edit one or more of the following information.

- **Business Term:** Update the name for the business term.
- **Definition:** Update the definition of the business term.
- **Status:** Select one of the choices from the drop-down list to update the status of the business term. Five statuses are available for business terms: New, In Progress, Waiting For Approval, Approved, and Rejected.

7. Click **Save** to save changes or click **Cancel** to cancel the changes.

Complete the following steps to edit a business term from the list of Business Terms in the **Keywords Exploration** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business term that you want to edit.
3. Optional: Select **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Select the **Keywords Exploration** tab. A list of all defined business terms in the workbook appears on the right.
5. Click the overflow menu (vertical ellipsis) icon on the right of the business term that you want to edit.
6. Select **Edit** from the overflow menu that appears. The **Edit Business Term** page appears.
7. Edit one or more of the following information.
 - **Business Term:** Update the name for the business term.
 - **Definition:** Update the definition of the business term.
 - **Status:** Select one of the choices from the drop-down list to update the status of the business term. Five statuses are available for business terms: New, In Progress, Waiting For Approval, Approved, and Rejected.
8. Click **Save** to save changes or click **Cancel** to cancel the changes.

Viewing the details of a business term

After you create business terms and associate them with snippets and keywords, you can view details of the business terms from the **Business Terms** tab of a workbook.

Complete the following steps to view the details of a business term.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains business terms that you want to view.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is what is associated with a workbook, no other tabs will be displayed.
4. Select the **Business Terms** tab. A list of business terms that are defined in the workbook appears in the table. You can find the summary information for the business terms such as definition, the number of artifacts relationship, the number of snippets associated, the list of implementation names (keyword associated), and the statuses of business terms.
5. Click the name of a business term that you want to view detailed information. A list of all implementation names that are associated to this business term appears on the left pane.
6. Click an implementation name to expand and view all the associated snippets that are defined by members of the workbook.

Notes:

- For an implementation name that is discovered from source files, you need to expand program name in order to view snippets.
- You can also select the overflow menu (vertical ellipsis) icon next to the implementation name and select **View Exploration Details** to go to the keyword exploration details page. The keyword exploration details page is filtered with the results down to the associations that were previously made when you defined the snippets and associated them with business terms.

7. Select a snippet item to view the snippet content.

- For implementation names that are discovered from source files, the snippet content shows the program name and line numbers.
- For implementation names that are discovered from enterprise artifacts, the snippet content shows the artifact name and line numbers.

You can find the snippet that is highlighted in the middle pane with artifact content and the snippet information that is shown on the right pane.

You can perform the following actions on the business term details page.

- [“Associating or disassociating with business rule packages” on page 259](#)
- [“Deleting the association between a snippet and a business term” on page 259](#)

Associating or disassociating with business rule packages

You can associate or disassociate a business term and its snippets with one or more business rule packages when you view the details of a business term.

To associate a business term and its snippets with one or more business rule packages, complete the following steps:

1. On the details page of a business term, expand and select the keyword usage from the left pane. ADDI Extensions highlights the statement usage of a keyword on the middle pane.

Note: You can only associate a business term with business rule packages when one or more snippets are associated with that business term.

2. Optional: Complete the **Snippet Description** field on the right pane.
3. Select one or more check boxes in front of the business rule packages to associate the business term and snippets with business rule packages. Or clear one or more check boxes in front of the business rule packages to remove the association of business term and its snippet with business rule packages.
4. Click **Save** to save the change or click **Cancel** to discard the change.

Deleting the association between a snippet and a business term

On the details page of a business term, you can delete the association between a snippet and the business term.

To delete a snippet that is associated with a business term, complete the following steps:

1. On the details page of a business term, expand and select the keyword usage from the left pane. ADDI Extensions highlights the statement usage of a keyword on the middle pane.
2. Click the overflow menu (vertical ellipsis) icon next to the snippet item on the left pane.
3. Select **Delete Association** to remove the association between the selected snippet and the business term.
4. Click **OK** on the confirmation dialog that appears to confirm the deletion or click **Cancel** to discard the change.

Deleting a business term

You can delete a business term directly from the **Business Terms** tab of a workbook or from the list of Business Terms in the **Keywords Exploration** tab of a workbook.

Complete the following steps to delete a business term from the **Business Terms** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business term you want to delete.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tab will be displayed.
4. Select the **Business Terms** tab. A list of all business terms in the workbook appears in the table.

5. Select the **Delete** icon on the right under the **Action** column of the business term you want to remove from the workbook. The Confirm Delete Business Term dialog box opens.
6. Click **OK** to delete the business term or click **Cancel** to cancel the deletion.

Complete the following steps to delete a business term from the list of Business Terms in the **Keywords Exploration** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business term you want to delete.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tab will be displayed.
4. Select the **Keywords Exploration** tab. A list of all defined business terms in the workbook appears on the right.
5. Click the overflow menu (vertical ellipsis) icon on the right of the business terms you want to delete.
6. Select **Delete** from the overflow menu that appears. The Confirm Delete Business Term dialog box opens.
7. Click **OK** to delete the business term or click **Cancel** to cancel the deletion.

Approving or rejecting a business term

As a member of a workbook, you can approve or reject the business terms from the **Business Terms** tab in order to track the progress of business term discovery process.

Complete the following steps to approve or reject a business term from the **Business Terms** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business term to approve or reject.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tab will be displayed.
4. Select the **Business Terms** tab. A list of all defined business terms in the workbook appears in the table.
5. Click the overflow menu (vertical ellipsis) icon on the right of the business term that you want to approve or reject and select **Edit** from the overflow menu. The Edit Business Term dialog box opens.
6. From the **Status** drop-down list, select **Approved** to approve the business term or select **Rejected** to reject the business term.

Exporting business terms through Business Rule Discovery APIs

IBM ADDI provides Business Rule Discovery (BRD) APIs that allow you to access Business Rule Discovery repository through REST services. You can export business terms by using those APIs.

You can complete the following steps to retrieve the list of business terms through BRD APIs:

1. Browse to `https://ADDI_Extensions_HOSTNAME: ADDI_Extensions_PORT/addi/brd/swagger-ui.html`, for example, to go to the Swagger UI for BRD APIs documentation. For example, `https://localhost:9443/addi/brd/swagger-ui.html`.
2. Click **business-term-storage-resource** to expand the details of business term storage resource APIs.
3. Click **GET /api/workbook/{workbook_uuid}/business_term** to expand the API details.
4. Click the **Try it out** button next to the Parameters header.
5. Enter the following parameter information:
 - **page**: Enter the page number that you want to retrieve the business terms data. The first page starts with 0.
 - **size**: Enter the number of business terms that you want to retrieve in one page. The maximum size is 2,000.

- **sort:** By default, the business terms are sorted in ascending order. You can change the order to descending order.
- **workbook_uuid:** Enter the UUID of the workbook. You can get the UUID of a workbook from the URL of the workbook summary page.

GET /api/workbook/{workbook_uuid}/business_term Retrieve pages of business terms for a specific workbook.

Parameters

Name	Description
page integer(\$int32) (query)	The page you want to retrieve. The first page number is 0. 0
size integer(\$int32) (query)	The size of each page. It should be larger than 0. 20
sort array[string] (query)	Sort setting in the format of: property(asc desc). Default sort order is ascending. Multiple sort settings are supported. Add Item
workbook_uuid string (path)	The uuid of the owning workbook. Must be provided. _1RpI4PiWEeIyrIKFZ11PeQ

Execute Clear

6. Click the **Execute** button below the Parameters section. Business terms and their details are returned as JSON under the Responses section.

Code Details

200

Response body

```
{
  "content": [
    {
      "id": 84,
      "uuid": "72fe51de-4b7f-4f11-98a9-266709925beb",
      "created": "2018-12-05T14:06:49.076+0000",
      "updated": "2019-01-03T16:06:42.720+0000",
      "workbookUuid": "_1RpI4PiWEeIyrIKFZ11PeQ",
      "name": "Patient ID",
      "nameLower": "patient id",
      "description": "Patient ID",
      "status": "IN_PROGRESS",
      "lookups": [
        {
          "id": 111,
          "uuid": "137cf834-2183-4777-86bf-2e8a6034b5a6",
          "created": "2018-12-05T23:43:41.643+0000",
          "updated": "2019-12-05T23:43:41.643+0000",
          "implementationName": {
            "id": 104,
            "uuid": "495c8853-9348-471e-a02a-3853feb9bcf0",
            "created": "2018-12-05T23:43:41.636+0000",
            "updated": "2018-12-05T23:43:41.636+0000",
            "name": "patnstr",
            "nameLower": "patnstr"
          }
        }
      ]
    }
  ]
}
```

Download

```
BusinessTerm {
  description: Represents a Business Terms, which a real world object that may be subject for a business rule or API.
  description: string
  Additional documentation for the Business Term, its usage, and meaning.
  lookups: > [...]
  name: string
  The name of the Business Term. Must be provided and unique within the workbook.
  snippets: > [...]
  status: string
  The status of Business Term.
  Enum:
  > Array [ 7 ]
  uuid: string
  minLength: 0
  maxLength: 36
  The uuid that uniquely identifies the entity. Cannot be omitted.
  workbookUuid: string
  minLength: 1
  maxLength: 36
  The uuid of the owning workbook. Must be provided.
}
```

Managing business rule packages

The business rule package groups business terms and snippets that are related and represents business logic or a business rule. Learn more about how to manage business rule packages.

Adding a business rule package

You can add a business rule package from the **Business Rule Packages** tab of a workbook.

Complete the following steps to add a business rule package from the **Business Rule Packages** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package to add.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Rule Packages** tab. A list of all the business rule packages in the workbook appears in the table.
5. Click the **Add (+)** icon on the right of the header section. The Add Business Rule Package dialog box opens.
6. In the Add Business Rule Package dialog box, provide the following information:
 - **Business Rule Package:** Provide a name or a short introduction of the business rule package.
 - **Description:** Provide a description of the business rule package.
 - **Tags:** Provide one or more tags for the business rule package.
 - **Status:** Select one of the choices from the drop-down list to specify the status of the business rule package. Five statuses are available for business terms: New, In Progress, Waiting For Approval, Approved, and Rejected. By default, the "New" status is selected when you create a business term.
7. Click **Save** to add the business rule package or click **Cancel** the operation.

Editing a business rule package

You can edit a business rule package directly from the **Business Rule Packages** tab of a workbook.

Complete the following steps to edit a business rule package from the **Business Rule Packages** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package to edit.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Rule Packages** tab. A list of all the business rule packages in the workbook appears in the table.
5. Click the **Edit** icon on the right under the **Action** column of the business rule package. The Edit Business Rule Package dialog box opens.
6. Edit one or more of the following information:
 - **Business Rule Package:** Update the name or the short introduction of the business rule package.
 - **Description:** Update the description of the business rule package.
 - **Tags:** Update the tags of the business rule package.
 - **Status:** Select one of the choices from the drop-down list to update the status of the business rule package. Five statuses are available for business terms: New, In Progress, Waiting For Approval, Approved, and Rejected.
7. Click **Save** to save changes or click **Cancel** to cancel changes.

Viewing the details of a business rule package

After you create a business rule package and associate them with snippets and business terms, you can view the details of the business rule packages from the **Business Rule Packages** tab of a workbook.

Complete the following steps to view the details of a business rule package:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package that you want to view.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is what is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Rule Packages** tab. A list of all the business rule packages in the workbook appears in the table. You can find the summary information for the business rule packages such as description, tags, associated business terms, the number of snippets associated, and the statuses of business terms.
5. Click the business rule package that you want to view detailed information. A list of all business terms and snippets associated to this business rule package appears on the left pane.
6. Expand the list of business terms on the left to view all the associated snippets that are defined by members of the workbook.

Note: For the implementation name under each business term, you can select the overflow menu (vertical ellipsis) icon next to the implementation name and select **View Exploration Details** to go to the keyword exploration details page. The keyword exploration details page is filtered with the results down to the associations that were previously made when you defined the snippets and associated them with business terms and (or) business rule packages.

7. Select a snippet item to view the snippet content. You can find the snippet that is highlighted in the middle pane with artifact content and the snippet information that is shown on the right pane.

Note: For each snippet, you can select the overflow menu (vertical ellipsis) icon next to the snippet and select **View Snippet** to view the snippet on the keyword exploration details page.

You can perform the following tasks on this page:

- [“Deleting the association of a business term with a business rule package” on page 263](#)
- [“Managing the snippet execution order” on page 264](#)

Deleting the association of a business term with a business rule package

To delete the association of a business term with a business rule package, complete the following steps:

1. On the details page of a business rule package, expand and select the business term from the left pane.
2. Select the overflow menu (vertical ellipsis) next to the business term item on the left pane.
3. Select **Delete Association** to remove the association between the selected business term and the business rule package.

Note: If there is one or more snippets under the business term, the relationship between snippets and business rule package is preserved after you remove the association between business term and business rule package. For more information about removing the association between snippets and business rule package, see [“Deleting the association of a snippet from keyword with a business rule package” on page 263](#).

4. Click **OK** on the confirmation dialog window to confirm the deletion or click **Cancel** to discard the change.

Deleting the association of a snippet from keyword with a business rule package

Complete the following steps to delete the association of a snippet from keyword with a business rule package:

1. On the details page of a business rule package, expand **Snippet from Keywords** and select a snippet from the left pane.
2. Select the overflow menu (vertical ellipsis) next to the snippet item on the left panel.
3. Select **Delete Association** to remove the association between the selected snippet and the business rule package.
4. Click **OK** on the confirmation dialog window to confirm the deletion or click **Cancel** to discard the change.

Managing the snippet execution order

Snippets that are associated with a business rule package represent sets of business logic that are implemented in the source code. These snippets can be executed in a sequential order based on the business logic that they represent.

Complete the following steps to manage the execution order of snippets.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package that you want to manage the snippet execution order.
3. Optional: Click the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tabs will be displayed.
4. Click the **Business Rule Packages** tab. A list of all the business rule packages in the workbook appears in the table.
5. Click the business rule package to go to the detailed page. Snippets are listed on the right pane and grouped into two groups: **Unordered** and **Ordered**.
6. Perform one or more of the following tasks.

Note: When you select the snippet from the list, the source code content of the snippet is displayed in the middle.

- Move the unordered snippets to **Ordered** group.
 - a. Select the snippet that you want to move from the **Unordered** group.
 - b. Click the down arrow next to the group header to move the snippets to the **Ordered** group.
- Order the sequence of the snippets in the **Ordered** group.
 - a. Click on the snippet which you want to move the order.
 - b. Click up or down blue arrow next to the group header to move the order of the snippet.
- Move the ordered snippets to unordered snippets.
 - a. Select the snippet that you want to move from the **Ordered** group.
 - b. Click the up arrow on the most right of the group header to move the snippets to the **Unordered** group.

Approving or rejecting a business rule package

As a member of a workbook, you can approve or reject the business rule packages from the **Business Rule Packages** tab to track the progress of business rule discovery process.

Complete the following steps to approve or reject a business rule package from the **Business Rule Packages** tab of a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package to approve or reject.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is what is associated with a workbook, no other tabs will be displayed.

4. Click the **Business Rule Packages** tab. A list of all the business rule packages in the workbook appears in the table.
5. Click the overflow menu (vertical ellipsis) icon on the right of the business rule package that you want to approve or reject and select **Edit** from the overflow menu. The Edit Business Rule Package dialog box opens.
6. From the **Status** drop-down list, select **Approved** to approve the business rule package or select **Rejected** to reject the business rule package.

Deleting a business rule package

You can delete a business rule package directly from the **Business Rule Packages** tab of a workbook.

Complete the following steps to delete a business rule package from the **Business Rule Packages** tab of a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that contains the business rule package you want to delete.
3. Optional: Select the **Business Rules Discovery** tab. If only the Business Rule Discovery data provider is associated with a workbook, no other tab will be displayed.
4. Select the **Business Rule Packages** tab. A list of all business rule packages in the workbook appears in the table.
5. Select the **Delete** icon on the left of the business rule package you want to remove from the workbook. The Confirm Delete Business Rule Package dialog box opens.
6. Click **OK** to delete the business rule package or click **Cancel** to cancel the deletion.

Exporting business rule packages through Business Rule Discovery APIs

IBM ADDI provides Business Rule Discovery (BRD) APIs that allow you to access Business Rule Discovery repository through REST services. You can export business rule packages by using those APIs.

Complete the following steps to retrieve the list of business rule packages through BRD APIs:

1. Browse to `https://ADDI_Extensions_HOSTNAME: ADDI_Extensions_PORT/addi/brd/swagger-ui.html`, for example, to go to the Swagger UI for BRD APIs documentation. For example, `https://localhost:9443/addi/brd/swagger-ui.html`.
2. Click **business-rule-package-storage-resource** to expand the details of business rule package storage resource APIs.
3. Click **GET /api/workbook/{workbook_uuid}/business_rule/package** to expand the API details. You can see the details of all parameters that are required for APIs and the details of the response content.
4. Click the **Try it out** button next to the Parameters header.
5. Enter the following parameter information:
 - **page**: Enter the page number that you want to retrieve the business rule packages data. The first page starts with 0.
 - **size**: Enter the number of business rule packages that you want to retrieve in one page. The maximum size is 2,000.
 - **sort**: By default, the business rule packages are sorted in ascending order. You can change the order to descending order.
 - **workbook_uuid**: Enter the UUID of the workbook. You can get the UUID of a workbook from the URL of the workbook summary page.

GET /api/workbook/{workbook_uuid}/business_rule/package Retrieve pages of business rule packages for a specific workbook.

Parameters Cancel

Name	Description
page integer(\$int32) (query)	The page you want to retrieve. The first page number is 0. <input type="text" value="0"/>
size integer(\$int32) (query)	The size of each page. It should be larger than 0. <input type="text" value="20"/>
sort array[string] (query)	Sort setting in the format of: property(.asc desc). Default sort order is ascending. Multiple sort settings are supported. <input type="button" value="Add item"/>
workbook_uuid string (path)	The uuid of the owning workbook. Must be provided. <input type="text" value="_f49okDTEemoUedUNAsCkw"/>

6. Click the **Execute** button below the Parameters section. The business rule packages and their details are returned as JSON under the Responses section.

Server response

Code	Details
200	<p>Response body</p> <pre>{ "content": [{ "id": 3, "uuid": "7263e19d-c5f7-49dd-8e07-14b009cae1b2", "created": "2019-02-20T04:31:53.548+0000", "updated": "2019-02-20T04:31:53.575+0000", "workbookUuid": "_f49okDTEemoUedUNAsCkw", "name": "Patient Identification", "description": "", "status": "IN_PROGRESS", "tags": [{ "id": 2, "uuid": "7b836794-9f87-478c-aa78-b9e92d226e9a", "created": "2019-02-20T04:31:53.589+0000", "updated": "2019-02-20T04:31:53.589+0000", "name": "patient" }] }, { "id": 16, "uuid": "6f2b756a-16d-489d-8854-602222c04ef", "created": "2019-02-20T04:10:27.932+0000" }], "terms": [{ "id": 16, "uuid": "6f2b756a-16d-489d-8854-602222c04ef", "created": "2019-02-20T04:10:27.932+0000" }] }</pre> <p><input type="button" value="Download"/></p> <p>Response headers</p> <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate content-language: en-CA content-type: application/json;charset=UTF-8 date: Mon, 25 Feb 2019 08:06:11 GMT expires: 0 pragma: no-cache strict-transport-security: max-age=31536000 ; includeSubDomains transfer-encoding: chunked x-content-type-options: nosniff x-frame-options: DENY x-powered-by: Servlet/3.1 x-xss-protection: 1; mode=block</pre>

```
BusinessRulePackage {
  description: Represents a Business Rule Package, which is a group of artifacts that contribute to a business rule.
  description string
  name string
  name Additional documentation for the Business Rule Package and its usage.
  snippets string
  status string
  status The status of Business Rule Package.
  Enum:
  > Array [ 7 ]
  tags > [...]
  terms > [...]
  uuid string
  uuid minLength: 0
  uuid maxLength: 36
  The uuid that uniquely identifies the entity. Cannot be omitted.
  workbookUuid string
  workbookUuid minLength: 1
  workbookUuid maxLength: 36
  The uuid of the owning workbook. Must be provided.
}
```

Analyzing and displaying project data from Application Discovery

Application Discovery (AD) data provides static analysis of projects. Project data such as number of artifacts, maintainability index, and number of statements that are collected from AD are analyzed and

displayed in different views. You can navigate through different views of project inventory data analysis results for a workbook.

Before you start this task, you need to add at least one connection with one Application Discovery data provider to the system. For more information, see [“Creating a connection” on page 197](#).

Navigating through different views of project data analysis

Static analysis data for a workbook can be analyzed and displayed in three main views: **Workbooks** view, detailed analysis of a workbook, and **Artifact Composition** view.

- **Workbooks** view

For the workbooks associated with the Application Discovery data provider, the **Workbook** view displays the number of Application Discovery projects within the workbook and their average maintenance status. When there is only one project within a workbook, the name of project is displayed instead of the number of projects. For more information on the report, see [“Project data from Application Discovery reports and information” on page 268](#).

- Detailed analysis of a workbook

From the **Workbooks** view, select the name of any workbook that is associated with Application Discovery data provider to browse the detailed analysis reports. The analysis reports and information of Application Discovery data are displayed on the **Static Analysis** tab of this view.

Four sub-tabs are available on the **Static Analysis** tab.

- The **Overview** tab displays the bubble chart as the summary of overall status for all projects within the scope of a workbook. You can select individual bubble on the bubble chart to navigate to the detailed reports of that project.
- The **Metrics** tab displays the radar chart of different metrics you are interested in. You can compare up to five projects along with the metrics.
- The **Structure** tab displays the information about the shared resources for selected projects.
- The **Trends** tab displays the historical trends of different metrics that you are interested in. You can compare up to five projects for each of the historical trends chart.

Note: You can select up to 5 projects to be analyzed on the **Static Analysis** tab.

For more information, see [“Project data from Application Discovery reports and information” on page 268](#).

Note: For Application Discovery data analysis, you can associate only one Application Discovery data provider with a workbook.

- The **Artifact Composition** view displays the information of all artifacts within a transaction. It can be switched between the following two reports.

- **Artifact Composition graph**

You can navigate to the **Artifact Composition graph** from the **Overview** view of a workbook by selecting the name of an Application Discovery project. When the page to select a transaction for analysis appears, select the transaction you want to analyze. The Artifact Composition graph view appears. For more information about the Artifact Composition graph, see [“Project data from Application Discovery reports and information” on page 268](#). For more information about how to analyze Artifact Composition, see [“Analyzing the Artifact Composition” on page 274](#).

- **Artifact Composition table**

You can navigate to the **Artifact Composition table** view from the **Artifact Composition graph** view by selecting the **Table View** icon on the top right of the view. For more information about the Artifact Composition table, see [“Project data from Application Discovery reports and information” on page 268](#). For more information about how to analyze Artifact Composition, see [“Analyzing the Artifact Composition” on page 274](#).

Project data from Application Discovery reports and information

You can find the details of reports and information that is displayed for Application Discovery data analysis.

Table 15. Static metrics available from Application Discovery data provider

Static Metrics	Description
Active Statement	The total number of statements that can be executed.
Artifacts	The total number of artifacts that are included in an application, where artifacts are entities such as programs, screens or classes.
Cyclomatic Complexity	An estimation of complexity by measuring the number of decision points within a program source code, in which the control flow can continue in more than one way. For more information, see Complexity Reports in Application Discovery .
Delivered Bugs	Delivered Bugs is an estimated number of errors. It correlates with the overall complexity of the software. For the project level, it is calculated by averaging the number of delivered bugs for all type of artifacts within a project.
Maintainability Index	Maintainability Index(MI) is a software metric that measures how maintainable (easy to support or change) the source code is. This is calculated based on Lines of Code, Cyclomatic complexity, and Halstead volume. The MI value is 0 - 100. A higher value indicates better maintainability. Halstead volume is an estimation of complexity measures based on number of variables, for example, the number of operands, and operators for expressions, program length, vocabulary and the number of delivered bugs.
Source Lines of Code	The total number of lines of code.
Unreachable Code	The percentage of statements that are within “live subroutines” (subroutines that can be reached by the control flow) out of the total number of statements.

Table 16. Data analysis reports for project data from Application Discovery

Reports Name	Description	Displayed Location																				
Maintenance Status	<p>The Maintenance Status is calculated based on the Maintainability Index (MI) and Unreachable Code threshold setting in the following table. The assumption is that the Maintainability Index metric has double weight of the Unreachable Code metric.</p> <table border="1"> <thead> <tr> <th colspan="4">Unreachable Code (Weight)</th> </tr> </thead> <tbody> <tr> <td>Maintainability Index (Weight)</td> <td>Poor (3)</td> <td>Acceptable (2)</td> <td>Good (0)</td> </tr> <tr> <td>Poor (6)</td> <td>Red</td> <td>Yellow-poor</td> <td>Yellow-good</td> </tr> <tr> <td>Acceptable (4)</td> <td>Yellow-medium</td> <td>Yellow-good</td> <td>Yellow-good</td> </tr> <tr> <td>Good (0)</td> <td>Yellow-good</td> <td>Blue</td> <td>Blue</td> </tr> </tbody> </table>	Unreachable Code (Weight)				Maintainability Index (Weight)	Poor (3)	Acceptable (2)	Good (0)	Poor (6)	Red	Yellow-poor	Yellow-good	Acceptable (4)	Yellow-medium	Yellow-good	Yellow-good	Good (0)	Yellow-good	Blue	Blue	Workbooks view
Unreachable Code (Weight)																						
Maintainability Index (Weight)	Poor (3)	Acceptable (2)	Good (0)																			
Poor (6)	Red	Yellow-poor	Yellow-good																			
Acceptable (4)	Yellow-medium	Yellow-good	Yellow-good																			
Good (0)	Yellow-good	Blue	Blue																			

Table 16. Data analysis reports for project data from Application Discovery (continued)

Reports Name	Description	Displayed Location
Overview	Bubble chart shows overall status of all projects within the scope of a workbook. By default, X-Axis represents percent of unreachable code, Y-Axis represents Cyclomatic Complexity value and Bubble size represents the size of project as number of source lines of code.	Workbooks view-> Static Analysis tab -> Overview tab
Artifact and Lines of Code Breakdown	<p>Artifact and Lines of Code Breakdown report displays total number of artifacts comparing with total number of lines of code breakdown by type of artifacts.</p> <p>The total number of artifacts calculated by the number of artifacts that are included in a project, where artifacts would be entities such as programs, screens, and classes.</p> <p>The total number of lines of code that is calculated by the number of lines of code in a project for each type of artifacts.</p>	Workbooks view-> Static Analysis tab -> Overview tab -> detailed project reports
Maintainability Index versus Delivered Bugs	Maintainability Index versus Delivered Bugs displays the line chart over time comparing maintainability index and delivered bugs.	Workbooks view-> Static Analysis tab -> Overview tab -> detailed project reports
Unreachable Code Trend	Unreachable code trend displays the percentage of statements that are within routines that cannot be reached by the control flow out of the total number of statements over time.	Workbooks view-> Static Analysis tab -> Overview tab -> detailed project reports
Total and Active Statements Trend	<p>Total and active statements display trend lines comparing total number of statements and total number of active statements within a project.</p> <p>Total number of statements that are calculated from total number of executable lines of all program within a project.</p> <p>Active number of statements that are calculated from total number of hit lines of all program within a project.</p>	Workbooks view-> Static Analysis tab -> Overview tab -> detailed project reports
Metrics	<p>The radar chart shows the information of project metrics. By default, the chart displays the following information:</p> <ul style="list-style-type: none"> • Cyclomatic Complexity • Number of source line of code • Maintainability Index • Unreachable Code • Number of estimated delivered bugs <p>You can select up to five projects to display on the radar chart in order to compare metrics values.</p>	Workbooks view-> Static Analysis tab -> Metrics tab

Table 16. Data analysis reports for project data from Application Discovery (continued)

Reports Name	Description	Displayed Location
Shared Resources table	<p>The table shows the list of all data sources that are shared between projects within the workbook. You can find the following types of data sources.</p> <ul style="list-style-type: none"> • Database tables • Dataset • IMS Database 	<p>Workbooks view-> Static Analysis tab -> Structure tab</p>
Shared Resources graph	<p>The network graph shows the relationship between projects and their data sources. Nodes on the right represent projects within the workbook and nodes on the left represent data sources that are used by projects. Direction of the relationship represents "read" or "write" activity.</p>	<p>Workbooks view-> Static Analysis tab -> Structure tab</p>
Shared Resources bar chart	<p>The stacked bar chart shows the number of shared resources by resource type.</p>	<p>Workbooks view-> Static Analysis tab -> Structure tab</p>
Trends	<p>The trend lines show the historical data of project metrics. By default, the chart displays the following information.</p> <ul style="list-style-type: none"> • Cyclomatic Complexity • Number of source line of code • Unreachable Code <p>You can select up to five projects to display on a trends line to compare the metrics values.</p>	<p>Workbooks view-> Static Analysis tab -> Trends tab</p>
Artifact Composition graph	<p>Artifact Composition graph is the connected graph showing all artifacts (for example, program and database table) within a transaction. The boxes represent artifacts within a transaction. The arrow lines represent the connection between two artifacts. The arrow direction indicates the calling direction between artifacts.</p>	<p>Artifact Composition view</p>
Artifact Composition table	<p>Artifact Composition table shows the list of all program artifacts within a transaction. The table is sorted based on the analysis of high risk to modify. The rank of risk to modify is calculated based on combination of the following criteria:</p> <ul style="list-style-type: none"> • Whether the artifact is in risk areas: Risk areas are based on the thresholds setting. • Whether the artifact is an outlier: Outlier artifacts are analyzed based on the outlier detection algorithm leveraging Mahalanobis distance of each artifact to the center of the distribution. Artifacts with large Mahalanobis distances mean they are outlier or abnormal because they very far away from majorities. • Metrics values sorting: Given the same risk area and outlier status, artifacts are finally sorted by the values of Maintainability Index, Unreachable Code, and Cyclomatic Complexity. 	<p>Artifact Composition view</p>

Table 17. Data analysis information for project data from Application Discovery

View	Information	Description
Shared Resources table	Resource Name	The name of the data sources that are shared between projects in a workbook.
	Resource Type	The type of the data source. Three types of data sources are available: <ul style="list-style-type: none"> • DB2 tables • Files • IMS segments
	Number of Projects	The number of projects that either "read" or "write" data from the resource.
Artifact Composition table	Outlier Status	Outlier Status indicates whether the artifact is an outlier. Outlier artifacts are analyzed based on the outlier detection algorithm leveraging Mahalanobis distance of each artifact to the center of the distribution. Artifacts with large Mahalanobis distances mean they are outlier or abnormal because they are far away from majorities.
	Artifact Type	The program type of the artifact.
	Program Name	The name of the artifact.
	Maintainability Index	See details in Table 1.
	Unreachable Code	See details in Table 1.
	Cyclomatic Complexity	See details in Table 1.
	Number of Incoming References	The number of links calling in the artifact.
	Number of Outgoing References	The number of links calling out of the artifact.

Analyzing static metrics for a workbook

ADDI Extensions provides the static metrics and reports such as project inventory, project complexity, and project quality for all projects within the scope of a workbook.

Before you begin, you need to add at least one workbook that is associated with the Application Discovery data provider. For more information, see [“Adding a workbook” on page 217](#). Complete the following steps to view the static analysis of projects within a workbook:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view the static analysis information about.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab. The Static Analysis reports of a workbook are displayed by tabs.

4. Select one of the following tabs to view information. For detailed report information, see [“Project data from Application Discovery reports and information”](#) on page 268.

- **Overview**
- **Metric**
- **Structure**
- **Trends**

5. Select up to five projects from the list of projects within the workbook on the left pane to analyze those projects. Static analysis reports for all tabs are updated based on the project selection.

6. Optional: On the **Overview** tab, **Metrics** tab, and **Trends** tab, choose different static metric to be displayed on the reports from the metric drop-down list on the left pane. You can find the following metrics.

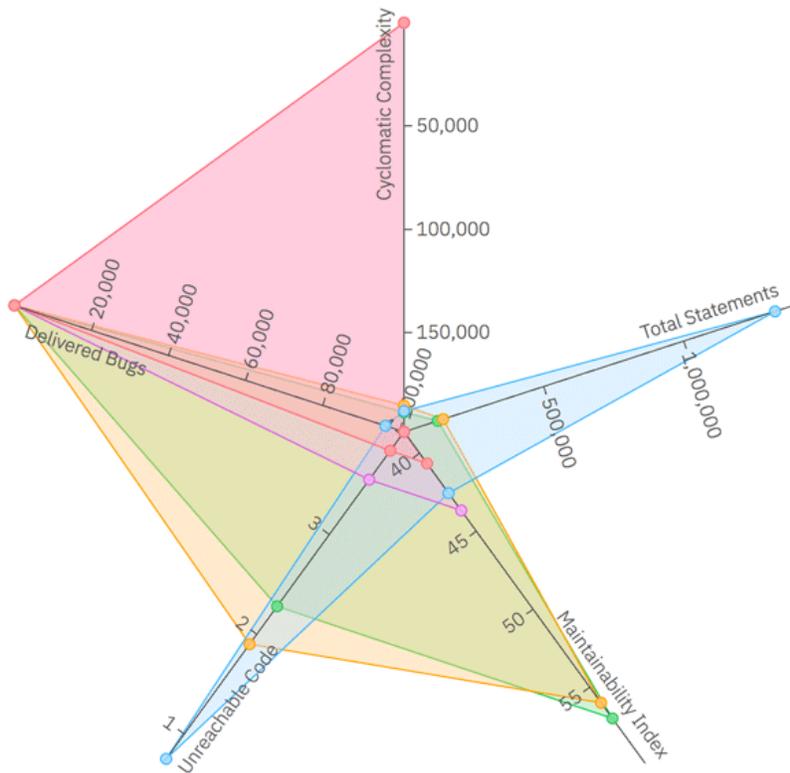
- Active Statements
- Artifacts
- Cyclomatic Complexity
- Delivered Bugs
- Maintainability Index
- Source Line of Code
- Unreachable Code

When you choose different metrics, the report on the corresponding tab is updated based on your metric selection.

Note: The higher value of a metric or the lower value of a metric can be good or bad depending on the metric itself. ADDI Extensions calls the rising or declining trend of metric value change as the metric direction. An increasing direction means that the higher value is better than the lower value. A decreasing direction means that the lower value is better than the higher value. For example, Active Statements metric has an increasing direction, which means that a higher number of Active Statements in the project indicates the better condition. While Unreachable Code metric has a decreasing direction, hence a lower value of unreachable code indicates a better condition. Some of the inventory metrics, such as the number of Artifacts or the number of Source Lines of Code, the metric directions might not indicate the condition of that metric. The following table shows the direction of all a available AD metrics.

<i>Table 18. AD metrics and their directions</i>	
Metrics	Direction
Active Statements	Increasing
Artifacts	No direction
Cyclomatic Complexity	Decreasing
Delivered Bugs	Decreasing
Maintainability Index	Increasing
Source Lines of Code	No direction
Unreachable Code	Decreasing

When the metrics are selected as axes on the bubble chart or the radar chart, the value of the axes are displayed in an increasing or decreasing trend depending on the metric direction. For the inventory metrics that have no direction, the value of the axes is increasing. You can check the following visualized metric directions for reference.



nt All Match Case Whole Words 2 of 12 matches

Analyzing static metrics for a project

For projects within the scope of a workbook, you can view project level metrics for a project such as project inventory, complexity, and quality.

Complete the following steps to analyze the static metrics for a project:

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view the static analysis information about.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab. The Static Analysis reports of a workbook that are organized in tabs display. You can find the following tabs.
 - **Overview**
 - **Metric**
 - **Structure**
 - **Trends**
4. On the **Overview** tab, select a bubble that represents a project of which you want to view the static analysis information on the bubble chart. The detailed project reports are displayed on the bottom with the following information that is calculated from the latest collected data from Application Discovery data provider. For detailed report information, see [“Project data from Application Discovery reports and information”](#) on page 268.
 - Artifact and lines of code breakdown
 - Maintainability index versus Delivered bugs
 - Unreachable code trend
 - Total and active statements trend

Next you can perform the composition analysis of artifacts within a project by navigating to the Artifact Composition function. For more information, see [“Analyzing the Artifact Composition” on page 274](#).

Analyzing shared resources

Data sources such as database table and data set can be shared among multiple projects. ADDI Extensions helps you to understand the data sources that are shared by multiple projects within the scope of a workbook.

Complete the following steps to analyze shared resources.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view analysis information about.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab.
4. Select the **Structure** tab under the **Static Analysis** tab. The shared resources reports and information are displayed. For more information, see [“Project data from Application Discovery reports and information” on page 268](#).
5. Select up to five projects from the list of projects within the workbook on the left pane to analyze those projects. The list of all the resources that are used by projects is displayed on the Shared Resources table.
6. Select up to 5 shared resources on the Shared Resources table. The Shared Resources graph and bar chart are updated based on the selected project and shared resources.
7. Optional: Uncheck any of the resource types under Resource Type Filters on the left pane to filter out the shared resources that are displayed on the table and charts.

Analyzing project data

ADDI Extensions provides the detailed project data analysis such as project inventory, project complexity, and quality.

Complete the following steps to view the analysis of project inventory, complexity, and quality. Before you begin this activity, you need to add at least one workbook with the associated Application Discovery data provider. For more information about how to add a workbook associated with Application Discovery, see [“Adding a workbook” on page 217](#).

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view analysis information.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab.
4. The Static Analysis reports of a workbook are displayed with the following information that is calculated from the latest collected data from Application Discovery data provider. For detailed report information, see [“Project data from Application Discovery reports and information” on page 268](#).
 - Artifact and lines of code breakdown
 - Maintainability index versus delivered bugs
 - Unreachable code trend
 - Total and active statements trend

For next step, you can perform the composition analysis of artifacts within a project by navigate to Artifact Composition function. For more information, see [“Analyzing the Artifact Composition” on page 274](#).

Analyzing the Artifact Composition

The transactions within a project can be analyzed using the Artifact Composition. The Artifact Composition allows you to analyze potential risk areas of all the artifacts within a transaction based on

relationships between artifacts and Application Discovery related metrics such as maintainability index, unreachable code, and cyclomatic complexity.

The Artifact Composition is displayed in two views: the Artifact Composition graph and the Artifact Composition table. You can switch between the two views after you navigate to the Artifact Composition function. For information about Artifact Composition graph and Artifact Composition table, see [“Project data from Application Discovery reports and information”](#) on page 268.

Complete the following steps to navigate to analyze the Artifact Composition.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view analysis information about.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab.
4. On the **Overview** tab under the **Static Analysis** tab, select one of the bubble that represents the project to be analyzed. The detailed project reports appear on the bottom.
5. Select the name of that project. The page to select transaction for analysis appears.
6. Select the transaction you want to analyze. The Artifact Composition map appears. You can perform multiple actions to help analyze the Artifact Composition. For more information, see [“Analyzing transactions with the Artifact Composition graph”](#) on page 275.
7. Select the **Table** icon () on the top right of the view to switch to Artifact Composition table.
8. The Artifact Composition table appears. You can perform multiple actions to help analyze the Artifact Composition. For more information, see [“Analyzing transactions with the Artifact Composition table”](#) on page 277.
9. Review the top-ranked artifacts on the table first as they are the potentially riskier ones to modify. For more information, see [“Reviewing the top-ranked artifacts”](#) on page 278.
10. Select the **Graph** icon () on the top right of the view to switch back to the Artifact Composition graph.

Note: You can navigate to the Artifact Composition view from the **Structure** tab by clicking the name of shared resources you are interested in. The list of all transactions that use the selected shared resources appears. Select one of the transaction to go to the Artifact Composition view and analyze as described above.

Analyzing transactions with the Artifact Composition graph

You can perform the following actions to help analyze the transactions via the Artifact Composition graph.

Risk Areas analysis

Risk areas are the indications of potential artifacts that could be risky. Risk areas are indicated by either blue warning icons or red warning icons. A blue warning icon indicates the artifacts that have Application Discovery metrics changes that can indicate potential source changes in the past week. A red icon indicates the value of Maintainability Index, Unreachable Code, or Cyclomatic Complexity is above or below the related threshold setting.

Complete the following steps to make a Risk Areas analysis.

1. Click the **Risk Areas** icon () on the top header. The dialog box that explains the risk areas and threshold setting opens.
2. Blue warning icons and red warning icons appear on the artifact boxes.
3. Slide the threshold sliders to adjust the threshold values.
4. The warning icons update based on the adjusted threshold values.
5. Click **X** on the upper right of the dialog box to close the Risk Areas dialog box.

Viewing program metrics

You can view all the metrics that are related to each program by clicking the artifact box. The selected artifact is highlighted and the dialog box with program metrics with the following information opens.

- The type of program and the program name
- Metrics
 - Maintainability Index
 - Unreachable Code
 - Active Statements
 - Source Lines of Code
 - Cyclomatic Complexity
 - Delivered Bugs
 - Total Statements

Note: You can select **View Trends** next to the metrics header to view historical trends of the above metrics. By default, the view displays Maintainability Index and Source Lines of Code. You can select the metric name to change to different metric.

- Program Reference Information
 - Number of incoming references
 - Number of outgoing references

Zooming in or out of the Artifact Composition Map

You can zoom in or zoom out on the artifact composition map by using the + icon, - icon and the slider bars on the lower left corner of the screen.

Filtering artifacts

The artifacts can be filtered based on two criteria: filter artifacts that are connected to DB2 and filter artifacts that are connected to File I/O.

Complete the following steps to filter artifacts.

1. On the top header, click **Filters** to open the drop-down list.
2. On the drop-down list, check either of the following choices or both.
 - Select DB2 to display only the artifacts, which are connected to DB2.
 - Select FILE I/O to display only the artifacts, which are connected to File I/O.

3. Click the **Expand All** icon () on the top header to clear all the filters.

Navigating through different projects and regions

A transaction can be associated with other projects and regions. You can view the Artifact Composition graph of these other projects and regions by selecting a project or a region from the **Project** drop-down list or the **Region** drop-down list on the top header.

Navigating through different transactions

There can be other transactions within a project and a region. You can view the Artifact Composition graph of other transactions by selecting a transaction from the **Transaction** drop-down list on the top header.

Highlighting a program

You can highlight a specific program in the artifact composition graph to display the relationship with other artifacts by selecting a program name from the **Select a Program** drop-down list on the top header.

Changing snapshot date

By default, the Artifact Composition graph is displayed based on the latest data that is collected from the Application Discovery server. You can view the Transaction Composition graph from different snapshots by selecting a different date from the **Date** drop-down list on the top header.

Highlighting artifacts with threshold setting

Select a metric from the **Select a Heat Map** drop-down list on the top header to display a color coded artifact composition graph based on the metric threshold values. Accordingly, the artifacts with the good, acceptable, and poor threshold values are highlighted with blue, yellow, and red color.

Analyzing transactions with the Artifact Composition table

You can perform the following actions to help analyze transactions via the Artifact Composition table.

Viewing metric trends

Select the program name of which you want to view historical trends of the static analysis metrics that is related to the program. By default, the view displays Maintainability Index and Source Lines of Code. You can select the metric name to change to different metric.

Risk Areas analysis

Risk areas are the indications of artifacts that could be potentially risky. Risk areas are indicated by either the blue warning icons or the red warning icons. A blue warning icon indicates the artifacts that have Application Discovery metrics changes that can indicate potential source changes in the past week. A red icon indicates the value of Maintainability Index, Unreachable Code, or Cyclomatic Complexity is above or below the related threshold setting.

Complete the following steps to make Risk Areas analysis.

1. Click the **Risk Areas** icon () on the top header. The dialogue box explaining the risk areas and threshold setting appears.
2. Slide the threshold sliders to adjust the threshold values.
3. The warning icons update based on the adjusted threshold values.
4. Click **X** on the top right of the dialogue box to close the **Risk Areas** dialogue box.

Changing snapshot date

By default the Artifact Composition graph is displayed based on the latest data collected from Application Discovery server. You can view the Transaction Composition table from different snapshots by selecting a different date from the **Date** dropdown list on the top header.

Navigating through other projects and regions

A transaction may be associated with other projects and regions. You can view the Artifact Composition table of these other projects and regions by selecting a project or a region from the **Project** dropdown list or the **Region** dropdown list on the top header.

Navigating through other transactions

There may be other transactions within a project and a region. You can view the Artifact Composition table of other transactions by selecting a transaction from the **Transaction** dropdown list on the top header.

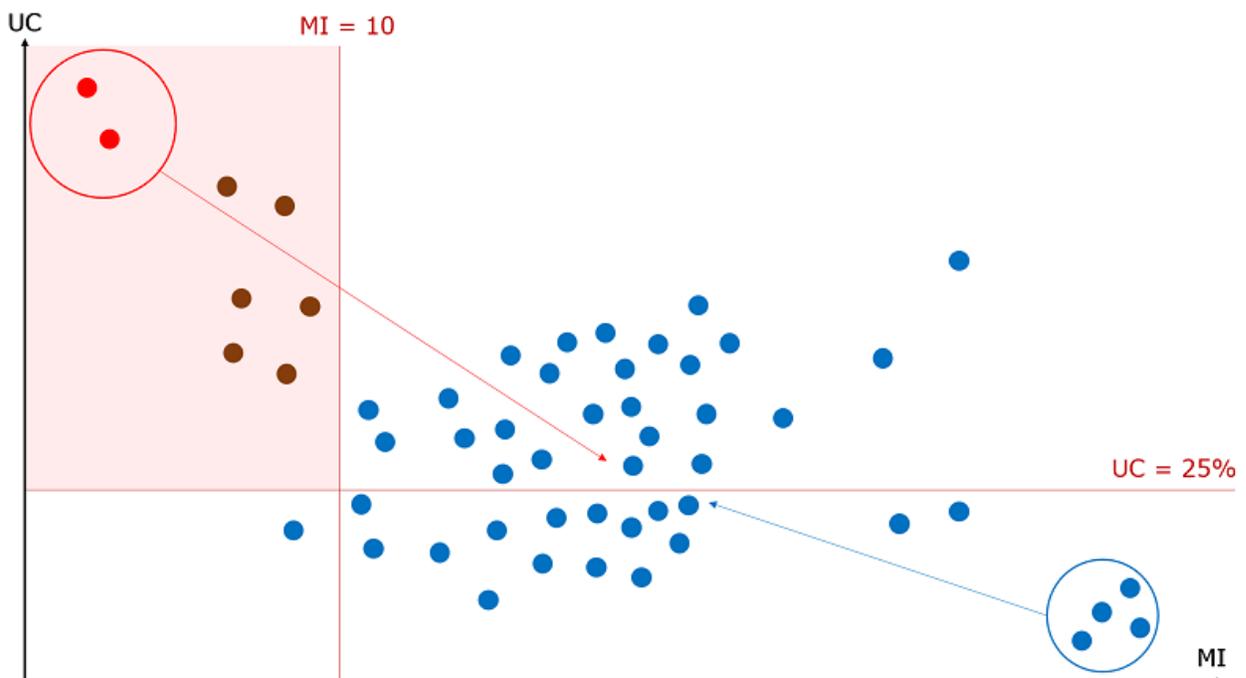
Reviewing the top-ranked artifacts

It is recommended that you review the top-ranked artifacts on the table for analysis. This top-ranked artifacts indicate the higher risk if there are changes to these artifacts.

The top-ranked artifacts are ranked comprehensively based on three factors:

1. Risk areas analysis.
2. Outlier analysis status.
3. Static metrics sorting

The basic idea of this ranking algorithm can be illustrated in the following figure. For simplicity, a two-dimensional graph is shown with the x-axis represents Maintainability Index (MI) and the y-axis represents Unreachable Code (UC). Each solid circle on the graph represents a program artifact with two metrics: MI and UC.



- The red rectangle shown on the graph is a risk area. It is defined by a horizontal line ($UC > 25\%$) and a vertical line ($MI < 10$), artifacts within the risk area (color labeled as red) are heuristically riskier than those outside the risk area (color labeled as blue). As such, checking whether a program artifact is in the risk area is the first factor to calculate the ranking.
- From the graph, you can also see that there are two red artifacts and four blue artifacts having much larger distances from the center compared to the others. Such distance is measured by Mahalanobis Distance, which is a commonly-used distance measurement between a data point and a distribution, introduced by P. C. Mahalanobis in 1936. These two artifacts can be considered as bad outliers as they are within risk area and far away from the others. The four blue artifacts can be considered as good outliers as they are outside the risk area and far away from the others. Thus, checking whether an artifact is a bad outlier is the second factor in calculating the rank.

Note: ADDI Extensions may generate false positives, that is, marking good outliers as bad outliers. It usually happens when you defined improper metrics threshold settings for risk area analysis. It is recommended that you use the default threshold settings or ask domain experts for advice.

- The last ranking factor is metrics sorting. ADDI Extensions sorts artifacts by their metric values following this order: Maintainability Index, Unreachable Code, Cyclomatic Complexity, Number Of Incoming References, and Number Of Outgoing References.

Comparing the projects within a workbook

ADDI Extensions allows you to compare up to 5 projects within a workbook. You can use the radar chart to compare the recent values of project metrics or use the trend line to compare the historical data of project metrics.

Complete the following steps to compare projects within a workbook.

1. Select the **Workbooks** tab on the header to go to the **Workbooks** page.
2. Select the name of the workbook that you want to view analysis information about.
3. On the summary view of that workbook, select the **Static Analysis** tab if you are not yet on the tab.
4. Select up to five projects from the list of projects within the workbook on the left pane to compare.
5. To compare the recent values of project metrics, select **Metrics** tab under the **Static Analysis** tab.
 - a. The radar chart is displayed with the recent values of Cyclomatic Complexity, Total Statement, Maintainability Index, Unreachable Code, and Delivered Bugs as default.
 - b. On the left pane, select different metrics to be displayed for each axis from the metric dropdown list. The radar chart is updated based on the metric you select.
6. To compare the historical data of project metrics, select **Trends** tab under **Static Analysis** tab.
 - a. Three trend charts showing historical value of Unreachable Code, Cyclometric Complexity and Source Line of Code are displayed on the **Trends** tab.
 - b. On the left pane, select different metrics to be displayed for each trend chart. The trend chart is updated based on the metric you select.
 - c. Optional: From the Project Aggregation drop-down list, select the aggregation calculation you want among the following three aggregation calculations available:
 - Metric min value: Calculate the minimum value of the metric for all the selected projects.
 - Metric max value: Calculate the maximum value of the metric for all the selected projects.
 - Metric average value: Calculate the average value of the metric for all the selected projects.

Chapter 12. Troubleshooting

Troubleshooting helps you identify and solve problems that occur while using ADDI Extensions.

Empty table displayed after deleting all the items within the last page of table view

When you display the last page of business terms or business rule packages and delete all the items within that page, this will cause the view to display an empty table. You can refresh the page to view the items that are not deleted (from the previous pages).

Information errors for adi-setup help document for non-English version

When you perform adi-setup with the help option to review adi-setup documentation for a non-English version, it displays internal parameters that are not supported by the tool. In addition, the description of the new option **bcryptPassword** is missing from the non-English command line help but this option is supported. You can refer to [adi-setup references](#) for all the available options.

Requirements for a self-signed certificate and browser configurations for the application server shipping with ADDI

After the installation, you need to configure a certificate authority that is a signed or self-signed certificate for your application server. You need to send instructions to your end users about how to import this certificate to avoid warnings or errors in their browsers.

- For instructions about installing a certificate into Liberty, see [“Installing a security certificate into Liberty” on page 163](#).
- For instructions about enabling SSL communication in Liberty, see [Enabling SSL communication in Liberty](#).

Note: SSL keystore and certificates are included in the Authentication Server (DEX) but only for evaluation purposes. You need to replace the pre-packaged SSL keystore and certificate with your own SSL keystore and certificate for your production environment.

SQL error (SQLCODE = -964) on massive data update transactions for DB2

When you perform massive DB2 update transactions, for example, deleting a data provider with 200,000 entries, you might observe SQL error with error code -964. The possible cause for the error is that your DB2 log setting is wrong or too restrictive. To solve this issue, you need to increase the size of your transaction logs. You can refer to the following sample statements to increase log size, supposing the name of your DB2 data warehouse is "DW".

- db2 get db cfg for DW
- db2 update db cfg for DW using LOGPRIMARY 64
- db2 update db cfg for DW using LOGSECOND 192

Note: The total number of primary logs plus secondary logs cannot exceed 256. It is recommended to set a larger secondary log size since they are constantly cleared by DB2.

Inadequate memory to run ADDI

If you're installing ADDI Extensions component together with other Application Discovery components with limited memory, you can reduce memory allocation to ADDI Extensions by updating the memory allocation parameters in the startup script located at `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/server.startup.bat` as follows:

```
set JAVA_OPTS=%JAVA_OPTS% -Xmx2G
set JAVA_OPTS=%JAVA_OPTS% -Xms2G
set JAVA_OPTS=%JAVA_OPTS% -Xmn1G
```

You can also consider reducing the memory allocation for elasticsearch in `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/config/jvm.options` as follows:

```
-Xms1g
-Xmx1g
```

However, it is recommended to reduce memory allocation parameters only in the evaluation environment. On the production server, the default memory allocations are required for ADDI Extensions to work properly.

For more information, see [“Hardware and software requirements” on page 139](#).

Application log files

Application logs contain important information about the ADDI Extensions server applications. Consult the application logs when you encounter errors in the application. If there is an error ID associated with the error message, the administrator can search for that ID in the application logs to ensure the appropriate message is being investigated. IBM Support may request log levels be adjusted for ADDI Extensions.

Perform the following steps to adjust the log levels.

1. Browse to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/conf/adi` and edit the `log4j.properties` file.
2. Go to the line 17 in the `log4j.properties` and switch the log level from WARN to INFO.

After starting up IBM ADDI Extensions, the shortcut to a folder containing log files is generated and located at `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server`.

server.startup.bat script hangs at starting the Elasticsearch server

When you start ADDI Extensions server for the first time on Windows and do not have the `JAVA_HOME` variable defined in the system environment, you will find that the `server.startup.bat` script hangs at starting the Elasticsearch server. You can complete the following steps to fix it:

1. Go to **Control Panel > View advanced system settings > Environment Variables**.
2. Click **New** to define a system variable with the name "JAVA_HOME". The value can be set to JDK8 or a later version.
3. Open a command prompt and navigate to the `<Installed ADDI Extensions folder>/adi611/server` directory by running the following command.

```
cd <Installed ADDI Extensions folder>/adi611/server
```

4. In the current directory, start the server by running the following command.

```
server.startup
```

You can see that the Elasticsearch server can be started now.

5. Shut down the server by running the following command.

```
server.shutdown
```

6. Refer to step 1 and step 2 to undo your operation of defining the `JAVA_HOME` variable.
7. Start the server again by running the following command.

```
server.startup
```

Starting ADDI Extensions Server for the first time hangs at starting Elasticsearch

When you start ADDI Extensions server for the first time after the installation and setup, you might notice that the script hangs at starting Elasticsearch and no error is reported. The cause is that Elasticsearch is unable to write the relative path 'logs/gc.log'.

To resolve the issue, complete the following steps:

1. Create a folder named "logs" on a file path without empty space, and make sure that the user to launch the ADDI Extensions server has full permissions to this folder.
2. Open
<ADDI_Extensions_installation_directory>\elasticsearch\config\jvm.options in an editor.
3. Find all occurrences of "logs".
Note: In V6.0.2 and later versions, you can find three relative paths, for example, logs/gc.log.
4. Replace those relative paths with a full path that you created in step 1, for example, replace logs/gc.log with c:/<a_file_path_without_empty_space>/logs/gc.log.
5. Save your changes.
6. Restart the server. If possible, restart your machine instead of restarting the server.

Verifying the Elasticsearch server is running

ADDI Extensions integrates Elasticsearch within it and requires that the Elasticsearch server is always up and running in the background. The Elasticsearch server is started by default when you start ADDI Extensions application. You can easily verify that the Elasticsearch server is running using one of the following two methods.

- Open your browser and navigate to `http://localhost:9200/?pretty`.
- Open your terminal/command prompt and run the command **curl http://localhost:9200/?pretty**.

If a JSON response is returned successfully, you can confirm that Elasticsearch server is up and running.

If you don't see a JSON response that is returned, you can conclude that the Elasticsearch server is not running. You need to investigate further to resolve this issue.

Note: For more information about what went wrong, refer to the Elasticsearch log files.

Elasticsearch log files

Elasticsearch log files contain important information about the Elasticsearch server that can help you troubleshoot errors pertaining to running/starting the Elasticsearch server.

You can find the Elasticsearch log files in the <addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/logs folder.

Elasticsearch failing to start up

When you start ADDI Extensions, you might notice an error message in the console stating that Elasticsearch failed to start. This error message might be different for different operating systems. Check the following tips for each operating system on how to resolve these errors.

Windows

On a Windows operating system, Elasticsearch is run as a background service. During ADDI Extensions startup, the "adi-elasticsearch" Windows service is installed and started. If you notice an error in the command prompt, such as "Failed to install 'adi-elasticsearch' service" or "Failed to start 'adi-elasticsearch' service", then it means that something went wrong while you were installing or starting the "adi-elasticsearch" service.

To resolve this error, complete the following steps.

1. Check the installed path. Elasticsearch cannot be installed in a path that contains parentheses.
2. Verify if the Elasticsearch server is running or not. You can perform this task by following the steps provided in the previous section of *Verifying the Elasticsearch server is running*.
3. If the Elasticsearch server is not running, review the Elasticsearch log files to understand what went wrong while you were installing or starting up the Elasticsearch service. You can find the Elasticsearch log files in the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/logs` folder.
4. Fix the errors reported in the log files.
5. Stop and restart ADDI Extensions application server.
6. Verify if the Elasticsearch service is installed and started successfully this time. To confirm it, verify that you see The service 'adi-elasticsearch' has been installed. and The service 'adi-elasticsearch' has been started. messages in the command prompt or complete the steps as described in the *Verifying the Elasticsearch server is running* section.
7. If you are unable to understand the nature of the problem, complete the steps described in the following *Workaround* section.

Linux

On a Linux operating system, Elasticsearch is run in the background as a daemon process and it is started during ADDI Extensions startup. If you notice any errors related to Elasticsearch in the terminal during startup, then it might indicate that something went wrong when you start the Elasticsearch.

To resolve this error, complete the following steps.

1. Check the installed path. Elasticsearch cannot be installed in a path that contains parentheses.
2. Verify if the Elasticsearch server is running or not. You can perform this task by following the steps provided in the previous section of *Verifying the Elasticsearch server is running*.
3. If the Elasticsearch server is not running, review the Elasticsearch log files to understand what went wrong while you were installing or starting up the Elasticsearch service. You can find the Elasticsearch log files in the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/elasticsearch/logs` folder.
4. Fix the errors reported in the log files.
5. Stop and restart ADDI Extensions application server.
6. Verify if the Elasticsearch service is installed and started successfully this time. To confirm it, complete the steps as described in the *Verifying the Elasticsearch server is running* section.
7. If you are unable to understand the nature of the problem, complete the steps described in the following *Workaround* section.

Workaround

If you are unable to understand the nature of the problem, contact technical support to get further help. IBM Support might request ADDI Extensions application or Elasticsearch log files for reference. Therefore, you need to preserve a copy of these log files from the session when you encountered these issues.

As a workaround, you can complete the following steps to start the Elasticsearch server manually in a separate terminal/command prompt.

1. Open terminal/command prompt and navigate to the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/logs` by running the following command:

```
cd <addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/logs
```

2. In the current directory, start the Elasticsearch server by running the following command.

```
startes
```

After you complete these steps, you have started the Elasticsearch server successfully and can continue to use the ADDI Extensions application.

Huge Elasticsearch log files

If you are not able to access the Enterprise Artifact data or keywords information and start seeing various 500 internal server error messages, check the size of your Elasticsearch log file. If the size of Elasticsearch log file is a lot larger than usual, complete the following steps:

1. Update the `rootLogger.level` property to be `error` logger level:
 - a. Open the `log4j2.properties` file in the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/config/` folder.
 - b. Update the `rootLogger.level` property as `rootLogger.level = error`.
2. Investigate if any zombie ADDI Extensions Java processes that might lock the Elasticsearch data folder are running.
3. Terminate the zombie ADDI Extensions Java processes if any.

Note: Avoid deleting anything in the `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch/data` folder, unless you want to clean the Elasticsearch server for a fresh restart.

Elasticsearch port conflicts

Elasticsearch Front Server application runs on a separate port with ADDI Extensions server. ADDI Elasticsearch Front Server applications by default would always run on **http:9081** and **https:9444** port and there is no automatic setup available to change this port number. When you change the host and port for ADDI Extensions, the changes are only applied to ADDI Business Rule Discovery, Metrics and Metadata applications but not Elasticsearch Front Server. You can manually change the port by completing the following steps. However, it is not recommended.

1. Navigate to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/liberty/wlp/usr/servers/adiServer/server.xml`.
2. Open `server.xml` and find the **httpEndpoint** element with `id="localhostOnly"`.
3. Within this **httpEndpoint** element update the property **httpsPort="9444"** to the appropriate port number you want to use for Elasticsearch Front Server application (for example, **httpsPort="9999"**).
4. (Optional) In the `server.xml`, find the element called **<virtualHost id="addi-web-es-host" . . .** that contains the child element **<hostAlias>localhost:9444</hostAlias>** and change the **9444** to the new Elasticsearch port, for example, change the port to **9999** as follows:

```
<hostAlias>localhost:9999</hostAlias>
```

5. Navigate to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/conf/adi/teamserver.properties`.
6. Open the `teamserver.properties` file and find the property `"com.ibm.team.repository.server.esfrontserver.url = https://localhost:9444/addi/es"`.
7. Update this property with Elasticsearch Front Server base url using the correct port number as defined in step 3. For example, `com.ibm.team.repository.server.esfrontserver.url = https://localhost:9999/addi/es`.
8. (Optional) If you are configuring ADDI Extensions for Business Rule Discovery analysis then you will also need to reflect the changed port number under `<addi_installed_directory>/IBM Application Discovery Batch Server/conf/adi.properties` file.

9. Open the `adi.properties` file and update the `es.serverUrl` property with correct port number as defined in step 3. For example, `es.serverUrl=https://localhost:9999/addi/es`.

Problems with running ADDI Extensions application on Linux using port 80 or port 443

On Linux, ports below 1024 can be opened only by `root`. If you are having problems trying to run your ADDI Extensions on Linux using port 80 or port 443 make sure you are running the ADDI Extensions application server as a `root` user.

The Elasticsearch service is started by default during ADDI Extensions server startup. If you are trying to run ADDI Extensions on Linux using port 80 or port 443 as a `root`, the Elasticsearch server startup will fail because Elasticsearch cannot be run as a `root` user. In this case, you need to complete the following steps to make a non-root user as the owner of the Elasticsearch directory and start it manually in a separate terminal.

1. Open a terminal and run the following command to navigate to `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch`:

```
cd <addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/elasticsearch
```

2. In the current directory, run the following command to change the ownership to a non-root user.

```
chown -R non-root user
```

3. Run the following command to switch to the non-root user.

```
su non-root user
```

4. Run the following command to start the Elasticsearch server.

```
startes
```

After you complete the previous steps, your ADDI Extensions will run as a `root` user while your Elasticsearch server will run separately as a non-root user.

Shutdown script is hanging

If the shutdown script (`server.shutdown`) hangs and never finishes, you may need to manually kill the linked java process (location `<addi_installed_directory>/IBM Application Discovery and Delivery Intelligence Extensions/adi611/server/jre/bin`) to force a shutdown of the server.

Known issues of ADDI Extensions

1. When you run ADDI installer on a Windows machine, Windows doesn't recognize the application and prevents your machine from running it.

Solution: Complete the following steps to run the ADDI installer.

- a. Select **More info** link from the **Window protected your PC** dialog box.
- b. Select **Run anyway** button from the next screen.

2. Uninstalling ADDI does not remove the IBM Application Discovery and Delivery Intelligence Extensions folder.

Solution: Simply remove the IBM Application Discovery and Delivery Intelligence Extensions folder or the `<addi_installed_directory>` manually.

3. The default ADDI Extensions installation path contains both forward slashes and backward slashes, which might cause issues when you install ADDI on a Linux machine.

Solution: Manually update the default path from the installation wizard.

4. While editing a Business Term and a Business Rule Package, you might not be able to update status and tag fields at the same time.

Solution: Either switch to Chrome or update status or tag fields separately.

5. When you run ADDI Extensions for awhile, the Authentication Service (DEX) might stop working unexpectedly and you will not be able to log into ADDI Extensions.

Solution: Restart the Authentication Service (DEX) to resolve the issue.

Getting errors when you connect to RTC server

You might get the following error message when you test the RTC connection. To fix the error, you need to restart the ADDI Extensions server.

```
CRIDA0632E A request to an external RTC server returned with an error or invalid data. Check your connection details or the external server.
```

Chapter 13. Providing feedback

You can provide feedback by opening a Service Request or sending a Request For Enhancement (RFE).

Opening a Service Request

In the case that you have problems or issues and need ADDI Extensions support, you can open a Service Request. You need to have an IBM ID and password to open a Service Request. You can register your IBM ID at [IBM site](#).

For more information about how to submit a Service Request, see [IBM Support: How To Submit a Service Request](#).

Sending a Request for Enhancement (RFE)

In the case that you want to request or suggest an enhancement of ADDI Extensions capabilities, you can submit a Request for Enhancement. You need to have an IBM ID and password to open a request. You can register IBM ID at [IBM site](#).

You can submit a Request for Enhancement (RFE) for ADDI at [ADDI Servers and Systems Software RFE Community](#) on IBM site.

Chapter 14. Accessibility features

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

ADDI Extensions supports the following accessibility features which complied with W3C Web Content Accessibility Guidelines.

- Keyboard navigation using Tab and Return
- Interfaces that are commonly used by screen readers

Related accessibility information

In addition to standard IBM help desk and support website, IBM has established a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

TTY service
800-IBM-3383 (800-426-3383)
(within North America)

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see [IBM Accessibility \(www.ibm.com/able\)](http://www.ibm.com/able).

Chapter 15. Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Index

A

abbreviations [3](#)
accessibility [291](#)
accessibility features for this product [291](#)
add a build [202](#)
add a connection [197](#)
Add a Manual Builds data providers [200](#), [201](#)
add a Rational Team Concert Builds data provider [208](#)
ADDI Extensions code coverage generator [185](#), [186](#),
[190–192](#), [194](#), [195](#)
ADDI Extensions code coverage on server generator [192](#),
[194](#), [195](#)
ADDI installer [141](#)
Adding a Manual Builds data provider [83](#)
Adding a new build [77](#)
Adding a new build as a test manager [77](#)
Adding a workbook [217](#)
adi-setup script [162](#)
adi-setup script references [162](#)
analysis results of the first build [74](#)
Analysis workbook [71](#), [221](#), [223](#), [232–236](#)
analyze Artifact Composition [274](#), [275](#), [277–279](#)
analyze project data [274](#)
analyze shared resources [274](#)
analyze static metrics [271](#), [273](#)
analyze static metrics for a project [271](#)
Analyzing Analysis workbook [221](#), [223](#), [232–236](#)
Analyzing operational data [237](#), [238](#), [242](#)
Analyzing pinned files [233](#)
Analyzing project data [266](#), [268](#), [271](#), [273–275](#), [277–279](#)
Apache Ant build scripts [185](#), [186](#), [190](#)
Application Discovery data providers [214](#)
Artifact Composition [274](#), [275](#), [277–279](#)
Artifact Composition graph [275](#)
Artifact Composition table [277](#)
Authentication Service (DEX) [154](#)
automated code coverage data collection [83](#)

B

back up data [163](#)
baseline builds [233](#)
BRD Rest APIs [163](#)
Build Analysis view [234](#)
build.properties [186](#)
build.xml [186](#)
Business Rule Discovery workbook [246](#), [257](#), [259](#)

C

code coverage analysis [79](#), [91](#)
Code Coverage Analysis [67](#)
code coverage data collections [83](#)
code coverage of level break-down
 flowpoint [236](#)
code coverage reports [97](#)

Code coverage reports and information [223](#)
code coverage results [167](#), [170–173](#)
code coverage results for batch applications [167](#)
code coverage results for CICS transactions [171](#)
code coverage results for COBOL and PL/I [167](#)
code coverage results for Java [178–180](#), [185](#), [186](#),
[190–192](#), [194](#), [195](#)
code coverage trends [232](#)
collect code coverage data [83](#)
collect code coverage data automatically [83](#)
collect data manually [200](#), [201](#)
collecting data automatically [202](#)
collecting data manually [202](#)
compare projects [279](#)
Comparing code coverage results [233](#)
Configure Authentication Server (DEX) [159](#)
configuring user groups [160](#)
connection [197](#)
connections [198](#)
coverage reports and dashboards [97](#)
coverage results [167](#), [178–180](#), [185](#), [186](#), [190–192](#), [194](#),
[195](#)
coverage results for COBOL and PL/I [172](#), [173](#)
Create a Business Rule Discovery data provider [213](#)
create an Application Discovery data provider [213](#)
create database [142](#)
Creating a connection [197](#)
Creating a Manual Builds data provider [68](#)

D

data automatically, collect code coverage [83](#)
data collections [83](#)
data from Application Discovery [266](#), [268](#), [271](#), [273–275](#),
[277–279](#)
data from IBM OMEGAMON for CICS [237](#), [238](#), [242](#)
data providers [199–210](#), [214](#)
data source for code coverage results [68](#)
data sources [167](#)
db2 database [142](#)
debug [281](#)
define Analysis workbook [71](#)
define baseline builds [204](#), [210](#)
define snippets [253](#)
Defining a new application [71](#)
delete a build [204](#), [210](#)
delete a business term [259](#)
delete a connection [198](#)
delete a workbook [220](#)
Deleting a connection [198](#)
display code coverage results [221](#)
displaying operational data [237](#), [238](#), [242](#)
displaying project data [266](#), [268](#), [271](#), [273–275](#), [277–279](#)
download code coverage data [205](#), [210](#)

E

edit a business term [257](#)
edit data provider information [202](#), [209](#), [214](#)
Editing the connection information [197](#)
End-to-end [116](#)
End-to-end system performance root cause analysis [116](#)
Exercising setting up a Manual Builds for code coverage analysis [91](#)
external data sources [167](#)

F

feedback [289](#)
Filtering data [234](#)
Filtering data in the Build Analysis view [234](#)

G

generate static analysis sample data [165](#)
Generating code coverage results [167](#), [170](#)
Generating code coverage results by using the IDz client [167](#)
Generating OMEGAMON for CICS data [164](#)
Generating project metadata [194](#), [195](#)
Generating sample data [164](#)
Generating sample data for evaluation [164](#)

H

hardware and software requirements [139](#)
hardware requirements [139](#)

I

IDz client [167](#)
Install a security certificate into Liberty [163](#)
Install and set up IBM ADDI Extensions [5](#)
Installation [139](#)
Installation and setup [139](#)
Installing ADDI Extensions with the ADDI installer [141](#)

L

legal statement [293](#)

M

Managing a connection [197](#)
Managing Business Rule Discovery workbook [246](#), [257](#), [259](#)
Managing connection [197](#)
Managing connections [198](#)
Managing data providers [199–210](#), [214](#)
Managing workbooks [217–220](#)
Manual Builds data provider [83](#), [91](#)
Manual Builds data providers [199–205](#)
Manual Builds for code coverage analysis [91](#)
migration [145](#)
Modify a
 Manual Builds data providers [202–205](#)
Modify a Rational Team Concert Builds data provider [209](#), [210](#)
modify a workbook [218](#)

modify an Application Discovery data provider [214](#)

N

Notices [293](#)

O

OMEGAMON for CICS data [164](#)
OMEGAMON for CICS data generation [164](#)
OMEGAMON for CICS data providers [199](#), [206](#)
operational data from IBM OMEGAMON for CICS [237](#), [238](#), [242](#)
overview [1](#)

P

parameters in the dex.yaml file [155](#)
performance root cause analysis [116](#)
Performing the code coverage analysis [79](#)
pin a workbook [219](#)
pinned files [233](#)
Preparing code coverage results [167](#)
Preparing code coverage results for batch applications [167](#)
Preparing code coverage results for CICS transactions [171](#)
Preparing code coverage results for COBOL and PL/I [167](#), [172](#), [173](#)
Preparing code coverage results for Java [178–180](#), [185](#), [186](#), [190–192](#), [194](#), [195](#)
Preparing external data sources [167](#)
Product overview [1](#)
project data [274](#)
project data from Application Discovery [266](#), [268](#), [271](#), [273–275](#), [277–279](#)
Providing feedback [289](#)

R

RAD client [178](#)
RAD code coverage solutions [178–180](#), [185](#), [186](#), [190–192](#), [194](#), [195](#)
Rational Team Concert Build and Apache Ant [190–192](#)
Rational Team Concert Builds data providers [199](#), [207–210](#)
refresh code coverage data [209](#)
refresh code coverage results [203](#)
refresh the builds [209](#)
reports and dashboards [97](#)
reports and information [238](#), [246](#), [268](#)
REST API [194](#)
Reviewing code coverage reports and dashboards [97](#)
root cause analysis [116](#)
RTC with RAD Quality Extensions [179](#), [180](#)

S

sample data [164](#)
sample data for evaluation [164](#)
Searching for within a build
 files, packages, or directories [235](#)
Security considerations [140](#)
server configuration [151](#)
server shutdown [151](#)
server startup [151](#)

- [setting up a Manual Builds](#) [91](#)
- [Setting up a Manual Builds data provider](#) [91](#)
- [setting up a Manual Builds for](#) [91](#)
- [Setting up a workbook](#) [95](#)
- [Setting up automated code coverage collections](#) [83](#)
- [Setting up automated code coverage data collections](#) [83](#)
- [Setting up Manual Builds](#) [67](#)
- [Setting up Manual Builds data provider](#) [67](#)
- [Setting up Manual Builds for Code Coverage Analysis](#) [67](#)
- [Setup](#) [139](#)
- [shared resources](#) [274](#)
- [software requirements](#) [139](#)
- [start Authentication Server \(DEX\)](#) [159](#)
- [static metrics](#) [271](#), [273](#)
- [static metrics for a project](#) [273](#)
- [system requirements](#) [139](#)

T

- [technical terms](#) [3](#)
- [Terminology](#) [3](#)
- [Terminology overview](#) [3](#)
- [test manager](#) [77](#)
- [Test to Run](#) [235](#)
- [the first build](#) [74](#)
- [top-ranked artifacts](#) [278](#)
- [transaction data](#) [242](#)
- [Troubleshooting](#) [281](#)
- [tutorial](#) [5](#)
- [Tutorial](#) [67](#), [68](#), [71](#), [74](#), [77](#), [79](#), [83](#), [91](#), [95](#), [97](#), [116](#)

U

- [update an existing build](#) [203](#)
- [upgrade](#) [145](#)
- [using RTC](#) [172](#), [173](#)
- [using the debug tool backend and IDz client](#) [171](#)
- [using the headless collector running on ADDI Extensions server](#) [170](#)

V

- [variables defined in startup scripts](#) [151](#)
- [view builds data](#) [209](#)
- [view data collection logs](#) [214](#)
- [Viewing the code coverage](#) [74](#)
- [Viewing the code coverage analysis results](#) [74](#)

W

- [workbook setup](#) [95](#)
- [workbooks](#) [217](#)–[220](#)



Part Number:
Product Number: 5737-B66

SC28-3122-05



(1P) P/N: