IBM Application Delivery Intelligence for IBM Z
V5.1.0.7

User Guide

IBM

Note

Before using this information and the product it supports, read the information in Chapter 12, "Notices," on page 219.

Contents

Chapter 1. Product overview

From the product overview, you can learn the basic knowledge about IBM Application Delivery Intelligence for IBM Z (IBM ADI), its key benefits, and its new features.

Overview of IBM Application Delivery Intelligence for IBM Z

Throughout the DevOps lifecycle, a very large amount of data can be produced, including code, requirements, designs, test artifacts, and operational information about applications. ADI is a web-based tool that is built on the Jazz® framework to facilitate enterprise DevOps adoption and drive continuous improvement across the lifecycle of product development, delivery, and maintenance.

The vision of ADI is to provide an insight dashboard with actionable recommendations to enterprise by:

- Analyzing and visualizing historical trend information on application, usage, code coverage, complexity, quality, and performance
- 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 before production time
- Making information accessible to stakeholders through a simple and intuitive interface

Key benefits of using ADI

ADI's capabilities allow users to analyze the runtime performance data and the testing details over time. It pro-actively alerts users on early problem detection so that resources can be shifted to the areas that need attention. By working with ADI, the typical users, such as the enterprise architects, test teams, developer teams, and operation teams, can benefit from ADI's functions to improve their DevOps process.

By using ADI code coverage function, a tester 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 ADI code coverage function, a test manager 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, a developer can:

- Be alerted about potential performance problems before the code goes into production.
- 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.

What's new

Starting from ADI V5.1.0.6, the Business Rule Discovery and Static Analysis features are moved off Jazz and hence moved out of ADI. Those features are now jazz-free and integrated into ADDI as the IBM ADDI Extension component. As a result, if you are on a previous version of ADI and want to migrate to ADI V5.1.0.6 or later, the data from IBM Application Discovery data providers and Business Rule Discovery data providers are not available. If you want to continue using both features, you need to migrate the IBM Application Discovery data and Business Rule Discovery data to ADDI Extension. The features for code

coverage analysis and performance analysis including OMEGAMON for CICS data and SMF/APA data are not affected so you still can continue using them.

For more information about IBM ADDI Extension, see IBM ADDI Extension User Guide.

For more information about migration to ADI V5.1.0.6 or later, see "Migration considerations" on page 100.

Chapter 2. Terminology overview

You can find the technical terms and abbreviations that are used in ADI in the terminology overview.

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

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.

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.
>
> Examples:
>
> - Build
> - Performance

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

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.

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

ADI 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 provide the step-by-step instructions with which you can explore the key capabilities of ADI.

ADI also provides tutorial videos that demonstrate how to work with ADI based on the scenarios designed in this Tutorial chapter. You can watch the videos that are included in the IBM Application Delivery Intelligence Tutorial Series playlist on YouTube.

Installing and setting up ADI

This tutorial guides you to install and set up ADI as an administrator.

Installing ADI and configuring JTS to be the foundation of ADI

This lab guides you through the installation of ADI and steps to configure IBM Jazz Team Server (JTS) as the foundation of ADI.

Before you begin the installation, you need to update the host file to understand the public URI you use for the setup. Complete the following steps to update the host file.

1. Go to **Start > All Programs > Accessories** and run Notepad as an administrator.
2. Open the host file to edit from `Windows > System32 > drivers > etc` directory.
3. Update the host file for the localhost as the following example.

> \# localhost name resolution is handled within DNS itself.
>
>   127.0.0.1   localhost
>
>   127.0.0.1   healthcare.example.com
>
> \#   ::1   localhost

4. Save the host file.

Complete the following steps to install ADI.

1. Unzip the ADI installer that you obtained.
2. Browse to the location where you unzipped the ADI installer and navigate to the `\adi\server` directory.
3. Run server.startup.bat in a command window. Wait until the server starts up.
4. Start the Mozilla® Firefox® browser and go to `https://healthcare.example.com:9753/jts/setup`.
5. Optional: Allow the browser to connect to the website when a message warning that your connection is not secure appears.

   a. Click **Add Exception**.
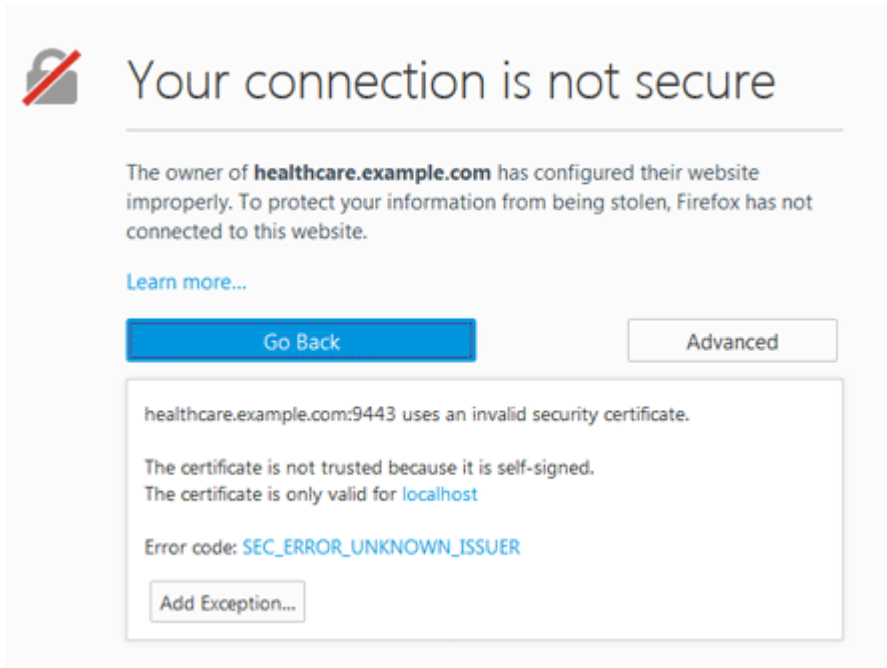
b. On the **Add Security Exception** pop-up window that appears, click **Confirm Security Exception**.
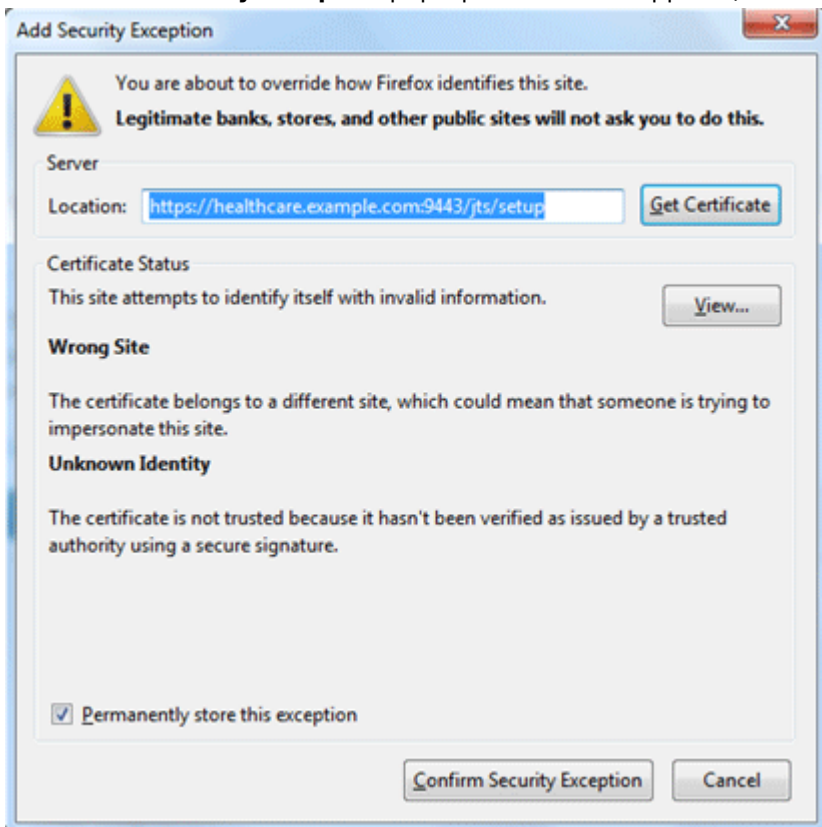
6. On the Jazz Team Server login page that appears, log in with ADMIN as the user ID and ADMIN as the password (case sensitive).
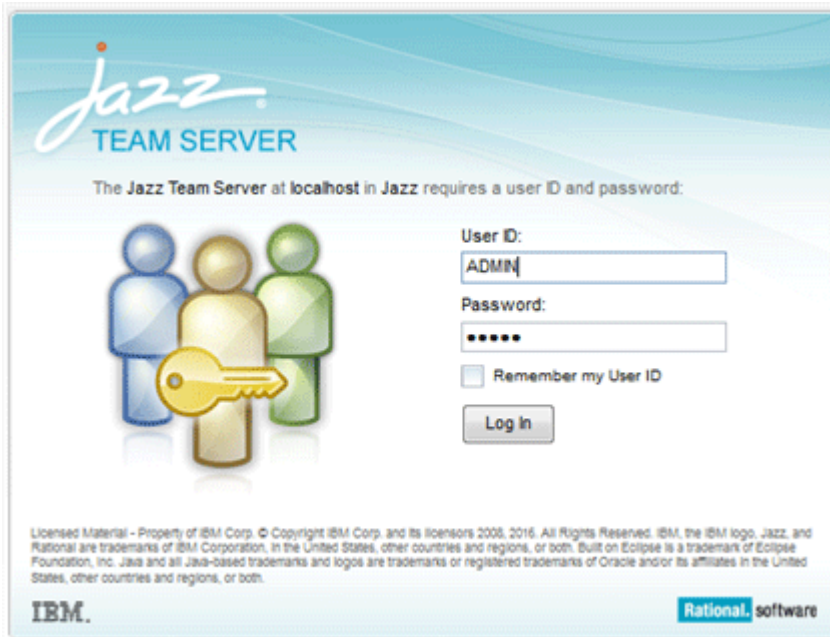
7. Take the following steps when the JTS setup wizard appears.

   a. On the **Welcome** page, select **Express Setup** and click **Next**.
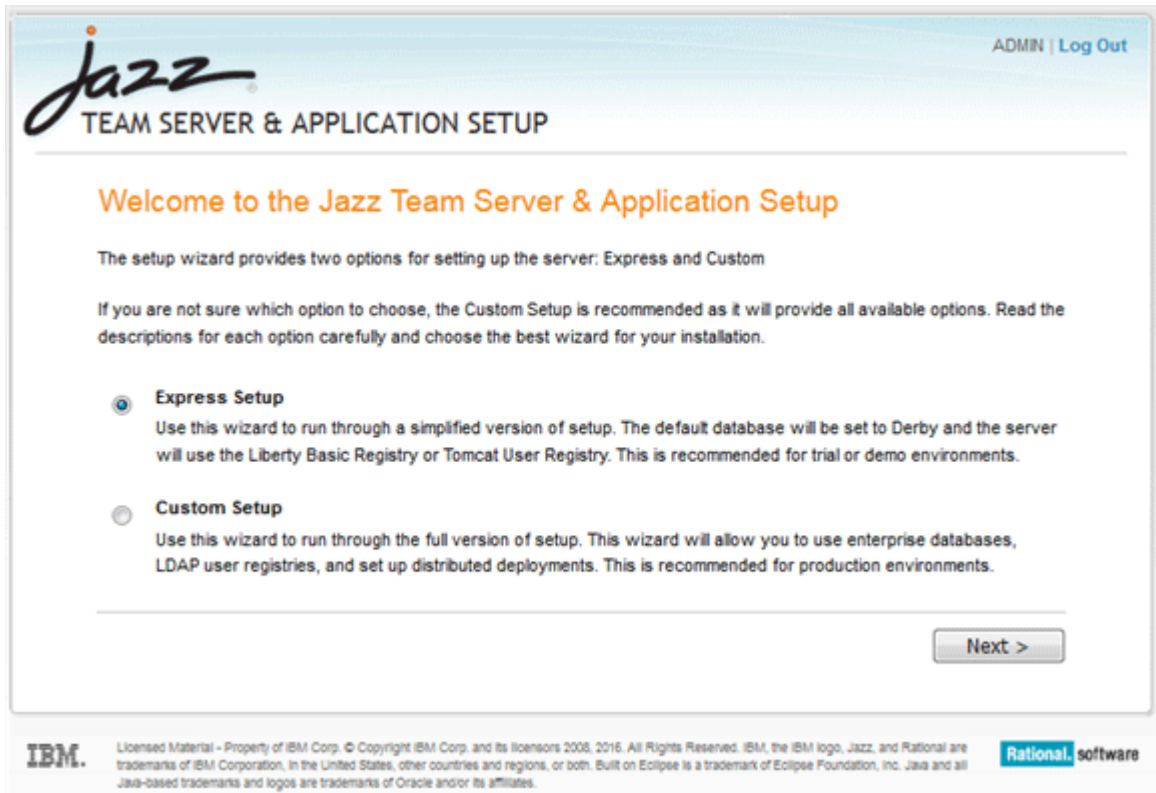
   b. On the **Configure Public URI** page, complete the following steps.

      1) **Step 1: Configure Public URI**

         The default Public URI at this step is the `https://healthcare.example.com:9753/jts` that you've set previously. You can also use the localhost as the Public URI for evaluation purpose. However, for the production server, you need to update the Public URI to a valid one. When the Public URI is set, you are not able to modify it.

Step 1: Configure Public URI

Property	Current Value
Public URI	https://healthcare.example.com:9443/jts
	Example: https://host.my.company.net:9443/jts

2) **Step 2: Confirm Public URI**

Check to confirm that you understand that when the Public URI is set, it cannot be modified.

Step 2: Confirm Public URI

⚠ The Public URI (which includes the fully qualified hostname) **must be chosen carefully** because once it is set, it becomes integral to the data on the server and any data on any other server or application that references it. This is particularly important when configuring a production server. Please review the information on this page and refer to the Planning your URIs topic for guidance and recommendations on configuring the Public URI.

☐ I understand that once the Public URI is set, it cannot be modified except with additional administrative commands and in a limited set of scenarios, which can result in broken links from other applications that do not support changing URLs.

3) Click **Test Connection** to test the connection to the server.

4) If the test is successful, click **Next**.

c. Complete the **Create Administrative User** page with the following information and click **Next** to create an administrator user for setting up ADI server in later steps.

- **User ID**: AdiAdmin
- **Name**: Adi Admin
- **Password**: AdiAdmin
- **Re-type Password**: AdiAdmin
- **E-mail Address**: adiadmin@healthcare.com

d. Click **Next** to proceed. The **Setup Progress** page appears.

e. Wait until the setup progress completes. Click **Next**.

   Note: Now you are switched to AdiAdmin user.

f. On the **Assign License** page, select the **Activate Trial** link next to the license of **IBM Application Delivery Intelligence – User**.

g. On the Client Access License Agreement dialog box that opens, select **I accept the terms in license agreement** and then click **Finish**.

h. Click **Finish** to complete the Express setup.

Setting up a project area to start applications to be analyzed

After you finish the Express setup for ADI, you need to create a project area for analysis.

Complete the following steps to set up a project area.

1. Open your Firefox browser and navigate to `https://healthcare.example.com:9753/jts/admin`. The JTS Administration home page appears.

2. Select the **Create Users** link in the **Manage Users** section.

3. Take the following steps to create users.

   Note: In these tutorials, you need to create 3 users for use.

   - Marco, a developer lead
   - Tammy, a test manager
   - Trish, a tester

   a. Complete the form for creating a user with the following information.

      - **User Name**: `Marco`
      - **User ID**: `Marco`
      - **E-mail Address**: `marco@healthcare.com`
      - In the **Client Access Licenses** section, select the checkbox in front of **IBM Application Delivery Intelligence – User**.
      - Click **Save** to create this user.

   b. Select **Users > Create User** on the top menu to create another user.
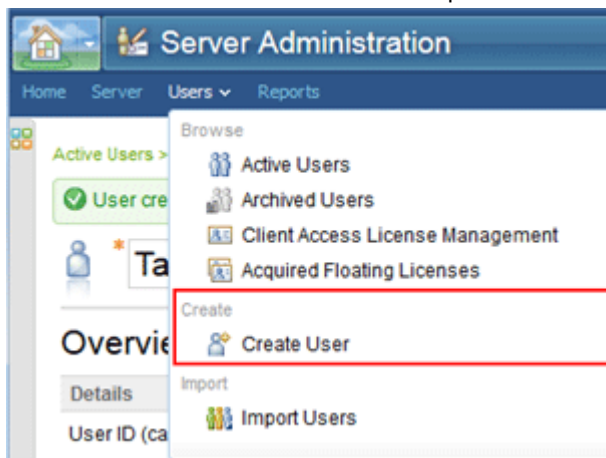
   c. Complete the form for creating a user with the following information.

      - **User Name**: `Tammy`
      - **User ID**: `Tammy`
      - **E-mail Address**: `tammy@healthcare.com`

- In the **Client Access Licenses** section, select the checkbox in front of **IBM Application Delivery Intelligence – User**.
- Click **Save** to create this user.

d. Select **Users > Create User** on the top menu to create the third user.

e. Complete the form for creating a user with the following information.

- **User Name**: `Trish`
- **User ID**: `Trish`
- **E-mail Address**: `trish@healthcare.com`
- In the **Client Access Licenses** section, select the checkbox in front of **IBM Application Delivery Intelligence – User**.
- Click **Save** to create this user.

4. Navigate your browser to ADI application from `https://healthcare.example.com:9753/adi/web` address.

   Note: You may want to disable the pop-up blocker from this page before you start.

5. On the **All Projects** page that appears, click the **project area management** link to set up a project.

   Note: For the first time user, no project is available.

6. The application navigates to the **Application Administration - IBM Application Delivery Intelligence (ADI)** page. Take the following steps to create a project area.

   a. Click **Create Project Area** on the upper right corner of the **Active Project Areas** section.
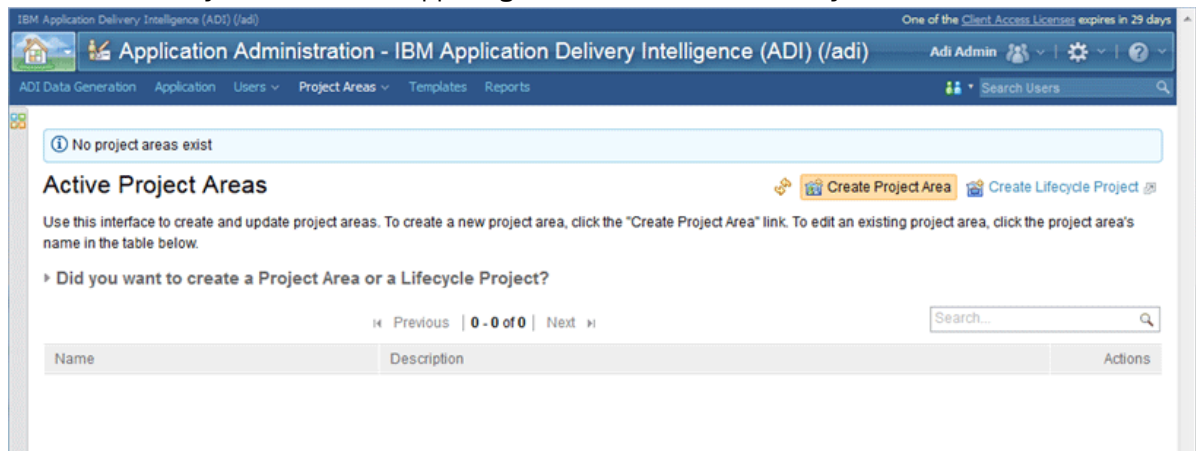
   b. Complete the **Create Project Area** form with the following information.

   - **Project Area Name**: `HealthCare4All`
   - **Summary**: `Health Care Systems`
   - **Description**: `Project to store data to be analyzed for Health Care systems.`
   - **Process**: Select **IBM Application Delivery Intelligence Application Default Process** if it is not yet selected.
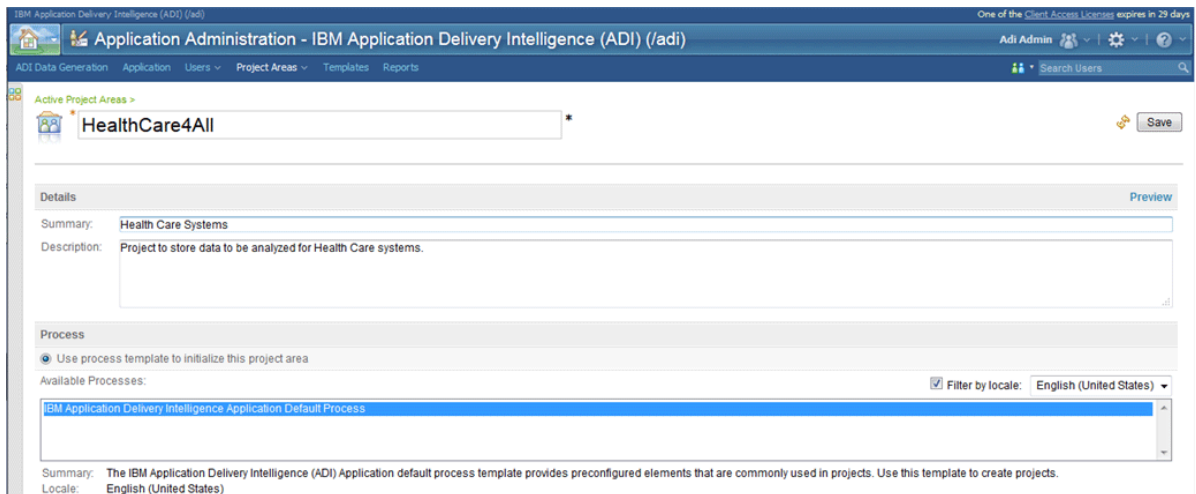
c. In the **Members** section, assign members to the project area.

1) Click the **Add** link on the upper right corner of the **Members** section.

2) On the **Select Users** pop-up window, click **Show All** to see all active members.

3) Multi-select Adi Admin, Marco, Tammy, and Trish from the **Matching users** list.

4) Click **Add and Close**.
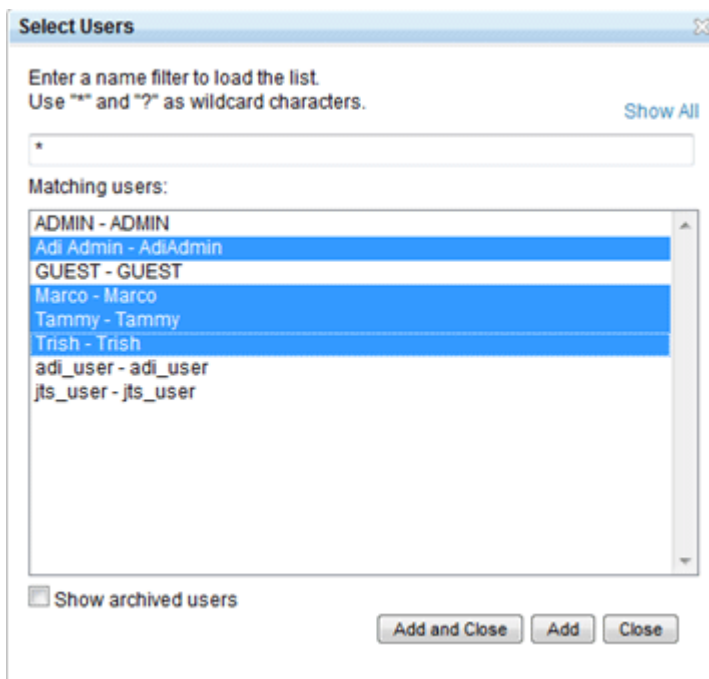
d. In the **Administrators** section, assign users as administrators of the project area.

1) Click the **Add** link on the upper right corner of the **Administrators** section.

2) On the **Select Users** pop-up window, type Ad on the search box.

3) Select **Adi Admin** from the **Matching users** list.

4) Click **Add and Close**.
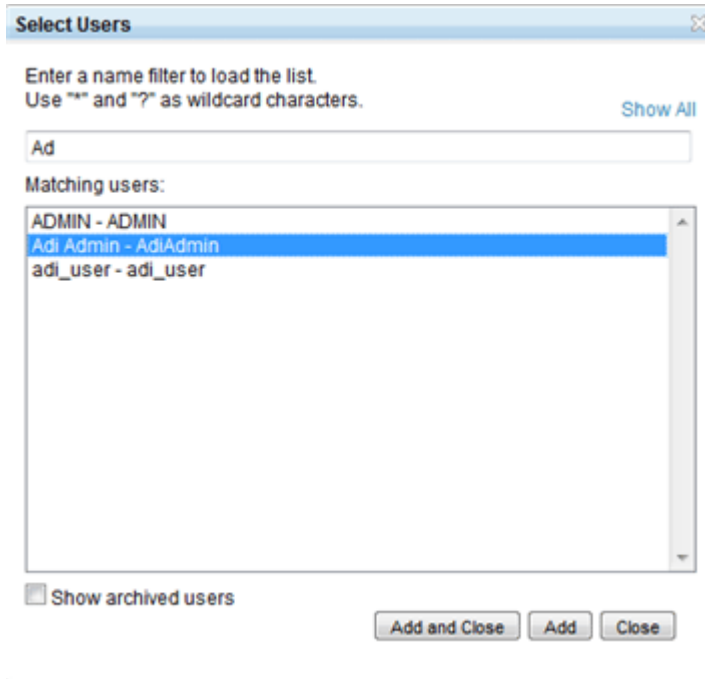
e. Click **Save** next to the project area name on the top to create the project area.

   Note: Upon saving the project area, a warning message appears showing that some new members were added without process roles. Those members will have the permissions of the "Everyone" role. The role permissions will be set up later in the next 2 steps.

f. On the pop-up window that appears, select **Cancel** to cancel sending team invitations to the new members .

   Note: If you are running the production server and you have set up the mail server, you can select the users which you would like to send team invitations to.
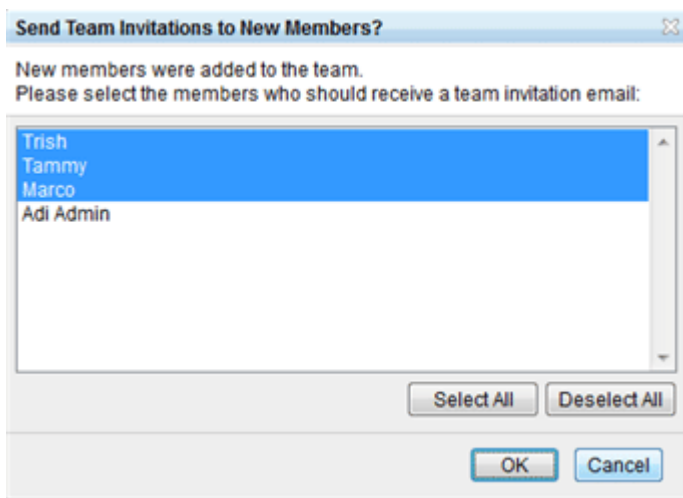
g. Go to the **Members** section to add process roles for all members.

   Note: Process roles are set in order to control the access for different users. The following three roles are available.

- **Data Provider Administrator**: As an administrator of this ADI project, you can create, modify, and delete data providers and workbooks.
- **Workbook Owner**: As a workbook owner, you can modify the memberships and settings of a workbook that you own. You can create builds and upload coverage results for Manual Builds data providers and Rational® Team Concert® Builds data providers. You can also request data collections for data providers and delete a workbook.
- **Workbook Member**: As a workbook member, you can review all data in a workbook and its data providers. You can modify only the settings of the workbook.

1) Hover over the name of Marco, the **Members** menu appears in front of the name.

2) Expand the **Members** menu and select **Add Process Roles**.
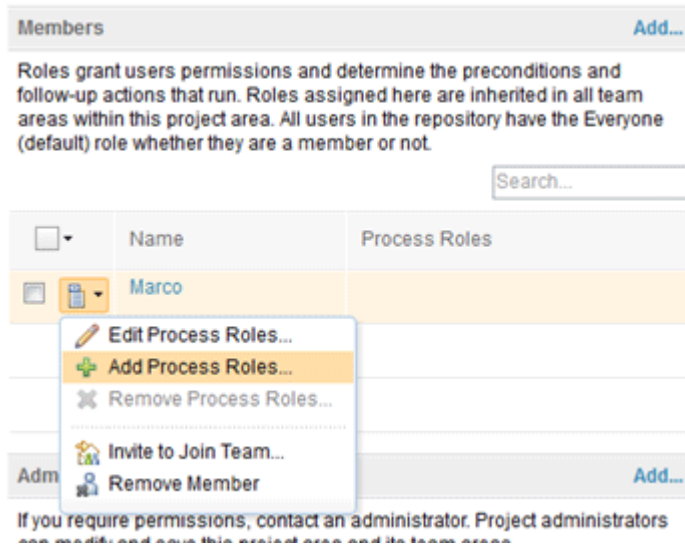
3) On the **Add Roles** pop-up window, select **Workbook Owner**.
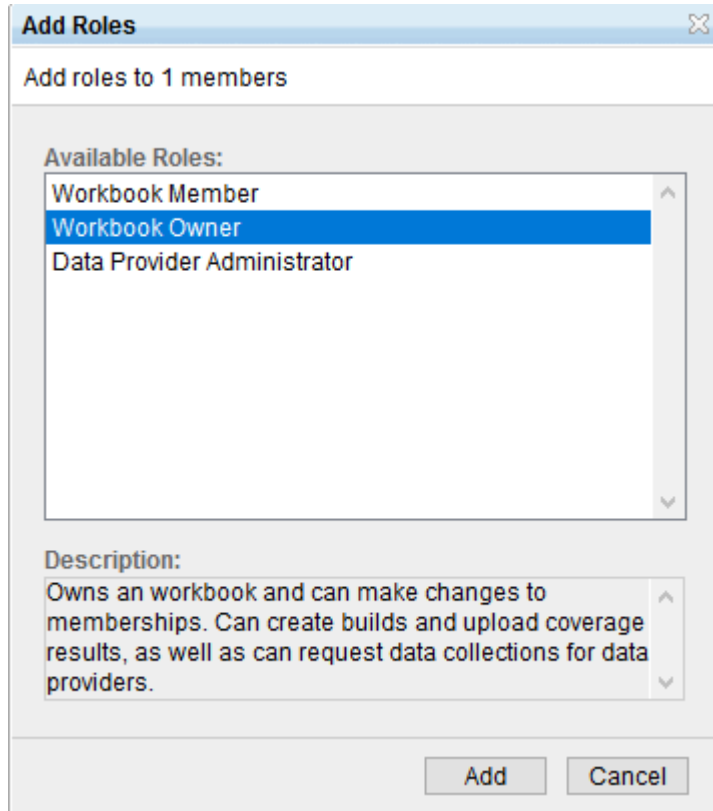
4) Click **Add** to add process role.

5) Repeat steps 1 - 4 to add process role for AdiAdmin, Tammy, and Trish. Assign the Data Provider Administrator role to AdiAdmin, the Workbook Owner role to Tammy, and the Workbook Member role to Trish.

h. Click **Save** on the upper right corner of window to save this project area.

You have now finished the setup of project area for the HealthCare4All project. You can navigate to HealthCare4All project on the ADI Application by clicking the **Home** icon on upper left corner of window and then select **IBM Application Delivery Intelligence (ADI) > HealthCare4All** from the drop-down menu that appears.
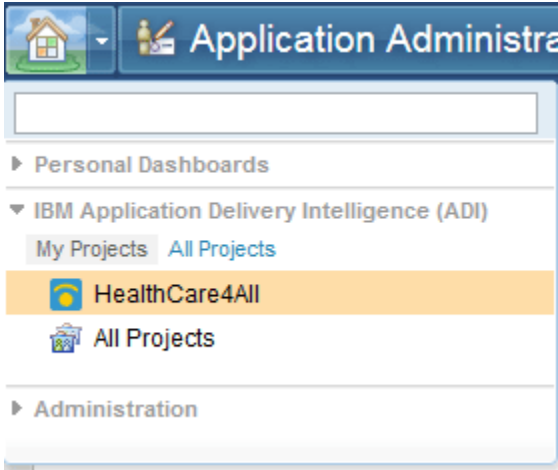
The following screen is the **Welcome** page of ADI application. ADI requires you to select a role before you can continue on the analysis.
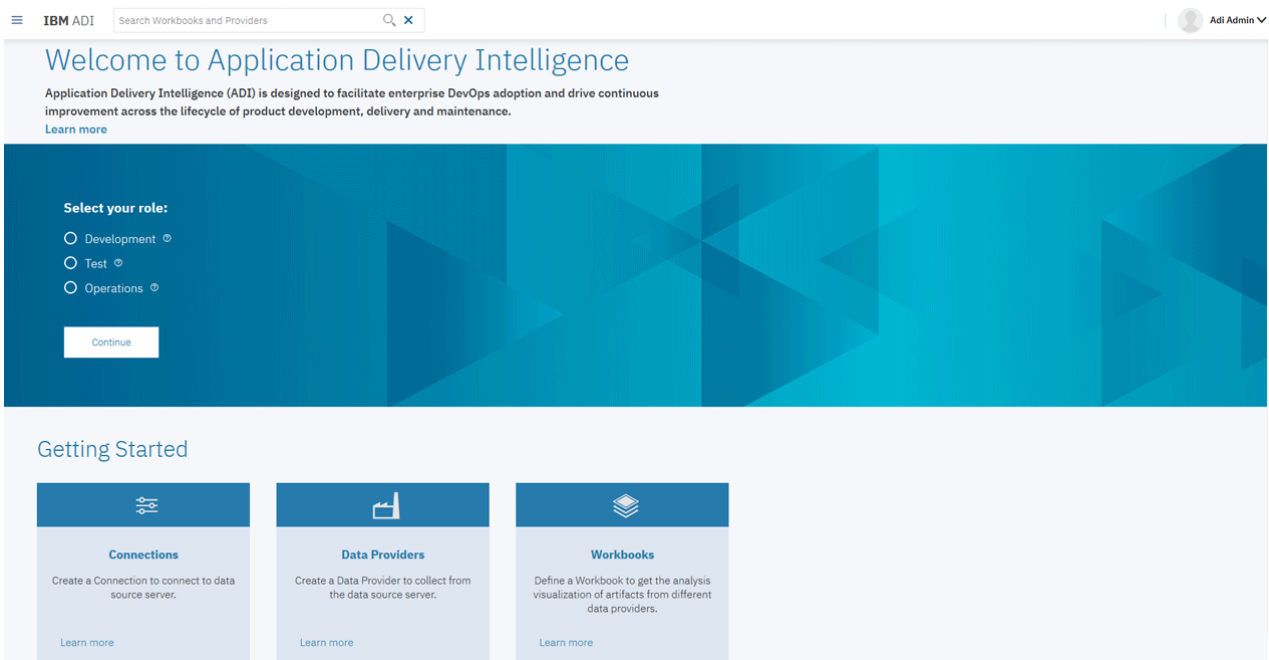
Setting up Manual Builds for Code Coverage Analysis

In this tutorial, you play the role of Alvin, an ADI administrator who is also known as Adi Admin, to create a Manual Builds data provider as a data source for code coverage results of SAM application. Then you will switch the role to Tammy, a test manager. Tammy wants to use ADI to perform a code coverage analysis for a new application, SAM Application.

You will perform the following activities in this tutorial:

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

- Defining a new application as an Analysis workbook.
- Adding the code coverage files to the first build.
- Viewing the code coverage analysis results of the first build.

Later, when the new build is available for testing, you as Tammy will add a new build to the SAM Manual Builds data provider as a place to store code coverage results from her testers. Then, you will switch your role to Trish, a tester, to test a new build and upload the code coverage results to perform analysis as well as communicate the results to the rest of your test team. Therefore, you will further perform the following activities.

- Adding a new build as a test manager.
- Adding new code coverage results to the existing build.
- Performing the code coverage analysis as a tester.

Prerequisite

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

- Installing and setting up ADI

Setting permissions for the Workbook Member role

By default, a workbook member can only view the analysis results of data provider through the workbook. But in the case that your organization would like multiple testers which are workbook members to upload their code coverage results files, you need to grant this permission to them first.

Complete the following steps to grant the permission to add and delete data of a data provider to the workbook members so that they can upload the code coverage result files.

1. Navigate your browser to `https://healthcare.example.com:9753/adi/admin`.
2. Log in as the ADI administrator with `AdiAdmin` as the user ID and `AdiAdmin` as the password.
3. Select **Project Areas** > **Browse Active Project Areas** on the top menu.
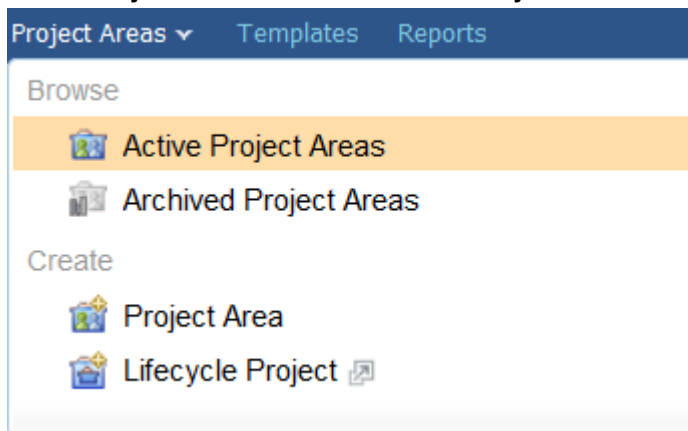
4. On the **Active Project Areas** page that appears, select **HealthCare4All**.

5. On the page for **HealthCare4All** project area that appears, select **Permissions** on the left menu.
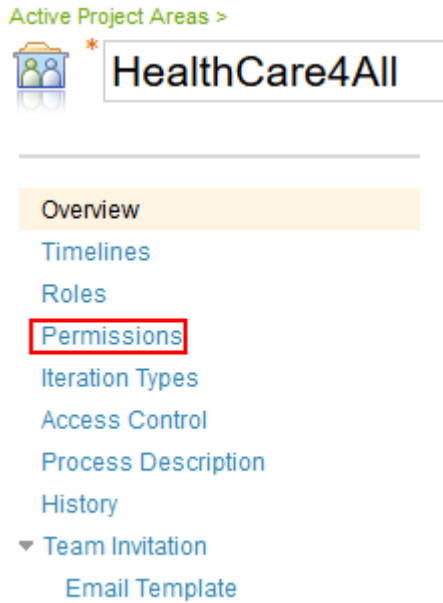
6. Select **Workbook Member** on the Permissions page.
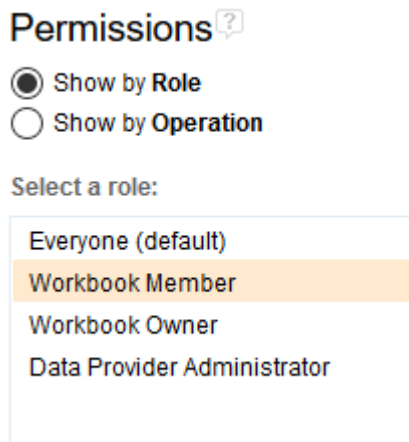
7. Expand the Data Provider row under the Operation on Permissions for Workbook Member.

8. Click **Grant Permission** action icon on the Add and delete data of a data provider row under the **Modify a data provider** section.

9. Click **Save** on the upper right corner to your settings on role permissions of the Workbook Member.

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, you need to start with adding a data provider. In this lab, you will play the role of ADI administrator to 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/adi/web/projects/HealthCare4All`.

2. Log in with `AdiAdmin` as the user ID and `AdiAdmin` as the password in the role of administrator Alvin. The Welcome page appears.

   Note: As an administrator, you do not need to select any role on this page. The role selection is only required when you want to view the analysis dashboards and reports.

3. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

4. Click **Data Providers** to go to the **Data Providers** page.

   On the **Data Provider** page, a warning message shows that no data providers have been specified yet.
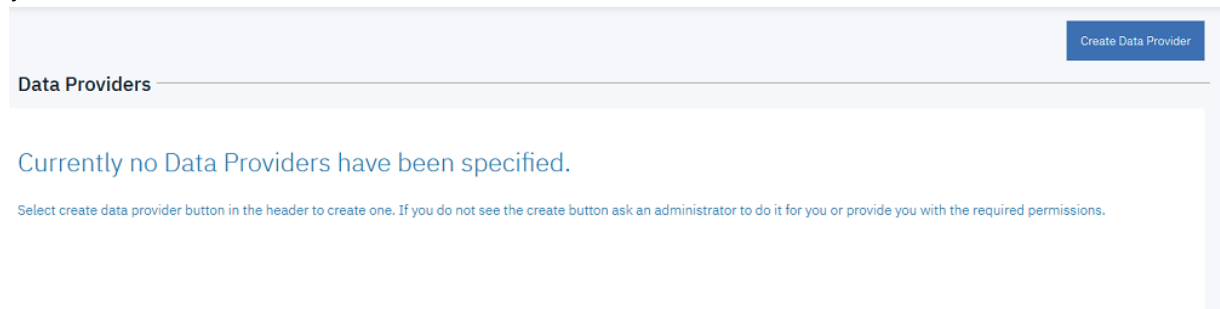
5. Click **Create Data Provider**. The **Add Data Provider** page loads with choices of available data providers.

6. Click the **Manual Builds** data provider card. A form for creating the Manual Builds data provider appears.

7. Complete the form with the following information.

   • **First Build**

     – **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.`

First Build

Build Name *
2015Nov10Build

Date Of Build *
11/10/2015, 12:00 AM

Description
November 10, 2015 build for SAM application.

- **Code Coverage Files**:

  You don't need to upload code coverage files at this point. You will upload the code coverage files as Tammy in the "Adding the code coverage files to the first build" on page 26 section.

- **Data Provider Details**

  - **Name**: `SAM Manual Builds`.
  - **Description**: `Manual builds for SAM application.`
  - **Enable headless collection support**: Make sure that the box is **unchecked**.
  - Select **All Days** for the number of days to keep data in the data warehouse.

Data Provider Details

Name *
SAM Manual Builds

Description
Manual builds for SAM application.

☐ Enable headless collection support

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

All Days

8. Click **Add** to create this data provider. The **Data Providers** page appears with the **SAM Manual Builds** data provider added.

   Note: Now you have created a Manual Builds data provider for the SAM application.

Data Providers

SAM Manual Builds

Manual builds for SAM application.
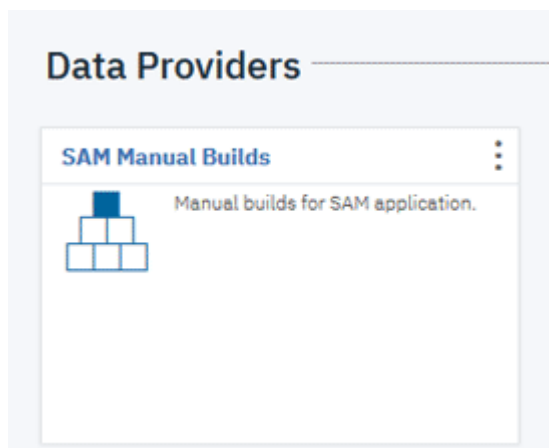
9. Optional: If you want to edit the Manual Builds data provider you created, take the following steps to update the data provider details.

   a. Click the **Options Menu** icon on the upper right corner of the **SAM Manual Builds** data provider card. An options menu appears.
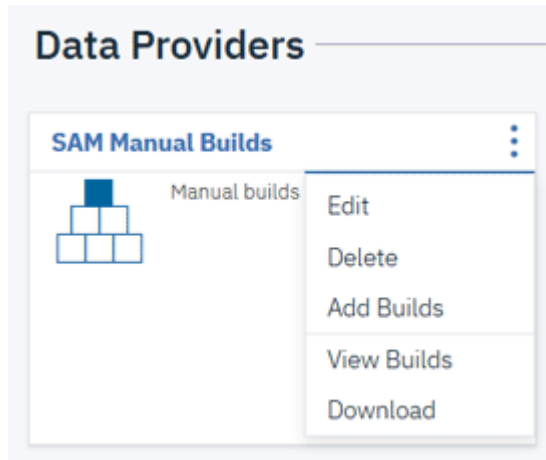
   b. Click **Edit** to edit the Manual Builds data provider information. The **Edit Data Provider** page loads.

   c. On the **Edit Data Provider** page, you can update the details of the Manual Builds data provider.
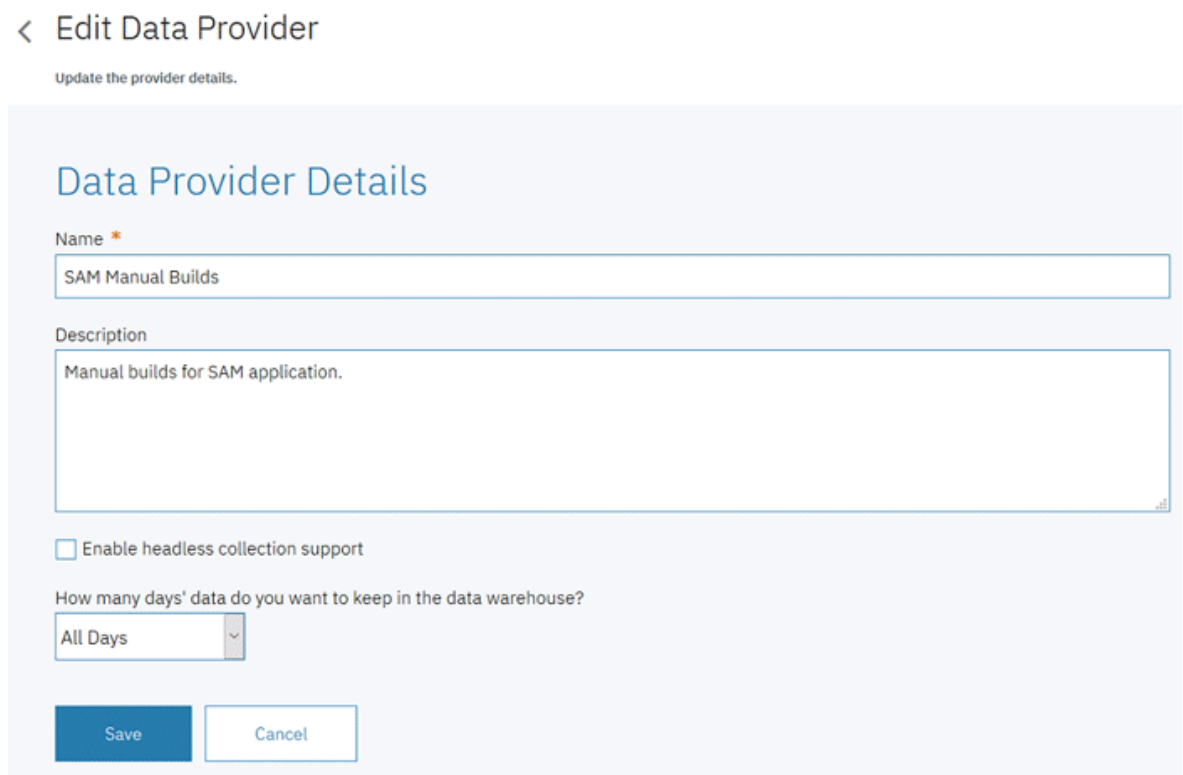
   d. Click **Save** to save your changes. You can also click **Cancel** to quite editing.

10. Click the User menu on the upper right corner of the view.
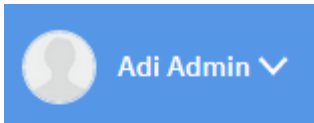
11. Select **Logout** to log out from ADI. The Login window appears. You are now logged out from ADI.

As the ADI administrator Alvin, you have now successfully created the **SAM Manual Builds** data provider. Next you will switch your role to Tammy, the test manager. You need to create a workbook to represent a new application that you want your team to use for code coverage analysis.

Defining a new application as an Analysis workbook

In ADI, 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 play the role of Tammy to define a new application as a workbook and then associate the workbook with a 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. On the ADI Login window, log in as Tammy the test manager, with Tammy as the user ID and Tammy as the password.
2. On the **All Projects** page, select the **HealthCare4All** project.
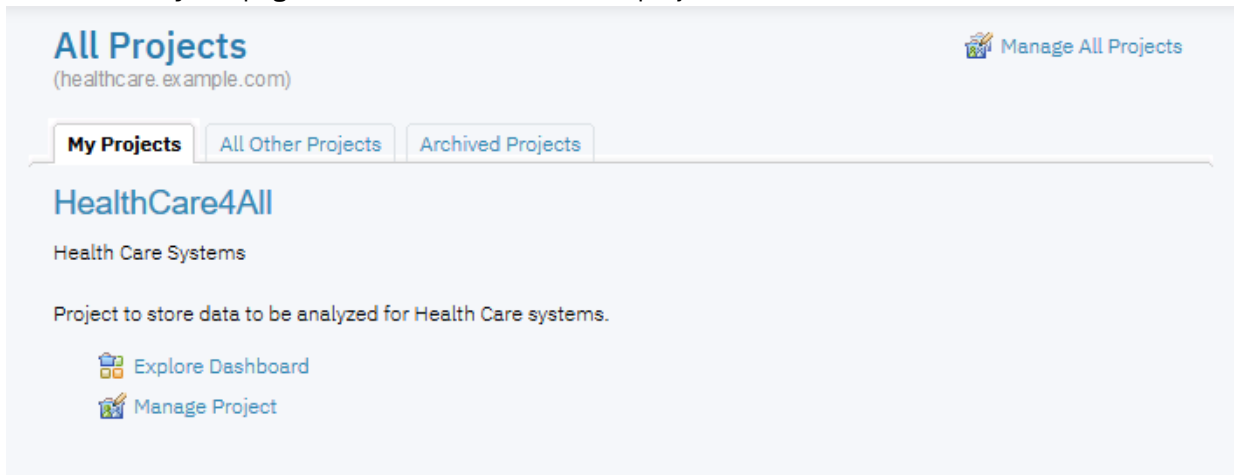
3. On the Welcome page that appears, select the **Test** role and click **Continue**.

The Workbooks page appears with a message that no workbooks have been specified at this point.

4. Click **Create Workbook** and then select **Analysis** form the drop-down list.

5. Complete the **Add Analysis Workbook** page that appears with the following information.

- **Name**: SAM Application
- **Description**: This is an analysis workbook for SAM application.
- **Contributors**: Select **Trish** as **Member**.

  Note: By default, the user who creates an Analysis workbook will be automatically added as the owner of this Analysis workbook.

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

  – **Code Coverage Threshold**: Slide the left slider to 50% and the right slider to 70%.

– **Refine Custom Scope**: You will not see any files to select at this point because you have not yet loaded the code coverage results to SAM Manual Builds.

6. Click **Add** to add this Analysis workbook. The Workbooks page appears with the Analysis workbook you added.
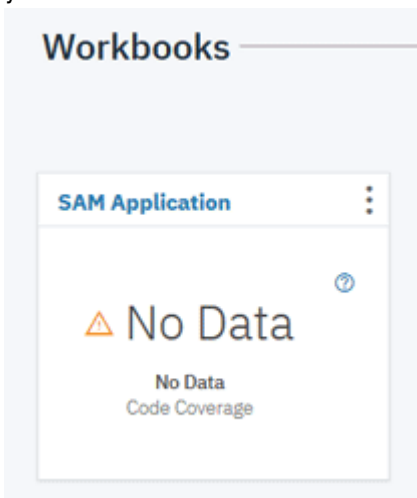
Now you have created the Analysis workbook. Next, you will load the code coverage results into the first build of the SAM Manual Builds data provider to perform the code coverage analysis.

Adding the code coverage files to the first build

In this lab, you play the role of Tammy, the test manager, to add the code coverage files for analysis to the first build that is being created in the Manual Builds data provider.

Complete the following steps to add the code coverage files to the first build.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the **SAM Manual Builds** data provider card. An options menu appears.

   Note: The options menu are different from what you have seen in the previous lab. This is because of the role permissions that are assigned to this user. Tammy has only the role permissions as a workbook owner. As a result, she can not edit nor delete a data provider. For more information about role permissions, see "Assigning role-based permissions to ADI users" on page 115.
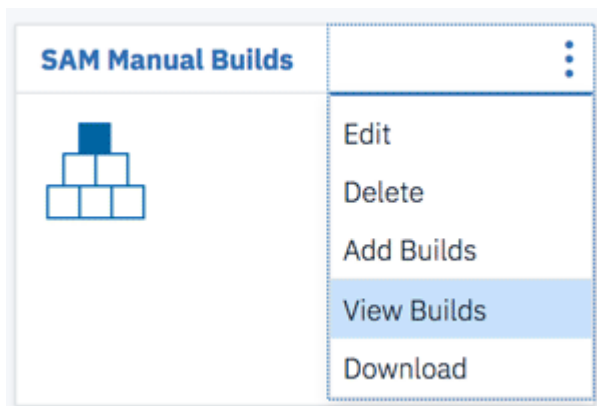
4. Select **View Builds** from the options menu that appears. The Builds page with all the builds in the data provider appears. Currently only one build is available, that is, **2015Nov10Build**.

5. Select the **Edit** icon ( ) on the **2015Nov10Build** row.

The **Edit Build** page appears.

6. On the **Edit Build** page, click the **Add** icon ( ) next to **Code Coverage Files** header section.

Edit Build

SAM Manual Builds

Build Name *

2015Nov10Build

Date Of Build *

11/10/2015, 12:00 AM

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.

No files have been uploaded

Save   Cancel

7. Select samt1.zip, samt2.zip, samt3.zip, samt4.zip, samt5.zip, samt6.zip, samt7.zip as the code coverage files.

Note: Browse to the sample data folder. Open the ***Installed ADI folder* > adi >examples> cobol-coverage > SAM App > November 10 2015**. Multi-select samt1.zip, samt2.zip, samt3.zip, samt4.zip, samt5.zip, samt6.zip, samt7.zip files.

Code Coverage Files ⊕

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

Filename	Upload Date	User	Description
samt1.zip			
samt2.zip			
samt3.zip			
samt4.zip			
samt5.zip			
samt6.zip			
samt7.zip			

Save   Cancel

8. Click **Save** to upload the code coverage files. The confirm dialog appears.

9. Click **OK** to confirm the files you upload. The Builds page appears with a message that shows the data collection is scheduled. You will be notified when the data collection is completed.
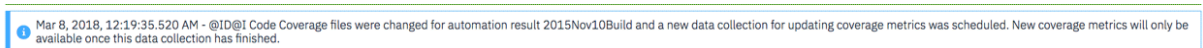
10. Click the **Back** icon on the Builds page header to go back to the **Data Providers** page.

11. Wait for a couple minutes till the data collection to complete. To view the status of data collection, click **Notifications** (bell) icon on the upper right corner of the window.
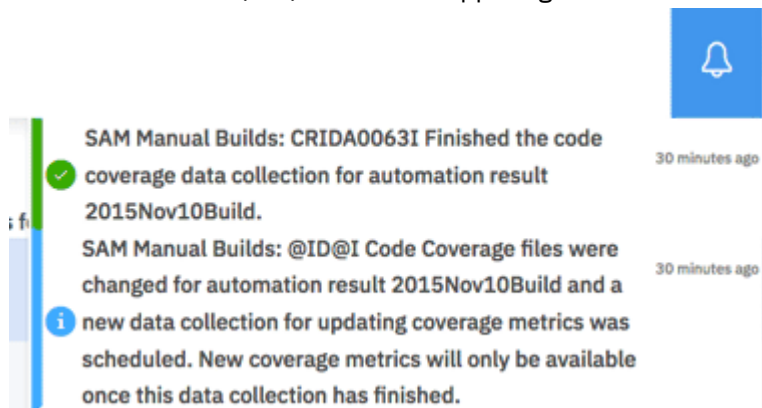
12. After the code coverage data collection is completed, you can go to the Analysis workbook to see which source files are tested based on the code coverage results that have been loaded. In real-life scenarios, there can be hundreds of source files for an application which makes it hard for the testers to focus the testing in different area. In this case, you can create multiple Analysis workbooks and define the focus area within each Analysis workbook by using the custom scope. You can complete the following steps to define the custom scope .

   a. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

   b. Click **Workbooks** to go to the Workbooks page.

   c. Click the **Options Menu** icon on the header of the **SAM Application** card. An options menu appears.

   d. Select **Edit** to edit the SAM Application information.
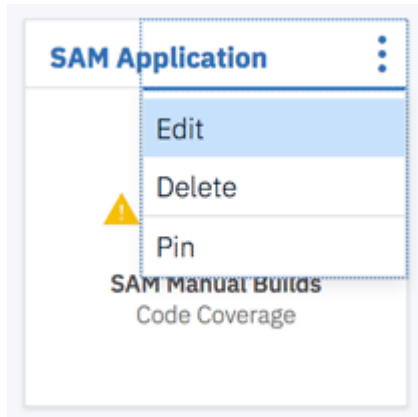
e. On the **Edit Analysis Workbook** page that appears, scroll down to **Refine Custom Scope** section and click the **Expand** icon to view the source files that are loaded to the Manual Builds data provider. These source files, SAM1.cbl and SAM2.cbl in this case, are tested as part of the code coverage results.

Notes:

- By default, if you do not select any files, ADI considers 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.
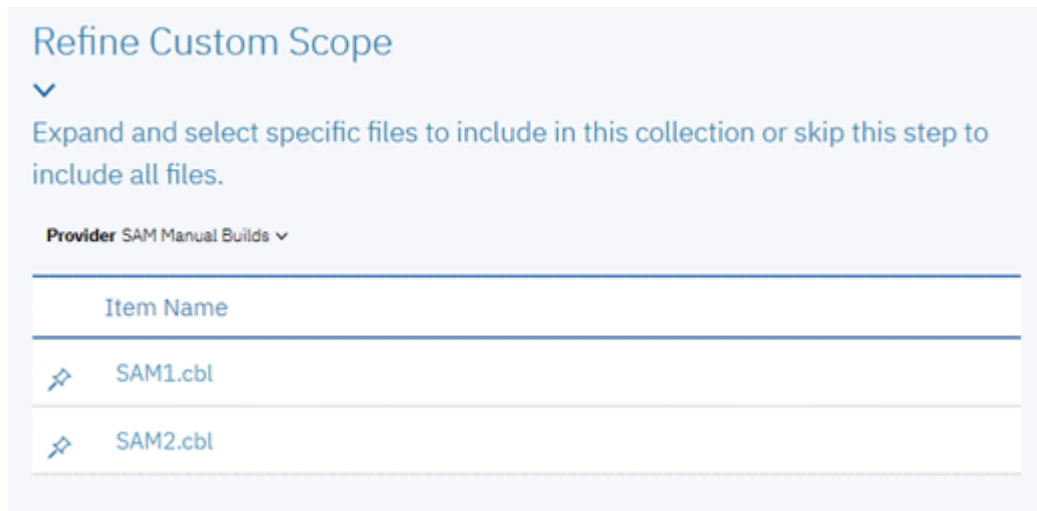
For this tutorial, you will not define the custom scope for SAM Application. Click **Cancel** to cancel the changes.

Now as Tammy you have added the code coverage files to the first build of the SAM Application Analysis workbook. You are now able to see the code coverage analysis results of the **2015Nov10Build** from the Workbooks page.

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 that you have loaded in the previous lab.

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

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Workbooks** to go to the **Workbooks** page. On the **SAM Application** card, you can see that SAM Application has only 66% code coverage.
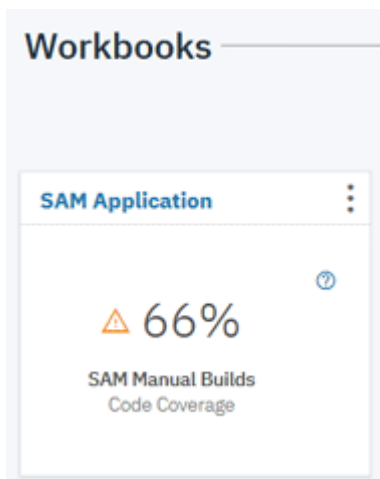
Workbooks

SAM Application

⚠ **66%**

SAM Manual Builds
Code Coverage

3. Click the header name of **SAM Application** card to view the detailed code coverage reports. The SAM Application page appears with summary charts.
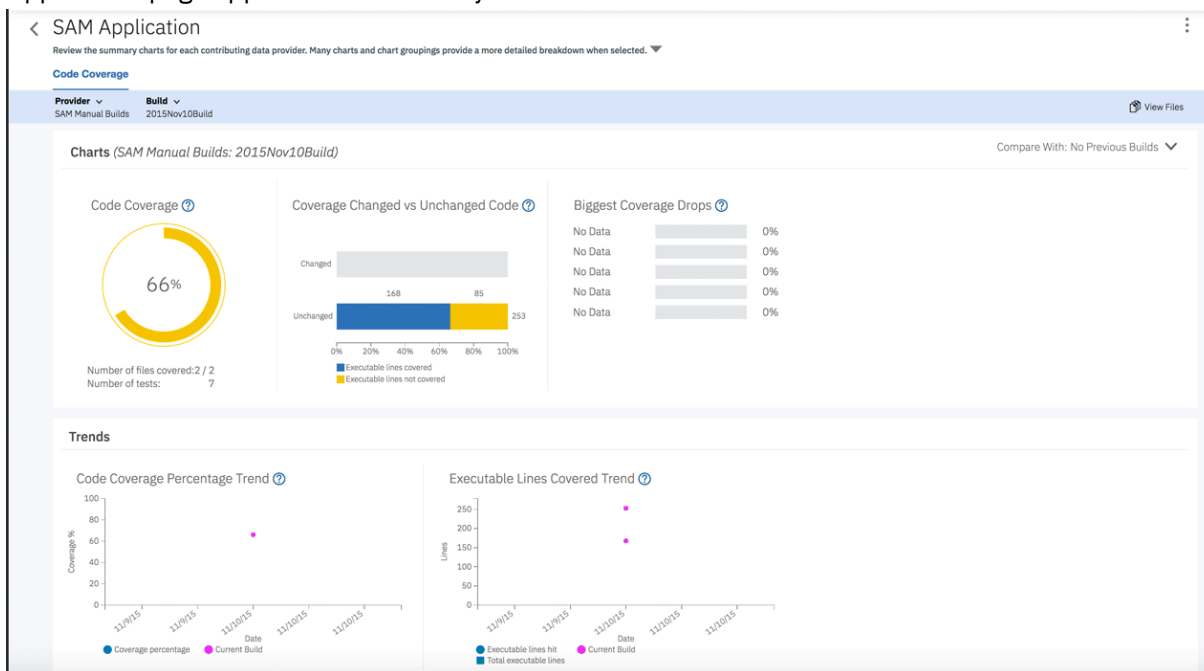
On the **SAM Application** page, you can see the analysis of **2015Nov10Build** in the **Charts** section. The first chart is the latest Code Coverage result of SAM Manual Builds that shows 66% Code Coverage. Next to the Code Coverage report are the comparison of code coverage for changed and unchanged executable lines of code and the files biggest code coverage drops. You can see no data for both reports since there is only one build for SAM application. For the same reason, you see no trend lines on the trends analysis reports in the **Trends** section.

For each report, you can hover over on the **Help** icon (⑦) to see the report information.

4. Under the SAM Application page header, click the **Expand** icon to view the **Workbook Summary**. You can see the summary information of all the data providers for this Analysis workbook.

5. Click the **Expand** icon to close the expanded section.

6. Select **View Files** icon on the upper right corner of the header to view detailed analysis.

Code Coverage

Provider ∨	Build ∨		🗐 View Files
SAM Manual Builds	2015Nov10Build		

7. Observe the **Build Analysis** page that appears. 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. 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. ADI analyzes that you can run only two tests to get the same code coverage results.

SAM Application Build Analysis	Provider ∨ SAM Manual Builds	Build ∨ 2015Nov10Build	Compare With ∨ No Previous Builds	Filters ∨	🗐 Files	📌 Pinned Files	▦ Tests	File Trends

⚠ 66%	0	0	7	2
Code Coverage	Modified Files	New Files	Historical Tests	Minimal Tests
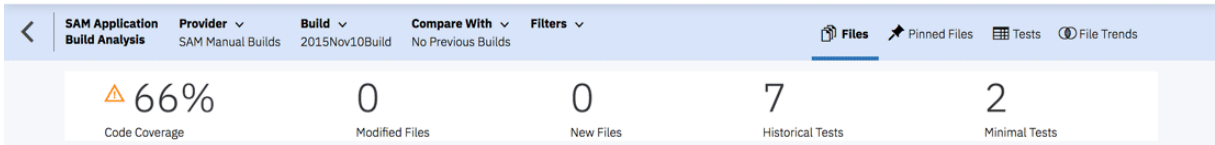
On the table section, you can see the code coverage analysis details of all the files within the build. SAM2.cbl has a red warning icon because SAM2.cbl has only 40% code coverage, which is lower than 50% threshold that you set earlier in the lab of "Defining a new application as an Analysis workbook" on page 23.

				Executable Lines ⓘ		Tests ⓘ	
Item Name ⓘ	Warnings ⓘ	Code Coverage ⓘ		Covered ⓘ	Total ⓘ	Historical ⓘ	Minimal ⓘ
📌 › SAM1.cbl		78% ▬▬▬		137	176	7	2
📌 › SAM2.cbl	⚠	40% ▬▬		31	77	3	1

8. Click the name of SAM2.cbl to view the list of Historical Tests to Run and Minimal Tests to Run. ADI recommends that for SAM2.cbl you can just run SAMT4 test to get 40% code coverage instead of running all the historical tests.

SAM2.cbl

Historical Tests to Run: 3	Minimal Tests to Run: 1
SAMT1	SAMT4
SAMT3	
SAMT4	
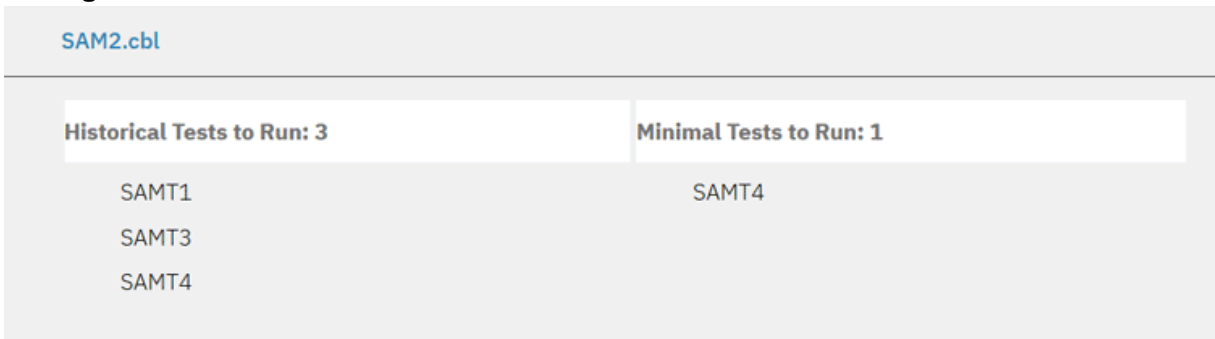
From the detailed code coverage analysis, you can see that only 31 executable lines are tested among the 77 executable lines. SAMT1, SAMT3, and SAMT4 are three test cases ran against SAM2.cbl. ADI suggested SAMT4 test case as minimal tests to run. This means that your test cases are not effective. SAMT4 covers the same executable lines that are covered in SAMT1 and SAM3.

9. Click the name of SAM2.cbl again to hide the information section.

10. Select the **Expand** icon (❯) in front of SAM2.cbl file to view the code coverage analysis result of the flowpoints within SAM2.cbl. 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.cbl.

⚠ ∨ SAM2.cbl	⚠	40% 🟥	31	77	3	1
000-MAIN		75% 🟦	6	8		
100-VALIDATE-TRAN	⚠	28% 🟥	7	25		
110-VALIDATE-TRAN-ACTION		88% 🟦	7	8		
200-PROCESS-TRAN	⚠	0%	0	24		
300-PROCESS-CPU-CRUNCH		100% 🟦	2	2		
310-CRUNCH-LOOP		88% 🟦	7	8		
(Show All 6 Flowpoints)						

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. As the test manager Tammy, you should pay attention to these test cases in the future tests.

11. Click the **Expand** icon in front of SAM2.cbl file again to close the flowpoint analysis section.

12. Select the **Tests** menu on the upper right menu.
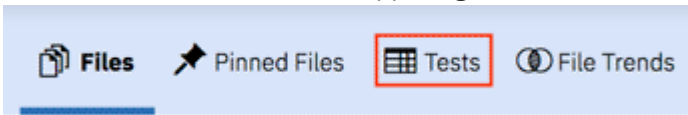
The Tests menu view loads. The view shows all the tests which are executed against the entire build and their code coverage details. You can notice the code coverage details of SAMT1, SAMT3 and SAMT4 which you observed in the previous steps. You can see that SAMT1, SAMT3, and SAMT4 have almost exactly 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.

		Executable Lines ⑦		Flowpoints ⑦		Files ⑦		
Item Name ⑦	Code Coverage ⑦	Covered ⑦	Total ⑦	Covered ⑦	Total ⑦	Tested ⑦	Missing ⑦	Elapsed Time ⑦
SAMT1	66%	168	253	21	23	2	0	4 sec.
SAMT2	55%	97	176	12	17	1	0	4 sec.
SAMT3	65%	165	253	21	23	2	0	4 sec.
SAMT4	65%	165	253	21	23	2	0	4 sec.
SAMT5	68%	120	176	15	17	1	0	4 sec.
SAMT6	62%	109	176	14	17	1	0	4 sec.
SAMT7	60%	106	176	14	17	1	0	4 sec.

SAM Application Build Analysis — Provider ∨ SAM Manual Builds — Build ∨ 2015Nov10Build — Compare With ∨ No Previous Builds — Filters ∨

⚠ 66% Code Coverage | 0 Modified Files | 0 New Files | 7 Historical Tests | 2 Minimal Tests

Now as Tammy you understand the quality issues of the **2015Nov10Build**. You communicate the results to your teams and ask them to do a better job of testing SAM2.cbl for the future builds. Now you have seen how to set up an Analysis workbook to analyze code coverage results. Next, you are informed by the development team that there is a new build available for testing. You will add a new build to the SAM Manual Builds data provider.

Adding a new build as a test manager

In this lab, you will continue to play the role as the test manager Tammy to add a new build to the existing manual builds data provider.

The development team informs Tammy that the new build will be available for testing. As Tammy, you add a new build to the SAM Manual Build without code coverage results and inform your team that the build is available for testing. Take the following steps to complete this task.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the **SAM Manual Builds** data provider card. An options menu appears.
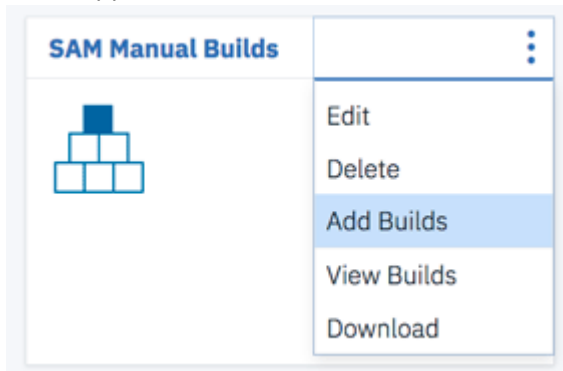
4. Select **Add Builds** from the options menu that appears. The **Add Build** page loads.

5. Complete the **Add Build** page 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**: Do not enter anything at this point.

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

You have now added a new build to SAM Manual Build. The testers can add code coverage results to the build and perform the analysis of the results they upload.

Adding new code coverage results to the existing build

In this section, you play the role of Trish, the tester, to add the new code coverage results to the existing **2015Nov11Build** build and to perform the code coverage analysis.

Trish is testing the new build. She wants to use ADI to analyze the code coverage results of the build that she is testing. She goes to ADI to upload the code coverage results to the build created by Tammy. She then reviews the results of code coverage analysis to see if she has improved the code coverage for `SAM2.cbl` as instructed by Tammy or if any other quality issues occurs. In the real-life scenario, you may have multiple testers testing the same files, each of the testers can come in to upload their own code coverage results of the builds.

Complete the following steps to add new code coverage results to an existing build and perform the code coverage analysis as Trish.

1. On the Login page, log in as Trish, a tester, with `Trish` as the user ID and `Trish` as the password.
2. On the **All Projects** page, select **HealthCare4All**.

The Welcome page appears.

3. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
4. Click **Data Providers** to go to the **Data Providers** page.
5. Click the **Options Menu** icon on the header of the **SAM Manual Builds** data provider card. An options menu appears.
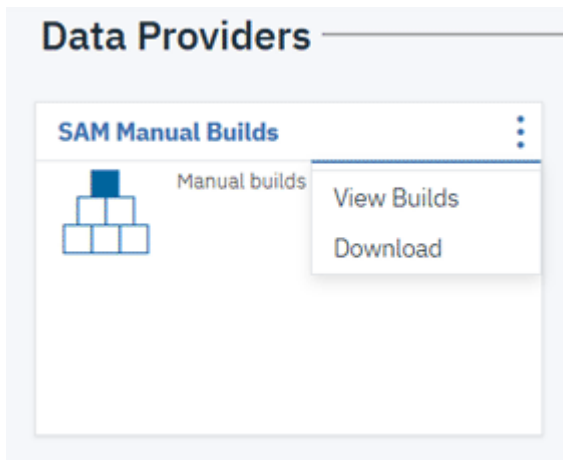
6. Select **View Builds**. The **Builds** page appears with a list of builds.

7. From the list of builds, click the **Edit** icon ( ) on the **2015Nov11Build** to edit this build.

‹ Builds ⊕

SAM Manual Builds

Create and manage builds. Select ⊕ to create a new build or use the Actions menu on the right to edit an existing build as well as upload more coverage files for a specific build.

Showing 1-2 First Page | Previous Page | Next Page

Name	Date of Build	Code Coverage	Action
2015Nov11Build	Nov 11, 2015, 12:00:00 AM	N/A	✎ ⊖
2015Nov10Build	Nov 10, 2015, 12:00:00 AM	66%	✎ ✓ ⊖

✅ Nov 30, 2017, 11:26:29.234 PM - CRIDA0066I Added 32 new rows of data to the warehouse for automation result 2015Nov10Build.

8. On the **Edit Build** page that appears, click the **Add** icon (⊕) next to the **Code Coverage Files** header.

9. Select samt1.zip, samt2.zip, samt3.zip, samt4.zip, samt5.zip, samt6.zip, samt7.zip as the code coverage files.
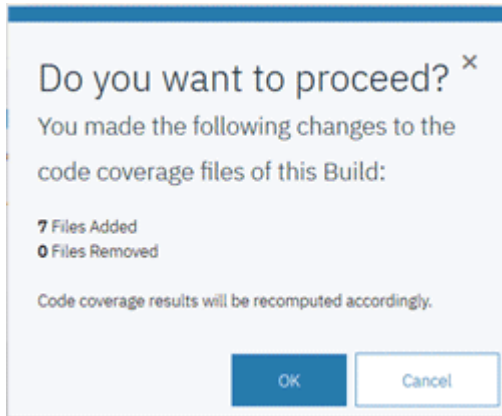
   Note: Browse to the sample data folder. Open the *Installed ADI folder* **> adi >examples> cobol-coverage > SAM App > November 11 2015**. Multi-select samt1.zip, samt2.zip, samt3.zip, samt4.zip, samt5.zip, samt6.zip, samt7.zip files.

Code Coverage Files ⊕

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

Filename	Upload Date	User	Description
samt1.zip			
samt2.zip			
samt3.zip			
samt4.zip			
samt5.zip			
samt6.zip			
samt7.zip			

Save   Cancel

10. Click **Save** to upload the code coverage files. The confirm dialog appears.

Do you want to proceed? ×

You made the following changes to the code coverage files of this Build:

7 Files Added
0 Files Removed

Code coverage results will be recomputed accordingly.

OK   Cancel
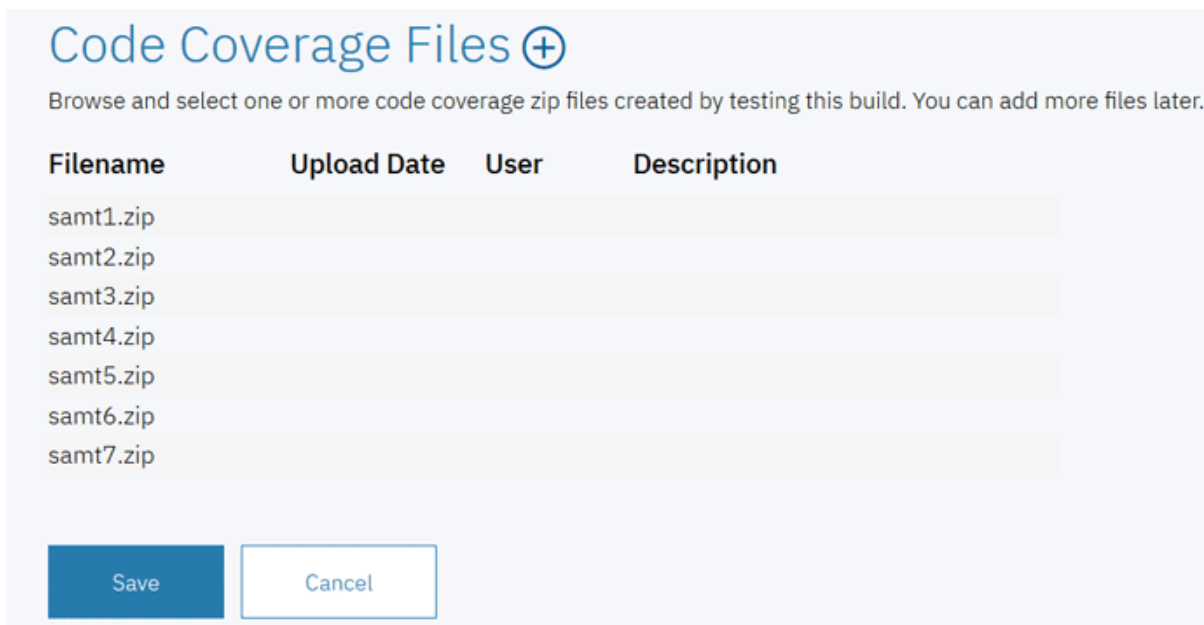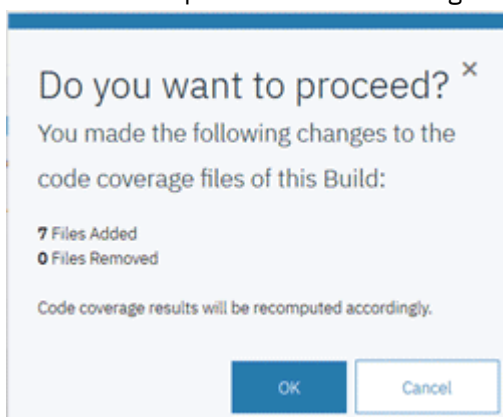
11. Click **OK** to confirm the files you upload. The **Builds** page appears with a message that shows the data collection is scheduled. You will be notified when the data collection is completed.

12. Wait for a couple minutes till the data collection to complete. To view the status of data collection, click **Notifications** (bell) icon on the upper right corner of the window.

SAM Manual Builds: CRIDA0063I Finished the code coverage data collection for automation result 2015Nov11Build.   4 minutes ago

SAM Manual Builds: @ID@I Code Coverage files were changed for automation result 2015Nov11Build and a new data collection for updating coverage metrics was scheduled. New coverage metrics will only be available once this data collection has finished.   4 minutes ago
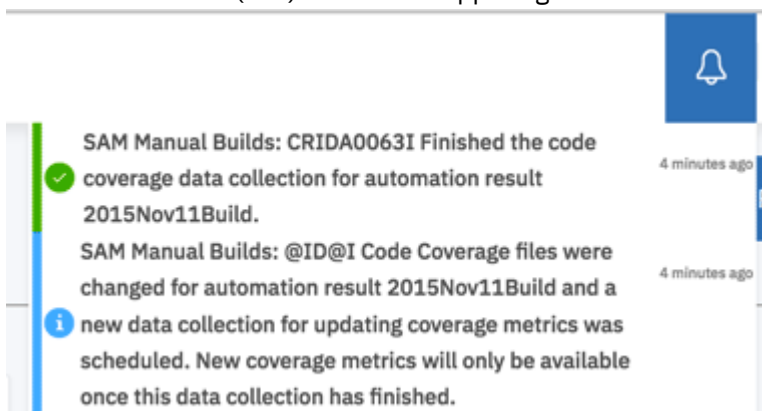
Now you have successfully added new code coverage results to the existing application. You can view the results of the code coverage analysis in the next lab.

Performing the code coverage analysis as a tester

In this lab, you will continue with the role of Trish to review the code coverage analysis results of the new build that you added in the previous lab. You can pin the critical files which 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. Click the IBM ADI header to go to the Welcome page.
2. Select the **Test** role and click **Continue**.
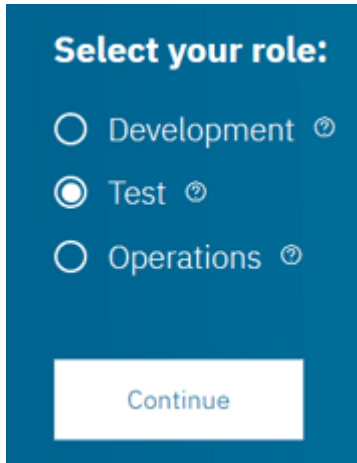
The Workbooks page appears. You can immediately see that the code coverage of the **SAM Application** is now improved to the acceptable area at 74% which is a good sign.
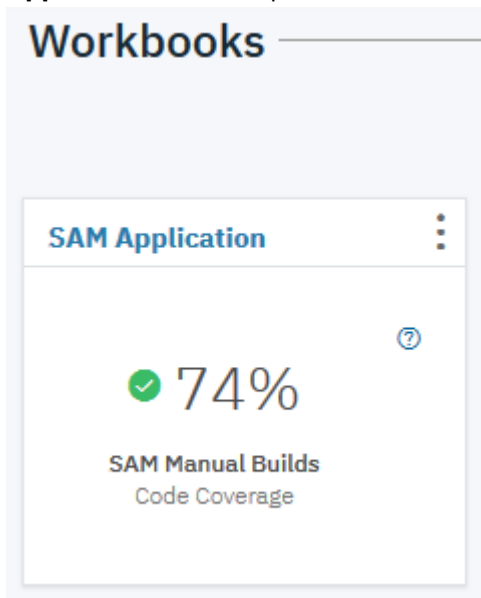
3. Click the header name of **SAM Application** card to view the detailed code coverage report. The SAM Application page appears with summary charts.
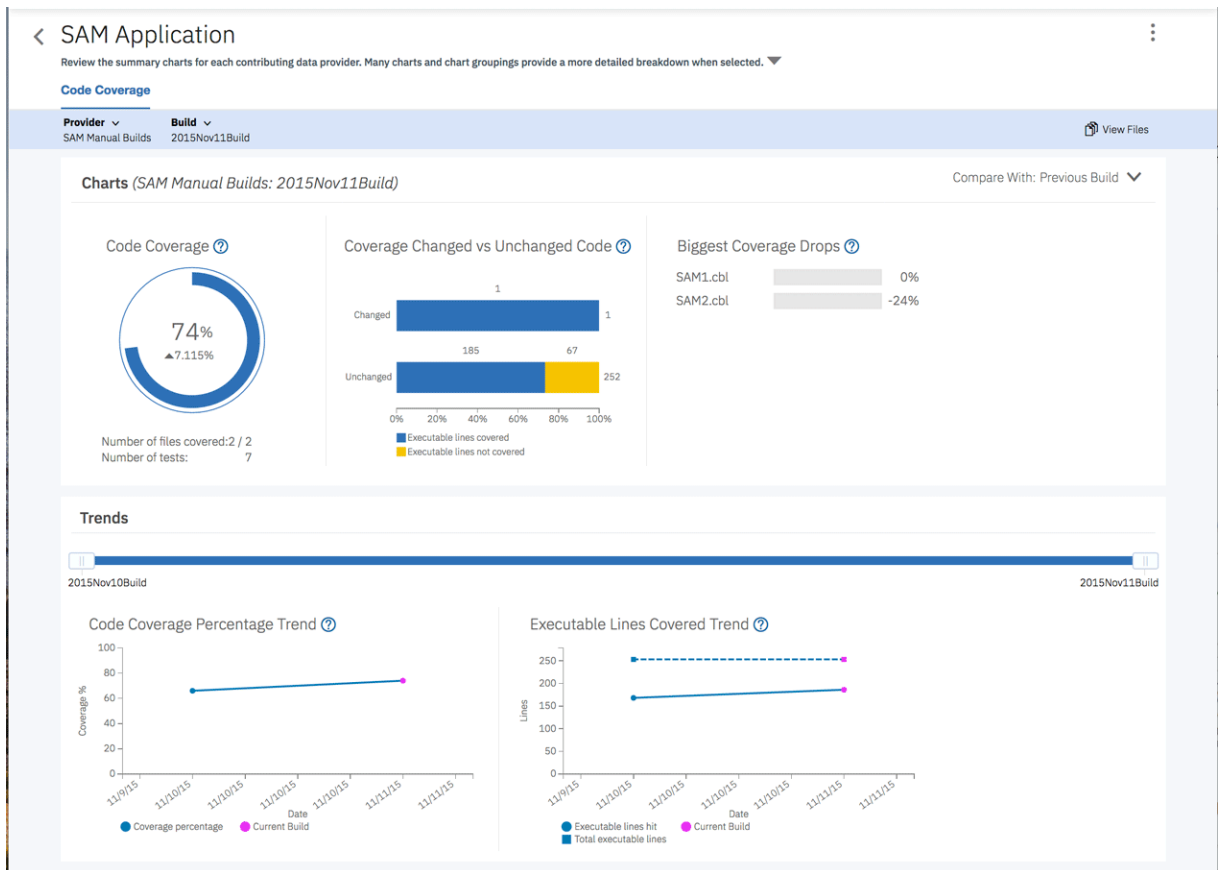
You can see that code coverage of SAM Application is now increased to 74%, up by 7.115%. The Code Coverage Percentage Trend and the Executable Lines Covered Trends are now updated with the data for Nov 10 and Nov 11 builds.

Note: From 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.

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

5. Click the **Expand** icon in front of SAM2.cbl to view the code coverage analysis for SAM2.cbl at the flowpoint level in the expended section.

Item Name	Warnings	Code Coverage	Change %	Covered	Total	Added	Updated	Change %	Deleted	Historical	Minimal
› SAM1.cbl		78%	0%	137	176	0	1	^1%	0	7	2
⌄ SAM2.cbl	⚠	64%	^24%	49	77	0	0	0%	0	5	1
000-MAIN		88%	^13%	7	8	0	0	0%	0		
100-VALIDATE-TRAN	⚠	60%	^32%	15	25	0	0	0%	0		
110-VALIDATE-TRAN-ACTION		100%	^12%	8	8	0	0	0%	0		
200-PROCESS-TRAN	⚠	33%	^33%	8	24	0	0	0%	0		
300-PROCESS-CPU-CRUNCH		100%	0%	2	2	0	0	0%	0		
310-CRUNCH-LOOP		88%	0%	7	8	0	0	0%	0		
Show All 6 Flowpoints											

6. Click **Show All 6 Flowpints** 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.

Flowpoint Analysis File SAM2.cbl ⌄ Filters ⌄				Executable Lines						Tests	
Item Name▲	Code Coverage	Change %	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	88%	0%	7	8	0	0	0%	0	5	1

Showing 1-6 First Page | Previous Page | Next Page

7. Click the **Filters** drop-down box on the top menu and check the **Code Coverage Percentage** box. The slider bars shows 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.
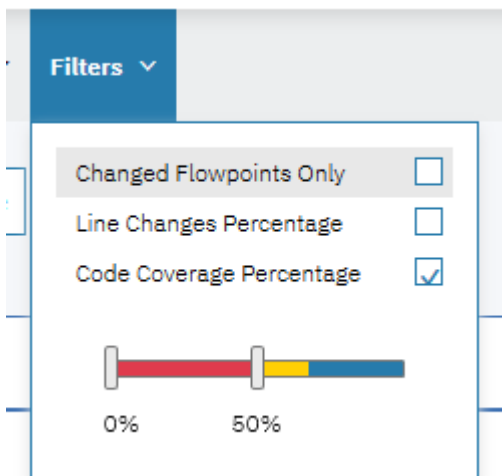
8. In the **Flowpoints Analysis** view, it now shows only the flowpoint where the code coverage is lower than 50%. You can move the right slider to 70%. The view is refreshed to show the flowpoints where the code coverage is lower than 70%.
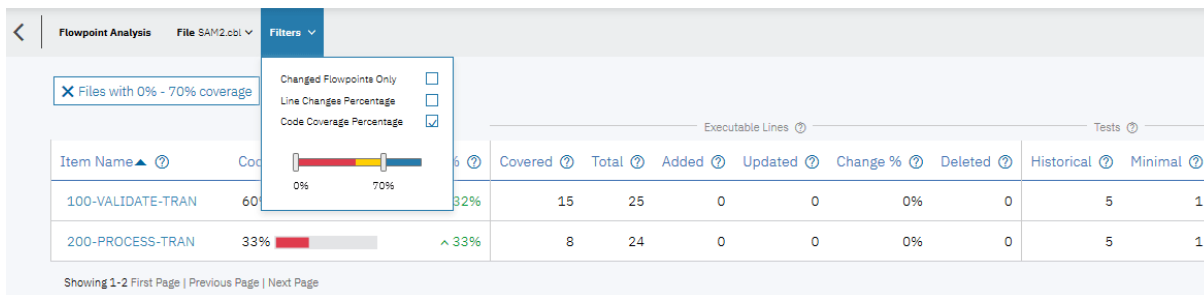
9. Click the **Back** icon to go back to the Build Analysis view.

10. Click the **Close** icon (**X**) in the filter dialog message box to clear the filter on the **Build Analysis** view. Both SAM1.cbl and SAM2.cbl are displayed.

11. Click the **Pin** icon in front of SAM2.cbl name. As Trish, you decide to pin the SAM2.cbl file since this is the file which you would like your team and yourself to pay attention to for the next round of testing for this build.

In the real-life situation, there would be hundreds of files within a build. It would be much more difficult for you to easily notice the problem areas. The filter function helps you to find these areas easier. The pin function allow you and your team to communicate on the files which need attention, such as the regression areas or the files that are more vulnerable than the others.

12. Click **Pinned Files** on the top menu.
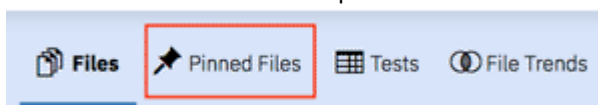
13. The updated view shows only the code coverage information of the files that are pinned. The header section shows the code coverage summary of the pinned files along with the code coverage summary for the entire build.

14. Under the **Summary** section, click on the right side number of the **Minimal Tests**.
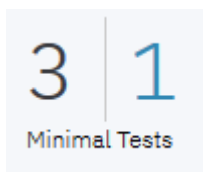
The list of the minimal tests file which ADI recommends to run for the pinned files appears. This is the list of minimal tests that ADI analyzed based on code coverage results. They can yield the same code coverage percentage as if you run all tests. In the real life scenario, when you have many test files but have limited resources to perform the manual tests, ADI can help you to reduce time to run all these tests.

| Item Name ⓘ | Code Coverage ⓘ | Executable Lines ⓘ | | Flowpoints ⓘ | | Files ⓘ | | |
		Covered ⓘ	Total ⓘ	Covered ⓘ	Total ⓘ	Tested ⓘ	Missing ⓘ	Elapsed Time ⓘ
SAMT1	74%	186	253	22	23	2	0	10 sec.

✕ Minimal Tests For Pinned Files   Search 🔍

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

Setting up automated code coverage data collections

This tutorial guides you through how to set up automated code coverage data collections.

ADI 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). ADI ships the headless collector and allows debuggers and code coverage engines to send code coverage data directly over the network so that ADI can collect the code coverage data automatically.
- Automated code coverage data collection from RTC (IBM Rational Team Concert 6.0 or higher) Build results. ADI 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 ADI.

In this tutorial, you can explore one of the automation scenarios that ADI supports. That is, the automated code coverage data collection utilizing the headless code coverage daemon program of RDz 9.5. You play the role as Tammy, the test manager, to add a new application in order to perform the code coverage analysis. This task is similar to "Setting up Manual Builds for Code Coverage Analysis" on page 17. However, in this tutorial, while creating the data provider, as Tammy you enable the headless collector for ADI to automatically collect the code coverage data.

Prerequisites

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

- "Installing and setting up ADI" on page 5
- "Setting up Manual Builds for Code Coverage Analysis" on page 17

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

In this section, Tammy the test manager wants to start monitoring code coverage for a new application called Health Care application. She wants the code coverage data to be automatically collected by ADI. So she requests Alvin, the ADI administrator, to create a new data provider to collect the data for her Health Care application.

As Alvin, you need to 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/adi/web/projects/HealthCare4All`.
2. Log in the ADI administrator with `AdiAdmin` as the user ID and `AdiAdmin` as the password.
3. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
4. Click **Data Providers** to go to the **Data Providers** page. The **Data Providers** page loads with a Manual Builds data provider: **SAM Manual Builds**.
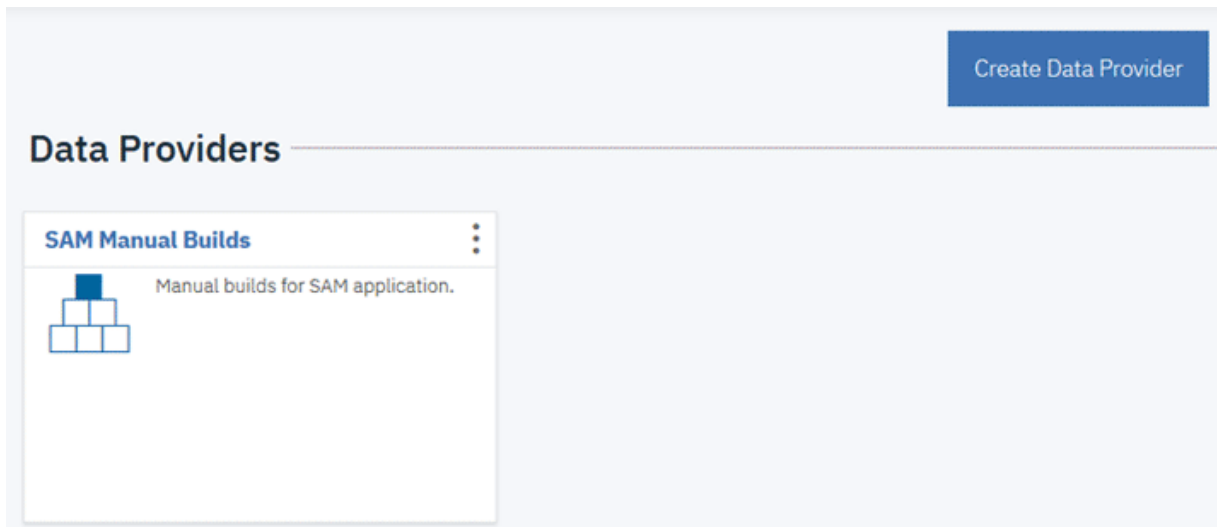
5. Click **Create Data Provider**. The **Add Data Provider** page loads with choices of available data providers.
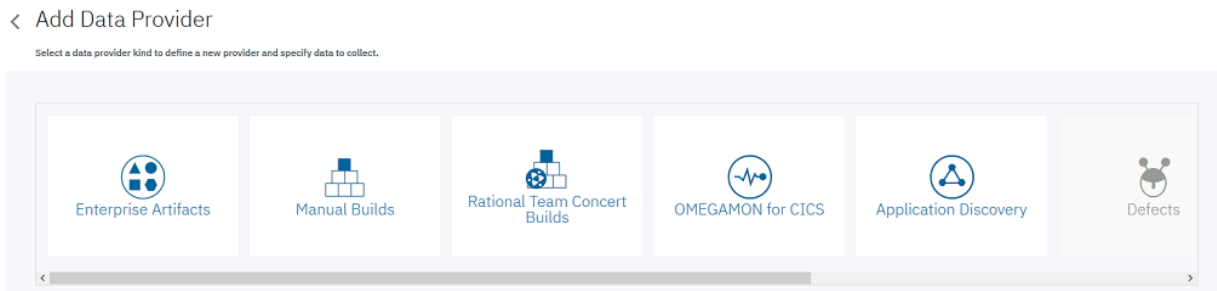
6. Click the **Manual Builds** data provider card. A form for creating the Manual Builds data provider appears.

7. Complete the form with the following information.

   a. **First Build**

      1) **Build Name**: 2016Feb2Build

      2) **Date of Build**: 2/2/2016, 12:00 AM. You can select the date from the drop-down calendar to fill in the **Date of Build** field.

      3) **Description**: February 2, 2016 build for Health Care application.

   b. **Code Coverage Files**

You won't load any code coverage files at this moment.
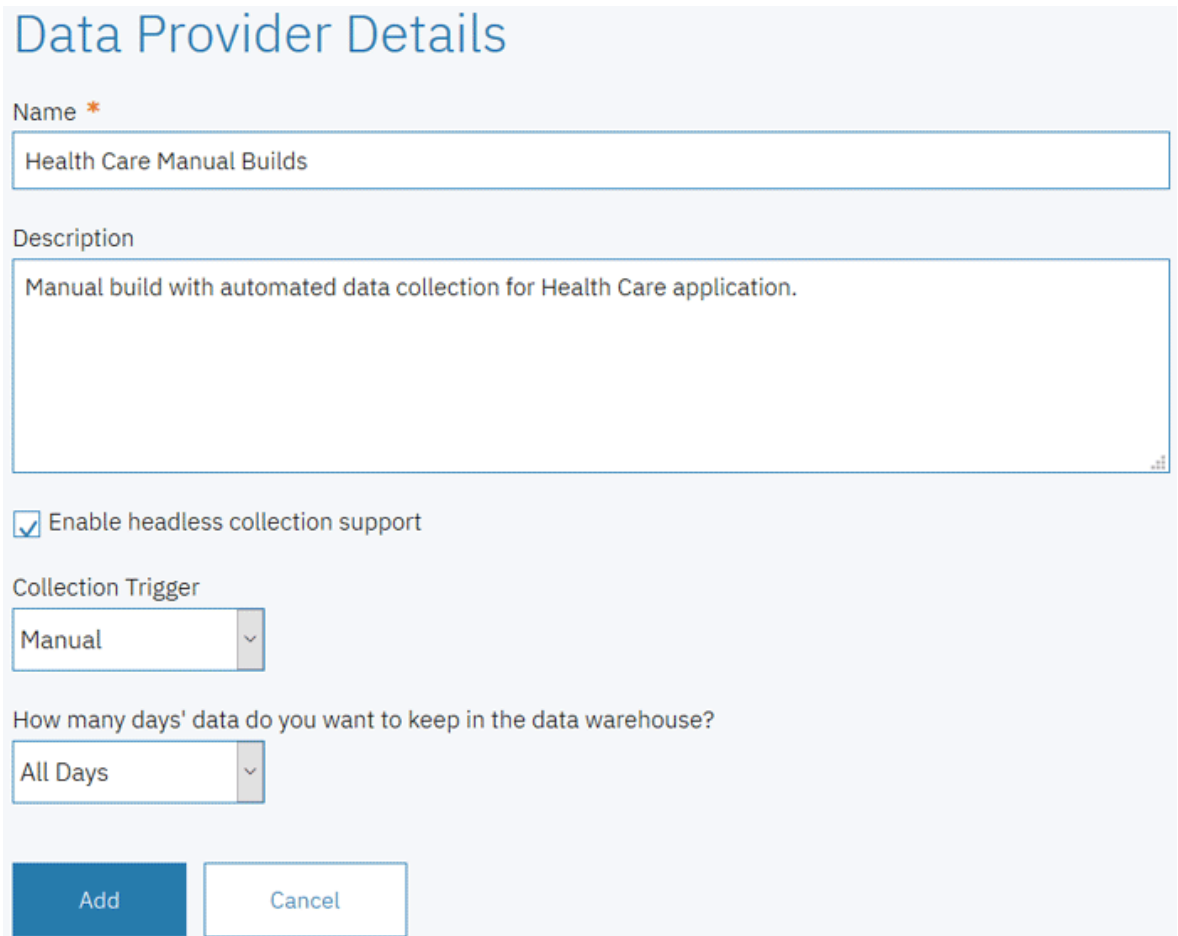
c. **Data Provider Details**

1) **Name**: `Health Care Manual Builds`

2) **Description**: `Manual build with automated data collection for Health Care application.`

3) **Enable headless collection support**: Select the check box.

4) **Collection Trigger**: Select **Manual**.

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

   This 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.

5) Select **All Days** in number of days you want 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**.

Data Provider Details

Name *

Health Care Manual Builds

Description

Manual build with automated data collection for Health Care application.

☑ Enable headless collection support

Collection Trigger

Manual

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

All Days

| Add | Cancel |

8. Click **Add** to add this provider. The **Data Providers** page appears with the **Health Care Manual Builds** added as the first card.
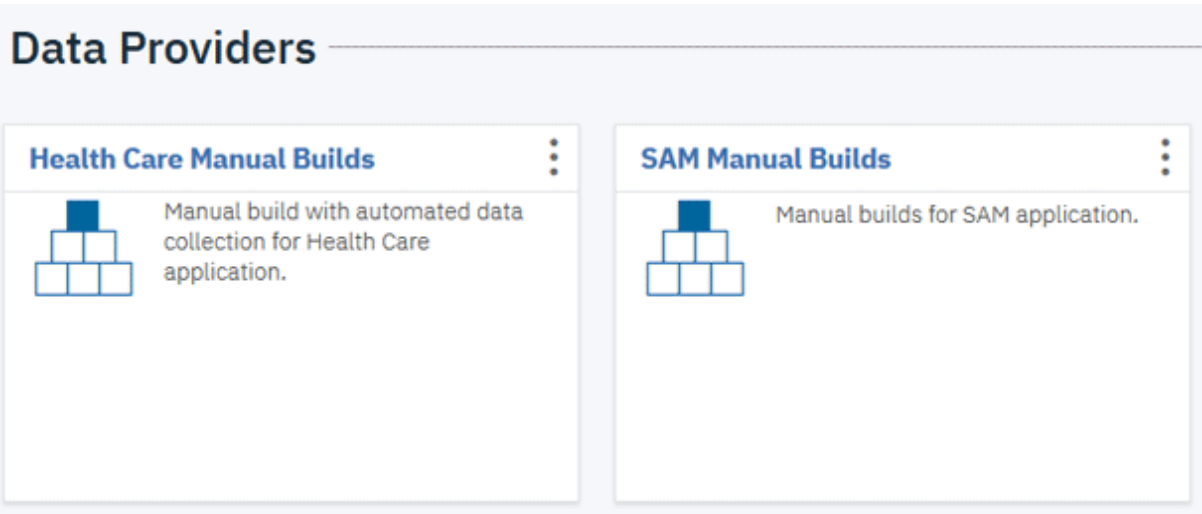
9. Click the **Options Menu** icon on the upper right corner of the **Health Care Manual Builds** card. An options menu appears.
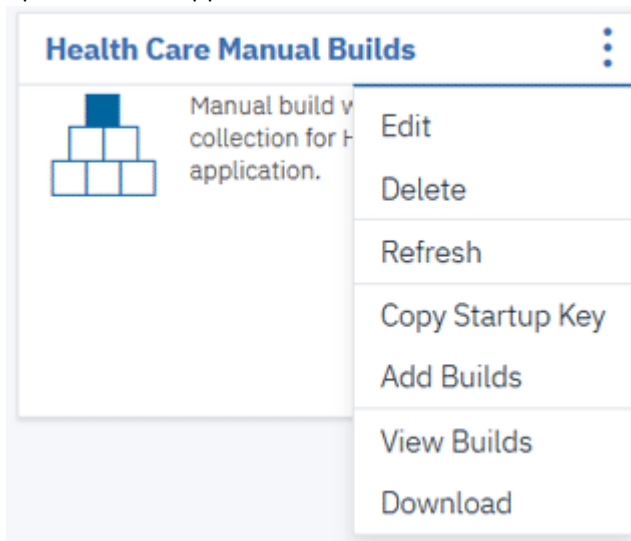
10. Select **View Builds** to view all the builds of this Manual Builds data provider. The Builds page appears with one build available.

    Note: The **Code Coverage** value indicates **N/A** at this point because ADI has not yet collected the code coverage data.

< Builds ⊕

Health Care Manual Builds

Create and manage builds. Select ⊕ to create a new build or use the Actions menu on the right to edit an existing build as well as upload more coverage files for a specific build.

Showing 1-1 First Page | Previous Page | Next Page

Name	Date of Build	Code Coverage	Action
2016Feb2Build	Feb 2, 2016, 12:00:00 AM	N/A	✎ ⊖
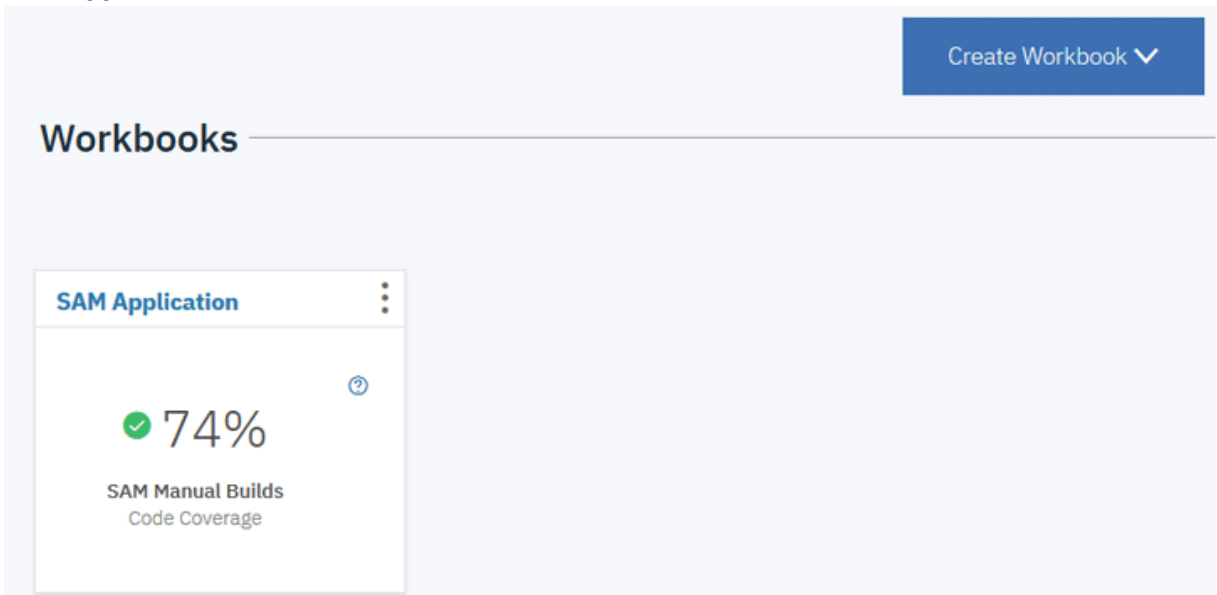
Showing 1-1 First Page | Previous Page | Next Page

11. Select the **Back** icon next to the Builds page header to go back to the Data Providers page.

    Alvin has now finished creating the data provider with automated data collection. Next he will create a workbook, associate the workbook with this data provider, and set up a build to collect code coverage data automatically for analysis.

    Note: In the real-life scenario, Tammy, the test manager can create her own workbook and trigger the data provider to collect code coverage data. For the purpose of this demo, you will continue as ADI admin.

12. Click the IBM ADI header to go to the Welcome page.

13. Select the **Test** role and click **Continue**. The Workbooks page appears with one workbook available: **SAM Application**.

Create Workbook ⌄

Workbooks

SAM Application ⋮

✔ 74% ⑦

SAM Manual Builds
Code Coverage

14. Click **Create Workbook** and then select **Analysis** form the drop-down list.
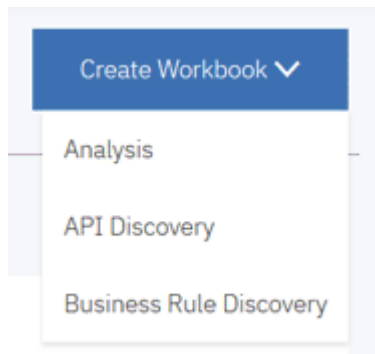
The Add Analysis Workbook page appears.

15. On the Add Analysis Workbook page, fill in the following information.

- **Name**: zMobile Health Care.
- **Description**: This is an analysis workbook for the core Health Care application.
- **Contributor Lis**t: Select **Tammy**, **Marco** as **Owner** and **Trish** as **Member**.
- **Provider**: Select **Health Care Manual Builds**. The **Code Coverage Data Provider Settings** section appears.
  - **Code Coverage Threshold**: Slide the left slider to 60% and the right slider to 80%.
  - **Refine Custom Scope**: You will not see any files to select at this point because you have not yet loaded the code coverage results to Health Care Manual Builds.

16. Click **Add** to add this Analysis workbook. The Workbooks page appears with the Analysis workbook you added.

17. Click the User menu on the upper right corner of the view.

18. Select **Logout** to log out from ADI.

You have successfully created the Health Care workbook with the first build and set it up in the way that it can automatically collect the code coverage files. In the next section, you will play the role of Tammy, the test manager, to load the code coverage files into the location where ADI can collect them for analysis. In real practices, the step to prepare and load the code coverage files into an ADI location can be done automatically depending on the programming languages and tools you are using to generate code coverage data. For more information, see Chapter 6, "Preparing external data sources," on page 127.

Loading the code coverage results and triggering the data collection from ADI

In this lab, you play the role of Tammy, the test manager, to load the code coverage results to the location where ADI can collect data automatically. Then you trigger the zMobile Health Care Manual Builds data provider to collect the code coverage results.

As Tammy you need complete the following steps.

1. Log in as the test manager with Tammy as the user ID and Tammy as password. The All Projects page appears.

2. Click **HealthCare4All**. The Workbooks page loads with two workbooks: **SAM Application** and **zMobile Health Care**.

Note: At this point, no data is available for code coverage analysis report of **zMobile Health Care**. You need to load the code coverage files into the ADI location and trigger ADI to collect code coverage data.

3. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

4. Click **Data Providers** to go to the **Data Providers** page.

5. On the Data Providers page, click the **Options Menu** icon on the upper right corner of the **Health Care Manual Builds** data provider card. An options menu appears.
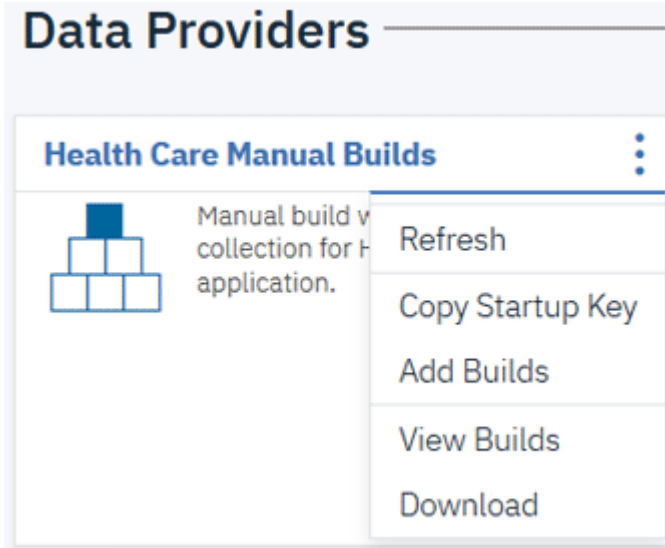
6. Select the **Copy Startup Key** from the option list to copy the start up key to the clipboard.

7. Open the Notepad program and paste the startup key to the Notepad. You will see the text similar to the following sample. Remember the provider ID. In this case the providerid is **LFB1QG**.

```
TEST(,,,TCPIP&127.0.0.1%8005:*)
ENVAR("EQA_STARTUP_KEY=CC,,providerid=LFB1QG,testid=<fill in your test-id>")
```

8. Browse to the sample data folder: `Installed ADI folder > adi > examples> cobol-coverage > zMobile Health App > Build1`. Select all the zip files under `Build1` folder and copy them to the clipboard.

9. Browse to the ADI location. Select **adi > server > conf > adi > headless-CC-files**.

10. Open the folder that has the same name as providerid. From this example it is the **LFB1QG-files** folder. Paste all the zip files you copied to this folder.

Name	Date modified	Type	Size
QueryVisit.zip	11/28/2017 11:10 ...	Compressed (zipp...	22 KB
QueryThreshold.zip	11/28/2017 11:10 ...	Compressed (zipp...	15 KB
QUERYPRI.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
QuerypatientErr2.zip	11/28/2017 11:10 ...	Compressed (zipp...	9 KB
QueryPatientErr1.zip	11/28/2017 11:10 ...	Compressed (zipp...	9 KB
QueryPatient.zip	11/28/2017 11:10 ...	Compressed (zipp...	9 KB
QueryNo.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
QueryMeds.zip	11/28/2017 11:10 ...	Compressed (zipp...	15 KB
QueryMedErr.zip	11/28/2017 11:10 ...	Compressed (zipp...	9 KB
QueryMD.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
QueryIn.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
QueryID.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
QueryDP.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
PrimaryMenu.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
ExitPrimMenu.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
ERRPRI.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
DELPRI.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
AddvisitErr.zip	11/28/2017 11:10 ...	Compressed (zipp...	10 KB
AddVisit.zip	11/28/2017 11:10 ...	Compressed (zipp...	22 KB
AddThreshold.zip	11/28/2017 11:10 ...	Compressed (zipp...	17 KB
ADDPRI.zip	11/28/2017 11:10 ...	Compressed (zipp...	6 KB
AddPatientErr2.zip	11/28/2017 11:10 ...	Compressed (zipp...	14 KB
AddPatientErr1.zip	11/28/2017 11:10 ...	Compressed (zipp...	7 KB
AddPatient.zip	11/28/2017 11:10 ...	Compressed (zipp...	16 KB
AddmedsErr.zip	11/28/2017 11:10 ...	Compressed (zipp...	14 KB
Addmeds.zip	11/28/2017 11:10 ...	Compressed (zipp...	17 KB

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 6, "Preparing external data sources," on page 127.

11. Go back to the Data Providers page and click the **Options Menu** icon on the upper right corner of the **Health Care Manual Builds** card. The option list appears again.
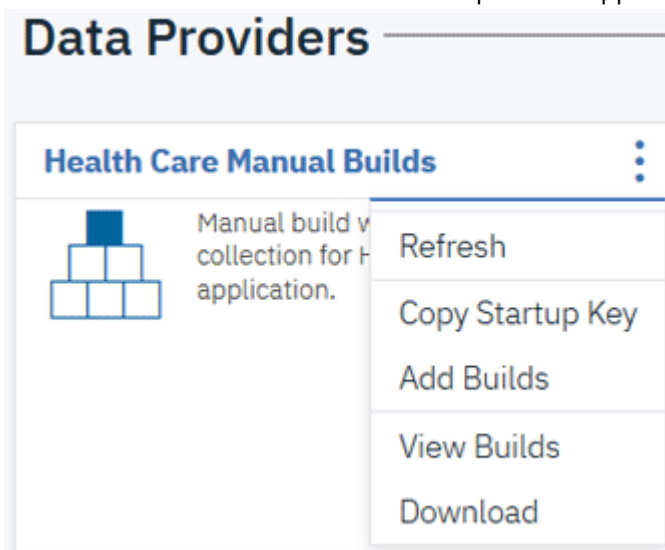
12. Select **Refresh** to trigger ADI to collect the code coverage data. The **Data Collection** window appears.
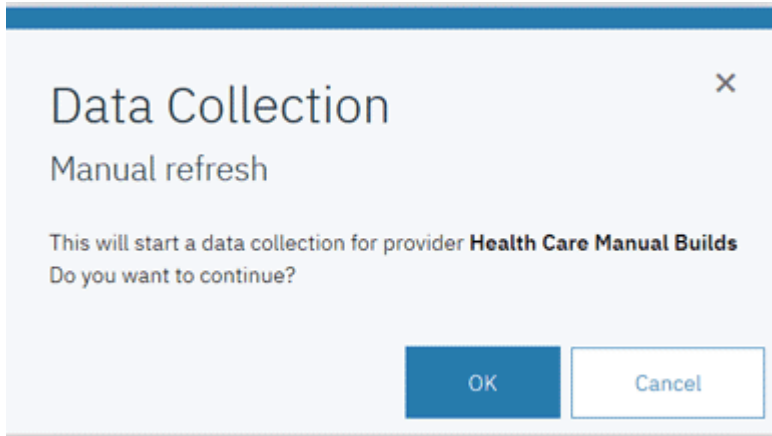
13. Click **OK** to start the data collection.

14. Click the **Options Menu** icon on the upper right corner of the **Health Care Manual Builds** card. An options menu appears.
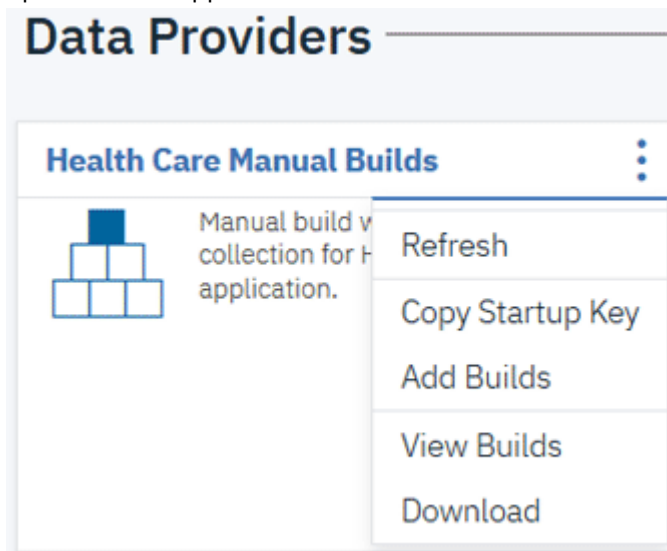
15. Select **View Builds** to view the list of builds. You can now see the information bar indicates that data is loaded to the warehouse. Code Coverage for this build now shows that the code coverage percentage of this build is 66%.

16. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
17. Click **Workbooks** to go to the Workbooks page.

On the Workbooks page, you can see the code coverage results for **zMobile Health Care** is 66%. You can click the name of the **zMobile Health Care** workbook card to view detailed analysis as you did in the "Setting up Manual Builds for Code Coverage Analysis" on page 17 tutorial.
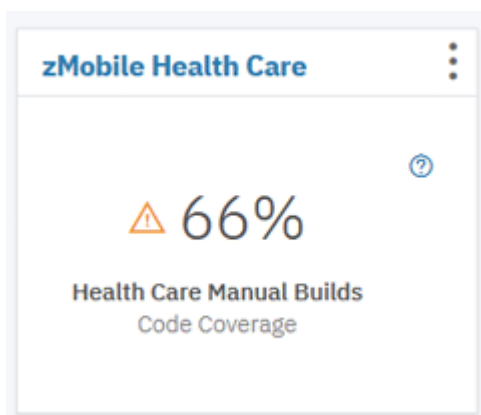
You have now explored how to add a new Manual Builds data provider to collect the code coverage data automatically. This scenario reduces the manual steps you have to perform in order for ADI to perform analysis of code coverage data. As mentioned in the beginning of "Setting up automated code coverage data collections" on page 41, ADI 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 ADI capability of automated data collection from an RTC build and already have an RTC server hosted with code coverage data, you can find more information in this online help in the Chapter of Chapter 7, "Managing connections, data providers, workbooks, and notifications," on page 155.

Exercising setting up a Manual Builds for code coverage analysis

This is a required exercise as prerequisite for the Analyzing code coverage tutorial. You need to perform the set up of a Manual Builds data provider and an Analysis workbook to analyze code coverage for Perfect Calculator application.

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

• Installing and setting up ADI

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/adi/web/projects/HealthCare4All`.

2. Log in as the ADI administrator with `AdiAdmin` as the user ID and `AdiAdmin` as the password. The Welcome page appears.

3. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

4. Click **Data Providers** to go to the **Data Providers** page.

   On the Data Provider page, a warning message shows that no data providers have been specified yet.

5. Click **Create Data Provider**. The Add Data Provider page loads with choices of available data providers.

6. Click the **Manual Builds** data provider card. A form for creating the Manual Builds data provider appears.

7. Complete the form with the following information.

   a. **First Build**

      1) **Build Name**: 2015Nov10Build

      2) **Date of Build**: 11/10/2015, 12:00 AM . You can select the date from the drop-down calendar to fill in the Date of Build.

      3) **Description**: November 10, 2015 build for Perfect Calculator application.

First Build

Build Name *

2015Nov10Build

Date Of Build *

11/10/2015, 12:00 AM

Description

November 10, 2015 build for Perfect Calculator application.

b. **Code Coverage Files**

1) Click the **Add** icon (⊕) next to the **Code Coverage Files** header.

2) Browse to the `installed folder>adi>examples>cobol-coverage>Perfcalc App> November 10 2015` folder.

3) Select all the zip files in the **November 10 2015** folder, that is , `Perfcalct1.zip`, `Perfcalct2.zip`, `Perfcalct3.zip`, and `Perfcalct4.zip`.

4) Click the **Open** to add the files.

Code Coverage Files ⊕

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

Filename	Description
Perfcalct1.zip	
Perfcalct2.zip	
Perfcalct3.zip	
Perfcalct4.zip	

5) **Data Provider Details**

  a) **Name**: Perfect Calculator Manual Builds

  b) **Description**: Manual builds for Perfect Calculator application

  c) **Enable headless collection support**: Make sure that the box is unchecked.

  d) Select **All Days** for number of days to keep data in the data warehouse.

Data Provider Details

Name *

Perfect Calculator Manual Builds

Description

Manual builds for Perfect Calculator application.

☐ Enable headless collection support

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

All Days ▾

[Add] [Cancel]

8. Click **Add** to create this data provider. The **Data Providers** page appears with the **Perfect Calculator Manual Builds** added.

Perfect Calculator Manual Builds   ⋮

Manual builds for Perfect Calculator application.
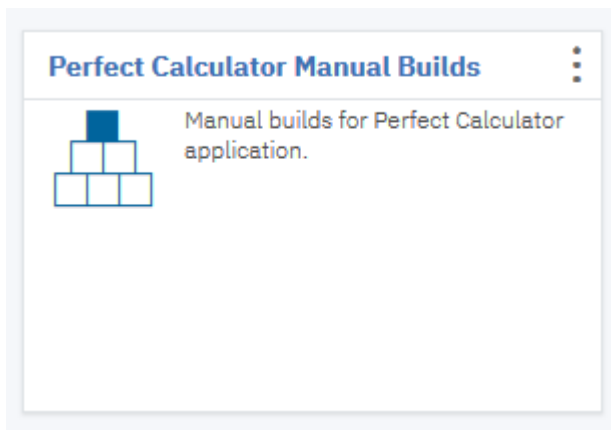
9. Click the **Options Menu** icon on the upper right corner of the **Perfect Calculator Manual Builds** cards. An options menu appears.

Perfect Calculator Manual Builds   ⋮

Manual builds application.

Edit

Delete

Add Builds

View Builds

Download
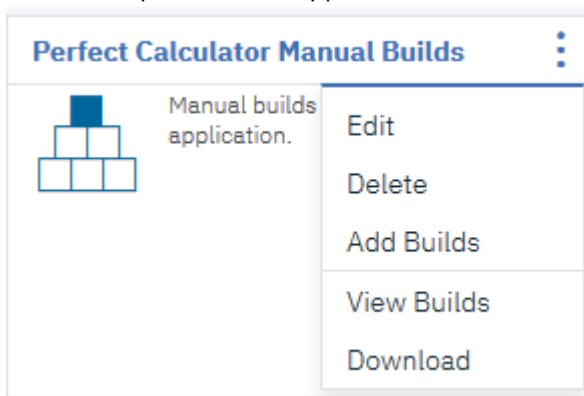
10. Select **Add Builds** to add another build. The Add Build form appears.

11. On the **Add Build** form, fill in the following information.

a. **Build Name**: 2015Nov11Build

b. **Date of Build**: 11/11/2015, 12:00 AM. You can select the date from the drop down calendar to fill in the Date of Build.

c. **Description**: November 11, 2015 build for Perfect Calculator application.

d. **Code Coverage Files**:

1) Click the **Add** icon (⊕) next to the **Code Coverage Files** header.

2) Browse to the `installed folder>adi>examples>cobol-coverage>Perfcalc App>
   November 11 2015` folder.

3) Select all the zip files in the `November 11 2015` folder, that is, `Perfcalct1.zip`,
   `Perfcalct2.zip`, `Perfcalct3.zip`, and `Perfcalct4.zip`.

4) Click **Open** to add the files.

e. Click **Add** to add this build.

12. The Builds page with the list of all builds appears. Click the **Add** icon (⊕) next to the Builds page
    header to add another build.

13. On the **Add Build** form, fill in 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 fill
   in the Date of Build.

c. **Description**: November 12, 2015 build for Perfect Calculator application.

d. **Code Coverage Files**

1) Click the **Add** icon (⊕) next to **Code Coverage Files** header.

2) Browse to the `installed folder>adi>examples>cobol-coverage>Perfcalc App>
   November 12 2015` folder.

3) Select all the zip files in the `November 12 2015` folder, that is, `Perfcalct1.zip`,
   `Perfcalct2.zip`, `Perfcalct3.zip`, `Perfcalct4.zip`, `Perfcalct5.zip`, and
   `Perfcalct6.zip`.

4) Click **Open** to add the files.

e. Click **Add** to add this build.

14. The page with the list of all builds appears. Click the **Add** icon (⊕) next to the Builds page header to
    add another build.

15. On the **Add Build** form, fill in 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 fill
   in the Date of Build.

c. **Description**: November 13, 2015 build for Perfect Calculator application.

d. Code Coverage Files:

1) Click the **Add** icon (⊕) next to the **Code Coverage Files** header.

2) Browse to the `installed folder>adi>examples>cobol-coverage>Perfcalc App>
   November 13 2015` folder.

3) Select all the zip files in the `November 13 2015` folder, that is, that is, `Perfcalct1.zip`,
   `Perfcalct2.zip`, `Perfcalct3.zip`, `Perfcalct4.zip`, `Perfcalct5.zip`, and
   `Perfcalct6.zip`.

4) Click **Open** to add the files.

e. Click **Add** to add this build.

16. The Builds page with the list of all builds appears. Click the **Add** icon (⊕) next to the Builds page header to add another build.

17. On the **Add Build** form, fill in 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 fill in the Date of Build.

   c. **Description**: November 17, 2015 build for Perfect Calculator application.

   d. Code Coverage Files:

      1) Click the **Add** icon (⊕) next to the **Code Coverage Files** header section.

      2) Browse to the `installed folder>adi>examples>cobol-coverage>Perfcalc App> November 17 2015` folder.

      3) Select all the zip files in the `November 17 2015` folder, that is, `Perfcalct1.zip`, `Perfcalct2.zip`, `Perfcalct3.zip`, `Perfcalct4.zip`, `Perfcalct5.zip`, and `Perfcalct6.zip`

      4) Click **Open** to add the files.

   e. Click **Add** to add this build.

   After adding all the builds, you should see 5 builds on the Builds page.

Name	Date of Build	Code Coverage	Action
2015Nov17Build	Nov 17, 2015, 12:00:00 AM	75%	✎ ✓ ⊖
✓ Dec 1, 2017, 1:30:09.578 PM - CRIDA0066I Added 22 new rows of data to the warehouse for automation result 2015Nov17Build.			
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%	✎ ✓ ⊖

18. Click the **Back** icon on the Builds page header to go back to the Data Providers page.

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. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Select the **Test** role and click **Continue**. The **Workbook** page appears.

The Workbooks page appears with a message that no workbooks have been specified at this point.

3. Click **Create Workbook** and then select **Analysis** form the drop-down list.

The **Add Analysis Workbook** page appears.

4. On the **Add Analysis Workbook** page, fill in the following information.

   a. **Name**: Perfect Calculator

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

   c. **Contributor List**: Select **Tammy** as Owner and **Trish** as Member.

   Note: By default, the user who is creating a workbook will be automatically added as the owner of the workbook.

d. **Provider**: Select Perfect Calculator Manual Builds. The **Code Coverage Data Provider Settings** section appears.

- **Code Coverage Threshold**: Slide the left slider to 50% and the right slider to 80%.

5. Click **Add** to add this workbook. The Workbooks page loads with **Perfect Calculator** workbook added.
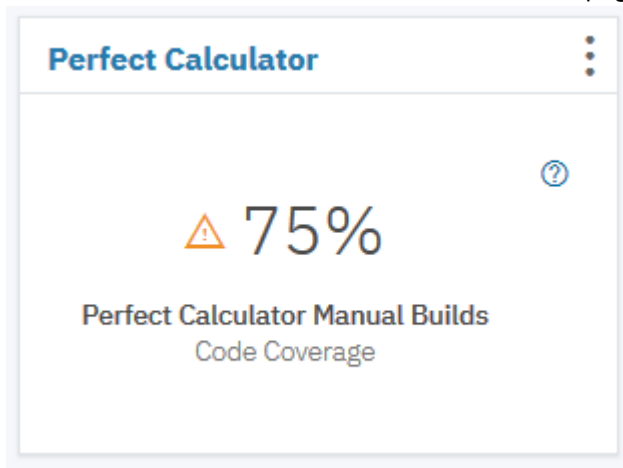
You have completed the setup of the Perfect Calculator Analysis workbook. You are now ready to analyze code coverage data for Perfect Calculator application. For more information, see "Analyzing code coverage" on page 60.

Analyzing code coverage

This tutorial guides you through how to analyze code coverage results with IBM ADI.

In this tutorial, you as Tammy, the test manager examines the code coverage analysis reports and dashboard provided in the ADI by performing the following activities.

- Reviewing the Workbooks page to see overall application code coverage status
- Deep driving to code coverage of application that having problem
- Comparing code coverage status of current build and previous build to identify the problem

Prerequisite

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

- "Installing and setting up ADI" on page 5
- "Setting up Manual Builds for Code Coverage Analysis" on page 17
- "Exercising setting up a Manual Builds for code coverage analysis" on page 52

Reviewing code coverage reports and dashboards

This lab guides you how to review code coverage reports and dashboards as Tammy, the test manager.

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

1. Navigate your browser to `https://healthcare.example.com:9753/adi/web/projects/HealthCare4All`.

2. Log in as the tester manager with Tammy as the user ID and Tammy as the password. The Workbook page appears. For this tutorial, you will see the code coverage report for **Perfect Calculator** and **SAM Application**.
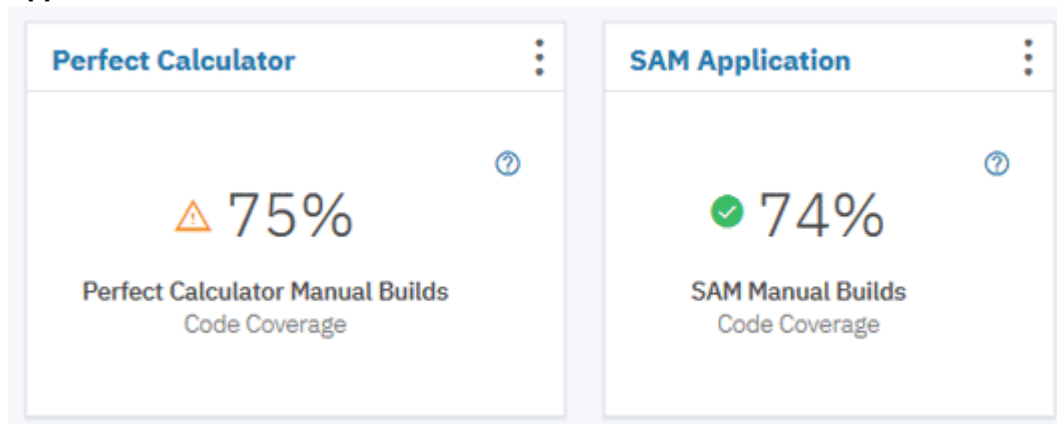
You can see that both applications have the **Aggregated Code Coverage** over 70% but Perfect Calculator is in warning state while SAM Application is in normal state. The code coverage report is a roll-up percentage of code coverage from all builds within one workbook. ADI allows users to define the code coverage threshold for a workbook to customize based on the quality level that application required.

For example in this lab, the code coverage for **Perfect Calculator** which is lower than 80% will be shown in yellow to indicate the warning status and the code coverage which is lower than 50% will be shown in red to indicate that it is not in acceptable number. For the **SAM Application**, the **Aggregated Code Coverage** which is lower than 70% will be shown in yellow.

These threshold values can be set when you create the workbook. See "Setting up Manual Builds for Code Coverage Analysis" on page 17 how to set up the threshold values.

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.
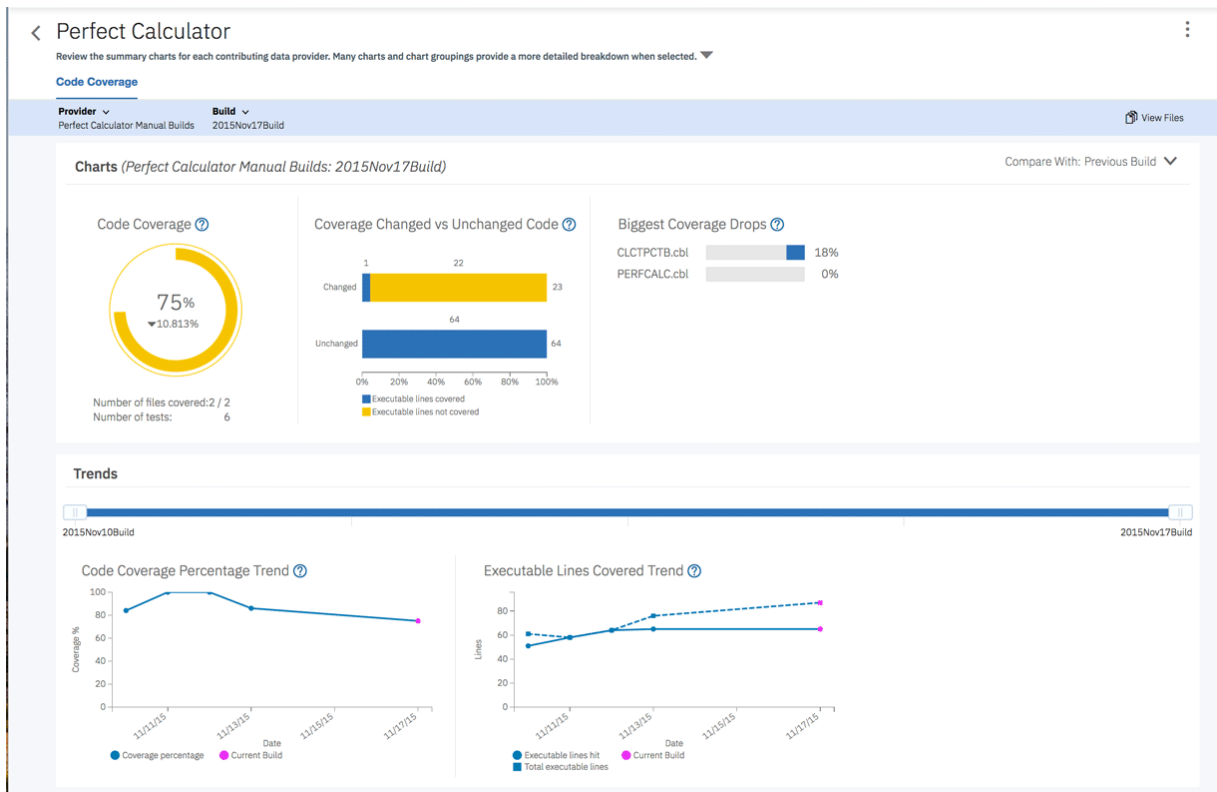
The reports show the code coverage analysis for the latest build of **Perfect Calculator Manual Builds**. The first report shows that the **Perfect Calculator** has 75% Code Coverage which is in the warning area. The code coverage is dropped from the previous build by 10.813%. From the next report, you can see that 23 executable lines of code are modified on this build but only 1 line 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%.

You can notice that CLCTPCTB.cbl status is in red warning and the percentage of code coverage change goes down by 18.286%. 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 18.286%.

Note: The "Change%" of executable lines is the percentage of executable lines that are changed within 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. The flowpoints analysis section displays 3 flowpoints but there are total of 5 flowpoints within the file.

Item Name ⑦	Warnings ⑦	Code Coverage ⑦	Change % ⑦	Covered ⑦	Total ⑦	Added ⑦	Updated ⑦	Change % ⑦	Deleted ⑦	Historical ⑦	Minimal ⑦
⌄ CLCTPCTB.cbl	⚠	46%	⌄ 18%	18	39	11	4	⌃ 38%	0	3	1
COMBINE-AND-WRITE	⚠	0%	0%	0	16	11	4	⌃ 94%	0		
FT-100-CALCULATE-TOP-VALUES	⚠	0%	0%	0	0	0	0	0%	0		
P-OUTPUT-AZUE0000	⚠	0%	0%	0	5	0	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 appears.

Flowpoint Analysis   File CLCTPCTB.cbl ⌄   Filters ⌄

Item Name▲ ⑦	Code Coverage ⑦	Change % ⑦	Executable Lines ⑦						Tests ⑦	
			Covered ⑦	Total ⑦	Added ⑦	Updated ⑦	Change % ⑦	Deleted ⑦	Historical ⑦	Minimal ⑦
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
COMBINE-AND-WRITE	0%	0%	0	16	11	4	⌃ 94%	0	0	0
FT-100-CALCULATE-TOP-VALUES	0%	0%	0	0	0	0	0%	0	0	0
P-OUTPUT-AZUE0000	0%	0%	0	5	0	0	0%	0	0	0

Similar to the **Files 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 15 executable lines from the total of 16 executable lines are either added or updated. This confirms Tammy suspicion that the executable lines that are modified within the `CLCTPCTB.cbl` are not tested.

8. Select the **Filters** dropdown list on the top menu and check **Changed Flowpoints Only** option to filter the view to show only the flowpoints that are modified.

Filters ⌄

- Changed Flowpoints Only ☑
- Line Changes Percentage ☐
- Code Coverage Percentage ☐
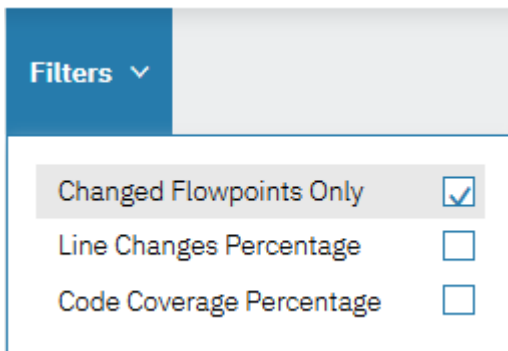
9. The view updates with only **COMBINE-AND-WRITE** flowpoint which confirms that this is the only flowpoint that is modified and it has not yet been tested.

Flowpoint Analysis   File CLCTPCTB.cbl ⌄   Filters ⌄

✕ Changed Flowpoints Only

Item Name▲ ⑦	Code Coverage ⑦	Change % ⑦	Executable Lines ⑦						Tests ⑦	
			Covered ⑦	Total ⑦	Added ⑦	Updated ⑦	Change % ⑦	Deleted ⑦	Historical ⑦	Minimal ⑦
COMBINE-AND-WRITE	0%	0%	0	16	11	4	⌃ 94%	0	0	0

Showing 1-1 First Page | Previous Page | Next Page

10. Click the **Back** icon on the top menu to go back to the **Build Analysis** view for **2015Nov17Build** page.

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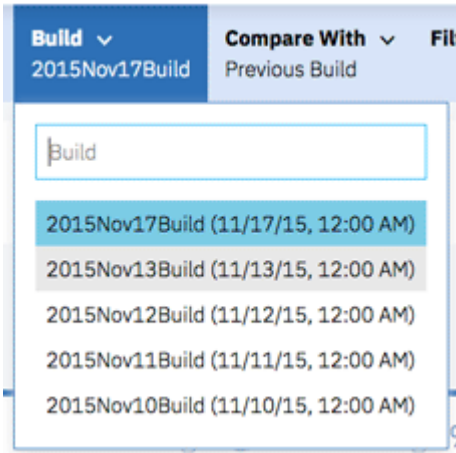
On the page for **2015Nov13Build**, you can see that the overall code coverage is dropped by 14.5% comparing with the previous build. For **2015Nov13Build**, the `CLCTPCTB.cbl` status is also in warning status for this build and the percentage of code coverage change goes down by 36% comparing with the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

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

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**. Tammy wants to compare the builds to get a better understanding of what has changed.

13. Click the **File Trends** on the top menu.
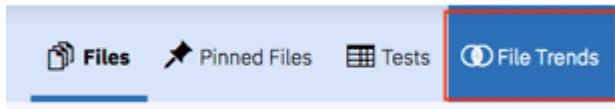
The **File Trends** page appears. By default, **2015Nov12Build** is preselected.

14. Select **2015Nov13Build** and **2015Nov17Build**.
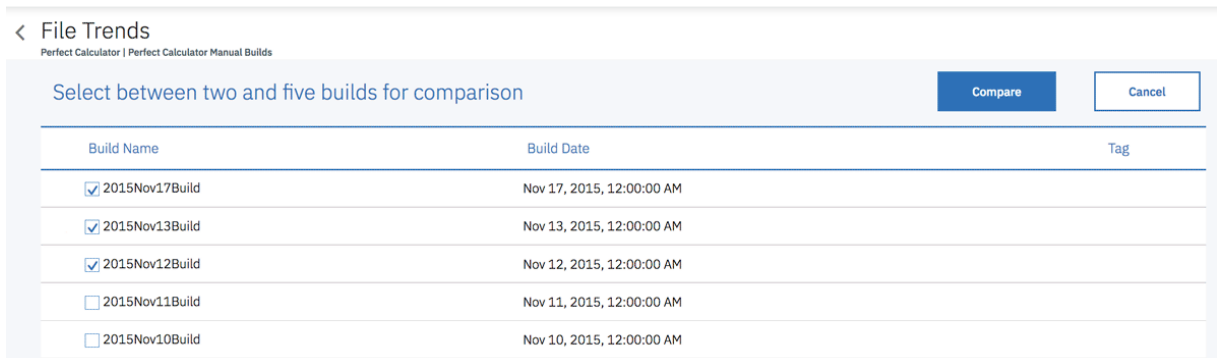
15. Click **Compare** to compare the three builds. The page of code coverage details for three builds appears.

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.cbl 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.clb** in the past 2 builds have not been properly tested. You can also see that the code coverage percentage of PERFCALC.cbl also drops 2% on 2015Nov13Build. There are 2 execution lines added but it seems not tested either.

16. Click the **Expand** icon in front of CLCTPCTB.cbl from all three builds to show the flowpoints analysis.

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.

Now as Tammy, you are satisfied with your investigation on the code coverage issue for Perfect Calculator application. You communicates the results to your team and ask Trish to update 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. Open the menu pane and select **Data Providers** to go to **Data Providers** page.
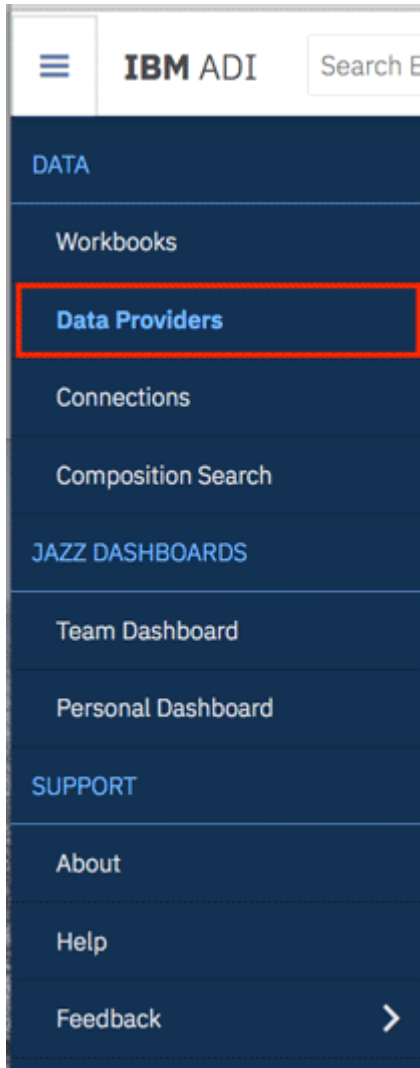
18. On the Perfect Calculator Manual Builds data provider, click the **Options Menu** icon and then select **View Builds**. The **Builds** page appears.
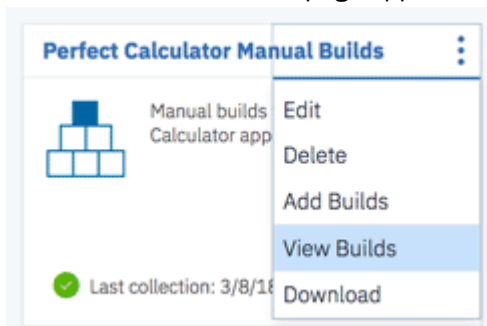
19. Select the **Star** icon in front of the **2015Nov12Build**. The **Star** icon in front of a build name is the indication of a baseline build.

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

21. Click **Start data collection** on the window for ADI 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 see a new notification message on the **Notifications** icon.

22. Click **Notifications** icon to view the data collection status.

23. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

24. Select **Perfect Calculator** workbook to view the code coverage reports of Perfect Calculator.

25. 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, ADI also calculates these information against the baseline builds that you define.

26. 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.
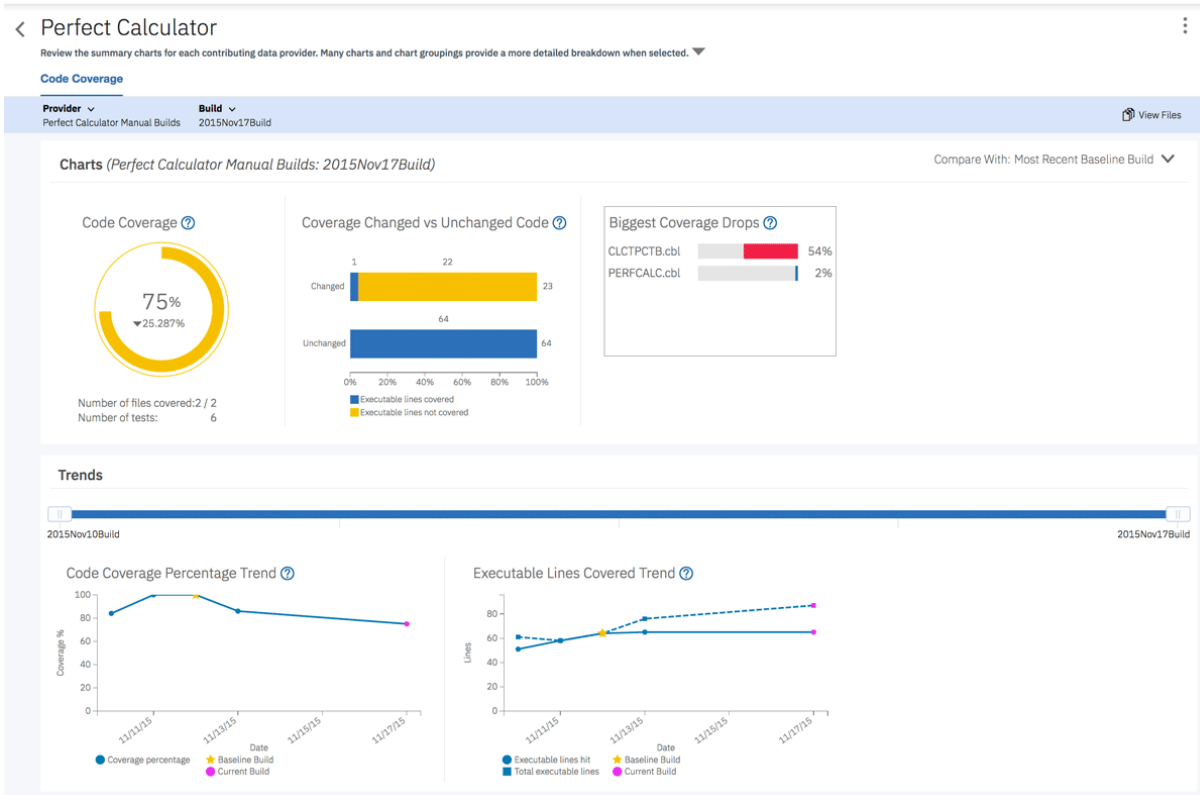
27. Select **View Files** icon on the upper right to go to the **Build Analysis** view.

28. Observe the information against the most recent baseline build on the **Build Analysis** page.
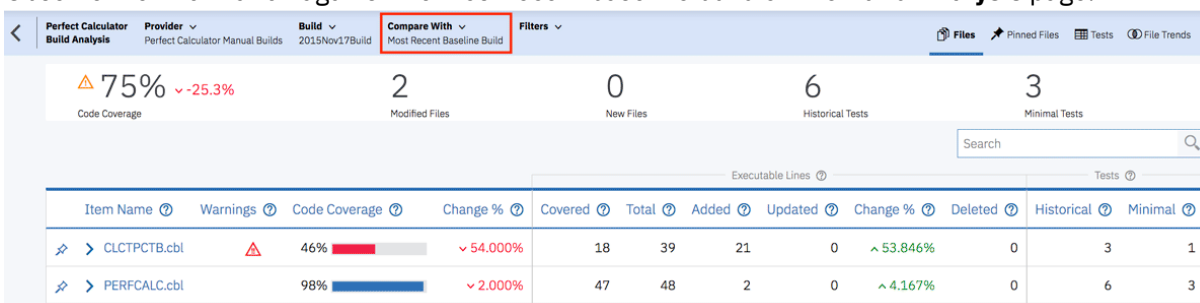
Now you are satisfied with your investigation. You suggest your team using ADI to always check the code coverage reports and check against the baseline build to ensure that the application is fully tested.

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.

Chapter 4. Installation and setup

Several ways are available to install and set up ADI. You can install and set up ADI by using web-based express setup, by using web-based custom setup, by using repotools command, or by using ADDI installer and adi-setup application. You can also install ADI into an existing IBM Rational Collaborative Lifecycle Management (CLM) solution. If you use a previous release of ADI and want to install a new version, you can migrate all the data from the old version to the new one.

Hardware and software requirements

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

A 64-bit operating system (Windows or Linux®) and a minimum of 8 GB server memory provide the best environment for running the Jazz Team Server (JTS).

Note: When you install IBM ADDI with both ADI and AD solutions, you need to combine the minimum hardware and software requirements from both products. For specific details about AD, see AD Installation Prerequisites.

A complete list of system requirements is as follows.

System requirements for ADI 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 2008 R2 Enterprise Edition, x86-64
  - Windows Server 2008 R2 Standard Edition, 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

Application servers

- Embedded WebSphere® Application Server (Liberty Profile 8.5.5)

Databases

- Embedded Apache Derby 10.10 (limited to 10 end users)
- DB2® Workgroup Server Edition, 11.1.1.1 and later versions

    Note: DB2 for z/OS® is not supported.

System requirements for ADI client

Browsers

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

For a complete list of system requirements for any particular version, see JTS online help or the System requirements section on the Jazz.net Deployment wiki. For ADI installation requirements on the Linux system, see Planing to install on Linux systems and ensure that your hardware and software meets all the requirements.

A list of software versions that are supported by ADI is as follows.

Code coverage analysis

- IBM Rational Developer for z Systems (RDz) 9.5.1 and IBM Developer for z Systems (IDz) 14.0.0
- IBM Rational Team Concert (RTC) 6.0 and later versions
- COBOL 4.x and later versions
- PL/I 4.2 and later versions
- IBM Debug Tool for z/OS 13.1 and later versions

OMEGAMON for CICS data provider

- IBM Tivoli OMEGAMON for CICS on z/OS 5.3.0

    Note: It is required to configure ITM TEPS REST API in OMEGAMON for CICS to be a data provider for ADI. For more information, see "Prerequisites for setting up OMEGAMON for CICS for ADI" on page 70.

To check the detailed system requirements for a specific version of ADI, go to the Software Product Compatibility Reports site at https://www.ibm.com/software/reports/compatibility/clarity/index.html

Prerequisites for setting up OMEGAMON for CICS for ADI

You can find the prerequisites and instructions on how to set up OMEGAMON for CICS for ADI.

Perform the following steps on all LPARs that participate in a CICSplex that you wish to monitor with ADI.

1. Install IBM Tivoli Monitoring 6.3.0 Fix Pack 6 or later versions. You can download it from IBM Support.
2. Install OMEGAMON XE for CICS on z/OS 5.3.0 or later versions.

    OMEGAMON XE for CICS on z/OS V5.3.0 must be installed with PTF UA80018 applied. If the self-describing agent (SDA) feature is enabled on the IBM Tivoli Monitoring components then the associated product support files for TEPS server will be installed automatically. For more information, see Self-describing monitoring agents.

    If SDA has been disabled, OMEGAMON XE for CICS on z/OS V5.3.0 must be installed on the TEPS server. You can access it from IBM Support.
3. Enable the dashboard data provider configuration option for the Tivoli Enterprise Portal Server.

    Verify that dashboard data provider configuration option for TEPS server has been enabled. This can be done by using Manage Tivoli Enterprise Monitoring Services utility or through command line. For more information, see Verifying the dashboard data provider is enabled.
4. Get appropriate permissions.

    You need to get permissions to see the data. Without permissions "/ibm/tivoli/rest/providers" will return empty.

5. If a firewall exists between the TEPS and the ADI server, you need to update the firewall settings appropriately.

   If TEPS server is running behind a firewall, you need to open appropriate ports in the firewall or use proxy.

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

IBM Application Delivery Intelligence for IBM Z only support HTTPS connection. You should always use Secure Sockets Layer (SSL) HTTPS connections to browse to the application from browser. SSL encrypts all data that is passed over an HTTPS connection. During installation, ADI security is enabled by default and all security is provided through the Jazz Team Server.

Enabling secure communication between multiple applications

ADI supports HTTP/HTTPS connections for all data provider connections.

ADI is built on the Jazz Team Server and its users authentication and users management mechanism is provided via the Jazz Team Server. The Jazz Team Server manages an LDAP connection . For more information, see Configure an LDAP or an LDAP/SDBM connection with the WebSphere Liberty server.

Ports, protocols, and services

The default port for ADI is 9753. If you do not want to use the default port numbers, see Changing the port numbers for the application server.

Notes:

- If you deploy ADI V5.1.0.6 or later on the same server as that of ADDI Extension component of ADDI, you cannot use the same port number for both ADI and ADDI Extension.
- If you are on a previous version of ADI and want to migrate to ADI V5.1.0.6 or later, URL and port number cannot be changed from the previous release. The default port of versions before ADI V5.1.0.6 is still 9443.

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 instruction about configuring the certificates, see this topic in the Jazz online help: Configuring security certificates.
- For instructions about configuring the certificate, you can follow the same instructions as described in Configuring security certificates for Report Builder servers.
- For instructions about how to configure a browser for such self-signed certificates, follow the instructions in Importing certificates and configuring browsers for Report Builder reports.

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.

You're required to specify an administrative user ID during ADI installation and setup. You can use the Jazz Team Server administration pages to change user IDs and passwords.

Setting up user roles and access

ADI requires an user ID and password to access ADI services. For more information about creating and managing user accounts, see "Creating and managing users" on page 114.

With the administrator permissions, you can manage users by using the Jazz Team Server administration or ADI administration pages. When a project area is created, users can be designated as members of that project and be assigned role permissions. Only members of project area can have access to that project area. The permission to perform the operations within the project area is controlled by the role permissions assigned to the users. For more information about permissions for each role, see "Assigning role-based permissions to ADI users" on page 115.

Privacy policy considerations

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

Installing ADI by using web-based express setup

The express setup is a simplified version of wizard setup. When you install and set up ADI by using this express setup, you use the Derby database by default and you're not allowed to use any other databases. It is recommended that you use this express setup in a demonstration or product trial but not for production. You can learn how to use this method to install and setup ADI in this topic.

For more information about the express setup, see Jazz online help.

Information required before the installation

You need to get a fully qualified host name and a public URI before you can install ADI application by using the web-based express setup.

The following information is required before the installation:

- A fully qualified host name
- A public URI

Important: A *fully qualified host name* is the host name along with the DNS domain reference of the machine on which the Jazz Team Server (JTS) is installed.

- An example of a *fully qualified host name* is `clmwb.example.com`.
- Examples of an **invalid** *fully qualified host name* include `localhost` and an IP address.

For more information about how to plan the server URI, see Planning your URIs

Starting up the server

To install ADI by using web-based express setup, you need to start up the server first. Unzip the ADI installer package and run commands for the start.

Note: You need to disable the pop-up blocker and add security exceptions before you start the installation.

You need to take the following steps to start up the server.

1. Unzip the ADI installer package.
2. Open terminal/command prompt and navigate to the *unzipped-adi-installer*/adi/server directory by running the following command.

   ```
   cd /unzipped-adi-installer/adi/server
   ```

3. In the current directory, start the server by running the following command.

   ```
   server.startup
   ```

Note: By default when you start the server, the Jazz server is running in the background. You can check the console.log file to see if the server is running. If you want to start the server in the foreground, you can use "-run" option when you start the server.

```
server.startup -run
```

Configuring the server

After you start the server, you need to configure the public URI and set up your administrative user. This topic describes how you configure the server when you install ADI by using web-based express setup.

To configure the server, take the following steps:

1. Open a web browser, go to `https://`*fully qualified hostname*`:9753/jts/setup`, and log on using **ADMIN** as the user ID and **ADMIN** as the password.

2. On the welcome page, select **Express Setup** and click **Next**.

3. On the **Configure Public URI** page:

- Provide a public URI root that is of the form `https://`*`fully qualified hostname`*`:9753/jts/`. For example, you can provide `https://insurance.example.com:9753/jts/` .

  Note: The repository might have an alternative host name through which it can be publicly accessed. When this property is set, this URL is used as the basis of absolute URIs that are made available outside of the client and server. When the URI is set, it cannot be modified.

- Check the box to confirm that when the public URI is set it cannot be modified.

- Click **Test Connection** to test the connection to the server. If the test is successful, click **Next**.

- For more information about URIs, see Planning URIs.

4. On the **Create Administrative User** page:

- Provide information for the administrative user. This is an administrative users for both JTS and ADI application. The ADMIN user account will be automatically disabled after this step. You are required to use this administrative user for future setup. For more information, see User management.

- Click **Next**.

TEAM SERVER & APPLICATION EXPRESS SETUP

✅ Click Next to begin the Express Setup.

Create Administrative User

To begin, create an administrative user for yourself. Once you have completed the form, click "Next" to start the setup.

Create an administrative user for yourself in the Jazz Team Server:

Property	Value	Description
User ID	AdiAdmin	Your user ID (e.g. 'jsmith')
Name	Adi Admin	Your full name (e.g. 'John Smith')
Password	••••••••	Enter a secure password.
Re-type Password	••••••••	
E-mail Address	adiadmin@example.com	Your e-mail address (e.g. 'jsmith@example.com')

[< Back] [Next >] [Finish]

5. On the **Setup Progress** page, wait until the wizard finishes configuring the installation, and click **Next**.

6. On the **Assign Licenses** page, click **Activate Trial** and accept the **terms of the license**. After you activate the trial, a license will be automatically assigned to the administrator you created previously.
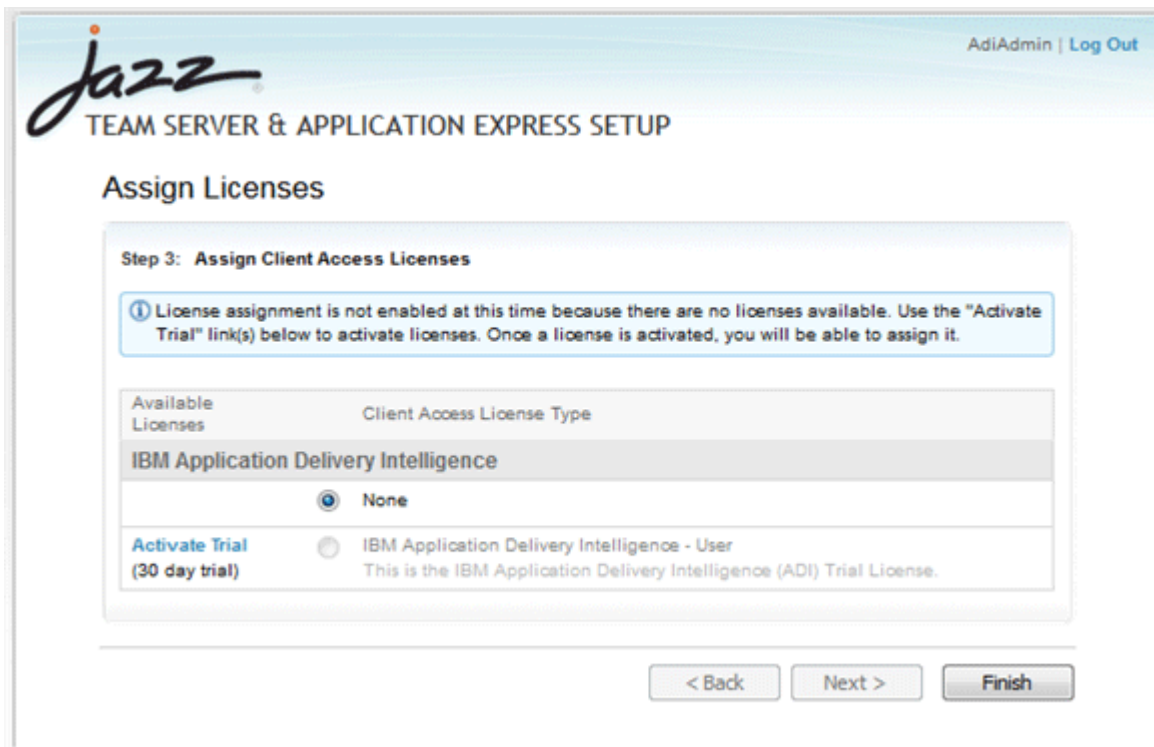
7. Click the **Finish** button to access the Jazz Team Server Administration page where you can perform administrative tasks such as managing the server, users, and licenses. For more information, see Administering Rational solution for Collaborative Lifecycle Management servers.

8. Restart the server by completing the following steps:

   a. "Shutting down the server" on page 79

   b. "Starting up the server" on page 72

Note: The ADI license are *authorized licenses*, which are permanent licenses that are assigned to a single user. The ADI license is managed by the Jazz license server. If you purchase activation kits that contain the activation key file, you can import the activation key to the license server. For more information, see Installing and managing client access license keys.

Setting up the ADI application

This topic describes how you set up the ADI after you install ADI by using the web-based express setup.

Before you begin, ensure that the pop-up blocker is disabled for `https://`*`fully qualified hostname`*`:9753/adi/web` page.

To set up the ADI application, take the following steps:

1. Browse to `https://`*`fully qualified hostname`*`:9753/adi/web` to open the ADI application.

2. Optional: Log on with the administrative user name and password that you created during the server configuration.

3. ADI application appears with the **All projects** page. For the first time user, there is no project area. Click the **project area management** link to set up a project area.

4. The application navigates to the **Application Administration - IBM Application Delivery Intelligence (ADI)** page. Use this page to create a project area.

a. Click the **Create Project Area** link on the top right corner.

b. Provide a **Name**, **Summary**, and **Description** for the project area.

c. Select **IBM Application Delivery Intelligence Application Default Process** as process if it is not yet selected.

d. Assign or create users as members of the project area. For more information, see "Creating and managing users" on page 114.

e. Assign or create users as administrators for the project area.

f. Click **Save** next to the project area name on the top to create the project area.

Note: After you save the project area, a message appears warning you about the members added without roles. You can ignore this warning message because currently no role to members is available.

5. On the upper left corner of window, click the **Home** icon to open the drop-down menu and select **IBM Application Delivery Intelligence (ADI)** . **My Projects** and **All Projects** appear.

6. Click **All projects**. The project area you previously created appears.

7. Click the project area to open it in your browser and then bookmark the page of your project area as the ADI home page.

Note: After you set up IBM ADDI, you need to restart the server. Otherwise, you will not be able to log in with the correct credentials and IBM ADDI will request you for a re-authentication. When you open the log file (`<ADI installation folder>\adi\server\liberty\servers\clm\logs\addi-brd.log`), you can see the warning of "Failed authentication with Jazz server".

To restart the server, complete the following steps:

1. Navigate to the `<unzipped-adi-installer>/adi/server` directory.

2. Shut down the server by running the `server.shutdown` script.

3. Restart the server after the you shut down the server successfully by running the `server.startup` script.

Shutting down the server

After you finish the installation and setup, you can shut down the server.

To shut down the server, complete the following steps:

1. Open terminal/command prompt and navigate to the `unzipped-adi-installer/adi/server` directory by running the following command:

```
cd /unzipped-adi-installer/adi/server
```

2. In the current directory, shut down the server by running the following command:

```
server.shutdown
```

Installing ADI by using web-based custom setup

You can learn how to install ADI by using web-based custom setup and set up the majority of the installation options. If you want to use DB2 database, you should use this method.

The custom setup is recommended for production environments. The wizard guides you through setting up your database, configuring email settings, configuring the user registry, and registering applications. The wizard detects and prompts you to set up the applications you installed.

For more information about the custom setup method, see Jazz online help.

Information required before the installation

You need to get a fully qualified host name and a public URI before you install ADI by using web-based custom setup.

The following information is required before the installation:

- A fully qualified host name
- A public URI

Important: A *fully qualified host name* is the host name along with the DNS domain reference of the machine on which the Jazz Team Server is installed.

- An example of a *fully qualified host name* is `clmwb.example.com`.
- Examples of an **invalid** *fully qualified host name* include `localhost` and an IP address.

For more information about how to plan the server URI, see Planning your URIs

Creating databases

You need to create databases before you set up ADI application. You can learn more about the default Derby database and how to create DB2 database for use.

If you are running against a database other than the default Derby database, which is included in the installation, you need to create a database before you set up the application. Currently, ADI supports only DB2 database. For more information about setting up the database, see Jazz online help.

To set up the ADI application, you need to create three databases for Jazz Team Server, ADI, and Data Warehouse (DW). Use the parameters that are provided in Setting up the database for Jazz Team Server and Data Warehouse. Use the same parameter values that are used for Jazz Team Server when you create database for the ADI application.

Derby database

By default, the builds use the Derby database included in the installation, so you do not need to create any database. This option is only available when you install ADI by using the web-based express setup installation.

DB2 database

Review Prerequisites and make sure that your DB2 database meets the requirements before you continue.

Before you create the DB2 databases for ADI, you must set the `DB2_COMPATIBILITY_VECTOR` registry variable with the following commands. Otherwise, ADI 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 JTS using codeset UTF-8 territory en PAGESIZE 16384
db2 create database ADI using codeset UTF-8 territory en PAGESIZE 16384
db2 create database DW using codeset UTF-8 territory en PAGESIZE 16384
```

Optional: Data Warehouse 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 cannot exceed 256. It is also recommended to set larger secondary log sizes since they are constantly cleared by DB2.

For more information about DB2 installation, see IBM DB2 10.5 for Linux, Unix and Windows documentation.

Starting up the server

To install ADI by using web-based custom setup, you need to start up the server first. Unzip the ADI installer package and run commands for the start.

You need to take the following steps to start up the server.

1. Unzip the ADI installer package.
2. Open terminal/command prompt and navigate to the *unzipped-adi-installer*/adi/server directory by running the following command:

   ```
   cd /unzipped-adi-installer/adi/server
   ```

3. In the current directory, start the server by running the following command:

   ```
   server.startup
   ```

   Note: By default when you start the server, the Jazz server is running in the background. You can check the console.log file to see if the server is running. If you want to start the server in the foreground, you can use "-run" option when you start the server.

   ```
   server.startup -run
   ```

Configuring the server

After you start the server, you need to configure the public URI and set up your administrative user. This topic describes how you configure the server when you install ADI by using web-based custom setup.

To configure the server, take the following steps:

1. Open a web browser, go to `https://localhost:9753/jts/setup`, and log on using **ADMIN** as the user ID and **ADMIN** as the password.

The Jazz Team Server at **localhost** in **Jazz** requires a user ID and password:

User ID:

ADMIN

Password:

•••••

Remember my User ID

Log In

Licensed Material - Property of IBM Corp. © Copyright IBM Corp. and its licensors 2008, 2015. All Rights Reserved. IBM, the IBM logo, Jazz, and Rational are trademarks of IBM Corporation, in the United States, other countries and regions, or both. Built on Eclipse is a trademark of Eclipse Foundation, Inc. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates in the United States, other countries and regions, or both.

IBM.

Rational. software

2. On the welcome page, select the **Custom Setup** option to customize the setup, and click **Next**.
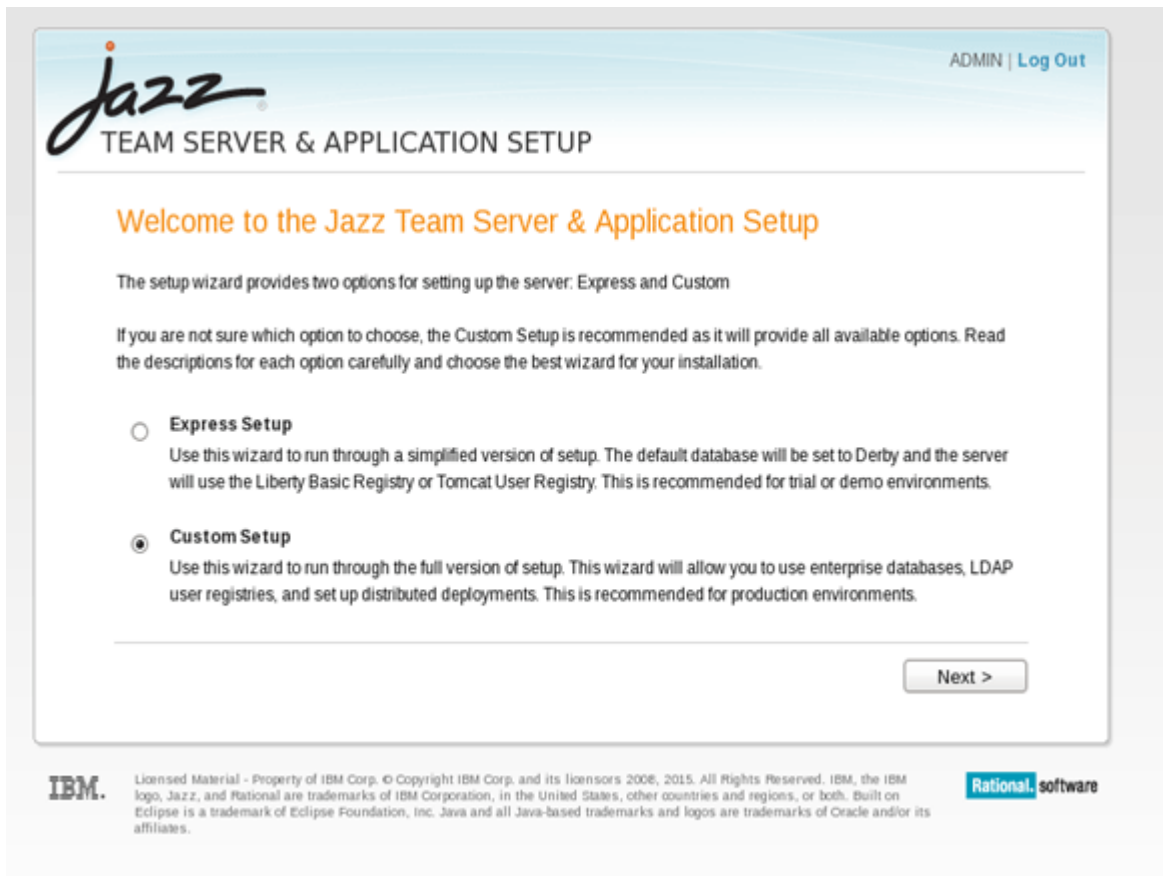
3. On the **Introduction** page, click **Next** to get started with the setup wizard.
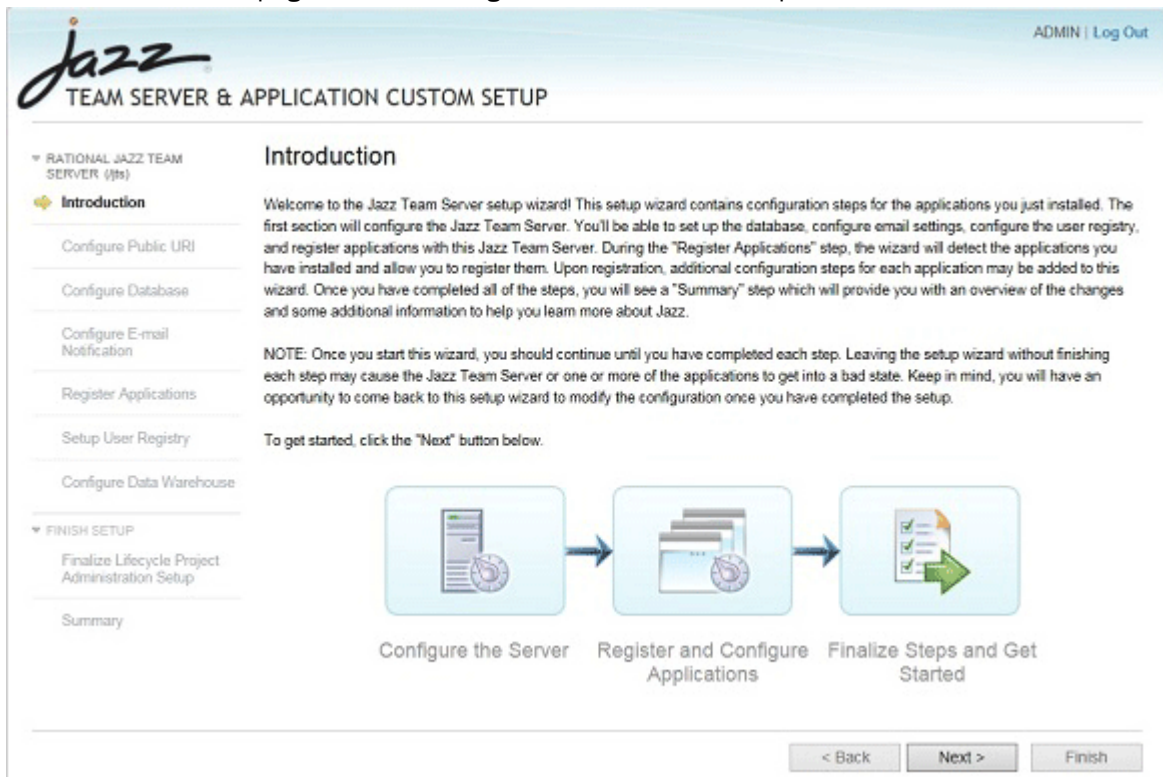
4. On the **Configure Public URI** page:

a. Provide a Public URI in the format of `https://`*`fully qualified hostname`*`:9753/jts/`. For example, `https://example.com:9753/jts/`.

b. Check to confirm that when the Public URI is set it cannot be modified.

c. Click **Test Connection** to test the connection to the server. If the test is successful, click **Next**.

Note: For more information about URIs, see Planning URIs.

5. On the **Configure Database** page, provide information to set up a connection to the Jazz Team Server database.

a. Select the database vendor and connection type. ADI currently supports only DB2 as a database vendor and JDBC as a connection type.

b. Configure the database connection properties.

   1) Insert the password of the JDBC database, that is, the password of your DB2 database.

   2) Provide the JDBC location in the following format.

   ```
   //server where your DB2 is installed:50000/JTS:user=Your DB2 user;password={password};
   ```

c. Click the **Test Connection**. You receive a message that the database exists but is not configured to contain Jazz.

d. Click the **Create Tables**, and wait several minutes for the command processing. When the processing is complete, click **Next**.

6. On the **Enable E-mail Notification** page, disable email notifications and click **Next**.

   Note: There is no email notification capabilities in the current version of ADI.
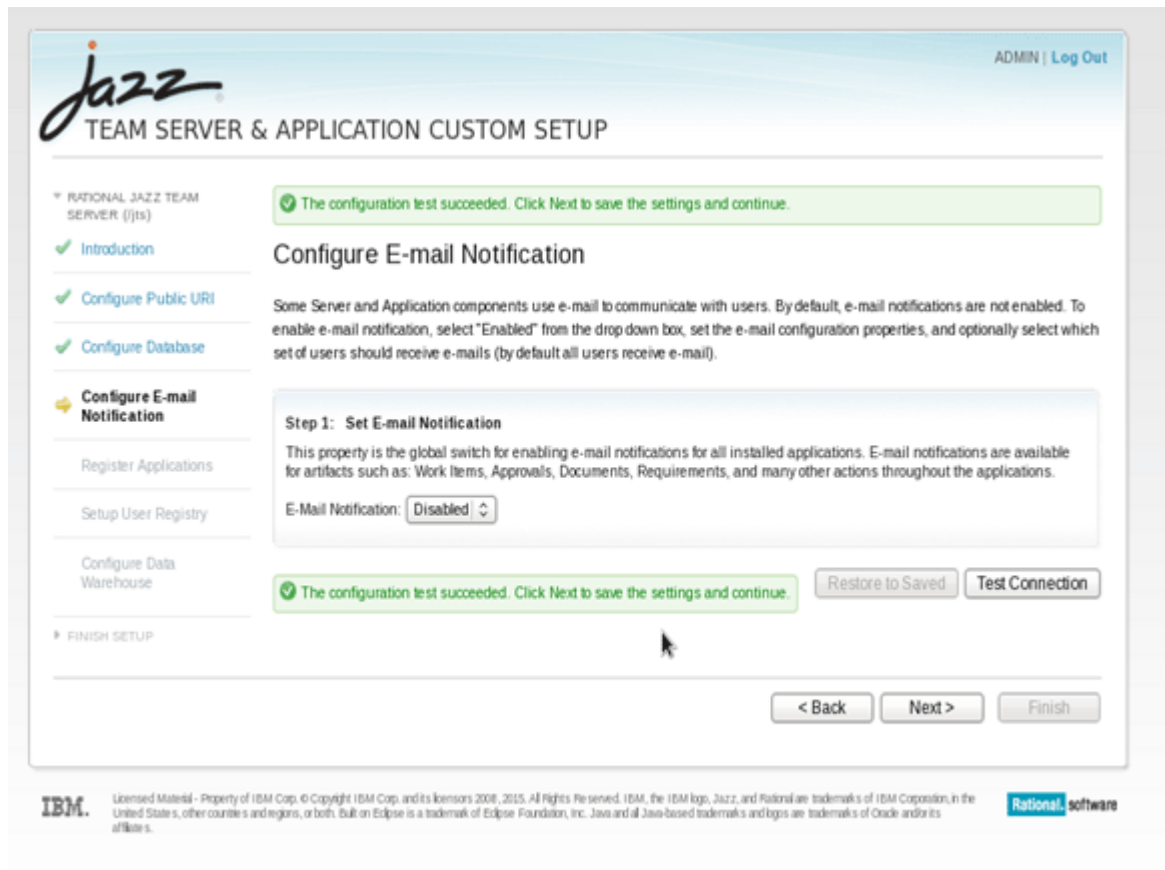
7. On the **Register Applications** page, the wizard detects the applications that were installed along with the Jazz Team Server. When the wizard detects the application, click **Register Applications**, and then click **Next**.

✅ Applications were successfully registered with this Jazz Team Server. Click Next to continue.

Register Applications

Use this step to register installed applications with the Jazz Team Server. For installations where all applications are on the same host, you can accept the defaults and continue.

Step 1: Verify and Update Discovered Applications

The following applications were detected on the local file system. The Application Instance name will appear in the user interface. An instance name proposal was generated based on the context root of the application. You can edit the proposal. The name should be unique on this Jazz Team Server and meaningful within your organization. The Discovery URL has been defaulted based on the public URI of this server and the context root of the application. A default functional user ID has been computed for each application. Associating a functional user ID allows applications to make requests within background activities, such as sending e-mail notifications, without user intervention. You can manually add a new application (for example, an application installed on a remote server), or choose not to register an application by clicking the "X" icon.
To add a new application enter its Discovery URL, which is usually in the following form, https://hostname.domain:port /contextRoot/scr, for example https://myserver.ibm.com:9443/qm/scr.

	Application Instance	Application Type	Discovery URL	Functional User ID	
✅	/adi	IBM Application Delivery Intelligence (ADI)	https://localhost:9443/adi/scr	adi_user	✕

Add Application...

Step 2: Register Applications

Once you have verified the applications and their URLs in the previous step, you should register the applications with this Jazz Team Server. As part of the registration process, the server will attempt to contact each application and send configuration data to it. This operation may take a few minutes to complete.

[Register Applications]   ✔ Registration Complete

8. On the **Create Administrative User** page:

   - Choose one of the options for your user registry. Jazz Team Server relies on a user registry to provide basic information about users and groups. For more information, see User management. Currently, ADI supports only IBM® WebSphere® Application Server Liberty Core. You can manage WebSphere Liberty server users by using the default basic user registry or Lightweight Directory Access Protocol (LDAP). If you need more information to create an LDAP connection, refer to this guide.

Property	value	Description
User ID	user	Your user ID (e.g. 'jsmith')
Name	user	Your full name (e.g. 'John Smith')
Password	••••	Enter a secure password.
Re-type Password	••••	Re-type your password to help prevent typos.
E-mail Address	user@mail.com	Your e-mail address (e.g. 'jsmith@example.com')

The default ADMIN user is defined in Jazz to allow you to log into a new server, configure the server, and create an initial JazzAdmins user. For enhanced server security we recommend that you disable default ADMIN access at this time.
☑ Disable default ADMIN access

   - Click **Activate Trial** and accept **the terms of the license**.
   - Click **Next**.

9. On the **Configure Data Warehouse** page, provide information to set up a connection to the Date Warehouse database. If you use Derby database, you do not need to change the default values, and you are not prompted to create database tables.

   - Select the database vendor and connection type. ADI currently supports only DB2 as a database vendor and JDBC as a connection type.

- Insert the password of your DB2 database.
- Provide the JDBC location in the following format.

```
//server where your DB2 is installed:50000/DW:user=Your DB2 user;password={password};
```


- Click **Test Connection**. You receive a message that the database exists but it's not configured to contain the data warehouse.
- Click the **Create Tables**, and wait several minutes for the command processing. When the processing is complete, click **Next**.

10. On the **Configure Data Warehouse** page, provide information to set up a connection to the ADI Database. If you uses the Derby database, you do not need to change the default values, and you are not prompted to create database tables.

- Select the database vendor and connection type. ADI currently supports only DB2 as a database vendor and JDBC as a connection type.
- Insert the password of your DB2 database.
- Provide the JDBC location in the following format.

```
//server where your DB2 is installed:50000/ADI:user=Your DB2 user;password={password};
```

- Click **Test Connection**. You receive a message that the database exists, but it is not configured to contain Jazz.
- Click **Create Tables**, and wait several minutes for the command to finish. When the command finishes, click **Next**.
- On the **Finalize Application** page, click **Finalize Application Setup**.
- When it finishes, click **Next**.

11. On the **Configure Data Warehouse** page, the wizard remembers the configuration.

   - Provide the DB2 password that you use for the DW database.
   - Click **Test Connection**.
   - Ignore the warning that you receive and click **Next**.

12. On the **Finalize Lifecycle Project Administration Setup** page, click **Finalize Application Setup**.

   - When it finishes, click **Next**.

13. On the **Summary** page, view the summary of all your configurations. Click **Finish** to access the Jazz Team Server Administration page where you can perform administration tasks such as managing the server, users, and licenses. For more information, see Administering Rational solution for Collaborative Lifecycle Management servers.

14. Restart the server by completing the following steps:

   a. "Shutting down the server" on page 91
   b. "Starting up the server" on page 81

Note: The ADI license are *authorized licenses*, which are permanent licenses that are assigned to a single user. The ADI license is managed by the Jazz license server. If you purchase activation kits that contain the activation key file, you can import the activation key to the license server. For more information, see Installing and managing client access license keys.

Setting up ADI application

This topic describes how you set up the IBM Application Delivery Intelligence for IBM Z (IBM ADI) application by using web-based custom setup.

Before you begin, ensure that the pop-up blocker is disabled for `https://fully qualified hostname:9753/adi/web` page.

To set up ADI application, take the following steps:

1. Browse to `https://fully qualified hostname:9753/adi/web` to open the ADI application.
2. Optional: Log on with the administrative user name and password that you created during the server configuration.
3. ADI application appears with the **All projects** page. For the first time user, there is no project area. Click the **project area management** link to set up a project area.

4. The application navigates to the **Application Administration - IBM Application Delivery Intelligence (ADI)** page. Use this page to create a project area.

    a. Click the **Create Project Area** link on the top right corner.

    b. Provide a **Name**, **Summary**, and **Description** for the project area.

    c. Select **IBM Application Delivery Intelligence Application Default Process** as process if it is not yet selected.

    d. Assign or create users as members of the project area. For more information, see "Creating and managing users" on page 114.

    e. Assign or create users as administrators for the project area.

    f. Click **Save** next to the project area name on the top to create the project area.

    Note: After you save the project area, a message appears warning you about the members added without roles. You can ignore this warning message because currently no role to members is available.

5. On the upper left corner of window, click the **Home** icon to open the drop-down menu and select **IBM Application Delivery Intelligence (ADI)** . **My Projects** and **All Projects** appear.
6. Click **All projects**. The project area you previously created appears.
7. Click the project area to open it in your browser and then bookmark the page of your project area as the ADI home page.

Note: After you set up IBM ADDI, you need to restart the server. Otherwise, you will not be able to log in with the correct credentials and IBM ADDI will request you for a re-authentication. When you open the log file (`<ADI installation folder>\adi\server\liberty\servers\clm\logs\addi-brd.log`), you can see the warning of "Failed authentication with Jazz server".

To restart the server, complete the following steps:

1. Navigate to the *<unzipped-adi-installer>*/adi/server directory.
2. Shut down the server by running the server.shutdown script.
3. Restart the server after the you shut down the server successfully by running the server.startup script.

Shutting down the server

After you finish the installation and setup, you can shut down the server.

To shut down the server, complete the following steps:

1. Open terminal/command prompt and navigate to the *unzipped-adi-installer*/adi/server directory by running the following command:

   ```
   cd /unzipped-adi-installer/adi/server
   ```

2. In the current directory, shut down the server by running the following command:

   ```
   server.shutdown
   ```

Installing ADI by using the repotools command

You can learn how to install and set up ADI by using the repotools command. This method sets up the Derby database by default. In this case, you cannot use any other data bases. You can use a repotools command to play back setup options with a pre-packed response file which enables you to set up everything from the command shell. The information that you provide by using repotools command is the same as for the web-based express setup.

Starting up the server

To install ADI by using repotools command, you need to start up the server first. Unzip the ADI installer package and run commands for the start.

You need to take the following steps to start up the server.

1. Unzip the ADI installer package.
2. Open terminal/command prompt and navigate to the *unzipped-adi-installer*/adi/server directory by running the following command:

   ```
   cd /unzipped-adi-installer/adi/server
   ```

3. In the current directory, start the server by running the following command:

   ```
   server.startup
   ```

   Note: By default when you start the server, the Jazz server is running in the background. You can check the console.log file to see if the server is running. If you want to start the server in the foreground, you can use "-run" option when you start the server.

   ```
   server.startup -run
   ```

Running the response file

You need to run the response file before you can successfully install ADI by using repotools command.

To run the response file, take the following steps:

1. Use a command shell to change the directory to *unzipped-adi-installer*/server directory where you can find the response file.
2. Run the following command:

   ```
   repotools-jts -setup repositoryURL=https://server-address:9753/jts adminUserID=ADMIN
   adminPassword=ADMIN parametersFile=./adi-response.properties prompt=missing
   ```

Note: Replace the *server-address* with a fully qualified server machine name.

3. To see the default values setting in the response file, you can find it in /*unzipped-adi-installer*/adi/server/adi-response.properties.

4. During the setup, you are prompted to provide the following information.

   - The repository URL from above again. It might theoretically be different if you are using aliases or proxies.
   - The user name, password, and email address of the admin user you want to create.
   - The license. Select licence 2.

5. When you complete, the command line shows a message to indicate the successful installation.

6. Restart the server by completing the following steps:

   a. "Shutting down the server" on page 93
   b. "Starting up the server" on page 91

Setting up ADI application

This topic describes how you set up the IBM Application Delivery Intelligence for IBM Z (IBM ADI) application after you install ADI by using the repotools command.

Before you begin, ensure that the pop-up blocker is disabled for `https://`*fully qualified hostname*`:9753/adi/web` page.

To set up the ADI application, take the following steps:

1. Browse to `https://`*fully qualified hostname*`:9753/adi/web` to open the ADI application.

2. Optional: Log on with the administrative user name and password that you created during the server configuration.

3. ADI application appears with the **All projects** page. For the first time user, there is no project area. Click the **project area management** link to set up a project area.

4. The application navigates to the **Application Administration - IBM Application Delivery Intelligence (ADI)** page. Use this page to create a project area.

   a. Click the **Create Project Area** link on the top right corner.

   b. Provide a **Name**, **Summary**, and **Description** for the project area.

   c. Select **IBM Application Delivery Intelligence Application Default Process** as process if it is not yet selected.

   d. Assign or create users as member of the project area. For more information, see "Creating and managing users" on page 114.

   e. Assign or create users as administrators for the project area.

f. Click **Save** next to the project area name on the top to create the project area.

   Note: After you save the project area, a message appears warning you about the members added without roles. You can ignore this warning message because currently no role to members is available.

5. On the upper left corner of window, click the **Home** icon to open the drop-down menu and select **IBM Application Delivery Intelligence (ADI)** . **My Projects** and **All Projects** appear.

6. Click **All projects**. The project area you previously created appears.

7. Click the project area to open it in your browser and then bookmark the page of your project area as the ADI home page.

Note: After you set up IBM ADDI, you need to restart the server. Otherwise, you will not be able to log in with the correct credentials and IBM ADDI will request you for a re-authentication. When you open the log file (`<ADI installation folder>\adi\server\liberty\servers\clm\logs\addi-brd.log`), you can see the warning of "Failed authentication with Jazz server".

To restart the server, complete the following steps:

1. Navigate to the `<unzipped-adi-installer>`/adi/server directory.

2. Shut down the server by running the `server.shutdown` script.

3. Restart the server after the you shut down the server successfully by running the `server.startup` script.

Shutting down the server

After you finish the installation and setup, you can shut down the server.

To shut down the server, complete the following steps:

1. Open terminal/command prompt and navigate to the *unzipped-adi-installer*/adi/server directory by running the following command:

   ```
   cd /unzipped-adi-installer/adi/server
   ```

2. In the current directory, shut down the server by running the following command:

   ```
   server.shutdown
   ```

Installing ADI into an existing CLM installation

You can find a roadmap for co-installing ADI with an existing IBM Rational Collaborative Lifecycle Management (CLM) solution, such as an existing Jazz Team Server (JTS) and IBM Rational Team Concert (RTC).

Before you begin

Check out the prerequisites before you install ADI into an existing CLM installation.

- ADI is implemented as a Jazz application so it can be co-installed in the same way as any other Jazz application, such as RTC or Doors Next Generation. For the Jazz Deployment guidelines and the proposed deployment topologies, see Jazz Deployment Guide.

- Decide on the topology that you want to implement. The main decision is to co-install on the same server machine as the JTS or to install it on a separate server machine. The following instructions cover both scenarios.

- The versions of the CLM solution and ADI must match. ADI can be co-installed only with CLM 6.0.2 or later version.

- These instructions work only for a new installation of ADI. If you use a stand-alone version of ADI, you cannot currently move that into the configuration as described in the "Procedures" on page 94. Because the ADI application must remain with the JTS for which it was originally installed. However, those instructions work with an existing installation of CLM and a fresh install when the versions match as listed previously. Another possibility is to install CLM applications such as RTC into an existing ADI

installation, that is, the JTS provided with ADI. In that case, review the CLM online help instructions about adding CLM applications to an existing JTS for more information.

Procedures

After you verify the prerequisites, you can complete steps to install ADI into a CLM installation depending on which application server you are using for your CLM environment.

Installing CLM

1. Use Installation Manager to install CLM 6.0.2 or a later version. If you have installed and set up CLM (perhaps it is a CLM already in use), jump ahead to following section of "Install the ADI application into IBM WebSphere Liberty Profile". and "Installing the ADI application into IBM WebSphere Application Server".

   - For details, see Installing the Rational solution for Collaborative Lifecycle Management by using IBM Installation Manager in the CLM Online help.

2. Run the setup for CLM installation.

   - For information on installing CLM, see CLM Interactive Installation Guide.

Java Security Configuration for CLM version 6.0.4 or earlier

If you install a Java version after April 2017 and use it to run Jazz Team Server, then you must perform the following Java security configuration for CLM version 6.0.4 or earlier versions.

1. Open the `java.security` configuration file in a text editor. You can find the file in `JDK Installed Directory/jre/lib/security/java.security`.

2. Change the following line

   ```
   jdk.jar.disabledAlgorithms=MD2, MD5, RSA keySize < 1024
   ```

   to

   ```
   jdk.jar.disabledAlgorithms=MD2, RSA keySize < 1024
   ```

3. Restart your Jazz Team Server.

Installing the ADI application into IBM WebSphere Liberty Profile

You can install ADI together with JTS installed on the same server machine and in the same installation directory. In this case, ADI runs on the same Liberty application server as CLM and has the same base URL as the JTS and the other locally running applications. As an alternative that offers more scalability, you can install ADI on a different server machine. In this case, ADI has a different base URL than the JTS. Other CLM applications will then most likely also be running on a different server. For details and how you can use a proxy server to unify URLs, see the CLM Online help and Deployment Guides mentioned previously.

Setting up a database

If you use DB2 for your CLM databases, then you need to create the ADI database in the existing DB2. You do not need to create JTS and DW database since they were created during the CLM's setup. For more information, see Installing ADI by using web-based custom Setup.

Alternative 1: Installing ADI on the same server machine

Prerequisite: Assumption is that the CLM server is set up and fully functional. The installation machine has at least the JTS setup and zero or more applications such as RTC. The CLM installation directory is referred to as *clm_install*.

1. Stop the CLM server with the following commands.

```
cd clm_install/JazzTeamServer/server
server.shutdown
```

Note: If the Jazz server is running in the background, you can check the console.log file to see if the Jazz server is running.

2. Extract the ADI distribution in a temporary directory referred to as *adi_unzip* below.
3. Merge some of the ADI installation files into the JTS installation by copying (only) the following files and directories from the *adi_unzip* directory into the corresponding *clm_install* directory.

Table 1. Files and directories involved in merging ADI installation into JTS installation

Copy from *adi_unzip*/adi/	To *clm_install*/**JazzTeamServer**/
cobol-coverage	(copy the entire folder into JazzTeamServer folder)
license/adi	license/
java-coverage	(copy the entire folder into JazzTeamServer folder)
repotools/adi	repostools
server/conf/adi	server/conf/
server/conf/jts/sites/ com.ibm.dimez.dashboard.activation.updatesite	server/conf/jts/sites
server/conf/jts/provision_profiles/dimez-product-activation.ini	server/conf/jts/provision_profiles/
server/headless-CC	server/
server/liberty/clmServerTemplate/apps/adi.war	server/liberty/servers/clm/apps/
server/liberty/clmServerTemplate/apps/ adihelp.war	server/liberty/servers/clm/apps/
server/liberty/clmServerTemplate/apps/ adi.war.zip	server/liberty/servers/clm/apps/
server/liberty/clmServerTemplate/apps/ adihelp.war.zip	server/liberty/servers/clm/apps/
server/codecov.shutdown.*	server/
server/codecov.startup.*	server/
server/repotools-adi.*	server/
server/server.shutdown.*	server/ (replacing the existing file; do file merge to be sure)
server/server.startup.*	server/ (replacing the existing file; do file merge to be sure)
swidtag/ ibm.com_IBM_Application_Delivery_Intelligence -5.0.3.swidtag	swidtag/

Note: If you are installing CLM and ADI at the same time and have not set it up yet, then the *clm_install*/server/liberty/servers directory does not exist, yet. In that case copy the files to *clm_install*/server/liberty/clmServerTemplate. Also, for the following step use the template path for editing the file. All template files will be copied to liberty/server/clm the first time the server starts.

4. Open the file *clm_install*/server/liberty/servers/clm/conf/application.xml in a text editor. Add the following lines at the end, but before this line: `</server>`

```
<application type="war" id="adi" name="adi" location="${server.config.dir}/apps/adi.war"></application>
<application type="war" id="adihelp" name="adihelp" location="${server.config.dir}/apps/adihelp.war"></application>
```

5. Start the CLM server again. Continue with the step of "Registering ADI with JTS".

Alternative 2: Installing ADI on a different server machine

Prerequisite: Assumption is that the CLM server is set up and fully functional. It is running on a different machine than where you want to install ADI. For this installation option, you don't need to shut down the JTS server.

1. Extract the ADI distribution into the directory where you want it installed (for example, /opt/ibm/adi on Linux or C:\ibm\adi on Windows). The location will be referred to as *adi_unzip* below.

2. Execute this command to deploy the Liberty server inside the installation directory:

```
cd adi_unzip/adi/server
server.startup -create
```

3. Delete the following files and directories in *adi_unzip*/adi to remove the JTS from this installation so that you can use the existing JTS running on that other machine:

   • /server/conf/jts
   • /server/liberty/servers/clm/apps/clm*
   • /server/liberty/servers/clm/apps/jts*

4. Start the ADI server with the following commands.

```
cd unzipped-adi-installer/server
server.startup
```

Note: By default when you start the server, the Jazz server is running in the background. You can check the console.log file to see if the server is running. If you want to start the server in the foreground, you can use "-run" option when you start the server.
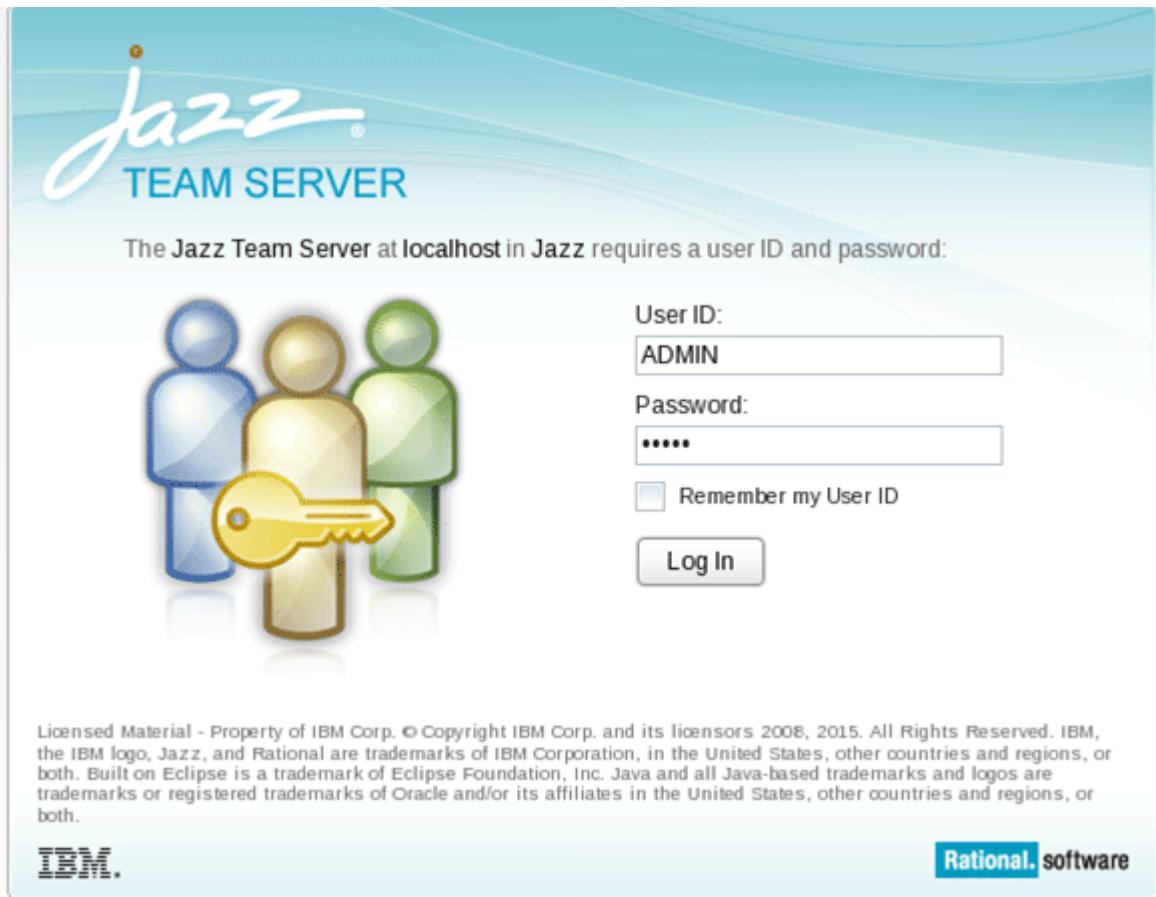
```
server.startup -run
```

5. Continue with the following step of "Registering ADI with JTS".

Installing the ADI application into IBM WebSphere Application Server

By default, the CLM applications are pre-configured to use IBM WebSphere Liberty. If you have set up IBM WebSphere Application Server instead of using Liberty to deploy CLM web applications, follow the instructions in this section. You can install ADI together with JTS that is installed on the same WebSphere Application Server machine and in the same installation directory. In this case, ADI runs on the same WebSphere Application Server as CLM and has the same base URL as the JTS and the other locally running applications.

Setting up a database

If you use DB2 for your CLM databases, then you need to create the ADI database in the existing DB2. You do not need to create JTS and DW database since they were created during the CLM's setup. For more information, see Installing ADI by using web-based custom Setup.

Installing ADI on WebSphere Application Server

Prerequisite:

- Assumption is that the WebSphere Application Server is installed and started. It has been set up appropriately to deploy CLM applications war files. For more information, see the instructions on CLM Online help: Setting up WebSphere Application Server.

- You have the WebSphere administrative user name and password.

- Verify that JAZZ_HOME, a system property that specifies where to find certain configuration files has been added correctly to WebSphere Application Server. This property uses a URL rather than a simple path, so specify the value as a file-style URL: `file://clm_install/JazzTeamServer/server/conf`. This can be verified under **Java Virtual Machine > Additional Properties**, click **Custom properties**. Ensure Jazz Team Server application is installed and started on the WebSphere Application Server.

- Jazz Team Server application has been updated to map security roles to users or repository groups.

- The database is created. If you use the default Derby database for evaluation purposes, ensure the location to the database in the `teamserver.properties` file is an absolute path.

The WebSphere Application Server profile directory into which the CLM applications (including JTS) are installed is referred to as *was_profiledir*. The CLM installation directory is referred to as *clm_install*..

1. Stop the WebSphere Application Server with the following commands.

```
cd was_profiledir/bin
stopServer
```

2. Extract the ADI distribution in a temporary directory referred to as *adi_unzip* below.

3. Merge some of the ADI installation files into the JTS installation by copying (only) the following files and directories from the *adi_unzip* directory into the corresponding *clm_install* directory.

Table 2. Files and directories involved in merging ADI installation into JTS installation

Copy from *adi_unzip*/adi/	To *clm_install*/JazzTeamServer/
cobol-coverage	(copy the entire folder into JazzTeamServer folder)
license/adi	license/
java-coverage	(copy the entire folder into JazzTeamServer folder)
repotools/adi	repostools
server/conf/adi	server/conf/
server/conf/jts/sites/ com.ibm.dimez.dashboard.activation.updatesite	server/conf/jts/sites
server/conf/jts/provision_profiles/dimez-product-activation.ini	server/conf/jts/provision_profiles/
server/headless-CC	server/
server/liberty/clmServerTemplate/apps/ adi.war.zip	server/webapps/
server/liberty/clmServerTemplate/apps/ adihelp.war.zip	server/webapps/
server/codecov.shutdown.*	server/
server/codecov.startup.*	server/
server/repotools-adi.*	server/

Table 2. Files and directories involved in merging ADI installation into JTS installation (continued)	
Copy from *adi_unzip*/adi/	To *clm_install*/JazzTeamServer/
swidtag/ ibm.com_IBM_Application_Delivery_Intelligence-5.0.3.swidtag	swidtag/

4. Rename the ADI application zip files in *clm_install*/JazzTeamServer/server/webapps/ from adi.war.zip and adihelp.war.zip to adi.war and adihelp.war respectively.

5. Start the WebSphere Application Server again by running the following commands.

```
cd was_profiledir/bin
startServer
```

6. Open a browser and log in to WebSphere Integrated Solutions Console at https://hostname.example.com:9043/ibm/console/logon.jsp.

7. Click **Applications > New Application > New Enterprise Application**.

8. On the **Path to the new application** page, select **Remote file system**.

9. Browse the adi.war ile in *clm_install* directory and cick **Next**.

10. Select **Fast Path** and click **Next**.

11. Click **Next** to accept all default options until you reach the **Map context roots for web modules** page.

12. In the Map context roots for web modules, set **Context Root** to /adi.

13. Click **Finish**.

14. Verify that your application was installed correctly and then click **Save** directly to the main configuration.

15. In WebSphere Integrated Solutions Console, click **Applications > Application Types > WebSphere enterprise applications**.

16. Select the check box next to adi_war, and then click **Start**. A green arrow appears that indicates the application is successfully started.

17. Repeat the preceding steps to deploy and start adihelp.war application. Note, in the Map context roots for web modules, set **Context Root** to /adihelp.

18. Continue with the step of "Registering ADI with JTS".

Registering ADI with JTS

After you install ADI, you still need to register it with the JTS. If you install CLM and ADI fresh together at the same time, then the registration step will be part of the regular jts/setup. In the following, it is assumed that CLM was already set up and you are adding ADI to that setup after the fact.

1. As a Jazz administrator, navigate to the JTS home page either through the Home menu of any application or by going to the URL: https://*server*:*port*/jts/admin.

2. Click the **Server** tab and select the **Registered Application** menu. A list of applications that are already registered with this JTS appears, among which ADI hasn't included yet.

3. Determine the root services URL for your ADI application:

   a. In case that you co-installed ADI on the same server machine and directory as you do with JTS, the URLs will have the same machine name and port as the JTS URL. For example, if your JTS URL is https://server.sample.org:9753/jts then the ADI root services URL will be https://server.sample.org:9753/adi/scr.

   b. In case you installed ADI on another machine, then the URL will be based on that other machine's hostname. For example, the JTS URL could be https://server01.sample.org:9753/jts then the ADI root services URL could be https://server02.sample.org:9753/adi/scr.

4. In the **Registered Application** admin page, click **Add**.

5. Under **Discovery URL**, fill in the URL you specified in Step 3 and click **Tab** on your keyboard. The dialog immediately tells you that it is verifying the URL you type, which can take a moment. If the verification succeeds, the dialog updates. If you received an error message, you most likely typed the URL wrong or you installed ADI on another machine.

6. Fill in the application name, for example, /adi to be consistent with the other apps. Fill in a Consumer Secret twice to confirm the password that applications use to authenticate their communications with each other. Keep all other fields as suggested by the dialog.

7. Click **Finish**. When the dialog closes, you see a green status message at the top. Click **Application Setup Wizard** in that message to continue. This leads you to the page `https://server:port/adi/customSetup`.

8. Walk through the setup wizard as instructed.

    a. If you configured a DB2 database for ADI then specify its details in the **Configure Database** step.

    b. In the **Configure Data Warehouse** step, you must configure the same warehouse database as the JTS. If you do not have the connection details and password, you must contact the JTS administrator who set it up. Because ADI will work only when you configure the warehouse correctly.

    c. Click **Finish** to complete the application setup.

9. Again log in to the administration page of the Jazz Team Server. Navigate your browser to `https://server:port/jts/admin`.

10. On the Administration page, click the **Server** tab.

11. In the **Licensing** pane, click **License Key Management**.

    Note: You must install entitled client access license before you can access ADI and assign it to all your users as described in "Managing client access licenses" on page 113.

12. Optional: Activate a trial license for evaluation purposes.

    a. In the **Client Access License Types** area, look for IBM Application Delivery Intelligence Product and click **Activate Trial**.

    b. In the **Client Access License Agreement** window, review the terms for the license and then click **I accept the terms in the license agreement**.

    c. Click **Finish** to make the trial license active.

    d. Assign it to all your users as described in "Managing client access licenses" on page 113.

What to do next

After you install ADI into an existing CLM, you can find the next steps to perform more operations.

After you set up the application, you can follow the steps for creating a project area as described in Creating and managing users and assign your existing CLM user base as ADI users of that project area. You must also install and assign ADI Client Access Licenses to all your users as described in Managing Client Access Licenses.

Setting up ADI with adi-setup application

The adi-setup application automates various setups that had to be executed manually in the past.

The adi-setup application is a command line program that is provided in the installation folder: `/adi/server/adi-setup` (`adi-setup.bat` on Windows). With the adi-setup application, users can perform the following operations:

- Create an ADI evaluation installation on a local machine with one simple command line.
- Migrate ADI from a previous release with one simple command line.
- Execute scripts that are provided by technical support for analyzing and fixing problems.

Reviewing the command line options for adi-setup application

To review all the options available in the adi-setup application, run the following command.

```
> adi-setup help
```

Creating an ADI evaluation installation with adi-setup application

You can use the adi-setup application to quickly set up ADI for evaluation purposes either on a local laptop or an evaluation sever. The idea is that you download the ADI zip file, unzip it, run the adi-setup command, and you are done.

Before you can run the following command, you need to start the server first. For more information, see "Starting up the server" on page 81.

Run the command with the following options:

```
adi-setup adiEvaluationSetup -- Creates an evaluation installation using all the defaults.
    [-baseUrl https://server.org:9753] -- Base URL without a path that should be used for this
server. If not provided it will assume http://localhost:9753, but then the server cannot be
used over the network.
    [-user.name username] -- user name of the admin user to be created. If you skip this option
the name "eg" will be used.
    [-user.password password] -- password of the admin user to be created. If you skip this
option the password "eg" will be used.
```

All options after `adiEvaluationSetup` are optional and the server uses defaults. If you skip them, it will create a server at localhost: 9753, that is, the server cannot be reached from another machine, with "eg" as both username and password. If the server is configured with a fully qualified DNS name, it is recommended that you use that in the `-baseUrl` parameter.

After you run the adi-setup command, restart the server by completing the following steps:

1. "Shutting down the server" on page 91
2. "Starting up the server" on page 81

Migrating from a previous release

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

Notes:

1. The following instructions assume that you had either installed a previous version in either of the following locations. If you installed 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* and *adi_new* as the installation directory names to differentiate the previous version from the new version. You can replace the names based on your migration scenarios. For example, if you migrate from ADI V5.0.5 to ADI V5.1.0, replace the corresponding names to `adi505` and `adi510`.

Migration considerations

Learn about the migration scenarios and results when you migrate from a previous version to ADI V5.1.0.6 or later.

Starting from V5.1.0.6, the Static Analysis and Business Rule Discovery features are moved from ADI to ADDI as the ADDI Extension component. Hence, after you migrate from a previous release of ADI to ADI 5.1.0.6 or later, the following components are not available to view or modify.

- Static analysis data within a workbook
- Business terms and business rules packages information within a workbook

- Application Discovery data providers
- Business Rule Discovery data providers
- Application Discovery connection to IBM Application Discovery server

Check the following table that summarizes the possible migration scenarios and results for migration.

Table 3. Migration scenarios and results					
	Data stored in the previous release of ADI		**Components required to install for migration**		
Scenarios	**Supported features** • Code Coverage Analysis • OMEGAMON for CICS Analysis • SMF / APA Data Analysis	**Features moved to ADDI Extension** • Static Analysis • Business Rule Discovery	**ADI V5.1.0.6 or later**	**ADDI Extension**	**Migration Results**
1	Included	Not included	X		All data from a previous release of ADI are migrated to ADI V5.1.0.6 or later.

	Table 3. Migration scenarios and results (continued)				
	Data stored in the previous release of ADI		**Components required to install for migration**		
2	Included	Included	X	X	The following data are migrated to ADI V5.1.0.6 or later. • Code coverage data • OMEGAMON for CICS data • SMF / APA data The following data are migrated to ADDI Extension. • Static analysis data • Business rule discovery data **Note:** When ADI and ADDI Extension are installed on the same server, you will need to consider changing the default port of ADDI Extension.
3	Not included	Included		X	All data from the previous release of ADI are migrated to ADDI Extension components on ADDI V5.1.0.6 or later.

Manually migrating from a previous release

For ADI migration, you can manually migrate the data files from a previous release into the new directory where you install the new release.

Note: Before migration, you need to remove all the Enterprise Artifact workbooks and Enterprise Artifact data providers. To save the business terms in the Enterprise Artifact workbooks, you can export the business terms from the workbooks before deletion and import them after migration.

Installing the new release into a new directory
You install every new release into a new installation directory parallel to the previous version's installation. You do not install releases on top of each other to upgrade.

Take the following steps to install the new release of ADI into a new directory.

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

   Linux:

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

   Windows:

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

2. 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
   ```

3. Download the new release of ADI and extract it into the same location as you did with the previous release. Rename the new release in the same way so that you can differentiate it from the previous one. For example, rename it to *adi_new*. Then you have the *adi_previous* and *adi_new* next to each other in the same installation location.

Backing up database files
If you use an enterprise database, such as DB2, you can make backups of your data to be able to repeat the migration in case of errors. In case of Derby, you copy your databases to the new location. Complete the steps in this topic to back up the database files.

If you are using DB2, make sure that you follow the instructions of Backing up your data to back up your three databases: JTS, ADI, DW.

If you are using Derby, no backup is required as you will have copies of these databases in your previous installation directory. You must copy them to the new installation location with the following steps.

Notes:

- Make sure you adjust the path to your installation directory if necessary.
- Do not perform these steps if you are using DB2.

1. Create the Liberty application directory to copy the files to.

   Linux:

```
cd /opt/ibm/adi_new/server
./server.startup -create
```

Windows:

```
cd C:\IBM\adi_new\server
server.startup -create
```

2. Delete the default Derby databases to replace them with the databases from the previous release:

   Linux:

   ```
   rm -rf /opt/ibm/adi_new/server/conf/jts/derby/repositoryDB
   rm -rf /opt/ibm/adi_new/server/conf/adi/derby/repositoryDB
   ```

   Windows:

   ```
   rd /s/q C:\IBM\adi_new\server\conf\jts\derby\repositoryDB
   rd /s/q C:\IBM\adi_new\server\conf\adi\derby\repositoryDB
   ```

3. Copy the Derby databases from the previous release to the new release.

   Note: The second command points to a path inside the Liberty application sever directory. That is where the Derby warehouse is located currently because a defect in the JTS.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/jts/derby/repositoryDB /opt/ibm/adi_new/server/
   conf/jts/derby/repositoryDB
   cp -R /opt/ibm/adi_previous/server/liberty/servers/clm/conf/jts /opt/ibm/adi_new/server/
   liberty/servers/clm/conf/jts
   cp -R /opt/ibm/adi_previous/server/conf/adi/derby/repositoryDB /opt/ibm/adi_new/server/
   conf/adi/derby/repositoryDB
   ```

   Windows:

   ```
   xcopy /s/i C:\IBM\adi_previous\server\conf\jts\derby\repositoryDB C:\IBM\adi_new\server\conf
   \jts\derby\repositoryDB
   xcopy /s/i C:\IBM\adi_previous\server\liberty\servers\clm\conf\jts\derby\warehouseDB C:\IBM
   \adi_new\server\liberty\servers\clm\conf\jts\derby\warehouseDB
   xcopy /s/i C:\IBM\adi_previous\server\conf\adi\derby\repositoryDB C:\IBM\adi_new\server\conf
   \adi\derby\repositoryDB
   ```

Copying key configuration and data files from the previous installation to the new one
You can run various commands to copy configuration files from the previous location to the new one. In some cases, you need to update the files to accommodate new settings.

You need to copy the key configuration files from the previous installation location to the new one.

1. Copy RDF index files for ADI.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/adi/indices /opt/ibm/adi_new/server/conf/adi/indices
   ```

   Windows:

   ```
   xcopy /s/i C:\IBM\adi_previous\server\conf\adi\indices C:\IBM\adi_new\server\conf\adi
   \indices
   ```

2. Copy the uploaded code coverage files for ADI.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/adi/coverageFileStorage /opt/ibm/adi_new/server/
   conf/adi/coverageFileStorage
   ```

   Windows:

```
xcopy /s/i C:\IBM\adi_previous\server\conf\adi\coverageFileStorage C:\IBM\adi_new\server
\conf\adi\coverageFileStorage
```

3. Copy the headless-CC-files for ADI.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/adi/headless-CC-files /opt/ibm/adi_new/server/
   conf/adi/headless-CC-files
   ```

   Windows:

   ```
   xcopy /s/i C:\IBM\adi_previous\server\conf\adi\headless-CC-files C:\IBM\adi_new\server\conf
   \adi\headless-CC-files
   ```

4. Copy the uploaded enterprise artifacts under the `businessRuleDiscovery` directory for ADI.

   Note: You need to take this step only when you migrate from ADI V5.1.0 or its later fix packs.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/adi/businessRuleDiscovery /opt/ibm/adi_new/server/
   conf/adi/businessRuleDiscovery
   ```

   Windows:

   ```
   xcopy /s/i C:\IBM\adi_previous\server\conf\adi\businessRuleDiscovery C:\IBM\adi_new\server
   \conf\adi\businessRuleDiscovery
   ```

5. Copy the ADI application's setting file. If a prompt to overwrite the existing files occurs, confirm it.

   Linux:

   ```
   cp /opt/ibm/adi_previous/server/conf/adi/teamserver.properties /opt/ibm/adi_new/server/
   conf/adi/
   ```

   Windows:

   ```
   copy /y C:\IBM\adi_previous\server\conf\adi\teamserver.properties C:\IBM\adi_new\server\conf
   \adi\
   ```

6. Copy the RDF index for JTS.

   Linux:

   ```
   cp -R /opt/ibm/adi_previous/server/conf/jts/indices /opt/ibm/adi_new/server/conf/jts/indices
   ```

   Windows:

   ```
   xcopy /s/i C:\IBM\adi_previous\server\conf\jts\indices C:\IBM\adi_new\server\conf\jts
   \indices
   ```

7. Copy the JTS settings file. If a prompt to overwrite the existing files occurs, confirm it.

   Linux:

   ```
   cp /opt/ibm/adi_previous/server/conf/jts/teamserver.properties /opt/ibm/adi_new/server/
   conf/jts/
   ```

   Windows:

   ```
   copy /y C:\IBM\adi_previous\server\conf\jts\teamserver.properties C:\IBM\adi_new\server\conf
   \jts\
   ```

8. Choose one of the following options to continue:

   • If you migrate from a version earlier than ADI V5.1.0, skip this step.

- If you migrate from ADI V5.1.0 or later versions, copy the Elasticsearch indexes that are created.

Linux:

```
cp -R /opt/ibm/adi_previous/elasticsearch/data /opt/ibm/adi_new/elasticsearch/data
```

Windows:

```
xcopy /s/i C:\IBM\adi_previous\elasticsearch\data C:\IBM\adi_new\elasticsearch\data
```

9. Create the Liberty application directory to copy files to.

Note:

If you are using Derby, you can skip the following step as you've executed it in the previous step Backing up database files.

If you are using DB2, you need to complete the following step.

Linux:

```
cd /opt/ibm/adi_new/server
./server.startup -create
```

Windows:

```
cd C:\IBM\adi_new\server
server.startup -create
```

10. Copy Liberty setting files. If a prompt to overwrite the existing files occurs, confirm it.

Linux:

```
cp /opt/ibm/adi_previous/server/liberty/servers/clm/server.xml /opt/ibm/adi_new/server/
liberty/servers/clm/
cp /opt/ibm/adi_previous/server/liberty/servers/clm/conf/basicUserRegistry.xml /opt/ibm/
adi_new/server/liberty/servers/clm/conf/
cp /opt/ibm/adi_previous/server/liberty/servers/clm/conf/ldapUserRegistry.xml /opt/ibm/
adi_new/server/liberty/servers/clm/conf/
```

Windows:

```
copy /y C:\IBM\adi_previous\server\liberty\servers\clm\server.xml C:\IBM\adi_new\server
\liberty\servers\clm\
copy /y C:\IBM\adi_previous\server\liberty\servers\clm\conf\basicUserRegistry.xml C:\IBM
\adi_new\server\liberty\servers\clm\conf\
copy /y C:\IBM\adi_previous\server\liberty\servers\clm\conf\ldapUserRegistry.xml C:\IBM
\adi_new\server\liberty\servers\clm\conf\
```

After you copy all the key configuration files to the new installation location, choose one of the following options to continue.

- If you migrate from ADI V5.1.0 or its later fix packs, skip this step.
- If you migrate from a version earlier than ADI V5.1.0, complete the following steps to update server.xml file to configure Lightweight Third Party Authentication (LTPA) in Liberty.

1. Open the server.xml file.

Linux:

```
/opt/ibm/adi_new/server/liberty/servers/clm/server.xml
```

Windows:

```
C:\IBM\adi_new\server\liberty\servers\clm\server.xml
```

2. Update the version numbers for the following two `feature` attributes.

```
<feature>jsp-2.3</feature>
<feature>servlet-3.1</feature>
```

3. Add the following line after `<webAppSecurity ssoRequiresSSL="true"/>`.

```
<ltpa expiration="12h"/>
```

4. Save your changes to the `server.xml` file.

Make sure that WebSphere Liberty knows about the newly added applications. Choose one of the following options to continue.

- If you migrate from a version earlier than ADI V5.1.0, update the `application.xml` file to let the WebSphere Liberty know about the newly added applications in the new release.

  1. Open the `application.xml` file.

     Linux:

     ```
     /opt/ibm/adi_new/server/liberty/servers/clm/conf/application.xml
     ```

     Windows:

     ```
     C:\IBM\adi_new\server\liberty\servers\clm\conf\application.xml
     ```

  2. Add new `<application>` lines in the `<server>` element.

     ```
     <application type="war" id="addi-web-brd" name="addi-web-brd" location="$
     {server.config.dir}/apps/addi-web-brd.war">
     </application>
     ```

     ```
     <application type="war" id="addi-web-es" name="addi-web-es" location="${server.config.dir}/
     apps/addi-web-es.war">
     </application>
     ```

     ```
     <application type="war" id="addi-web-metrics" name="addi-web-metrics" location="$
     {server.config.dir}/apps/addi-web-metrics.war">
     </application>
     ```

  3. Save your changes to the `application.xml` file.

- If you migrate from ADI V5.1.0 or its later fix packs, open the `application.xml` file and verify that the following new applications are listed in it. If you find any missing applications, add them to the `application.xml` file.

  ```
  <application type="war" id="addi-web-brd" name="addi-web-brd" location="${server.config.dir}/
  apps/addi-web-brd.war">
  </application>
  ```

  ```
  <application type="war" id="addi-web-es" name="addi-web-es" location="${server.config.dir}/
  apps/addi-web-es.war">
  </application>
  ```

  ```
  <application type="war" id="addi-web-metrics" name="addi-web-metrics" location="$
  {server.config.dir}/apps/addi-web-metrics.war">
  </application>
  ```

Migrating the JTS data
You can execute an interactive command to migrate the JTS configuration files, the data repository, and the warehouse databases to the latest version.

Run the following command to migrate the JTS data:

Linux:

```
cd /opt/ibm/adi_new/server
upgrade/jts/jts_upgrade.sh -oldJTSHome /opt/ibm/adi_previous/server/conf -noPrompt -
noStepPrompt -noEditor
```

Windows:

```
cd C:\IBM\adi_new\server
upgrade\jts\jts_upgrade.bat -oldJTSHome C:\IBM\adi_previous\server\conf -noPrompt -noStepPrompt
-noEditor
```

If you configured your server with nondefault options or you want more control over all the steps that this migration script perform, you can see Jazz Online Help for more information.

Migrating the ADI application data
You can execute a command to update the data repository that stores ADI data.

Run the following command to migrate the repository database of ADI application.

Linux:

```
cd /opt/ibm/adi_new/server
./repotools-adi.sh -addTables noPrompt
```

Windows:

```
cd C:\IBM\adi_new\server
repotools-adi.bat -addTables noPrompt
```

Now the server has all the data of your previous server. However, you might still want to add other configuration options, for example, renewing server certificates. For more information, see Jazz Online Help.

Starting the server and waiting for additional migration to finish
After all the files are migrated, you can start the server. It executes additional migration tasks that the admin user can review for success.

When you start the server, it executes more migration operations automatically. An administrator can review the progress of these operations in the log files that get generated on the server machine.

1. Start the new ADI server.

   Linux:

   ```
   cd /opt/ibm/adi_new/server
   ./server.startup
   ```

   Windows:

   ```
   cd C:\IBM\adi_new\server
   server.startup
   ```

2. Use a browser to navigate to the applications main page: `https://`*fully qualified hostname*`:9753/adi/web`

3. Observe the progress of the migration in the log file located in *adi_new*`/server/logs/adi.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.

4. Wait for these two lines to be shown in the log.

   ```
   CRIDA0059I The DIMEZ warehouse schema validation has completed.
   ```

   and

```
CRIDA0052I Completed checking the ADI data migration status.
```

Also review previous lines shown for status messages and warnings about the migration. If there are errors, review the error message to understand the nature of the problem. You can also start over by redoing the steps described in this guide with a new fresh installation. Contact technical support to get help with any migration error messages.

5. When the migration is shown as successful in the log, you can notify your users that the new server is available for use.

Migrating from a previous release with adi-setup application

For ADI V5.0.4 and later versions, you can run adi-setup application to automate the steps for migrating ADI from a previous release.

To migrate ADI from a previous release, you can perform manual steps as described in "Migrating from a previous release" on page 100. Alternatively, you can run adi-setup application command by using the following options, which automates all these steps for migration.

Notes:

- It is recommended that you manually back up the existing ADI database before you start the migration process. For more information, see "Backing up database files" on page 103.
- For the migration of Business Rule Discovery data providers, check the following instructions.

    – Migration from V5.0.5 to V5.1.0

    Before migration, you need to remove all the Enterprise Artifact workbooks and Enterprise Artifact data providers. To save the business terms in the Enterprise Artifact workbooks, you can export the business terms from the workbooks before deletion and import them after migration.

    – Migration from V5.1.0 to V5.1.0.x

    The statuses of business terms in V5.1.0 are not preserved after you migrate from ADI V5.1.0 to ADI V5.1.0.x. You need to manually correct the statuses based on the following mapping.

Table 4. The status mapping between V5.1.0 and V5.1.0.x

V5.1.0 Status	V5.1.0.x Status
New	Waiting For Approval
In Progress	New
Waiting For Approval	Rejected
Approved	In Progress
Rejected	Approved

Important:

1. The following instructions assume that you had either installed a previous version in either of the following locations. If you installed 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* and *adi_new* as the installation directory names to differentiate the previous version from the new version. You can replace the names based on your migration scenarios. For example, if you migrate from ADI V5.0.5 to ADI V5.1.0, replace the corresponding names to `adi505` and `adi510`.

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

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

    Linux:

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

Windows:

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

2. 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
   ```

3. Download the new release of ADI and extract it into the same location as you did with the previous release. For example, /opt/ibm for Linux or C:\IBM for Windows.

   Then, rename the new release's folder in a similar way so that you can positively identify its code level and differentiate it from the previous ADI installation level and folder name. For example, rename it to *adi_new*. Then you have the *adi_previous* and *adi_new* at the same relative file path under the same installation location. Check the following examples.

   Linux:

   The location of the previous release is /opt/ibm/adi_previous. Then, the location of the new release is /opt/ibm/adi_new.

   Windows:

   The location of the previous release is C:\IBM \adi_previous. Then, the location of the new release is C:\IBM\adi_new.

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

   Linux:

   ```
   cd /opt/ibm/adi_new/server
   ```

   Windows:

   ```
   cd C:\IBM\adi_new\server
   ```

   Note: 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".

5. Run one of the following commands and wait for the command to complete.

   • If you migrate from a modification level to a later modification level, such as migrating from ADI V5.0.4 to ADI V5.0.5, run the following command:

     Linux:

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

     Windows:

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

- If you migrate from a fix pack level to a later fix pack level, such as migrating from ADI V5.1.0.0 to ADI V5.1.0.x, run the following command:

  Linux:

  ```
  adi-setup migration -fromDir /opt/ibm/adi_previous -skipJazzUpdate true
  ```

  Windows:

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

  For more information about what is a modification level or a fix pack level, see <u>V.R.M.F Maintenance Stream Delivery Vehicle terminology explanation</u>.

6. Choose one of the following options to continue.

   - If you migrate from ADI V5.1.0 or its later fix packs, skip this step.
   - If you migrate from a version earlier than ADI V5.1.0, complete the following steps to update `server.xml` file to configure Lightweight Third Party Authentication (LTPA) in Liberty.

     a. Open the `server.xml` file under the new installation directory.

        Linux:

        ```
        /opt/ibm/adi_new/server/liberty/servers/clm/server.xml
        ```

        Windows:

        ```
        C:\IBM\adi_new\server\liberty\servers\clm\server.xml
        ```

     b. Update the version numbers for the following two `feature` attributes.

        ```
        <feature>jsp-2.3</feature>
        <feature>servlet-3.1</feature>
        ```

     c. Add the following line after `<webAppSecurity ssoRequiresSSL="true"/>`.

        ```
        <ltpa expiration="12h"/>
        ```

     d. Save your changes to the `server.xml` file.

7. Choose one of the following options to continue.

   - If you migrate from a version earlier than ADI V5.1.0, update the `application.xml` file to let the WebSphere Liberty know about the newly added applications in the new release.

     a. Open the `application.xml` file.

        Linux:

        ```
        /opt/ibm/adi_new/server/liberty/servers/clm/conf/application.xml
        ```

        Windows:

        ```
        C:\IBM\adi_new\server\liberty\servers\clm\conf\application.xml
        ```

     b. Add new `<application>` lines in the `<server>` element.

        ```
        <application type="war" id="addi-web-brd" name="addi-web-brd" location="$
        {server.config.dir}/apps/addi-web-brd.war">
        </application>
        ```

        ```
        <application type="war" id="addi-web-es" name="addi-web-es" location="$
        {server.config.dir}/apps/addi-web-es.war">
        </application>
        ```

```
<application type="war" id="addi-web-metrics" name="addi-web-metrics" location="$
{server.config.dir}/apps/addi-web-metrics.war">
</application>
```

  c. Save your changes to the `application.xml` file.

- If you migrate from ADI V5.1.0 or its later fix packs, open the `application.xml` file and verify that the following new applications are listed in it. If you find any missing applications, add them to the `application.xml` file.

```
<application type="war" id="addi-web-brd" name="addi-web-brd" location="$
{server.config.dir}/apps/addi-web-brd.war">
</application>
```

```
<application type="war" id="addi-web-es" name="addi-web-es" location="${server.config.dir}/
apps/addi-web-es.war">
</application>
```

```
<application type="war" id="addi-web-metrics" name="addi-web-metrics" location="$
{server.config.dir}/apps/addi-web-metrics.war">
</application>
```

8. Start the new server as described in .

Integration with IBM License Metric Tool

IBM Application Delivery Intelligence for IBM Z generates IBM Software License Metric Tag (SLMT) files. The versions of IBM License Metric Tool that support IBM Software License Metric Tag can generate License Consumption Reports.

Each instance of a running environment generates an IBM Software License Metric Tag file. The USER metrics are monitored. The values are refreshed every 15 minutes.

IBM Application Delivery Intelligence for IBM Z generates an SLMT tag file that records license usage at 15 minutes interval in which users logged on and used the application. When the application was not used, no activities will be logged. The recorded information is:

AUTHORIZED_USER: containing the name of the authorized account that logged on ADI.

IBM SLM tag files are automatically updated and stored in the server folder `adi/server/conf/adi/`. The file name will use a random identifier such as 538b3879154264d5ec0e606276ee98fb.slmtag.

Following is an example of the content of an SLM tag file:

```
<SchemaVersion>2.1.1</SchemaVersion>
        <SoftwareIdentity>
                <PersistentId>a09471a5d2b04cf48d14b5a7d6287aa0</PersistentId>
                <Name>IBM Application Delivery Intelligence</Name>
                <InstanceId>/home/ibmadmin/ADI/adi5.0.5/server/conf/adi</InstanceId>
        </SoftwareIdentity>

        <Metric logTime="2561-06-04T14:42:53+07:00">
                <Type>AUTHORIZED_USER</Type>
                <SubType></SubType>
                <Value>2</Value>
                <Period>
                        <StartTime>2561-06-04T14:27:53+07:00</StartTime>
                        <EndTime>2561-06-04T14:42:53+07:00</EndTime>
                </Period>
        </Metric>

        <Metric logTime="2561-06-04T14:57:53+07:00">
                <Type>AUTHORIZED_USER</Type>
                <SubType></SubType>
                <Value>2</Value>
                <Period>
                        <StartTime>2561-06-04T14:27:53+07:00</StartTime>
                        <EndTime>2561-06-04T14:57:53+07:00</EndTime>
                </Period>
        </Metric>
```

Chapter 5. Administration

As a administrator for ADI, you can learn what administration tasks you can do and how to do them.

Logging in to administer the server

`/jts/admin` and `/adi/admin` are two admin pages available that you can log in to administer the server. One is for administering the Jazz Team Server (JTS), and the other is for the ADI application. Users can be created and managed in either of them. Licenses need to be managed in the `jts/admin`.

The following example demonstrates the steps to log in to a JTS. For more information, see Logging in to administer the server.

1. Go to `https://localhost:9753/jts/admin`.
2. Log in with the administrative user name and password that you created during the Jazz Team Server setting up.

Administration tasks for user management

For user management, you can manage client access licenses, create and manage users, and change password.

When you add users to ADI, you need to perform the following workflow:

1. Activate the trial or import the purchased ADI client access licenses into the Jazz Team Server.

   Note: The ADI license are *authorized licenses*, which are permanent licenses that are assigned to a single user. The ADI license is managed by the Jazz license server. If you purchase activation kits that contain the activation key file, you can import the activation key to the license server. For more information, see Installing and managing client access license keys.

2. Create new users or import users in the Jazz Team Server.
3. Assign licenses to the users.
4. Create one or more ADI project areas in the ADI admin UI.
5. Assign users to the project area.

For more instructions on those steps, see the following topics.

Managing client access licenses

To access ADI, you must have a valid user access license and import the activation key to the license server.

Administrators can import the license key in the administrative web UI of that Jazz Team Server (`/jts/admin`).

Note: The ADI license are *authorized licenses*, which are permanent licenses that are assigned to a single user. The ADI license is managed by the Jazz license server. If you purchase activation kits that contain the activation key file, you can import the activation key to the license server. For more information, see Installing and managing client access license keys.

If you do not have the activation key, ADI ships with trial licenses that you can enable during setup or within that same administrative UI for importing licenses. Administrators can import and configure the purchased user access license keys to replace the trail licenses anytime after or during the trial period.

After you import and configure the user access licenses, you can create users and then assign the client access license to users. For more information, see ""Creating and managing users" on page 114" and "Assigning client access licenses to users" topics.

Creating and managing users

As administrators, you can learn how to create or import users, assign licenses to users, grant users as members of project areas, and assign role-based permissions to users.

To add a user to ADI, you need to complete the following procedures.

- Create a user or import it from an external user management service.
- Assign client access licenses to the new user.
- Create a project area or use an existing one.
- Assign the user to that project area.
- Assign role-based permissions to ADI users.

Creating and managing user accounts

Currently ADI supports user management by the embedded Websphere Liberty Profile application server and LDAP user registries. You can learn more about how to create and manage user accounts as administrators.

For configuring either option, see Installing ADI by using web-based custom setup page for references.

The Jazz Online Help describes all the details for creating new users locally in Liberty and how to import from LDAP. All these operations are performed in the JTS administrative web UI (`/jts/admin`):

- For the general overview to user management, see Managing users.
- For information about how to create users locally, see Creating users.
- For information about how to import users from LDAP if it was configured during setup: Importing users from an external user registry.
- For information about how to assign user client access licenses, see Assigning client access licenses to users.
- For information about how to assign user roles and permissions, see Assigning roles to a member.
- For information about how to modify the user permissions, see Modifying role-based permissions.

Note: When you use a local user registry in Liberty and create new users, the password of this new user is the same as the user id specified. The new user can then log on with that temporary password and use the user profile page to change it. For more details, see Users changing their password.

Creating and managing project areas

After you create users in the JTS and assign them client access licenses, the user accounts will be replicated and made available to the ADI application. They will also be visible in the ADI administrative Web UI (`/adi/admin`). You as an administrator can then assign users to one or more ADI project areas. Project areas partition all data managed in ADI and provide access to user to that data. They can be used to organize the content of different teams that do not want the others to have access to their data.

- To learn more about project areas, see the Jazz Online help: Project area.

  Note: The current release of ADI does not support all the capabilities described here. In particular, it doesn't yet support Behavior and Timelines.

- After the setup, the administrator must create at least one ADI project area and assign each user to at least one of these project areas.

  - Procedures:

    1. To create a project area, you need to navigate to the ADI Administrative Web UI located at URL: `https://`*`fully qualified hostname`*`:9753/adi/admin`.
    2. Select **Project Areas > Active Project Areas** from the menu.
    3. Select the **Create Project Area** operation.
    4. Complete the form by providing at least a value for the Project Area Name.
    5. Select the default ADI Process.

6. Click **Add** in the **Members** section to add one or more users as team members. Only the users who are listed here are able to access the data for this project area.

7. Click **Add** in the **Administrators** section to add one or more administrative users as administrators of this project area. Only the administrators of the project area can modify the information of the project area such as adding a member to the project area.

8. Click **Save** to start the creation.

   Note: After you click **Save**, a warning about missing user roles appears.

9. Assign the user roles and permissions to each user. For more information about ADI role-based permissions, refer to the table of "Assigning role-based permissions to ADI users" on page 115.

   – To learn more about creating Project Areas, see the Jazz Online Help: Creating a project area.

- Once the project area is created users that are listed as members are able to access it from this URL: `https://fully qualified hostname:9753/adi/web`.

- At any time, the administrator can add more users to the project area. To learn more about managing user for a project area, see the Jazz Online Help: Adding and modifying users as members of project areas.

   Notes:

   – Any other user that you add to the **Administrators** section is able to follow the previous steps to manage membership of that project area.

   – Jazz Online Help also shares information about Team Areas. ADI also uses team areas, but they should be created as part of the main ADI Web UI by creating workbooks. Workbooks map one-to-one to a Jazz Team Area. For more details, see "Managing workbooks" on page 173.

Assigning role-based permissions to ADI users

ADI 5.0.2 and later versions leverages Jazz Team Server (JTS) role-based permissions in order to manage operations that the users can perform within workbooks. From this topic, you can find more information about roles and permissions in ADI.

The table below shows the list of user roles and permissions set by default. You can edit the following permissions based on your needs.

Note: A user can be assigned to more than one user roles.

Table 5. User roles and permissions for ADI		
User Roles	**ADI View**	**Operations**
Data Provider Administrator	Connections	Create, modify, and delete connections.
	Data Providers	Create, modify, and delete data providers
	Workbooks	Create, modify, and delete workbooks. View analysis reports of workbooks of which they are members.

Table 5. User roles and permissions for ADI (continued)		
User Roles	**ADI View**	**Operations**
Workbook Owner	Data Providers	View data providers information and modify part of information. Such as adding new builds, uploading coverage results, refreshing build data, deleting builds from an existing Manual Builds data provider or downloading data from data providers.
	Workbooks	Create and modify workbooks. Such as making changes to memberships and setting thresholds. View analysis reports of workbooks of which they are owners.
Workbook Member	Data Providers	View build information, view data collection logs, and download data from data providers. For Manual Builds data providers, the members can upload code coverage results to the existing builds.
	Workbooks	View workbooks. View analysis reports of workbooks of which they are members.

Users changing their password

If you are using the local user management facilities provided by ADI (and not, for example, an LDAP server), users need to change their password in their user profile. You can learn how to manage user accounts from user profiles.

ADI end users can use their account profile to manage their user accounts. Primarily, they can use this profile page to change their passwords in cases a local user directory, that is, not an LDAP user directory, is used.

Procedure:

1. To access user profile page from the ADI Web UI, select the small triangle in the left navigation bar with the **User** icon.

2. In the pop-up menu that opens, select **My Profile**. The Jazz Team Server administration page appears. Log in with your username and password if you are required to do so.

3. To change your password, select the **Change Password..** button on your profile page.

   Note: In the case that your server uses a third party user directory, such as LDAP, this button will be disabled.

For more information about changing password, see Changing your password in the Jazz Online Help.

Backing up your data

You can learn the process on how to back up your data.

To back up your data, follow the instructions described on Backing up and restoring Rational solution for Collaborative Lifecycle Management applications.

- If you configured a Derby database, you can find all the process on Backing up and restoring the Derby database.
- If you configured a DB2 database, you can find all the process on Backing up and restoring other supported databases.

Note: You also need to back up the directory of `adi-install-directory`/server/conf/adi/coverageFileStorage` that stores all the uploaded coverage compressed files.

Server configuration settings

As an administrator, you can learn how to use various settings for controlling the ADI server.

This documentation only focuses on ADI-specific settings. For more information about the other options that are offered by the Jazz framework, see Jazz Online Help .

Settings that determine the behavior of startup scripts

You can learn about those startup scripts and how their behaviors are defined by variables.

The following table outlines the variables that define the behavior of those startup scripts. For example, they determine what processes should be started and on which port they operate.

Table 6. Variables defined in startup scripts			
Script File Name	**Property Name / Parameter Option**	**Default Value**	**Description**
`adi/server/ server.startup`	HEADLESSCC / - noheadlesscc	true	Determines if the headless code coverage daemon process should be started when the ADI server starts. The default is true, that is, the daemon is always started.
			Change the default in the script or start ADI with - `noheadlesscc` to prevent starting the coverage daemon.
			Example: `server.startup - noheadlesscc`
	-Xmx	4G	Determines the maximum memory allocated to ADI.
			Note: When you set up ADI on a machine with inadequate resources for evaluation, you can set the maximum memory allocation to 2G.
	-Xms	4G	Determines the initial memory allocated to ADI.
			Note: When you set up ADI on a machine with inadequate resources for evaluation, you can set the maximum memory allocation to 2G.
	-Xmn	1G	Determines the minimum memory allocated to ADI.

	Property Name /		
Script File Name	**Parameter Option**	**Default Value**	**Description**
adi/server/ codecov.startup	CODECOV_PORT	8005	Defines the port on which the headless code coverage collection daemon should listen for incoming coverage data.
			If you change the value, you also need to change it in the codecov.shutdown script and the teamserver.properties variable *com.ibm.dimez.dashboard.coverageHeadlessCC.port*. For more details, see "Server settings that need to be defined in teamserver.properties before the server starts " on page 120.
adi/server/ codecov.shutdown	CODECOV_PORT	8005	Defines the port on which the headless code coverage collection daemon listens for incoming coverage data.
			If you change the value, you also need to change it in the codecov.startup script and the teamserver.properties variable *com.ibm.dimez.dashboard.coverageHeadlessCC.port*. For more details, see "Server settings that need to be defined in teamserver.properties before the server starts " on page 120.
adi/elasticsearch/ config/jvm.options	-Xmx	2G	Determines the maximum memory allocated to elasticsearch.
	-Xms	2G	Determines the initial memory allocated to elasticsearch.

Table 6. Variables defined in startup scripts (continued)

Server settings that need to be defined in teamserver.properties before the server starts

Before the server starts, you must set several settings that define the behavior and storage locations of the ADI server. For example, the location of the databases or the location of the code coverage resource files. Those settings are provided through a properties file in the installation location: `adi/server/conf/adi/teamserver.properties`. You can learn more about the server settings that need to be defined in teamserver.properties.

Many settings have defaults. If you don't define them in the property file, those settings use the default values automatically. Therefore, if you want to change such a setting and you do not see it in the current file on your server, add the property with that new value. Make sure that you always stop the server first before you make any changes to the property file as some settings can also be set through the Administrative UI.

The following list provides the ADI-specific settings. For other settings in the file that is defined by the Jazz framework that ADI uses, see Jazz Online Help.

Table 7. Server settings that need to be defined in teamserver.properties before the server starts

Property Name	Default Value	Description
com.ibm.dimez.dashboard.coverageFileStoragePath	adi/server/conf/adi/coverageFileStorage	The location in which uploaded code coverage files will be managed. You need to back up the location together with the databases. If you change this value after the initial setup, make sure that you move all directories available in the old location to the new one to avoid breaking ADI's consistency.
com.ibm.dimez.dashboard.coverageFileStorageCheckoutTimeout	60000	For two server threads try to checkout the same coverage file directory at the same time for update, the property defines the amount of time the second threads waits to receive a file lock until it throws a timeout exception. If you see timeout errors frequently when you collect coverage files, increase this value.

Property Name	Default Value	Description
com.ibm.dimez.dashboard.coverageHeadlessCCFilesPath	adi/server/conf/adi/headless-CC-files	The base-directory in which the headless code coverage collector demon is placing code coverage files received by the collector. A subdirectory is available for every data provider. ADI collects files from there and move them into the coverageFileStorage location. If you change this value, you currently have to also change the same path in a second location: in adi/server/codecov.startup.[bat|sh] change the value of the PROVIDERIDDIR variable. This operation configures the headless collection daemon process responsible for receiving coverage data from coverage engine over the net.
com.ibm.dimez.dashboard.coverageHeadlessCC.port	8005	The port of the headless collection daemon process under which code coverage data can be received. This value is only used for creating the Startup Key examples in the ADI UI. To change it, you must also change the value of the CODECOV_PORT variable in adi/server/codecov.startup.[bat|sh].

Server settings that can be defined at runtime in the /adi/admin user interface

When you navigate to the /adi/admin pages of your application, you find an **Advanced Properties** section. It defines variables that can change the ADI server's runtime behavior. Check for more information about variables defined in the /adi/admin from this topic.

An example for a direct URL to the /adi/admin page is https://adi.example.com:9444/adi/admin#action=com.ibm.team.repository.admin.configureAdvanced.

Table 8. Variables Defined in the /adi/admin Page

ADI Component Name	Task Name	Property Name	Default Value	Description
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.AuditScheduledTask	ADI Audit Log Writing Interval (seconds)	900	Defines the interval time logging the information to log files for auditing. Reduce this value if you want the log file to be recorded more often.

Table 8. Variables Defined in the /adi/admin Page (continued)				
ADI Component Name	**Task Name**	**Property Name**	**Default Value**	**Description**
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.AuditService	ADI Service - Cache Size	100	Defines the maximum size of the cache for log files. Increase this value if your log records are run out of memory.
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.DataCollectionService	Data Collection Data Provider - IBM Code Coverage Data - Heap Memory	512M	Defines the maximum heap memory size (by using the -Xmx JVM parameter) to be defined for the data collection jobs that collect code coverage data. Change this value if your data collections report out of memory errors.
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.DataCollectionService	Data Collection Data Provider - Operations Data from OMEGAMON for CICS - Heap Memory	512M	Defines the maximum heap memory size (by using the -Xmx JVM parameter) to be defined for the data collection jobs that collect operations data from IBM OMEGAMON for CICS. Change this value if your data collections report out of memory errors.

Table 8. Variables Defined in the /adi/admin Page (continued)				
ADI Component Name	**Task Name**	**Property Name**	**Default Value**	**Description**
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.DataCollectionService	Data Collection Data Provider - Static Analysis Data from IBM Application Discovery - Heap Memory	512M	Defines the maximum heap memory size (by using the -Xmx JVM parameter) to be defined for the data collection jobs that collect static analysis data from IBM Application Discovery. Change this value if your data collections report out of memory errors.
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.DataCollectionService	Data Collection Data Provider - Data Scrub Interval (hours)	24	The time interval in hours between executions of the scheduled data scrubbing background task.
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.DimezAutoCollectionScheduleTask	ADI Data Collection Schedules Task Fixed Delay (seconds)	900	The time interval in seconds at which ADI checks whether to perform another data collection or not from code coverage or IBM Application Discovery data providers. By default, ADI checks every fifteen minutes. Each scheduled code coverage or IBM Application Discovery data provider checks whether the specified amount of time has passed or not. And if yes, ADI runs a new collection.

Table 8. Variables Defined in the /adi/admin Page (continued)				
ADI Component Name	**Task Name**	**Property Name**	**Default Value**	**Description**
ADI Warehouse Collections and Access	com.ibm.dimez.dashboard.service.internal.OmDataAutoCollectionScheduleTask	ADI Operations Data Collection Schedules Task Fixed Delay (seconds)	300	The time interval in seconds at which ADI checks whether to perform another data collection or not from theOMEGAMON for CICS data provider. By default ADI checks every five minutes. Each scheduled OMEGAMON for CICS data provider checks whether the specified amount of time has passed or not. And if yes, ADI runs a new collection. If you want ADI to check more frequently, you can change the value. However, currently, changing the value is not recommended to ensure that you collect for each 5-minute slot. However, if your IBM OMEGAMON for CICS server is not able to provide all of its data within a 5-minute timeframe, you need to set this value to a larger value that reflects your IBM OMEGAMON for CICS performance.

Generating sample data for evaluation

For ADI, you can generate sample OMEGAMON for CICS data in order to evaluate the ADI functionality.

Generating OMEGAMON for CICS data

You can learn how to generate OMEGAMON for CICS data as sample data to evaluate the ADI functionality.

Complete the following steps to generate OMEGAMON for CICS data.

1. Navigate your browser to ADI admin page `https://<fully qualified hostname>:9753/adi/admin`.
2. Log on with Jazz administrator user ID and password. **Application Administration** page appears.
3. Click **ADI Data Generation** to go to the data generation page.
4. Select **OMEGAMON for CICS** card. The form of **Data Generation Settings** for OMEGAMON for CICS appears.
5. Fill in the following information in **Data Generation Settings**. For more information about terms, see Chapter 2, "Terminology overview," 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.
   - **Project Areas**: Select the project area where you want the data generator to store. By default, it is your current project area.
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.

Chapter 6. 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 ADI 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 ADI).

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 ADI)

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 ADI)

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:

```
//CEEOPTS 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.
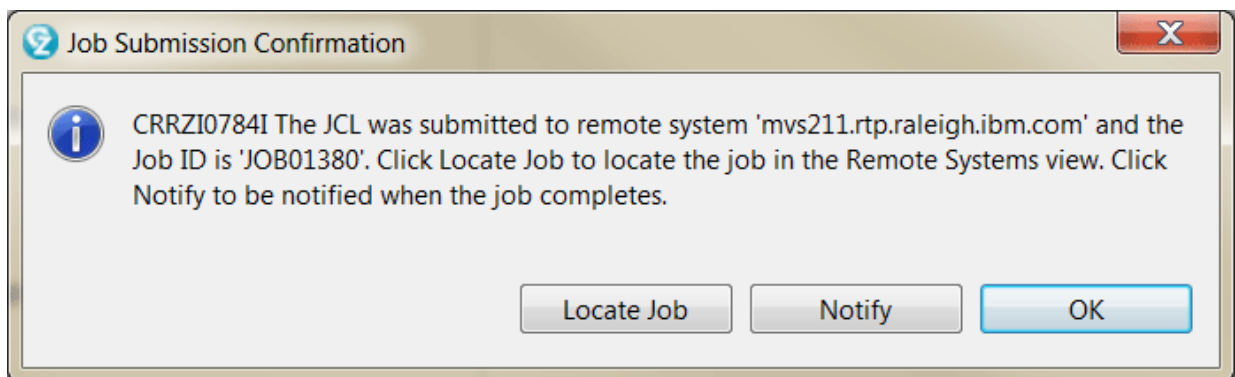
Figure 1. Dialog showing the successful submission of the job

6. On successful execution, view the code coverage results from the current run in the Code Coverage Report.
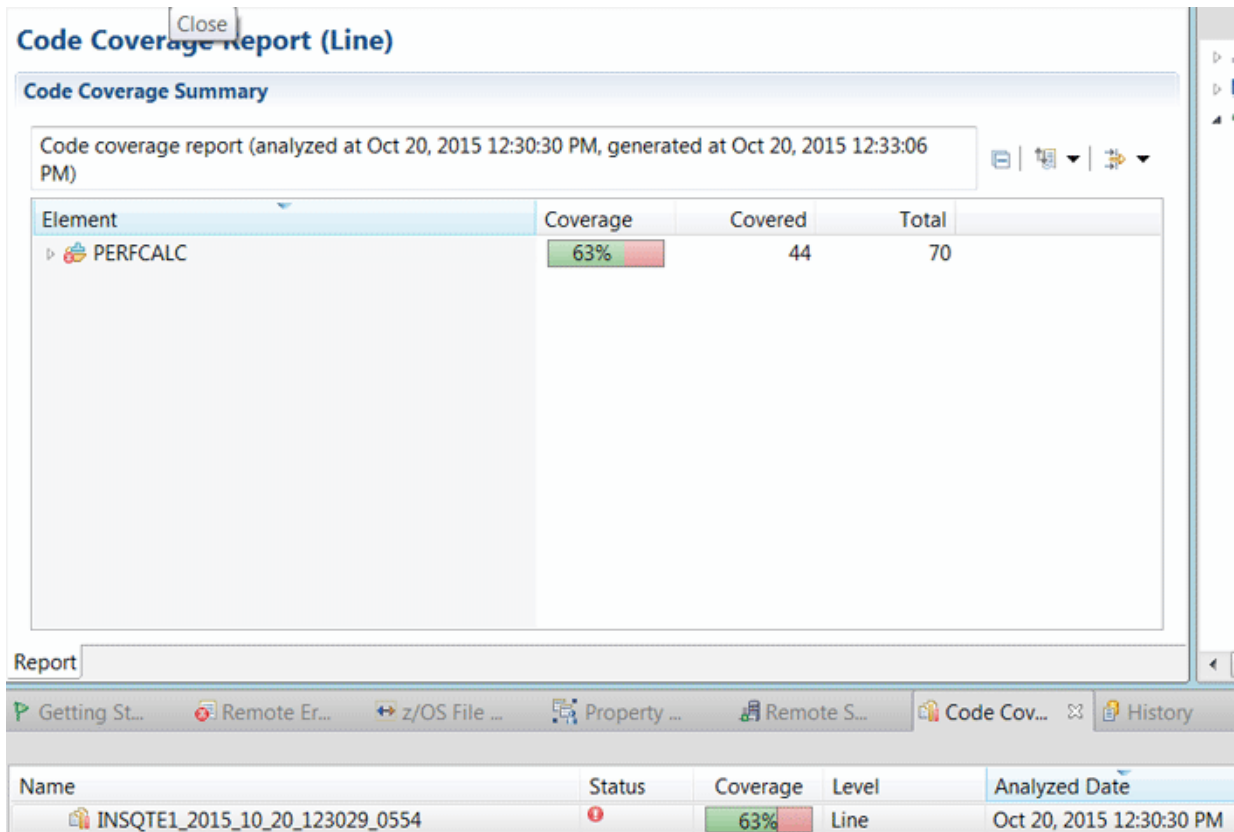
Figure 2. Sample showing the Code coverage result displayed in Code Coverage results view

7. Right click the code coverage result and select **Export**.
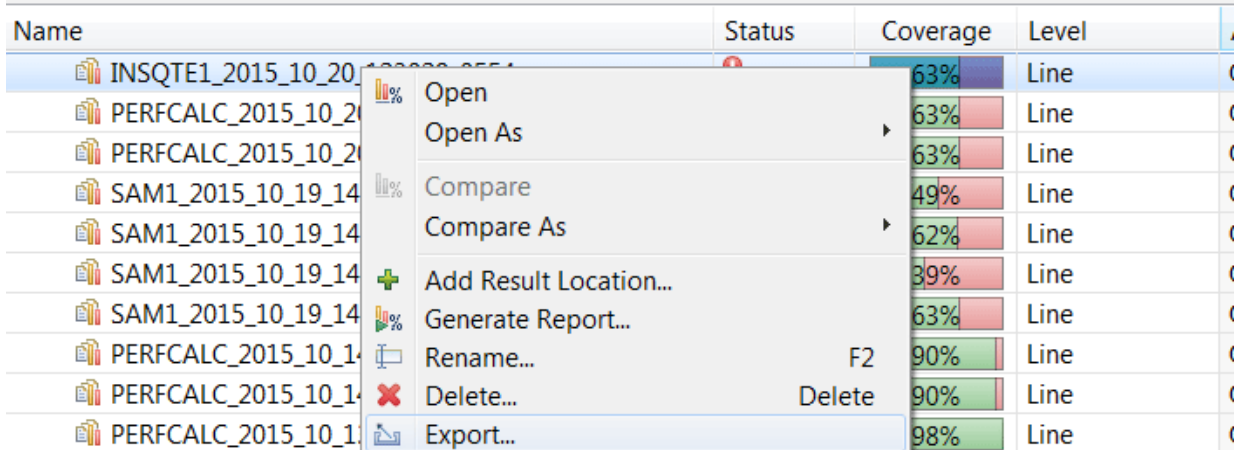
Figure 3. Sample showing the Export menu in Code coverage results view

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.
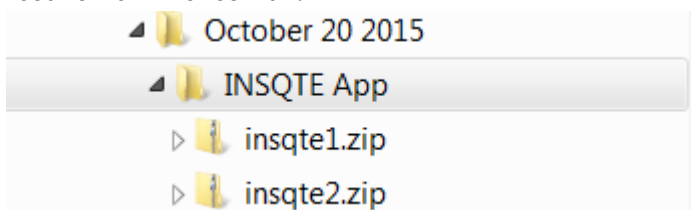
Figure 4. Sample showing file system organization of code coverage results

9. 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 ADI server
You can learn how to generate code coverage results by using the headless collector running on ADI 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 ADI server is started and running. As a part of the ADI 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.

1. Create the data provider by following the instructions in "Adding a Manual Builds data provider to collect data automatically" on page 159.
2. On the **Data Providers** page, click the **Options Menu** icon on the data provider card you created. The options menu appears.
3. Select **Copy Startup Key** from the options menu to copy the startup key to clipboard.
4. 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>")
```

5. 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 ADI 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 ADI 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.
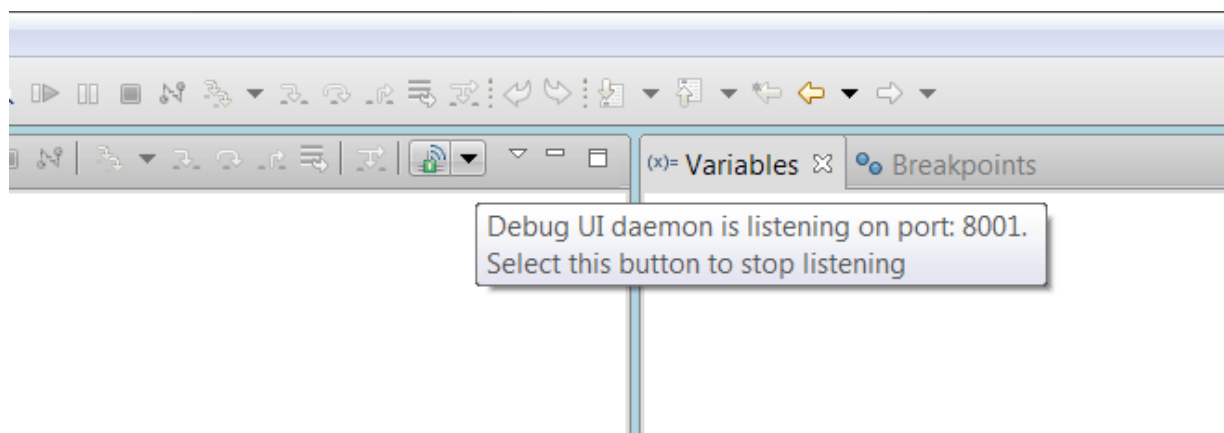
Figure 7. Screen shot showing the verification of debug daemon running

3. Run the transaction, for example, SEPS.

4. Get the code coverage results in the IDz code coverage view.

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 ADI 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 ADI 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 ADI, as ADI 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
<?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>
```

3. 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.

4. 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:

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 ADI.

   a. For information about setting up an ADI data provider that connects to your RTC server and build definition, see "Adding a Rational Team Concert Builds data provider" on page 167.

   b. For how to collect the coverage from the latest build available in ADI, see "Refreshing the builds and code coverage data" on page 168.

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 it owns results file for ADI.
- 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 ADI. For example, only with the sources included can ADI 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 ADI.
   - 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 ADI 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 online help.

6. Right-click the new result and select **Export...** from the context menu. Also, see Import and export results in the RAD 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 ADI 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 ADI.

   - 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 ADI can map the test and compare its current results against it 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 ADI. When you create a new build of your code base and rerun the test to find regression, your 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 ADI can compare the result of the first with the second test.

In ADI:

1. Log on to ADI.

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 ADI 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 ADI 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 ADI 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 ADI, that is, all JUnit tests executed map to
  one ADI 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 ADI test case. And each group results into a separate ADI 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 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 ADI, as ADI 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 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 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 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 ADI, 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 ADI under which the coverage results will be listed.

b. The extension of the file should be .cczip and .zip so that ADI 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 167.

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 ADI.

a. For setting up an ADI data provider that connects to your RTC server and build definition, see "Adding a Rational Team Concert Builds data provider" on page 167

b. For how to collect the coverage from the latest build available in ADI, see "Refreshing the builds and code coverage data" on page 168

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 ADI, 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 detials 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"
classname="com.ibm.rational.llc.engine.instrumentation.anttask.CoverageApplicationAnalyzerT
ask">
                                        <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}.${project.artifactId}.baseline" />
                                    <taskdef name="startBuildActivity"
classname="com.ibm.team.build.ant.task.StartBuildActivityTask">
                                        <classpath>
                                            <fileset dir="${Build_System_HOME}/
buildtoolkit">
                                                <include name="*.jar" />
                                            </fileset>
                                        </classpath>
                                    </taskdef>
                                    <startBuildActivity buildResultUUID="$
{buildResultUUID}"
                                        repositoryAddress="${repositoryAddress}"
userId="${UserId}"
                                        passwordFile="${PasswordFile}" verbose="true"
autoComplete="true"
                                        label="Building and testing $
{project.groupId}.${project.artifactId}." />
                                </target>
                            </configuration>
```

```xml
                                <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" />
                                        <!-- Publish the code coverage results as RTC
contributions -->
                                        <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
downloads -->
                                        <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-probescript=${project.basedir}/${project.groupId}.$
{project.artifactId}.probescript</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 ADI 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 the 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.
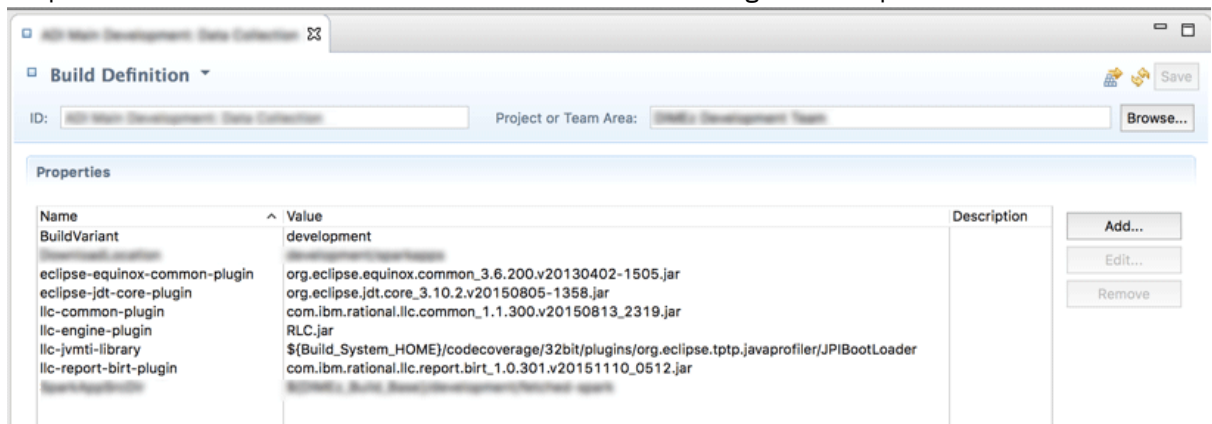
- Now you can run your build. When you finish, verify in RSA or RAD Eclipse client by opening the build result editor that you have coverage results available in a Coverage tab. Check the Downloads tab for a coverage .cczip file available for downloads.

- You are now ready to consume these build results with ADI.

a. For setting up an ADI data provider that connects to your RTC server and build definition, see "Adding a Rational Team Concert Builds data provider" on page 167

b. For how to collect the coverage from the latest build available in ADI, see "Refreshing the builds and code coverage data" on page 168

Preparing code coverage results by using the ADI code coverage generator

You can learn how to use the code coverage generator that ships with ADI to prepare code coverage results for Java.

The ADI 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 ADI to analyze and relate code changes to code coverage results.

2. It generates test result on the JUnit test method level, which gives ADI 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 ADI 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 ADI 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 145.

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

ADI 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 ADI 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

Table 9. Reference of properties in build.properties

Attribute	Description	Required
analyzeArchive	Specify whether to analyze internal achieve 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 achieve, 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: • 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 9. 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: • 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 9. 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: • 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: • 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: • `.metadata` • `.probescript` source (if available) **Note:** This property applies to only the build.properties file in the `JavaProjectAnalyzer` directory.	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 9. Reference of properties in build.properties (continued)

Attribute	Description	Required
serverURL	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. For more information about starting the application server, see "Generating Java code coverage by using the ADI code coverage on server generator" on page 151. An ServerURL example: `http://localhost:9080`. **Note:** This property applies to only the build.properties file in the `JavaProjectAnalyzer` directory.	No. This value is empty by default. If either serverURL or projectAnalysisResultZip is empty or not valid, no project analysis result artifacts will be uploaded to server.
serverAppNames	Specify a list of applications to configure WAS/Liberty server to collect code coverage exclusively for these applications. **Note:** This property applies to only the build.properties file in the JavaProjectAnalyzer directory.	No. This value is empty by default. 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.
testCase	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.	Yes

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 ADI code coverage on server generator.

Note: If the javacc service is not available on the specify server, no operations will be performed.

Table 10. 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 exists. 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 ADI with a first build. 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 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 ADI 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 ADI 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 ADI 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.
- ADI is installed as described in Installation and setup. You are granted with read access to the ADI server machine and the ADI installation directory.
- RTC Build and ADI 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 ADI 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 ADI server machine, find the ADI 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 ADI 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 ADI 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.

After you generate the code coverage results, you can find the next steps to perform more operations.

In ADI, 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 ADI code coverage on server generator

You can learn how to use the ADI code coverage on server generator to generate Java code coverage and project metadata.

Generating Java code coverage by using the ADI code coverage on server generator
You can use the ADI 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 ADI for further analysis.

Assumptions
Verify the assumptions before you generate Java code coverage by using the ADI code coverage on server generator.

Make sure that the ADI 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 ADI code coverage on server generator to generate Java code coverage.

Complete the following steps to prepare Java code coverage by using the ADI 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 ADI code coverage generator" on page 153. 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 ADI code coverage generator" on page 153.

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 153.

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 ADI. 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 158 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 ADI code coverage generator

Assumption

Verify the assumption before you generate project metadata by using the ADI 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 ADI 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 145.

Chapter 7. Managing connections, data providers, workbooks, and notifications

You can learn how to manage connections, data providers, workbooks, and notifications.

For the terminology of data providers and workbooks, see Terminology Overview page.

Managing connections

For a data provider that stores data on the server, you must manage connections associated with the data provider in order to connect the server for data collection. By default, the users with data provider administrator role can add, edit, and delete connections from the system.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Adding 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. Check for the steps to add a connection.

Go to the Connections page and create a connection to connect the data provider server. Currently, ADI offers two types of connections, including IBM Rational Team Concert and IBM OMEGAMON for CICS.

Notes:

- You need a data provider administrator role permission in order to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.
- Application Discovery connection is no longer available as part of ADI. If you migrated from a previous version of ADI, which has Application Discovery connections, the AD connections are still stored in ADI warehouse but are not displayed. As a result, if you are creating a new connection with the same name as the existing Application Discovery connections from the previous release of ADI, you will receive an error message that there is a duplicate name.

Complete the following steps to create a connection.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Connections** 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 **Add Connection** form.

3. Click **Create Connection**. The Add Connection page appears.
4. Provide the following information in the **Add Connection** form.

   - **Connection type**: Select type of connection you want to create. Currently, IBM Rational Team Concert and IBM OMEGAMON for CICS are the two types of connections that ADI offers.
   - **Name**: Provide a unique name of the connection.
   - **Description**: Provide short description of the connection.
   - **Connection URL**: Provide the URL address of the location you want to connect to. For IBM Rational Team Concert, you need to provide the address of root service.

     Examples of URL addresses for different data providers:

     - IBM Rational Team Concert - `https://hostname:9753/jazz`
     - IBM OMEGAMON for CICS - `http://hostname:15200`

- **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.

5. Click **Test** to test the connection.

6. If the connection is a success, click **Add** to create the connection.

Editing a connection

You can edit the information of a connection after you create at least one connection. Check for the steps to edit a connection.

Note: You need a data provider administrator role permission in order to edit the connection information. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to edit the connection information:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Connections** to go to the **Connections** page.

3. Click the **Menu** icon on the header of the provider card you want to edit. You can search the connection card by typing part of the connection name in the search box on the upper right corner. The connection cards with corresponding name appear.

4. Click **Edit**.

5. Edit the information such as name, description, connection URL, username, and password or all the information. If you update connection URL, you are required to retest the connection before you save the changes.

6. Click **Save** to update the changes or click **Cancel** to cancel the changes.

Deleting a connection

Check for the steps to delete a connection from the system. When you delete a connection, you cannot undo the deletion or restore the connection. You cannot delete a connection when it is being used by data providers. Before you delete a connection, you need to make sure that you deleted the data providers that use the connection.

Note: You need a data provider administrator role permission in order to delete a connection. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to delete a connection.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Connections** to go to the Connections page.

3. Click the **Menu** icon on the header of the provider card you want to delete. You can search the connection card by typing part of the connection name in the search box on the upper right corner. The connection cards with corresponding name appear.

4. Click **Delete**. The dialog box to confirm that the connection deletion opens.

5. Click **OK** to confirm the deletion or **Cancel** to cancel the deletion.

Managing data providers

The data provider is the data source for analysis. By default, only the users with a data provider administrator role can create, modify, and delete data providers.

The following types of data sources are available for ADI:

- Manual Builds data providers
- Rational Team Concert Builds data providers
- OMEGAMON for CICS data providers

- System Management Facility data providers

Notes:

- The Application Discovery data provider and the Business Rule Discovery data provider are no longer available as part of ADI. If you migrated from a previous version of ADI, which has the Application Discovery data provider, the Business Rule Discovery data provider, or both, the data providers data are still stored in ADI warehouse but are not displayed. As a result, if you are creating a new data provider with the same name as the existing Application Discovery data provider or Business Rule Discovery data provider from the previous release of ADI, you will receive an error message that there is a duplicate name.

- The Enterprise Artifact data provider, which is available as a technical preview feature, is deprecated in ADI V5.1.0 and later versions. The enterprise artifacts data can be collected through Business Rule Discovery data providers instead. If you have one or more Enterprise Artifact data providers that are created, you need to back up your enterprise artifacts from the location that is specified in the Enterprise Artifact data providers and delete these data providers before you upgrade to ADI V5.1.0.

To manage data providers, you need to have specific role-permissions. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

In addition to the role-based permissions, there is an access control for a workbook in which only the members of a workbook can view the analysis data of the data providers that are associated with a workbook.

To improve the efficiency and performance of data collection, ADI automatically scrubs historical data at certain intervals (for example, every 24 hours) in your data warehouse. There are two aspects of data scrub configuration:

- Specify the number of days that you want to keep data in the data warehouse for analysis in the data provider setting. You need to have a data provider administrator role permission to perform this task.

- Configure the interval time for the data scrub task to run automatically in the ADI administration page. The default value is 24 hours for ADI to run the data scrub task in the background every 24 hours. You need to be an ADI administrator to perform this task. For more information, see "Server settings that can be defined at runtime in the /adi/admin user interface" on page 121.

Data providers

To get started with ADI, first you need to define the data providers to collect data from the data sources for analysis. Currently, ADI offers several types of data providers, that is, Application Discovery data providers, Business Rule Discovery data providers, Manual Builds data providers, OMEGAMON for CICS data providers, Rational Team Concert Builds data providers, and System Management Facility 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 127.

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 ADI 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 ADI 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 ADI for performance analysis such as CPU time, Execute Channel Program time, Service units, and Elapsed 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.

Only the users with the data provider administrator role permission can add and modify the Manual Builds data provider. The users with the workbook owner role permission can only modify the Manual Builds data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permission in order to add a Manual Builds data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a Manual Builds data provider for manual data collection.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click **Create Data Provider** on the upper right of the Data Providers page.
4. Select the **Manual Builds** card from the Add Data Provider page that appears.
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 build 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 127 and "Preparing

code coverage results for CICS transactions by using the debug tool backend and IDz client" on page 131 .

- **Data Provider Details** form.

  – **Name**: Provide a unique name for the data provider you create.

  – **Description**: Provide a short description of the data provider.

  – **Enable headless collection support**: Make sure the checkbox is unchecked.

  – 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.

6. Click **Add** 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.

Notes:

1. 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.

2. You need a data provider administrator role permission in order to add a Manual Builds data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a Manual Builds data provider for automated data collection:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click **Create Data Provider** on the upper right of the Data Providers page.

4. Select the **Manual Builds** card from the Add Data Provider page that appears.

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 build 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. For procedure to set up the automated data collection, see "Generating code coverage results by using the headless collector running on ADI server" on page 130.

   - **Data Provider Details** form.

     – **Name**: Provide a unique name for the data provider you create.

     – **Description**: Provide a short description of the data provider.

     – **Enable headless collection support**: Check to enable headless collection support. A dropdown menu of **Collection Trigger** 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.

- 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.

6. Click **Add** 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 6, "Preparing external data sources," on page 127, you need to refresh the data provider in order to collect data from the headless-cc-file directory. For more information, see "Refreshing the code coverage results for automatically data collection" on page 161.

- If you have not put the code coverage data collection in the headless-cc-file directory, follow the steps described Chapter 6, "Preparing external data sources," on page 127 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. ADI prevents the provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

By default, the users with data provider administrator role permission modify the Manual Builds data provider. The users with the workbook owner role can add and modify builds information. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Editing the Manual Builds data provider information
You can edit name, description, or both of the data provider.

Note: By default, you need a data provider administrator role permission to edit the data provider information. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to edit the data provider information:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to edit.

4. Select **Edit** from the options menu that appears. The Edit Data Provider form appears.

5. Modify the data provider details, such as name, description, connection and so on.

6. 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.

7. Modify number of days you want to keep data in the data warehouse.

8. 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.

Note: By default, you need either a data provider administrator role or a workbook owner role to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a build to a Manual Builds data provider which collects data manually.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to add a build to.

4. Select **Add Builds** form the options menu that appears. The Add Build page appears.

5. 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.

6. 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.

Note: You need either a data provider administrator role or a workbook owner role to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a build to a data provider which collects data automatically.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to add a build to.

4. Select **Add Builds** form the options menu that appears. The Add Build page appears.

5. 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 ADI to collect data automatically. See "Generating code coverage results by using the headless collector running on ADI server" on page 130 for more information.

6. 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 "Refreshing the builds and code coverage data" on page 168.

Refreshing the code coverage results for automatically data collection
When the code coverage results are available in headless-cc-file folder of ADI, 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 ADI server" on page 130.

Note: You need either a data provider administrator role or a workbook owner role to refresh the code coverage results for automatically data collection. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to refresh the code coverage results for collect code coverage data from headless-cc-file folder.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to refresh the code coverage results.

4. Select **Refresh** from the options menu that appears. The Data Collection dialog box opens.

5. Click **OK** to perform data collection or **Cancel** to cancel the operation.

Updating an existing build

You can update an existing build by modifying the build information.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to update an existing build associated with the data provider:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to update the build information.

4. Select **View Builds** from the options menu that appears. The Builds page with a list of all builds associated with the data provider card appears.

5. On the row which displays a build that you want to modify, click the **Edit** icon.

6. 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.

7. 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. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to define baseline builds.

4. Select **View Builds** form the options menu that appears. The **Builds** page that lists all builds that are associated with the Manual Builds data provider card appears.

5. 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.

6. Click **Save baselines builds**. The dialog box to confirm the save operation opens.

7. Click **Start Data Collection** for ADI 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. ADI will send you a notification when data is ready to use.

8. 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.

Note: By default, you need either a data provider administrator role or workbook owner role to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to delete a build from its associated Manual Builds data provider.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to delete a build from.

4. Select **View Builds** form the options menu that appears. The Builds page that lists all builds that are associated with the Manual Builds data provider card appears.

5. 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.

6. 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.

Note: By default, the users with any role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to download the code coverage data to a `.csv` file or `.JSON` file.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Manual Builds data provider card that you want to download the code coverage data.

4. Select **Download** from the options menu that appears. The Download Data page appears.

5. 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.

6. Click **Download** to download or **Cancel** to cancel the download action. When you click **Download**, the download box appears.

7. Select the **Save File** option and click **OK**.

8. 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.

Table 11. The list of files in the downloaded 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.

Only the users with data provider administrator can add and edit the OMEGAMON for CICS data provider. Any of the role permissions can view data collection logs and download OMEGAMON for CICS data. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permission in order to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add an OMEGAMON for CICS data provider for data collection.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click **Create Data Provider** on the upper right of the Data Providers page. The Add Data Provider page appears.

4. Select the **OMEGAMON for CICS** card.

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 "Adding a connection" on page 155.

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 or delete. Then select **Edit** or **Delete**.

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 in the **Data Provider Details** section.

   - **Name**: Provide a unique name for the data provider you create.
   - **Description**: Provide a short description of the data provider.
   - **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 settings that can be defined at runtime in the /adi/admin user interface" on page 121.

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. ADI prevents the data provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

Only the users with a data provider administrator role can edit the data provider information. Any of the role permissions can view the data collection logs for OMEGAMON for CICS data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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. ADI prevents the provider to be updated by multiple users. The page rejects all but one of many parallel updates by multiple users.

Note: You need a data provider administrator role permission in order to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to edit the OMEGAMON for CICS data provider information.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the OMEGAMON for CICS data provider card that you want to edit.

4. Select **Edit** from the options menu that appears. The Edit Data Provider form appears.

5. Modify the name, description, data collection, or all information of the data provider details.

   Note: The Connection for the data provider cannot be modified.

6. 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.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to view OMEGAMON for CICS data collection.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the OMEGAMON for CICS data provider card that you want to view the data collection logs.
4. Select **View Logs** from the options menu that appears. The list of the date and time data appears. The logs are group by the hours when they are collected.
5. 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 settings that can be defined at runtime in the /adi/admin user interface" on page 121.

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.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to download the OMEGAMON for CICS data to a `.csv` file or `.json` file.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the OMEGAMON for CICS data provider card that you want to download data.
4. Select **Download** from the options menu that appears. The **Download Data** page appears.
5. 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.

6. Click **Download** to download or **Cancel** to cancel the export option. When you click **Download**, the download box appears.
7. Select the **Save File** option and click **OK**.
8. 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.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permission to add a Rational Team Concert Builds data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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 132.

Procedures

Take the following steps to add a Rational Team Concert Builds data provider.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click **Create Data Provider** on the upper right of the Data Providers page.

4. Select the **Rational Team Concert Builds** card from the Add Data Provider page that appears.

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 the **Add** icon next to the Connection header to add a connection. For more information, see "Adding a connection" on page 155.

   Note: You can also edit connection information or delete connection from here by clicking the **Options Menu** icon on the connection card header that you want to edit or delete. Then select **Edit** or **Delete** from the options menu.

6. Click **Connect** to fetch build and code coverage data from RTC server.

7. When the **Project Area** appears, select **Project Area** you want the data provider to collect the data from the drop down list.

8. **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 right corner of **Build Definitions** section.

9. 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.

10. By default all files available for Download in an RTC build result that end with .cczip will be downloaded and analyzed by ADI. 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 check box 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.

11. Provide the following information in the **Data Provider Details** forms.

    • **Name**: Provide a unique name for the data provider you create.

    • **Description**: Provide a short description of the data provider.

    • **Collection Trigger**: 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 117.

12. Click **Add** to create the Rational Team Concert Builds data provider.

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. ADI prevents the data provider update by multiple users. The page rejects all but one of many parallel updates by multiple users.

To edit or delete the Rational Team Concert builds provider, you need to have a data provider administrator role permission.

To refresh the builds data, you need to have either a data provider administrator role permission or a workbook owner role permission.

Any of the role permissions can view the builds data and download the code coverage data.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permissions to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to edit the Rational Team Concert Builds data provider information:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to edit.
4. Select **Edit** icon from the options menu that appears. The Edit Data Provider page appears.
5. 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
6. 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.
7. Click **Save** to update the changes or click **Cancel** to quit editing.

Refreshing 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.

Note: You need either a data provider administrator or a workbook owner role to perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to refresh the builds and code coverage data.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to refresh builds and code coverage data.

4. Select **Refresh** from the options menu that appears. The Data Collection dialog box opens.

5. Click **OK** to perform manual refresh or **Cancel** to cancel the operation.

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.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to view builds data.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to view builds data.

4. Select **View Builds** from the options menu that appears. The Builds page that lists all builds that are associated with the data provider card appears.

5. Select the **Check** icon on the right 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 settings that can be defined at runtime in the /adi/admin user interface" on page 121.

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. Click the **Menu** icon next to the IBM ADI header to open the menu pane.

2. Click **Data Providers** to go to the **Data Providers** page.

3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to define baseline builds.

4. Select **View Builds** form the options menu that appears. The **Builds** page that lists all builds that are associated with the Rational Team Concert Builds data provider card appears.

5. 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.

6. Click **Save baselines builds**. The dialog box to confirm the save operation opens.

7. Click **Start Data Collection** for ADI 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. ADI will send you a notification when data is ready to use.

8. 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.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to delete a build from its associated data provider.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to delete a build from.
4. Select **View Builds** from the options menu that appears. The Builds page that lists all builds that are associated with the data provider card appears.
5. 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.
6. 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.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to download the code coverage data to a .csv file or .json file.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the Rational Team Concert Builds data provider card that you want to download.
4. Select **Download** from the options menu that appears. The **Download Data** page appears.
5. 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.
6. Click **Download** to export or **Cancel** to cancel the export option. When you click **Download**, the download box appears.
7. Select the **Save File** option and click **OK**.
8. 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.

Table 12. The list of files in the downloaded 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 System Management Facility data providers

You can manage System Management Facility data providers on the **Data Providers** page. Only the users with a data provider administrator role can add and edit the System Management Facility data provider. Any of the role permissions can refresh the Application Discovery data, view data collection logs, and download Application Discovery data.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permission to add a System Management Facility data provider.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a System Management Facility data provider for performance analysis.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click **Create Data Provider** on the upper right of the **Data Providers** page.
4. Select the **System Management Facility** card from the **Add Data Provider** page that appears.
5. Enter ADI user credentials to be used for authenticating distributed data collection of the provider.
6. Provide the following information in the **Data Provider Details** section.

   - **Name**: Provide a unique name for the data provider you create.
   - **Description**: Provide a short description of the data provider.

7. Click **Add** to create the System Management Facility data provider.
8. Import the System Management Facility and Application Performance Analyzer files in the **Import Data** form.

   - **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, ADI 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.

9. Click **Import** to import the data or click **Cancel** to cancel the import.

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. ADI prevents the data provider from updating by multiple users. The page rejects all but one of many parallel updates by multiple users.

Only the users with a data provider administrator role can edit the data provider information. The data provider administrator or the workbook owner can import SMF and APA data.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

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.

Note: You need a data provider administrator role permission to edit a System Management Facility data provider. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to edit the System Management Facility data provider information.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the System Management Facility data provider card that you want to edit.
4. Select **Edit** from the options menu that appears. The **Edit Data Provider** form appears.
5. Update the data provider setting information such as the name or the description.
6. Click **Save** to update the data provider information or click **Cancel** to quit editing.

Viewing the System Management Facility data collection logs
You can view the status of the System Management Facility data collection.

Note: Any of the role permissions can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to view the System Management Facility data collection logs.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the System Management Facility data provider card that you want to view the data collection logs.
4. Select **View Logs** from the options menu that appears. The list of the date and time data that are collected appears. The logs are grouped by the hours when they are collected.

   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 settings that can be defined at runtime in the /adi/admin user interface" on page 121.

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. ADI prevents the provider to be uploaded by multiple users. The page rejects all but one of many parallel uploaded by multiple users.

Note: You need a data provider administrator role or workbook owner permission to import the SMF and APA data. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to import the SMF and APA data:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the System Management Facility data provider card that you want to import the SMF and APA data.
4. Select **Import** from the options menu that appears. The Import Data dialog box opens.
5. 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, ADI 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.

6. Click **Import** to import the files or click **Cancel** to quit the import.

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.

Note: You need a data provider administrator role permission in order to perform this task.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to delete a data provider.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Data Providers** to go to the **Data Providers** page.
3. Click the **Options Menu** icon on the header of the data provider card that you want to delete.
4. Select **Delete** from the options menu that appears. The dialog box to confirm the data provider deletion opens.
5. Click **OK** to confirm the deletion or **Cancel** to cancel the deletion.

Managing workbooks

To perform data analysis, you can associate data providers with a workbook in the **Workbooks** view. By default, the users with a data provider administrator role or a workbook owner role can add workbooks to the system. To edit information of a workbook or delete a workbook, the users need to be a workbook owner. As the members of the workbook, users can only view the reports and dashboards of that workbook.

For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Note: The static analysis and business rule discovery features are no longer available as part of ADI. If you migrated from a previous version of ADI, which has workbooks that are associated with either Application Discovery data provider or Business Rule Discovery data provider, the workbooks are displayed with additional unsupported message at the end of workbook name. You cannot view or modify the static analysis or business rule discovery data from these workbooks. Both data providers data are still stored in ADI warehouse but are not displayed. As a result, if you are creating a new data provider with the same name as the existing Application Discovery data provider or Business Rule Discovery data provider from the previous release of ADI, you will receive an error message that there is a duplicate name.

Workbooks

ADI 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.

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.

Note: By default, only the users with the data provider administrator or workbook owner role permission can perform this task. For more information about roles and permissions for ADI, see "Assigning role-based permissions to ADI users" on page 115.

Complete the following steps to add a workbook.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Workbooks** to go to the Workbooks page.
3. Click **Create Workbook** on the upper right of the **Workbooks** page.
4. Provide the following information on the **Add Workbook** page.

   - **Name**: The workbook name.
   - **Description**: A brief description of the workbook.
   - **Contributors**: Only the contributors of the workbook can modify, view, and analyze the Analysis workbook. You can find the list of all members that are associated with the project area in the left box. For information about how to set up project area, see Installation and Set-up page.
     - Set owners of the workbook: A workbook owner can modify and delete the workbook. By default, the user who creates a workbook will be assigned as an owner of that workbook. There can be more than one owner for a workbook.

       a. Select one or more names of the workbook owners from the left box.

       b. Select the **Owner** role from the right box.

       c. Click **>** icon to add owners to the contributor list.

     - Set members of the workbook: A member of the workbook can view and analyze the workbook.

       a. Select one or more names of the workbook members from the left box.

       b. Select **Member** role from the right box.

       c. Click **>** icon to add members to the contributor list.
   - **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. Then one or more sections of data providers settings for each of the data providers you selected appear.
   - **Data Providers Setting**: 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**. ADI 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**, ADI takes all the data collected during past week to calculated the information that displays on the OMEGAMON for CICS analysis views. The default timeframe is **Last Day**.

5. Click **Add** to add the workbook or **Cancel** to cancel adding the workbook.

Modifying a workbook

You can edit the workbook information, update contributors, and update data providers of the workbook.

Note: You need to be assigned as an owner of the workbook to modify that workbook.

Modifying a workbook

Complete the following steps to modify a workbook:

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Workbooks** to go to the Workbooks page.
3. Click the **Options Menu** icon on the header of the Analysis workbook you want to modify information.
4. Select **Edit** from the options menu that appears. The Edit Analysis Workbooks page appears.
5. Modify the workbook information. You can perform one or more of the following activities.
   - Modify name, description, or both of the workbook from the **Name** and **Description** field.
   - Add owners of the workbook.

     a. Select one or more names of the workbook owners from the left box.

     b. Select **Owner** role from the right box.

     c. Click **>** icon to add owners to the contributor list.

   - Remove owners of the workbook.

     a. Select one or more names of the workbook owners from the right box.

     b. Click **<** icon to remove owners from the contributor list.

   - Add members of the workbook.

     a. Select one or more names of the workbook members from the left box.

     b. Select **Member** role from the right box.

     c. Click **>** icon to add members to the contributor list.

   - Remove members of the workbook.

     a. Select one or more names of the workbook members from the right box.

     b. Click **<** icon to remove members from the contributor list.

   - Update the threshold.

     – Slide the right and left bars to change the threshold level.

   - Update 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.

6. 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 which you are interested in or often work with.

Complete the following steps to pin a workbook:

1. Go to the **Workbooks** view. By default, you are on the **Workbooks** view after you log in ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.
2. Click **Options Menu** icon on the header of workbook you want to pin.
3. Select **Pin** from the drop-down list that appears. The Workbooks view reloads with the pinned workbook under the **Pinned Workbook** section.

     Note: To unpin the workbook, click **Options Menu** icon and select **Unpin**.

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 don't delete the data providers associated with that workbook.

Note: You need to be assigned as an owner of the workbook to be able to delete the workbook.

Complete the following steps to delete a workbook.

1. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
2. Click **Workbooks** to go to the Workbooks page.
3. Click the **Options Menu** icon on the header of the workbook you want to delete.
4. Select **Delete** from the options menu that appears. A confirmation dialog appears.
5. Click **OK** to delete the workbook or **Cancel** to cancel the deletion.

Preserving user settings

When you work with ADI, you can preserve user settings as an individual user.

For more information about the user settings you can preserve, see the following list:

• Selection of the user role on the Welcome page

• Pinned workbook by a user

• The log messages and notifications that are read by a user

• The order of workbook listed on the **Workbooks** page

     Note: A user can select how the workbook is sorted on the Workbooks page by selecting the **Sort By** drop-down list on this page. The last used sorting option is saved for that user.

- The last used filters for Code Coverage analysis views

Managing notifications

Notifications are messages that inform users about the data collection status or metrics violation.

For the notifications about data collection status, ADI displays the following log messages of data collection activities for all the data providers within your ADI.

- Manual Builds data providers: All log messages from data collection.
- Rational Team Concert data providers: All log messages from scheduled data collection.
- OMEGAMON for CICS data providers: Only warnings and errors messages of data collection.

Note: This type of notifications cannot be turned off.

Viewing notifications

You can view the details for notifications of data collection status and metric violation.

To view notifications, complete the following steps:

1. Log in to ADI. After you log in to ADI, you can see the **Notifications** (bell) icon on the upper right of the page next to your **Profile** icon. The **Notifications** icon is displayed along with the number of new notifications since you check notifications last time.

2. Select the **Notifications** icon to view the list of ten latest notifications. From the notifications list, the latest notifications are highlighted.

3. Click any of the notifications to go to the detailed information.

   - For notification of data collection status, the Logs message page of that data provider is displayed.
   - For notification of metric violation, the trend line of the metric that is violated is displayed.

4. Select **View all notifications** on the bottom to go to the **Notifications** page. The **Notifications** page appears with all notifications that are grouped by hours they occur.

For notification of data collection status, most of the notifications are the same as the log messages. However, some of the messages on the **Logs** page are too long to be displayed on the notification list. ADI displays short forms of these messages as follows.

Notification messages	Log messages
CRIDA0344E Error in the data collection for automation result %s.	CRIDA0062E There was an error in the code coverage data collection process automation result %s. Error message returned: %s. Contact your administrator to check the log file on the server at /server/conf/adi/logs/adi-collections.log for details.
CRIDA0351W Ended with a warning for the last data collection.	CRIDA0095W The last data collection ended with a warning: %s. Contact your administrator to check the log file on the server at /server/conf/adi/logs/adi-collections.log for details.
CRIDA0350E Ended with an error for all the automation results in last data collection.	CRIDA0094E The last data collection ended with an error for all the automation results in this data provider. Contact your administrator to check the log file on the server at /server/conf/adi/logs/adi-collections.log for details.

Notification messages	Log messages
CRIDA0353E No valid regular expression for the code coverage data files.	CRIDA0068E The automated data provider %s does not specify a valid regular expression for the code coverage data files. Please correct the problem, delete this automation result, and run the collection again.
CRIDA0352E No code coverage file was found for automation result %s.	CRIDA0067E No code coverage file was found for automation result %s. Please verify that this build contains such files in the build record and that it matches the regular expression specified for the provider.
CRIDA0349W Data Collection was skipped.	CRIDA0093W The data received from IBM OMEGAMON for CICS was older or the same as the latest data already available in the warehouse. Data Collection was skipped.
CRIDA0348W Data maybe missed because last data collection took too long.	CRIDA0102W The last data collection took more time than the time interval between two collections, which can result in data being missed during an auto collection. The recommendation is to either increase the fixed delay between two collections in the ADI Administration page or collect data for smaller number of plexes.
CRIDA0340E The last data collection ended with an error.	CRIDA0082E The last data collection ended with this error: %s.

Note: %s in the table represents the message strings that ADI generates based on different message types. For example, %s could be the name of a build or a data provider.

Chapter 8. Analyzing workbooks

You can access the analysis dashboards and reports from the **Workbooks** view on ADI home page. The **Workbooks** view is the view that shows the status of all the workbooks within the system that the user has access to.

For each workbook, it shows its latest data analysis results from their associated data provider. For example, for the workbooks associated with Manual Build data provider, the view displays the summary report of code coverage of all builds within the data provider.

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.

Note: Starting from ADI V5.1.0.6, the static analysis and business rule discovery features are deprecated. Both features are integrated into ADDI as the ADDI Extension component. For more information, see IBM ADDI Extension User Guide.

Role-based dashboards and reports for a workbook

Before you can view the analysis results of a workbook, ADI requires you to select your role to start the analysis. Three roles are available: Development, Test, and Operations. The order of dashboards reports displays differently based on the role you select.

Development role

This role is for people whose focus is on software implementation. You can find the order of tabs on the dashboard and reports that are displayed within a tab for the development role in the following list.

- The order of tabs that are displayed on the dashboard is as follows.
  - **Code Coverage**
  - **Performance**
- The order of reports within a tab is as follows.
  - **Code Coverage**:

    The code coverage reports are displayed in the same order for all roles.

    For report details, see "Code coverage reports and information" on page 183.
  - **Performance**:

    The reports within the **Performance** tab are in the following order.
    - Transactions with Performance and Reliability Issues
    - Average Response Time
    - Average CPU Time
    - Average Transactions

    For report details, see "OMEGAMON for CICS related reports and information" on page 198.

Test role

This role is for people whose focus is on software testing. You can find the orders of tabs on the dashboard and reports that are displayed within a tab for the test role in the following list.

- The order of tabs that are displayed on the dashboard is as follows.
  - **Code Coverage**
  - **Performance**
- The order of reports within a tab is as follows.

- **Code Coverage**:

  The code coverage reports are displayed in the same order for all roles.

  For report details, see "Code coverage reports and information" on page 183.

- **Performance**:

  The reports within the **Performance** tab are in the following order.

  - Transactions with Performance and Reliability Issues
  - Average Response Time
  - Average CPU Time
  - Average Transactions

  For report details, see "OMEGAMON for CICS related reports and information" on page 198.

Operations role

This role is for people whose focus in on monitoring the performance of the operation server. You can find the order of tabs on the dashboard and reports that are displayed within a tab for the operations role in the following list.

- The order of tabs that are displayed on the dashboard is as follows.
  - **Performance**
  - **Code Coverage**
- The order of reports within a tab is as follows.
  - **Performance**:

    The reports within the **Performance** tab are in the following order.

    - Average Response Time
    - Average CPU Time
    - Average Transactions
    - Transactions with Performance and Reliability Issues

    For report details, see "OMEGAMON for CICS related reports and information" on page 198.
  - **Code Coverage**:

    The code coverage reports are displayed in the same order for all roles.

    For report details, see "Code coverage reports and information" on page 183.

Note: The tabs and reports that are related to a data provider are displayed only when you associate that type of data provider with the workbook that is analyzed. For example, if you do not associate the OMEGAMON for CICS data provider with a workbook that is analyzed, the **Performance** tab and reports within the **Performance** tab will not be displayed.

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 Chapter 7, "Managing connections, data providers, workbooks, and notifications," on page 155.

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 overview," on page 3.

- **Workbooks** view

  The **Workbooks** view is the home page of ADI.

  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 183.

- **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 183.

  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 build and the details of the files within the build that have insufficient code coverage. You can click the upside-down triangle icon next to the current build id to select other builds available. 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 following 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. For more information about how to filter data that is displayed in the view, see "Filtering data in the Build Analysis view" on page 193.

  To view the code coverage results by source files, pinned files, or test files, 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).

  – 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 183

- **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 "Filtering data in the Flowpoints Analysis view" on page 194.
  – 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 Lines: The number of executable lines that are covered.
    - Total Exec. 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 183.

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

Table 13. Code coverage reports		
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
Code coverage	Two metrics are provided in the report. • 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

Table 13. Code coverage reports (continued)		
Reports Name	**Description**	**Displayed Location**
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

Note: The order of the reports that are displayed in the **Summary** view of a workbook depends on your role. For more information, see "Role-based dashboards and reports for a workbook" on page 179.

Information that is displayed in Build Analysis Views

Table 14. Information that is displayed in Build Analysis views			
View	**Displayed Location**	**Information**	**Description**
View by Source Files	Summary for Entire Build	Warning	The code coverage status of the build. There are two statuses: red and yellow. • Yellow indicates the poor code coverage percentage. • Red indicates the insufficient code coverage percentage. The threshold of red and yellow statuses are set during the workbook creation. For more information, see "Managing workbooks" on page 173.
		Code Coverage	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.

Table 14. Information that is displayed in Build Analysis views (continued)			
View	**Displayed Location**	**Information**	**Description**
		Change%	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. **Note:** You can select baseline build to compare from the **Compare With** drop-down menu on th top.
		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.
	Hover-over Summary For Entire Build Section	Covered / Total Executable Lines	The number of executable lines that are exercised by the test and number of total executable lines within the entire build.

Table 14. Information that is displayed in Build Analysis views (continued)			
View	**Displayed Location**	**Information**	**Description**
	Table	Item Name	The name of the item within the build. The item can be source file, program, package, flowpoints, and method.
		Warnings	The code coverage status of the build. There are two statuses: red and yellow. • Yellow indicates the poor code coverage percentage. • Red indicates the insufficient code coverage percentage. The threshold of red and yellow statuses are set during the workbook creation. For more information, see "Managing workbooks" on page 173.
		Code Coverage	The code coverage percentage of the item displayed. For more information, see Chapter 2, "Terminology overview," on page 3.
		Change %	The difference in code coverage of the item comparing to the previous build or the selected baseline build as a percentage.
		Covered (Executable Lines)	The number of the executable lines for an item displayed that are exercised by the tests.

Table 14. Information that is displayed in Build Analysis views (continued)			
View	**Displayed Location**	**Information**	**Description**
		Total (Executable Lines)	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.
		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.

Table 14. Information that is displayed in Build Analysis views (continued)			
View	**Displayed Location**	**Information**	**Description**
	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.
View by Tests	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.

View	Displayed Location	Information	Description
		File Tested	The number of source files that are exercised by the test for the current build being analyzed.
		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		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**.

Table 14. Information that is displayed in Build Analysis views (continued)

Information Display in Flowpoints Analysis View

View	Information	Description
Summary	Code Coverage	The code coverage percentage of a source file. See Terminology Overview for Code Coverage.

Table 15. Information that is display in Flowpoints Analysis views

View	Information	Description
	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.
	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.

Table 15. Information that is display in Flowpoints Analysis views (continued)

Table 15. Information that is display in Flowpoints Analysis views (continued)		
View	**Information**	**Description**
	Change %	The percentage of executable lines that is added or updated in the function or method comparing to the previous build.
	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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.
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 **File Trends** from the menu options on the upper right corner. 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 183.

Comparing code coverage results with baseline builds

ADI 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 162 and "Defining baseline builds for a Rational Team Concert Builds data provider" on page 169.

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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.
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.

Complete the following steps to select baseline builds on the **Build Analysis** view for comparison.

1. From the summary view of the workbook, select **View Files**.
2. The **Build Analysis** view appears with the code coverage information of all the files within the selected build.
3. Select **Compare With** drop-down menu from the menu options on the top.
4. Select a build that you want to use as a baseline for analysis. The code coverage analysis information 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.

Note: Everyone who has access to the workbook can see the same set of the files that are pinned.

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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.
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. On the top menu, select **Pinned Files** menu on the right.

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 183.

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 183.

11. Click the **Pinned Files** menu to go back to the list of pinned files or click the **Files** menu to go 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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

Filtering data in the Flowpoints Analysis view

In this topic you can learn how to filter information in the **Flowpoints 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

2. Select the name of the workbook that you want to view analysis information.

3. On the summary view of that workbook, click the name of Manual Builds data provider or Rational Team Concert Builds data provider with the latest build name that you want to view analysis information. The latest build information is displayed in the **Build Analysis** view.

4. Click the file or program name that you want to view analysis information. The **Flowpoints Analysis** view appears.

5. On the top menu, click the **Filters** dropdown menu.

6. In the **Filters** dropdown menu, select one or more of the following options 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 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.

   Note: By default, the **Build Analysis** view is displayed by applying this filter and setting the slider bar between 0% and insufficient code coverage threshold value. So it shows only the files that have insufficient code coverage.

   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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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 ADI 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**, ADI 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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. ADI 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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 183.

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 Chapter 7, "Managing connections, data providers, workbooks, and notifications," on page 155.

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 ADI. The page is displayed after you log in ADI. 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 s associated with OMEGAMON for CICS data provider, an average response time gauge report displays on the **Workbooks** view. For more information, see "OMEGAMON for CICS related reports and information" on page 198

- **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. The order of reports that is displayed depends on your role. For more information, see "Role-based dashboards and reports for a workbook" on page 179.

    - 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 198.

- **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 174.

- For more information, see "OMEGAMON for CICS related reports and information" on page 198.

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 16. OMEGAMON for CICS data analysis reports		
Reports Name	**Description**	**Displayed Location**
Average of Average response time and its indicator status	`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. Red indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 400% Orange indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 200 - 400% Yellow indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time by 100 - 200% Blue indicates that the average response time for all CICS transactions within the workbook exceeds the goal response time less than 100%	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 16. OMEGAMON for CICS data analysis reports (continued)		
Reports Name	**Description**	**Displayed Location**
Average response time	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.. Red indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 400% Orange indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 200 - 400% Yellow indicates that the average response time for all CICS transactions within the service class exceeds the goal response time by 100 - 200% Blue indicates that the average response time for all CICS transactions within the service class exceeds the goal response time less than 100%	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 16. 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	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. 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). 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).	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

Note: The display order of the reports in the summary view of a workbook depends on your role. For more information, see .

Table 17. 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.

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.

Table 17. OMEGAMON for CICS data analysis information (continued)

Analyzing transaction data

ADI 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.

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 198.

   - 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, ADI 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 Chapter 7, "Managing connections, data providers, workbooks, and notifications," on page 155.

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 ADI. 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 204.

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 204.

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 204.

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.

Table 18. System Management Facility (SMF) and Application Performance Analyzer (APA) Reports		
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 18. 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 19. 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 ADI. If you are not yet on the **Workbooks** view, click the **Menu** icon next to the IBM ADI header, and select **Workbooks** from the menu pane that appears.
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

Chapter 9. Troubleshooting

Troubleshooting helps you identify and solve problems that occur while using ADI.

Limited platform support

This application is built by using the Jazz infrastructure. However, ADI doesn't support all the platforms that are currently listed as supported for Jazz. For more information about the supported environment for ADI, see "Hardware and software requirements" on page 69.

Requirements on the pop-up blocker configuration

Currently, when you need to authenticate, the ADI log-on dialog pops up. Therefore, the browser that you are using must be configured to allow all the pop-up windows from the ADI server machine. In other words, you do not need to disable your pop-blocker, but you must configure it to allow the pop-up window from the ADI server. You can typically accomplish it in the browser's preferences settings by specifying an exception. Another method is to let the browser block the first initial pop-up window and then click the configuration banner or button that is displayed to inform you that a pop-up window was blocked. You need to select the option to allow pop-ups from this site in the future and then reload the page to make the pop-up log-on window appear.

Requirements on a self-signed certificate and browser configurations for the application server shipping with ADI

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 of errors in their browsers.

- For instruction about configuring the certificates, see the Jazz online help: Configuring security certificates.
- For information about how to configure the certificate, you can follow the same instructions as described in Configuring security certificates for Report Builder servers.
- For information about how to configure a browser for such self-signed certificates, follow the instructions in Importing certificates and configuring browsers for Report Builder reports.

Login problem after setting up ADDI

After you set up IBM ADDI, you need to restart the server. Otherwise, you will not be able to log in with the correct credentials and IBM ADDI will request you for re-authentication. When you open the *ADI installation folder*\adi\server\liberty\servers\clm\logs\addi-brd.log, you can see the warning of "Failed authentication with Jazz server". To restart the server, complete the following steps:

1. Navigate to the *<unzipped-adi-installer>*/adi/server directory.
2. Shut down the server by running the server.shutdown script.
3. Restart the server after the you shut down the server successfully by running the server.startup script.

Icon display error on ADI browser page

ADI uses a custom font to display all of its icons and browser settings might prevent the custom font from being downloaded. In that case, when you browse ADI, the UI might be filled with strange characters and numbers that don't make sense. Therefore, you need to make sure that your browser preferences are set to allow pages to choose their own fonts. This icon display error occurs in Mozilla Firefox browser.

To verify that your Mozilla Firefox browser is configured to download the custom font, make sure that the following settings are correct:

1. Select **Tools>Options**.
2. Select **Content** in the left navigation area.
3. In the **Fonts & Colors** section, click **Advanced...**.
4. On the **Fonts** window, verify that the checkbox for **Allow pages to choose their own fonts, instead of my selections above** is selected and then press **OK**.

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 larger secondary log size since they are constantly cleared by DB2.

Migration error (Error 503: CRJAZ1173W) when migrating from ADI 1.0.1 or ADI 1.0.1 ifix01 to ADI 1.0.2

When you complete the migration steps, start the new server, and navigate to the applications main page: `https://servername:9753/adi/web`, you might hit the following error.

```
Error 503: CRJAZ1173W The com.ibm.team.repository.service.internal.ILogoutRestService service is
not available.
```

This error is caused by services trying to access Jazz Team Server (JTS) root services too early in the application. This means that JTS is not fully started and the root services that are required for authentication are not yet available.

To fix the error, you should try the following steps:

1. Open a browser and navigate to the JTS admin page: `https://servername:9753/jts/admin` to ensure that Jazz Team Server is fully started.
2. If you can successfully access the JTS admin page, you can retry accessing the applications main page: `https://servername:9753/adi/web`.
3. Review the progress of migration tasks in the log file located in adi102/server/logs/adi.log to verify that your data is migrated successfully.
4. Confirm that these two lines are shown in the log.

   ```
   CRIDA0059I The DIMEZ warehouse schema validation has completed.
   ```

   and

   ```
   CRIDA0052I Completed checking the ADI data migration status.
   ```

5. If you find errors in the log, review the error message to understand the nature of the problem. Contact technical support to get further help with any migration error messages.

Incorrect statuses of business terms after the migration from ADI 5.1.0 to ADI 5.1.0.2

The statuses of business terms in V5.1.0 are not preserved after you migrate from ADI V5.1.0 to ADI V5.1.0.2. You need to manually correct the statuses based on the following mapping.

Table 20. The status mapping between V5.1.0 and V5.1.0.2	
V5.1.0 Status	**V5.1.0.2 Status**
New	Waiting For Approval
In Progress	New
Waiting For Approval	Rejected
Approved	In Progress
Rejected	Approved

A DB2 problem after the migration from ADI 5.1.0 to ADI 5.1.0.2 or later versions

After you migrate from ADI V5.1.0 to ADI V5.1.0.2 or later versions, you might fail to save the association between a code snippet and a business term due to a DB2 problem. The problem might occur when you try to complete the following steps:

1. Create a Business Rules Discovery data provider and a workbook.
2. Navigate to the Keyword details page.
3. Associate a code snippet to a business term.

You might find the association cannot be saved and nothing happens when you click the **Save** button. If you check the ADI server console log, you might find the following error messages:

```
org.springframework.web.util.NestedServletException: Request processing failed; nested
exception is org.hibernate.exception.GenericJDBCException: could not extract ResultSet
```

and

```
ERROR org.hibernate.engine.jdbc.spi.SqlExceptionHelper   - DB2 SQL Error: SQLCODE=-668,
SQLSTATE=57016, SQLERRMC=7;ADDI.OCCURRENCE, DRIVER=4.21.29
```

As a workaround to the problem, an ADI server admin and a DB2 admin should complete the following steps:

1. Navigate to the *<unzipped-adi-installer>*/adi/server directory.
2. Shut down the server by running the server.shutdown script.
3. Start the DB2 command line processor and run the following commands.

   a. Run CONNECT command to connect to the ADI warehouse database.

   ```
   DB2 CONNECT TO <ADI warehouse database name>
   ```

   b. Run REORG TABLE command to reorganize the table.

   ```
   DB2 REORG TABLE ADDI.OCCURRENCE
   ```

   c. Run DISCONNECT command to disconnect from the ADI warehouse database.

   ```
   DB2 DISCONNECT <ADI warehouse database name>
   ```

4. Navigate to the *<unzipped-adi-installer>*/adi/server directory.
5. Start the server by running the server.startup script.

Error in importing archive files from a Mac machine

When you create an archive file on a Mac machine, a metadata file will be created in th archive file. You can import that file to ADI but an error message will occur because ADI doesn't understand the metadata file.

You can ignore the error message. The import function is still working correctly.

Application log files

Application logs contain important information about the JTS server and ADI 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 ADI application.

Perform the following steps to adjust the log levels.

1. Browse to `Installed_ADI_folder > adi > server > conf > adi` and edit log4j.properties files.
2. Go to the line 17 on the log4j.properties and switch the log level from WARN to INFO.

Other key log files are available to collect to aid the troubleshooting. The key log files are:

• Log file for ADI application: `Installed_ADI_folder\adi\server\liberty\servers\clm\logs\adi.log`
• Log file for JTS: `Installed_ADI_folder\adi\server\liberty\servers\clm\logs\jts.log`
• Log file for ADI data collection: `Installed_ADI_folder\adi\server\conf\adi\logs\adi-collections.log`
• Log file to check which patches have been applied: `Installed_ADI_folder\adi\server\liberty\servers\clm\logs\adi-patch.log`

Slow performance of OMEGAMON for CICS data provider

If you collect many months of data for millions of transactions from an OMEGAMON for CICS data provider, you might see a slowdown for the **Transactions with Performance and Reliability Issues** viewlet. For DB2 database, you can apply the following scripts to generate additional indexes and constraints that improve the performance significantly.

If you create your database with ADI V5.0.3 or earlier versions, run the following scripts.

```
CREATE INDEX "DIMEZ"."IDX1712012235210" ON "DIMEZ"."FACTTRANSACTIONMETRICS" ( "SERVICECLASS",
"PLEX", "SNAPSHOT" ) ALLOW REVERSE SCANS COLLECT STATISTICS;

CREATE INDEX "DIMEZ"."IDX1712012235540" ON "DIMEZ"."FACTTRANSACTIONMETRICS" ( "SNAPSHOT",
"NUMTRANSEXCRESPTIMEGOAL", "NUMABENDS", "DB2WAIT", "FILEWAIT", "AVGCPUTIME",
"AVGRESPONSETIME" , "TOTALTRANSACTIONS", "TRANSACT", "MAXBUCKET", "SERVICECLASS", "PLEX" )
ALLOW REVERSE SCANS COLLECT STATISTICS;

CREATE INDEX "DIMEZ"."IDX1712020039000" ON "DIMEZ"."FACTTRANSACTIONMETRICS" ( "SERVICECLASS",
"NUMTRANSEXCRESPTIMEGOAL", "SNAPSHOT", "TRANSACT", "MAXBUCKET", "PLEX") ALLOW REVERSE SCANS
COLLECT STATISTICS;

ALTER TABLE DIMEZ.FACTTRANSACTIONMETRICS ALTER COLUMN SERVICECLASS SET NOT NULL;
ALTER TABLE DIMEZ.FACTTRANSACTIONMETRICS ALTER COLUMN MAXBUCKET SET NOT NULL;
ALTER TABLE DIMEZ.FACTTRANSACTIONMETRICS ALTER COLUMN PLEX SET NOT NULL;
CALL SYSPROC.ADMIN_CMD('REORG TABLE DIMEZ.FACTTRANSACTIONMETRICS');
ALTER TABLE DIMEZ.FACTTRANSACTIONMETRICS ALTER COLUMN SNAPSHOT SET NOT NULL;
ALTER TABLE DIMEZ.FACTTRANSACTIONMETRICS ALTER COLUMN TRANSACT SET NOT NULL;
CALL SYSPROC.ADMIN_CMD('REORG TABLE DIMEZ.FACTTRANSACTIONMETRICS');
```

In addition, if you create your database with ADI V5.0.2 or earlier versions, run the following scripts to perform an additional operation.

```
CREATE INDEX "DIMEZ"."IDX1705190119390" ON "DIMEZ"."FACTTRANSACTIONMETRICS" ( "SERVICECLASS"
ASC, "NUMTRANSEXCRESPTIMEGOAL" ASC, "NUMABENDS" ASC, "DB2WAIT" ASC, "FILEWAIT" ASC,
"AVGCPUTIME" ASC, "AVGRESPONSETIME" ASC, "TOTALTRANSACTIONS" ASC, "SNAPSHOT" ASC, "TRANSACT"
ASC, "MAXBUCKET" ASC, "PLEX" ASC ) ALLOW REVERSE SCANS COLLECT STATISTICS
```

Verifying the Elasticsearch server is running

Starting from V5.0.4, ADI integrates Elasticsearch in it and requires that elasticsearch server is always up and running in the background. The Elasticsearch server is started by default when you start ADI

application. You can easily verify that the Elasticsearch server is running through 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 *Installed ADI folder*`/adi/elasticsearch/logs` folder.

Elasticsearch failing to startup

When you start ADI application, 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 ADI application 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. Verify if the Elasticsearch is running or not. You can perform this task by following the steps provided in the previous section of **Verifying Elasticsearch server is running** .

2. If the Elasticsearch 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 *Installed ADI folder*`/adi/elasticsearch/logs` folder.

3. Fix the errors reported in the log files.

4. Stop and restart ADI application server.

5. 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 Elasticsearch server is running* section.

6. 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 ADI application 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. Verify if the Elasticsearch is running or not. You can perform this task by following the steps provided in the previous section of *Verifying Elasticsearch server is running* .

2. If the Elasticsearch 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 *Installed ADI folder*/adi/elasticsearch/logs folder.

3. Fix the errors reported in the log files.

4. Stop and Restart ADI application server.

5. Verify if the Elasticsearch service is installed and started successfully this time. To confirm it, complete the steps as described in the *Verifying Elasticsearch server is running* section.

6. 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 ADI 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 *Installed ADI folder*/adi/ elasticsearch by running the following command:

   ```
   cd Installed ADI folder/adi/elasticsearch
   ```

2. In the current directory, start the Elasticsearch server by running the following command.

   ```
   startes
   ```

After you complete these steps, you can start the Elasticsearch server successfully and continue to use ADI 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 *Installed ADI folder*/adi/elasticsearch/ config path.

   b. Update the rootLogger.level property as rootLogger.level **= error**.

2. Investigate if any zombie ADI Java processes that might lock Elasticsearch data folder are running.

3. Terminate the zombie ADI Java processes if any.

   Note: Avoid deleting anything in the adi\elasticsearch\data folder, unless you want to clean Elasticsearch server for a fresh restart.

Problems with running ADI application on Linux using port 80 or port 443

Starting from ADI V5.0.4, the Elasticsearch is started by default during ADI application server startup. The startes script is used on Linux to start Elasticsearch as a non-root user when the rest of ADI is started as a root user.

ADI application cannot be configured (jts/setup) as a non-root user. On Linux, ports below 1024 can be opened only by a root user. If you have problems to run your ADI application on Linux by using port 80 or port 443, make sure that you run the ADI application server as a root user.

If you run ADI application on Linux by using port 80 or port 443 as a root user, 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 terminal and run the following command to navigate to the *Installed ADI folder*/adi/elasticsearch:

   ```
   cd Installed ADI folder/adi/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
   ```

   Note: If you are using SSH, you need to keep the session alive. Otherwise, the Elasticsearch will terminate when you quit the SSH connection.

After you complete the previous steps, your ADI application server will run as a root user while your Elasticsearch server will run separately as a non-root user.

Troubleshooting tip

For more troubleshooting information on ADI, you can check the Q&As on dW Answers page.

Chapter 10. Providing feedback

You can provide feedback to ADI by sending an email, opening a Service Request, or sending a Request For Enhancement (RFE).

Sending an email

Complete the following steps to send an email to ADI.

1. Log in to ADI with your user ID and password.
2. Click the **Menu** icon next to the IBM ADI header to open the menu panel.
3. Click **Feedback** from the menu panel.
4. Select **Send Email** from the dropdown menu. Your default email program is opened with the default recipient and subject.

Opening a Service Request

In the case that you have problems or issues and need ADI 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.

Complete the following steps to open a Service Request to ADI.

1. Login to ADI with your user ID and password.
2. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
3. Click **Feedback** from the menu panel.
4. Select **Open Service Request** from the dropdown menu. You will be redirected to the Service Request submission page on IBM site.

Sending a Request for Enhancement (RFE)

In the case that you want to request or suggest an enhancement of ADI 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.

Complete the following steps to open a Request for Enhancement for ADI.

1. Login to ADI with your user ID and password.
2. Click the **Menu** icon next to the IBM ADI header to open the menu pane.
3. Click **Feedback** from the menu panel.
4. Select **Request for Enhancement** from the dropdown menu. You will be redirected to the Servers and Systems Software RFE Community on IBM site.

Chapter 11. Accessibility features for ADI

Accessibility features assist users who have a disability, such as restricted mobility or limited vision, to use information technology content successfully.

Accessibility features

ADI includes the following major accessibility features:

- Keyboard-only operation
- Operations that use a screen reader

ADI uses the latest W3C Standard, WAI-ARIA 1.0 (http://www.w3.org/TR/wai-aria/), to ensure compliance to US Section 508 (http://www.access-board.gov/guidelines-and-standards/communications-and-it/about-the-section-508-standards/section-508-standards), and Web Content Accessibility Guidelines (WCAG) 2.0 (http://www.w3.org/TR/WCAG20/). To take advantage of accessibility features, use the latest release of your screen reader in combination with the latest web browser that is supported by this product.

The ADI online product documentation in IBM Knowledge Center is enabled for accessibility. The accessibility features of IBM Knowledge Center are described at https://www.ibm.com/support/knowledgecenter/en/about/releasenotes.html#accessibility.

Keyboard navigation

This product uses standard navigation keys.

ADI uses Tab and Return as the keyboard shortcuts.

To enable keyboard navigation on Apple Mac OS X for your browser, click **System Preferences** > **Keyboard** > **Shortcuts**, and select the **All controls** radio box at the bottom of the dialog.

Related accessibility information

In addition to standard IBM help desk and support websites, 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).

Chapter 12. 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

IBM®

Part Number:
Product Number: 5725-Y87

(1P) P/N: