Oracle® Cloud

Using the FTP Adapter with Oracle Integration 3

F45554-19
February 2025

Oracle Cloud Using the FTP Adapter with Oracle Integration 3,

F45554-19

Primary Author: Oracle Corporation

Contents

Preface

1 Understand the FTP Adapter

2 Create an FTP Adapter Connection

7 FTP Adapter Samples

Preface

This guide describes how to configure this adapter as a connection in an integration in Oracle Integration.

> **Note:**
>
> The use of this adapter may differ depending on the features you have, or whether your instance was provisioned using Standard or Enterprise edition. These differences are noted throughout this guide.

Topics:

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Resources
- Conventions

Audience

This guide is intended for developers who want to use this adapter in integrations in Oracle Integration.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Resources

See these Oracle resources:

- Oracle Cloud at `http://cloud.oracle.com`
- *Using Integrations in Oracle Integration 3*
- *Using the Oracle Mapper with Oracle Integration 3*
- Oracle Integration documentation on the Oracle Help Center.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Understand the FTP Adapter

Review the following conceptual topics to learn about the FTP Adapter and how to use it as a connection in integrations in Oracle Integration. A typical workflow of adapter and integration tasks is also provided.

Topics:

- FTP Adapter Capabilities
- FTP Adapter Restrictions
- What Application Version Is Supported?
- Common Message Exchange Patterns
- About FTP Adapter Use Cases
- Workflow to Create and Add an FTP Adapter Connection to an Integration

FTP Adapter Capabilities

The FTP Adapter enables the integration of servers supporting File Transfer Protocol (FTP) and Secure Shell (SSH) File Transfer Protocol (sFTP) into Oracle Integration. The connectivity agent must be used for the latter environment.

The FTP Adapter has the following capabilities:

- Transfers (reads and writes) files to any publicly accessible server in either binary or ASCII format. The FTP Adapter can connect to FTP and sFTP servers that are publicly available on the internet.
- Supports using the connectivity agent to read and write files from an FTP server that is not publicly accessible. Encryption and decryption of these files is supported with the connectivity agent. The FTP server may be behind a firewall, but is accessible from the connectivity agent host. See Configure the Endpoint Access Type and Invoke Operations Page.
- Supports connecting to private resources that are in your virtual cloud network (VCN) with a private endpoint. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Configure the Endpoint Access Type. This type of connection does not use the connectivity agent.
- Supports dynamically updating the invoke connection to use at runtime. This feature enables you to reuse the same integration to send requests to multiple end systems or instances of the same application without creating separate integrations for each. See Dynamically Update Invoke Connections at Runtime.
- Supports streaming for transferring large files.
- Enables the integration developer to build an integration for polling (retrieving files) from remote FTP and sFTP servers and for uploading the files onto remote FTP/sFTP servers. Oracle Integration provides the capability for scheduling the time and frequency at which to run these integrations.

  See Define the Integration Schedule in *Using Integrations in Oracle Integration 3*.

- Enables you to specify the path of the source directory and schema of the files to consider while polling. You can also specify the directory that does not take part in polling. Once the schema matches with that of the file in the input directory, the file can either be moved/archived to a specific location or deleted. Files that do not support the chosen schema are transferred to the rejection directory.
- Supports the optional describing of the schema (data definitions) for the files uploaded/downloaded to and from FTP servers using one of the following formats.
  - Sample delimited document (for example, .CSV).

    > **Note:**
    >
    > Supported delimiters in the file are single space, comma, semicolons, pipe, or tab.

  - XML schema (XSD) document
  - Sample XML document (Single or No NameSpace)
  - Sample JSON document
- Enables you to encrypt a file that is being uploaded to remote FTP/sFTP servers using Pretty Good Privacy (PGP) cryptography. Encryption and signing of the file while uploading to remote FTP/s(FTP) servers are currently available only when the servers are publicly accessible. For privately hosted FTP/(s)FTP servers, you must encrypt and/or sign the file prior to uploading the file to the privately-hosted server using the *Using Integrations in Oracle Integration 3*.
- Enables you to decrypt a file being read or downloaded from a remote FTP/sFTP server using PGP cryptography. Decryption and verification of the file while downloading from a remote FTP/s(FTP) server to Oracle Integration are currently available only when the servers are publicly accessible. For privately hosted FTP/(s)FTP servers, you must download the file from privately-hosted FTP/(s)FTP servers to Oracle Integration using the FTP Adapter prior to decrypting and verifying the contents of the file.
- Supports implicit and explicit FTP over SSL.
- Supports FTP, sFTP, and FTP over SSL connections. For FTP over SSL connections, you must use an SSL certificate in Public-Key Cryptography Standard (PKCS12) format. For sFTP connections using a key, you must use a key in PEM format.

  See Using Secure FTP with the Oracle FTP Adapter.
- Supports self-diagnosing connectivity issues that may occur when integrating with external sFTP servers.
- Supports host key authentication, public key authentication, and multilevel authentication:
  - Supports different authentication schemes
  - Supports different types of file handling operations (List File, Read File, Write File, Download File, Move File, and Delete File)
  - Supports compression and decompression of files in ZIP and GZIP formats
- Supports signing the content before sending the file to an FTP server. You can upload the PGP sign private key and PGP sign private key password on the Connections page. Signing can be combined with PGP encryption.
- Supports verification of the signature after downloading the file from the FTP server. You must upload the PGP sign public key on the Connections page to verify the signature.

- Supports using File Server with the FTP Adapter to read and write files. File Server is an sFTP server and uses sFTP connection details. You can also use a File server action to connect to File Server.

  To learn more, see:

  - About File Server in *Using File Server in Oracle Integration 3*.
  - Interact with Files in File Server in *Using Integrations in Oracle Integration 3*.

You can configure the FTP Adapter as an invoke connection in an integration in Oracle Integration. The FTP Adapter is one of many predefined adapters included with Oracle Integration. See the Adapters page in the Oracle Help Center.

Video

FTP Adapter Restrictions

Note the following FTP Adapter restrictions.

- Authenticated encryption with associated data (AEAD) encryption/decryption is not supported.
  Perform the following steps to remove AEAD from your keys on GNU Privacy Guard (GnuPG) 2.4.5. See GnuPG.

  > **Note:**
  >
  > Some lower versions of GnuPG may not support these steps.

  1. Open GPG for editing with your key fingerprint.

     ```
     gpg --expert --edit-key key_fingerprint
     ```

  2. Show the preferences for the key user name and email address.

     ```
     gpg> showpref
         [ unknown] (1). Key_User's_Name email@oracle.com
          Cipher: AES256, AES192, AES, 3DES
          AEAD: OCB
          Digest: SHA512, SHA384, SHA256, SHA224, SHA1
          Compression: ZLIB, BZIP2, ZIP, Uncompressed
          Features: MDC, AEAD, Keyserver no-modify
     ```

  3. Set your preferences.

     ```
     gpg> setpref AES256 AES192 AES SHA512 SHA384 SHA256 SHA224 ZLIB BZIP2
     ZIP
         a. Set preference list to:
             Cipher: AES256, AES192, AES, 3DES
             AEAD:
             Digest: SHA512, SHA384, SHA256, SHA224, SHA1
             Compression: ZLIB, BZIP2, ZIP, Uncompressed
             Features: MDC, Keyserver no-modify
     ```

4. Confirm when prompted to update your preferences.

   ```
   Really update the preferences? (y/N) y
   ```

5. Save your updates.

   ```
   gpg> save
   ```

- RSA-based PGP keys less than 1024 bits in length are not supported for FTP public key authentication. The recommended length is 2048 bits.
- If you configure the FTP Adapter as an FTP connection or an FTP over SSL connection, file names that have non-ASCII characters are not supported. Any operation such as read, write, list, and others that takes file names or directory paths that have non-ASCII characters does not work. This issue does not impact sFTP connections. See Configure Connection Properties.
- The FTP Adapter does not support TLS resumption.
- The FTP Adapter supports files up to 50 MB in size for schema-based inbound polling using the connectivity agent.
- The FTP Adapter supports files up to 100 MB in size for schema based inbound polling using cloud connectivity through the public gateway.
- The FTP Adapter supports files up to 100 MB file size for schema based inbound polling using cloud connectivity via Public gateway
- The FTP Adapter supports files up to 50 MB in size for schema-based inbound polling using the connectivity agent.
- The FTP Adapter does not support verification on the signed file downloaded from an external FTP server when using a detached signature.
  If a detached signature is used, the signature verification fails with the following exception:

  ```
  Signature Verification Failed:Application Error
  ```

- The FTP Adapter does not support the IBM FTP server.
- Oracle Integration does not support the form feed char (0xc) feature in the Extensible Markup Language (XML) 1.0 specification.
- You must generate SSH keys in PEM format with the `ssh-keygen` tool to connect to an on-premises sFTP server. See Generate SSH Keys in PEM Format to Connect to a Public or On-Premises sFTP Server.
- Note the following restrictions when using the connectivity agent with the FTP Adapter:
  - Unzipping is not supported with the Download File operation.
  - When configuring the FTP Adapter to communicate with an sFTP server through the connectivity agent, you must select an algorithm from the **SFTP Key Exchange Algorithm** field on the Connections page. Do *not* select the **diffie-hellman-group1-sha1** algorithm. This algorithm is not supported with the connectivity agent.
- The FTP Adapter only supports homogeneous arrays in JSON. Heterogeneous arrays in JSON samples and payloads are not supported. An example of a heterogeneous array is as follows:

  ```
  {
      "example": [
           "kumar",
  ```

```
        {
            "firstName": ["John"],
        },
        {
            "length": 100,
            "width": 60,
            "height": 30
        }
]
}
```

- The FTP Adapter does not support the processing of files from the same folder by two integrations. File deletion causes a conflict. For example, if one integration deletes a file, the second integration cannot find the file to delete, and fails.

> **Note:**
>
> There are overall service limits with Oracle Integration. A service limit is the quota or allowance set on a resource. See Service Limits.

What Application Version Is Supported?

For information about which application version is supported by this adapter, see the Connectivity Certification Matrix.

Common Message Exchange Patterns

You can use the FTP Adapter in a variety of message exchange patterns, including the following:

- Triggered by the SOAP Adapter, the REST Adapter, or a schedule in an integration.
- Polling an FTP directory:
  1. Create a schedule integration.
  2. Configure the FTP Adapter with the List File operation.
  3. Create a for-each action.
  4. Configure the FTP Adapter with the Read File Operation inside the for-each action to read a file.
  5. Map the file name and directory header from the for-each action loop.
- Downloading a file from an embedded SFTP server with Oracle Managed File Transfer / Oracle Managed File Transfer Cloud Service:
  – A file is placed in an embedded SFTP server.
  – Oracle Managed File Transfer sends a multipart SOAP notification to Oracle Integration.
  – Oracle Integration receives the SOAP message and extracts the file name.
  – The FTP Adapter is invoked and sets the file name.
  – The file is downloaded to Oracle Integration.

About FTP Adapter Use Cases

The FTP Adapter can be used in the following scenarios.

- FTP Adapter to Oracle ERP Cloud Adapter Integration
- JSON to XML Special Character Conversion

FTP Adapter to Oracle ERP Cloud Adapter Integration

This use case describes how to load data from a secure FTP location to Oracle ERP Cloud by using a trigger file mechanism. The data is sent to Oracle Integration for processing. Oracle Integration schedules the integration flow for polling the trigger file.

- Create a schedule integration.
- From the **Invoke** palette, drag an FTP Adapter into the integration canvas.
- Configure the FTP Adapter to perform the following tasks:
  - Specify a name of `readTriggerFile`.
  - Specify a file name pattern of `*.TRG`.
  - Specify an input directory from which to read a trigger file.
  - Define a schema from a CSV file as the trigger file format.
- From the **Actions** panel, add an Assign action below the FTP Adapter to declare the set of variables.
  - Specify a name of `ExtactFileName`.
  - In the Expression Builder, configure variables, such as extracting the ZIP file name of `*.TRG` from a relative path.
- Add a second invoke FTP Adapter below the Assign action to download and stage the file temporarily in Oracle Integration.
- Configure the FTP Adapter to perform the following tasks:
  - Specify a name of `DownloadFile`.
  - Select the **Download to ICS** operation.
  - Specify the input directory and download directory path.
  - Select the **Unzip the File** option.

    A mapper is automatically added to the integration whenever a Stage File action or new endpoints such as the FTP Adapter are added to the integration. The mapper appears between the Assign action and the **DownloadFile** FTP Adapter.
- In the mapper, map the source extracted ZIP file to the target `DownloadFileToICS` filename element.
- Because the ZIP file may contain multiple files, drag a For-Each action below the **DownloadFIle** FTP Adapter in the integration. This action enables iteration over a repeated element.
- Specify the repeating element and current element name.
- To read each file from the input directory, add a Stage File action below the For-Each action and configure it as follows:

  - In the Expression Builder, specify the file name and directory from which to read the file.
  - Specify the schema file by loading a CSV file that contains the data structure.
- When a file is staged in Oracle Integration, add a second For-Each action below the Stage File action to iterate through each record.
  - Add the repeating element and current element name.
- Add a second Stage File action to write each record into a new file.
  - Specify a name of `WriteRecordToFile`.
  - In the Expression Builder, set the name and the output directory. All the enriched data is appended to the staged file. The schema file for the new ERP file uses the same structure as the source file.
- Perform the data mapping and transformation in the automatically added mapper. Functions, operators, and XSLs are useful to enrich and transform the new file for ERP.
- Add a Stage File action below the `WriteRecordToFile` Stage File action to write a manifest file with the business data details.
  - Specify a name of `CreateManifest`.
  - In the Expression Builder, select the **Write File** option and **Append to Existing File**.
- In the mapper that is automatically created above the **CreateManifest** Stage File action, define the manifest file content.
- Add a Stage File action below the **CreateManifest** Stage File action to rezip the file before sending it to the ERP cloud endpoint.
  - In the Expression Builder, select the **ZIP Files** option.
  - Select the zip file name to concatenate and the directory to zip.
- Add an Oracle ERP Cloud Adapter at the end of the integration to import the data into the ERP Financial application.
  - Select the **Import Payable Invoices** operation.

    Once the data loading and processing are complete, an email notification must be triggered.
  - Select **Email Notification**.
  - Define a callback to retrieve any details about failed records.
- In the mapper above the Oracle ERP Cloud Adapter, define the file references for the endpoint so that the ERP system processes the rezipped file accordingly.
- Activate the integration and monitor the activity stream from the Runtime Health page. The activity stream and diagnostic logs are available for download.
- Save and activate your integration.
- Invoke the integration.
- Monitor the activity stream and the diagnostics log.

JSON to XML Special Character Conversion

If the JSON payload has special characters that are not valid in XML, those characters are replaced by a string when converted from JSON to XML.

For example, assume you have the following JSON payload:`{ "_id": { "$oid": "52cdef7f4bab8bd67529c6f7" } }`

You then select the **JSON Sample** payload format and **<<inline>>** to copy and paste the payload into the text field in the Adapter Endpoint Configuration Wizard.

In the mapper, the field `$oid` is represented with a string value of `_0x646c72_oid`.

The list of special characters and their corresponding XML conversion strings are as follows:

Special Character	Converted Value Represented in the Mapper
`" "`	`_0x737063_`
`"/"`	`_0x736c68_`
`"\\"`	`_0x626c68_`
`":"`	`_0x636c6e_`
`";"`	`_0x73636e_`
`"("`	`_0x6c7072_`
`")"`	`_0x727072_`
`"&"`	`_0x616d70_`
`","`	`_0x636d61_`
`"#"`	`_0x706e64_`
`"?"`	`_0x717374_`
`"<"`	`_0x6c7374_`
`">"`	`_0x677274_`
`"start"`	`_0x737472_`
`"@"`	`_0x617472_`
`"$"`	`_0x646c72_`
`"{"`	`_0x6c6362_`
`"}"`	`_0x726362_`
`"%"`	`_0x706572_`

Workflow to Create and Add an FTP Adapter Connection to an Integration

You follow a very simple workflow to create a connection with an adapter and include the connection in an integration in Oracle Integration.

Step	Description	More Information
1	Create the adapter connections for the applications you want to integrate. The connections can be reused in multiple integrations and are typically created by the administrator.	Create an FTP Adapter Connection
2	Create the integration. When you do this, you add trigger and invoke connections to the integration.	Create Integrations and Add the FTP Adapter Connection to an Integration

Step	Description	More Information
3	Map data between the trigger connection data structure and the invoke connection data structure.	Map Data in *Using Integrations in Oracle Integration 3*
4	(Optional) Create lookups that map the different values used by those applications to identify the same type of object (such as gender codes or country codes).	Manage Lookups in *Using Integrations in Oracle Integration 3*
5	Activate the integration.	Manage Integrations in *Using Integrations in Oracle Integration 3*
6	Monitor the integration on the dashboard.	Monitor Integrations During Runtime in *Using Integrations in Oracle Integration 3*
7	Track payload fields in messages during runtime.	Assign Business Identifiers for Tracking Fields in Messages and Track Integration Instances in *Using Integrations in Oracle Integration 3*
8	Manage errors at the integration level, connection level, or specific integration instance level.	Manage Errors in *Using Integrations in Oracle Integration 3*

2 Create an FTP Adapter Connection

A connection is based on an adapter. You define connections to the specific cloud applications that you want to integrate.

Topics:

- Prerequisites for Creating a Connection
- Create a Connection

Prerequisites for Creating a Connection

Satisfy the following prerequisites appropriate to your environment to create a connection with the FTP Adapter.

- FTP Server Prerequisites
- FTP Over SSL Prerequisites
- Secure FTP with a Public Key Prerequisites

FTP Server Prerequisites

- Ensure that you have write permissions on the FTP server directory to which to transfer files.
- Ensure that you have read permissions on the FTP server directory from which to transfer files.
- Know the host name or IP address and port number of the FTP server.
- Know the user name and password for connecting to the FTP server.

FTP Over SSL Prerequisites

If you want to use FTP over SSL, know the location of the FTP server certificate in PKCS12 format.

Secure FTP with a Public Key Prerequisites

If you want to use secure FTP with a public key, know the location of the private key of the server and optionally the passphrase to access the private key. If you are using Secure FTP (sFTP) with HostKey-based authentication, know the location of hostkey to upload:

- The default location of the RSA key is `/etc/ssh` on the server.
- If you are using Oracle Managed File Transfer Cloud Service, download the host key from the Oracle Managed File Transfer Cloud Service console.
- If you are using `vsftpd`, you will find the host key under `/etc/ssh`.

Create a Connection

Before you can build an integration, you must create the connections to the applications with which you want to share data.

> **Note:**
>
> You can also create a connection in the integration canvas. See why working with projects is preferred.

To create a connection in Oracle Integration:

1. Decide where to start:
   - Work in a project (see why working with projects is preferred).
     a. In the navigation pane, click **Projects**.
     b. Select the project name.
     c. Click **Integrations** .
     d. In the **Connections** section, click **Add** if no connections currently exist or **+** if connections already exist. The Create connection panel opens.
   - Work outside a project.
     a. In the navigation pane, click **Design**, then **Connections**.
     b. Click **Create**. The Create connection panel opens.
2. Select the adapter to use for this connection. To find the adapter, scroll through the list, or enter a partial or full name in the **Search** field.
3. Enter the information that describes this connection.

Element	Description
Name	Enter a meaningful name to help others find your connection when they begin to create their own integrations.
Identifier	Automatically displays the name in capital letters that you entered in the **Name** field. If you modify the identifier name, don't include blank spaces (for example, `SALES OPPORTUNITY`).

Element	Description
Role	Select the role (direction) in which to use this connection. **Note**: *Only* the roles supported by the adapter you selected are displayed for selection. Some adapters support all role combinations (trigger, invoke, or trigger and invoke). Other adapters support fewer role combinations. When you select a role, only the connection properties and security policies appropriate to that role are displayed on the Connections page. If you select an adapter that supports both invoke and trigger, but select only one of those roles, you'll get an error when you try to drag the adapter into the section you didn't select. For example, assume you configure a connection for the Oracle Service Cloud (RightNow) Adapter as only an **invoke**. Dragging the adapter to a **trigger** section in the integration produces an error.
Keywords	Enter optional keywords (tags). You can search on the connection keywords on the Connections page.
Description	Enter an optional description of the connection.
Share with other projects	**Note**: This field only appears if you are creating a connection in a project. Select to make this connection publicly available in other projects. Connection sharing eliminates the need to create and maintain separate connections in different projects. When you configure an adapter connection in a different project, the **Use a shared connection** field is displayed at the top of the Connections page. If the connection you are configuring matches the same type and role as the publicly available connection, you can select that connection to reference (inherit) its resources. See Add and Share a Connection Across a Project.

4. Click **Create**.

   Your connection is created. You're now ready to configure the connection properties, security policies, and (for some connections) access type.

5. Follow the steps to configure a connection.
   The connection property and connection security values are specific to each adapter. Your connection may also require configuration with an access type such as a private endpoint or an agent group.

6. Test the connection.

Configure Connection Properties

The FTP Adapter supports the following types of FTP connections.

- FTP connection: The FTP Adapter supports passive communication to an FTP server. You must configure passive configurations in the FTP server.

- FTP over SSL connection: The FTP Adapter supports FTP over SSL, which supports explicit FTP over SSL.
- sFTP connection: The FTP Adapter supports communication with a secure FTP server.

> **Note:**
>
> File Server only supports sFTP connections.

Configure an FTP Connection

Enter FTP connection information so your application can process requests.

1. Go to the **Properties** section.
2. Enter the following details:

Element	Description
FTP Server Host Address	Enter the host address of the FTP/FTPS server.
FTP Server Port	Enter the FTP server port number.
SFTP Connection	Select **No** from the list.
Passive IP as Host Address	If using a different IP in a passive configuration, select **Yes** from the list.
FTP Server OS	Select either **Unix** or **Windows** as the operating system of the host on which the FTP server is installed. The list operation requires this information to parse the response because Unix and Microsoft Windows use different line-ending characters. **Note**: This is an optional field used only with the **List Files** operation on the Invoke Operations page. See Invoke Operations Page.
FTP Server Time Zone	Select the time zone of the FTP server. **Note**: This is an optional field. If you plan to specify a processing delay, use the **Minimum Age** field of the **List Files** operation on the Invoke Operations page.

Configure an FTP over SSL Connection

Enter FTP over SSL connection information so your application can process requests.

1. Go to the **Properties** section.
2. Enter the following details:

Element	Description
FTP Server Host Address	Enter the host address of the FTP/FTPS/sFTP server.
FTP Server Port	Enter the FTP server port number.
SFTP Connection	Select **No** from the list.
Passive IP as Host Address	If using a different IP in a passive configuration, select **Yes** from the list.

Element	Description
FTP SSL Type	• **Explicit** • **Implicit**
SSL Certificate	**Note**: This field is now optional. The PKCS12 certificate format is no longer required. You can instead upload a public certificate on the Certificates page. See Upload an SSL Certificate in *Using Integrations in Oracle Integration 3*. If you are using an FTP over SSL certificate, click ⬆ to upload a certificate in PKCS12 format (`.p12` extension).
FTP Server OS	Select either **Unix** or **Windows** as the operating system of the host on which the FTP server is installed. The list operation requires this information to parse the response because Unix and Microsoft Windows use different line-ending characters. **Note**: This is an optional field used only with the **List Files** operation on the Invoke Operations page. See Invoke Operations Page.
Channel Mask	If you are using FTP over SSL, select a channel encryption option: • **Control**: Encrypts the control channel. Data is transferred in plain text. • **Data**: Encrypts the data transferred. Commands in the control channel are in plain text. • **Both**: Encrypts both the control and data channels. • **None**: No encryption is performed.
FTP Server Time Zone	Select the time zone of the FTP server.

Configure an sFTP Connection

Enter sFTP connection information so your application can process requests.

> **Note:**
>
> File Server only supports this type of connection.

1. Go to the **Properties** section.
2. Enter the following details:

Element	Description
FTP Server Host Address	Enter the host address of the FTP/FTPS/sFTP server.

Element	Description
FTP Server Port	Enter the FTP server port number. For connecting to File Server, use the port number provided in Oracle Integration on the File Server Settings page. See Configure File Server Settings in *Using File Server in Oracle Integration 3*.
SFTP Connection	Select **Yes** from the list.
Host Key	This is an optional field for adding extra security for host key authentication. Host key authentication is used by a server to verify its identity to a client (the FTP Adapter, in this case). This authentication guards against man-in-the-middle-style attacks. The FTP Adapter currently supports keys of type RSA, ECDSA, and EDDSA/ED-25519 in OpenSSH format for host key verification. a. Click the **Host Key** check box. b. Click **Upload** to upload the host key. • The default location of the RSA key is `/etc/ssh` on the server. • You can identify the host keys supported by the server with the following command: `ssh-keyscan -p port_number Server_IP` • If you are using Oracle Managed File Transfer Cloud Service, download the host key from the Oracle Managed File Transfer Cloud Service Console. • If you are using `vsftpd`, you can find the host key under `/etc/ssh`. See Generate SSH Keys in PEM Format to Connect to a Public or On-Premises sFTP Server.
SFTP Key Exchange Algorithm	This selection is required if you are using an sFTP connection. If your sFTP server is restricted to a specific algorithm, select an algorithm to use from the list. **Note**: If you use the FTP Adapter with the connectivity agent, you must select a value for this field. Do *not* select the algorithm **diffie-hellman-group1-sha1**. This algorithm is not supported with the connectivity agent.

Element	Description

SFTP Preferred PKI Algorithm	This selection is required if you are using host key verification. You must select the algorithm specified in the **Host Key** field. You can find the algorithm specified with the following command:
	`cat path_to_host_key`
	Where *`path_to_host_key`* is the location of the host key present in the local system of the user. The user uploads this key to the Connections page. For example:
	`cat Documents/OIC/FTP_enhance/`
	`test_host_key_rsa.pub`
	This `ssh-rsa` output provides the PKI algorithm to use:
	`ssh-rsa`
	`AAAAB3NzaC1yc2EAAAABIwAAAQEA7LidUL6b`
	`vwlG61oTd/`
	`9InpmNdyB7BuRdJx+D76tn868hNFUg1OFZ24`
	`t7qRrgatKeWH0I3AjDbljSEtvtlK88wEZPn`
	`EJpFNO3YqBaaTdtzQvDpcSWlUVNjf+u2XWET`
	`DNXe5JFCM07q5SRkBO6ja+tfsPyNG3buYvRX`
	`t+/l2V3DllCKDS4iOgr5f6/`
	`DgbKSpvyxduCZje6Vj89rAQwPzCWH1kqA7Wp`
	`wO3`
	`. . .`
	`. . .`
	`/1MEmrJuw==`
	By default, the FTP Adapter is configured to use the ssh-ed25519 host key (if a host key is uploaded, but no PKI algorithm is selected).

Disable Directory Check	Select **Yes** to disable checks of the SFTP directory. This action may be required if you cannot list files in the SFTP directory after an FTP server upgrade. See Cannot List Files in the SFTP Directory After FTP Server Upgrade.

Configure Connection Security

Security policies capture information about how the FTP Adapter must authenticate against the target FTP server.

The following security policies are supported:

- **FTP Server Access Policy**: This policy uses the user name and password for authentication and enables users to configure the PGP values.
- **FTP Public Key Authentication**: This policy connects to the sFTP server using a key. This is used only for sFTP connections. The user enters a user name and uploads the

private key file. A passphrase for the private key is optional. The user can also configure the PGP values in this connection.

- **FTP Multi Level Authentication**: This policy uses multiple independent credentials to log in to the server. This process creates an extra layer of defense against unauthorized users. With this policy, you provide a user name, user password, private key, and private key passphrase to connect to the sFTP server. You also configure the first authentication sequence between the password and the public key.

> **Note:**
>
> File Server does *not* support use of this security policy.

In addition, each security policy provides options for specifying PGP encryption and decryption and signing verification details. You can generate the PGP keys to use. See Generate PGP Keys to Use in Oracle Integration.

- Encryption
  - Provide encryption details if the FTP Adapter connection is used to encrypt the contents while writing the file to a target FTP server.
  - Do not provide encryption details if the file being written to the target FTP server is already encrypted using the stage file action.
- Decryption
  - Provide decryption details if the FTP Adapter connection decrypts the contents while downloading the file from the source FTP server.
  - Do not provide decryption details if the stage file action is used to decrypt the downloaded file.

Signing and verification details:

- Signing
  - Provide signing details if the FTP Adapter connection is used to write a file to the target FTP server that must be optionally signed.
- Signature verification
  - Provide signature verification details if the FTP Adapter connection reads and downloads a file that is digitally signed from the source FTP server.

Configure FTP Connection Security

Configure FTP connection security.

1. Go to the **Security** section.
2. Complete the following fields to configure an FTP connection.

Element	Description
Security Policy	Select **FTP Server Access Policy**.
User Name	Enter the username to connect to the FTP server.

Element	Description
Password	The FTP Adapter supports a nonmanaged connection factory. Enter the password to connect to the FTP server, then enter the password a second time for confirmation.
SSL Certificate Password	If you uploaded an FTP over SSL certificate, enter the password for the .p12 format certificate. Enter the password a second time for confirmation.

3. If required for your integration, specify PGP encryption and decryption and signing verification details. See Configure a PGP Encryption Decryption Connection.

Configure FTP over SSL Connection Security

Configure FTP over SSL connection security.

1. Go to the **Security** section.
2. Complete the following fields to configure an FTP over SSL connection.

Element	Description
Security Policy	Select **FTP Server Access Policy**.
User Name	Enter the username to connect to the FTP server.
Password	The FTP Adapter supports a nonmanaged connection factory. Enter the password to connect to the FTP server, then enter the password a second time for confirmation.
SSL Certificate Password	If you uploaded an FTP over SSL certificate, enter the password for the .p12 format certificate. Enter the password a second time for confirmation.

3. If required for your integration, specify PGP encryption and decryption and signing verification details. See Configure a PGP Encryption Decryption Connection.

Configure sFTP Connection Security

Configure sFTP connection security.

1. Go to the **Security** section.
2. Select a security policy. The fields that are displayed for configuring are based on your selection.

> **Note:**
>
> Public and private keys created using OpenSSH are only supported if they are created on a version below 7.8.

- **FTP Server Access Policy**

- **FTP Public Key Authentication**
- **FTP Multi Level Authentication**

3. If you select **FTP Server Access Policy**:
   a. Complete the following fields.

Element	Description
User Name	Enter the username to connect to the FTP server.
Password	Enter the password to connect to the FTP server. The FTP Adapter supports a nonmanaged connection factory.

   b. If required for your integration, specify PGP encryption and decryption and signing verification details. See Configure a PGP Encryption Decryption Connection.

4. If you select **FTP Public Key Authentication**:
   a. Complete the following fields.

Element	Description
User Name	Enter the username to connect to the FTP server.
Private Key File	If you have a private key, click the checkbox and then click **Upload** to upload the key. You do not need to enter a password to access the server. However, some private keys are encrypted and require a passphrase. If that is the case, enter it in the following field.
PassPhrase	If your private key file is passphrase protected, enter the passphrase here.

   b. If required for your integration, specify PGP encryption and decryption and signing verification details. See Configure a PGP Encryption Decryption Connection.

5. If you select **FTP Multi Level Authentication**:
   a. Complete the following fields.

Element	Description
User Name	Enter the username to connect to the FTP server.
Password	The FTP Adapter supports a nonmanaged connection factory. Enter the password to connect to the FTP server.
First Authentication	This provides the sequence of authentication. If the first authentication is a password, then first password authentication is used. After a successful authentication, public key authentication is performed.

Element	Description
Private Key File	If you have a private key, click the checkbox and then click **Upload** to upload the key. You need to enter a password to access the server. However, some private keys are encrypted and require a passphrase. If that is the case, enter it in the following field.
PassPhrase	If your private key file is passphrase protected, enter the passphrase.

b. If required for your integration, specify PGP encryption and decryption and signing verification details. See Configure a PGP Encryption Decryption Connection.

Configure a PGP Encryption Decryption Connection

Each security policy (FTP Server Access Policy, FTP Public Key Authentication, and FTP Multi Level Authentication) provides options for specifying PGP encryption and decryption and signing verification details.

Connection Type	Supported Security Policy	See
FTP Server	FTP Server Access Policy	Specify PGP Encryption Decryption and Signing Verification for FTP Server Access Policy
FTP over SSL	FTP Server Access Policy	Specify PGP Encryption Decryption and Signing Verification for FTP Server Access Policy
sFTP	• FTP Server Access Policy • FTP Public Key Authentication • FTP Multi Level Authentication	• Specify PGP Encryption Decryption and Signing Verification for FTP Server Access Policy • Specify PGP Encryption Decryption and Signing Verification for FTP Public Key Authentication • Specify PGP Encryption Decryption and Signing Verification for FTP Multi Level Authentication

Specify PGP Encryption Decryption and Signing Verification for FTP Server Access Policy

1. Specify PGP encryption and decryption and signing verification details.

Element	Description
PGP Public Key	If using a PGP public key, click the check box, then click **Upload** to upload the key for encrypting the payload. Pretty Good Privacy (PGP) is a data encryption and decryption program that provides cryptographic privacy and authentication for encrypting and decrypting message files. Message file encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography. Each step uses one of several supported algorithms. Each public key is bound to a user name, an e-mail address, or both.
ASCII-Armor Encryption Format	Select to format the encrypted message in ASCII armor. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content. If not selected, the message is sent in binary format.
Cipher Algorithm	Select the symmetric cryptographic algorithm to use. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. • **CAST5** • **TDES** • **AES128** • **AES192** • **AES256**
Use Secure RNG	If using a ECDSA or EDDSA key pair for PGP signing and verification, select **Yes** from the drop-down list.
PGP Private Key	If using a PGP private key, click the check box, then click **Upload** to upload the key for decrypting the payload.
PGP Private Key Password	Enter the password to encrypt the payload. Enter the password a second time for confirmation.
PGP Sign Public Key	Click the check box, then click **Upload** to upload the public key to verify a digitally-signed certificate.
PGP Sign Private Key	Click the check box, then click **Upload** to create a digitally-signed certificate.
PGP Sign Private Key Password	Enter the sign private key password, then enter the password a second time for confirmation.

Specify PGP Encryption Decryption and Signing Verification for FTP Public Key Authentication

1. Specify PGP encryption and decryption and signing verification details.

Element	Description
PGP Public Key	If using a PGP public key, click the check box, then click **Upload** to upload the key for encrypting the payload. Pretty Good Privacy (PGP) is a data encryption and decryption program that provides cryptographic privacy and authentication for encrypting and decrypting message files. Message file encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography. Each step uses one of several supported algorithms. Each public key is bound to a user name, an e-mail address, or both.
ASCII-Armor Encryption Format	Select to format the encrypted message in ASCII armor. This option is used if you want the encrypted file in readable format. Readable format does not mean that anyone can view the decrypted data. • If you select **Yes**, the file has a `BEGIN PGP MESSAGE` header. • If you select **No**, the file is not readable and has junk characters. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content. If not selected, the message is sent in binary format.
Cipher Algorithm	Select the symmetric cryptographic algorithm to use for encryption. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. • **CAST5** • **TDES** • **AES128** • **AES192** • **AES256**
Use Secure RNG	If using a ECDSA or EDDSA key pair for PGP signing and verification, select **Yes** from the drop-down list.
PGP Private Key	If using a PGP private key, click the check box, then click **Upload** to upload the key for decrypting the payload.
PGP Private Key Password	Enter the password to encrypt the payload. Enter the password a second time for confirmation. If the PGP private key is passphrase-protected, enter the passphrase. Otherwise leave it blank. This field is optional.
PGP Sign Public Key	Click the checkbox, then click **Upload** to upload the public key to verify a digitally-signed certificate.
PGP Sign Private Key	Click the checkbox, then click **Upload** to create a digitally-signed certificate.

Element	Description
PGP Sign Private Key Password	Enter the sign private key password, then enter the password a second time for confirmation. If the PGP sign private key is passphrase-protected, enter the passphrase. Otherwise leave it blank.

Specify PGP Encryption Decryption and Signing Verification for FTP Multi Level Authentication

1. Specify PGP encryption and decryption and signing verification details.

Element	Description
PGP Public Key	If using a PGP public key, click the checkbox, then click **Upload** to upload the key for encrypting the payload. Pretty Good Privacy (PGP) is a data encryption and decryption program that provides cryptographic privacy and authentication for encrypting and decrypting message files. Message file encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and public-key cryptography. Each step uses one of several supported algorithms. Each public key is bound to a user name, an e-mail address, or both.
ASCII - Armor Encryption Format	Select to format the encrypted message in ASCII armor. This option is used if you want the encrypted file in readable format. Readable format does not mean that anyone can view the decrypted data. • If you select **Yes**, the file has a `BEGIN PGP MESSAGE` header. • If you select **No**, the file is not readable and has junk characters. ASCII armor is a binary-to-textual encoding converter. ASCII armor formats encrypted messaging in ASCII. This enables messages to be sent in a standard messaging format. This selection impacts the visibility of message content. If not selected, the message is sent in binary format.
Cipher Algorithm	Select the symmetric cryptographic algorithm to use for encryption. Symmetric-key algorithms for cryptography use the same cryptographic keys for both encryption of plain text and decryption of cipher text. • **CAST5** • **TDES** • **AES128** • **AES192** • **AES256**
Use Secure RNG	If using a ECDSA or EDDSA key pair for PGP signing and verification, select **Yes** from the drop-down list.

Element	Description
PGP Private Key	If using a PGP private key, click the checkbox, then click **Upload** to upload the key for decrypting the payload.
PGP Private Key Password	Enter the password to encrypt the payload, then enter it a second time for confirmation. Enter the password to encrypt the payload. Enter the password a second time for confirmation. If the PGP private key is passphrase-protected, enter the passphrase. Otherwise leave it blank. This field is optional.
PGP Sign Public Key	Click the checkbox, then click **Upload** to upload the public key to verify a digitally-signed certificate.
PGP Sign Private Key	Click the checkbox, then click **Upload** to create a digitally-signed certificate.
PGP Sign Private Key Password	Enter the sign private key password, then enter the password a second time for confirmation. If the PGP sign private key is passphrase-protected, enter the passphrase. Otherwise leave it blank.

Configure the Endpoint Access Type

Configure access to your endpoint. Depending on the capabilities of the adapter you are configuring, options may appear to configure access to the public internet, to a private endpoint, or to an on-premises service hosted behind a fire wall.

- Select the Endpoint Access Type
- Ensure Private Endpoint Configuration is Successful

Select the Endpoint Access Type

Specify an agent group only when the FTP server is not publicly accessible from Oracle Integration. Install the connectivity agent on the same network as the FTP server. This enables the connectivity agent to access the FTP server. Troubleshoot any connectivity agent issues that occur. See Troubleshoot Connectivity Agent Issues with the FTP Adapter.

1. Go to the **Access type** section.
2. Select the option for accessing your endpoint.

Option	This Option Appears If Your Adapter Supports ...
Public gateway	Connections to endpoints using the public internet.

Option	This Option Appears If Your Adapter Supports ...
Private endpoint	Connections to endpoints using a private virtual cloud network (VCN). **Note**: To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Infrastructure Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3* and Troubleshoot Private Endpoints in *Using Integrations in Oracle Integration 3*.
Connectivity agent	Connections to on-premises endpoints through the connectivity agent. a. Click **Associate agent group**. The Associate agent group panel appears. b. Select the agent group, and click **Use**. To configure an agent group, you must download and install the on-premises connectivity agent. See Download and Run the Connectivity Agent Installer and About Creating Hybrid Integrations Using Oracle Integration in *Using Integrations in Oracle Integration 3*.

Ensure Private Endpoint Configuration is Successful

- To connect to private endpoints, you must complete prerequisite tasks in the Oracle Cloud Infrastructure Console. Failure to do so results in errors when testing the connection. See Connect to Private Resources in *Provisioning and Administering Oracle Integration 3*.
- When configuring an adapter on the Connections page to connect to endpoints using a private network, specify the fully-qualified domain name (FQDN) and *not* the IP address. If you enter an IP address, validation fails when you click **Test**.
- IPSec tunneling and FastConnect are not supported for use with private endpoints.

Test the Connection

Test your connection to ensure that it is successfully configured. If necessary, you can self-diagnose connectivity issues that occur with the sFTP server.

1. In the upper right corner of the page, click **Test**.
2. Select the type of connection testing to perform:
   - **Diagnose & Test**: If you receive issues after selecting **Test**, select this option to diagnose sFTP server networking issues. The diagnostics can take more than 15 minutes to perform. Once selected, you cannot cancel this option. After completing network diagnostics, a response is displayed for debugging the issue. Resolve the issues, or, if necessary, contact your network support for additional assistance.
   - **Test**: Performs a normal connection test.

   If successful, the following message is displayed and the progress indicator shows 100%.
   `Connection` ***`connection_name`*** `was tested successfully.`
3. If your connection was unsuccessful, an error message is displayed with details. Verify that the configuration details you entered are correct. Select **Diagnose & Test** to perform further diagnosis.

4. When complete, click **Save**, then click
   <
   .

3

Add the FTP Adapter Connection to an Integration

When you drag the FTP Adapter into the trigger or invoke area of an integration, the Adapter Endpoint Configuration Wizard is invoked. This wizard guides you through configuration of FTP Adapter endpoint properties.

The following sections describe the wizard pages that guide you through configuration of the FTP Adapter as a trigger or invoke in an integration.

Topics:

Basic Info Page

You can enter a name and description on the Basic Info page of each adapter in your integration.

Element	Description
What do you want to call your endpoint?	Provide a meaningful name so that others can understand the responsibilities of this connection. You can include English alphabetic characters, numbers, underscores, and hyphens in the name. You can't include the following characters: • No blank spaces (for example, `My Inbound Connection`) • No special characters (for example, `#;83&` or `righ(t)now4`) except underscores and hyphens • No multibyte characters
What does this endpoint do?	Enter an optional description of the connection’s responsibilities. For example: `This connection receives an inbound request to synchronize account information with the cloud application.`
B2B Trading Partner Mode (Appears only for invoke connections.)	Select this check box only if you plan to use this FTP endpoint for each of the following: • Transfer files to or from a B2B trading partner • Translate file content to or from Electronic Data Interchange (EDI) format with a B2B action in an integration See Manage Trading Partners in *Using B2B for Oracle Integration 3*.

Trigger Configure File Read Page

Specify the following parameters for reading points from the endpoint.

Element	Description
Specify an Input directory	The name of the directory that contains the file to be read. For example: `/tmp/Oracle/input/file` .
Specify a File name pattern	Specify a wild card pattern to use for listing files from the input directory. For example: `order*.csv` .
Specify the Rejection Directory	The name of the directory that contains the file that did not match the schema specified for polling. For example: `/Oracle/rejected` .
Specify the Exclusion Directory	The name of the directory that should not be considered to read files. For example: `/Oracle/exclusion` .
Maximum Files	The maximum number of file names that should be listed. The maximum value is 1000. **Note**: The **List File** operation returns the file list in a sorted order according to the last modified time. If you selected **10** as the maximum number of files and the last modified time of the eleventh file is the same as the tenth file, the eleventh file is also added. This continues until you get a file with a different timestamp. For example, assume the directory has 15 files and you select **10** as the maximum number of files. If the tenth, eleventh, twelfth, and thirteenth files have the same time stamp, the list file returns thirteen files.
Polling Frequency	The frequency at which to poll the files.
Delete Files After Successful Retrieval	Deletes the files that have been successfully retrieved during polling.
Ignore File not found error in Delete	No error is thrown if the file being searched for has already been deleted.
Move to Directory after Successful Retrieval	Moves the files successfully retrieved to a specific directory.

Element	Description
Polling Delay (In Seconds)	Specify the time duration between two polling cycles.
Specify the directory where file is moved	The name of the directory where successfully retrieved files are moved. For example: `/tmp/Oracle/input`
Read Files Recursively	Read files within directories recursively beneath the input directory.

Invoke Dynamic Connections Page

Select to specify a dynamic invoke connection to use at runtime instead of the design-time invoke connection. This page enables you to specify an XPath expression or lookup table result to use with the dynamic connection at runtime.

Element	Description
Enable Dynamic Connection	Click to specify a dynamic connection.
Dynamic Connection ID	Specify a dynamic connection ID in either of two ways: • **Input sources**: Click to pass the connection ID as an XPath expression. The XPath expression can be a combination of the input source and a function. • **Functions**: Click to use a lookup table function to pass different connection IDs representing the different regions. See Dynamically Update Invoke Connections at Runtime.

Invoke Operations Page

Enter the FTP Adapter operation properties. All FTP operations in integrations are synchronous.

Choose one of the following operations to perform on files. Depending on which operation you choose, you are presented with different options and parameters that apply specifically to that operation. All operations support dynamic file name and directory via mapping.

> **Note:**
>
> The operation to read multiple files has been deprecated. This option is not available in the Adapter Endpoint Configuration Wizard when configuring the FTP Adapter connection in a new integration, but can appear in an existing, older integration. As an alternative, Oracle recommends that you use looping functionality (such as a for-each action) to read multiple files.

- Read a File: Reads a file from the FTP/FTPS/sFTP location. The file sizes supported by this operation are based on whether or not you have selected a schema. See Service Limits in *Provisioning and Administering Oracle Integration 3*.
- Write File: Writes a single file onto the target FTP server.
- List Files: Lists the names of the files in the input directory. This operation returns the list of files without any data. It is similar to running the `ls` command in Linux.
- Move a File: Moves a file from one location to another on the same target FTP server.
- Delete a File: Deletes a single file on the FTP server.
- Download File: Specifies a directory to be used for staged activity. Use this directory to stage large files for processing. This operation downloads a single file in the logical directory you specify. Understand the file sizes supported. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

> **Note:**
>
> If you try to decrypt an encrypted file that is less than 1 GB in size, and after decryption it exceeds 1 GB, the operation fails. This also applies to ZIP files. If a ZIP file is less than 1 GB and you select to unzip it, if the file size exceeds 1 GB after unzipping, the operation also fails.

> **Note:**
>
> If you want to read/download multiple files, use the following integration pattern. FTP List File (Operation) > For Each Loop > FTP Read File (Operation) > End For Each Loop

The following tables describe the key information on the FTP Adapter invoke Operations page. Depending on which operation you choose, you are presented with different options and parameters that apply specifically to that operation. Each table corresponds to a specific operation.

Each operation can be performed against ZIP and GZIP file formats.

> **Note:**
>
> Elements marked with “+” can be mapped in the mapper and can be passed dynamically at runtime.

Read a File

When using the connectivity agent, the **Read a File** operation when used without a schema supports files of up to 1 GB in size. You can specify sample XML and JSON documents for the payload when configuring the FTP Adapter for read operations. Understand the file sizes supported based on whether or not you are using a schema. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Element	Description
Select a Transfer Mode	Select the transfer mode: • **ASCII**: Transfers special control characters to format the data. • **Binary**: Transfers raw bytes of the file data.
+Input Directory	The name of the directory that contains the file to be read. For example, `/tmp/Oracle/input`.
+File Name	The name of the file to be read. The name can also be provided by the mapper. Understand the file sizes supported. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Write File

When using the connectivity agent, the **Write File** operation when used without a schema supports files of up to 1 GB in size. This limit is the same as the **Download File** operation. You can specify sample XML and JSON documents for the payload when configuring the FTP Adapter for write operations. Understand the file sizes supported based on whether or not you are using a schema. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Element	Description
Select a Transfer Mode	Select the transfer mode: • **ASCII**: Transfers special control characters to format the data. • **Binary**: Transfers raw bytes of the file data.
+Output Directory	The directory path to which to write the file. For example, `/tmp/Oracle/output`.
+File Name Pattern	The pattern of file names to transfer to the output directory. Use the pattern inside `%%`. For example, `Oracle%SEQ%ICS.txt` creates files in sequence, such as `Oracle1ICS.txt`, `Oracle2ICS.txt`, and so on. For a list of supported file patterns, click the information icon. **Note:** Use of these patterns leads to message loss if the messages are written to the file system at the same time or files are written to a separate node. To ensure that files are not overwritten, use the mapper to assign the file names explicitly. You can use XPath functions to ensure that file names are unique.

Element	Description
Append to Existing File	If selected, the file content is appended to the existing file content and is not overwritten. **Notes**: • If you append JSON or XML content, the final file is invalid XML or JSON. • Writing record by record using the **Append to Existing File** option creates too many network calls and eventually slows down the process. Instead, use a stage file action and write the records to a stage directory. Once all records are written, use the **List File** option in the stage file action and the FTP Adapter **Write File** option to transfer the file to an FTP location. This approach reduces processing time and prevents too many calls to the FTP server. • You may be performing successive appends by putting multiple write operations or using a write operation in a for-each loop. In this scenario, if the FTP server is running on multiple nodes or on a cloud service, it should synchronize the data immediately among all nodes. Otherwise, if the next request goes to a different node, there is inconsistency in the final output file. As an alternative, you can also place a wait activity before or after every append to give the data time to synchronize.
PGP Encryption / Decryption	Oracle Integration can perform PGP encryption using a public key on the file to send to external FTP servers for protecting sensitive data and preserving confidentiality and privacy. Oracle Integration can also perform PGP decryption using a private key on the incoming file to decrypt the encrypted contents. • **Perform PGP Encryption on the file to be sent to external FTP Server**: Encrypts the data. This requires that you specify a public key, ASCII Armor, and cipher algorithm on the Connections page. • **Perform PGP Decryption on an encrypted file to be sent to external FTP Server as clear text**: Decrypts the encrypted data. This mode requires that you create a private key and private key password on the Connections page. • **No PGP Encryption/Decryption on the file to be sent to external FTP Server**: No encryption or decryption is performed.
Include Modification Detection Code	Select this check box when using Elliptic-Curve Diffie–Hellman (ECDH) keys to encrypt/decrypt a file. Otherwise, you receive an error in the activity stream during decryption at runtime. See PGP File Decryption Failure When Using the Elliptic-Curve Diffie–Hellman (ECDH) Key Pair. This selection is optional when using RSA keys to encrypt/decrypt a file.

Element	Description
Sign / Verify Signature	Oracle Integration can perform signing using the private key to allow the receiver to verify that the file contents were not altered during transit. Oracle Integration can also perform signature verification using the public key on the incoming file to verify that the contents were not altered during transit. • **Perform signing on the file to be sent to external FTP Servers**: Generates a signature only and attaches it. This requires that you create an ASCII armor, cipher algorithm, sign private key, and sign private key password for signing on the Connections page. • **Perform verification on the incoming signing file that needs to be sent to external FTP Server**: Verifies the signature present in the data. This requires that you create a sign public key on the Connections page. • **No Signing/Verification on the file to be sent to external FTP Server** : No signing or signing verification is performed.

List Files

Element	Description
+Input Directory	The name of the directory that contains the file to be read. For example, `/tmp/Oracle/input`.
+File Name Pattern	Specify a wild card pattern to be used for listing files from the input directory. For example: `order*.csv`.
Exclude File Name Pattern	Specify a wild card pattern to be used while excluding or ignoring files from the input directory. For example, `order*.csv`.
Max Files	The maximum number of file names that should be listed. The maximum value is 1000. **Note**: The **List File** operation returns the file list in a sorted order according to the last modified time. If you selected **10** as the maximum number of files and the last modified time of the eleventh file is the same as the tenth file, then the eleventh file is also added. This continues until you get a file with a different timestamp. For example, assume the directory has 15 files and you select **10** as the maximum number of files. If the tenth, eleventh, twelfth, and thirteenth files have the same time stamp, then the list file returns thirteen files.
Minimum Age	The minimum age in seconds for files to be displayed. For example, if the last time a file was modified is 02:28:45 AM and the minimum age is defined as 80 seconds, the file will not be listed at 02:29:00 AM and 02:30:00 AM. It will be listed after 02:30:05.
List Files Recursively	List files within directories recursively beneath the input directory.

Element	Description
Ignore File Permissions	The associated file permissions are not considered while listing.
Use NLST [DEPRECATED]	Provides a list of only the file names and the corresponding directory names.

> **Note:**
>
> Oracle Integration plans to deprecate the NLST feature. You are advised not to use it moving forward.

Move a File

Element	Description
+Directory Path	The name of the directory that contains the file to be moved. For example, `/tmp/Oracle/input`.
+File Name	The name of the file to be moved. There is no limit to the size of the file to move.
+Target Directory Path	The name of the directory to which the file will be moved.
+Target File Name	The name of the file as it will be written in the target directory.
Overwrite	Select to overwrite the file in the target directory.

Delete a File

Element	Description
+Directory Path	The directory path to the file you wish to delete.
+File Name	The name of the file to delete.

Download File

The **Download File** operation enables you to map the download directory dynamically. This works fine for new integrations. You can also edit an existing **Download File** operation in the Adapter Endpoint Configuration Wizard to add more file download operations, then click **Done** to regenerate the adapter artifacts. You can download file sizes of up to 1 GB. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Using the **Download File** operation with the connectivity agent enables you to upload the file from the connectivity agent server to Oracle Integration. This operation enables you to download and upload files of up to 1 GB in size. This operation may take more time depending on the network conditions between FTP and the connectivity agent and the connectivity agent and Oracle Integration. Understand the file sizes supported based on whether or not you are using a schema. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

Element	Description
Select a Transfer Mode	Select the transfer mode: • **ASCII**: Transfers special control characters to format the data. • **Binary**: Transfers raw bytes of the file data.
+Input Directory	Specify the name of the directory that contains the file to be read. For example, `/tmp/Oracle/input`.
+File Name	The name of the file to download. This can be overridden using the mapper.
+Download Directory	The name of the directory to be used for stage file action. See Processing Files in Schedule Integrations with a Stage File Action. Do not enter the directory path in double quotes. The quotes are mistakenly included as part of the directory name.
Perform unzip on compressed file downloaded from external FTP server	Select if the file to download is in compressed mode. Selecting this option causes the **Retain the zip directory structure** option to be displayed. If selected, this option preserves the directory structure while unzipping content in the download directory.
Perform PGP Decryption on an encrypted file downloaded from external FTP server	Select if the file to download was encrypted by the sender using PGP encryption to protect sensitive data and preserve confidentiality and privacy. The private key required for decryption must be uploaded to the connection specific to this FTP server.
Perform verification on the signed file downloaded from external FTP server	Select if the file to download must have signature verification to ensure that the file contents were not altered during transit and it came from the actual sender. The corresponding public key required for signature verification must be uploaded to the connection specific to this FTP server.

Trigger or Invoke Schema Page

Enter the schema properties.

Element	Description
Do you want to specify the structure for the contents of the file?	• **Yes**: Select to define a schema format to use for the file to transfer. This option enables you to read and write files. Understand the file size supported. See Service Limits in *Provisioning and Administering Oracle Integration 3*. • **No**: Select if a schema is not required and you want to send the file as an attachment. You typically select **No** if you want to stream large files to or from the servers. This option enables you to read and write files up to 1 GB in size.

Element	Description
Which one of the following choices would be used to describe the structure of the file contents?	Select an option: • **Sample delimited document (e.g. CSV)**: Select to create a new schema file from a comma-separated value (CSV) file. On a subsequent page of this wizard, you are prompted to select the CSV file from which to create the schema. • **XML schema (XSD) document**: Select an existing schema file. On a subsequent page of this wizard, you are prompted to select the existing schema (XSD) file from the file system. You can also upload non-native schemas that are created outside of Oracle Integration. See XSD File Capabilities. • **Sample XML document (Single or No Namespace)**: Select to provide sample XML content for the payload. • **Sample JSON document**: Select to provide sample JSON content for the payload.

XSD File Capabilities

- The adapter supports the upload of an XSD file without a target namespace. In these cases, a surrogate namespace is added to the XSD file that all messages then use:

  ```
  http://xmlns.oracle.com/cloud/adapter/nxsd/surrogate
  ```

- The adapter supports complex XSDs that can import and include other XSDs. The included XSDs in the ZIP file can import the XSD from an HTTP location. All XSD files must be added to a ZIP file and uploaded when configuring the adapter for read and write operations in the Adapter Endpoint Configuration Wizard.

  In the following example, the hierarchy of the ZIP file to upload is as follows:

  ```
  zipxsd.zip
    first.xsd
    second (folder)
      second.xsd
  ```

  `first.xsd` imports `second.xsd`.

  ```
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://xmlns.oracle.com/first"
  targetNamespace="http://xmlns.oracle.com/first"
        xmlns:tns1="http://xmlns.oracle.com/second">
  <xs:import schemaLocation="./second/second.xsd"
  targetNamespace="http://xmlns.oracle.com/second"/>
  <xs:import schemaLocation="https://example.com/fscmService/ItemServiceV2?
  XSD=/xml/datagraph.xsd" targetNamespace="commonj.sdo"/>
  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="isbn" type="xs:string"/>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="author" type="tns1:author"/>
      </xs:sequence>
  ```

```
  </xs:complexType>
</xs:element>
</xs:schema>
```

The contents of `second.xsd` are as follows.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://xmlns.oracle.com/second"
targetNamespace="http://xmlns.oracle.com/second">
<xs:import schemaLocation="https://example.com/fscmService/ItemServiceV2?
XSD=/mycompany/apps/scm/productModel/items/itemServiceV2/
ItemAttachment.xsd"
targetNamespace="http://xmlns.oracle.com/apps/scm/productModel/items/
itemServiceV2/"/>
<xs:complexType name="author">
    <xs:sequence>
       <xs:element name="name" type="xs:string"/>
       <xs:element name="address" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Admin">
    <xs:complexType>
          <xs:sequence>
             <xs:element name="AdminName" type="xs:string"/>
             <xs:element name="AdminAdd" type="xs:string"/>
          </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

> **Note:**
>
> If you are importing from HTTPS locations, ensure that you import the SSL certificates into Oracle Integration.

Trigger or Invoke File Contents - Definition Page

Enter the format definition parameters.

The fields that display on the Format Contents - Definition page are determined by your selection on the Schema page:

- Creating a New Schema from a CSV File
- Select an existing XML schema or schema archive from the file system
- Provide a sample XML document from the file system
- Provide a sample JSON document from the file system

> **Note:**
>
> - If configuring the adapter in the trigger (inbound) direction, schema selection is *not* supported. If you select **Yes** to define a schema for the endpoint on the Basic Info page, nothing prevents you from uploading a schema on the Format Contents - Definition page. However, this schema is not used. You must select **No** on the Basic Info page to transfer files as an attachment.
> - The adapter can only be configured as an invoke connection in an integration.

Creating a New Schema from a CSV File

Element	Description
Select the Delimited Data File	Select the delimited comma-separated value (CSV) file from which to create the schema file. The content of the file is then displayed at the bottom of the page.
Enter the Record Name	Enter the record name. This becomes the parent element in the created schema file for the record names selected as column headers from the CSV file.
Enter the Recordset Name	Enter the recordset name. This becomes the root element of the created schema file.
Select the Field Delimiter	Select one of the following supported file delimiter options: • Single space • Comma • Semicolon • Tab • Pipe (for example, Name|City|Country)
Character Set	Select a character set. The selected value will be used as the encoding format while reading the sample data file. This field is used for character encoding during file transfer. If data sent to the adapter is in a specific encoding format, then select that same encoding format in the adapter. Otherwise, there may be some character loss in the final written file. This also corrupts the file. For example, if a REST Adapter is using UTF-8 encoding and the adapter is configured with the ASCII character set, this may corrupt the file. If you select the same UTF-8 encoding in the adapter, the problem is resolved.

Element	Description
Optionally Enclosed By	This value causes occurrences of the selected delimiter to be ignored during processing. For example, when processing the following record: `Fred,"2 Old Street, Old Town,Manchester",20-08-1954,0161-499-1718` If the selected **Field Delimiter** is “,” and the **Optionally Enclosed By** value is `quot;` (“), then the value `2 Old Street, Old Town,Manchester` is treated as a single record column. If **Optionally Enclosed By** is used, the character must not be part of any field. Therefore, the following is invalid: • `a,b",c` (In this scenario, `"` is part of the second field) • `a,"b,c",d` (In this scenario, the second field is created using **Optionally Enclosed By** (`'"'`). However, it has `','` as content for the second field value.)
Terminated By	Displays by default the first row of the selected CSV file as the column headers. Select the option to terminate the end of a line. • **${eol}** • **\n** • **\r\n** • **\r**
Detach	Select to edit the CSV file in a separate window.
Use First Row as Column Headers	Select to use the first row as the column headers.
Mark All As Optional	Select to mark elements as optional in the schema file. By default, all elements are mandatory. You can also select the data type (for example, string, byte, integer, and so on) to use for each column in the table and mark specific elements as optional. While this option enables you to select all elements as optional, you must have at least one mandatory element to validate this page. This checkbox provides a convenient method to select the majority of elements as optional.

Select an existing XML schema or schema archive from the file system

The schema archive can have a single top level schema with nested input and include elements containing absolute or relative paths.

Element	Description
Select File	Select the existing schema file to use.
Selected File Name	Displays the selected schema file name.
Select the Schema Element	Select the schema element. This field is displayed after the schema file is selected. The element name is treated as the root element in the uploaded schema file.

Provide a sample XML document from the file system

The XML document should contain no namespaces or only a single namespace.

Element	Description
Select File	Select the existing XML document to use.
Selected File Name	Displays the selected schema file name.
Select the Schema Element	Select the schema element. This field is displayed after the schema file is selected. The element name is treated as the root element in the uploaded schema file.

Provide a sample JSON document from the file system

Element	Description
Select File	Select the existing JSON file to use.
Selected File Name	Displays the selected file name.
Select the Schema Element	Select the schema element. This field is displayed after the schema file is selected. The element name is treated as the root element in the uploaded schema file.

Summary Page

You can review the specified adapter configuration values on the Summary page.

Element	Description
Summary	Displays a summary of the configuration values you defined on previous pages of the wizard. The information that is displayed can vary by adapter. For some adapters, the selected business objects and operation name are displayed. For adapters for which a generated XSD file is provided, click the XSD link to view a read-only version of the file. To return to a previous page to update any values, click the appropriate tab in the left panel or click **Go back**. To cancel your configuration details, click **Cancel**.

4 Install and Configure FTP Over SSL on Solaris and Linux

You can install and configure secure FTP for Solaris and Linux:

Topics:

Install and Configure OpenSSL

OpenSSL is an open source implementation of the SSL protocol. OpenSSL implements basic cryptographic functions and provides utility functions. Install and configure OpenSSL on the Solaris or Linux host to be used as the FTP server.

1. Go to the following URL:

   ```
   http://www.openssl.org/source
   ```

2. Locate `openssl-0.9.7g.tar.gz` in the list of available files. For example:

   ```
    3132217 Apr 11 17:21:51 2005 openssl-0.9.7g.tar.gz (MD5) (PGP sign)
   ```

3. Download the following files:
   - `openssl-0.9.7g.tar.gz`
   - `openssl-0.9.7g.tar.gz.md5` (under the `MD5` link)
   - `openssl-0.9.7g.tar.gz.asc` (under the `PGP sign` link)

4. Unzip the following file using `gunzip`.

   ```
   gunzip openssl-0.9.7g.tar.gz
   ```

5. Untar the following file:

   ```
   tar xvf openssl-0.9.7g.tar
   ```

6. Change directories to the following location:

   ```
   cd openssl-0.9.7g
   ```

7. Run the following command:

   ```
   ./config --prefix=/usr --openssldir=/usr/local/openssl
   ```

8. Change to the Bourne shell (if you are not using it):

   ```
   sh
   ```

9. Configure and export the `PATH` variable:

   ```
   PATH=${PATH}:/usr/ccs/bin; export PATH
   ```

10. Run the following command:

    ```
    make
    ```

11. Exit the Bourne shell:

    ```
    exit
    ```

12. Run the following command:

    ```
    make test
    ```

13. Log in as the super user:

    ```
    msu
    ```

14. Enter the password when prompted.

15. Run the following command:

    ```
    make install
    ```

Install and Configure vsftpd

The vsftpd server is a secure and fast FTP server for UNIX systems. Install and configure vsftpd on the Solaris or Linux host to be used as the FTP server.

1. Go to the following location:

   ```
   ftp://vsftpd.beasts.org/users/cevans/
   ```

2. Download `vsftpd-2.0.5` (You must have the tar and signature file (`.asc` file)). For example:

   ```
   [BINARY]     vsftpd-2.0.5.tar.gz. . . . . . . . . . .    [Mar 19 21:26]    149K
   [FILE]       vsftpd-2.0.5.tar.gz.asc. . . . . . . . .    [Mar 19 21:26]    189B
   ```

3. Unzip the following file using `gunzip`.

   ```
   gunzip vsftpd-2.0.5.tar.gz
   ```

4. Unzip the tar file:

   ```
   tar xvf vsftpd-2.0.5.tar
   ```

5. Change directories to the following location:

   ```
   cd vsftpd-2.0.5
   ```

6. Make the following change in the `builddefs.h` file:

   ```
   #undef VSF_BUILD_SSL
   ```

   to

   ```
   #define VSF_BUILD_SSL
   ```

7. Log in as the super user:

   ```
   msu
   ```

8. Enter the password when prompted.

9. Create a file named `vsftpd.conf` with the following settings in the `/etc` directory:

   ```
   # Standalone mode
   listen=YES
   max_clients=200
   max_per_ip=4
   ```

```
# Access rights
anonymous_enable=YES
#chroot_local_user=YES
#userlist_enable=YES
ftp_username=ftp
local_enable=YES
write_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
anon_other_write_enable=YES
chown_uploads=YES
chown_username=ftp
# Security
anon_world_readable_only=NO
allow_anon_ssl=YES
ssl_enable=YES
connect_from_port_20=YES
hide_ids=YES
pasv_min_port=40000
pasv_max_port=49999
# Features
ftpd_banner="Welcome to the FTP Service"
xferlog_enable=YES
ls_recurse_enable=NO
ascii_download_enable=NO
async_abor_enable=YES
# Performance
one_process_model=NO
idle_session_timeout=120
data_connection_timeout=300
accept_timeout=60
connect_timeout=60
anon_max_rate=50000
```

> **Note:**
>
> Copies of the `vsftpd.conf` file appear in several locations in the `vsftpd-2.0.5` directory structure. If you use one of those files to create the `vsftpd.conf` file in the `/etc` directory, ensure that it only includes the parameters and settings described in Step 9.

10. Run the following commands:

```
mkdir /var/ftp
useradd -d /var/ftp ftp
chown root /var/ftp
chmod og-w /var/ftp
mkdir /usr/share/empty
mkdir /usr/share/ssl
mkdir /usr/share/ssl/certs
```

11. Run the following command:

```
openssl req -x509 -nodes -newkey rsa:1024 -keyout /usr/share/ssl/certs/vsftpd.pem -
out /usr/share/ssl/certs/vsftpd.pem
```

12. Run the `vsftpd` daemon from the `vsftpd-2.0.5` directory:

```
./vsftpd
```

Create PKCS#12 Certificates and Keys

You can manage and edit security credentials by creating PKCS#12 certificates and keys.

1. Export `vsftpd.pem` from Step 11 of Install and Configure vsftpd into PKCS#12 format:

   ```
   openssl pkcs12 -export -out vsfptd.p12 -in vsfptd.pem  -inkey vsftpd.pem
   ```

5

Implement Common Patterns Using the FTP Adapter

You can use the FTP Adapter to implement the following common patterns.

Topics:

- Dynamically Update Invoke Connections at Runtime
- Generate PGP Keys to Use in Oracle Integration
- Connect to a Private FTP Server Using the FTP Adapter

> **Note:**
>
> Oracle Integration offers a number of prebuilt integrations, known as *recipes*, that provide you with a head start in building your integrations. You can start with a recipe, and then customize it to fit your needs and requirements. Depending upon the solution provided, a variety of adapters are configured in the prebuilt integrations. See the Recipes and Accelerators page on the Oracle Help Center.

Dynamically Update Invoke Connections at Runtime

You can dynamically update the invoke connection to use at runtime. This feature lets you use a single integration to send requests to multiple end systems or instances of the same application and eliminates the need to create a separate integration for each endpoint or multiple invoke connections configured in a switch action in a single integration.

- Dynamic Connection Capabilities
- Configure and Use a Dynamic Connection

Dynamic Connection Capabilities

Dynamic connections provide the following capabilities.

- Connections can be local to a project, shared between projects, or standalone (that is, available globally and not under the context of a project).
- Each connection can be defined with different properties and security policies (including connectivity agent configurations). For example, an FTP over SSL invoke connection configured with the FTP Public Key Authentication security policy at design-time can be overridden at runtime by an sFTP connection configured with the FTP Server Access Policy security policy.
- The runtime connection to use is defined by a property in the payload of the incoming message or through the result of a lookup table function.

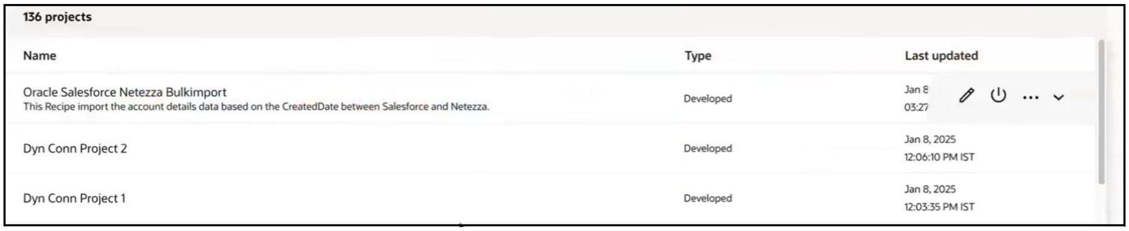
> **Note:**
>
> Lookup table-based expressions are supported only in the design-time lookup wizard and not in the mapper at runtime.

- For any invoke connection, you can specify an XPath expression or optionally a lookup table function result on the source payload. Once evaluated against the actual data, the value of the connection ID to use at runtime is provided for that particular invoke connection.
- The invoke connection used dynamically can be tracked and monitored in the activity stream at runtime.

Configure and Use a Dynamic Connection

This example walks you through how to create and use a dynamic connection across two projects. You can also create and use dynamic connections in a single project or in standalone environments (that is, outside of projects).

For this example, two projects are created.

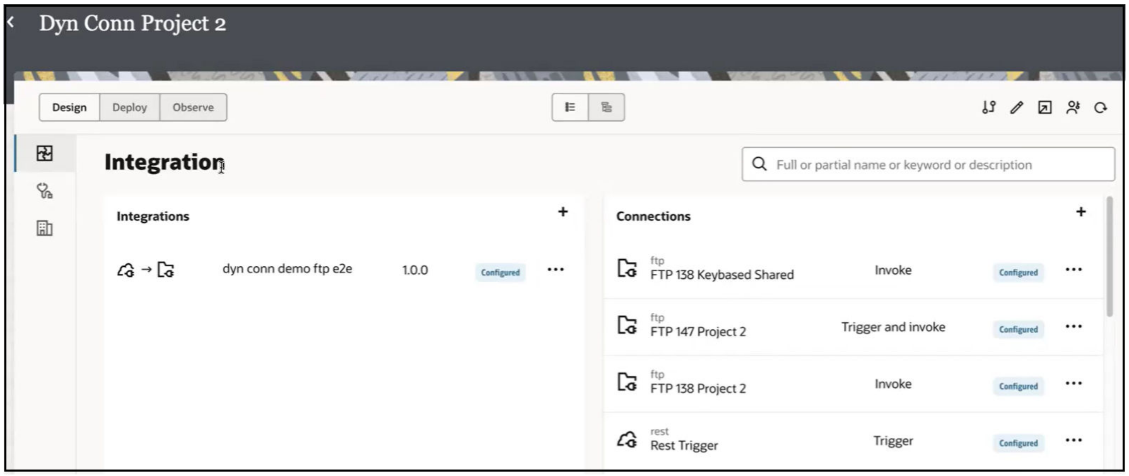


136 projects

Name	Type	Last updated
Oracle Salesforce Netezza Bulkimport This Recipe import the account details data based on the CreatedDate between Salesforce and Netezza.	Developed	Jan 8 03:27
Dyn Conn Project 2	Developed	Jan 8, 2025 12:06:10 PM IST
Dyn Conn Project 1	Developed	Jan 8, 2025 12:03:35 PM IST

Project **Dyn Conn Project 2** consists of the following:

- Integration **dyn conn demo ftp e2e**.
- Connection **FTP 138 Keybased Shared**: This key-based connection (FTP Public Key Authentication) is shared across both projects. The use of this connection dynamically at runtime in **Dyn Conn Project 1** is demonstrated in this section.
- Connection **FTP 147 Project 2**: This is a user name/password-based connection (FTP Server Access Policy).
- Connection **FTP 138 Project 2**: This is a user name/password-based connection (FTP Server Access Policy).
- Connection **Rest Trigger**: This connection triggers the integration.

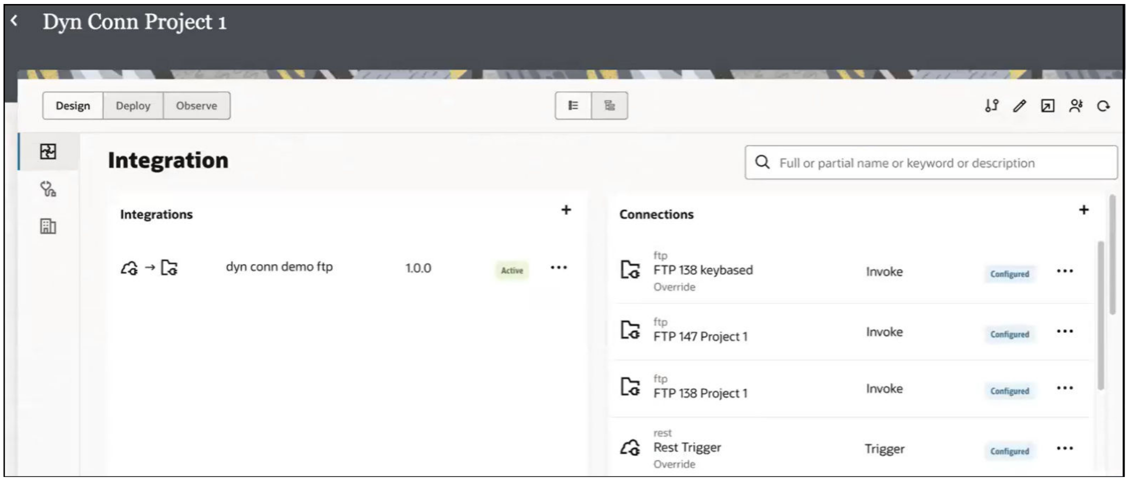


Integrations	Version	Status
dyn conn demo ftp e2e	1.0.0	Configured

Connections	Role	Status
ftp FTP 138 Keybased Shared	Invoke	Configured
ftp FTP 147 Project 2	Trigger and invoke	Configured
ftp FTP 138 Project 2	Invoke	Configured
rest Rest Trigger	Trigger	Configured

Project **Dyn Conn Project 1** consists of the following:

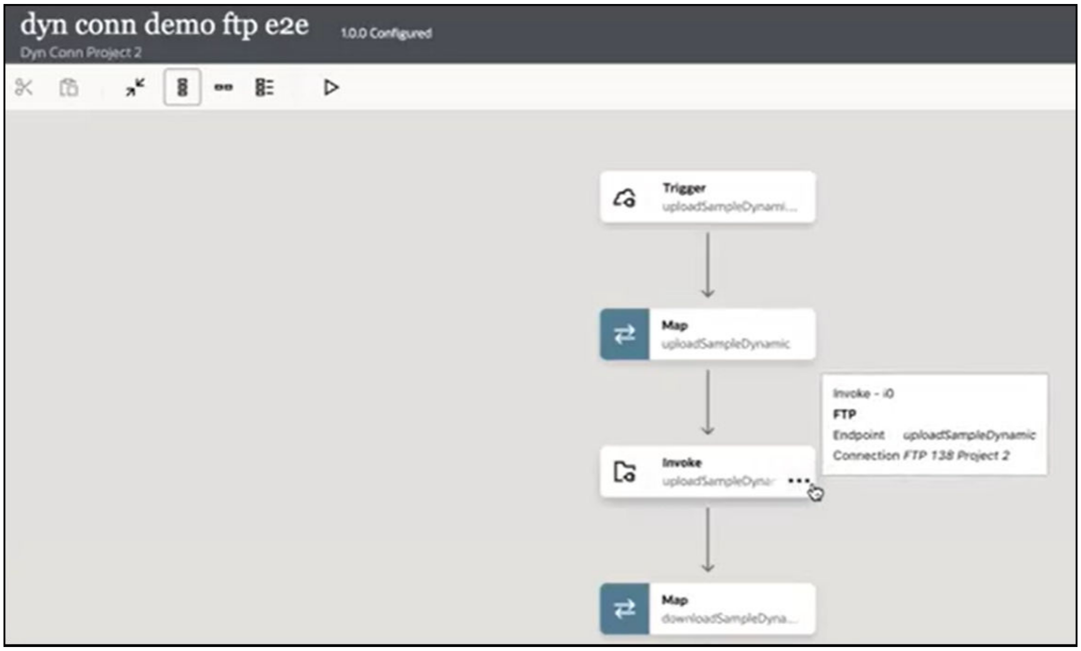
- Integration **dyn conn demo ftp**.
- Connection **FTP 138 Keybased**: This connection references the shared key-based connection (FTP Public Key Authentication) in the **Dyn Conn Project 2** project.
- Connection **FTP 147 Project 1**: This is a user name/password-based connection (FTP Server Access Policy).
- Connection **FTP 138 Project 1**: This is a user name/password-based connection (FTP Server Access Policy).
- Connection **Rest Trigger**: This connection triggers the integration.



Integrations	Version	Status
dyn conn demo ftp	1.0.0	Active

Connections	Role	Status
ftp FTP 138 keybased Override	Invoke	Configured
ftp FTP 147 Project 1	Invoke	Configured
ftp FTP 138 Project 1	Invoke	Configured
rest Rest Trigger Override	Trigger	Configured

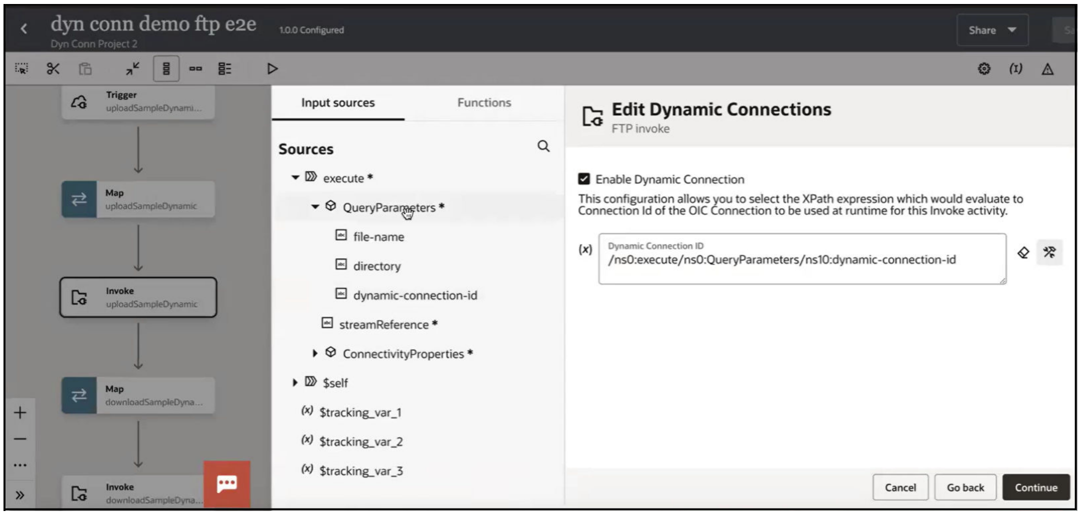
1. In **Dyn Conn Project 2**, open the **dyn conn demo ftp e2e** integration for editing. Note that the **FTP 138 Project 2** FTP Adapter invoke connection is being used in the design.



Invoke - i0
FTP
Endpoint uploadSampleDynamic
Connection FTP 138 Project 2

2. Open this invoke connection for editing in the integration canvas.
3. On the Basic Info page, click **Continue**.
4. On the Dynamic Connections page, select **Enable Dynamic Connection** to specify a dynamic connection ID to use at runtime.
   The **Dynamic Connection ID** field is displayed.
5. Click **Switch to Developer View** .
6. Specify a dynamic connection ID in either of two ways.
   - **Input sources**: Click to pass the connection ID as an XPath expression with the **dynamic-connection-id** query parameter. The XPath expression evaluates to the connection ID of the dynamic invoke connection used at runtime.
   - **Functions**: Click to use a lookup table function to pass different connection IDs representing the different regions.

   For this example, the **dynamic-connection-id** XPath expression is dragged to the field.



Edit Dynamic Connections

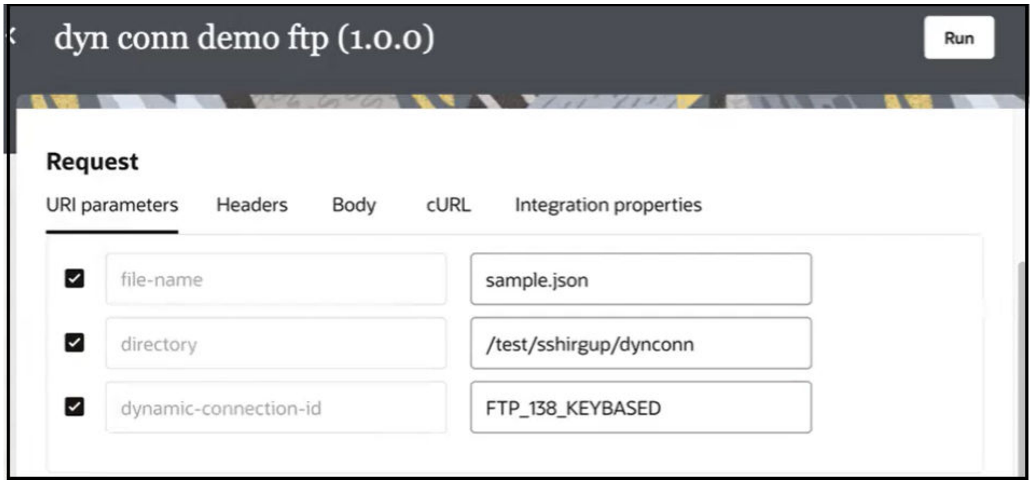
FTP invoke

☑ Enable Dynamic Connection

This configuration allows you to select the XPath expression which would evaluate to Connection Id of the OIC Connection to be used at runtime for this Invoke activity.

Dynamic Connection ID: /ns0:execute/ns0:QueryParameters/ns10:dynamic-connection-id

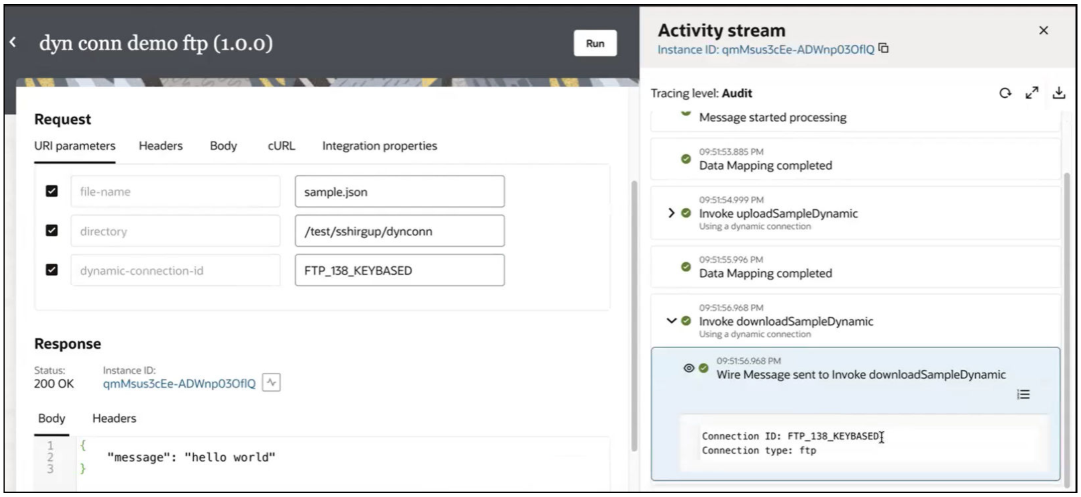
7. Save the updates.
   It's time to run the integration.
8. Go to the **Dyn Conn Project 1** project.
9. Hover over the **dyn conn demo ftp** integration to run.
10. From the **Actions** ••• menu, select **Run**.
11. In the **Body** tab, upload the payload file to use (for this example, named **sample.json**).
12. In the **URI parameters** tab, specify the following:
    - **file-name**: File name of the payload.
    - **directory**: Target directory in which to write the file.
    - **dynamic-connection-id**: FTP Adapter invoke connection to use dynamically at runtime. Begin typing letters to filter the display of invoke connections available for selection. For this example, **FTP 138 Keybased** is selected. This is the shared invoke connection on which you defined the dynamic XPath expression in Step 7. If you leave this field blank, the connection added at design time, and not a dynamic connection, is used.



☑	file-name	sample.json
☑	directory	/test/sshirgup/dynconn
☑	dynamic-connection-id	FTP_138_KEYBASED

13. Click **Run**.

14. Expand the connection and wire message in the activity stream.
    A connection message indicates the dynamic connection was used instead of the design-time connection. The wire message identifies the connection ID used (**FTP 138 Keybased**) and connection type (**ftp**) used.



☑ file-name: sample.json
☑ directory: /test/sshirgup/dynconn
☑ dynamic-connection-id: FTP_138_KEYBASED

Status: 200 OK
Instance ID: qmMsus3cEe-ADWnp03OflQ

```
{
  "message": "hello world"
}
```

Instance ID: qmMsus3cEe-ADWnp03OflQ

Message started processing

09:51:53.885 PM
Data Mapping completed

09:51:54.999 PM
Invoke uploadSampleDynamic
Using a dynamic connection

09:51:55.996 PM
Data Mapping completed

09:51:56.968 PM
Invoke downloadSampleDynamic
Using a dynamic connection

09:51:56.968 PM
Wire Message sent to Invoke downloadSampleDynamic

```
Connection ID: FTP_138_KEYBASED
Connection type: ftp
```

Generate PGP Keys to Use in Oracle Integration

You can generate PGP keys to use in Oracle Integration. The FTP Adapter currently supports PGP keys with RSA algorithms. ECSDA and ED25519 algorithm keys are supported for signing and verification.

Generate the PGP Keys with RSA Algorithms

Perform the following steps to generate the PGP keys with RSA algorithms:

1. Install GnuPG.
   - On Linux:

     ```
     sudo apt-get install gnupg
     ```

   - On the Mac:

     ```
     brew install gnupg
     ```

2. Generate the PGP keys.

   ```
   gpg --full-generate-key
   ```

   a. Select the RSA and RSA option (creates a key to use for both encryption and signing).

      > **Note:**
      >
      > The default option is ECC (Elliptic-curve cryptography). ECC PGP keys are supported for signing/verification.

   b. Select the key size.
   c. Specify the expiration for the key.
   d. Enter details for key identification, when prompted.

3. List the keys (with the fingerprint) you can use.

```
gpg --list-keys
gpg --list-secret-keys
```

4. Export the keys to a file.

```
gpg --export -a keyid > publickeyname.key
gpg --export-secret-key -a keyid > privatekeyname.key
```

5. Get details about the key (that is, what the key consists of).

```
cat path_to_pgp_private.key | gpg --list-packets
```

6. Understand the following about the output.

   - `:secret key packet: version 4, algo 1, created 1687877507, expires 0`

     Where `algo` represents the public key algorithm:

     – `1` for RSA (Encrypt or Sign)
     – `2` for RSA Encrypt-Only
     – `3` for RSA Sign-Only
     – `19` reserved for ECDSA, and so on

     > **Note:**
     >
     > `19` is not supported.

   - `iter+salt S2K, algo: 7, SHA1 protection, hash: 2, salt: C774580FF5CBDF79`

     Where `algo`: represents the Symmetric-Key Algorithm:

     – `7` for AES with 128-bit key
     – `8` for AES with 192-bit key
     – `9` for AES with 256-bit key, and so on.

   - `digest algo 8, begin of digest d9 35`

     Where `digest algo` represents Hash algorithm:

     – `8` for SHA256
     – `9` for SHA384
     – `10` for SHA512

  - `11` for SHA224

Approved/Unapproved Algorithms in FIPS mode:

The following command lists all supported algorithms by GnuPG.

```
gpg --version
```

You can configure GPG keys from the command line. For example:

```
gpg --gen-key --s2k-cipher-algo AES256 --cipher-algo AES256
```

The following algorithms are not approved for use in FIPS mode (FIPS 140-2):

- Cipher algorithms (For data encryption and decryption): CAST5, CAMELLIA128, CAMELLIA192, CAMELLIA256, BLOWFISH, and TWOFISH
- Message digest/hash algorithms: MD5 and RIPEMD
- Digital signature and verification: RSA with keys smaller than 2048 bits

The following algorithms are approved for use in FIPS mode:

- Cipher algorithms (For data encryption and decryption): AES128, AES192, AES256, and 3DES
- Message digest/hash algorithms: SHA1, SHA256, SHA384, SHA512, and SHA224
- Digital signature and verification: RSA with keys greater than 2048 bits

Connect to a Private FTP Server Using the FTP Adapter

You can integrate Oracle Integration with an FTP server, even when that server is in a private network and not accessible publicly. This scenario is possible when you configure the connectivity agent with the FTP Adapter.

The FTP Adapter supports connectivity to the following servers:

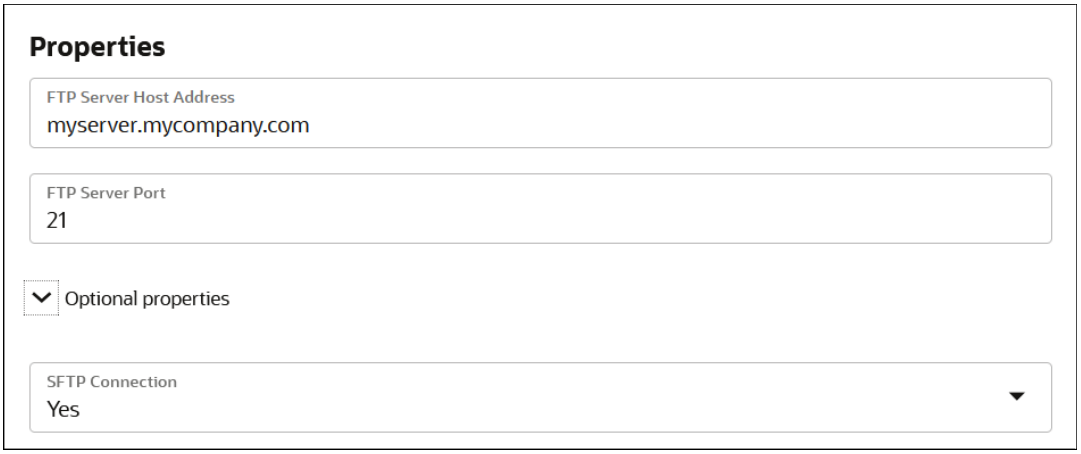
- FTP/sFTP server hosted on-premises through the connectivity agent
- FTP/sFTP server hosted in the cloud without the connectivity agent

See Manage the Agent Group and the On-Premises Connectivity Agent in *Provisioning and Administering Oracle Integration 3*.

Configure Connection Properties

Provide connection property values:

1. In the **Properties** section on the Connections page:
   a. Enter the FTP/sFTP host address and port.
   b. If using a secure FTP server, select **Yes** for the **SFTP Connection** field. Otherwise, select **No**.



Properties

FTP Server Host Address: myserver.mycompany.com

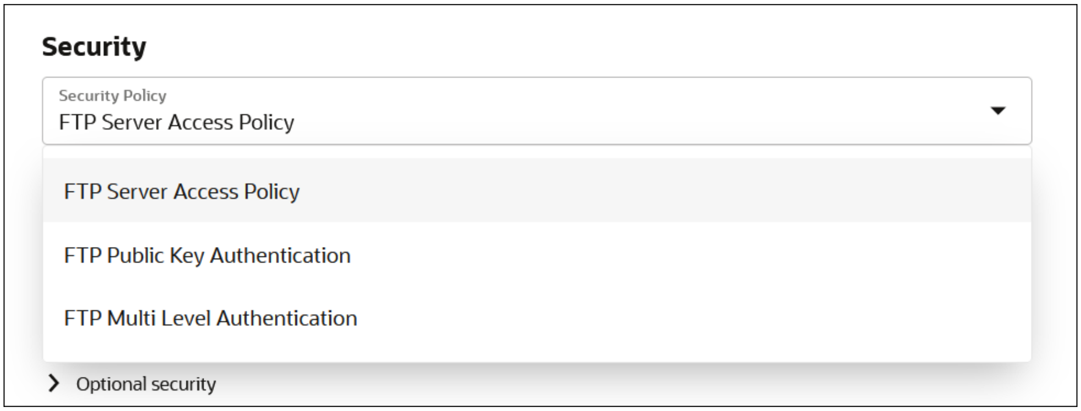
FTP Server Port: 21

SFTP Connection: Yes

Configure Connection Security

Provide connection security values:

1. In the **Security** section on the Connections page, select the security policy:
   - **FTP Server Access Policy**: For username/password authentication.
   - **FTP Public Key Authentication**: For public key authentication.
   - **FTP Multi Level Authentication**: For authentication using both username/password and public key.

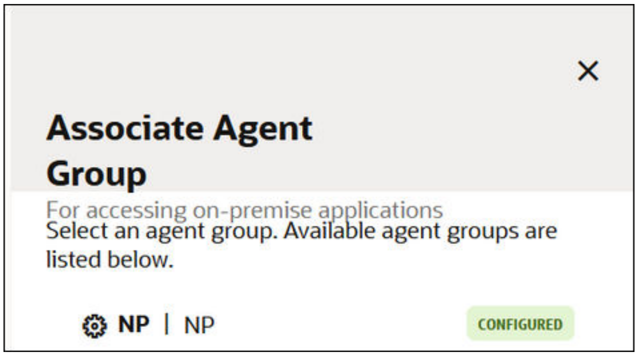


Security

Security Policy: FTP Server Access Policy

Configure Agent Group

If the FTP server is not directly accessible from Oracle Integration (for example, the server is installed on-premises or behind a firewall), you must configure the connectivity agent for this connection. This can be done in the **Associate Agent Group** section.



Associate Agent Group

For accessing on-premise applications

Select an agent group. Available agent groups are listed below.

NP | NP CONFIGURED

There are file size limits when using the FTP Adapter. For example, the **Download File** operation on the Operations page does not support a schema, and can send a file of up to 1 GB in size. The download may take time to complete considering the network latency between the connectivity agent and Oracle Integration. Understand the file sizes supported. See Service Limits in *Provisioning and Administering Oracle Integration 3*.

There are also restrictions when the FTP Adapter is configured with the connectivity agent. See FTP Adapter Restrictions.

6 Troubleshoot the FTP Adapter

Review the following topics to learn about troubleshooting issues with the FTP Adapter.

Topics:

Cannot List Files in the SFTP Directory After FTP Server Upgrade

Upgrading your SFTP server may result in the FTP Adapter not listing files in an SFTP directory. This occurrence is due to stricter enforcement of the SFTP spec and is more prevalent with OpenSSH-based SFTP servers.

If this occurs, perform the following steps:

1. Set the optional property **Disable Directory Check** to **Yes** on the Connections page for the SFTP connection.
2. Save the changes.
3. Test the connection.
4. Reactivate the integration.

See Configure an sFTP Connection.

CASDK-0002: Unable to access the host sftp.*xxxxx.xxxxx* Error

If the connection test for the FTP Adapter fails with the following error:

```
CASDK-0002: Unable to access the host sftp.xxxxx.xxxxx
```

This error occurred because the FTP protocol was selected instead of the SFTP protocol during connection creation on the Connections page.

Perform the following steps:

1. Select **Yes** from the **SFTP Connection** list in the **Connection Properties** section of the Connections page.
2. Retest your connection.

Override Design-Time Endpoint Configurations in the Mapper

You can override design-time endpoint configurations for the FTP Adapter in the mapper. During runtime, the values you specified in the mapper override the configurations you set in the Adapter Endpoint Configuration Wizard.

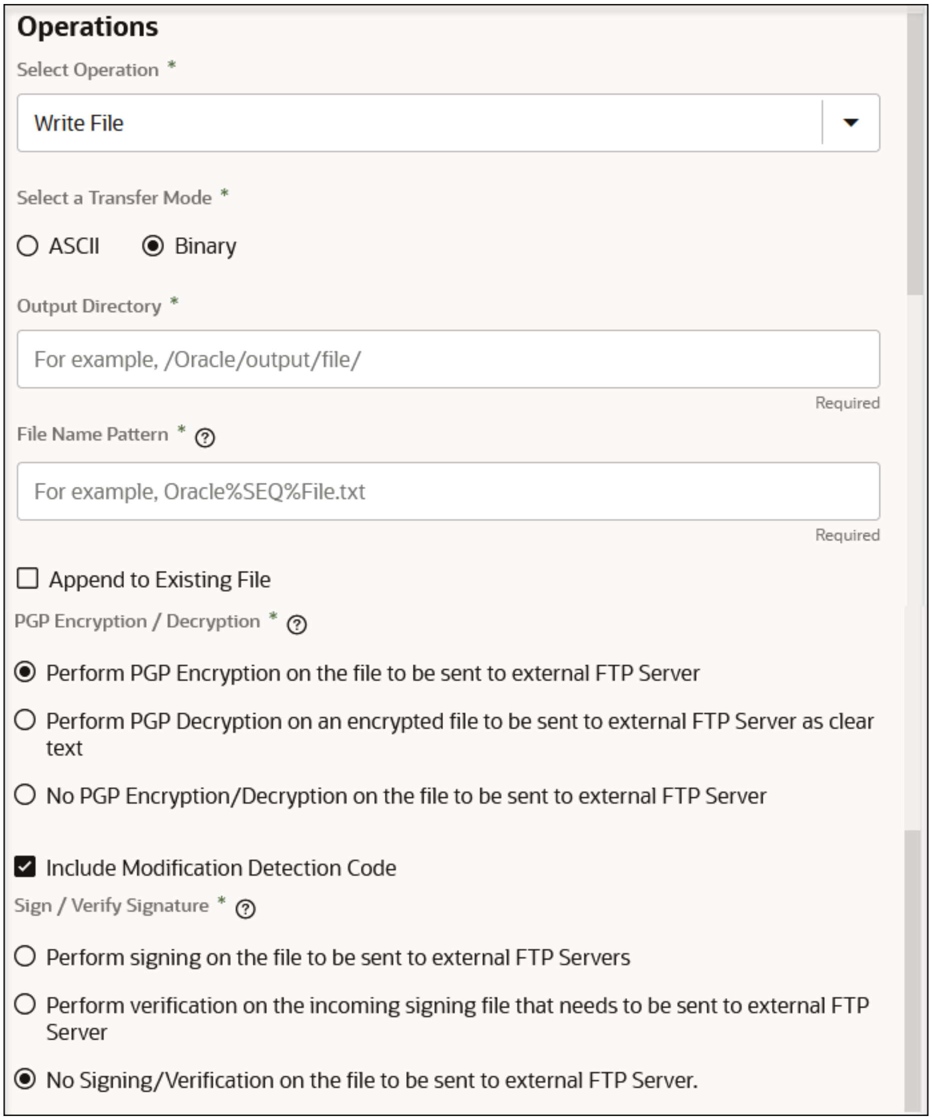
1. In the **Target** (right hand) section of the mapper, expand **FTPWrite Request (FTP)** > **OutboundFTPHeaderType**.
2. Map values to the **directory** and/or **filename** nodes. The mapped values are used at runtime instead of the default values specified during FTP Adapter design-time configuration in the Adapter Endpoint Configuration Wizard.

PGP File Decryption Failure When Using the Elliptic-Curve Diffie–Hellman (ECDH) Key Pair

If using an Elliptic-Curve Diffie–Hellman (ECDH) key pair to encrypt/decrypt a file, you must use the Modification Detection Code (MDC). Otherwise, decryption fails with a `null:Application Error` error at runtime.

If you see this error in the activity stream at runtime during decryption and you are using EDCH keys, you have two options:

- Regenerate your PGP keys using RSA, and then use the RSA keys for encryption/decryption.
- Enable MDC during encryption of the file when using EDCH keys by selecting the **Include Modification Detection Code** checkbox on the Operations page of the Adapter Endpoint Configuration Wizard.



Operations

Select Operation *

Write File ▾

Select a Transfer Mode *

○ ASCII ● Binary

Output Directory *

For example, /Oracle/output/file/

Required

File Name Pattern * ?

For example, Oracle%SEQ%File.txt

Required

☐ Append to Existing File

PGP Encryption / Decryption * ?

● Perform PGP Encryption on the file to be sent to external FTP Server

○ Perform PGP Decryption on an encrypted file to be sent to external FTP Server as clear text

○ No PGP Encryption/Decryption on the file to be sent to external FTP Server

☑ Include Modification Detection Code

Sign / Verify Signature * ?

○ Perform signing on the file to be sent to external FTP Servers

○ Perform verification on the incoming signing file that needs to be sent to external FTP Server

● No Signing/Verification on the file to be sent to external FTP Server.

This MDC requirement applies even if the file is being encrypted by a third party.

See Invoke Operations Page.

LastModifiedTime in the FTP Adapter Response Time

The `LastModifiedTime` in the FTP Adapter response time is in milliseconds. In Java, the time in milliseconds can be obtained with `System.currentTimeInMiliseconds`.

Low Entropy in the System Causes FTP Adapter Timeouts When Using the Connectivity Agent

When running the connectivity agent on Oracle Cloud Infrastructure compute, you may see timeouts/slowness in FTP Adapter invocations to SFTP servers. The FTP Adapter uses cryptographic libraries that need a SecureRandom randomizer to seed cryptographic key generation. Seeding makes use of entropy in the system and an insufficient amount of entropy leads to slowdowns and results in slowness in the FTP Adapter and timeouts in some cases.

To resolve this issue, ensure that the system has an entropy value greater than 3000. Entropy can be obtained on Linux by executing the following command:

```
cat /proc/sys/kernel/random/entropy_avail
```

See doc: 2738113.1 at My Oracle Support to set entropy to more than 3000.

Creation and Release of an FTP Adapter Connection Can Cause Conflict for FTP Servers with Less Connections

The FTP Adapter creates a fresh connection every time and releases it as soon as work is done. This approach creates issues for some FTP servers in which the maximum number of FTP connections are less.

Some FTP servers take a minimum of one minute to flush the connection that is closed before making it available again. Check whether there is a need for multiple flows or multiple instances to read or write to the FTP server at the same time. If so, the FTP server administrator must adjust the number of connections.

You can encounter this issue if you call the FTP Adapter inside a loop. It is recommended that you use the write file operation of a stage file action to append the data and transfer the whole file at once. You can reduce flush time in your server or use a wait action inside a for-each loop for these cases.

Troubleshoot Connectivity Agent Issues with the FTP Adapter

Troubleshoot issues with the connectivity agent based on the type of FTP Adapter connection you are using.

- When Using an FTP Connection Type
- When Using an FTP over SSL Connection Type
- When Using an sFTP Connection Type

When Using an FTP Connection Type

- The FTP server must be publicly accessible when the agent group is not configured.
  Try to connect using FileZilla or `winscp` to see whether the connection is occurring from a public laptop.
- If the FTP server is *not* publicly accessible, ensure that the connectivity agent is installed either on-premises or on a private cloud that has network connectivity with the FTP server. Enter the following command from the connectivity agent host instance.

  ```
  ftp user@host
  ```

- If using privately-hosted FTP servers, ensure that the proxy configured for the connectivity agent to use allows an FTP connection. If not, also try including the FTP host address as the `noproxyHost` for the connectivity agent configuration.

When Using an FTP over SSL Connection Type

- If you have not used the agent group, the FTP server should be publicly accessible.
  Use an FTP client such as `winscp` or FileZilla and enter the correct password to confirm whether it's a connectivity issue.

- If you used the connectivity agent, the FTP server should be accessible from the agent host.
  Enter the following command from the connectivity agent host:

  ```
  ftp user@host
  ```

- If you used a proxy server in the agent configuration, check if the proxy used allows an FTP server connection. Also, try using the FTP host address as the `nonProxyHost` for the agent configuration.

When Using an sFTP Connection Type

- If you have not used the agent group, the sFTP server should be publicly accessible. Verify the connectivity agent connection is working from outside the Oracle network using `winscp` or FileZilla.
- If you are getting a private key or a passphrase is not correct, open the private key file and see whether it has the following header:

  ```
  -----BEGIN RSA PRIVATE KEY-----
  ```

  If the key is any other format, convert the key in this format using PuTTYgen. PuTTYgen is available on Windows.

  Load the keys in PuTTYgen, then export the open SSH key.
- If using multilevel authentication, check the first authentication.
- If using the connectivity agent, the `sftp` command should work from the connectivity agent host.
- If using a proxy on the connectivity agent host, run the `sftp` command using the proxy or add the sFTP host as the `nonProxyHost`.
- If you uploaded the host key on the Connections page and are receiving a `Host Key Exception`, remove the host key and test the connection.

Generate SSH Keys in PEM Format to Connect to a Public or On-Premises sFTP Server

You receive the following error when testing your connection after using an upgraded `ssh-keygen` tool to generate SSH keys in OPENSSH format. OPENSSH is a proprietary format. Oracle Integration requires the keys to be in PEM format.

```
CASDK-0004: Failed to authenticate against the application with the
credentials provided; Private Key or Passphrase is incorrect. Please verify
the Private Key and Passphrase.
```

1. Verify the key by opening the file in Notepad. The key must start with the following phrase. Oracle Integration supports keys in this format:

   ```
   -----BEGIN RSA PRIVATE KEY-----
   ```

The following format is not supported. You must regenerate your keys in PEM format.

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

2. Use `-m PEM` with `ssh-keygen` to generate private keys in PEM format:

   ```
   ssh-keygen -t rsa -m PEM
   ```

Two Integrations Processing Files from the Same Folder is Not Supported

The FTP Adapter does not support the processing of files from the same folder by two integrations. File deletion causes a conflict. For example, if one integration deletes a file, the second integration cannot find the file to delete and fails.

FTP Adapter Cannot Connect to FTP/SFTP Server in Oracle Cloud

When trying to connect from Oracle Integration to an FTP server running in Oracle Cloud, the connection may not work. For example, if Oracle Support Services runs the following command:

```
sftp -o "ProxyCommand /usr/bin/nc -X connect -x  proxy_host:proxy_port
%h %p" FTP_username@FTP_hostname
```

The following error message can be displayed. This occurs because routing is not enabled from Oracle Integration to the FTP server.

```
Connecting to FTP_hostname...
nc: port range not valid
ssh_exchange_identification: Connection closed by remote host
Couldn't read packet: Connection reset by peer
```

Either file a service request to ensure that the proxy allows connections to the port on which the FTP server is running or use an FTP server not running in Oracle Cloud.

Use of FTP Adapter Connections Created Before Release 16.3.3 in Integrations Created in Release 16.3.3 or Later

FTP Adapter connections created before release 16.3.3 had the option to select **User Name Password Token** as the security policy. Beginning with release 16.3.3, this security policy was changed to **FTP Server Access Policy**. If you open the Connections page for an FTP Adapter connection created before release 16.3.3 for editing, the security policy field is empty and the connection is unusable in integrations. To use this connection, you must manually select **FTP Server Access Policy** as the security policy, enter the user credentials, and test the connection. After that, you can use this connection in integrations.

If you created the FTP Adapter connection in release 16.3.3 or later, this problem does not occur.

Read Multiple File Operation of FTP Adapter Not Available in 17.2.5

The Read Multiple File operation is no longer available for selection when configuring the FTP Adapter in the Adapter Endpoint Configuration Wizard. Only backwards compatibility is supported (meaning that a pre-17.2.5 .IAR file imported into Oracle Integration that was designed with the Read Multiple File operation has this option).

Additional integration troubleshooting information is provided. See Troubleshoot Oracle Integration in *Using Integrations in Oracle Integration 3*.

Connect to an On-Premises FTP Server

To make an on-premises FTP server publicly accessible, enable the port for public internet access. After that, you can add firewall rules to restrict or limit IP address access. With limited IP access, you can add Oracle Integration IP addresses so that only Oracle Integration can connect to an FTP server.

7 FTP Adapter Samples

You can use the FTP Adapter in end-to-end scenarios such as the following:

Topics:

Tutorial: Create an Integration to Import and Process Bulk Files

In this tutorial, you'll learn how to create a schedule integration in Oracle Integration to load and transform files from a secure File Transfer Protocol (FTP) location.

For the purpose of this tutorial, let's consider a simple use case where you'll require to communicate with an external application through an FTP server, import raw data, and convert it into a standard format. The flow for this scenario is as follows:

1. A user or an application uploads a file or set of files to an FTP server for processing.
2. The schedule integration imports these files into Oracle Integration using an FTP Adapter.
3. Using a set of relevant of actions, the integration transforms or enriches the data within the files and uploads the modified files back to the FTP server.

Upon completing this tutorial, you'll be familiar with several important skills in Oracle Integration, such as configuring an FTP Adapter, iterating over repeating elements, staging files, mapping data, and creating schedules.

Topics:

Prerequisites to Set Up the Integration

To complete this tutorial, you need:

- Access to Oracle Integration. Don't have access? Use free credits to try Oracle Integration now.
- Sign-in credentials (user name, password, data center/region, and identity domain) for your Oracle Integration user account.
- A secure FTP (sFTP) server.
- An FTP client to access the sFTP server.
- The sample file, *test-data.zip*. To create this file, see Create a Sample Zip File.

Create a Sample Zip File

On your local machine, create a zip file, containing a few individual files, to use in this tutorial.

1. Create a `.txt` file and name it `test-file1`. Add the following data into it.

   ```
   Test Data Row1 From File1
   Test Data Row2 From File1
   ```

2. Create two more `.txt` files and name them `test-file2` and `test-file3`. Add the following data into them, respectively.

   ```
   Test Data Row1 From File2
   Test Data Row2 From File2
   Test Data Row3 From File2
   Test Data Row4 From File2
   Test Data Row5 From File2
   ```

   ```
   Test Data Row1 From File3
   Test Data Row2 From File3
   Test Data Row3 From File3
   ```

3. Create a folder named `test-data` and copy the three files you created into it.
4. Zip the `test-data` folder. Your sample file is now ready to use.

Access Your FTP Server and Upload the File

Obtain an sFTP server and ensure that you're able to access it.

1. Log in to the server using your user name and password through an FTP client; for example, FileZilla.
2. Create a directory (`FTP Bulk Transfer`) on the server from which to read and write files.
3. Upload the sample file (`test-data.zip`) into the directory you've created.

Create an FTP Connection

In the integration you'll set up, you'll use an FTP Adapter to connect to the sFTP server and retrieve files for transformation. In order to use an FTP Adapter, you'll have to first define a connection based on it to your server.

1. In the navigation pane, click **Design**, then **Connections**.
2. Click **Create**.
   The Create Connection — Select Adapter dialog is displayed.
3. Select the FTP Adapter from the dialog.
   The Create Connection dialog is displayed.
4. Enter the information to describe the connection.
   a. Enter a name for your connection (`FTP Connection`).
   b. Select **Trigger and Invoke** in the **Role** field.
   c. Enter an optional description of the connection.
5. Click **Create**.
   Your connection is created and you are now ready to configure other details.
6. In the **Properties** section, enter the following details.

Field	Information to Enter
FTP Server Host Address	Enter the host address of your sFTP server.
FTP Server Port	Enter 22.
SFTP Connection	Select **Yes** from the list.

7. In the **Security** section, enter the following details.

Field	Information to Enter
Security Policy	Select **FTP Server Access Policy**.
User Name	Enter the user name to connect to your sFTP server.
Password	Enter the password to connect to your sFTP server, and then enter the password a second time for confirmation.

8. Click **Save**.
9. Click **Test** to ensure that your connection is successfully configured. In the resulting dialog, click **Test** again. If necessary, you can self-diagnose connectivity issues that occur with the sFTP server, see Test the Connection.
   A confirmation message is displayed if your test is successful.

CONFIRMATION

Connection **FTP Connection** was tested successfully.

10. Click **Back** < to return to the Connections page. Click **Save** again if prompted.
    Now you're ready to create the integration.

Create a Schedule Integration

Let's create a schedule integration in which an invoke FTP Adapter reads the sample zip file from the sFTP server and another invoke FTP Adapter writes the transformed zip file back to the server. You'll create a schedule to run this integration at a specified frequency later in this example.

1. In the navigation pane, click **Design**, then **Integrations**.
2. On the Integrations page, click **Create**.
   The Create Integration - Select a Style dialog is displayed.
3. Select **Schedule** in the Create integration panel.
4. Enter the following information.

Field	Information to Enter
What do you want to call your integration?	Provide a name for the integration, `FTP Bulk Import`.
Identifier	Accept the default identifier value.
Version	Accept the default version number of `01.00.0000`.
What does this integration do?	Enter the following text: `Loads and transforms files from a secure FTP location.`

5. Click **Create**. The integration canvas is displayed, where you can configure your integration.

Configure an FTP Adapter to Download the Zip File

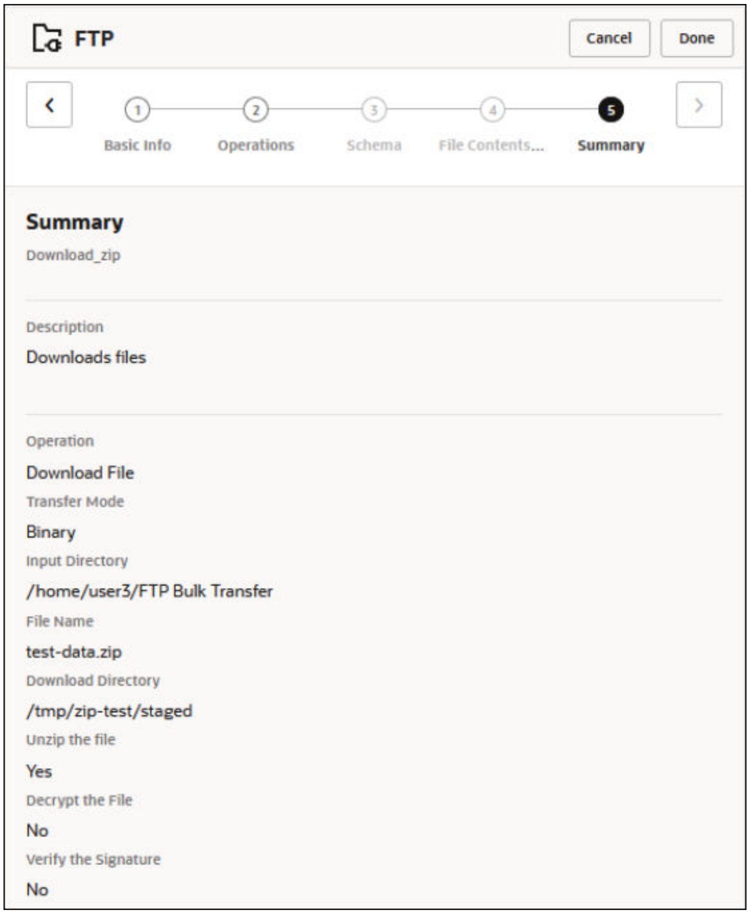
Add an FTP Adapter to your flow to download the sample zip file present on the sFTP server.

1. Click **Invokes** ◎ in the pane next to the canvas.
2. Expand **FTP** and drag **FTP Connection** to the **plus** sign in your integration flow.
   The Oracle Adapter Endpoint Configuration Wizard is displayed.
3. On the Basic Info page, enter a name without a space (`Download_Zip`) and a description for the adapter. Click **Next**.
4. On the Operations page, enter the following details.

Element	Information to Enter
Select Operation	Select **Download File**.
Select a Transfer Mode	Select **Binary**.
Input Directory	Specify the path of the directory on your FTP server that contains the file to be read. For example, `/home/user2/FTP Bulk Transfer`.
File Name	Enter `test-data.zip`.
Download Directory	Specify the directory to use in Oracle Integration to stage the downloaded file temporarily; for example, `/tmp/zip-test/staged`.

Element	Information to Enter
Perform unzip on compressed file downloaded from external FTP server	Select this field to unzip the sample file.

5. Click **Next**. On the Summary page, review the data you've entered and click **Done**.



Summary
Download_zip

Description
Downloads files

Operation
Download File
Transfer Mode
Binary
Input Directory
/home/user3/FTP Bulk Transfer
File Name
test-data.zip
Download Directory
/tmp/zip-test/staged
Unzip the file
Yes
Decrypt the File
No
Verify the Signature
No

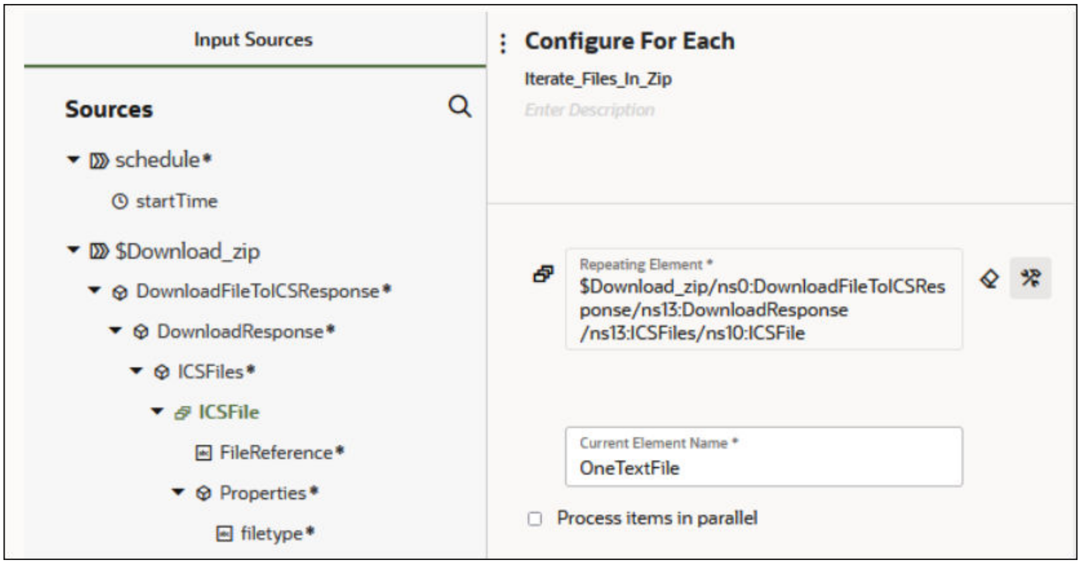
6. Note that a map action corresponding to the **Download_Zip** action is automatically added to the canvas. However, it doesn't require any configuration changes.
7. In the toolbar, click **Layout** and select **Horizontal** from the list. Click **Save.**

Add a For-Each Action to Iterate Over Individual Files

The for-each action enables you to loop over individual files within the zip file and stage each file in Oracle Integration for transformation.

1. On the canvas, click **Actions** ϟ and drag and drop the **For Each** action (present under **Collection**) after the **Download_Zip** action.
   The Create Action dialog is displayed.
2. Enter a name for the action, `Iterate_Files_In_Zip`.
3. Provide an optional description.
4. In the **Repeating Element** field, specify the element over which to iterate.

For this example, the element over which to loop is **ICSFile**, which represents the individual files within the zip file downloaded in the previous action. Expand the **Source** tree to select this element. It is present under `$Download_Zip > DownloadFileToICSResponse > DownloadResponse > ICSFiles > Load more`. After you've selected the element, drag to the **Repeating Element** field to populate the field with this element.



Iterate_Files_In_Zip

Repeating Element *
$Download_zip/ns0:DownloadFileToICSResponse/ns13:DownloadResponse/ns13:ICSFiles/ns10:ICSFile

Current Element Name *
OneTextFile

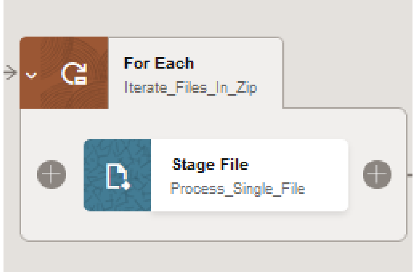
□ Process items in parallel

5. In the **Current Element Name** field, enter an alias for the current file of the iteration; for example, `OneTextFile`. You'll use this alias for processing individual files downstream.
6. Click **Create**.

Configure a Stage File Action to Read Individual Files

Add a stage file action to read the contents of the individual files within the zip file. For the files to be successfully read, you must also specify the schema file (by loading a comma-separated value (CSV) or `.txt` file) that contains the required data structure.

1. On the canvas, click **Actions** and drag and drop the **Stage File** action (present under **Actions**) inside the **Iterate_Files_In_Zip** action. Placing the stage file action within the loop ensures that each file is read iteratively.



The Configure Stage File Action dialog is displayed.

2. On the Basic Info page, enter a name for the action (`Process_Single_File`). Click **Next**.
3. On the Configure Operation page, enter the following details.

Field	Information to Enter
Choose Stage File Operation	Select **Read Entire File**.
Configure File Reference	Select **No** to process files using names.
Specify the File Name	Click **Switch to Developer View** to build an expression to specify the name of the file to be read. We'll use the alias you specified for the current file under iteration in the previous action (that is, OneTextFile) and extract the actual file name from it. In the **Source** tree, search for **OneTextFile**. Select the `filename` field present under `$OneTextFile > ICSFile > Properties`, and drag it to the field to populate the value for the **Specify the File Name** field on the right.
Specify the Directory to read from	Click **Switch to Developer View** to build an expression to specify the directory from which to read files. Select the `directory` field present under `$OneTextFile > ICSFile > Properties`, and drag to populate the value for the **Specify the Directory to read from** field on the right.

Click **Next**.

4. Leave the configurations on the Schema Options page as they are. Click **Next**.
5. On the Format Definition page, perform the following actions to upload a file from which to create a schema.

Field	Information to Enter
Select the Delimited Data File	Drag and drop a file to the **Drag and Drop** area and upload the `test-file3.txt` file you created. See Create a Sample Zip File. The content of the file is then displayed at the bottom of the page.
Enter the Record Name	Enter the name you want to assign to the records in the schema file. This becomes the parent element in the created schema file for the record names selected as column headers from the CSV file. Enter `Row`.
Enter the Recordset Name	Enter the name you want to assign to the recordset in the schema file. This becomes the root element of the created schema file. Enter `RowSet`.
Use the First Row as Column Headers	Deselect this check box.

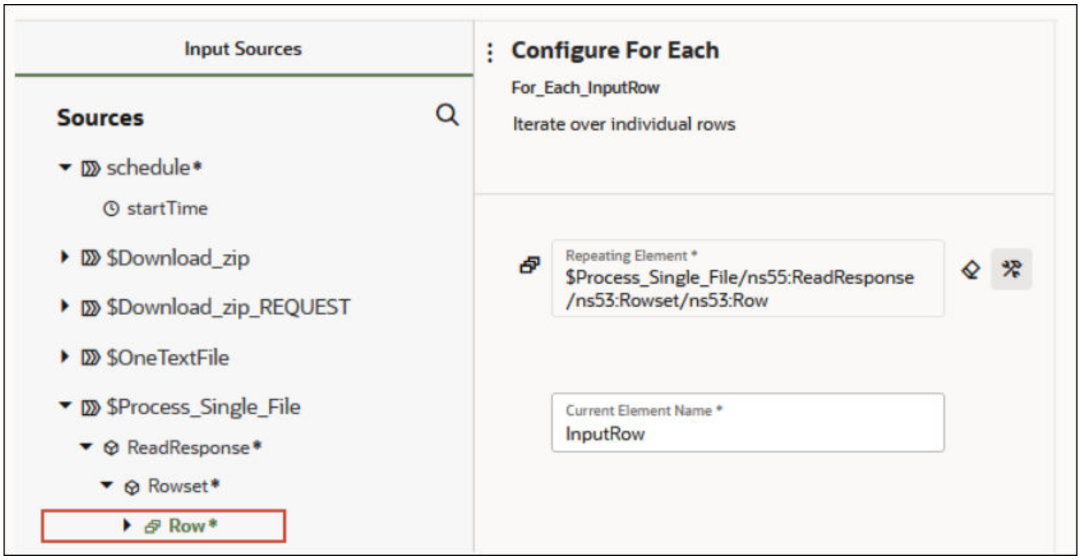
The rest of the fields are automatically populated from the uploaded CSV file. Click **Next**.

6. On the Summary page, review the data you've entered and click **Done**.

Add a For-Each Action to Iterate Over Individual Records

When an individual file is staged in Oracle Integration, add a second for-each action to iterate through each record in the file.

1. On the canvas, click **Actions** and drag and drop the **For Each** action after the **Process_Single_File** action but within the first for-each loop.
   The Create Action dialog is displayed.
2. Enter a name for the action, `For_Each_InputRow`.
3. Provide an optional description.
4. In the **Repeating Element** field, specify the element over which to iterate.
   Here, the element over which to loop is **Row**, which is the name you've provided for individual rows/records within a file in the previous action. Expand the **Source** tree to find this element. It's present under `$Process_Single_File > ReadResponse > RowSet`. After you've selected the element, drag to the **Repeating Element** field to populate the field with the element.



Input Sources

Sources
- schedule*
  - startTime
- $Download_zip
- $Download_zip_REQUEST
- $OneTextFile
- $Process_Single_File
  - ReadResponse*
    - Rowset*
      - Row*

Configure For Each

For_Each_InputRow

Iterate over individual rows

Repeating Element *
$Process_Single_File/ns55:ReadResponse/ns53:Rowset/ns53:Row

Current Element Name *
InputRow

5. In the **Current Element Name** field, enter an alias for the current record of the iteration, for example, `InputRow`. You'll use this alias for processing individual records downstream.
6. Click **Save**.

Add a Stage File Action to Write Modified Records into a New File

Add a second stage file action to read individual records of each file, modify or transform them, and write the modified records into a new file.

1. On the canvas, click **Actions** and drag and drop the **Stage File** action inside the **For_Each_InputRow** action. Placing this stage file action within the second loop ensures that each record in a file under iteration is read.
   The Configure Stage File Action dialog is displayed.

2. On the Basic Info page, enter a name for the action (`Write_Row`). Click **Next**.
3. On the Configure Operation page, enter the following details.

Field	Information to Enter
Choose Stage File Operation	Select **Write File**.
Specify the File Name	Click **Switch to Developer View** to build an expression to specify the name for the new file that is created. Let's build an expression such that the names of the new individual files are: their old names plus the suffix, `- modified`. In the **Source** tree, select the `filename` field present under `$OneTextFile > ICSFile > Properties` (this represents the old names of the files), and drag to populate the value of the **Specify the File Name** field on the right. Add a comma after `filename` and enter `'-modified'`. Enclose the entire expression in parenthesis and prefix the term `concat` as follows: `concat($OneTextFile/nsmpr2:ICSFile/nsmpr2:Properties/nsmpr2:filename, '-modified')`
Specify the Output Directory	Click **Switch to Developer View** to build an expression to specify the output directory, where the files will be written. Type the following expression into the value of the **Specify the Output Directory** field: `'/tmp/zip-test/staged/modified'`.
Append to Existing File	Select to append records to the existing file.

Click **Next**.

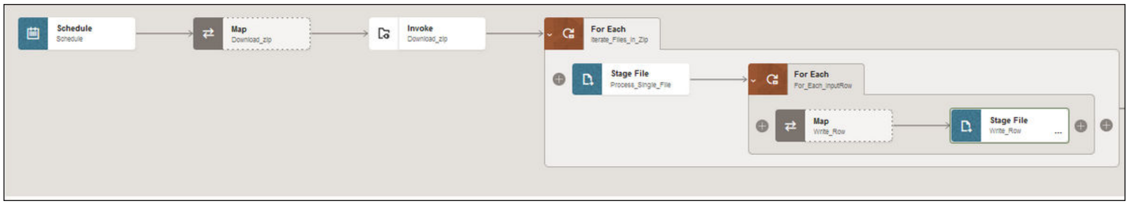
4. Leave the configurations on the Schema Options page as they are. Click **Next**.
5. On the Format Definition page, perform the following actions to upload a file from which to create a schema.

Field	Information to Enter
Select the Delimited Data File	Click **Choose File** and upload the `test-file1.txt` file you created. See Create a Sample Zip File. The content of the file is then displayed at the bottom of the page.
Enter the Record Name	Enter the name you want to assign to the records in the schema file. This becomes the parent element in the created schema file for the record names selected as column headers from the CSV file. Enter `ModifiedRow`.
Enter the Recordset Name	Enter the name you want to assign to the recordset in the schema file. This becomes the root element of the created schema file. Enter `ModifiedRowSet`.
Use the First Row as Column Headers	Deselect this check box.

The rest of the fields are automatically populated from the uploaded CSV file. Click **Next**.

6. On the Summary page, review the data you've entered and click **Done**.

At this point, the integration flow appears as follows:

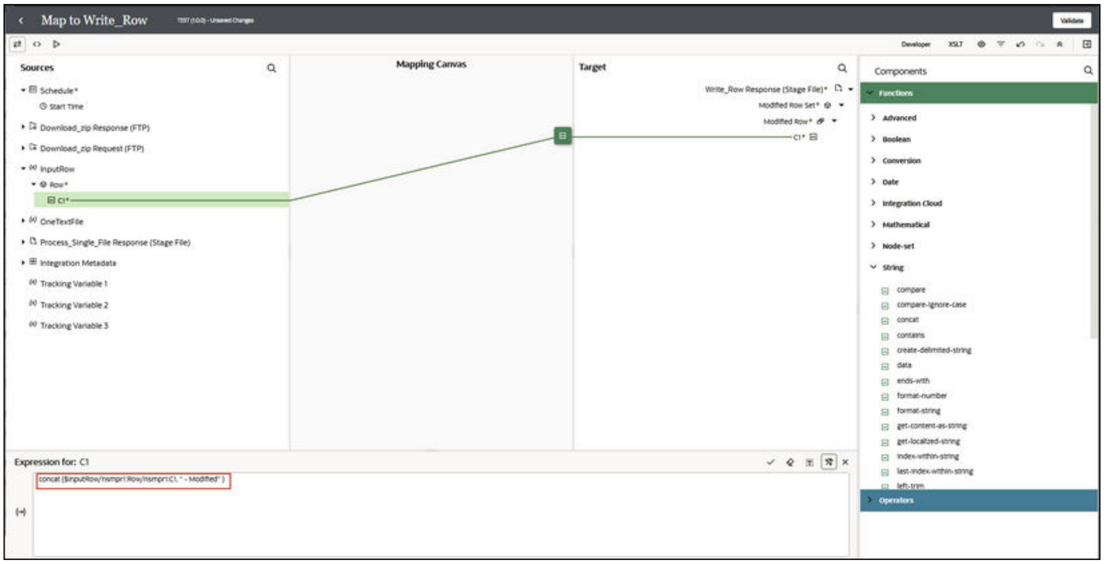


Configure Data Mapping to Transform Data in Files

Configure the data mapping in the automatically added mapper to modify the content of the files. You can use functions and operators to enrich or transform each record before it's written into a new file.

In this topic, you'll configure a simple mapping that suffixes the term `- Modified` to each row or record of a file as a way of transforming the file contents.

1. Click **Map to Write_Row** and select **Edit** ✎ to open the mapper.
2. Click **XSLT** on the toolbar to enter an advanced view.
3. Expand **$InputRow** (which corresponds to an old row or record under iteration) on the left and map the **C1** element within it to the **C1** element under **ModifiedRowSet** (which corresponds to a new record in a new file) on the right.
   The expression builder with a corresponding mapping expression opens.
4. Click **Toggle Functions** in the toolbar to view the Components pane. Click **Switch to Developer View** . Expand **String** and drag the **concat** function to the expression builder pane.
5. Copy the existing expression in the pane (for example, `$InputRow/nsmpr6:Row/nsmpr6:C1`) into the concat function as the first operator and add `" - Modified"` as the second operator. The final expression in the pane will be of the following format: `concat ( $InputRow/nsmpr6:Row/nsmpr6:C1 , " - Modified" )`.
   This ensures the term `- Modified` is appended to each old record before it's written to a new file.
6. Click **Save** ✓ and click **Validate**.
7. Click **Go back** ‹ to return to the integration canvas.



```
concat ($inputRow/nsmpr1:Row/nsmpr1:C1, " - Modified" )
```

Add a Stage File Action to Zip the Modified Files

Let's add a stage file action to combine all the individual files, which have been transformed, into a new zip file that will be sent back to your FTP server.

1. On the canvas, click **Actions** and drag and drop the **Stage File** action outside both the for-each loops.
   The Configure Stage File Action dialog is displayed.
2. On the Basic Info page, enter a name for the action (`Create_Zip`). Click **Next**.
3. On the Configure Operation page, enter the following details.

Field	Information to Enter
Choose Stage File Operation	Select **Zip Files**.
Specify the File Name	Type the following expression into the **Specify File Name** field: `'test-data-modified.zip'`.
Specify the Directory to zip	Type the following expression into the **Specify the Directory to zip** field: `'/tmp/zip-test/staged/modified'`.
Specify the Output Directory	Type the following expression into the **Specify the Output Directory** field: `'/tmp/zip-test/staged/modified'`.

   Click **Next**.
4. On the Summary page, review the data you've entered and click **Done**.
5. Click **Save**.

Configure an FTP Adapter to Write the Modified Zip File

Add another FTP Adapter at the end of the integration to write the transformed file back into your sFTP server, from where other applications can pick the file up for further processing.

1. Click **Invokes** ◎ in the pane next to the canvas.
2. Expand **FTP**, drag and drop **FTP Connection** after the **Create_Zip** action.
   The Oracle Adapter Endpoint Configuration Wizard is displayed.
3. On the Basic Info page, enter a name without a space (`Send_Zip`) and a description for the adapter. Click **Next**.
4. On the Operations page, enter the following details.

Field	Information to Enter
Select Operation	Select **Write File**.
Select a Transfer Mode	Select **Binary**.
Output Directory	Specify the path of the directory on your FTP server where the new file needs to be written. For example, `/home/user2/FTP Bulk Transfer`.
File Name Pattern	Enter `test-data-modified.zip`.
Append to Existing File	Leave it unchecked.

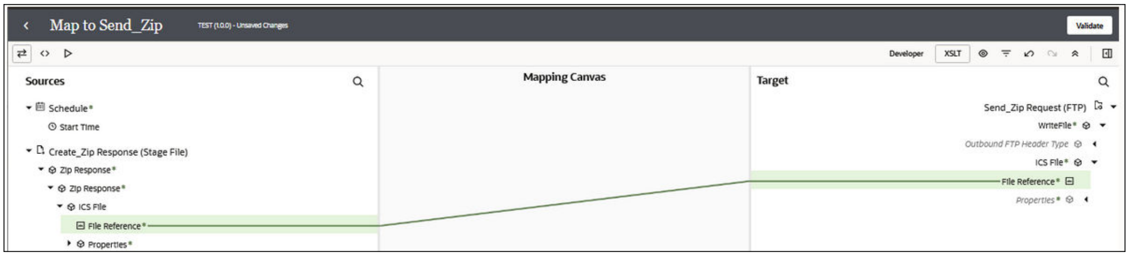
   Leave the selections for other fields unchanged, and click **Next**.

5. On the Schema page, select **No** in the **Do you want to specify the structure for the contents of the file?** field. Click **Next**.
6. On the Summary page, review the data you've entered and click **Done**.

Configure Data Mapping for the FTP Adapter

Note that a map action corresponding to the **Send_Zip** action is automatically added to the integration flow. Edit this action to map the file reference of the newly-created zip file to the file reference field of the FTP Adapter.

1. Click the **Map to Send_Zip** element and select **Edit** ✎.
2. In the mapper, click **XSLT** on the toolbar to enter an advanced view.
3. Expand **$Create_Zip Response** on the left and map the **FileReference** element within it to the **FileReference** element on the right as shown in the following image.
4. Click **Validate**, and **Go back** ‹ .



Activate and Test Your Integration

Check for errors, save, and activate the integration flow.

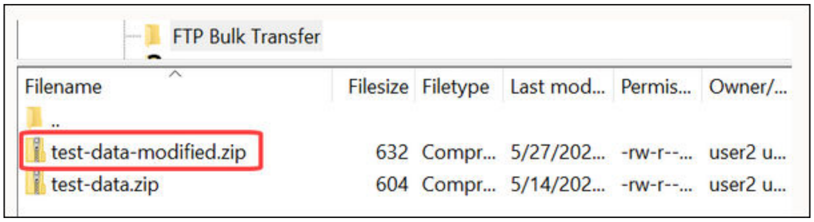
You'll notice an error notification on the canvas. To resolve it, assign a primary business identifier for tracking. Business identifiers enable you to track payload fields in messages during runtime. A primary business identifier is required to activate an integration. See Assign Business Identifiers for Tracking Fields in Messages.

To assign an identifier:

1. Click **Business Identifiers** (I) in the top-right corner of the canvas.
2. In the resulting dialog, select **startTime** on the left and move it to the table on the right.
3. Click **Save**.
4. Click **Business Identifiers** (I) to close the dialog.

Now, activate and test your integration.

1. On the Integrations page, click **Activate** ⏻ to activate your integration.
2. Click **Activate** in the Activate Integration dialog.
3. After the integration is activated, click **Run** to test run the integration.
4. Select **Ad hoc request** in the resulting dialog and click **Run**.
   You've now successfully submitted the integration for a test run. To create a sample schedule for your integration, see Create a Sample Schedule.
5. Log in to your sFTP server and check for the modified files returned by the integration.



FTP Bulk Transfer

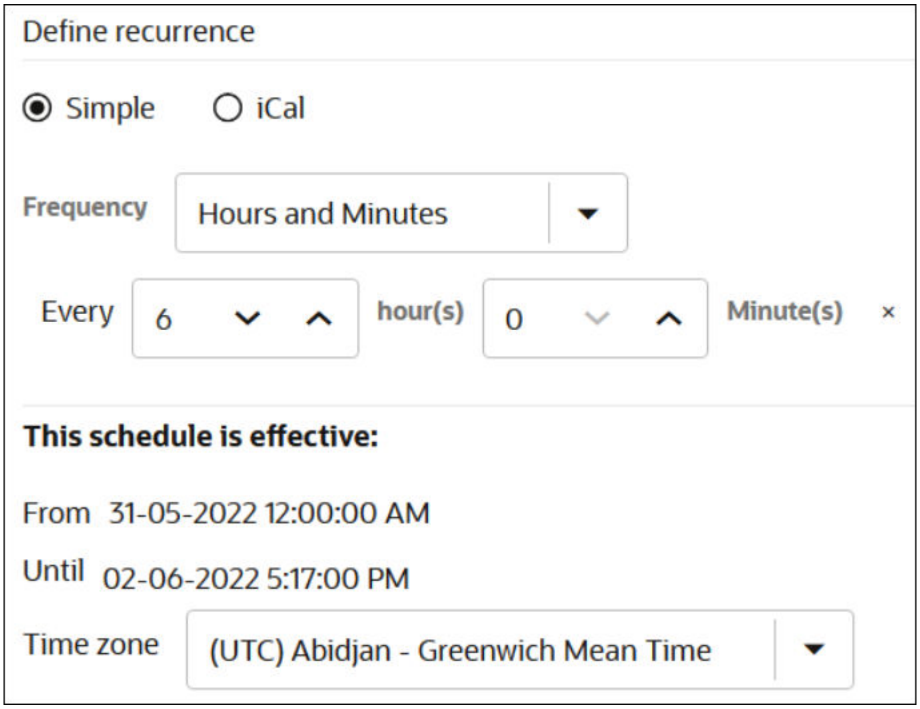
Filename	Filesize	Filetype	Last mod...	Permis...	Owner/...
..					
test-data-modified.zip	632	Compr...	5/27/202...	-rw-r--...	user2 u...
test-data.zip	604	Compr...	5/14/202...	-rw-r--...	user2 u...

Create a Sample Schedule

Create a schedule for your integration to pick up files from the FTP server at a date, time, and frequency of your choosing.

1. On the Integrations page, hover your cursor over your integration, then click **Actions** ••• then **Add Schedule**.
   The page for defining the schedule execution details is displayed.
2. For this example, let's create a basic schedule, so leave the **Simple** radio button selected under the **Define recurrence** section.
3. In the **Frequency** field, select **Hours and Minutes**, and change the hours field to **6**.
   You've now created a schedule that runs the integration every six hours. Similarly, you can create a schedule that runs on particular days or weeks.

4. Additionally, you can specify the start and end dates for this schedule. In the **This schedule is effective:** section, click the **From** field and select **Modify start date** to specify a date and time from when this schedule should run. Similarly, specify an expiry date and time in the **Until** field. In addition, you can also specify your preferred timezone.



Define recurrence

● Simple ○ iCal

Frequency: Hours and Minutes

Every 6 hour(s) 0 Minute(s)

This schedule is effective:

From 31-05-2022 12:00:00 AM

Until 02-06-2022 5:17:00 PM

Time zone: (UTC) Abidjan - Greenwich Mean Time

5. Click **Save** and then click **Back** ‹ to return to the Integrations page. To create advanced schedules, see
See Define the Integration Schedule in *Using Integrations in Oracle Integration 3*.

.