Oracle Utilities Opower Data Transfer Standards Documentation

Opower Data Transfer Standards Documentation

F14406-45
December 2025

Oracle Utilities Opower Data Transfer Standards Documentation Opower Data Transfer Standards Documentation,

F14406-45

Contents

1 Oracle Utilities Opower Data Transfer Standards Documentation

2 Getting Started with Data Transfer

Data Transfer Process Overview 1
- Phase 1: Sample Historical File Integration 1
- Phase 2: Full Historical File Integration 1
- Phase 3: Incremental File Integration 2

Data Transfer Standards by Product 2
- APIs 2
- Behavioral Demand Response 3
- Business Customer Engagement Digital Self-Service Web Portal 3
- Customer Service Interface 4
- Customer Service Interface - Bill Advisor 4
- Digital Self Service - Energy Management 5
- Distributed Energy Resources (Solar) 5
- Email Home Energy Reports 6
- Energy Efficiency Web Portal 6
- High Bill Alert AMI 7
- High Bill Alerts (non-AMI) 7
- Home Energy Reports 8
- Inside Opower 8
- Load Shifting 9
- Peak Time Rebates 9
- Rate Education Reports 10
- Weekly Energy Updates 10

Documentation Conventions and Common Data Formats 10
- Example and Allowed Values 10
- Empty Fields and Null Values 11
- Date and Time Handling 11
- Supported Boolean Formats 11
- Supported Decimal Formats 12

Utility Data Obligations 12

Contact Your Delivery Team 12

3 Setting Up Secure File Transfer

File Transfer Methods 1
Microsoft SharePoint 1
Secure File Transfer Protocol (SFTP) 1
Encryption 2
File Errors 2
File Retention Policy 2
File Transfer Setup - Windows 2
Downloading and Installing PuTTY 2
Generating SSH Keys with PuTTY 5
Transferring Files with WinSCP 7
File Transfer Setup - Unix or Linux 9
Generating SSH Keys with ssh-keygen 9
Transferring Files with SFTP 10
File Retention Policy 10
File Transfer FAQs 11
Fileset Transfer Manifest Specification 12
Transfer Manifest File Contents 12
File Indentation Structure 12
File Naming Conventions 13
File Transfer Limits 13

4 Core Data Standards

Account Data Transfer 1
Account Data Integration Process 2
Account Data File Specifications 2
Main Account File 2
Person Address File 12
Service Agreement Service Points File 13
Account Data Order and File Format 15
Data File Order 15
File Format 15
Account Data File Name Conventions 16
File Naming Example 17
Additional Notes 17
Account Data Version History 17
Billing Data Transfer 19
Billing Data Integration Process 20

Billing Data File Specifications 20
Bill Fields 20
Bill Segment Fields 22
Line Item Fields 25
Service Quantity Fields 27
Billing Data Order and File Format 29
Data File Order 29
File Format 29
Billing Data File Name Conventions 30
File Naming Examples 31
Additional Notes 31
Billing Data Version History 31
Premise Data Transfer 33
Premise Data Integration Process 34
Premise Data File Specifications 34
Premise Fields 34
Service Point Fields 37
Device Installation Fields 40
Device Fields 42
Measuring Component Fields 44
Premise Data Order and File Format 49
Data File Order 49
File Format 50
Premise Data File Name Conventions 50
File Naming Example 51
Additional Notes 51
Premise Data Version History 51
Legacy Billing Data Transfer 53
Legacy Billing Data Integration Process 54
Sample Historical File Integration 54
Full Historical File Integration 54
Incremental Data Integration 55
Legacy Billing Data Entity Definitions 56
Data Entity Relationships 56
Customer 57
Premise 57
Service Point 58
Account 59
Usage Value 60
Legacy Billing Data File Specifications 61
Customer Fields 61
Account Fields 67

Service Point Fields 68
Usage Fields 75
Legacy Billing Data File Order and File Format 77
Data File Order 77
File Format 78
Legacy Billing Data File Name Conventions 78
File Name Examples 79
Additional Notes 79
Additional Business Rules for Legacy Billing Data 80
Usage Value Corrections 80
Multiple Service Points for a Single Fuel Type 83
Landlords 84
Customer Transfers to a New Site 84
Legacy Billing Data Version History 84

5 Supplemental Data Standards

Call Center Data Transfer 1
Call Center Data File Specifications 1
File Format 3
File Name Conventions 3
Call Center Data Version History 4
Customer Classification Data Transfer 6
Customer Classification Data File Specifications 6
Customer Classification Definition File 6
Customer Classification File 8
Customer Classification Data Order and File Format 9
Data Order 9
File Format 10
Customer Classification File Name Conventions 10
File Name Examples 11
Additional Notes 11
Customer Classification Data Verification Checklist 11
Customer Classification Data Version History 12
Entity Attribute Data Transfer 13
Entity Attribute Data File Specifications 14
Entity Attribute Definition File 14
Entity Attributes Data File 17
Entity Attribute Data Order and File Format 20
Data Order 20
File Format 20
Entity Attribute File Name Conventions 20

File Name Examples 21
Additional Notes 21
Entity Attribute Data Version History 22
Interval Data Transfer 22
Interval Data Integration Process 22
Interval Data Definitions 23
Logical Diagram of Data Entity Relationships 23
Service Point 24
Usage Value 24
Interval Data File Specifications 25
Field Descriptions 25
Time Field Examples 29
Daylight Saving Time 30
File Format 31
File Transfer Limit 32
Interval Data File Name Conventions 32
File Name Examples 33
Additional Notes 33
Interval Data Version History 33
Program Participation Data Transfer 36
Program Participation Data File Specifications 36
Program Participation Definition File 36
Program Participation File 38
Program Participation Data File Format 40
Program Participation Data File Name Conventions 41
File Naming Example 41
Additional Notes 42
Program Participation Data Verification Checklist 42
Program Participation Data Version History 43
Rates Data Transfer 44
Rates Data Integration Process 44
Supported Rate Components 45
Rates Data File Specifications Overview 45
Overview of Files 45
Example Scenarios 46
Rate Attributes File 52
Examples of Rate Attribute Keys 53
Rate Attributes Sample Data 54
Baselines File 55
Baselines Sample Data 56
Rate Prices File 57
Rate Prices Sample Data 58

Rate Plan Configuration Files 59
Rate Tier Definitions File 60
Rate Period Definitions File 61
Holidays and Seasons File 63
Peak Time Rebates File 66
Guidelines for Common Data Elements 67
Account and Rate Plan Identifiers 67
Start and End Dates 68
Rates Data File Format 68
Rates Data File Name Conventions 69
Example Names 70
Additional Notes 70
Rates Data Version History 70
Rate Engine Plus Data Transfer 72
Rate Engine Plus Data Integration Process 73
Rate Engine Plus Data File Specifications 73
Rate Engine Plus Rate Eligibility 74
Rate Engine Plus Data File Format 74
Rate Engine Plus Data File Name Conventions 74
Example File Names 75
Additional Notes 75
Rate Engine Plus Data Version History 75
Unsubscribe Data Transfer 76
Unsubscribe Data Integration Process 77
Unsubscribe Data File Specifications 77
Email Unsubscribe Fields 77
Interactive Voice Response (IVR) Unsubscribe Fields 78
SMS Unsubscribe Fields 79
Unsubscribe Data Order and File Format 79
Data Order 79
File Format 80
Unsubscribe Data File Name Conventions 80
File Name Examples 81
Additional Notes 81
Unsubscribe Data Version History 81

1

Oracle Utilities Opower Data Transfer Standards Documentation

Welcome to the Oracle Utilities Opower data transfer standards documentation. Use this information to learn how to transfer your data to Oracle Utilities during the setup and launch of your program. Have a question? Contact your Delivery Team or visit My Oracle Support.

Quick Links

- Getting Started with Data Transfer
- Setting Up Secure File Transfer
- Core Data Transfer Standards
  - Account Data Transfer
  - Billing Data Transfer
  - Premise Data Transfer
  - Legacy Billing Data Transfer
- Supplemental Data Transfer Standards
  - Call Center Data Transfer
  - Customer Classification Data Transfer
  - Entity Attribute Data Transfer
  - Interval Data Transfer
  - Program Participation Data Transfer
  - Rates Data Transfer
  - Rate Engine Plus Data Transfer
  - Unsubscribe Data Transfer

2

Getting Started with Data Transfer

Oracle Utilities Opower provides a range of data-driven solutions, software products, and personalized insights to utility customers about their energy use. In order for these solutions and products to work, utilities must set up regular data transfer feeds and send data files to Oracle Utilities that conform to predefined specifications. The data transfer requirements will vary depending on which products and solutions will be implemented.

- Data Transfer Process Overview
- Data Transfer Standards by Product
- Documentation Conventions and Common Data Formats
- Utility Data Obligations
- Contact Your Delivery Team

Data Transfer Process Overview

Transferring your data to the Oracle Utilities Opower platform generally involves the following phases:

- Sample historical file integration
- Full historical file integration
- Incremental file integration

In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

Phase 1: Sample Historical File Integration

Your Delivery Team will work with your technical lead and data extract engineer to review our data specification and map your data elements to the fields in the Oracle Utilities Opower data file format. Your engineering team then generates a sample historical data file.

The relatively small size of the sample file allows your Delivery Team to quickly analyze and diagnose issues. Your Delivery Team will provide you with a list of any issues found in the sample file and ask that you provide an updated file. This process is repeated until a sample historical file passes diagnostic testing. It typically takes several rounds of sample file generation, analysis, and feedback to approve a sample historical file.

Phase 2: Full Historical File Integration

After the sample file has passed validation, we can begin the process of generating, validating, and approving a full historical file. This file needs to include actual production data for all active and inactive customer accounts for a specific timeframe (typically January of the previous year). Full historical file approval typically requires several rounds of generation and validation. The approved historical file will be loaded into the Oracle Utilities Opower production

application, and the data within it will be displayed to customers in reports, email messages, alerts, and web applications.

Phase 3: Incremental File Integration

After your Delivery Team approves and loads the full historical data, the team will generate an incremental file that includes any new customer information and data recorded since the most recent record in the full historical file that was delivered. After the first set of incremental files are approved, your team will schedule an automated file transfer. During this transfer, your system will push ongoing incremental files to the Oracle Utilities Opower SFTP server.

> **Note**
>
> The number of phases you need to follow may vary slightly depending on which Oracle Utilities Opower products you are implementing and which data transfer standards are required for those products. See Data Transfer Standards by Product for an overview of which products require which data standards. Also, note that the type of data that you must extract from your system can vary, as well as the frequency with which you must transfer it. In addition to the overview here, you should read the data integration process documentation for each individual data transfer standard.

Data Transfer Standards by Product

The data you need to send to Oracle Utilities Opower depends on which products and features you are implementing. Use the information below to determine which data transfer standards are applicable to your situation.

> **Note**
>
> Some Oracle Utilities Opower products and features (such as Customer Education Reports) do not require data transfer and are therefore omitted from this list. Contact your Delivery Team if you have any questions.

APIs

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for the bill comparison API and neighbor comparison API. Your Delivery Team will work with you to determine which data standards to use.

Data Standards Required	Notes
Interval Data Transfer	Required for the following APIs and use cases: • Bill forecast API • To calculate customer-specific heating and cooling coefficients for the bill comparison API • To calculate personalized breakdowns for the seasonal disaggregation API
Rates Data Transfer	Required for the following use cases: • To display information about customer rate plan changes in the bill comparison API • To display cost information in the bill forecast API

Behavioral Demand Response

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required to provide the following features: • Neighbor Comparison email module • Neighbor Rank email module • Post Season Report • Post-Event Voice Message • Pre-Event Voice Message

Business Customer Engagement Digital Self-Service Web Portal

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.

Data Standards Required	Notes
Interval Data Transfer	Required for the following features: • Bill or Usage Forecast widget • Data Browser - by day views • Demand Heatmap widget • Download My Data - Green Button - for daily or subdaily data
Premise Data Transfer	Required for the following features: • Display of weather data in the Data Browser • Display of commercial and industrial data in the Data Browser
Rates Data Transfer	Required for the following features: • Bill Comparison widget (for rates-based insights) • Bill Forecast widget (for rates-based insights) • Data Browser widget (for cost insights in certain parts of the Energy Costs View)

Customer Service Interface

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Call Center Data Transfer	Required for customer call measurement.
Interval Data Transfer	Required for interval data features.

Customer Service Interface - Bill Advisor

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.

Data Standards Required	Notes
Call Center Data Transfer	Required for customer call measurement.
Interval Data Transfer	Required for interval data features.

Digital Self Service - Energy Management

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for the following features: • Bill or Usage Forecast widget • Neighbors Comparison widget - by bill and by day views
Premise Data Transfer	Required for the following features: • Display of water data in the Data Browser • Display of commercial and industrial data in the Data Browser
Rates Data Transfer	Required for the following features: • Bill Forecast widget • Compare My Bills widget

Distributed Energy Resources (Solar)

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Rates Data Transfer	Required to display cost insights in the Distributed Energy Resources web experience.
Rate Attributes File	Required to identify customers who generate energy using solar power.

Email Home Energy Reports

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Customer Classification Data	Required to send targeted communications to customer groups.
Interval Data Transfer	Required for modules which contain interval data.
Rates Data Transfer	Required to display cost information.

Energy Efficiency Web Portal

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for the following features: • Bill or Usage Forecast • Heating and Cooling comparison • Costs View by bill and by day views • Demand View • Neighbors View by bill and by day views • Usage View by bill and by day views • Weather View by bill and by day views • What Uses Most heating and cooling breakdowns • My Rates • Additional options for exporting and downloading data, such as range of days or bill periods

Data Standards Required	Notes
Rates Data Transfer	Required for the following features: • Bill Forecast • Costs View by bill and by day views • Compare my Bills change in rate plan factor • Usage View displays energy use as price tiers • My Rates • Additional options for exporting and downloading data, such as range of days or bill periods
Customer Classification Data Transfer	Required for targeting communications for customer groups.

High Bill Alert AMI

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Call Center Data Transfer	Required for customer call measurement.
Interval Data Transfer	Required for base product features.
Rates Data Transfer	Required to display cost information.

High Bill Alerts (non-AMI)

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Call Center Data Transfer	Required for customer call measurement.
Interval Data Transfer	Required for base product features.

Home Energy Reports

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for modules which contain interval data.
Customer Classification Data Transfer	Required to send targeted communications to customer groups.
Rates Data Transfer	Required to display cost information.

Inside Opower

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Call Center Data Transfer	Required for call center insights in reports and the Data Exploration dashboard.
Customer Classification Data Transfer	Required for customer classification insights in reports and the Data Exploration dashboard.
Interval Data Transfer	Required for interval data insights in reports and the Data Exploration dashboard.
Rates Data Transfer	Required for cost-related insights in reports and the Data Exploration dashboard.
Program Participation Data Transfer	Required for program participation insights in reports and the Data Exploration dashboard.

Load Shifting

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Call Center Data Transfer	Required for customer call measurement.
Interval Data Transfer	Required for base product features.
Rates Data Transfer	Required to display cost information.

Peak Time Rebates

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for base product features.

Rate Education Reports

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for interval data features.
Rates Data Transfer	Required for base product features.

Weekly Energy Updates

Data Standards Required	Notes
Option 1 • Account Data Transfer • Billing Data Transfer • Premise Data Transfer *Option 2* • Legacy Billing Data Transfer	Required for base product features. Your Delivery Team will work with you to determine which data standards to use.
Interval Data Transfer	Required for base product features.
Rates Data Transfer	Required to display cost information.

Documentation Conventions and Common Data Formats

The data standards documentation uses conventions to describe data fields in a consistent manner. Additionally, the data fields in the Oracle Utilities Opower data platform rely on common data formats, such as date, time, and Boolean values. These formats should be strictly followed so as to minimize errors in the data transfer process. Use the sections below to understand the conventions you will encounter and the data formats that Oracle Utilities expects.

Example and Allowed Values

Throughout the documentation you will see field descriptions that include **Example Values** or **Allowed Values** labels. The **Example Values** are meant to give you an idea or recommendation of the data you can send. However, you are not required to follow the examples. You can send other values instead.

The **Allowed Values** label denotes a set of values that is exhaustive and must be strictly followed. You must choose from the list provided, and cannot use other values.

Empty Fields and Null Values

Each field description in the documentation includes an indication of whether the field can be empty when you send data to Oracle Utilities. This is shown in the **Can Be Empty?** label.

If the label is followed by **Yes**, the field can be left blank or a value of NULL can be used.

If the label is followed by **No**, then a value must be provided or the data import will fail. A value of NULL is not acceptable in such cases unless the field has a default value that the system can revert to.

Date and Time Handling

All fields containing date or date time values must fully specify an exact point in time rather than a local date or date and time. The following table shows how Oracle Utilities Opower will use default time and offset values if these are not provided:

Format	Description
`YYYY-MM-DD<T>hh:mm:ss-<timezone>`	The exact value specified will be loaded after adjusting the date time offset to UTC. Note that the offset itself will *not* be preserved. Time zones should be specified according to the **TZ database name** column in List of tz database time zones (Wikipedia). **Example**: `2007-06-05T12:30:00-America/New_York` Oracle Utilities Opower uses the time zone to calculate the time in UTC, resulting in the value `2007-06-05T16:30Z` being stored in the system (where Z stands for Zulu or UTC time).

Supported Boolean Formats

Many fields in the Oracle Utilities Opower data platform require Boolean data values. The following Boolean formats are supported. Other Boolean values or formats may be supported for specific fields, in which case this is documented in the field description.

Boolean	Supported Formats
TRUE	• YES • Y • TRUE • T • 1
FALSE	• NO • N • FALSE • F • 0

Supported Decimal Formats

For a DECIMAL column, the declaration syntax is DECIMAL(M,D). M refers to the maximum number of digits (the precision). It has a range of 1 to 65. D refers to the number of digits to the right of the decimal point (the scale). It has a range of 0 to 30 and must be no larger than M.

All decimal numbers must use a period (.) for the decimal separator. No thousands separator is allowed. For example, decimal numbers must be formatted like `100000000.00`. Values like the following are not allowed:

- `100000000,00`
- `100,000,000.00`
- `100.000.000,00`
- `100_000_000.00`

Utility Data Obligations

The utility must agree to certain data obligations to ensure that data quality and integrity are maintained during the launch and operations of the Oracle Utilities Opower platform. The statement below describes these obligations. (The “You” in the statement refers to the utility.)

For any initial data load, data migration, or iterative data file, You must provide Oracle Utilities with data, in formats requested by Oracle Utilities, and You are responsible for the correct formatting of any data provided to Oracle Utilities as part of any initial or ongoing data load.

You must appoint a point of contact within Your organization authorized to work with Oracle Utilities to resolve incidents related to any data loading process.

If data quality issues are identified with any data received in this Oracle Utilities Cloud Service, then You are responsible for providing corrected, validated, formatted and cleansed data. Oracle Utilities is not responsible for correcting, validating, formatting or cleansing data from Your source application.

As reasonably required by Oracle Utilities, You will provide Oracle Utilities, on a timely basis, with complete and accurate information and responses to questions as needed to support the performance of the Oracle Utilities Cloud Services.

Oracle Utilities will remove data stored on the SFTP server periodically based on the data retention periods applicable to the Oracle Utilities Cloud Services. You are responsible for ensuring that any data You require is downloaded to a storage location local to Your organization.

Contact Your Delivery Team

Your Oracle Delivery Team is the group responsible for setting up, configuring, launching, or expanding your Oracle Utilities Opower program. Contact your Delivery Team if you have any questions about your program products and implementation.

To contact your Delivery Team:

1. Sign in to Inside Opower (https://inside.opower.com). This is your portal for questions and information related to your program.
2. Go to the Community tab to see who is on your Delivery Team.
3. Contact any of the team members using the information provided.

If you need to report an issue or get technical support, contact My Oracle Support.

3

Setting Up Secure File Transfer

To transfer data files to Oracle Utilities Opower, you must set up and configure a Secure Shell (SSH) tool and private/public key pair for authentication. Instructions for these tasks are provided for the operating systems and tools listed below. (Other SSH tools may also be used, but they are not covered in this documentation.)

- Microsoft Windows XP, Windows Vista, and Windows 7 using WinSCP
- Linux or UNIX using OpenSSH

Once you have set up SSH and generated a private/public key pair, you will transfer files to predefined directories at a secure file transfer protocol site (SFTP) set up by your Oracle Utilities Delivery Team.

Oracle Utilities also offers an optional fileset transfer manifest specification to optimize file transfer and provide troubleshooting support. See Fileset Transfer Manifest Specification for more information.

File Transfer Methods

Oracle Utilities Opower has two secure methods of receiving files from utilities: a secure file sharing system for manual file transfer, and an SFTP site for automatic file transfer.

Microsoft SharePoint

Oracle Utilities primarily uses Microsoft SharePoint for receiving manually transferred sample files at the beginning of the setup process before SFTP access is configured. Coordinate with your Service Delivery Manager to receive your SharePoint repository location. Your Service Delivery Manager can provide access to select individuals on your team. SharePoint remains an option throughout the program for manually exchanging sensitive files and is a good option for non-technical users who cannot access the SFTP site.

Secure File Transfer Protocol (SFTP)

Prior to launch, your will need to verify that your system can automatically transfer files to the Oracle Utilities Opower SFTP site. We exclusively use key-based authentication for secure connectivity between your data center and the Oracle Utilities Opower data center. In order to establish that connection, your team will need to provide an SSH public key and install your private key on your own server(s). Your Delivery Team will use your public key to create your SFTP account and provide your user ID.

After files have been submitted to the proper directory, the Oracle Utilities Opower file reception system begins processing them within a few minutes of receiving them.

> **Note**
>
> Canadian utilities are expected to use the SFTP site for the duration of the program.

The directories within your SFTP site are listed below.

- **/shared_files**: Use this directory to transfer files that are not meant to be loaded directly into the application. That includes all sample files, historical files, and incremental files until your Delivery Team approves at least one incremental file as production-ready.
- **/Uploads**: When up and running, your production instance of the Oracle Utilities Opower platform automatically loads all files placed in this directory. Once your Delivery Team approves an incremental file, your team will need to configure your automated file transfer process to drop files in this directory exclusively.

Encryption

For added security, Oracle Utilities can accept Pretty Good Privacy (PGP) encrypted files. If you choose this option, your Delivery Team will send you a public key that you can use to encrypt files. These will then be decrypted with the Oracle Utilities Opower private key. The Oracle Utilities Opower application will then automatically decrypt all inbound files using this private key.

File Errors

If file errors occur during the file transfer process (for example, a corrupt or erroneous file is added to the **Uploads** folder and cannot be processed), then utility users with the proper credentials can log in to the Oracle Utilities Opower SFTP server to perform one of the following basic self-service tasks in order to resolve the error:

- Delete the corrupted file from the **Uploads** folder and upload a new file
- Replace the corrupted file by overwriting it with a new version that has the same file name

However, because these tasks involve modifying files in a production environment, they should only be done after receiving an overview of the process from Oracle Utilities. Your Oracle Utilities Opower Delivery Team will work with you to explain the process and provide more information.

File Retention Policy

Oracle Utilities Opower will only retain files for a set number of days before deleting them. See File Retention Policy for more information.

File Transfer Setup - Windows

The Windows file transfer setup procedure uses two tools: PuTTY for SSH key management and WinSCP for the graphical user interface (GUI). Although other options exist, Oracle Utilities has found this to be a simple and free option that can be used by most clients.

Downloading and Installing PuTTY

1. If you do not already have PuTTY installed, download it from the following location:

   http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

2. Install the executable (.exe) under the heading: **A Windows installer for everything except PuTTYtel**.

For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)

A .ZIP file containing all the binaries (except PuTTYtel), and also the help files

Zip file:	putty.zip	(or by FTP)	(RSA sig)	(DSA sig)

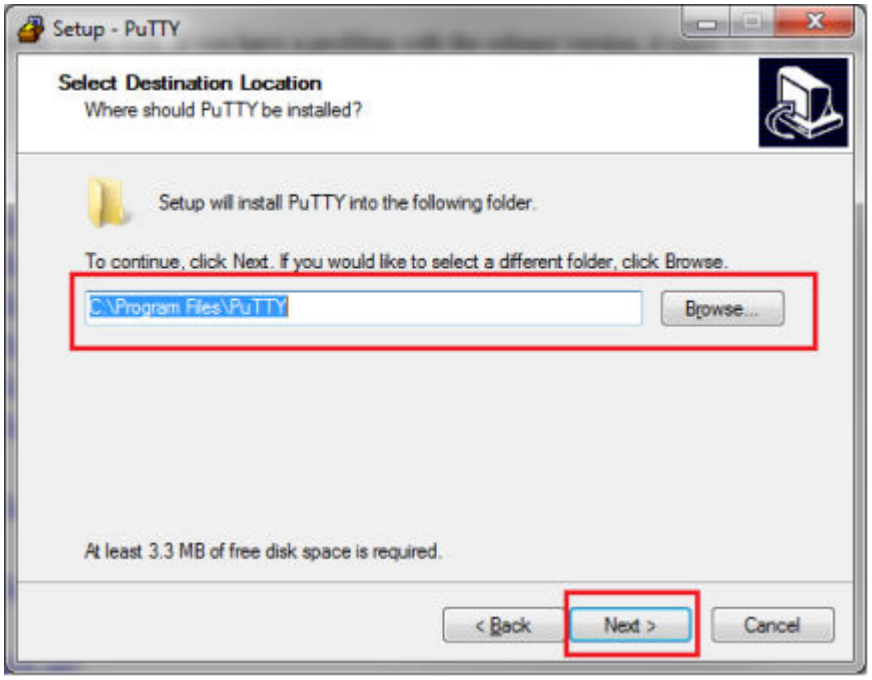
A Windows installer for everything except PuTTYtel

Installer:	putty-0.63-installer.exe	(or by FTP)	(RSA sig)	(DSA sig)

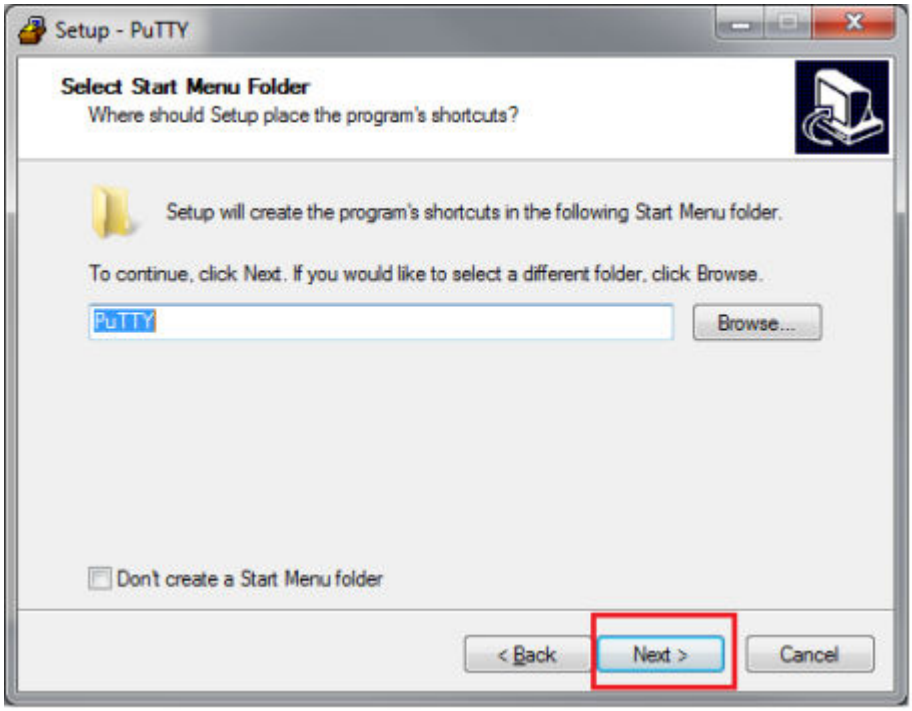
Checksums for all the above files

MD5:	md5sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-1:	sha1sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-256:	sha256sums	(or by FTP)	(RSA sig)	(DSA sig)
SHA-512:	sha512sums	(or by FTP)	(RSA sig)	(DSA sig)

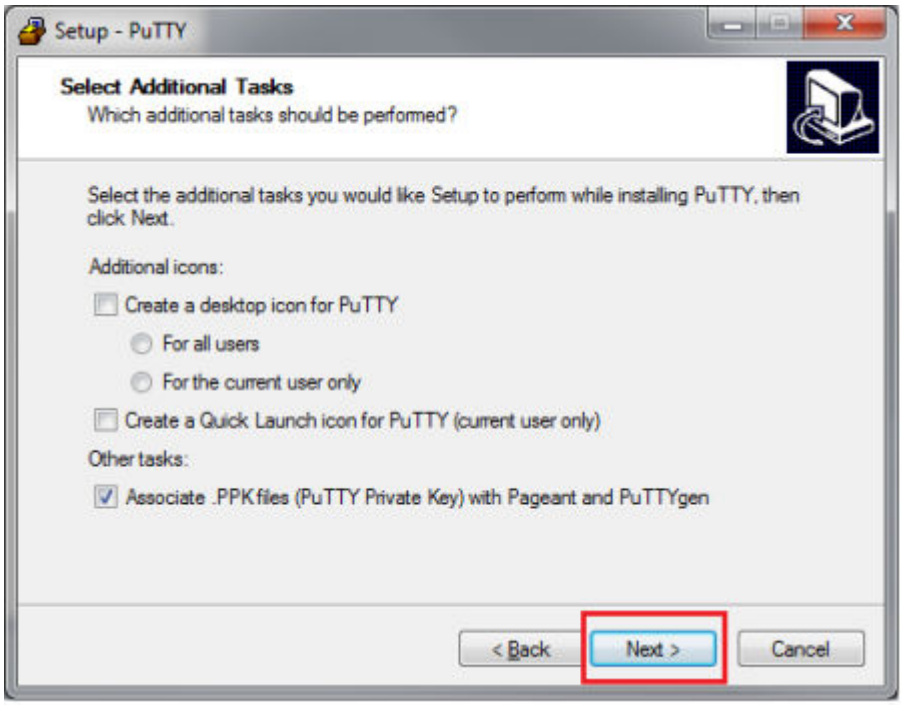
3. Retrieve and launch the **putty-0.62-installer.exe** from the location where it was saved.
4. When the initial setup window appears, click **Next**.
5. Identify where the program should be installed (by default, it is saved in **C:\Program Files\PuTTy**) and click **Next**.



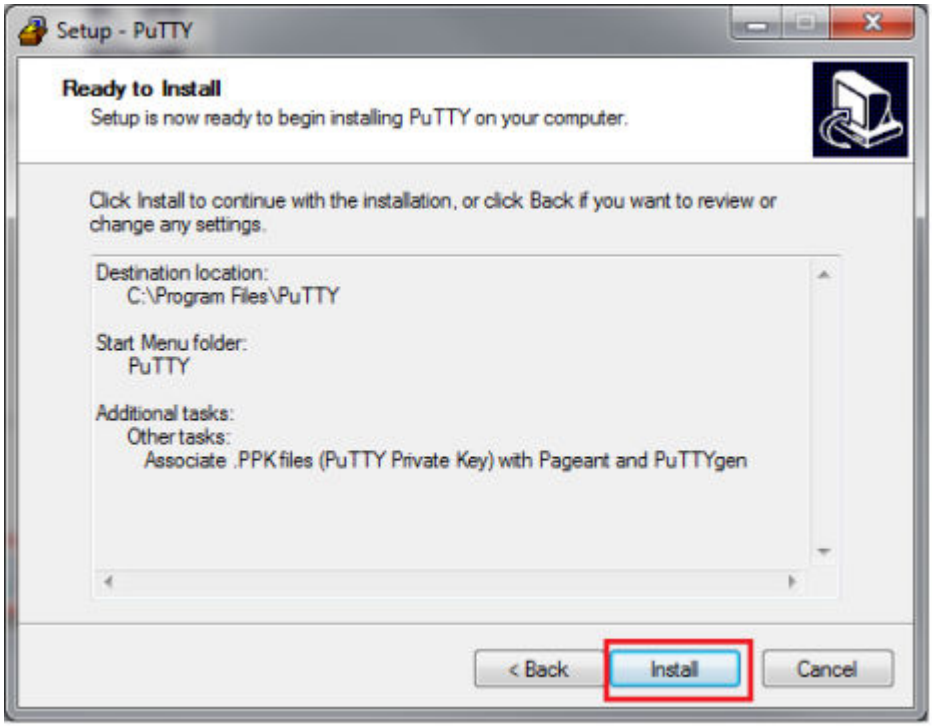
6. Name the shortcut folder and click **Next**.



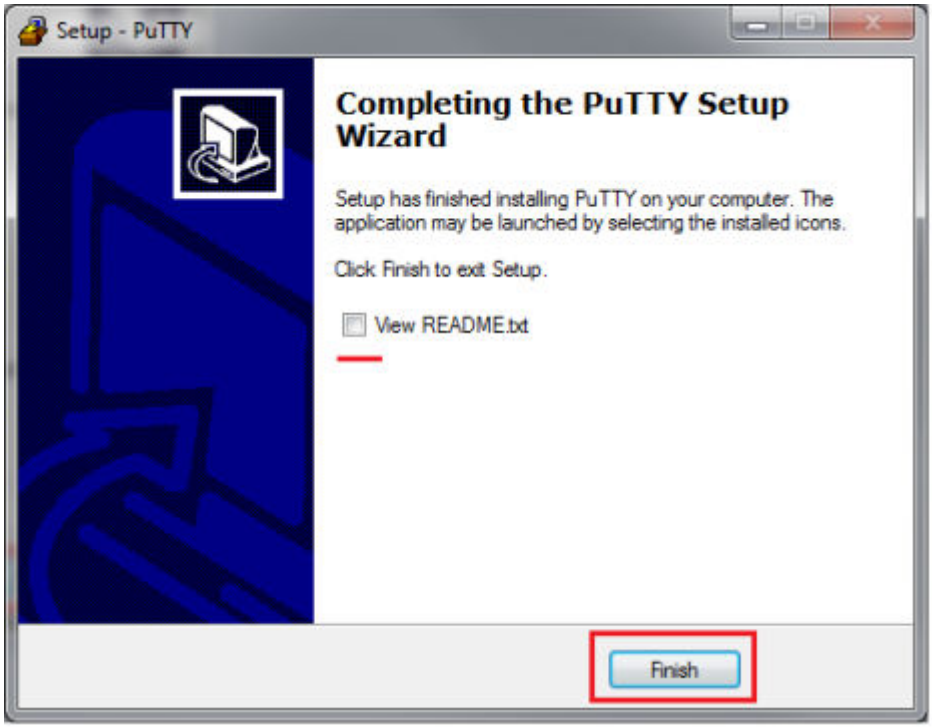
7. Make sure that the Associate .PPK files (PuTTY Private Key) with Pageant and PuTTYgen option is selected.



8. On the next screen, click **Install**.



9. Uncheck **View README.txt** and click **Finish** to complete setup.

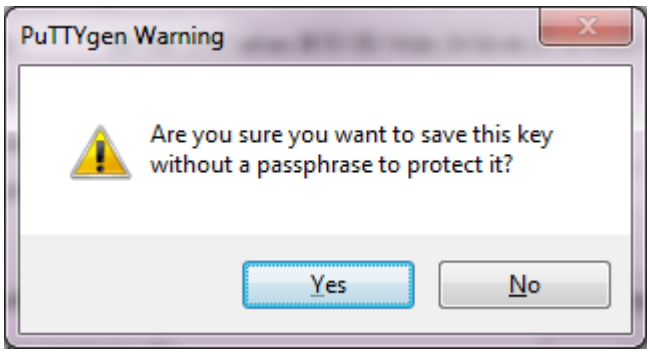


Generating SSH Keys with PuTTY

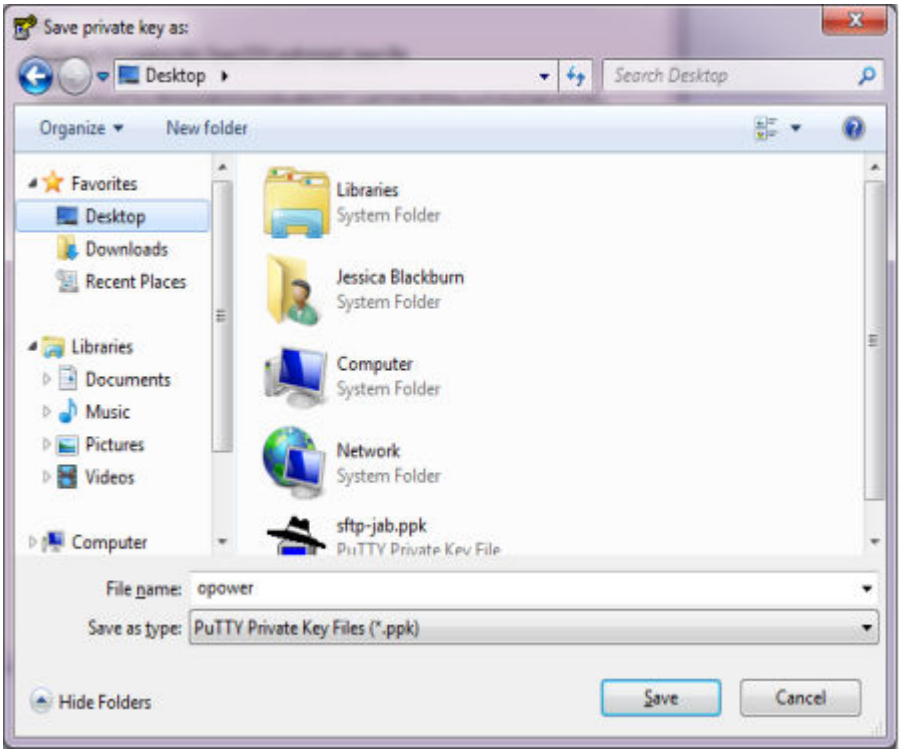
1. To launch PuTTYgen from Windows, select **Start > All Programs > PuTTY > PuTTYgen**.
2. In the **Parameters** area, select **SSH-2 RSA**.
3. Enter **4096** as the number of bits in the generated key.

4. Click **Generate**.
5. When prompted, move the mouse around in the window to generate randomness for the key.
6. *Optional*. Update the **Key comment** field to something more descriptive.

7. *Optional*. Set a passphrase by supplying an identical value in the **Key passphrase** and **Confirm passphrase** dialog boxes. **Note**: Before setting a passphrase, confirm that your operating system supports creating an automation script with a passphrase. Some Windows-based operating systems do not.
8. Click **Save private key**.
9. If you did not set a passphrase in the previous step, the dialog box below appears. Click **Yes** to continue.



10. Select a location in which to store the private key and click **Save**.



11. Save the public key by copying the content shown in **Public key for pasting into OpenSSH authorized_keys** section or clicking the **Save public key** button, selecting location, and pressing the **Start** button. Add a **.pub** extension to the file name.
12. Close the PuTTY Key Generator window.
13. Send the public key to your Oracle Utilities Service Delivery Manager.

> **Warning**
>
> Do not send the private key to Oracle Utilities. The private key should remain with your utility for security purposes. If you send the private key to Oracle Utilities, then it will be deleted, and a new public-private key pair will need to be generated.

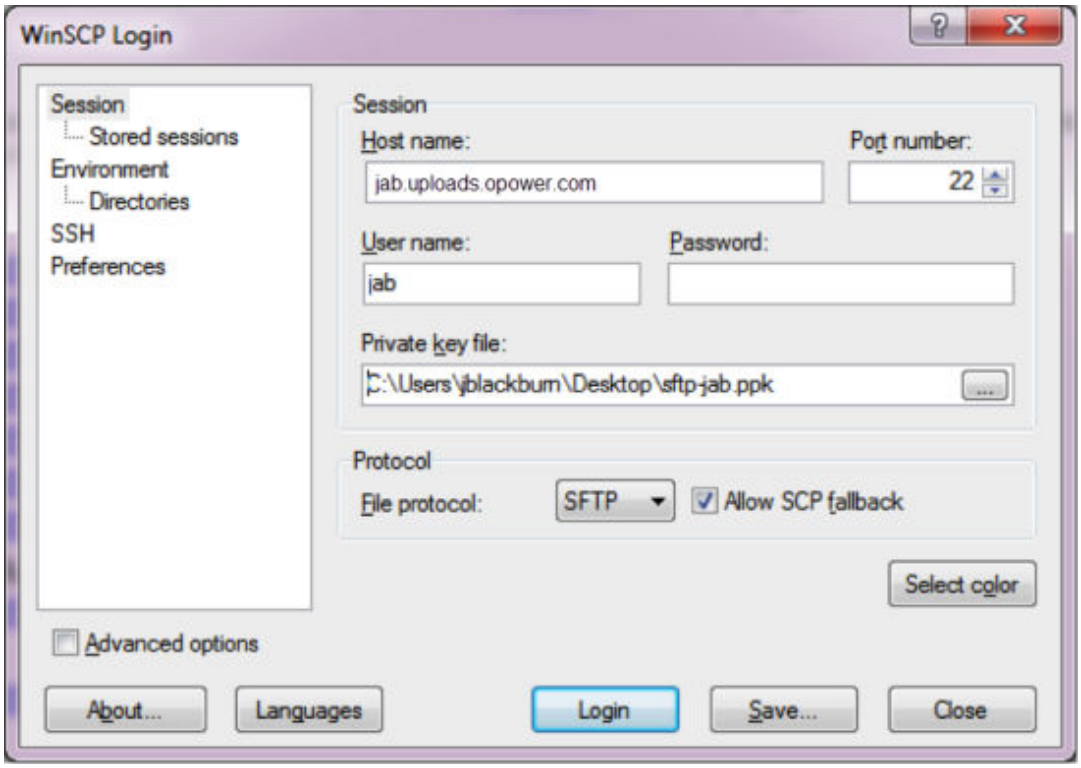
Transferring Files with WinSCP

1. Download WinSCP from the following location:

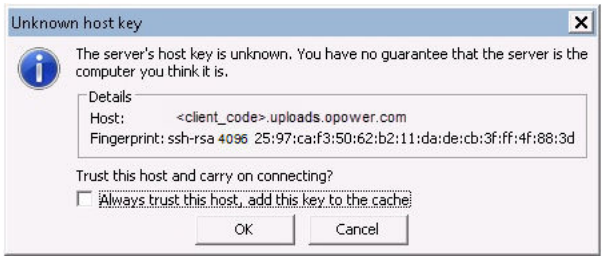
   http://winscp.net/eng/index.php
2. To launch the application from Windows, select **Start > All Programs > WinSCP > WinSCP**.
3. Set up a session by entering the following information in the **Session** area:
   a. **Host name**: Enter <*client_code*>.uploads.opower.com, where <*client_code*> is a specific three- or four-letter code provided by your Oracle Utilities Delivery Team. This is typically an acronym for your utility name. For example, if your utility name is the Great Energy Company, your client code might be gec.
   b. **User name**: Your client code. This user name must be entered as lower case.
   c. **Password**: Leave blank.

d. **Private key file**: Browse to the location of the saved private key file.

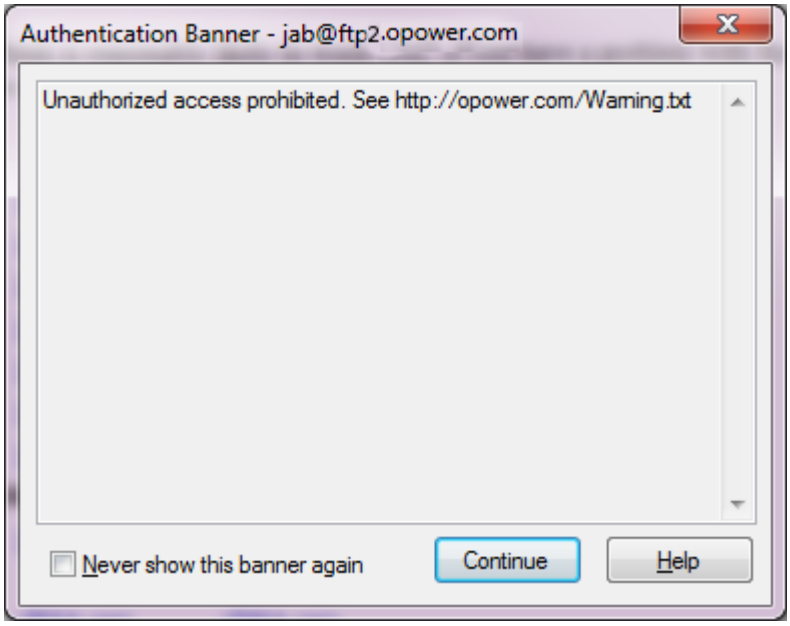
The screenshot below shows how the WinSCP Login dialog should look once the session details are entered.



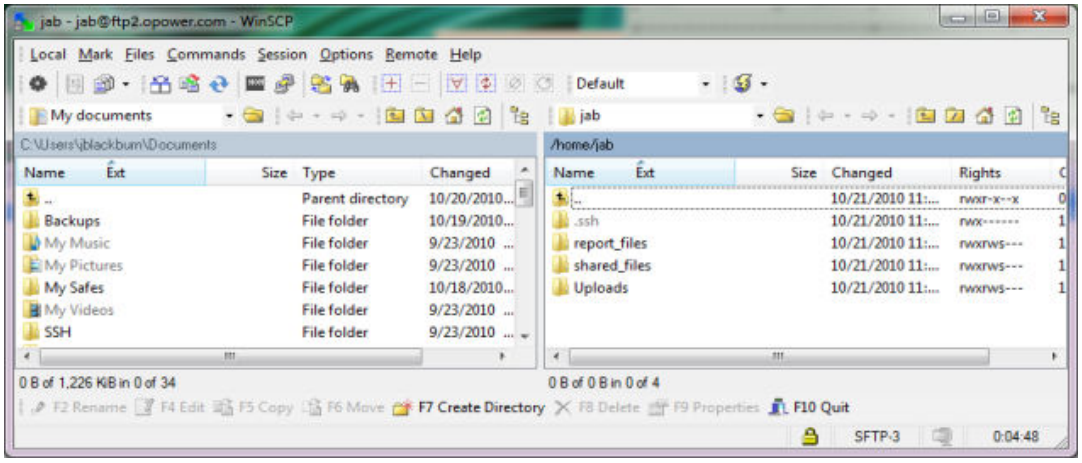
4. Click **Save** to save the session parameters.
5. Click **Login**. An **Unknown host key** dialog should appear, displaying a host name and an SSH server fingerprint. An example is shown below.**Note**: The fingerprint will be different depending on the host name that your Service Delivery Manager instructs you to use.



6. Click **OK**.
7. When the **Authentication Banner** window appears, click **Continue**.



8. Enter your passphrase for your private key if you have set one. If you are logged in successfully, the application should appear like the following:



9. To transfer a file to Oracle Utilities, select a file in the left panel and drag it to one of the directories in the right panel.

File Transfer Setup - Unix or Linux

The Unix or Linux file transfer setup procedure uses OpenSSH to create a key pair and the command line to transfer files to an SFTP site.

Generating SSH Keys with ssh-keygen

1. Install OpenSSH from the following location: http://www.openssh.com.
2. Create an SSH key pair using the following command:

```
[username@host tmp]$ ssh-keygen –b 4096 -t rsa -f
                     ~/.ssh/id_rsa_opower
```

3. Send the public key (~/.ssh/id_rsa_opower.pub) to your Oracle Utilities Service Delivery Manager.

Warning: Do not send the private key to Oracle Utilities. The private key should remain with your utility for security purposes. If you send the private key to Oracle Utilities, then it will be deleted, and a new public-private key pair will need to be generated.

Transferring Files with SFTP

Logging In

You can log in using the following command:

```
sftp -o IdentityFile=~/.ssh/id_dsa_opower
               <username>@<client_code>.uploads.opower.com
```

In this case, <*username*> is your client code. Your client code is provided by your Oracle Utilities Delivery Team, and is typically an acronym for your utility name. For example, if your utility name is the Great Energy Company, your client code might be gec. The username must be entered as lower case.

If you do not know what names to enter for either of these variables, please contact your Delivery Team. An example of a session is shown below:

```
[username@host tmp]$  sftp -o
IdentityFile=~/.ssh/id_rsa_opower
abc@abc.uploads..opower.com

Connecting to abc.uploads.opower.com...

Unauthorized access prohibited.
See http://opower.com/Warning.txt

sftp>
```

It is expected behavior to see the “Unauthorized access prohibited” warning.

Transferring a File

To transfer a file, use the cd command to change the directory and enter the command to send the file. In the example below, the file **hello.txt** is placed in the **Uploads** directory. (Note that the name of the **Uploads** directory is capitalized and case-sensitive.)

```
sftp> cd Uploads

sftp> put hello.txt

Uploading hello.txt to
/home/abc/Uploads/hello.txt

hello.txt   100%   20
0.0KB/s   00:00
```

File Retention Policy

According to the retention policy described in the Oracle Cloud Hosting and Delivery Policies document (which can be found online at Oracle Cloud Services), all incoming and outgoing files added to the Oracle Utilities Opower SFTP server will be retained for a maximum of 60

days. After this time period, they will be automatically deleted. Copies of files added to the SFTP server are stored in a file archive until the termination of the contract.

For utilities in the European Union, files are retained for a maximum of 30 days in both the SFTP server and the file archive. After this period the files are deleted.

File Transfer FAQs

What tools are required to transfer files securely to Oracle Utilities**?**

Secure Shell (SSH) and a secure file transfer protocol (SFTP) site are required. Oracle Utilities has adopted file transfers over SSH because it offers strong encryption, has free (OpenSSH) and commercial versions, and is supported across multiple desktop and server operating systems.

Can I transfer files using a secure copy (SCP)?

No, only SFTP is supported. An error message like the following will be displayed if you attempt to use SCP:

```
username@host tmp]$ scp -i id_rsa_opower hello.txt
abc@abc.uploads. opower.com:/Uploads/

Unauthorized access prohibited. See
http://opower.com/Warning.txt

unknown user 1632

lost connection
```

Can I log in to the SFTP server using a password instead of an SSH key?

No. For security reasons, Oracle Utilities only allows authentication through the use of SSH keys.

Are there any minimal encryption requirements for SSH Key generation?

Yes. As shown in the examples in this documentation, SSH keys should be 4096 bit, RSA, SSH2 format. See File Transfer Setup - Unix or Linux and File Transfer Setup - Windows for the examples.

Is there a limited number of SFTP connection attempts?

Yes. The Oracle Utilities Opower firewall imposes a limit of 20 connection attempts for every 60 second period. If the number of connection attempts exceeds this threshold, the file transfer process will fail.

Is there a file size restriction for transferred files?

Yes. Files can be up to 4 gigabytes uncompressed. Additionally, there are limits on the number of files that can be transferred in a day depending on how large they are. See File Transfer Limits for details.

Is there a file name length restriction for transferred files?

Yes. The name of a data file cannot exceed 183 characters.

What happens if there are any issues with the file transfer?

Oracle Utilities Opower uses an internal file transfer monitoring service that automatically sends alerts to internal team members if a file transfer fails. This allows issues to be quickly

identified, and corrective action to be taken. Oracle Utilities Opower will contact you if there are any file transfer issues that require your attention.

Fileset Transfer Manifest Specification

A transfer manifest is a small metadata file that describes the set of files that follow it. This file is optional in most cases. Depending on the terms of your contract, your Oracle Utilities Delivery Team will inform you whether this file is required. However, it is highly recommended for several reasons.

- Transfer manifests allow Oracle Utilities to validate that we have received all the files that we are supposed to receive, and that all of them have arrived intact. In the event of a transfer failure, such as a file that gets truncated due to a connection dropping, a transfer manifest will allow Oracle Utilities to detect the error condition and notify the utility. This typically leads to faster resolution.
- Transfer manifests also allow the utility to specify the order in which files should be considered. File order coercion is currently supported in some Oracle Utilities Opower importers, but not in others. If you expect to depend on the order in your manifest, contact your Oracle Utilities Delivery Team about whether the files will be processed by an importer that currently supports file order coercion.

Oracle Utilities recommends that utilities use a transfer manifest every time they send a set of files. Files sent without a manifest will also be processed, but in some cases Oracle Utilities will have no other way to detect issues with the file.

If your utility chooses to use the transfer manifests, note that the transfer manifest file must always precede the first data file. A data file cannot be sent to Oracle Utilities before the manifest file is sent.

Transfer Manifest File Contents

Fileset Transfer Manifest files must be delivered in YAML format. The file must contain the following:

- `meta.version`: Version identifier.
- `transfer.validity_window (int)`: Expiration time in seconds for this manifest, measured from when the file was received. If any of the specified files have not appeared when the time window ends, then the transfer is failed and notifications are sent. Files that have started but not yet finished transferring count as having appeared.
- `fileset`: A list of fileset entries. This must be a list to preserve ordering.
- `fileset.[FILENAME].size (int)`: Size in octets of the object being transferred.
- `fileset.[FILENAME].cksum (str)`: Checksum of the object being transferred.
- `fileset.[FILENAME].ckalg (str-enum)`: Checksum algorithm used to generate the checksum string. Valid values are `md5`, `sha1`, `sha256`, `ripemd-160`. If your system cannot support any of these checksum algorithms, you can discuss alternatives with your Delivery Team.

File Indentation Structure

Fileset Transfer Manifest files adhere to the following indentation rules:

- Section headers, like meta, transfer, and fileset, are preceded by zero spaces.

- Section content, like meta.version and transfer.validity_window, is preceded by two spaces.
- Names of the files in the file list, like opwr_glbx_customer_ABCDEFG.gz.pgp, are preceded by two spaces, a '-' (ASCII code 0x2d), and then another space.
- Contents of the file list, like size: 10945113, are preceded by six spaces.

Example Fileset Transfer Manifest File

The following shows the contents of an example fileset transfer manifest.

```
meta:

  version: 0

transfer:

  validity_window: 600

fileset:

  - opwr_glbx_interval15_electric_ABCDEFG.DAT.gz.pgp:

      size: 10945113

      cksum: abcdef0123456789abcd

      ckalg: md5

  - opwr_glbx_interval15_gas_ABCDEFG.DAT.gz.pgp:

      size: 10112608

      cksum: 0123456789abcdabcdef

      ckalg: md5

  - opwr_glbx_customer_ABCDEFG.gz.pgp:

      size: 2149602

      cksum: abc123abc123abc123ab

      ckalg: md5
```

File Naming Conventions

Manifest files must use the following naming convention:

```
transfer_manifest_<UNIQUE_ID>.yaml
```

File Transfer Limits

There is a limit to the file size and file count that the Oracle Utilities Opower data ingest system can process in a day. The table below indicates the limits for *uncompressed* data files.

> **Note**
>
> Given the minimum setup and break down time that the data transfer jobs require to ingest data, it is recommended that you send a small number of large files to Oracle Utilities Opower rather than a large number of small files. A small number of large files can be processed faster than a large number of small ones.

File Size (Gb)	Daily File Count Limit
1	45
2	20
3	15
4	10

As an example, the first row indicates that if your files are 1 gigabyte in size, then you can transfer a maximum of 45 such files in a day to Oracle Utilities Opower. The second row indicates that if your files are 2 gigabytes in size, then you can transfer a maximum of 20 such files in a day.

4

Core Data Standards

Core data standards contain the foundational data elements used in the Oracle Utilities Opower program. They include data elements for uniquely identifying utility customers and tracking customer energy usage. These data standards are required to run the base components of the Opower program. As shown below, there are two options for following the core data standards. Your Delivery Team will work with you to identify the best option for your situation.

Option 1 (Preferred)

Option 1 includes newer data standards and a broader range of data fields.

- Account Data Transfer
- Billing Data Transfer
- Premise Data Transfer

Option 2

Option 2 includes a legacy data standard that is used among existing utilities. It will eventually be deprecated and replaced by the data standards in option 1.

- Legacy Billing Data Transfer

Account Data Transfer

The account data files contain information about a utility customer's service agreement, contact details, rate plan, and other account information. The key files include:

- Main Account File
- Person Address File
- Service Agreement Service Point File

Not all of the data in each file is required. Your Delivery Team will work with you to determine what is required for your program, and help you construct the relevant files during program launch. See Account Data File Specifications for more information.

Account data can be generated as a data extract using the Oracle Utilities DataConnect solution. DataConnect is available for utilities that have the following Oracle Utilities products:

- Customer Care and Billing 2.8 or higher
- Customer to Meter 2.8 or higher
- Meter Data Management 2.8 or higher

If DataConnect is available to you, your Delivery Team will work with you to leverage it for the account file.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Account Data Integration Process

Integration of your account data into the Oracle Utilities Opower platform involves the following phases: sample historical file integration, full historical file integration, and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

See Data Transfer Process Overview for more information.

Account Data File Specifications

The account data files contain information about a utility customer's service agreement, contact details, rate plan, and other account information. There are several distinct account files that Oracle Utilities Opower will ask you to provide. Each file consists of multiple fields and has a unique naming convention.

- Main Account File
- Account Fields
- Person Fields
- Service Agreement Fields
- Service Agreement Rate Fields
- Person Address File
- Service Agreement Service Point File

Not all of the data in each file is required. Your Delivery Team will work with you to determine what is required for your program. There are a few general rules that govern the account data:

- Only one person may be the main account customer at any point in time.
- Switching main account customers can be achieved by supplying two rows in the input file. One row must mark the current main account customer as not the main account customer. Another row must identify the new main account customer. See Example Data: Switching the Main Customer on the Account for more information.
- The `end_date` field must be null or must be strictly later than `start_date`, for any entity that specifies both.

Main Account File

The main account file consists of multiple data fields containing a customer's account information. Although information about the fields is organized into subtopics for easier reference, the fields should be included in a single file and sent to Oracle Utilities Opower.

- Account Fields
- Person Fields

- Service Agreement Fields
- Service Agreement Rate Fields

The main account file must follow a specific naming convention. See Account Data File Name Conventions for details.

Account Fields

The account fields include data related to the account ID, name, setup, classification, and billing cycle of a customer's account.

> **Note**
>
> Although the fields below are described separately from the other fields in the Main Account Data File, they must be included in the same file as the other fields, with all columns included in every row. See Account Data File Specifications for more information.

Field	Description
account_id	The primary identifier of an account. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary key.
account_name	A user-friendly name for the account. The purpose of this field is to make the account name easier to understand when displayed in the user interface of an Oracle Utilities product. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
setup_datetime	The identified setup date for an account. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
customer_class	The classification for the customer. Depending on what channels and products are enabled for your Oracle Utilities program, this field segments your customer population for different experiences on printed reports or on the web. **Allowed Values**: • AGRICULTURAL • COMMERCIAL • INDUSTRIAL • RESIDENTIAL • SMB • OTHER **Type**: VARCHAR(40) **Can Be Empty?**: Yes. When empty, this defaults to RESIDENTIAL.

Field	Description
bill_cycle_code	Any unique alphanumeric string that specifies the time of month at which the meter read occurs. For Oracle Utilities products that include normative comparison features, this data is used to ensure customers are compared against other customers with recent usage reads. This field should designate the service points that are read at approximately the same time. Utilities typically have 21 distinct read cycles corresponding to the business days in a month. **Example Values**: M01,M02 **Type**: VARCHAR(24) **Can Be Empty?**: Yes, but preferred to not be empty.
bill_cycle_description	A description of the bill cycle. **Example Values**: • Monthly bill cycle - day 1 • Monthly bill cycle - day 2 **Type**: VARCHAR(255) **Can Be Empty?**: Yes.
account_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Person Fields

The person fields contain information related to the person associated with an account, such as first name, last name, language preference, and key contact information.

Notes:

- Although the fields below are described separately from the other fields in the Main Account Data File, they must be included in the same file as the other fields, with all columns included in every row. See Account Data File Specifications for more information.
- Oracle Utilities Opower supports up to three phone numbers and email addresses per account. In such cases, each field should have a suffix of _1, _2, and _3 to differentiate them. See Phone Number Examples and Email Address Examples below for details.
- Empty values in fields or missing fields will be interpreted as deletions, and previously-received data for those fields will be removed. See Empty Values and Deletions below for details.

Field	Description
person_id	A unique identifier for a person in a utility's Customer Information System or other system of record. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
person_type	A flag to indicate whether the person is a business or not. **Allowed Values**: • Person • Business **Type**: ENUM **Can Be Empty?**: No.

Field	Description
first_name	The customer's first name. If `person_type` = `person`, then at least one of `first_name` and `last_name` must be specified. If `person_type` = `Business`, then `first_name` and `last_name` are optional if a `business_name` is provided. **Type**: VARCHAR(408) **Can Be Empty?**: Yes. Business Key.
last_name	The customer's last name. If `person_type` = `person`, then at least one of `first_name` and `last_name` must be specified. If `person_type` = `Business`, then `first_name` and `last_name` are optional if a `business_name` is provided. **Type**: VARCHAR(408) **Can Be Empty?**: Yes. Business Key.
business_name	The name of the business. If `person_type` = `Business`, this should be populated. If `person_type` = `PERSON`, this can be empty. **Type**: VARCHAR(408) **Can Be Empty?**: Yes.
language_preference	A country and language code combination. For a full list of country codes, see Wikipedia - Officially Assigned Code Elements. For a full list of language codes, see Wikipedia - List of ISO 639-1 Codes. The full language code is a combination of the ISO 639-1 two-character language code and the ISO 3166-1 two-character country code. **Example Value**: en_US **Type**: VARCHAR(5) **Can Be Empty?**: Yes.
is_main_customer	A flag to indicate whether the person is the primary customer for the account. See Example Data: Switching the Main Customer on the Account below for instructions on how to change who is the main customer. **Note**: Only one person can be the primary customer of an account at any one time. Oracle Utilities Opower communicates with the primary account holder for many types of communications. **Allowed Values**: • Y This person is the primary customer. • N This person is not the primary customer. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: No.
bill_route_type	The method used to transmit the bill to the customer. **Example Values**: • EBILL Route the bill by email. • POSTAL Route the bill by postal service. **Type**: VARCHAR(8) **Can Be Empty?**: No.

Field	Description
person_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
phone_type_X	The type of phone number, such as Home or Mobile. Up to three sets of phone number columns are allowed for a customer. If one row includes two or more sets of phone number columns, then every other record in the file must also have two or more sets of phone number columns even if many of the columns are left empty. See Phone Number Examples below for details. **Type**: VARCHAR(45) **Can Be Empty?**: Yes.
phone_number_X	The customer's number in one of the following formats: (999) 999-9999 or 999 999 9999. International phone number formats are supported using the E.164 standard. Up to three sets of phone number columns are allowed for a customer. See Phone Number Examples below for details. **Type**: VARCHAR(20) **Can Be Empty?**: Yes.
phone_extension_X	The extension of the phone number. **Type**: VARCHAR(45) **Can Be Empty?**: Yes.
email_address_X	The customer's email address. Up to three emails may be provided. If one row includes two or more sets of email columns, then every other record in the file must also have two or more sets of email columns even if many of the columns are left empty. See Email Address Examples below for more information. **Type**: VARCHAR(344) **Can Be Empty?**: Yes.
email_type_X	The type of email address, such as for personal or work-related use. Up to three emails may be provided. See Email Address Examples below for more information. **Type**: VARCHAR(45) **Can Be Empty?**: Yes.

Example Data: Switching the Main Customer on the Account

To switch the main account customer, supply two rows in the input file. One row must mark the current main account customer as *not* the main account customer in the `is_main_customer` field. Another row must identify the new main account customer in the same field.

For example, the original data file would have this data:

person_id	<other fields>	is_main_customer	account_id
1234	...	Y	9012
5678	...	N	9012

Then the second iterative data file would have the data below. The only change is that the `is_main_customer` field was updated to mark one customer as the main person and the other as not.

person_id	<other fields>	is_main_customer	account_id
1234	...	N	9012
5678	...	Y	9012

Phone Number Examples

Up to three phone numbers may be provided for a single account. In the example below, one customer account has three phone numbers, another has two phone numbers, another has only one phone number, and another has no phone numbers at all. Note that despite these differences, each row has three "sets" of phone number columns even though many columns are empty. In other words, even if one customer has three phone numbers while another does not, there must still be three sets of phone number columns in every row of the data file.

person_id	[Other Fields]	phone_type_1	phone_number_1	phone_extension_1	phone_type_2	phone_number_2	phone_extension_2	phone_type_3	phone_number_3	phone_extension_3
12223	...	Home	240-123-4567		Mobile	240-111-2345		Work	240-222-2345	123
34445	...	Home	780-765-4321		Mobile	780-567-1234				
56667	...	Mobile	910-876-5432							
78889	...									

Email Address Examples

Up to three email addresses may be provided for a single account. In the example below, one customer account has three email addresses, another has two addresses, another has only one address, and another has no address at all. Note that despite these differences, each row has three "sets" of email address columns even though many columns are empty. In other words, even if one customer has three email addresses while another does not, there must still be three sets of email address columns in every row of the data file.

person_id	[Other Fields]	email_type_1	email_address_1	email_type_2	email_address_2	email_type_3	email_type_3
12223	...	Personal	abc123smith@gmail.com	Personal	abc123smithfamily@hotmail.com	Work	abc123smithjr@work.com
34445	...	Personal	def.456.martin@gmail.com	Work	steve.martin@work.com		
56667	...	Personal	ghi.789.jones@yahoo.com				
78889	...						

Empty Values and Deletions

Empty values in fields or missing fields will be interpreted as deletions, and previously-received data for those fields will be removed. For example, if a previous file had a `phone_number_3` field filled in, and a subsequent file is missing the field or transmits it with an empty number, the `phone_number_3` field value will be deleted from the Oracle Utilities Opower operational data store. The same principle applies for any `email_address_x` fields.

Service Agreement Fields

The service agreement data entity contains information about the name, duration, and status of a utility customer's service agreement.

> **Note**
>
> Although the fields below are described separately from the other fields in the Main Account Data File, they must be included in the same file as the other fields, with all columns included in every row. See Account Data File Specifications for more information.

Field	Description
service_agreement_id	The unique identifier of a service agreement. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
service_agreement_name	A user-friendly name for the service agreement. The purpose of this field is to make the service agreement name easier to understand when it appears in the user interface of an Oracle Utilities product. **Type**: VARCHAR(40) **Can Be Empty?**: Yes. Business Key.
service_agreement_start_datetime	The start date of a service agreement. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
service_agreement_end_datetime	The end date of a service agreement. If left blank, this will be considered an active service agreement. If another service agreement with a later `start_datetime` is sent later on, you should update the `end_datetime` of the prior service agreement to the second before the `start_datetime` of the new service agreement. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
service_agreement_type_code	A code designating characteristics about the service agreement in addition to the information in the service_type_code column. For example, this column could be used to designate whether the service agreement is for residential or non-residential customers, customers who are on budget billing, customers who have solar power or an electric vehicle, and so on. **Type**: VARCHAR(8) **Can Be Empty?**: Yes.
service_agreement_type_description	A description of the service agreement type code defined in `service_agreement_type_code`. **Type**: VARCHAR(60) **Can Be Empty?**: Yes.

Field	Description
special_role_flag_code	A code designating a special role for the service agreement. **Example Values**: These example values are described in special_role_flag_code_description. • BC • BD • CD • IN • LO • NB • PA • WO **Type**: VARCHAR(8) **Can Be Empty?**: Yes.
special_role_flag_description	A description of the special role of the service agreement. **Example Values**: • Billable Charge • Bill Determinants Required • Cash Deposit • Interval • Loan • Non-Billed Budget • Payment Arrangement • Write Off **Type**: VARCHAR(60) **Can Be Empty?**: Yes.
service_type_code	A code designating the type of service provided through the service agreement. This is based on the service agreement fuel type or resource type. **Allowed Values**: • E • G • W • WW • O **Type**: ENUM **Can Be Empty?**: No.
service_type_desc	A description of the code that represents the type of service provided through the service agreement. This is based on the service agreement fuel type or resource type. **Allowed Values**: • ELECTRIC • GAS • WATER • WASTE_WATER • OTHER **Type**: VARCHAR(100) **Can Be Empty?**: Yes.

Field	Description
status_code	A code designating the status of the service agreement. You will need to map your service agreement statuses to one of the valid statuses below. Your Delivery Team can work with you on the exact mapping. **Allowed Values**: • 05 Incomplete: A service agreement setup has been started, but not all information is available for it yet. The data file can omit service agreements that are in this state. • 10 Pending Start: Service agreement has been set up, but service has not yet started. • 20 Active: A service agreement is fully active and service is being delivered. • 30 Pending Stop: A stop to a service agreement has been initiated, but service has not yet stopped. • 40 Stopped: Service agreement is fully stopped and no service is being delivered. Bills may still be outstanding. • 50 Reactivated: Service agreement has been reactivated after being stopped (for example, temporary suspension of service for non-payment). • 60 Closed: Service agreement has been fully terminated, and all related bills are closed. • 70 Canceled: Service was never delivered, there are no outstanding bills, and the service agreement can be canceled. **Type**: VARCHAR(50) **Can Be Empty?**: No.
status_description	A description for the status code of the service agreement. **Allowed Values (from Oracle Utilities Customer Care and Billing)**: • Incomplete • Pending Start • Active • Pending Stop • Stopped • Reactivated • Closed • Canceled **Type**: VARCHAR(100) **Can Be Empty?**: No.

Field	Description
business_classification_code	The code for classifying a business according to a specific kind of economic activity. These codes may come from a classification system such as the NAICS (North American Industry Classification System), SIC (Standard Industrial Classification), or ISIC (International Standard of Industrial Classification). You do not have to specify which classification system your CIS uses. However, the codes should come from only one classification system. Contact your Delivery Team to ensure Oracle Utilities Opower sets up the right configuration for the classification system you are using. For example, if you are using NAICS, the values should match one of the codes from the NAICS reference: https://www.naics.com/search/ **Type**: VARCHAR(20) **Can Be Empty?**: Yes.
service_agreement_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Service Agreement Rate Fields

The service agreement rate data entity contains information about the rate used to calculate a service agreement's bill amounts. Rates will be associated with the service agreement on the same line of the file and should be the current rate code in effect for the service agreement at the time of the extract. Oracle Utilities Opower will keep a history of rate codes that were sent. It is assumed that a new `effective_date` is a new entry in the history. Having the same value as a previously-sent `effective_date` is treated as a correction.

> **Note**
>
> Although the fields below are described separately from the other fields in the Main Account Data File, they must be included in the same file as the other fields, with all columns included in every row. See Account Data File Specifications for more information.

Field	Description
sa_rate_code	The rate used to calculate the service agreement's bill segments. Multiple rows will only exist if the service agreement's rate changes over time. **Type**: VARCHAR(40) **Can Be Empty?**: No.
effective_datetime	The date the rate becomes effective. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
service_agreement_rate_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Example Data

For example, consider the following sequence of files sent to Oracle Utilities Opower and the resulting operational state.

File	service_agreement_id	other fields	sa_rate_code	effective_date	Result
1	1234567890	...	ER-BASIC	2018-01-01 00:00:00	Inserted as first entry, effective from 2018-01-01 00:00:00 until the end of time.
2	1234567890	...	ER-DMINU	2018-05-01 00:00:00	Inserted as second entry, effective from 2018-01-01 00:00:00 until the end of time. Previous entry now made to end as of 2018-04-30 23:59:59.
3	1234567890	...	ER_BASIC	2018-05-01 00:00:00	Correction of data in previous file #2, and `sa_rate_code` updated.
4	1234567890	...	ERES1	2018-06-01 00:00:00	Inserted as third entry, effective from 2018-01-06 00:00:00 until the end of time. Previous entry now made to end as of 2018-05-31 23:59:59.

Person Address File

The person address file contains data related to the mailing address associated with a person. The data for this entity is expected to be sent in a separate file. Only the latest address for a `person_id` is stored. The file must follow a specific naming convention. See Account Data File Name Conventions for details.

Field	Description
person_id	The person ID that this address is related to. **Type**: VARCHAR(40) **Can Be Empty?**: No.

Field	Description
mail_address_1	Address line 1 of the person. **Type**: VARCHAR(254) **Can Be Empty?**: No.
mail_address_2	Address line 2 of the person. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
mail_address_3	Address line 3 of the person. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
mail_address_4	Address line 4 of the person. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
locality	The city of the person. **Type**: VARCHAR(90) **Can Be Empty?**: No.
mail_postal_code	The postal code of the person. **Type**: VARCHAR(12) **Can Be Empty?**: No.
admin_area_1	A first-order civil entity below the country level, such as states in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: No.
admin_area_2	A second-order civil entity below the country level, such as counties in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: Yes.
admin_area_3	A third-order civil entity below the country level, such as suburbs and neighborhoods in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: Yes.
country	The two-character code representing the country of the premise. For a full list of country codes, see Wikipedia - Officially Assigned Code Elements. **Type**: VARCHAR(2) **Can Be Empty?**: Yes.
person_address_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Service Agreement Service Points File

The service agreement service points file includes fields related to the relationship between a customer's service point and service agreement. The data for this entity is expected to be sent in a separate file that follows a specific naming convention. See Account Data File Name Conventions for details.

If there are any changes to a service agreement's service point list, then Oracle Utilities Opower will attempt to merge them into a cohesive history. However, the `sa_sp_start_date` and `sa_sp_end_date` history must not overlap for a single service agreement and service point combination. If the same combination of `service_agreement_id`, `service_point_id`, and `sa_sp_start_date` is sent again, it is assumed to be a correction and will overwrite previously-imported values.

Field	Description
service_agreement_id	The unique identifier of a service agreement. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
service_point_id	The unique identifier of a service point on a service agreement. This relates to the `service_point_id` field in the Premise file. See Service Point Fields for more information. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
service_point_name	A user-friendly name for the service point. The purpose of this field is to make the service point name easier to understand when displayed in the user interface of an Oracle Utilities product. **Type**: VARCHAR(40) **Can Be Empty?**: Yes. Business Key.
usage_flag	A flag that indicates whether usage for this service point is additive or subtractive for the overall service agreement, or if the usage is informational only. **Allowed Values**: • + (*plus sign*) Add the value for the overall usage of the service agreement. • - (*minus sign*) Subtract the value from the overall usage of the service agreement. • c Check. Check service points are ignored by billing. They may be used to track informational meters such as those that directly track solar generation for customers. This value can be uppercase (C) or lowercase (c). **Note**: Rows containing any values for this field other than plus (+), minus (-), or c will be rejected. **Type**: ENUM **Can Be Empty?**: No.
usage_percent	The percentage contribution to the total usage of the related service agreement. For example, a service agreement in a roommate arrangement may have one service point but is only contributing 50% of its usage to each of two different service agreements. This field defaults to 100 if no value is provided. **Type**: DECIMAL(3,0) **Can Be Empty?**: Yes.
sa_sp_start_datetime	The date on which the service agreement was associated with the service point. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No. Primary Key.

Field	Description
sa_sp_end_datetime	The date on which the service agreement was disassociated from the service point. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Account Data Order and File Format

Account data must be sent in a particular order and file format.

Data File Order

The Oracle Utilities Opower platform processes files according to the order received, then dated. If we receive multiple files on the SFTP server in the same time frame (as could be done for full historical data files), they will be automatically queued by their filename date stamps.

For example, if a file is received on the SFTP server named *opwr_util_20140520.txt* and then receive one named *opwr_util_20140519.txt* an hour later, *opwr_util_20140520.txt* will be processed first despite the fact that its date stamp is later than the *opwr_util_20140519.txt*. However, if these files are received at the same time, the file name data stamp will take precedence and *opwr_util_0140519.txt* will be processed before *opwr_util_20140520.txt*.

To ensure that all files are queued in the correct order, files should be named so that the date stamps are consistent with the intended load order of the files.

Data Order Matters!

Data (rows) for each service point should always be arranged from the oldest to the most recent to ensure that your data is processed in the correct order.

- **Multiple Usage Reads for a Single Service Point**: Any time a single file contains more than one usage read for a single service point covering the same period of time, the most recent read needs to appear last in the file. If the usage periods for those two reads are the same, Oracle Utilities considers the last record read to be a correction.
- **Multiple Records for the Same Entity**: If there are multiple records for the same entity (such as customer, account, or service point), in the same file, only the last record for that entity is loaded. Oracle Utilities assumes that the last record represents the most current data from the utility.
- **Customer Information**: The data in the last row of the last file processed is always the active data in the Oracle Utilities database for that customer.

File Format

All data files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- All data must be sent as tab-delimited text.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported. **Note**: The only field that does not yet support non-Latin characters is `premise_id`. Contact your Delivery Team if you need to send non-Latin characters for this field, so that a workaround can be provided.

- Some fields must have English values. These are fields that have an enumerated set of expected values in the field tables.
- All files must include a header row containing column names from the tables specified in this document. Header column names must exactly match the names specified in this data specification.
- All rows should have the same number of tabs even if some fields have null values. It is important that null values are represented by empty strings and not denoted by **/N**, **NA**, **null**, or another value.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Account Data File Name Conventions

All account data files transferred to Oracle Utilities must follow a standard naming convention. Your Delivery Team will work with you to define the appropriate file names for any files that must be transferred.

File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Main Account File:

opwr_<utility>_<specification version number>_billing_account_<yyyyMMddHHmmss>.<gz or zip>

Person Address File:

opwr_<utility>_<specification version number>_billing_account_address_<yyyyMMddHHmmss>.<gz or zip>

Service Agreement Service Points File:

opwr_<utility>_<specification version number>_billing_account_sa_sp_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Account Data Version History.
- **Data Type**: The filename must contain the phrase *billing_account* to indicate the type of data it contains. As shown in the file name examples above, additional data type components may be required depending on which account file is being sent.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.

- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Naming Example

Suppose the specification version number of the document is v2-0-0 and your utility is named "The Great Energy Company." An example of your utility's file name would be as follows:

opwr_gec_v2-0-0_billing_account_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's file name would be as follows:

opwr_gec_v2-0-0_billing_account_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.v2.0.0.billing.account.20210316131415.gz

Account Data Version History

The following table lists updates to the account data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Account Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update description of `business_name` in Person Fields to more clearly explain what is required.	v2-0-9	09/05/2025
Update description of `first_name` and `last_name` in Person Fields to more clearly explain what is required. Update `status_code` in Service Agreement Fields to more clearly separate allowed values from definitions.	v2-0-8	08/26/2025
Update the description of the `service_agreement_end_date time` field in Service Agreement Fields to better advise utilities on how to complete the field. Update the Service Agreement Service Points File to add a new allowed value to the `usage_flag` field.	v2-0-7	03/17/2025

Update	Specification Version Number	Date
Update Main Account File, Person Address File, and Service Agreement Service Points File to include a cross-reference to Account Data File Name Conventions. Remove redundant file naming conventions.	v2-0-6	10/01/2024
Delete redundant copy of `account_id` column from Service Agreement Fields.	v2-0-5	04/17/2023
Update `locality` in Person Address File from VARCHAR(64) to VARCHAR(90).	v2-0-4	01/04/2023
Update Person Fields to improve accuracy of phone number and email address examples. Update Service Agreement Fields to add new columns: • `service_agreement_type_code` • `service_agreement_type_description` • `special_role_flag_code` • `special_role_flag_description` Update Person Address File to clarify that only the latest address for a `person_id` is stored, and to remove legacy columns: • `is_seasonal_address` • `seasonal_address_start_datetime` • `seasonal_address_end_datetime`	v2-0-3	08/02/2022
Update `bill_cycle_code` column in Account Fields to say it is no longer required but is preferred. Add `person_type` and `business_name` to Person Fields. Add `business_classification_code` to Service Agreement Fields. Update Account Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v2-0-2	04/11/2022

Update	Specification Version Number	Date
Update examples in Account Data File Name Conventions to use incremented major version number and current year.	v2-0-1	12/06/2021
Increment to a major version number to designate Account Data Transfer as a core data standard for utilities. Clarify that zip files should only contain one file each.	v2-0-0	11/23/2021
Update supported version number for DataConnect extracts.	v1-2-0	07/28/2021
Update `admin_area_x` fields in the Person Address File to match the same fields in the Premise File. Update `effective_datetime` in Service Agreement Rate Fields to say it cannot be empty. Update `usage_percent` field in Service Agreement Service Points File to say it can be empty.	v1-1-0	10/09/2020
Update multiple field descriptions to improve clarity. Merge phone and email field descriptions into the Person Fields topic. Update introductory topics to simplify explanations and improve cross-references. Add new information about common data formats. For a more detailed list of changes, contact your Delivery Team.	v1-0-0	05/14/2020
Initial draft of account data file specification.	v0-9-0	11/12/2019

Billing Data Transfer

The billing data files contain a range of information about a customer's billed energy use. Oracle Utilities uses this information to deliver personalized energy usage insights through printed reports, email messages, alerts, and web applications. The key fields in the files include:

- Bill Fields
- Bill Segment Fields
- Line Item Fields
- Service Quantity Fields

Not every field is required. Your Delivery Team will work with you to set up the file transfer during program launch. See Billing Data File Specifications for more information.

The billing files can be generated as a data extract using the Oracle Utilities DataConnect solution. DataConnect is available to utilities that have the following Oracle Utilities products:

- Customer Care and Billing 2.8 or higher
- Customer to Meter 2.8 or higher
- Meter Data Management 2.8 or higher

If DataConnect is available to you, your Delivery Team will work with you to leverage it for the billing file.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Billing Data Integration Process

Integration of your billing data into the Oracle Utilities Opower platform involves the following phases: sample historical file integration, full historical file integration, and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

See Data Transfer Process Overview for more information.

Billing Data File Specifications

There are four billing data files containing a range of information about a customer's billed energy use. The files are based on each of the following data entities:

- Bill Fields
- Bill Segment Fields
- Line Item Fields
- Service Quantity Fields

The order in you which you send the files to Oracle Utilities can be flexible. Not all four files necessarily need to be sent at the same time. Your Delivery Team will work with you to determine what is required for your program.

As an alternative, you can send all billing data fields in a single file. In this case, every column must be included in every row of the file, and the order of the columns must be strictly followed. This approach is not recommended. If you would like to explore it, contact your Delivery Team.

Bill Fields

The bill fields data entity includes information related to a bill's status, due date, and balance.

Field	Description
bill_id	The primary utility identifier of a bill. This is used to disambiguate bills. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary key.
account_id	The utility account associated with the bill. This identifies the customer who is responsible for the bill. Note that this field corresponds to a utility account in the Oracle Utilities Customer Care and Billing data model. See Account Fields for more information. **Type**: VARCHAR(40) **Can Be Empty?**: No.
bill_datetime	The business date that was used when the bill batch generated the bill. For example, if batch billing starts on 11:55 p.m. on 1-Jan-2018, and a bill is produced at 1:00 a.m. on 2-Jan-2018, then this bill's bill date would be 1-Jan-2018. If no offset is specified, the offset will default to the utility company's time zone. If no time component is specified, the time will be assumed to be the start of the day. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
due_datetime	The date on which the bill is due. If no offset is specified, the offset will default to the utility company's time zone. If no time component is specified, the time will be assumed to be the start of the day. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
currency_code	The ISO-4217 3-letter currency code for the bill. See https://www.currency-iso.org/en/home/tables/table-a1.html for a list of allowed values. **Example Values:** • USD • EUR • GBP **Type**: VARCHAR(3) **Can Be Empty?**: No.
current_balance	The overall account balance within the bill time window. The current balance shows how much the customer owes after the current bill. The current balance can differ from the payoff balance for budget billing service agreements, deposit service agreements, charitable contribution service agreements, or payment arrangements. For budget billing customers, the current balance is the monthly payment obligation. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: No.
payoff_balance	The outstanding balance or total amount owed for the account, after the current bill. The payoff amount contains how much the customer really owes for the "cash out amount" if the customer wanted to clear all debt. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: No
previous_period_balance	The balance from the account's last bill. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.

Field	Description
total_payments	The total amount that was paid by the customer. The total amount of frozen or canceled payment segment financial transactions linked to this bill. Frozen payments appear as negative numbers (decreasing the amount owed by the customer), while canceled payments appear as positive numbers (increasing the amount owed). **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
total_adjustments	The total amount of frozen or canceled adjustment financial transactions linked to this bill. For example, this can include late payment charges, connection charges, deposit refund, rebates, and bill write-off. Adjustments can be positive, negative, or zero. A zero amount is unusual, but necessary when you need to use an adjustment to correct the general ledger code. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
total_bill_corrections	The total amount of any cancels and rebills. Corrections can be positive, negative, or zero. A zero amount is unusual, but may be necessary when you need to correct the general ledger code. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
total_current_billing_charges	The total amount of frozen bill segment financial transactions linked to this bill. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
total_balance_current_period	The amount owed by the customer for this bill. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
bill_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Bill Segment Fields

A bill typically contains one bill segment for every active service agreement linked to its account. A bill segment is a portion of a bill that corresponds to one or more line items in a bill. Line items could be taxes, usage charges, and other fees.

It is expected that for consecutive bills, there will be no gaps between the `bill_segment_**end**_date` of the first bill segment and the `bill_segment_**start**_date` of the next bill's segment.

For example, if the first bill's segment had a `bill_segment_**end**_date` of 2019-01-01 00:00:00, then the next bill's corresponding segment would be expected to have a `bill_segment_**start**_date` of 2019-01-01 00:00:00 as well.

Field	Description
bill_id	The primary utility identifier of a bill. This is used to disambiguate bills. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note**: This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
bill_segment_id	The unique identifier of the bill segment in the source system. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note**: This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
service_agreement_id	The utility's service agreement identifier. This field corresponds to a service agreement in the Oracle Utilities Customer Care and Billing data model. See Service Agreement Fields for more detail. **Type**: VARCHAR(40) **Can Be Empty?**: No.
bill_segment_start_datetime	The start of the period of time covered by the bill segment's charges. The bill period is inclusive of this datetime value. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
bill_segment_end_datetime	The end of the period of time covered by the bill segment's charges. The bill period is exclusive of this datetime value. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
usage_period_start_datetime	The start date used by Oracle Utilities Meter Data Management to determine the correct interval data to process for the bill segment. The usage period is inclusive of this datetime value. It is typically a day behind the `bill_segment_start_datetime` (but could also be the same or different by more days, depending on your particular setup). **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
usage_period_end_datetime	The end date used by Oracle Utilities Meter Data Management to determine the correct interval data to process for the bill segment. The usage period is exclusive of this datetime value. It is typically a day behind the `bill_segment_end_datetime` (but could also be the same or different by more days, depending on your particular setup). **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
estimated	A Boolean value to indicate whether or not this value was estimated. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.

Field	Description
closing	A Boolean value to indicate whether this is the final bill segment for the service agreement (that is, the service agreement has been stopped and this is the final bill segment). **Allowed Values**: • Y Value indicating the final bill segment. • N (Default) Value indicating that this is not the final bill segment. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.
bill_segment_status	The status of the bill segment. Your Delivery Team will work with you during the launch process to confirm the status values to use. **Example Values**: • Incomplete Bill segments are initially created in the Incomplete state since they do not have bill lines. • Freezable After successful generation of a bill segment, the status of the bill segment becomes Freezable. • Frozen A bill segment may now appear on customer's bill and cannot be changed. • Pending Cancel A new financial transaction was generated to correct the bill segment. • Canceled A new financial transaction was generated to correct the bill segment. **Note**: The `Incomplete`, `Freezable`, and `Pending Cancel` statuses are not expected to appear in Oracle Utilities DataConnect extracts. The DataConnect tool only extracts bills that are complete. **Type**: VARCHAR(20) **Can Be Empty?**: No.
bill_cycle_code	The bill cycle of the bill segment's bill. This information is only populated on bills when they are generated on-schedule by the batch billing background process. **Example Values**: M01,M02 **Type**: VARCHAR(20) **Can Be Empty?**: Yes, but preferred to not be empty.
bill_cycle_description	A description of the bill cycle. **Example Values**: Monthly bill cycle - day 1, Monthly bill cycle - day 2 **Type**: VARCHAR(255) **Can Be Empty?**: Yes.
current_amount	The aggregate amount owed on a service agreement for the bill period. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: No.

Field	Description
outstanding_amount	The aggregate total outstanding amount for the service agreement. This is likely the same as the current amount unless there is a budget billing arrangement. Note that budget billing arrangements are made per Service Agreement, and then rolled up to the bill/account level. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: No.
master_bill_segment_id	The identifier of the master service agreement's bill segment. The `master_bill_segment_id` must reference another `bill_segment_id` that was previously sent, or is part of the same file that this bill segment was sent in. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
bill_segment_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Line Item Fields

The line item fields include data related to the specific lines and charges that appear on a customer's bill.

Field	Description
bill_id	The primary utility identifier of a bill. This is used to disambiguate bills. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note**: This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
bill_segment_id	The unique identifier of the bill segment in the source system. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note**: This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
calculation_header_sequence_number	The identifier for the calculation header within the bill segment. This identifier will typically be 1 unless multiple versions of the rate were in effect during the bill period. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
bill_line_sequence	A system-assigned unique identifier of the bill line item. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
calculated_amount	The calculated amount associated with the bill line item. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: No.

Field	Description
description_on_bill	The information about the bill line that appears on the customer's bill. **Type**: VARCHAR(255) **Can Be Empty?**: No.
print	A flag that indicates whether information about this line will print on the customer's bill. **Allowed Values**: • Y Display information on the customer's bill. • N Do not display information on the customer's bill. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes. When empty, this defaults to N.
bill_line_item_type	The category of the line item charge, such as a tax, fee, distribution charge, transmission charge, etc. The categories are defined by the utility. Oracle Utilities Opower will map them to internal categories. **Type**: VARCHAR(30) **Can Be Empty?**: Yes, but preferred to not be empty.
bill_line_item_unit_of_measure	The unit of measure (UOM) of the line item. There is also a UOM at the segment read for cases when a measured quantity is different than billed quantity. For example, gas prices are expressed in an amount per therm, but gas meters are expressed in volume. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
bill_line_item_time_of_use	The time-of-use code of the service quantity priced on the calculation line. There will be one bill line item per TOU period. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
bill_line_item_service_quantity_identifier	The service quantity priced on the calculation line. This value is used to further distinguish between measured quantities that have identical UOM/TOU combinations. For example, this value could distinguish between On-Peak KWH generation and On-Peak KWH consumption for solar customers. **Type**: VARCHAR(255) **Can Be Empty?**: Yes.
billable_service_quantity	The service quantity priced on the calculation line. For example, On-Peak KWH is 159 kWh. **Type**: DECIMAL(18,6). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
base_amount	A value used by calculation lines (such as taxes) that are dependent on other values (such as volumetric charges). The Base Amount shows the total amount derived from the cross-referenced line(s) that the current line then used to calculate its billed amount. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.

Field	Description
exempt_amount	The amount of the calculated charge that the customer doesn't have to pay because they are tax exempt. **Type**: DECIMAL(15,2). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
measures_peak_quantity	A flag that indicates if the UOM priced on the calculation line is used to measure a peak quantity. **Allowed Values**: • Y Line measures peak quantity. • N (Default) Line does not measure peak quantity. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.
bill_line_item_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Service Quantity Fields

For service agreements that charge for a commodity, the Service Quantity data entity contains information about the consumption that will be priced by the service agreement rate. Service Quantity is populated based on usage requests that are sent during billing to Oracle Utilities Meter Data Management. One row exists for every unique combination of unit of measure (UOM), time-of-use (TOU) code, and service quantity identifier (SQI) associated with the service agreement.

> **Note**
>
> The `bill_segment_id`, `bill_service_quantity_unit_of_measure`, `bill_service_quantity_time_of_use`, and `service_quantity_identifer` fields below are meant to be used together to form a unique key. The first field is required, and the other fields can be empty. Your Delivery Team will work with you to ensure that these fields are used correctly.

Field	Description
bill_id	The primary utility identifier of a bill. This is used to disambiguate bills. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note** This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
bill_segment_id	The unique identifier of the bill segment in the source system. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key. **Note** This field is not required if you are sending your billing data in a single file to Oracle Utilities. See Billing Data File Specifications for more information.
bill_service_quantity_unit_of_measure	The unit of measure of the service quantity. **Type**: VARCHAR(30) **Can Be Empty?**: No. Primary Key.
bill_service_quantity_time_of_use	The time of use (TOU) of the service quantity. An identifier of the TOU used in a rate plan. **Example Value**: WEEKEND_SUMMER **Type**: VARCHAR(30) **Can Be Empty?**: No. Primary Key.
service_quantity_identifier	The service quantity identifier of the service quantity. **Example Values:** • CONSUMED • DAYS • DELIVERED • GEN • NET_USAGE • RECEIVED **Type**: VARCHAR(30) **Can Be Empty?**: No. Primary Key.
start_read_datetime	The start time of the first read in the group. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.
end_read_datetime	The end time of the last read in the group. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.

Field	Description
billed_service_quantity	The service quantity that will be priced by the service agreement's rate. **Type**: DECIMAL(18,6). See Supported Decimal Formats for details. **Can Be Empty?**: No.
bill_service_quantity_updated_at	The date at which the record was updated in the source system. If left blank, the field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Billing Data Order and File Format

Billing data must be sent in a particular order and file format.

Data File Order

The Oracle Utilities Opower platform processes files according to the order received, then dated. If we receive multiple files on the SFTP server in the same time frame (as could be done for full historical data files), they will be automatically queued by their filename date stamps.

For example, if a file is received on the SFTP server named *opwr_util_20140520.txt* and then receive one named *opwr_util_20140519.txt* an hour later, *opwr_util_20140520.txt* will be processed first despite the fact that its date stamp is later than the *opwr_util_20140519.txt*. However, if these files are received at the same time, the file name data stamp will take precedence and *opwr_util_0140519.txt* will be processed before *opwr_util_20140520.txt*.

To ensure that all files are queued in the correct order, files should be named so that the date stamps are consistent with the intended load order of the files.

Data Order Matters!

Data (rows) for each service point should always be arranged from the oldest to the most recent to ensure that your data is processed in the correct order.

- **Multiple Usage Reads for a Single Service Point**: Any time a single file contains more than one usage read for a single service point covering the same period of time, the most recent read needs to appear last in the file. If the usage periods for those two reads are the same, Oracle Utilities considers the last record read to be a correction.
- **Multiple Records for the Same Entity**: If there are multiple records for the same entity (such as customer, account, or service point), in the same file, only the last record for that entity is loaded. Oracle Utilities assumes that the last record represents the most current data from the utility.
- **Customer Information**: The data in the last row of the last file processed is always the active data in the Oracle Utilities database for that customer.

File Format

All data files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- All data must be sent as tab-delimited text.

- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- Some fields must have English values. These are fields that have an enumerated set of expected values in the field tables.
- All files must include a header row containing column names from the tables specified in this document. Header column names must exactly match the names specified in this data specification.
- All rows should have the same number of tabs even if some fields have null values. It is important that null values are represented by empty strings and not denoted by **/N**, **NA**, **null**, or another value.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Billing Data File Name Conventions

All billing data files transferred to Oracle Utilities must follow a standard naming convention. The naming convention is shown below, and is followed by concrete examples. It is strongly recommended that you also send the Fileset Transfer Manifest Specification so that Oracle Utilities can validate that all expected files have been received intact.

File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

opwr_<utility>_<specification version number>_<billing data type>_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Billing Data Version History.
- **Billing Data Type**: The file name must indicate the type of billing data being sent in the file. Use one of the following designations depending on which file you are sending.
- *billing_bill*: Indicates that the file contains data related to the bill fields entity.
- *billing_billsegment*: Indicates that the file contains data related to the bill segments entity.
- *billing_lineitem*: Indicates that the file contains data related to the line items entity.
- *billing_servicequantity*: Indicates that the file contains data related to service quantity entity.

- *billing*: Indicates that the file contains all bill-related data in a single file. Note: This name should only be used if you are sending your billing data in a single as opposed to four separate files. This approach is not recommended. Contact your Delivery Team if you need to discuss this approach.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Naming Examples

Suppose the specification version number of the document is v2-0-0 and your utility is named "The Great Energy Company." Examples of your billing file names are shown below.

Utility Sending Separate Files (Default)

- File with *bill* fields:

  opwr_gec_v2-0-0_billing_bill_20210316131415.gz
- File with *bill segment* fields:

  opwr_gec_v2-0-0_billing_billsegment_20210316131415.gz
- File with *line item* fields:

  opwr_gec_v2-0-0_billing_lineitem_20210316131415.gz
- File with *service quantity* fields:

  opwr_gec_v2-0-0_billing_servicequantity_20210316131415.gz

Utility Sending Single File (Not Recommended)

opwr_gec_v2-0-0_billing_20210316131415.gz

Utility Using PGP Encryption

If your utility is using PGP-encrypted files, then an example of your utility's file name for the Bill Fields entity would be as follows:

opwr_gec_v2-0-0_billing_bill_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.v2.0.0.billing.20210316131415.gz

Billing Data Version History

The following table lists updates to the billing data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Billing Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update `service_quantity_identifier` column in Service Quantity Fields to include NET_USAGE as an example value.	v2-0-3	06/27/2025
Update `bill_cycle_code` column in Bill Segment Fields to say it is no longer required but is preferred. Remove `premise_id` from Bill Segment Fields. Update `bill_line_item_type` in Bill Line Item Fields to say it is no longer required but is preferred. Delete note about `premise_id` from File Format section. Update Billing Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v2-0-2	04/11/2022
Update examples in Billing Data File Name Conventions to use incremented major version number and current year.	v2-0-1	12/06/2021
Increment to a major version number to designate Billing Data Transfer as a core data standard for utilities. Update other sections to reflect that utilities can send four separate billing data files, rather than a single file. (For example, the Billing Data File Name Conventions section includes new file names for the four files.) Delete appears_in_summary from Line Item Fields, as it is no longer supported. Clarify that zip files should only contain one file each.	v2-0-0	11/23/2021
Update definition of `total_bill_corrections` in Bill Fields. Add `appears_in_summary` back to Line Item Fields.	v1-4-0	09/14/2021
Update supported version number for DataConnect extracts.	v1-3-0	07/28/2021
Update the definition of bill segments in Bill Segment Fields. Remove `appears_in_summary` from Line Item Fields.	v1-2-0	01/25/2021

Update	Specification Version Number	Date
Update `service_quantity_identifier` description in Service Quantity Fields to show additional example values.	v1-1-0	10/09/2020
Update multiple field descriptions to improve clarity. Update introductory topics to simplify explanations and improve cross-references. Add new information about common data formats. For a more detailed list of changes, contact your Delivery Team.	v1-0-0	05/14/2020
Initial draft of billing data file specification.	v0-9-0	11/12/2019

Premise Data Transfer

The premise data file identifies a utility customer's physical, unchanging location. This file is required for utilities that need to display water data or commercial and industrial data in the Oracle Utilities Opower Data Browser. The key fields in the file include:

- Premise Fields
- Service Point Fields
- Device Installation Fields
- Device Fields
- Measuring Component Fields

Your Delivery Team will work with you to determine if the file is required for your program, and to help you construct the file during program launch. See Premise Data File Specifications for more information.

The premise file can be generated as a data extract using the Oracle Utilities DataConnect solution. DataConnect is available for utilities that have the following Oracle Utilities products:

- Customer Care and Billing 2.8 or higher
- Customer to Meter 2.8 or higher
- Meter Data Management 2.8 or higher

If DataConnect is available to you, your Delivery Team will work with you to leverage it for the premise file.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Premise Data Integration Process

Integration of your premise data into the Oracle Utilities Opower platform involves the following phases: sample historical file integration, full historical file integration, and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

See Data Transfer Process Overview for more information.

Premise Data File Specifications

The premise data file identifies a utility customer's physical, unchanging location. It is organized into a hierarchy of data entities, with data about the premise at the root of the hierarchy. The data in these entities allows Oracle Utilities to provide the right premise information for the right fuel type to the right customer in the right location.

- Premise Fields
- Service Point Fields
- Device Installation Fields
- Device Fields
- Measuring Component Fields

Not all of the data in each data entity is required. Your Delivery Team will work with you to determine what is required for your program.

Note: Although each entity is described separately, any data elements that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file.

Premise Fields

The premise entity includes fields related to the qualities of a physical location, such as the premise type, mailing address, country, and time zone.

Note: Although the fields below are described separately from the other fields in the premise data file, any premise data that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file. See Premise Data File Specifications for more information.

Field	Description
premise_id	Premise identifier in a utility's Customer Information System or other system of record. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
premise_name	The business-oriented name of the premise from a utility company's perspective. **Type**: VARCHAR(40) **Can Be Empty?**: Yes. Business Key.

Field	Description
external_premise_id	The premise identifier from an external system. For example, this may be a premise identifier from a system other than Oracle Utilities Customer to Meter solution or the Oracle Utilities Customer Care and Billing solution. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
parent_premise_id	The unique identifier of the parent premise. This is typical for buildings that include multiple residences, such as apartments and condos. **Type**: VARCHAR(40) **Can Be Empty?**: Yes
premise_type_code	A code that represents the type of premise. A standard for this value is not enforced. Any non-standard value provided will be handled with a mapping during transformation to the canonical model. Source values from the utility's system should be consistent. For example, values like town home and bungalow may be mapped to the value for single family homes. To map source values accurately, Oracle Utilities Opower must be assured that the source values are part of a standardized, predefined list rather than a set of free-form text entries. **Example Values**: SFR (single-family residence), MFR (multi-family residence) **Type**: VARCHAR(8) **Can Be Empty?**: Yes.
premise_type_description	A short description of the premise. **Example Values**: Single Family, Multi Family, Hospital, Apartment **Type**: VARCHAR(100) **Can Be Empty?**: Yes.
address_1	Address line 1 of the premise. **Note**: At least one address line must be populated. Other address lines are optional. **Type**: VARCHAR(254) **Can Be Empty?**: No.
address_2	Address line 2 of the premise. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
address_3	Address line 3 of the premise. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
address_4	Address line 4 of the premise. **Type**: VARCHAR(254) **Can Be Empty?**: Yes.
locality	The locality of the premise. This is typically a city in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: No.
postal_code	The postal code of the premise. **Type**: VARCHAR(12) **Can Be Empty?**: No.

Field	Description
admin_area_1	A first-order civil entity below the country level, such as states in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: Yes.
admin_area_2	A second-order civil entity below the country level, such as counties in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: Yes.
admin_area_3	A third-order civil entity below the country level, such as suburbs and neighborhoods in the United States. **Type**: VARCHAR(90) **Can Be Empty?**: Yes.
country	The two-character code representing the country of the premise. For a full list of country codes, see Wikipedia - Officially Assigned Code Elements. **Type**: VARCHAR(2) **Can Be Empty?**: Yes.
division	The distinct business entities or jurisdictions of the premise in a utility's Customer Information System or other system of record. **Type**: VARCHAR(3) **Can Be Empty?**: Yes.
premise_time_zone	The timezone of the premise. The value provided must conform to the specification in the Internet Assigned Numbers Authority database. **Type**: VARCHAR(32) **Can Be Empty?**: No.
latitude	The latitude of the service address, expressed as a decimal number of degrees between -90 and 90, with no directional indicators. Any latitude data should have three-digit precision or better. If it is less specific than this, leave this field null. If no latitude value is available, leave this field empty instead of supplying a value of 0. **Type**: DECIMAL(9,7). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
longitude	The longitude of the service address, expressed as a decimal number of degrees between -180 and 180, with no directional indicators. Any longitude data should be rooftop or better. If it is less specific than rooftop, leave this field null. If no longitude value is available, leave this empty instead of supplying a value of 0. **Type**: DECIMAL(10,7). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
landlord_id	The unique account ID of the landlord agreement that covers the premise, if any. This value must match the value of the `account_id` field listed in the Account Fields section of the Account File. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.

Field	Description
premise_updated_at	The date on which the record was updated in the Customer Information System or other system of record. If left blank, this field does not store any data. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Service Point Fields

The service point entity includes fields related to the device location where service is delivered.

Note: Although the fields below are described separately from the other fields in the premise data file, any premise data that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file. See Premise Data File Specifications for more information.

Field	Description
service_point_id	The unique service point identifier in a utility's Customer Information System or other system of record. A service point identifies an unchanging device location where service is delivered. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
service_point_name	The name of the service point as displayed in the utility Customer Information System or other system of record, if applicable. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
service_point_type_description	A short description of the service point code. **Example Values**: Electric Commercial, Gas Residential, Water Industrial **Type**: VARCHAR(100) **Can Be Empty?**: Yes.
service_point_type_category	A flag that indicates the type of device installed at service points of this type. **Allowed Values**: • D1MT Indicates that a single meter can be installed at service points of this type. • D1IT Indicates that a single "badged" item can be installed at service points of this type. A badged item refers to objects like street lamps that have a unique ID. • D1MI Indicates that one or more "unbadged" items can be installed at service points of this type. **Type**: VARCHAR(4) **Can Be Empty?**: No.

Field	Description
service_point_type_code	The service type provided by service points of this type. **Allowed Values**: • E Electric • G Gas • W Water • M Miscellaneous • WW Waste water • O Other **Type**: VARCHAR(2) **Can Be Empty?**: No.
service_point_start_datetime	The date on which the service agreement was associated with the service point. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes. **Note**: This field should only be populated if it will be used in the Oracle Utilities DataRaker solution. Otherwise it will be ignored.
service_point_end_datetime	The date on which the service agreement was disassociated from the service point. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes. **Note**: This field should only be populated if it will be used in the Oracle Utilities DataRaker solution. Otherwise it will be ignored.
service_cycle_code	The service point's measurement cycle. A measurement cycle can define the schedule for manual meter reading of devices at service points, or it can define when to calculate bill determinants for usage subscriptions associated to service points in the cycle. **Example Values**: CYCLE01, CYCLE02 **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
service_cycle_description	The description for the service cycle code. **Example Values**: Read Cycle 01, Read Cycle 02 **Type**: VARCHAR(300) **Can Be Empty?**: Yes.
service_route_code	The route used to collect measurements. **Example Values**: ROUTE-100, ROUTE-110 **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
service_route_description	The description for the route code. **Example Values**: Route 100, Route 110 **Type**: VARCHAR(100) **Can Be Empty?**: Yes.

Field	Description
facility_level_1	The substation network node level closest to the service point. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
facility_level_2	The feeder or distribution circuits that carry power from the substations to the transformers closest to the service point. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
facility_level_3	The step down transformer closest to the premise. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
life_support_sensitive_load_code	A Boolean value that identifies whether the premise has life support or sensitive load equipment. **Allowed Values**: • Y Premise has life support equipment. • N Premise does not have life support equipment. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes. If left empty, this field defaults to N.
status_code	The current status of the service point. **Allowed Values**: ACTIVE, INACTIVE **Type**: VARCHAR(12) **Can Be Empty?**: Yes.
source_status_code	A status code that identifies if the service point is connected or disconnected to the distribution network. **Example Values**: • D1CN Service point is connected to the distribution network. • D1DC Service point is disconnected from the distribution network. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
disconnect_location	A code that identifies whether the service is disconnected at the device or at a source such as a service pole. **Example Values**: • D1DV Service is disconnected at the device. • D1SR Service is disconnected at another source. **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
remote_disconnect	A Boolean value to indicate whether to remotely disconnect a customer's service point from receiving service. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes. If left empty, this field defaults to FALSE.

Field	Description
service_point_time_zone	The timezone of the service point. The value provided must conform to the specification in the Internet Assigned Numbers Authority database. **Type**: VARCHAR(32) **Can Be Empty?**: No.
disconnect_for_nonpay	A Boolean value to indicate whether to stop providing service to a service point due to the customer not paying bills. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.

Device Installation Fields

The device installation entity represents the installation lifecycle of a device at a service point. The lifecycle reflects a business process consisting of commands to commission, connect, disconnect, and decommission the device. Rules for this process include:

1. The `removal_date` field must be null or must be strictly later than `install_date`.
2. Once specified, fields about the device cannot be changed other than setting the `removal_date` to indicate the device is no longer in service. This ensures that any fields on the device used for calculating values displayed to a customer do not impact historically displayed values. If a change is needed, the existing device should be taken out of service and a new device with the appropriate values should be created in its place.
3. Device installation and removal periods may not overlap for a service point.

> **Note**
>
> Although the fields below are described separately from the other fields in the premise data file, any premise data that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file. See Premise Data File Specifications for more information.

Field	Description
install_event_id	The unique ID of the install event. **Type**: VARCHAR(80) **Can Be Empty?**: No. Primary Key.
device_installation_external_id	The ID used by the external system to identify the install event. **Type**: VARCHAR(60) **Can Be Empty?**: Yes.

Field	Description
device_installation_status	The installation lifecycle stage of a device at a service point. The lifecycle reflects a business process consisting of commands to commission, connect, disconnect, and decommission the meter. **Example Values**: • Pending • Connected / Pre-Commission • Pre-Connected / Commissioned • Connected / Commissioned • Connected / Decommissioned • Disconnected / Commissioned • Disconnected/ Decommissioned • Remove **Type**: VARCHAR(40) **Can Be Empty?**: No.
arming_status	A Boolean value that indicates if the device has been armed or not. Some AMI devices have special arming switches that require manual manipulation before the commodity can flow through the device. Devices that require arming (those with an Arming required flag set to Arming Required) must be armed before they can be connected. **Allowed Values**: • Any supported Boolean values. • Armed (Default) The device is armed and ready to be connected. • Not Armed The device is not yet armed. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.
device_on_off_status	A Boolean value indicating the status of the device. **Allowed Values**: • Any supported Boolean values • D1ON The device is on. • D1OF The device is off. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: No.
installation_constant	A value other than 1 used to indicate that when calculating consumption, the installation requires that measurement data be multiplied by this value to get accurate results. This installation constant is used for all device measuring components. The application includes two multipliers: this installation constant and a measuring component register multiplier. These multipliers can be used individually or together to adjust measurements. **Type**: DECIMAL(12,6). See Supported Decimal Formats for details. **Can Be Empty?**: No.
install_datetime	The date and time of the installation. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: No.

Field	Description
removal_datetime	The date and time of the installation. Leave null to indicate that the meter is in service. **Note**: There can be only one active device at a service point. The `removal_datetime` field should be populated for any device that is not active. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Device Fields

The device entity includes fields related to the name, type, and description of a device.

> **Note**
>
> Although the fields below are described separately from the other fields in the premise data file, any premise data that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file. See Premise Data File Specifications for more information.

Field	Description
device_id	The device or meter identifier in the source system. **Type**: VARCHAR(40) **Can Be Empty?**: No. Primary Key.
device_name	The device name as displayed to a user in the source system. **Type**: VARCHAR(40) **Can Be Empty?**: Yes.
device_external_id	The device identifier from an external system. If your utility has the Oracle Utilities DataConnect solution, this field can be used to map the external ID from the head-end system to the `external_id` field. The source in your meter data management system could be a number of fields, including `badge_number`, `serial_number`, or `internal_meter_number`. Consult the administrator of your meter data management system for more information. **Type**: VARCHAR(120) **Can Be Empty?**: Yes.
device_type_code	The type of device. Typically, devices are categorized into smart meters, manual meters, items, or communications components. The code used to denote the device type will vary by utility. **Example Value**: E-SMART-MTR **Type**: VARCHAR(30) **Can Be Empty?**: No.
device_type_description	A description of the device type code. **Example Value**: Electric Smart Meter **Type**: VARCHAR(100) **Can Be Empty?**: Yes.

Field	Description
device_type_service_type_code	The service type (electric, gas, water, etc.) provided by devices of this type. **Allowed Values**: • E Electric • G Gas • W Water • M Miscellaneous • O Other • WW Waste water **Type**: VARCHAR(2) **Can Be Empty?**: No.
manufacturer	The short name or acronym of the manufacturer of the device. **Example Values**: GE, LG **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
manufacturer_desc	The full name of the meter manufacturer. **Example Values**: General Electric, Landis+Gyr **Type**: VARCHAR(100) **Can Be Empty?**: Yes.
model_number	The specific model number of the device provided by the manufacturer. **Example Values**: I-210+c **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
model_desc	The full name of the device model. **Example Values**: I-210+c Smart Meter **Type**: VARCHAR(100) **Can Be Empty?**: Yes.
device_manufacture_date	The date on which the device was manufactured. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
dial_count	The number of dials in a meter. **Type**: DECIMAL(5,0) **Can Be Empty?**: Yes.
sub_meter_flag	A Boolean value to indicate whether a device is a sub meter as opposed to a separate, independent meter. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes. When left empty, the default value is FALSE.

Field	Description
net_meter_flag	A Boolean value to indicate whether a device uses net metering. This is typical for customers who generate energy using solar technology in addition to consuming energy. **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: No.
device_status	The current status of the meter. **Note**: There can be only one active device at a service point. **Allowed Values**: The lifecycle status values configured for a meter in the Oracle Utilities Meter Data Management system. **Example Values**: ACTIVE, RETIRED **Type**: VARCHAR(12) **Can Be Empty?**: No.
head_end_system	The head-end system associated with the device. Head-end systems are systems that collect measurement data and meter events, and send command requests to meters. **Example Values**: Echelon, Itron, Sensus **Type**: VARCHAR(30) **Can Be Empty?**: Yes.
head_end_registration_status	A flag that indicates the registration status of the meter in the head-end system. This is used by smart meter commands and other processes to determine if the meter is currently registered in the head-end system. **Example Values**: D1DR, D1NR, D1RE • D1NR (Default) Not Registered: The meter is not registered in the head-end system. • D1RE Registered: The meter is registered in the head-end system. • D1DR Deregistered: The meter has been deregistered from the head-end system. **Type**: VARCHAR(4) **Can Be Empty?**: Yes.
device_classification	A flag that indicates if devices of this type are processed as meters or items. **Example Values**: D1IT, D1MT • D1IT The device is processed as an item. • D1MT The device is processed as a meter. **Type**: VARCHAR(4) **Can Be Empty?**: Yes.

Measuring Component Fields

Measuring components are single points for which data is received and stored in the utility's meter data management system. A measuring component can be associated with a device, which can have one or more measuring components. The measuring component entity includes fields related to channel, class, units of measure, and time of use.

> **Note**
>
> Although the fields below are described separately from the other fields in the premise data file, any premise data that you send to Oracle Utilities must be included in the same file, with all columns included in every row of the file. See Premise Data File Specifications for more information.

Field	Description
measuring_component_id	The measuring component or channel identifier in a utility's Customer Information System or other system of record. It should be a value that allows identification of a particular channel or register. **Note**: This field is not a unique ID. The combination of this field with the `device_id` field should be unique. See Device Fields for more information about the `device_id`. It also acceptable to use either the `measuring_component_type_code` or `channel_id` fields below to serve as the `measuring_component_id`. **Type**: VARCHAR(40) **Can Be Empty?**: No.
measuring_component_external_id	The external identifier of the channel, used in external systems such as head-end systems. **Type**: VARCHAR(60) **Can Be Empty?**: Yes.
channel_id	The identifier of the channel for interval data. If scalar data is being used, then this field should contain the identifier of the register. **Type**: VARCHAR(120) **Can Be Empty?**: Yes.
number_of_digits_left	The number of digits to the left of the decimal point that the reading can contain. **Type**: DECIMAL(5,0). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
number_of_digits_right	The number of digits to the right of the decimal point that the reading can contain. **Type**: DECIMAL(5,0). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
channel_multiplier	The factor used for multiplying the measurement obtained from the measuring component. **Type**: DECIMAL(12,6). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
measuring_component_type_code	The code for the measuring component type. For records that are null and the type of measuring component is unknown, use the value `UNK` to signify that it is unknown. **Example Value**: E-RES-KWH-60 **Type**: VARCHAR(30) **Can Be Empty?**: No.
measuring_component_type_description	The description of the measuring component type. **Example Value**: Electric kWh 60min **Type**: VARCHAR(100) **Can Be Empty?**: Yes.

Field	Description
measuring_component_class	The class of the measuring component. **Allowed Values**: • D1AG Aggregator • D1PH Physical • D1SP Scratchpad • D1SR Standalone Data Request • D1SS Statistics • D1ST Standalone **Type**: VARCHAR(4) **Can Be Empty?**: Yes.
interval_size	The size of the intervals measured by the measuring component. This is represented as a number of seconds, such as 3600 for hourly intervals. If the meter is scalar, use a value of 0. **Allowed Values**: • 0 • 300 • 900 • 1800 • 3600 • 86400 **Type**: DECIMAL(5,0). See Supported Decimal Formats for details. **Can Be Empty?**: No.
interval_scalar_flag	A flag that identifies if the attribute is interval or scalar. If the meter is not interval, then this field should be populated with SCALAR. **Allowed Values**: • INTERVAL • SCALAR • D1IN • D1SC **Type**: VARCHAR(30) **Can Be Empty?**: No.
allows_negative_consumption	A flag that indicates if measuring components of this type can record negative consumption, such as in the case of two-way meters. **Allowed Values**: True, False **Type**: BOOLEAN. See Supported Boolean Formats for more information. **Can Be Empty?**: Yes.

Field	Description
unit_of_measure	The unit of measure for the quantity being recorded. **Allowed Values**: • A Amps • C Celsius • CCF Hundred Cubic Feet • CF Cubic Feet • CGL Hundred Gallons • CT Count • F Fahrenheit • GAL Gallons • KL Kilolitres • KVA Kilovolt-Ampere • KVAH Kilovolt-Ampere Hours • KVAR Kilovolt-Ampere Reactive • KVARH Kilovolt-Ampere Reactive Hours • KW Kilowatt • KWH Kilowatt-Hours • L Litres • M3 Cubic Metre • MCF Thousand Cubic Feet • MGL Mega Gallons • ML Megalitres • MWH Megawatt-Hours • PERC Percentage • PF Power Factor • TH

Field	Description
	Thermal Units • V Volts • VA Volt-Ampere • VAH Volt-Ampere Hours • VAR Volt-Ampere Reactive • VARH Volt-Ampere Reactive Hours • W Watt • WH Watt-Hours **Type**: VARCHAR(30) **Can Be Empty?**: No.
time_of_use	The modifier for a given unit of measure that indicates a period of time during which a quantity has been used, such as Critical Peak or On-Peak. **Example Values**: • On-Peak The time when the greatest quantity of some consumable is being used. • Off-Peak The time when the least amount of some consumable is being used. **Type**: VARCHAR(12) **Can Be Empty?**: Yes.

Field	Description
service_quantity_identifier	An identifier for distinguishing between measured quantities that have identical unit of measure (UOM) and time of use (TOU) combinations. **Example Values**: • ADU Average Daily Usage • COIN Coincident • CONSUMED Consumed • DELIVERED Delivered • GEN Generation • NET_USAGE Net Usage • OFFDMD Contract Off Peak Demand • RECEIVED Received **Type**: VARCHAR(30) **Can Be Empty?**: Yes.

Premise Data Order and File Format

Premise data must be sent in a particular order and file format.

Data File Order

The Oracle Utilities Opower platform processes files according to the order received, then dated. If we receive multiple files on the SFTP server in the same time frame (as could be done for full historical data files), they will be automatically queued by their file name date stamps.

For example, if a file is received on the SFTP server named *opwr_util_20140520.txt* and then receive one named *opwr_util_20140519.txt* an hour later, *opwr_util_20140520.txt* will be processed first despite the fact that its date stamp is later than the *opwr_util_20140519.txt*. However, if these files are received at the same time, the file name data stamp will take precedence and *opwr_util_0140519.txt* will be processed before *opwr_util_20140520.txt*.

To ensure that all files are queued in the correct order, files should be named so that the date stamps are consistent with the intended load order of the files.

Data Order Matters!

Data (rows) for each service point should always be arranged from the oldest to the most recent to ensure that your data is processed in the correct order.

- **Multiple Usage Reads for a Single Service Point**: Any time a single file contains more than one usage read for a single service point covering the same period of time, the most recent read needs to appear last in the file. If the usage periods for those two reads are the same, Oracle Utilities considers the last record read to be a correction.

- **Multiple Records for the Same Entity**: If there are multiple records for the same entity (such as customer, account, or service point), in the same file, only the last record for that entity is loaded. Oracle Utilities assumes that the last record represents the most current data from the utility.
- **Customer Information**: The data in the last row of the last file processed is always the active data in the Oracle Utilities database for that customer.

File Format

All premise data files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- All data must be sent as tab-delimited text.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported. **Note**: The only field that does not yet support non-Latin characters is `premise_id`. Contact your Delivery Team if you need to send non-Latin characters for this field, so that a workaround can be provided.
- Some fields must have English values. These are fields that have an enumerated set of expected values in the field tables.
- All files must include a header row containing column names from the tables specified in this document. Header column names must exactly match the names specified in this data specification.
- All rows should have the same number of tabs even if some fields have null values. It is important that null values are represented by empty strings and not denoted by **/N**, **NA**, **null**, or another value.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Premise Data File Name Conventions

All premise data files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

opwr_<utility>_<specification version number>_premise_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Premise Data Version History.

- **Data Type**: The filename must contain the word *premise* to indicate the type of data it contains.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Naming Example

Suppose the specification version number of the document is v2-0-0 and your utility is named "The Great Energy Company." An example of your utility's file name would be as follows:

opwr_gec_v2-0-0_premise_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's file name would be as follows:

opwr_gec_v2-0-0_premise_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.v2.0.0.premise.20210316131415.gz

Premise Data Version History

The following table lists updates to the premise data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Premise Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update `status_code` allowed values in Service Point Fields to be all caps. Update `device_status` allowed values in Device Fields to be all caps.	v2-0-7	08/26/2025
Remove `service_agreement_id` from Service Point Fields since it is a redundant column. Add `300` as an allowed value in `interval_size` in Measure Component Fields.	v2-0-6	08/07/2025

Update	Specification Version Number	Date
Update `service_quantity_identifier` column in Measure Component Fields to include NET_USAGE as an example value.	v2-0-5	06/27/2025
Remove `register_id` reference from Measuring Component Fields. This column has been replaced by other columns in the Interval Data Transfer specification, and so it no longer applies.	v2-0-4	07/20/2023
Update `service_point_type_code` column description in Service Point Fields to specify VARCHAR(2) instead of VARCHAR(30). Update `interval_size` column description in Measuring Component Fields to specify that the column cannot be empty.	v2-0-3	08/01/2022
Expand the list of allowable values for the `unit_of_measure` field in Measuring Components. Update Premise Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v2-0-2	04/11/2022
Update examples in Premise Data File Name Conventions to use incremented major version number and current year.	v2-0-1	12/06/2021
Increment to a major version number to designate Premise Data Transfer as a core data standard for utilities. Clarify that zip files should only contain one file each.	v2-0-0	11/23/2021
Update supported version number for DataConnect extracts.	v1-2-0	07/28/2021

Update	Specification Version Number	Date
Update `measuring_component_id` description in Measuring Component Fields to explain how it maps to another field in the Oracle Utilities interval data file. Rename `updated_at` to `premise_updated_at` in Premise Fields. Update `install_event_id` in Device Installation Fields to accept VARCHAR(80). A variety of additional updates were made to improve clarity and meaning. Contact your Delivery Team for a more detailed list of changes.	v1-1-0	10/09/2020
Update multiple field descriptions to improve clarity. Update introductory topics to simplify explanations and improve cross-references. Add new information about common data formats. For a more detailed list of changes, contact your Delivery Team.	v1-0-0	05/14/2020
Initial draft of premise data file specification.	v0-9-0	11/12/2019

Legacy Billing Data Transfer

Oracle Utilities delivers personalized energy usage insights through printed reports, email messages, alerts, and web applications. In order to provide these insights, Oracle Utilities needs a regular feed of billed usage data for your customers.

> **Note**
>
> This is a legacy data transfer specification. You may need to use the new specification instead. Your Delivery Team will work with you to determine which is applicable to your situation.

> **Note**
>
> Terms used by Oracle Utilities to describe data fields might differ in meaning from your internal system's terminology. Data fields are defined in this documentation to facilitate communication by ensuring use of common definitions. In order to avoid delays in launching your program, it is important that you read this content carefully and quickly contact your Oracle Utilities Delivery Team if you have any questions.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Legacy Billing Data Integration Process

Integration of your data into the Oracle Utilities Opower platform involves the following phases: sample historical file integration, full historical file integration, and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

Oracle Utilities recommends that you extract data from your production billing system. This ensures that the customer contact information and energy usage data you transfer to Oracle Utilities is as accurate and current as possible.

> **Warning**
>
> We strongly recommend that you keep a searchable copy of the sample historical, full historical, and incremental files. These files will make it easier for you and your Delivery Team to work together to diagnose issues.

Sample Historical File Integration

Your Delivery Team will work with your technical lead and data extract engineer to review our data specification and map your data elements to the fields in the Oracle Utilities Opower data file format. Your engineering team then generates a sample historical data file. The sample historical file needs to include all the production-quality billed usage data for (at minimum) a random sample of 10,000 - 20,000 customer accounts—both inactive and active—and all their billed usage reads recorded since January of last year. For example, if your Delivery Team planned to launch your program for the first time on December 1, 2016, the team would need data going back to January 1, 2015.

The relatively small size of the sample file allows your Delivery Team to quickly analyze and diagnose issues. Your Delivery Team will provide you with a list of any issues found in the sample file and ask that you provide an updated file. This process is repeated until a sample historical file passes diagnostic testing. It typically takes several rounds of sample file generation, analysis, and feedback to approve a sample historical file.

Tip: Your Delivery Team is committed to making it as easy as possible for you to generate the files needed to launch your program. If it is easier for you to skip the sample historical file and generate a full historical file immediately, that is fine. You can discuss the easiest path with your Delivery Team.

Full Historical File Integration

After the sample file has passed validation, we can begin the process of generating, validating, and approving a full historical file. This file needs to include actual production data for all active and inactive customer accounts since January of the previous year. It is not expected that historical data since January of the last year will be available for every customer account.

The data set should also include accounts for customers who have moved in or moved out since January of the previous year. Oracle Utilities needs data for all utility customer accounts, *including inactive accounts*, even if only a small subset of those customers will be enrolled in the program. Within a given customer population, some customers will not meet the program eligibility criteria for reasons like low usage values or gaps in data. By providing more customer data, you create a larger pool of customers for statistical comparison. This allows Oracle Utilities to compare customers that are as similar as possible. Better comparisons will result in more actionable insights, more energy savings, and a better experience for your customers.

Tip: If possible, write your extract program so that it can receive an “end date” parameter at runtime. You can use this to generate the full historical file as of a few weeks in the past. This will keep a set of data available to use for generating and validating incremental files.

Full historical file approval typically requires several rounds of generation and validation. The approved historical file will be loaded into the Oracle Utilities Opower production application, and the data within it will be displayed to customers in reports, email messages, alerts, and/or web applications.

Data Extraction Flexibility

If your historical data set is so large that you need to extract it into multiple files, that is fine. If you need to send multiple files, your Delivery Team can append them together for analysis. In this case, make sure to follow the row and file-level sorting rules. See Legacy Billing Data File Specifications for more information.

Incremental Data Integration

After your Delivery Team approves and loads the full historical data, the team will generate an incremental file that includes any new customer information and billed usage data recorded since the most recent record in the full historical file that was delivered. Move in and move out data must be included as part of the incremental file.

Avoid Read Gaps!

Incremental file data should start immediately after the last read in the full historical file or files. There should be no gaps in reads for any customers between historical and incremental files.

Your Delivery Team will validate the file with a special focus on gaps, overlaps, and corrections.

After Delivery Team approves the first set of incremental files (preferably a full billing cycle’s worth), your team will schedule an automated file transfer. The file transfers should initially go to the `/shared_files` folder until you receive approval from the Delivery Team to start transferring to the `/uploads` folder. (See File Transfer Methods for more information.) During this transfer, your system will push ongoing incremental files to the Oracle Utilities Opower SFTP server. Ongoing incremental files should be sent at least once a week, but daily file transfer is also possible. Because reports are only created when new data is available for a customer, interval data must be provided at least once a week for Oracle Utilities to generate weekly reports. Files need to arrive on the SFTP server by the start of business Monday for any week in which we plan to generate Oracle Utilities Opower Home Energy Reports.

Tip: Send fresh data frequently. Even if we are not generating reports, the more often we receive data, the more often we can refresh the energy insights we deliver to customers across channels. Extracting data directly from your production database and sending it to Oracle Utilities directly after extraction will help ensure that it is as fresh as possible, so customers will receive the most current and accurate data.

After the initial incremental transfer, your Delivery Team will configure automated alerts for missing and incomplete files. These alerts enable both Oracle Utilities and your team to quickly

respond to any file transfer issues that may emerge after your Oracle Utilities Opower product is deployed to production.

Once incremental files are approved, the automated transfer is up and running and our alerts are listening for anomalies, then the data integration process is complete.

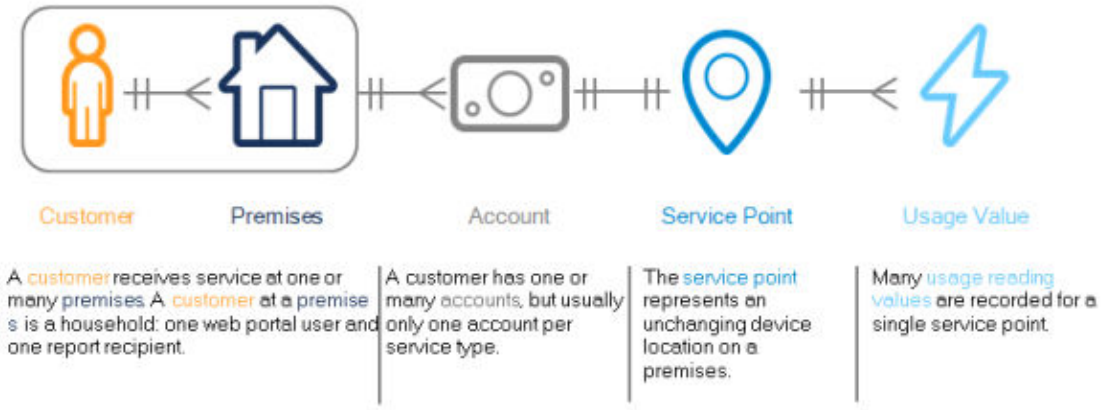
Tip: Usage data should be extracted based on last update, not last read date, to ensure it includes all reads, updates, and corrections since the last file. This will help to avoid gaps and inaccurate data.

Legacy Billing Data Entity Definitions

The Oracle Utilities Opower data model includes five key data entities. These five entities allow Oracle Utilities to provide the right usage information for the right fuel type to the right customer in the right location. Clearly understanding the relationships between these entities and representing them correctly in data files are crucial to meeting our shared implementation schedule, reaching your program goals, and delivering the desired experience for your customers. You are encouraged to ask questions about these entities early and often in the data integration process. The five key entities of the data module are:

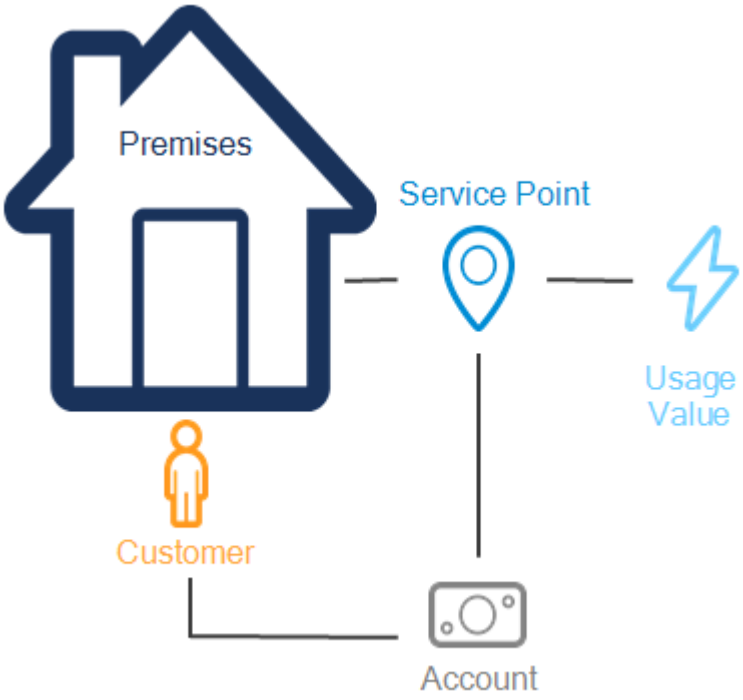
1. Customer
2. Premise
3. Account
4. Service Point
5. Usage Value

This section describes these five distinct data entities, the business rules that apply to each of them, and the relationships between them.



Data Entity Relationships

The following image illustrates the relationships between these entities. A single row in the data file always includes all five of these entity values.



Customer

A customer is an individual who receives service. The best **customer_id** to send is the one that uniquely identifies the customer and appears on the customer's bill. By default, Oracle Utilities will display the **customer_id** or a portion of it on outbound communications and the web, but if a different ID should be displayed on these communications and the web, you can provide it to Oracle Utilities in the **secondary_id** column.

- **Customer and Premise**: Oracle Utilities offers analysis of a customer's energy consumption at a specific location. For this reason, identifiers are needed for the customer (**customer_id**) and the service location (**premise_id**). Therefore, a "customer" is defined as a person at a premise rather than just an individual.
- **Multiple Premises**: Oracle Utilities does not aggregate usage across premises **(premise_id**). Customers that receive service at more than one premise are treated as separate potential report recipients and web users for each customer and premise combination. Customers with active accounts at more than two premises are ineligible for communications that include neighbor comparisons. It is assumed that these customers are property managers rather than residents.

Premise

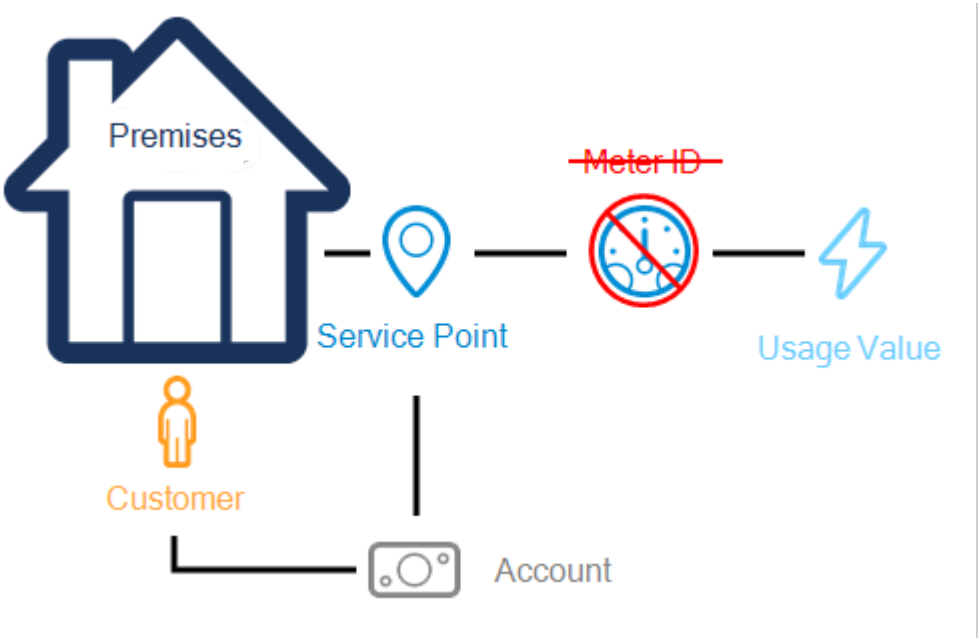
A premise is a physical, unchanging location. Oracle Utilities requires a premise identifier (**premise_id**) that uniquely identifies the location of the service. Individual units of multi-family dwellings, such as apartments or condos, must have unique premise identifiers. Premise identifiers should not change when residents change. When a customer moves out of a premise and another moves in, the original premise identifier should continue to appear in the data file alongside the new individual customer (**customer_id**) who begins to receive service at that premise.

Service Point

A service point identifies an unchanging device location where service is delivered. The IDs of meters located at a premise may change over time as hardware is replaced. A service point ID must remain unchanged from the time of construction until the location no longer exists. Therefore, it is important that the service point ID be distinct from a meter number ID and *not change* if a meter is replaced.

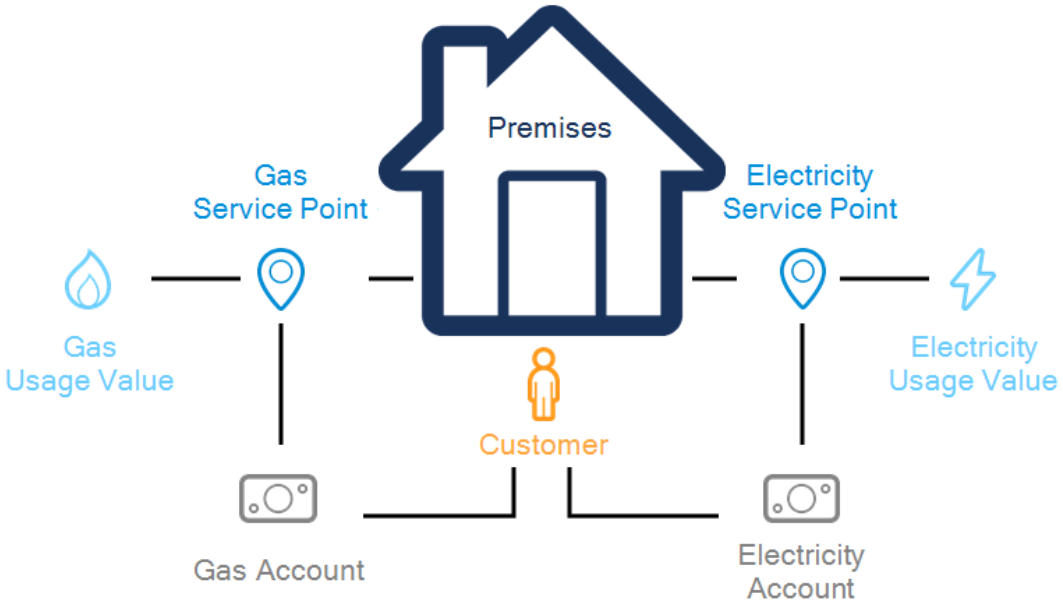
Do Not Change the Service Point ID

A change in service point ID will result in a loss of usage history at that service point, and render the premise ineligible to receive reports or serve as another premise's neighbor.



Service Point and Account IDs: A **service_point_id** can only be associated with one **account_id** at any given point in time, but over the life of the service point it will likely be associated with different accounts at different times. Therefore, the **active_date** and **inactive_date** on the account records are important to associate usage with the correct customer. See Account Fields for more information.

Dual Fuel Insights: Oracle Utilities offers the ability to deliver insights on a customer's total energy usage. A customer who has both gas and electric service should have one service point for each type of service. In order to aggregate the usage, both of the service points must be associated with the same customer and the same premise.



Service Point and Premise Solutions: Some dual fuel utilities do not maintain separate premise and service point identifiers. In this case, a common solution for creating the **service_point_id** is to concatenate the **premise_id** to the service type (**G, E**) or concatenated to a dash and a service number ("**-1**", "**-2**", "**-3**" …).

For example, given a **premise_id** of 123456, you may have a gas service point where the **service_point_id** is 123456-G and an electric service point where **service_point_id** is 123456-E. However, if you expect to have multiple service points of the same fuel type at a given premise, you may also need to add an additional, unchanging identifier for each of these service points, such as 123456-1-E, 123456-2-E, and so on. Discuss the options with your Delivery Team to determine what is appropriate for your use case.

Account

An account links a customer to a service point for a period of time. The **account_id** also associates a customer with usage values by way of the service point.

The most important attributes of the account are the **active_date** and **inactive_date** values, which indicate the period of time that service was provided to a customer at a given premise. A null **inactive_date** indicates a currently active account. Any non-null value in the **inactive_date** field discontinues the outbound Oracle Utilities Opower communications and/or web access for a customer whose account has been terminated.

- **Rate Changes**: If a customer switches to a new rate, the new rate code should be sent along with the existing account data with the same **active_date**, **account_id**, and all other required fields. Be sure to inform your Delivery Team of any new rates that should be eligible for reports. Oracle Utilities uses an allowlist for rate codes to exclude medical rates, for example, so if a customer switches to a new rate that has not been allowed, the customer will become ineligible for reports.
- **Temporary Lapses in Service or Suspension**: For a customer that has a temporary lapse in service or has their account suspended, the **inactive_date** should still be null. The **inactive_date** should only be set for a customer once they have received their final bill at that premise. You should continue to send usage reads for a customer that has a

temporary lapse in service. The usage value for any reads during a temporary lapse should be zero.

- **Attrition**: Once an account is deactivated, it can only be reactivated if you send a row with a null **inactive_date** where the **active_date** and **account_id** matches the earlier reads. Prematurely sending **inactive_date** values can result in high rates of attrition from the program and lower-than-forecast energy savings.

Never modify the active_date!

> **Warning**
>
> *Never modify the active_date*. When Oracle Utilities receives a service point record with a more recent active date than currently on file, the old account is automatically deactivated to hide prior usage history from the account and protect the customer's privacy. If you mistakenly send a newer active date, the customer will become ineligible for the program and lose access to usage data on relevant web applications (if applicable) prior to the new date.

Usage Value

A **usage_value** is the amount of energy consumed at a specific service point over a period of time that has specific start and end dates. This period of time is called the billing period, which is defined in the Oracle Utilities Opower system by the **date_to** and **duration** fields. The **date_to** should be the last day of usage that has been billed during the billing period. The **duration** is expected to reflect the total number of days inclusive to the billing period so that the first day of the billing period can be derived from the following equation:

date_to - **duration** + 1

For example, if a customer's first day of energy use for a given billing period is October 1, 2013 and the last day of consumption is October 28, 2013, then the **date_to** field is 20131028 and the **duration** is 28.

To prevent creation of a gap or an overlap in usage, the **date_to** minus the **duration** should equal the previous **date_to** value. Using our previous example with the most recent **date_to** being 20131028 and the **duration** is 28, the previous **date_to** date should be 20130930.

- **Usage Gaps and Overlaps**: If a gap or an overlap in usage is detected, the customer associated with that service point is ineligible to receive Oracle Utilities Opower communications for up to two years. Any gaps in usage should indicate an actual gap in service and should be a rare occurrence.
- **Usage Charge**: Each usage value has an associated **usage_charge**. This charge should only represent billed costs that vary with usage. Flat fees and/or other customer charges, for example, should not be included in the usage charges, because customers cannot control these fixed costs by reducing their usage. Costs are displayed throughout the platform because it motivates customer to take action, but only costs that customers can control will be shown.
- **Net Metering**: If your system supports net metering, add the delivered usage to the received usage and put the summed result in the **usage_value** field. You should do the same for the **usage_charge** value. If you mistakenly send two separate rows (one for delivered usage and another for received usage) that cover the exact same calendar days, Oracle Utilities treats the second row as a correction to the first row. You can also exclude net metering customers through a **rate_code** if desired.

Legacy Billing Data File Specifications

This section describes the distinct data fields that the utility must provide in the billing data file. The descriptions are organized into tables for each data entity represented in the billing data file: Customer, Account, Service Point, and Usage. (For an overview of these data entities, see Legacy Billing Data Entity Definitions.) Your Oracle Utilities Opower Delivery Team will work with you to determine whether you should provide all of the data in a single data file or in two separate data files. If you use a single file, then all of the data described must be included in the same file. If you use two separate files, then the two files should be organized be as follows:

1. One file must contain the data fields associated with the Customer, Account, and Service Point entities.
2. One file must contain the data fields associated with the Usage entity.

Your Delivery Team can also provide a billing data mapping template to assist with the process of relating the data in your system with the data that Oracle Utilities requires. Contact your Delivery Team for more information.

Column Order Matters!

In order for the data importer to interpret your data correctly, the fields must appear (from left to right) in a consistent order. Even if you do not send data for optional fields, you cannot remove columns from the file. Imports will also fail if you add additional columns to the middle of the file. For more information about the order of individual data files and file formatting rules, see Legacy Billing Data File Order and File Format.

Customer Fields

The billing data customer fields capture details such as the utility customer's name, address, and other contact information.

Each table row represents a column in the data file. The header row of the data file must contain all table "Column Name" values, even if all values in those columns are blank. The columns must appear from left to right in the order in which they appear in the table and be separated by tab characters. See Billing Data File Specifications for more information.

Column Name	Description
customer_id	Unique identifier for the customer that is familiar to the customer. This field is the default id that will be displayed on Oracle Utilities Opower communications. It is also used for web portal account creation and by Customer Service Representatives to locate customer accounts in the Customer Service Interface. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
premise_id	An identifier for a site occupied by an individual customer. Multiple service points (one per fuel type) may be tied to one premise, but only one customer can be tied to a premise at any given time. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.

Column Name	Description
secondary_id	Additional field for a secondary customer id. If `customer_id` is unknown to your customers, you can use this field as the value included on Oracle Utilities Opower communications, for Energy Efficiency Web Portal - Classic account creation, and for CSRs searching for customer accounts in the Customer Service Interface. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
ccb_account_id	Unique identifier for an account in the Oracle Utilities Customer Care and Billing (CCB) system. For more information about CCB, see the Oracle Utilities Customer Care and Billing documentation. **Note** If you are using Oracle Utilities CCB, this is a required field. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
mail_address_line_1	The first line of the mailing address that should contain the street address. **Note** The mailing address does not necessarily match the service point address. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
mail_address_line_2	Optional second line of the mailing address. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
mail_address_line_3	Optional third line of the mailing address. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
mail_city	The city in which the mailing address is located. For example, *Richmond*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.

Column Name	Description
mail_state	The 2-character state code where the mail address is located. For example, *VA* for *Virginia*. **Type**: STRING - 2 characters **Can Be Empty?**: No. **Default**: Not applicable.
mail_zip_code	5-digit zip or 9-digits of zip+4. For example, *22209* or *22209-1234*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
mail_address_line_4	May be necessary for some addresses with a building name or secondary thoroughfare and sublocality in mailing address. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
mail_locality	Post town / city of the mailing address. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
mail_postal_code	The 5-7 character postcode, For example, *M1 1AA*, or *DN55 1PT*. The six-character postal code for the mailing address. For example, *B3J 2Y3*. 4-digit numeric postal code. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
mail_admin_area_1	The abbreviated province for the mailing address. For example, *NS*. The abbreviated State or Territory for the mailing address. For example, *NSW*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
mail_country	Matches ISO 3166-1 alpha-2 country code. Allows you to distinguish customers with international mailing addresses so they can be rendered ineligible. **Type**: STRING **Can Be Empty?**: No. **Default**: Utility's country.

Column Name	Description
first_name	Customer's first name. **Type**: STRING **Can Be Empty?**: Yes.* *For residential customers, at least one of the following must be specified: `first_name`, `last_name`. If the name on a customer's bill includes a middle initial, this should be concatenated in the `first_name` field. If you have the `first_name` of a contact for a business customer (not residential), enter it here. If the name on a customer's bill includes a middle initial, this should be concatenated in the `first_name` field. For business customers, the `business_name` must be specified, as well as either `first_name` or `last_name`. **Default**: Null.
last_name	Customer's last name. **Type**: STRING **Can Be Empty?**: Yes.* *For residential customers, at least one of the following must be specified: `first_name`, `last_name`. If you have a `last_name` contact name for a business customer (not residential), enter it here. For business customers, the `business_name` must be specified, as well as either `first_name` or `last_name`. **Default**: Null.
phone_1	The primary phone number for a customer. For most utilities, this will be the only phone number that is used. Example US formats: (xxx)xxx-xxxx, xxxxxxxxxx, xxx-xxx-xxxx, x-xxx-xxx-xxxx **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
phone_2	A secondary phone number. This is used to denote SMS numbers. If this field is completed, the number must be able to receive SMS messages. Example US formats: (xxx)xxx-xxxx, xxxxxxxxxx, xxx-xxx-xxxx, x-xxx-xxx-xxxx **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
email	Customer's email address. This is required to enable delivery of email communications. Only one address should be sent. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.

Column Name	Description
owner	A flag to indicate whether a customer is an owner of the home. **Type**: STRING - 1 character **Allowed Values**: • O Customer is an owner. • R Customer is a renter. **Can Be Empty?**: Yes. If left blank, the customer's status as owner is marked "unknown." **Default**: Null.
customer_type	A classification for the customer, such as a residential or commercial customer. Only customers marked as `RESIDENTIAL` or `R` will receive a normative comparison experience. A normative comparison experience is a feature that compares a customer's energy use against that of their neighbors, such as the Neighbor Comparison in Home Energy Reports. **Type**: STRING **Allowed Values**: • A Customer is agricultural. • AGRICULTURAL Customer is agricultural. • C Customer is commercial. This value maps to the SMB value below. • I Customer is industrial. • INDUSTRIAL Customer is industrial. • OTHER Customer is another type that does not fall into the commercial, industrial, residential, or SMB categories. • R Customer is residential. • RESIDENTIAL Customer is residential. • SMB Customer is a small and medium business or commercial customer. **Can Be Empty?**: Yes. **Default**: Null. If left Null, the customer will be marked as RESIDENTIAL.

Column Name	Description
language_preference	A five-character code specifying a customer's preferred language/country. The language is specified by the two letter lowercase ISO language code defined in ISO-639. The country is specified by the two letter uppercase ISO country code defined in ISO-3166. The code must begin with the language and then the country separated by an underscore. If the value in this field is empty or NULL, then the default language code that is configured for the utility will be used for the customer. **Note** This field is primarily designed to support the multilingual functionality that is provided for utilities whose customers speak multiple languages. It is not frequently used. Contact your Delivery Team if you have any questions. **Type**: STRING **Example Values**: • fr_FR • en_US **Can Be Empty?**: Yes. **Default**: None.

Additional Customer Fields for Businesses

Column Name	Description
business_name	The name of the business. Use this field if the `customer_type` field has a value of `C` or `SMB` to refer to commercial customers. **Type**: STRING **Can Be Empty?**: Yes. **Note** If the `business_name` field is filled, you must also include information in `first_name` and `last_name` fields. If `business_name` is filled, but there is no information in `first_name` or `last_name`, the record will be rejected. **Default**: Not applicable.

Column Name	Description
other_business_segment	Utility specified business segment if the utility has a unique segmenting schema other than NAICS or SIC code. **Note** Only enter values in this column if your utility is part of the Oracle Utilities Opower SMB Energy Efficiency Pilot program. If your utility is not part of this pilot program, do not enter any values in this column. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.

Account Fields

The billing data account fields capture details such as the utility customer's account active dates and any applicable rate codes.

Each table row represents a column in the data file. The header row of the data file must contain all table "Column Name" values, even if all values in those columns are blank. The columns must appear from left to right in the order in which they appear in the table and be separated by tab characters. See Billing Data File Specifications for more information.

Column Name	Description
account_id	Utility's unique identifier for the account. An account is a contract between a person and a utility to receive a particular service at a certain location. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
active_date	Activation date for the account. The `active_date` must never change for a given `account_id`. The activation date should represent the beginning of the first usage period. Do not modify the `active_date`. **Type**: STRING (yyyymmdd) **Can Be Empty?**: No. **Default**: Not applicable.
inactive_ date	Termination date for the account, with null means that the account is currently active. This should only be set if the account has actually been terminated. Accounts that are temporarily inactive should not have this field populated. For example, *20140121*. **Type**: STRING (yyyymmdd) **Can Be Empty?**: Yes. **Default**: Null.

Column Name	Description
is_account_deletion	Deletion flag for the account. **Allowed Values**: • true • 1 • t Deletion is disabled by default to ensure that accounts are not erroneously deleted. If you want to allow deletions, make a request to your Delivery Team. We only allow deletion of the most recent account for a service point. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
read_cycle	Any unique alphanumeric string that specifies the time of month at which the meter read occurs. Used to ensure customers are compared against other customers with recent usage reads. This field should designate the service points that are read at approximately the same time. Utilities typically have 21 distinct read cycles corresponding to the business days in a month. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
rate_code	Rate code abbreviation for the rate plan associated with the account. The utility rate codes include, but are not be limited to: residential, commercial, low-income, time-of-use, and medical or disability rates. It is critical that the utility provide all rate codes applicable to residential customers. Inaccurate `rate_code` data could result in inaccurate savings numbers for customers and limit segmentation capabilities. A document describing the rate plans and structure must also be provided separately to Oracle Utilities. See Rates Data Transfer for details. **Type**: STRING **Can Be Empty?**: No. **Default**: Null.
termination_reason	This should indicate the account termination reason if `inactive_date` is not null. For example, *Move out*. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.

Service Point Fields

The billing data service point fields capture details such as the service point location, meter type, and any applicable parcel data.

Each table row represents a column in the data file. The header row of the data file must contain all table "Column Name" values, even if all values in those columns are blank. The columns must appear from left to right in the order in which they appear in the table and be separated by tab characters. See Legacy Billing Data File Specifications for more information.

Column Name	Description
service_point_id	Utility's unique identifier for the service point. A service point identifies an unchanging device location where service is delivered. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_house_number	The house number portion of the street address. For example, *123* for *123 Main Street*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_street_name	Street name. If applicable in your service territory, this field should include predirection (beginning of string), street suffix, or postdirection (end of string) as appropriate. For example, *W 24th St NE*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_unit	The unit number. For example, *2A* for *Apartment 2A*. **Note** This field is required for multi-unit premises. If possible, do not supply data such as "Third pole" that indicates where the meter is located. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
service_city	The city in which the service point is located. Do not include the state. For example, *Richmond*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_state	The 2-character state code where the service point is located. For example, *VA* for *Virginia*. **Type**: STRING (2 characters) **Can Be Empty?**: No. **Default**: Not applicable.
service_zip_code	5-digit zip or 9-digits of zip+4. For example, *22209* or *22209-1234*. Zip+4 is requested if available to help geocode service point addresses. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.

Column Name	Description
service_premise	A named location, usually a building or collection of buildings with a common name. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
service_street_number	The house number portion of the street address excluding floor, apartment, flat, or suite number/indicator. For example, *123* for *123 Main Street*.For example, *123* for *123 Main Street, Apartment A*. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
service_street	Street name of the service address - includes predirection (left), street suffix, or postdirection (right) as appropriate. For example, *24th St West*. Street name of the service address - includes street type and suffix as appropriate. For example, *24th St West*. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_subpremise	Apartment, flat, or floor # of the service address. For example, *Flat 2* or *5th Floor*. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
service_locality	Post town / city / locality of the service address. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
service_sublocality	Sublocality is required if the street name is not unique within the post town area. Include suburb in this field if it is a customary component of this address. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.
service_admin_area_1	Include the county of the service point, if available. Abbreviated province for the service address. For example, NS. Abbreviated state or territory for the service address. Province or territory abbreviation of the service point. **Type**: STRING **Can Be Empty?**: Yes. **Default**: Not applicable.

Column Name	Description
service_postal_code	5-7 character postcode. For example, *M1 1AA*, *DN55 1PT* of the service address. Six-character postal code for the service address. For example, *B3J 2Y3*. 4-digit numeric postal code. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
meter_type	The fuel type reported by the meter. **Allowed Values**: • E Electric • G Gas • W Water • WW Waste water **Type**: STRING **Can Be Empty?**: Yes. **Default**: E

Column Name	Description
meter_units	The unit of energy used to represent usage values. **Allowed Values**: • C Hundred cubic feet (CCF) • G or GAL Gallons • K or kWh Killowatt hours • KV or KVAR Kilovolt-ampere reactive • KVARH or KVH Kilovolt-ampere reactive hours • KW Kilowatts • L Liters • M or MCF Thousand cubic feet • M3 or 3 Cubic meters • T or TH Therms Only one `meter_units` unit type can be supported for each `meter_type`. For example, the therm and M3 units cannot both be used for the gas `meter_type`. In addition, it is not possible to change the `meter_units` after one has been set at the creation of a service point, which is typically at the beginning of a program. For example, if a gas unit of therms is used at the beginning of the program, the gas unit cannot be changed to M3 later in the program. If you need support to address either of these restrictions, contact your Delivery Team. **Type**: STRING **Can Be Empty?**: Yes. **Default**: K
voltage_class	A grouping of commercial customers based on how they receive electricity. Determines what discounts are earned by a commercial customer. **Allowed Values**: • PRIMARY • SECONDARY • TRANSMISSION **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.

Column Name	Description
latitude	Latitude of the service address, expressed as a decimal number of degrees between -90 and 90, no directional indicators. Any latitude data should be rooftop or better. If less specific than rooftop, leave as null (empty/blank). If no latitude value is available, leave this empty instead of supplying a value of 0. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
longitude	Longitude of the service address, expressed as a decimal number of degrees between -180 and 180, no directional indicators. Any longitude data should be rooftop or better. If less specific than rooftop, leave as null (empty/blank). If no longitude value is available, leave this empty instead of supplying a value of 0. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
dwelling_type	An indication of the type of dwelling in which a customer lives. **Allowed Values**: • SING • MULT • CONDO • DUPLEX • TRIPLEX • TOWNHOUSE • MOBILE • COOP • QUADRUPLEX • OTHER_RESIDENTIAL • NON_RESIDENTIAL **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
bldg_sq_foot	Square footage of the living area of the building. This generally does not include the garage, unfinished basement, or porch. This information is used to select neighbors or similar homes for the normative comparison. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
parcel_sq_foot	Square footage of the parcel lot. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
year_built	Year the structure was built. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.

Column Name	Description
total_rooms	Total number of rooms in the structure. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
assess_value	Either the assessed value of the property, or a number or letter representing a range for the value. A blank field indicates that the assessed value is unknown. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
heat_type	The heat type used by the customer. **Allowed Values**: • ELEC • GAS • NONE • OTHER • LPG • OIL • WOOD • DISTRICT_HEATING • GEOTHERMAL • Null (blank) = unknown **Type**: STRING **Can Be Empty?**: Yes. **Default**: Null.
photovoltaic	A flag to indicate whether the customer generates photovoltaic or solar power. **Allowed Values**: • 0 Customer does not generate photovoltaic / solar power. • 1 Customer does generate photovoltaic / solar power. • Blank It is unknown whether the customer generates photovoltaic / solar power. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.

Column Name	Description
pool	A flag to indicate whether the customer has a pool. **Allowed Values**: • 0 Customer does not have a pool. • 1 Customer does have a pool. • Blank It is unknown whether the customer has a pool. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
air_cond	A flag to indicate whether the customer has air conditioning. **Allowed Values**: • 0 Customer does not have air conditioning. • 1 Customer does have air conditioning. • Blank It is unknown whether the customer has air conditioning. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
bedrooms	Number of bedrooms in the structure. This can be used in place of `bldg_sq_foot` to identify the size of a home for the purpose of selecting neighbors. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
amperage	The amperage value, expressed as a floating point number. For example, *1.24*. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Not applicable.

Usage Fields

The billing data usage fields capture details such as the customer's total energy usage amount, bill cycle duration, and total billing charge.

Each table row represents a column in the data file. The header row of the data file must contain all table "Column Name" values, even if all values in those columns are blank. The columns must appear from left to right in the order in which they appear in the table and be separated by tab characters.

> **Note**
>
> If you are sending data in two files to Oracle Utilities, then this data should be included in its own, separate file. See Billing Data File Specifications for more information.

Usage data should be transmitted in a separate file.

Column Name	Description
service_point_id	Utility's unique identifier for the service point. See Legacy Billing Data Entity Definitions for more information. **Type**: STRING **Can Be Empty?**: No. **Default**: Not applicable.
usage_value	Usage value amount based on meter read in units determined by meter's units. Commas cannot be used in this field. The value must be positive and the usage cannot exceed 999999999 kWhs. Values that are too large will be rejected. Negative usage values can be accepted, but this requires configuration. Contact your Delivery Team for more information. **Type**: NUMBER **Can Be Empty?**: No. **Default**: Not applicable.
date_to	End date of meter read period. **Type**: STRING (yyyymmdd) **Can Be Empty?**: No. **Default**: Not applicable.
duration	Number of days the usage value represents. For example, this could be 30 for a monthly billing cycle, 60 for a bimonthly billing cycle, and so on. Durations for monthly, bi-monthly, quarterly, and annual billing cycles are supported. **Type**: NUMBER **Can Be Empty?**: No. **Default**: Not applicable.
is_estimate	A flag to indicate whether a usage read was estimated or not. **Allowed Values**: • A Actual read by utility. • E Estimated read. • U User-reported read. A user-reported read is treated as an estimated read when it is imported into the Oracle Utilities Opower system. **Type**: STRING **Can Be Empty?**: No. **Default**: E

Column Name	Description
usage_charge	Amount billed to the customer for usage during this period. This should only include the variable charge for the actual energy usage, and should exclude taxes, account fees, rebates, or other bill adjustments. If the reported `usage_value` is the sum of usage from multiple meters at the same `service_point`, the `usage_charge` is required to provide an effective usage rate. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.
demand_peak	Highest usage mark within a month based on meter read in units determined by the meter's units. Generally, this will be present only if the utility has a special high-water mark meter installed at a business site. **Note** Only enter values in this column if your utility is part of the Oracle Utilities Opower Small and Medium Business Energy Efficiency pilot program. If your utility is not part of this program, do not enter any values in this column. **Type**: NUMBER **Can Be Empty?**: Yes. **Default**: Null.

Legacy Billing Data File Order and File Format

Billing data must be sent in a particular order and file format.

Data File Order

The Oracle Utilities Opower platform processes files according to the order received, then dated. If we receive multiple files on the SFTP server in the same time frame (as could be done for full historical data files), they will be automatically queued by their file name date stamps.

For example, if a file is received on the SFTP server named *opwr_util_full_20140520.txt* and then receive one named *opwr_util_full_20140519.txt* an hour later, *opwr_util_full_20140520.txt* will be processed first despite the fact that its date stamp is later than the *opwr_util_full_20140519.txt*. However, if these files are received at the same time, the file name data stamp will take precedence and *opwr_util_full_20140519.txt* will be processed before *opwr_util_full_20140520.txt*.

To ensure that all files are queued in the correct order, files should be named so that the date stamps are consistent with the intended load order of the files.

Data Order Matters!

Data (rows) for each service point should always be arranged from the oldest to the most recent to ensure that your data is processed in the correct order.

- **Multiple Usage Reads for a Single Service Point**: Any time a single file contains more than one usage read for a single service point covering the same period of time, the most recent read needs to appear last in the file. If the usage periods for those two reads are the same, Oracle Utilities considers the last record read to be a correction. See Legacy Billing Data Entity Definitions for more information.
- **Multiple Records for the Same Entity**: If there are multiple records for the same entity (such as customer, account, or service point), in the same file, only the last record for that entity is loaded. Oracle Utilities assumes that the last record represents the most current data from the utility.
- **Customer Information**: The data in the last row of the last file processed is always the active data in the Oracle Utilities database for that customer.

File Format

All files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.**Note**: The only fields that do not yet support non-Latin characters are `premise_id` and `email`. Contact your Delivery Team if you need to send non-Latin characters for these fields, so that a workaround can be provided.
- Some fields must have English values. These are fields that have an enumerated set of expected values in the field tables. For example, for the **customer_type** field, the values must be **AGRICULTURAL**, **RESIDENTIAL**, **SMB**, or **OTHER**. The Oracle Utilities Opower platform can only accept these exact strings.
- All files must include a header row containing column names from the tables specified in this document. Header column names must exactly match the names specified in this data specification.
- All rows should have the same number of tabs even if some fields have null values. It is important that null values are represented by empty strings and not denoted by **/N**, **NA**, **null**, or another value.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file. Compressed gzip or zip files must contain only one text file each.

Legacy Billing Data File Name Conventions

All billing data files transferred to Oracle Utilities must follow a standard naming convention. The convention to use depends on whether you are sending your data in a single file or two separate files. Your Delivery Team will work with you to determine which convention to use.

File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

opwr_<utility>_res_<specification version number>_full_<yyyyMMddHHmmss>.<gz or zip>

opwr_<utility>_res_<specification version number>_customer_<yyyyMMddHHmmss>.<gz or zip>

opwr_<utility>_res_<specification version number>_usage_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Legacy Billing Data Version History.
- **Data Type**: If you are sending data in a single file, then the filename must include the word *full* to indicate that all recipient and usage information headers are in the same file. If you are sending two files, the filename must contain the word *customer* or *usage*. The Service Point data file should include the word *customer*. The file that contains the Usage data should include the word *usage*.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company.” An example of your utility’s file name would be as follows:

opwr_gec_res_v1-0-0_full_20210316131415.gz

opwr_gec_res_v1-0-0_customer_20210316131415.gz

opwr_gec_res_v1-0-0_usage_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility’s file name would be as follows:

opwr_gec_res_v1-0-0_full_20210316131415.pgp

opwr_gec_res_v1-0-0_customer_20210316131415.pgp

opwr_gec_res_v1-0-0_usage_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system

considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.full.20210316131415.gz

Additional Business Rules for Legacy Billing Data

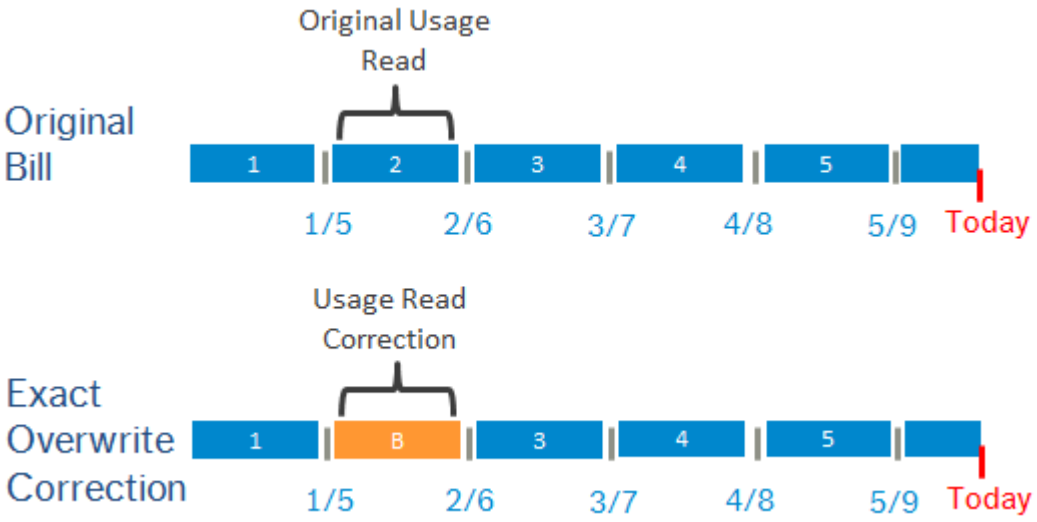
Usage Value Corrections

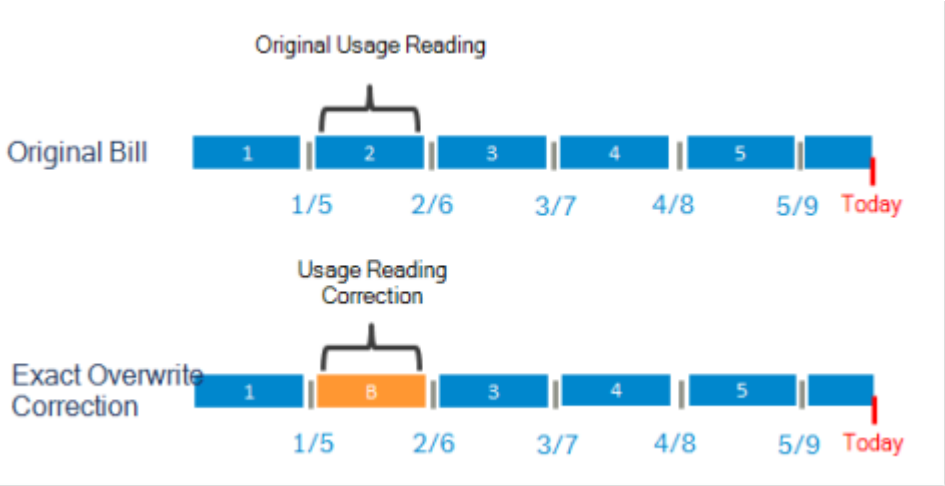
Oracle Utilities supports corrections when a usage value or charge needs to be updated after a customer has been billed. Two kinds of usage value corrections are accepted: exact overwrites and overlapping overwrites.

Usage Value Assumptions

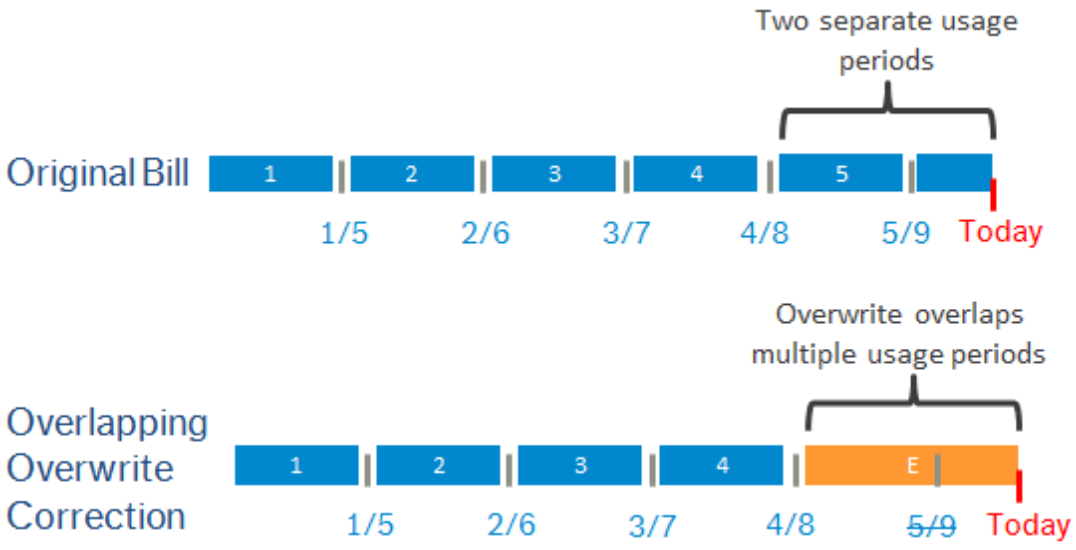
The examples in this section assume a month-long read cycle that begins on the first day of each month.

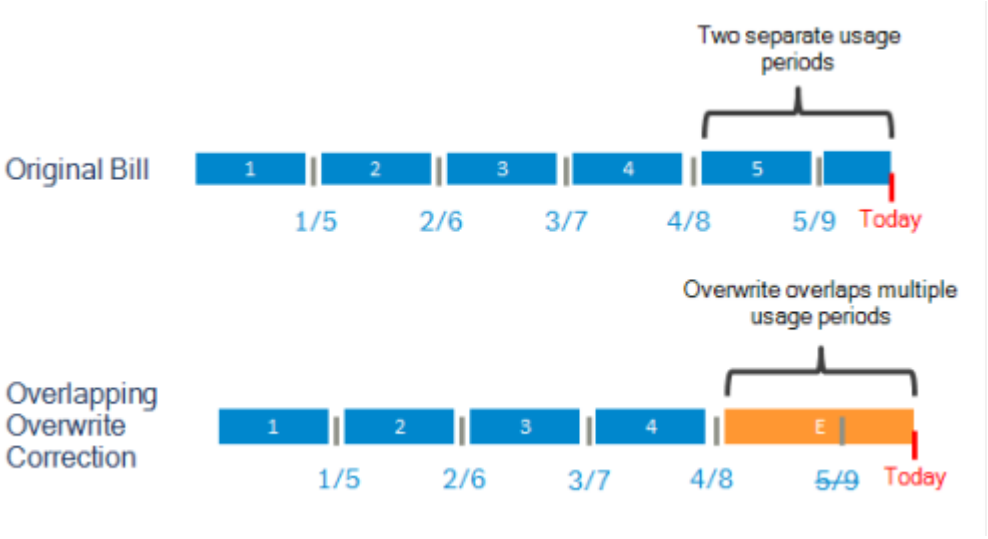
Exact Overwrites: You can send a new row that has the same **service_point_id**, **date_to**, and duration along with the corrected usage value and/or charge. The original usage value and charge will be replaced with the new ones. In the example below, exact overwrite correction B replaces the original usage value of billing period 2.



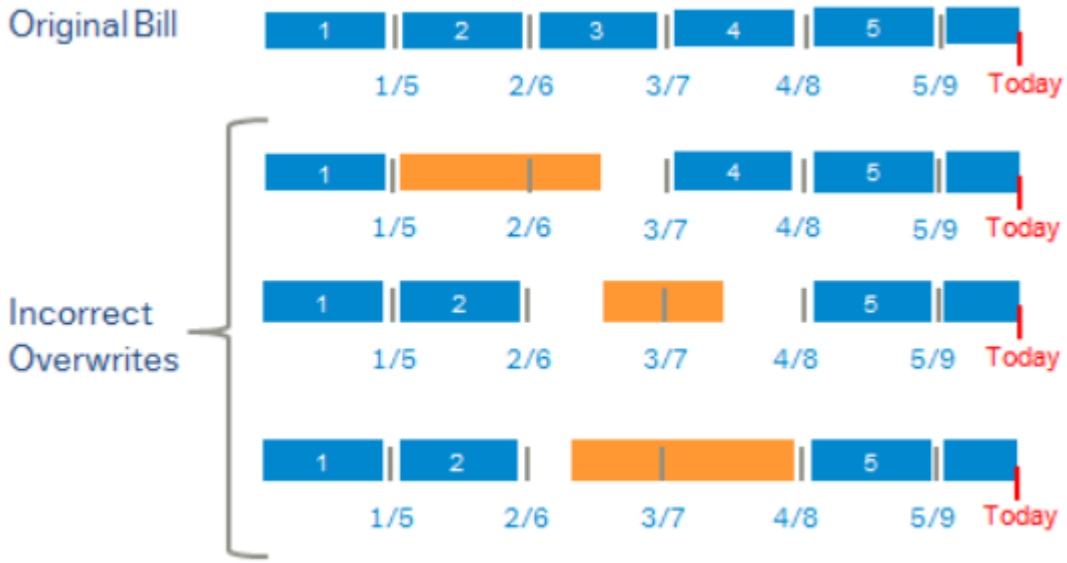


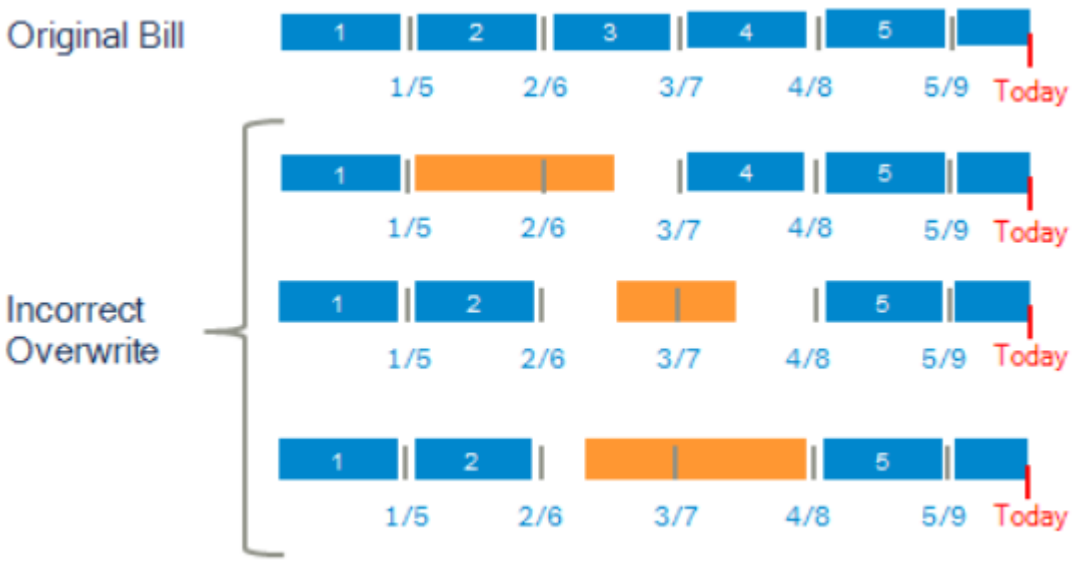
Overlapping Overwrites: You can send a new row that completely covers the original usage read. In this case, the **date_to** will be more recent and the duration will be longer than latest **date_to** on record for that **service_point_id**. The new **date_to** must make the corrected read the most recent read received for the service point, and the derived start date of the new usage period must match the start date of an old usage period. When an overlapping overwrite correction is loaded, the Oracle Utilities Opower application removes any old intermediate reads that fall within the duration of the new read. The usage value and usage charge of the new read must reflect all usage and all usage-based charges for the new, longer usage period. In the example below, overlapping overwrite correction E replaces the original usage values of billing period 5 though today.





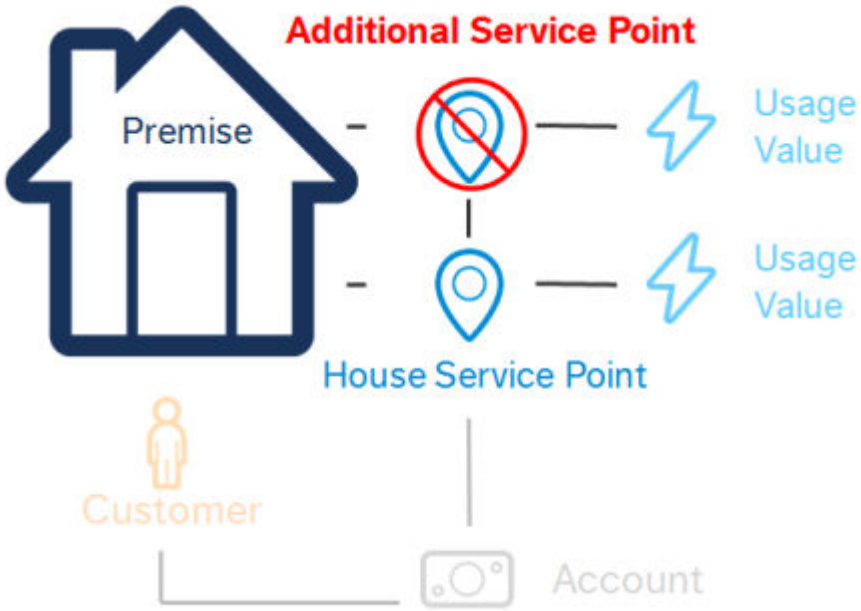
Incorrect Overwrites: Below are several examples of incorrect corrections. Note how they violate the rules above by starting or ending in the middle of a usage read periods.

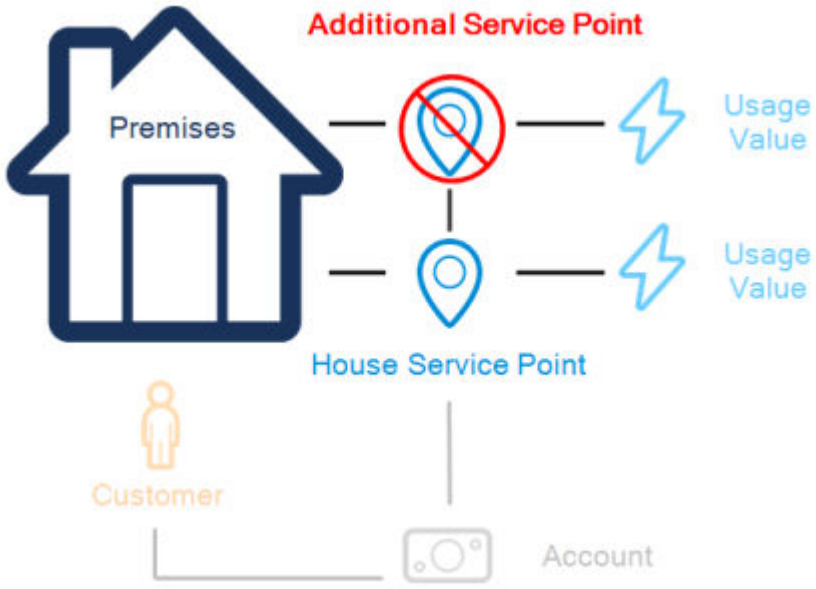




Multiple Service Points for a Single Fuel Type

Oracle Utilities will only generate usage comparisons for premises with a single service point for a given fuel type.





Making this subset of customers ineligible ensures that we do not attempt to compare residential usage to usage for non-residential service points such as garages, barns, and pools. Discuss options for customers with multiple active service points with your Delivery Team.

Landlords

When a site moves between a tenant and a landlord, it is treated as a standard move-out, move-in scenario. For example, if a tenant moves out, their account should have `inactive_date` set to the date they move out, and the landlord should have `active_date` set to the following day. The tenant and landlord should have their own unique `customer_id` and `account_id`. When a tenant moves back in to the site, the landlord should have an `inactive_date` set for their account associated with that site, and the customer should have an `active_date` set for the following day. This prevents usage data from being exposed when the site transfers ownership.

Customer Transfers to a New Site

When a customer moves to a new site but continues service at their new location, it is treated as though their old account closed, and they opened a new account. Their old account should have `inactive_date` set, and they should receive a new `active_date`, `account_id`, and `premise_id` at their new location. Their `customer_id` can remain the same. This is due to Oracle Utilities considering an account as the combination of a customer at a premise.

Legacy Billing Data Version History

The following table lists updates to the billing data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Legacy Billing Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update Legacy Billing Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-19-0	04/11/2022
Clarify that zip files should only contain one file each.	v1-18-0	11/23/2021
Update Usage Fields section to clarify that the `usage_value` field supports negative values. Update Account Fields, Service Point Fields, and Usage Fields sections to use two-column table format.	v1-17-0	09/14/2021
Update Customer Fields section to clarify the `customer_type` field and `business_name` field descriptions.	v1-16-0	06/09/2021
Update a few terms to use more inclusive language.	v1-15-0	05/12/2021
Update Customer Fields section to clarify the `customer_type` description and convert table to a simpler two-column format.	v1-14-0	10/09/2020
Move information about the file retention policy to a more prominent place.	v1-13-0	05/14/2020
Update titles and naming conventions where applicable to rebrand the specification as a legacy version. Clarify file extension requirements and expectations. Clarify that the `service_unit` field is required in the case of multi-unit premises.	v1-12-0	11/12/2019
Update the list of allowable values in the `customer_type` field in the Customer Fields section. Clarify purpose of the `duration` field in the Usage Fields section.	v1-11-0	06/14/2019
Clarify that commas cannot be used in `usage_value` field. Fix typo in `active_date` description. Update supported values for `customer_type`, `meter_type`, and `meter_unit` fields.	v1-10-0	04/10/2019

Update	Specification Version Number	Date
Edit sections and headings to simplify content and clarify instructions. No substantive changes to technical specifications.	v1-9-0	12/12/2018
Updated file retention section to reflect regional differences.	v1-8-0	05/08/2018
Updated the descriptions for `meter_type` and `meter_units`. Added statement about valid end-of-line separators.	v1-7-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-6-0	03/03/2018
Added new `ccb_account_id` field in Customer table. Removed duplicate `account_id` field from Usage table.	v1-5-0	01/12/2018
Updated definition of the bldg_sq_foot value to clarify that it does not generally include spaces such as the garage, unfinished basement, or porch.	v1-4-0	06/26/2017
Added a statement about the utility's data obligations. Removed information about `register_id`.	v1-3-0	03/31/2017
Added document number and updated branding.	v1-2-0	02/10/2017
Added legal notices. Updated document branding.	v1-1-0	01/11/2017
Added baseline specification version number to the document. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file. Added more detail to `language_preference` description.	v1-0-0	09/22/2016

Update	Specification Version Number	Date
Added note to the `is_estimate` description to clarify that user-reported reads are treated as estimated reads and to standardize the default value for all locales. Updated the `usage_value` description to change the upper bound from 999999 to 999,999,999.	Not applicable.	05/27/2016
Clarified that data from the utility's production system is required. Fixed minor typos.	Not applicable.	05/09/2016

5

Supplemental Data Standards

Supplemental data standards contain data elements used to support additional Oracle Utilities Opower products and capabilities. For example, they include data elements that enable the display of interval (AMI) data insights or cost insights in the Opower products. The supplemental data standards are not required for the base components of the Opower program.

- Call Center Data Transfer
- Customer Classification Data Transfer
- Interval Data Transfer
- Program Participation Data Transfer
- Rates Data Transfer
- Rate Engine Plus Data Transfer
- Unsubscribe Data Transfer

Call Center Data Transfer

Call center data files contain records for each contact that the customer has with a utility's call center, including wait time, talk time, and hold time. This data can then be used to provide insights about customer service.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Call Center Data File Specifications

This section describes the distinct data fields that you must provide in the call center data file. It also provides information about the data file format and file name conventions. If your utility only has some of the data listed below, or has the data in a different format, we will work with your utility to meet your specific needs. Contact your Oracle Utilities Delivery Team to discuss these circumstances.

> **Warning**
>
> Although the `customer_id` and `premise_id` fields listed below are marked as optional, the utility should plan to provide this data to Oracle Utilities for any row for which it is available.

Field	Description
customer_id	The utility's unique identifier for the customer. This customer ID should be the same customer ID that is sent in the billed usage files, as described in Customer Fields. Although it is marked as optional, the utility should plan to fill in this field wherever it is available. **Type**: VARCHAR **Can Be Empty?**: Yes.
premise_id	An identifier for a site occupied by an individual customer. This premise ID should be the same premise ID that is sent in the billed usage data files, as described in Customer Fields. Although it is marked as optional, the utility should plan to fill in this field wherever it is available. **Type**: VARCHAR **Can Be Empty?**: Yes.
phone_number	The phone number used by the customer to contact the utility. Example US formats: (xxx) xxx-xxxx, xxxxxxxxxx, xxx-xxx-xxxx, x-xxx-xxx-xxxx **Type**: STRING **Can Be Empty?**: No.
start_tstamp	Timestamp with the date and time of when the call began. Data should be transmitted in the format YYYYMMDD hhmmss. **Type**: DATETIME **Can Be Empty?**: No.
end_tstamp	Timestamp with the date and time of when the call ended. Data should be transmitted in the format YYYYMMDD hhmmss. This can be left empty only if duration is filled in. **Type**: DATETIME **Can Be Empty?**: No. **Note** This can be left empty if only the `duration` field is populated.
duration	Duration of the call in seconds. The maximum value is 99999. This is not needed if you provide the `start_tstamp` and `end_tstamp`. **Type**: INT **Can Be Empty?**: Yes. **Note** If `end_tstamp` is not provided, then this field is required.
wait_time	Duration customer was on hold before a representative first answered the call, in seconds. The maximum value is 99999. **Type**: INT **Can Be Empty?**: Yes.

Field	Description
talk_time	Duration representative and customer were in conversation, in seconds. The maximum value is 99999. **Type**: INT **Can Be Empty?**: Yes.
wrap_time	Duration representative spent after the customer hung up to perform closing activities, in seconds. The maximum value is 99999. **Type**: INT **Can Be Empty?**: Yes.
hold_time	Duration customer was put on hold during the call, after the first answer, in seconds. The maximum value is 99999. **Type**: INT **Can Be Empty?**: Yes.
csr_id	Unique identifier of responding customer service representative (CSR). **Type**: VARCHAR(100) **Can Be Empty?**: Yes.
type	A reason code or call disposition code consistent across all calls so that Oracle Utilities can interpret the reason for the call. This must be in a human comprehensible format, or a decoding must be provided. **Type**: VARCHAR(100) **Can Be Empty?**: No.
notes	Any notes the CSR recorded during the call. **Type**: VARCHAR(5000) **Can Be Empty?**: Yes.

File Format

The following conventions apply to all files generated for transfer:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files. CSV format is also supported.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- All files should include a header line containing column names from the tables specified in this document.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- Do not remove columns or add columns to the middle of the file.

File Name Conventions

All files transferred to Oracle Utilities must follow the standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

opwr_<utility>_res_<specification version number>_callcenter_<yyyyMMddHHmmss>.gz

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.

- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Call Center Data Version History.
- **Data Type**: The filename must include the term *callcenter* to indicate the type of data included.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company.” An example of your utility’s file name would be as follows:

opwr_gec_res_v1-0-0_callcenter_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility’s file name would be as follows:

opwr_gec_res_v1-0-0_callcenter_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.callcenter.20210316131415.gz

Call Center Data Version History

The following table lists updates to the call center data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See File Name Conventions for more information.

Update	Specification Version Number	Date
Update File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-11-0	04/11/2022
Update table in Call Center Data File Specifications to use two-column format. Clarify that zip files should only contain one file each.	v1-10-0	11/23/2021
Move information about the file retention policy to a more prominent place.	v1-9-0	11/23/2021
Clarify file extension requirements and expectations.	v1-8-0	11/12/2019
Edit sections and headings to simplify content. No substantive changes to technical specifications.	v1-7-0	12/21/2018
Updated file retention section to reflect regional differences.	v1-6-0	05/08/2018
Added statement about valid end-of-line separators.	v1-5-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-4-0	03/02/2018
Added a statement about the utility's data obligations.	v1-3-0	03/31/2017
Added document number and updated branding.	v1-2-0	02/09/2017
Added legal notices. Updated document branding.	v1-1-0	01/11/2017
Added baseline specification version number to the document. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file.	v1-0-0	10/27/2016
Made the csr_id field required. Clarified that either the end_time or duration field is required, though one can be empty if the other is filled in. Minor updates to document formatting and organization.	Not applicable.	06/16/2016

Update	Specification Version Number	Date
Updated document formatting and organization. Added new rows for phone_number, wait_time, talk_time_wrap_time, and hold_time. Made customer_id and premise_id optional.	Not applicable.	05/10/2016

Customer Classification Data Transfer

Customer classification data allows a utility to categorize its customers into different groups or “classifiers” for tracking or targeting purposes. Based on this information, Oracle Utilities can then create tailored communications and user experiences. For example, utilities may want to designate a group of their customers as being "low income" through the customer classification data files. These customers can then be sent tailored marketing messages and tips for saving energy that best suit them.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Customer Classification Data File Specifications

There are two separate customer classification data files that you must send to Oracle Utilities: the customer classification definition file and customer classification file. This section describes the distinct data elements that these data files must contain.

> **Note**
>
> The Customer Classification Definition file must be sent prior to the Customer Classification file. This is because the classifiers defined in the definition file are used to validate the data in the classification file.

Customer Classification Definition File

The customer classification definition file describes the segments (classifications) into which customers can be placed. The following table defines the elements that should be provided in the columns of the file.

Column	Definition
classifier	The main classification name. • Be consistent with upper and lower case. For example, do not use both `Low_Income` and `low_income`. Choose one. • Must include only letters, underscores, hyphens, and periods. • Special characters are not accepted. **Type**: VARCHAR **Can Be Empty?**: No.
classifier_description	Description of the classifier and its purpose for the utility. **Type**: VARCHAR **Can Be Empty?**: No.
allowed_value	Possible value for the classifier. • Be consistent with upper and lower case. For example, do not use both `True` and `true`. Choose one. • Must include only letters, underscores, hyphens, and periods. • Special characters are not accepted. **Type**: VARCHAR **Can Be Empty?**: No.
allowed_value_description	Description of the allowed value and its purpose for the utility. **Type**: VARCHAR **Can Be Empty?**: No.
inactive_date	The date and time at which the `allowed_value` is no longer valid for data import. • This date must not be prior to the import. If it is, the row will be rejected. • If left blank, this field will default to leave the `allowed_value` valid indefinitely. **Type**: DATETIME (YYYYMMDD HHmm). See Date and Time Handling for details. **Can Be Empty?**: Yes.

Additional Notes

To add a new classifier, simply add a new row containing the `classifier` and `allowed_value` pair in the file. To de-activate an existing classifier, assign an `inactive_date` to the classifier-allowed value pair. If an inactive date is assigned to a classifier-allowed value pair, then any new assignments of customers to that classifier-allowed value that fall after the inactive date will be rejected.

> **Note**
>
> Consult with your Delivery Team to make ensure that the products exhibit the correct behavior for each classifier.

Example Customer Classification Definition File

The table below is an example of the Customer Classification Definition File containing different classifiers. These are meant as examples only. Utilities may choose to use these classifier-allowed value pairs or others of their choosing.

classifier	classifier_description	allowed_value	allowed_value_description	inactive_date
low_income	Indicates whether the customer is low-income	True	Tags a low-income customer to receive tips relevant to low-income customers	
language	The preferred language of the customer	English	Tags a customer to receive communications in English	
language	The preferred language of the customer	French	Tags a customer to receive communications in French	
DoNotSolicit	Customers in the "do not solicit" list	True	Exclude these customers from future expansions	

Customer Classification File

The customer classification file lists the customers and their respective classifications. The table below defines the elements that must be included in the columns of the customer classification file.

> **Note**
>
> If the Oracle Utilities Opower system encounters any classifiers that have not been already defined in the Customer Classifier Definition file, the corresponding rows in the Customer Classification file will be rejected.

Column	Definition
customer_id	Unique identifier for the customer. This customer ID should be the same customer ID that is sent in the billed usage files, as described in Customer Fields. **Type**: VARCHAR **Can Be Empty?**: No.
premise_id	Identifier for the premise with which the customer is currently associated. This premise ID should be the same premise ID that is sent in the billed usage data files, as described in Customer Fields. **Type**: VARCHAR **Can Be Empty?**: No.
classifier	Key identifying the classifier. The classifier should be the same as one of the classifiers already listed in the corresponding definition file. **Type**: VARCHAR **Can Be Empty?**: No.

Column	Definition
value	Value for the classification. The value should be the same as one of the allowed values already listed in the corresponding definition file. **Type**: VARCHAR **Can Be Empty?**: No.
start_date	The date and time at which the customer is classified according to the classifier/value. This is useful for classifers that are time-sensitive. For example, this could be used if a customer is part of a low-income group for a certain period and then is removed from the group. If left blank, the start date is assumed to be the date of file import. **Type**: DATETIME (YYYYMMDD HHmm). See Date and Time Handling for details. **Can Be Empty?**: Yes.
end_date	The date and time at which the customer is no longer classified according to the classifier/value. If left blank, the classification value is assumed to be true indefinitely. If the value for a customer's classification is changed, and the end date is not explicitly sent by the utility, the end date will be set to the date prior to the start date of the new classification. **Type**: DATETIME (YYYYMMDD HHmm). See Date and Time Handling for details. **Can Be Empty?**: Yes.

Example Customer Classification File

The table below is an example of customer classification data. It is intended as an example only.

customer_id	premise_id	classifier	value	start_date	end_date
779222413	603400	low_income	True	20130101 1200	20180101 1200
822022136	143344	low_income	True	20130101 1200	
530432863	261411	low_income	True		
572410557	400657	low_income	True		
579494002	259691	language	French	20170601 1200	
559409440	572129	language	French		
799909879	573400	language	English		

Customer Classification Data Order and File Format

Customer classification data is processed in a particular order and file format.

Data Order

The Oracle Utilities Opower platform processes files according to the order received and dated. The last record received for a row of identical date and time will be considered a replacement and only the final updates will be retained in the system. Therefore, if multiple files are received on the SFTP server in the same timeframe (as would be done for historical data files), then the files must be named so that the file with the oldest data has the oldest date and timestamp, allowing the system to process them in the appropriate order. See Customer Classification File Name Conventions for more information.

File Format

The following file format conventions apply to all files generated for transfer:

- All data files must be sent as tab-delimited text in .tsv or .txt format. The same number of tabs must exist in each row.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.**Note**: The only fields that do not yet support non-Latin characters are `premise_id` and `email`. Contact your Delivery Team if you need to send non-Latin characters for these fields, so that a workaround can be provided.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Customer Classification File Name Conventions

All customer classification files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Customer Classification Definition File:

opwr_<utility>_res_<specification version number>_classification_customer_definition_<yyyyMMddHHmmss>.gz

Customer Classification File:

opwr_<utility>_res_<specification version number>_classification_customer_<yyyyMMddHHmmss>.gz

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Customer Classification Data Version History.
- **Data Type**: The filename must contain the words *classification_customer* or *classification_customer_definition* to indicate the type of data it contains.

- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All data files must be sent in tab-delimited text in .tsv or .txt format. All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company." An example of your utility's customer classification definition specification file name would be as follows:

opwr_gec_res_v1-0-0_classification_customer_definition_20210316131415.gz

An example of your utility's customer classification specification file name would be:

opwr_gec_res_v1-0-0_classification_customer_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's customer classification definition specification file name would be as follows:

opwr_gec_ res_v1-0-0_classification_customer_definition_20210316131415.pgp

An example of your utility's customer classification specification file name would be as follows:

opwr_gec_res_v1-0-0_classification_customer_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.classification.customer.definition.20210316131415.gz

Customer Classification Data Verification Checklist

Use the checklist in this section to ensure you have completed all required actions before sending the data files to Oracle Utilities.

File Format Checks

- The filename follows the appropriate pattern. See Customer Classification File Name Conventions for more information.
- The file is tab-delimited and is in .tsv or .txt format.
- The same number of tabs exists in each row.

Header Row Checks

Column headers must match exactly the headers shown below.

Customer Classification Definition File:

```
classifier classifier_description allowed_value allowed_value_description
                inactive_date
```

Customer Classification File:

```
customer_id premise_id classifier value start_date end_date
```

Data Field Checks

- All data fields in the Customer Classification Definition file follow the rules described in this document. See Customer Classification Definition File.
- All data fields in the Customer Classification file follow the rules described in this document. See Customer Classification File.
- Null values are empty strings. (They must be not denoted by /N, NA, null, or some other non-supported notation.)

Customer Classification Data Version History

The following table lists updates to the customer classification data document specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Customer Classification File Name Conventions for more information.

Update	Specification Version Number	Date
Update Customer Classification File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-12-0	04/11/2022
Clarify that zip files should only contain one file each.	v1-11-0	11/23/2021
Clarify file extension requirements and expectations.	v1-10-0	11/12/2019
Clarify that data files must be in .tsv or .txt file format before being compressed and sent to Oracle Utilities. Add an example to clarify when the `start_date` field is useful.	v1-9-0	01/28/2019
Edit sections and headings to simplify content. No substantive changes to technical specifications.	v1-8-0	01/10/2019
Updated file retention section to reflect regional differences.	v1-7-0	05/08/2018
Added statement about valid end-of-line separators.	v1-6-0	03/19/2018

Update	Specification Version Number	Date
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-5-0	03/02/2018
Added a statement about the utility's data obligations.	v1-4-0	03/31/2017
Added document number and updated branding.	v1-3-0	02/09/2017
Added legal notices. Updated document branding.	v1-2-0	01/11/2017
Made stylistic updates to improve clarity and organization. Updated example data. Added a verification checklist.	v1-1-0	01/05/2017
Added baseline specification version number to the document. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file.	v1-0-0	09/22/2016
Updated document formatting. Updated inactive_date, start_date, and end_date field descriptions to say that YYYYMMDD is an acceptable format. Reorganized some sections to follow a more standard structure. Removed Appendix, which contained redundant examples of data.	Not applicable.	06/16/2016

Entity Attribute Data Transfer

Entity attribute data allows a utility to send attributes about customers, accounts, service agreements, service points, and premises for tracking or targeting purposes. Based on this information, Oracle Utilities Opower can then create tailored communications and user experiences. For example, utilities may want to use the entity attribute data to designate a group of their customers as having an electric vehicle, being low income, or having installed a heat pump. These customers can then be sent tailored marketing messages and tips for saving energy that best suit them.

Entity attribute data also includes fields related to customer billing and rate attributes.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Entity Attribute Data File Specifications

There are two separate files that you must send to Oracle Utilities: the Entity Attributes Definition file and Entity Attributes Data file. This section describes the distinct data elements that these data files must contain.

> **Note**
>
> The Entity Attributes Definition file must be sent prior to the Entity Attributes Data file. This is because the attributes defined in the definition file are used to validate the records in the data file.

Entity Attribute Definition File

The Entity Attributes Definition file describes the attributes with their allowed values. The following table defines the elements that should be provided in the columns of the file.

Field	Description
entity_type	The core entity type for the key. **Supported Values**: • CUSTOMER Attribute key associated with CUSTOMER. • UTILITY_ACCT Attribute key associated with UTILITY_ACCT. • SERVICE_POINT Attribute key associated with SERVICE_POINT. • PERSON Attribute key associated with PERSON. • ACCOUNT Attribute key associated with ACCOUNT. • SERVICE_AGREEMENT Attribute key associated with SERVICE_AGREEMENT. • PREMISE Attribute key associated PREMISE. **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.
key	The key identifying the attribute. The key is case- insensitive (it can be in lower, upper, or any case). See Example Entity Attributes Definition File for common use cases, such as for preferred languages and low-income customers. **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.
key_description	A description of the key and its purpose for the utility. **Type**: VARCHAR. **Can Be Empty?**: Yes. **Default**: Not applicable.

Field	Description
allowed_value	A possible value for the key. • Be consistent with upper and lower case. For example, do not use both `True` and `true`. Choose one. • Must include only letters, underscores, hyphens, and periods. • Special characters are not accepted. If you are not using enumerated values, then use one of the reserved keywords to define the type of value. Enumerated values are any fixed sets of known values. For example, for the attribute key LANGUAGE, known enumerated values could be "ENGLISH", "FRENCH". **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.
allowed_value_description	A description of the allowed value and its purpose for the utility. **Type**: VARCHAR. **Can Be Empty?**: Yes. **Default**: Not applicable.
allowed_min_value	The minimum value allowed for this key. This field is applicable only if the `allowed_value` field above is set to <NUMBER>. This field will be ignored for all other value types. Both, either, or neither of `allowed_min_value` and `allowed_max_value` can be populated. **Type**: DECIMAL (15,4). See Supported Decimal Formats for details. **Can Be Empty?**: Yes. **Default**: Not applicable.
allowed_max_value	Maximum value allowed for this key. This field is applicable only if the `allowed_value` field above is set to <NUMBER>. This field will be ignored for all other value types. Both, either, or neither of `allowed_min_value` and `allowed_max_value` can be populated. **Type**: DECIMAL (15,4). See Supported Decimal Formats for details. **Can Be Empty?**: Yes. **Default**: Not applicable.
source	Source of the key. For example, this should be UTILITY if the utility is sending the data. **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.

RESERVED Keywords in Definition File

Keyword	Allowed Value Details	Examples of Allowed Values	Examples of Not Allowed Values
<NUMBER>	**Type**: DECIMAL(15,4). See Supported Boolean Formats for more details.	• 123.89 • 0.01234 • -2147483647 • 2147483647 • 100000000.00	• "123Test" • "stringValue" • "alphaNumeric data" • 1.234,56 • 100000000,00 • 100,000,000.00 • 100.000.000,00 • 100_000_000.00
<DATETIME>	**Type**: DATETIME. See Date and Time Handling for more details.	2007-06-05T12:30:00-America/New_York	2016-10-12 09:12:00 -7:00
<BOOLEAN>	**Type**: BOOLEAN. See Supported Boolean Formats for more details.	• TRUE • T • F • FALSE • 0	• on • off • TR • ACCEPTED • NA • N/A • NULL • NONE

Example Entity Attributes Definition File

The table below is an example of the Entity Attribute Definition File data containing different keys and descriptions. These are meant as examples only. Utilities may use these allowed value pairs or others of their choosing.

key	key_description	entity_type	allowed_value	allowed_value_description	allowed_min_value	allowed_max_value	source
language	The preferred language of the customer	CUSTOMER	English	Tags a customer to receive communications in English			UTILITY
language	The preferred language of the customer	CUSTOMER	French	Tags a customer to receive communications in French			UTILITY
has_ev	Customer has electric vehicle	CUSTOMER	<BOOLEAN>	True/Yes if customer has electric vehicle			UTILITY

key	key_description	entity_type	allowed_value	allowed_value_description	allowed_min_value	allowed_max_value	source
event_date	Date of start of the event	CUSTOMER	<DATETIME>	Actual event start date. Does not represent event creation date.			UTILITY
performance_dollar	Rebate earned during this period of time.	CUSTOMER	<NUMBER>	Rebates should be negative or zero amounts. No positive amounts should be provided.	-99999	0	UTILITY
total_dues	Total dues	CUSTOMER	<NUMBER>			9999.99	UTILITY
tax_charge	Tax charged	CUSTOMER	<NUMBER>		100		UTILITY
household_income	Income for current calendar year	CUSTOMER	<NUMBER>				UTILITY

Entity Attributes Data File

The Entity Attributes Data file lists the attributes and their respective values. The table below defines the elements that must be included in the columns of the file.

Field	Description
entity_id	Entity identifier. The entity ID should be established following the rules and requirements in the Legacy Billing Data File Specifications. **Type**: VARCHAR **Can Be Empty?**: No.

Field	Description
entity_type	The core entity type for the key. **Supported Values**: • CUSTOMER Attribute key associated with CUSTOMER. • UTILITY_ACCT Attribute key associated with UTILITY_ACCT. • SERVICE_POINT Attribute key associated with SERVICE_POINT. • PERSON Attribute key associated with PERSON. • ACCOUNT Attribute key associated with ACCOUNT. • SERVICE_AGREEMENT Attribute key associated with SERVICE_AGREEMENT. **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.
key	Key identifying the attribute. See Example Entity Attributes Data File for common use cases, such as for solar customers and low-income customers. **Type**: VARCHAR **Can Be Empty?**: No. **Default**: Not applicable.
value	Value for the attribute. See Example Entity Attributes Data File for common use cases. If you are not using enumerated values, then use one of the reserved keywords to define the type of value. See RESERVED Keywords in Definition File for details. Enumerated values are any fixed sets of known values. For example, for the attribute key LANGUAGE, known enumerated values could be ENGLISH, FRENCH. **Type**: VARCHAR **Can Be Empty?**: No.
start_datetime	The effective start date and time of the specified attribute value. This field may be left NULL if the value is effective from the beginning of time. The start datetime value is inclusive (for example, from 2019-02-27T00:00:00-America/New_York, and onwards). See Date and Time Handling for more guidance. **Type**: DATETIME (YYYY-MM-DD<T>hh:mm:ss-<timezone>) **Can Be Empty?**: Yes.

Field	Description
end_datetime	The date and time on which the specified attribute value is no longer effective. This datetime value is not part of the period covered by the attribute key. This field may be left NULL if the value is effective indefinitely. The end datetime value is exclusive (for example, until 2024-04-13T00:00:00-America/New_York, excluding that instant). See Date and Time Handling for more guidance. **Type**: DATETIME (YYYY-MM-DD<T>hh:mm:ss-<timezone>) **Can Be Empty?**: Yes.
source	Source of the key. For example, this should be UTILITY if the utility is sending the data. **Type**: VARCHAR. **Can Be Empty?**: No. **Default**: Not applicable.

Example Entity Attributes Data File

The table below is an example of entity attributes data. It is intended as an example only.

entity_id	entity_type	key	value	start_datetime	end_datetime	source
779222413	CUSTOMER	has_ev	True	2018-06-09T12:30:00-America/New_York		UTILITY
822022136	CUSTOMER	has_ev	True	2019-06-05T12:30:00-America/New_York		UTILITY
530432863	CUSTOMER	low_income	True		2007-06-05T12:30:00-America/New_York	UTILITY
572410557	CUSTOMER	has_solar	True	2022-06-05T12:45:00-America/New_York		UTILITY
579494002	CUSTOMER	language	French	2021-11-10T10:30:00-America/New_York	2022-11-09T10:30:00-America/New_York	UTILITY
662410557	CUSTOMER	language	English	2022-12-15T11:45:00-America/New_York		UTILITY
662410557	CUSTOMER	performance_ dollars	-10.35	2022-12-15T11:45:00-America/New_York		UTILITY
662410557	CUSTOMER	performance_value	0.445			UTILITY

entity_id	entity_type	key	value	start_datetime	end_datetime	source
662410557	CUSTOMER	event_date	2023-01-15T00:00:00-America/New_York	2022-03-05T12:30:00-America/New_York		UTILITY

Entity Attribute Data Order and File Format

Entity attribute data is processed in a particular order and file format.

Data Order

The Oracle Utilities Opower platform processes files according to the order received and dated. The last record received for a row of identical date and time will be considered a replacement and only the final updates will be retained in the system. Therefore, if multiple files are received on the SFTP server in the same timeframe (as would be done for historical data files), then the files must be named so that the file with the oldest data has the oldest date/timestamp, allowing the system to process them in the appropriate order. See Entity Attribute File Name Conventions for more information.

File Format

The following file format conventions apply to all files generated for transfer:

- All data files must be sent as tab-delimited text in .tsv or .txt format. The same number of tabs must exist in each row.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported. **Note**: The only fields that do not yet support non-Latin characters are `premise_id` and `email`. Contact your Delivery Team if you need to send non-Latin characters for these fields, so that a workaround can be provided.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Entity Attribute File Name Conventions

All entity attribute data files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Entity Attribute Definition File:

opwr_<utility>_[res_|non_res_]<specification version number>_core_entity_attribute_definition_<yyyyMMddHHmmss>.gz

Entity Attribute File:

opwr_<utility>_[res|non_res_]<specification version number>_core_entity_attribute_<yyyyMMddHHmmss>.gz

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Entity Attribute Data Version History.
- **Data Type**: The filename must contain the words *core_entity_attribute* or *core_entity_attribute_definition* to indicate the type of data it contains.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All data files must be sent in tab-delimited text in .tsv or .txt format. All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0, your utility is named "The Great Energy Company”, and your data applies to a residential customer population. An example of your utility’s entity attribute definition specification file name would be as follows:

opwr_gec_res_v1-0-0_core_entity_attribute_definition_20210316131415.gz

An example of your utility’s entity attribute specification file name would be:

opwr_gec_res_v1-0-0_core_entity_attribute_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility’s entity attribute definition specification file name would be as follows:

opwr_gec_ res_v1-0-0_core_entity_attribute_definition_20210316131415.pgp

An example of your utility’s entity attribute specification file name would be as follows:

opwr_gec_res_v1-0-0_core_entity_attribute_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.core.entity.attribute.definition.20210316131415.gz

Entity Attribute Data Version History

The following table lists updates to the entity attribute data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Entity Attribute File Name Conventions for more information.

Update	Specification Version Number	Date
Update components of DATETIME references in Entity Attribute Definition File and Example Entity Attributes Data File to use underscores. Update `entity_type` column description in Entity Attribute Definition File to include PREMISE as a supported value.	v1-3-0	06/27/2025
Update naming convention in the Entity Attribute File Name Conventions section.	v1-2-0	07/31/2024
Initial version.	v1-1-0	12/03/2023

Interval Data Transfer

Interval data allows Oracle Utilities Opower to deliver personalized energy insights to customers through printed reports, email messages, alerts, and web applications. This documentation provides detailed requirements for generating files with interval usage data and transferring those files to Oracle Utilities on an ongoing basis.

> **Note**
>
> Terms used by Oracle Utilities to describe data fields might differ in meaning from your internal system's terminology. Data fields are defined in this documentation to facilitate communication by ensuring use of common definitions. In order to avoid delays in launching your program, it is important that you read this content carefully and quickly contact your Oracle Utilities Delivery Team if you have any questions.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Interval Data Integration Process

Integration of your interval data into the Oracle Utilities Opower platform involves the following phases: sample historical file integration and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The

files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover.

Oracle Utilities recommends that you extract data from your production billing system. This ensures that the customer contact information and energy usage data you transfer to Oracle Utilities is as accurate and current as possible.

For smart meter interval data, Oracle Utilities recommends that meter read and usage data be retrieved from the utility Meter Data Management System (MDMS) after the validation, estimation, and editing (VEE) process has been performed. This ensures that data displayed by Oracle Utilities applications is consistent with data received by the billing system for final customer billing and processing.

> **Tip**
>
> Oracle Utilities recommends that the data be of billing quality when possible. This is because we treat the data provided as the system of record for information, and we do not alter it for presentation to the customer or performing calculations. Lower quality of data will affect the customer experience and analytic calculations that may be presented to a customer.

Full Historical File Integration

Historical data only needs to be provided once at the beginning of the project. Oracle Utilities requests data history for all currently active accounts for at least 13 months in order to support robust data disaggregation modules. If historical data is not available for a full 13 months of history, Oracle Utilities requests the full amount of historical data that is available.

Incremental File Integration

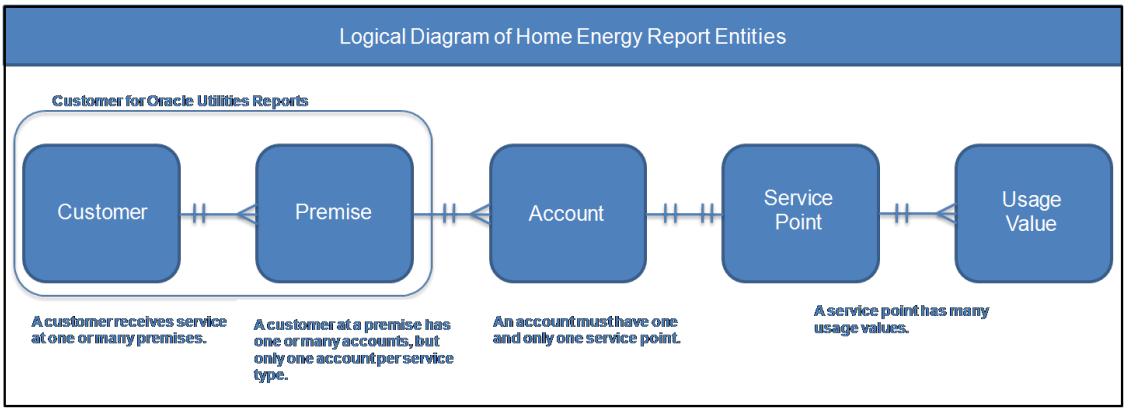
Interval data should be transferred as frequently as possible, but no less than on a daily basis to ensure timely display of data for online applications. Iterative files should contain new usage data since the previous file.

Interval Data Definitions

Oracle Utilities requires the following data to be sent from the utility: billing data (including data about the customer, premise, service point, and billed usage) and interval data (specifically interval usage values). This documentation focuses specifically on the file formats and requirements for obtaining the interval usage values. It is assumed that this information will augment data provided by your billing system.

Logical Diagram of Data Entity Relationships

The diagram below illustrates the logical relationships between the entities and is not intended to translate the data model for any particular utility or Oracle Utilities Opower system.



It is assumed that all of the data elements described in this diagram will be provided. Interval data will augment the customer, premise, account, and billing data that are provided from the billing system. The common data element that Oracle Utilities uses to unify the billing to interval data is the unique service point identifier. Therefore, it is of utmost importance that the service point identifier definition is consistent across both billing and interval data extracts.

Service Point

A service point is an unchanging identifier that is specific to a service type at a location. A service point is the logical “hole in the wall” through which the gas or electricity flows. The `service_point_id` must remain unchanged from the time of construction until the location is destroyed.

A customer who has both gas and electric accounts must have one service point for each account. In the case of an individual with service at multiple locations, this person would be treated as a separate customer at each of the person's locations. For this reason, Oracle Utilities requests both an identifier for the customer, `customer_id` and an identifier for the service location, `premise_id`. Contact your Delivery Team if your customers have multiple service points per service type at a location.

- A service point is logically a meter, but the `service_point_id` should not be the actual meter id. Meters can be replaced or upgraded, but the `service_point_id` must remain unchanged even when meters are replaced or customers at the location change.
- Each account is associated with one and exactly one service point. Likewise, a service point must be associated with only one account at any given point in time, but over the history of the service point it can be associated with different accounts.
- All service points for a customer with multiple service types must be at the same premise.
- The `service_point_id` can be the `premise_id` concatenated to the service type (`G`, `E`, `W`) or concatenated to a dash and a service number (`-1`, `-2`, `-3`, ….).

Usage Value

A usage value is the amount of energy consumed at a specific service point over a given period of time. The term “usage value” refers to the amount of energy delivered to or generated at a service point during the time interval that is specified. Usage values do not represent the cumulative value displayed on the meter at the end of the interval.

Interval Usage Values

The aggregation of the interval usage values for a service point should illustrate a linear progression of the energy usage for that service point over a period of time.

- Any gaps in usage values should indicate an actual gap in service and should be a rare occurrence.
- There should be no overlaps in usage periods for a service point.
- All reads in a given file should be equivalent in interval size. For example, all reads within a file should be 15 minute intervals, and not a mix between 15/30/60 minute intervals.
- Data should be provided in fixed intervals (for example, always ending at fixed 15 minute increments of the hour.) Valid time values are 0015, 0030…2345, 0000.
- Data should be grouped by `service_point_id` and `date`, and ordered by `time`.

Important: If a service point has net-metering, only one usage value for each interval is supported. If you have values for gross consumption and gross generation, and they are both associated with the same service point ID, you must combine them into net consumption before transferring that data to Oracle Utilities.

Interval Data File Specifications

This section describes the distinct data elements that must be included in the interval data file that you send to Oracle Utilities. Additionally it describes how to handle time fields, daylight saving time, and file format requirements, and file transfer and size limits.

Field Descriptions

The following table defines the elements that should be provided in the columns of the interval usage data file.

Field	Description
service_point_id	Utility's unique identifier for the service point. This is the same as the `service_point_id` in the billing data files. See Interval Data Definitions for more information. **Type**: VARCHAR. **Can Be Empty?**: No. **Default**: Not applicable.
usage_value	Usage value amount based on meter read in units determined by meter's units. **Type**: Number. **Can Be Empty?**: No. **Default**: Not applicable.
date	End date of meter read interval. **Type**: VARCHAR (yyyyMMdd). **Can Be Empty?**: No. **Default**: Not applicable.

Field	Description
time	Time representation in 24-hour format in the local time zone. The timestamp represents the end of the read, not the beginning of the read. The following time intervals are supported: 5 minute, 15 minute, 30 minute, 60 minute, and 1440 minute (one day). Times should always be in fixed interval periods, and the intervals within the file should always represent the same length of time. For example, within an hourly interval data file, valid interval values are `0000` through `2300`, all ending in `00`. However, for daily interval files denoting an interval of 24 hours, the only valid interval value is `0000`. If you cannot provide ending times, contact your Delivery Team. For more examples on how you can specify the time field, see Time Field Examples below. For guidance on how to handle daylight saving time, see Daylight Saving Time below. **Type**: VARCHAR (HHmm). **Can Be Empty?**: No. **Default**: Not applicable.

Field	Description
units	Unit of measurement for the usage value. **Supported Values**: • A Amps • C Celsius • CCF Hundred Cubic Feet • CF Cubic Feet • CGL Hundred Gallons • CT Count • F Fahrenheit • GAL Galllons • KL Kilolitres • KVA Kilovolt-Ampere • KVAH Kilovolt-Ampere Hours • KVAR Kilovolt-Ampere Reactive • KVARH Kilovolt-Ampere Reactive Hours • KW Kilowatt • KWH Kilowatt-Hours • L Litres • M3 Cubic Meter • MCF Thousand Cubic Feet • MGL Mega Gallons • ML Megalitres • MWH Megawatt-Hours • PERC Percentage • PF Power Factor • TH

Field	Description
	Thermal Units • THERM Thermal Units • V Volts • VA Volt-Ampere • VAH Volt-Ampere Hours • VAR Volt-Ampere Reactive • VARH Volt-Ampere Reactive Hours • W Watt • WH Watt-Hours **Type**: VARCHAR. **Can Be Empty?**: No. **Default**: Not applicable.
is_estimate	A flag to indicate whether the usage read is estimated or not. **Supported Values**: • A Actual read by utility. • E Estimated read. **Type**: VARCHAR. **Can Be Empty?**: No. **Default**: E
UTC_offset	Provide the offset from UTC time for the `service_point_id` in +/-HHmm format. For example, send `-0800` for a `service_point_id` that is 8 hours and 0 minutes behind UTC, or `+0100` for a `service_point_id` that is one hour and 0 minutes ahead of UTC. For the two calendar days that are affected by daylight saving time (DST), send each interval with a DST offset to represent the amount of time lost or gained by the DST event. See Daylight Saving Time below for more information and examples. **Type**: VARCHAR. **Can Be Empty?**: No. **Default**: Not applicable.

Field	Description
service_type	The service type provided by service points of this type. **Allowed Values**: • E Electric • G Gas • W Water • M Miscellaneous • WW Waste water • O Other **Type**: VARCHAR. **Can Be Empty?**: Yes. **Default**: Not applicable.
service_quantity_identifier	An identifier for distinguishing between measured quantities that have identical unit of measure (UOM) combinations. **Example Values**: • ADU Average Daily Usage • COIN Coincident • CONSUMED Consumed • DELIVERED Delivered • NET_USAGE Net Usage • GENERATED Generation • OFFDMD Contract Off Peak Demand • RECEIVED Received **Type**: VARCHAR. **Can Be Empty?**: Yes. If this field is left blank, then it will be assumed that the usages represent `NET_USAGE` values. **Default**: `NET_USAGE`

Time Field Examples

The following table describes several timestamp scenarios that may be used with the time field in the interval usage data file. The scenarios described represent data from meters with 15 minute intervals:

- The first interval of the day such as the interval between midnight and 00:15

- A second interval during the day such as the interval between 11:45 and noon
- The final interval in the day, for example the interval between 23:45 and midnight. (Note that intervals that cover the end of the day end at the beginning of the next day, or 00:00.)

service_point_id	usage_value	date	time	units	is_estimate	utc_offset
12345	10	20140101	0015	KWH	A	-0400
12345	5	20140101	1200	KWH	A	-0400
12345	7	20140102	0000	KWH	A	-0400

Daylight Saving Time

For utilities participating in daylight saving time (DST), two days in the calendar year require additional usage value rules: DST-forward when the time is reset one hour later, and DST-backward when the time is reset one hour earlier.

Warning

It is important to send daylight saving time and UTC offset data to Oracle Utilities as described. Otherwise, confusing information may display in the user interface of certain product features (most notably the Digital Self Service Data Browser) that depend on AMI data reads. If you have questions about daylight saving time, or if there are limitations in what your system can handle, contact your Delivery Team to discuss options.

Note

Oracle Utilities does not support user-configurable DST settings. DST is calculated using standardized calendars.

DST-Forward

In the US, on DST-forward days the 2 a.m. hour is skipped, and the hour after 1 a.m. is 3 a.m. For a 15-minute interval file, the end time should go from `0145` to `0300`, and the `utc_offset` for `0300` should be reduced by one hour.

See the table below for an example of DST-forward. In the last row, the `time` value goes from `0145` to `0300` to indicate a shift forward due to daylight saving time. Notice that the `utc_offset` field value changes from `-0600` to `-0500` to adjust for this change.

service_point_id	usage_value	date	time	units	is_estimate	utc_offset
12345	10	20190310	0115	KWH	A	-0600
12345	9	20190310	0130	KWH	A	-0600
12345	9	20190310	0145	KWH	A	-0600
12345	5	20190310	0300	KWH	A	-0500

> **Note**
>
> In a standard 24-hour period, there are 96 reads of fifteen minutes. On DST-forward days, there will be 92 reads since the four reads in the `0200` hour will be skipped. These 92 reads will be displayed in the **By day** view of the Digital Self Service Data Browser.

DST-Backward

In the US, on DST-backward days the 1 a.m. hour is repeated, and there are two hours between 1 a.m. and 2 a.m. Therefore, there should be two sets of reads between 1 a.m. and 2 a.m. For a 15-minute interval file, the end time should fall back from `0145` to `0100`, and the second `utc_offset` for `0100` should be increased by one hour.

See the table below for an example of DST-backward. In the sixth row, the `time` field value goes from `0145` to `0100` to indicate a fallback due to daylight savings time. Notice that the `utc_offset` field value changes from `-0500` to `-0600` to adjust for this change.

service_point_id	usage_value	date	time	units	is_estimate	utc_offset
12345	10	20181104	1245	KWH	A	-0500
12345	10	20181104	0100	KWH	A	-0500
12345	6	20181104	0115	KWH	A	-0500
12345	5	20181104	0130	KWH	A	-0500
12345	5	20181104	0145	KWH	A	-0500
12345	5	20181104	0100	KWH	A	-0600
12345	3	20181104	0115	KWH	A	-0600
12345	2	20181104	0130	KWH	A	-0600
12345	1	20181104	0145	KWH	A	-0600
12345	1	20181104	0200	KWH	A	-0600

> **Note**
>
> In a standard 24-hour period, there are 96 reads of fifteen minutes. On DST-backward days, there will be 100 reads since the four reads in the `0100` hour will be repeated. These 100 reads will be displayed in the **By day** view of the Digital Self Service Data Browser.

File Format

The following conventions apply to all files generated for transfer:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files. CSV format is also supported.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- Some fields must contain English values. These are fields that have an enumerated set of expected values. For example, the `customer_type` field must contain the values

`AGRICULTURAL`, `RESIDENTIAL`, `SMB`, `OTHER`, or `INDUSTRIAL`. These values must not be translated as the Oracle Utilities Opower system expects these exact strings.

- All files should include a header line containing column names from the tables specified in this document.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- All data fields specified in this document must be included in each file that is generated.
- Reads within the files should be sorted according to `service_point_id`, `date`, and `time` for fastest processing.
- Files and interval reads will be processed in the order that they are received. If multiple files are received within the same timeframe, the files will be sorted according to name prior to loading.
- If duplicate records (same `service_point_id`, `date`, `time`) are received in the same file, Oracle Utilities will keep and use the last row received.
- Do not remove columns or add columns to the middle of the file.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

File Transfer Limit

There is a limit to the file size and file count that the Oracle Utilities Opower data ingest system can process in a day. See File Transfer Limits for details.

Interval Data File Name Conventions

All interval data files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

opwr_<utility>_res_<specification version number>_interval<mm>_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an

underscore. This must match the latest number shown in the Specification Version Number column in Interval Data Version History.

- **Data Type**: The filename should contain the word *interval* to indicate that the file contains interval and not billing data.
- **Interval**: The interval that is specified must be in minutes. The following indicators are accepted: *interval05*, *interval15*, *interval30*, *interval60*, and *interval1440*. To indicate that the interval is 24 hours, use *interval1440*.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company." An example of your utility's filename would be as follows:

opwr_gec_res_v1-0-0_interval15_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's filename would be as follows:

opwr_gec_res_v1-0-0_interval15_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.interval15.20210316131415.gz

Interval reads of different units of measure should be separated into separate files for processing. For example, interval reads with therms and kWh should not be sent in the same file.

Interval Data Version History

The following table lists updates to the interval data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Interval Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update the value of GEN to GENERATED in the `service_quantity_identifier` description in Interval Data File Specifications.	v2-0-5	08/26/2025

Update	Specification Version Number	Date
Add information about support for five-minute interval AMI data reads to Interval Data File Specifications and Interval Data File Name Conventions.	v2-0-4	03/17/2025
Remove `device_label` from Interval Data File Specifications since the same data is being supported by Device Fields in the Premise data transfer specification.	v2-0-3	07/31/2024
Update the `units` field in Interval Data File Specifications to have the same units of measure as the `units_of_measure` field in the Measuring Components section of the Premise data transfer specification.	v2-0-2	12/20/2023
Update all instances of "String" to "VARCHAR" in Interval Data File Specifications.	v2-0-1	07/20/2023
Begin new major version due to significant expansion of the specification. Add the following columns to Interval Data File Specifications to capture more attributes about meters and better support multi-channel interval reads: • `device_label` • `service_type` • `service_quantity_identifier` Remove the `register_id` column. This column has been replaced by the `service_quantity_identifier` and `units` columns.	v2-0-0	07/19/2023
Update Interval Data File Specifications to include a cross-reference to the new File Transfer Limits topic available.	v1-17-0	04/17/2023
Update Interval Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-16-0	04/11/2022
Fix typo in date format in Interval Data File Specifications.	v1-15-0	12/06/2021

Update	Specification Version Number	Date
Update the `date` and `time` fields in Interval Data File Specifications to use clearer date and time references. Clarify that zip files should only contain one file each.	v1-14-0	11/23/2021
Update some terms to use more inclusive language.	v1-13-0	05/12/2021
Update `register_id` description to explain how it maps to another field in the Oracle Utilities premise data file.	v1-12-0	10/09/2020
Update note in the `units` field to clarify the rules regarding units of measurement for service points with multiple registers. Move information about the file retention policy to a more prominent place.	v1-11-0	05/14/2020
Update accepted value representing water gallons for the `units` field. Clarify how to provide data for the `time` field for 24-hour interval data files. Clarify how to handle daylight saving time in the Interval Data File Specifications section.	v1-10-0	02/11/2020
Clarify file extension requirements and expectations.	v1-9-0	11/12/2019
Update the list of supported values for the `units` field in the Interval Data File Specifications section. Add new `register_id` field.	v1-8-0	06/14/2019
Edit sections and headings to simplify content. No substantive changes to technical specifications.	v1-7-0	01/10/2019
Updated file retention section to reflect regional differences.	v1-6-0	05/08/2018
Added statement about valid end-of-line separators.	v1-5-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-4-0	03/02/2018
Added a statement about the utility's data obligations.	v1-3-0	03/30/2017
Added document number and updated branding.	v1-2-0	02/14/2017

Update	Specification Version Number	Date
Added legal notices. Updated document branding.	v1-1-0	01/11/2017
Added baseline specification version number to the document. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file.	v1-0-0	09/26/2016
Added note about maximum file size of 4 gigabytes. Updated document formatting.	Not applicable.	6/3/2016

Program Participation Data Transfer

In addition to impacting a customer's energy usage behavior, Oracle Utilities can measure customer participation in other utility-sponsored energy efficiency programs. Program participation data is needed to do this. Program participation data provides Oracle Utilities with additional segmentation data about the utility's population, which enables the creation of more targeted communications. This data is also required for analysis of energy savings.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Program Participation Data File Specifications

There are two separate program participation data files that you must send to Oracle Utilities: the program participation definition file and the program participation file. This section describes the distinct data elements that these data files must contain.

- Program Participation Definition File
- Program Participation File

Program Participation Definition File

The program participation definition file lists the different energy efficiency programs in which utility customers can participate. The following table defines the elements that should be provided in the columns of the file. Each row in the table is a column in the data file.

> **Note**
>
> The Program Participation Definition file must be sent prior to the Program Participation file. This is because the programs listed in the definition file are used to validate the data in the participation file.

Column Name	Format
program_id	A unique identifier for the program. This can only contain letters, numbers, underscores, hyphens, dashes, and periods. Spaces and special characters are not accepted. Be consistent with upper and lower case. For example, do not use both `Home_Energy_Audit` and `home_energy_audit`. Choose one. **Type**: varchar **Can Be Empty?**: No. **String Limit**: 40 characters.
program_description	A description of the program and its purpose for the utility. **Type**: varchar **Can Be Empty?**: Yes. **String Limit**: 255 characters.
measure	A standardized identifier for a procedure or action that was implemented. Use a standard set of measures and be consistent with upper and lower case. For example, do not use both `Replaced_Light_Bulbs` and `replaced_light_bulbs`. Choose one. **Type**: varchar **Can Be Empty?**: No. **Format**: This field can only contain letters, numbers, underscores, hyphens, dashes, and periods. Spaces may not be used. **String Limit**: 255 characters.
measure_description	A description of the program measure. **Type**: varchar **Can Be Empty?**: No. **String Limit**: 255 characters.
active_date	Date and time that the program began. If an invalid datetime format is used, the record will be rejected. **Type**: datetime (YYYYMMDD HHmm) **Can Be Empty?**: Yes.
inactive_date	Date and time that the program ended. If left blank, this will leave the program valid indefinitely. If an invalid datetime format is used, the record will be rejected. **Type**: datetime (YYYYMMDD HHmm) **Can Be Empty?**: Yes.

Additional Notes

To add new program definitions, add a new row containing the `program_ID` and `measure` pair in the file. To deactivate an existing `program_id` and `measure` value pair, assign an `inactive_date` to an existing row. After this is done, any new assignments of customers to that pair that fall after the inactive date are rejected.

> **Note**
>
> Consult with your Oracle Utilities Delivery Team to make ensure that the products exhibit the correct behavior for each `program_id` and `measure` pair.

Example Program Participation Definition File

The following table is an example of a Program Participation Definition file.

program_id	program_description	measure	measure_description	active_date	inactive_date
Home_Energy_Audit	An audit of a customer's home	Replaced_light_bulbs	As part of the audit, the auditor replaced the customer's light bulbs with compact florescent light bulbs	20130110 1200	20200110 1200
Home_Energy_Audit	An audit of a customer's home	Replaced_HVAC_filters	As part of the audit, the auditor replaced the customer's HVAC filters	20130110 1200	
Recycling	Recycling of old appliance	Freezer_recycled	Customer recycled freezer		
Recycling	Recycling of old appliance	Washer_recycled	Customer recycled washer		

Program Participation File

The program participation file lists the customers and their energy efficiency programs. The following table defines the elements that must be provided in the columns of the file. Each row in the table is a column in the data file.

> **Note**
>
> If Oracle Utilities encounters any program that has not already been defined and sent in a Program Participation Definition file, the corresponding rows in the Program Participation file will be rejected. Oracle Utilities recommends that you confirm with your Delivery Team that the latest definition file has been received before sending any customers with newly-defined program-allowed value pairs. Additionally, if an inactive date is assigned for a program-measure pair, Oracle Utilities will reject any new assignments of customers to that program-measure after that inactive date. Existing customers with the program-measure assignment will not be deactivated.

Column Name	Format
customer_id	Unique identifier for the customer. **Note**: This customer ID should be the same customer ID that is sent in the billed usage files, as described in Customer Fields. **Type**: varchar **Can Be Empty?**: No.

Column Name	Format
premise_id	An identifier for a site occupied by an individual customer. **Note**: This premise ID should be the same premise ID that is sent in the billed usage data files, as described in Customer Fields. **Type**: varchar **Can Be Empty?**: No.
program_id	Key identifier for the program. The `program_id` must be the same as a `program_id` already listed in the corresponding program participation definition file. **Type**: varchar **Can Be Empty?**: No.
measure	A standardized identifier for a procedure or action that was implemented. The `measure` should be the same as a `measure` already listed in the corresponding program participation definition file. **Type**: varchar **Can Be Empty?**: No.
deemed_annual_savings	Energy saved over the course of the year (in kWh, CCFs, or Therms) for the customer. **Type**: floating point **Can Be Empty?**: Yes.
deemed_ annual_savings_unit	Unit of measure for `deemed_annual_savings`. **Type**: enum - KWH, K, THERM, T, CCF, C **Note**: K, T, and C are alternate short form representations of KWH, THERM, and CCF, respectively. **Can Be Empty?**: Yes.
deemed_lifetime_savings	Energy saved over the course of the program (in kWh, CCFs, or Therms) for the customer. **Type**: floating point **Can Be Empty?**: Yes.
deemed_lifetime_savings_unit	Unit of measure for `deemed_lifetime_savings`. **Type**: enum - KWH, K, THERM, T, CCF, C **Note**: K, T, and C are alternate short form representations of KWH, THERM, and CCF, respectively. **Can Be Empty?**: Yes.
measure_life	The number of years over which the benefits of the energy efficiency measure are expected to accrue. **Type**: floating point **Can Be Empty?**: Yes.
start_date	The start date of the customer's enrollment in the program. The start date must fall inside the active and inactive program dates specified in the program participation definition file. Otherwise, it will be marked as invalid. **Type**: datetime (YYYYMMDD HHmm) **Can Be Empty?**: No.

Column Name	Format
end_date	The end date of the customer's enrollment in the program. If this field is left empty, then it is assumed that there is no end date for the customer's enrollment. If this field is populated, then the end date must come after the `start_date` defined above and must fall inside the active and inactive program dates specified in the program participation definition file. Otherwise, it will be marked as invalid. **Type**: datetime (YYYYMMDD HHmm) **Can Be Empty?**: Yes.

Example Program Participation File

The following table is an example of a Program Participation file.

customer_id	premise_id	program_id	measure	deemed_annual_savings	deemed_annual_savings_unit	deemed_lifetime_savings	deemed_lifetime_savings_unit	measure_life	start_date	end_date
4268110141	304892	Home_energy_audit	Replaced_filter	20	KWH	40	KWH	1	201301011200	
8427500652	492783	Home_energy_audit	Replaced_light_bulbs	10	KWH	35	KWH	1.2	201301011200	
3934977649	622774	Recycling	Washer_recycled						201301011200	
1080095742	654078	Recycling	Washer_recycled						201301011200	
7611477224	208315	Recycling	Freezer_recycled						201301011200	

Program Participation Data File Format

All program participation data files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- All rows should have the same number of tabs even if some fields have null values. It is important that null values are represented by empty strings and not denoted by **/N**, **NA**, **null**, or another value.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.

Program Participation Data File Name Conventions

All program participation data files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Program Participation Definition File:

opwr_<utility>_res_<specification version number>_energy_efficiency_program_definition_<yyyyMMddHHmmss>.<gz or zip>

Program Assignment File:

opwr_<utility>_res_<specification version number>_customer_energy_efficiency_program_<yyyyMMddHHmmss>.<gz or zip>

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Program Participation Data Version History.
- **Data Type**: The filename identifies the data type that the file contains.
- For the program participation definition file, use *energy_efficiency_program_definition*.
- For the program participation file, use *customer_energy_efficiency_program*.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Naming Example

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company.” An example of your utility’s file name would be as follows:

opwr_gec_res_v1-0-0_energy_efficiency_program_definition_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's file name would be as follows:

opwr_gec_res_v1-0-0_energy_efficiency_program_definition_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.customer_energy_efficiency_program.20210316131415.gz

Program Participation Data Verification Checklist

Use the checklist in this section to ensure you have completed all required actions before sending the data files to Oracle Utilities.

File Format Checks

- The filename follows the appropriate pattern. Program Participation Data File Name Conventions
- The file is tab-delimited.
- The same number of tabs exists in each row.

Header Row Checks

Column headers must match exactly the headers shown below.

Program Participation Definition File:

```
program_id program_description measure measure_description active_date
               inactive_date
```

Program Participation File:

```
customer_id premise_id program_id measure deemed_annual_savings
               deemed_annual_savings_unit deemed_lifetime_savings
deemed_lifetime_savings_unit
               measure_life start_date end_date
```

Data Field Checks

- All data fields in the Program Participation Definition file follow the rules described in this document. See Program Participation Definition File for more information.
- All data fields in the Program Participation file follow the rules described in this document. See Program Participation File for more information.
- Null values are empty strings. (They must be not denoted by /N, NA, null, or some other non-supported notation.)

Program Participation Data Version History

The following table lists updates to the program participation data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Program Participation Data File Name Conventions for more information.

Update	Specification Version Number	Date
Clarify `start_date` and `end_date` definitions in the Program Participation File specification.	v1-14-0	10/10/2023
Update Program Participation Definition File to streamline the column descriptions and list the string limit for the `program_id` column.	v1-13-0	09/07/2022
Update Program Participation Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-12-0	04/11/2022
Clarify that zip files should only contain one file each.	v1-11-0	11/23/2021
Clarify file extension requirements and expectations.	v1-10-0	11/12/2019
Edit sections and headings to simplify content. No substantive changes to technical specifications.	v1-9-0	01/10/2019
Updated file retention section to reflect regional differences.	v1-8-0	05/08/2018
Added statement about valid end-of-line separators.	v1-7-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-6-0	03/02/2018
Updated the character limit details of the measure field.	v1-5-0	06/26/2017
Added a statement about the utility's data obligations.	v1-4-0	03/31/2017
Added document number and updated branding.	v1-3-0	02/13/2017
Added legal notices. Updated document branding.	v1-2-0	01/11/2017
Made stylistic updates to improve clarity and organization. Updated example data. Added a verification checklist.	v1-1-0	01/05/2017

Update	Specification Version Number	Date
Added baseline specification version number to the document. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file.	v1-0-0	10/07/2016

Rates Data Transfer

Rates data allows Oracle Utilities Opower to deliver personalized energy cost insights to utility customers through print reports, email messages, user-friendly cost graphs, rate plan comparisons, and other such features. This information helps customers better understand the cost implications of their energy use, and empowers them to be more energy efficient.

To make rates-based insights available, Oracle Utilities Opower requires the following data feeds from utilities:

- Rates data files containing rate plan and pricing information. See Rates Data File Specifications Overview for details.
- Billed usage data files containing account and customer information. See either Billing Data Transfer or Legacy Billing Data Transfer. Your Delivery Team will work with you to identify which data specification is the correct one for your program.
- Interval data files containing information about how much energy a customer at a particular service point consumed in a given interval of time. See Interval Data Transfer for details.

Additionally, during the setup phase of a new program launch or expansion, your Delivery Team will work with you to follow a rates modeling process to develop the proper rate plans and compute personalized cost insights for customers. For more information about how rates are modeled, see Rates Modeling.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Rates Data Integration Process

Your Delivery Team will use the following process to set up and configure utility-specific rates:

1. **Scoping**: A product expert is assigned to review your rate plans and confirm the applicable rate components for those plans with the support of your rates subject matter expert. See Rates Modeling for more information.
2. **Calculation Logic Definition**: Once the rate plans in the scope are solidified, the Delivery Team works with you to determine the calculation logic for each rate plan before implementation.
3. **Data Extract Build and Analysis**: The Delivery Team defines the rate data files that are applicable for your utility based on the rate plans that will be supported and the calculation

logic required. You work with the Delivery Team to build the data files required. The Delivery Team analyzes this data to confirm the quality standards are met for consumption by the Oracle Utilities Opower Rate Engine. This process generally consists of two phases:

- **Phase 1 - Full Historical File Integration**: For historical cost data display and rate analysis, the utility should perform one-time data file transfer for all periodic and historical data files that are applicable. The length of the history will be dictated by each utility's requirements, but it should coincide with the length of history for which customer and billed usage data is being provided from the utility production billing system. This is typically January 1st of the previous calendar year to the present. For example, if your Delivery Team planned to launch your program for the first time on December 1, 2016, the team would need data going back to January 1, 2015.
- **Phase 2 - Incremental File Integration**: Ongoing data transfer frequency for each of the data elements is dependent on the frequency in which the data changes. Your Delivery Team will work with you to determine the data transfer frequency (weekly, daily) that is required based on the rate structures being used and the likelihood of change to prices or other configurable factors.

4. **Configuration and Quality Assurance**: You provide the Delivery Team with a set of sample test data and calculation for validation of the implemented logic. The Delivery Team configures the rate plans as defined in the definition phase and tests the configurations using the test data sets provided.
5. **User Acceptance**: The Delivery Team works with you to conduct user acceptance testing of the calculations being presented by the end product prior to rolling it out.

Supported Rate Components

Oracle Utilities supports a variety of rate components, which can be mixed and matched to create custom "rate plans" for each utility. The sections below describe the supported rate components. A member of the Delivery Team will work with you to determine which rate components are applicable for your utility's implementation. This process will include defining the unique identifiers used for each individual component for use throughout subsequent file specifications. This refers to the `rate_component` column used throughout this document.

Rates Data File Specifications Overview

There are several rates data files that Oracle Utilities Opower requires in order to provide personalized cost insights to customers. These files serve as inputs to the Oracle Utilities Opower Rate Engine, a software computing model that calculates the cost insights based on each customer's rate plans and energy use. Your Delivery Team will work with you to determine which files are applicable to your situation, to construct sample files when necessary, and to test that the files are correct and are being transferred as expected.

Overview of Files

The data you provide in each file should be limited to the fields described in each file's specification. Including additional fields will complicate the data transfer process, and the data will not be used. The responsibility of who must provide each file varies, and not every file is required for every type of rate plan.

File	Responsibility
Rate Attributes File The Rate Attributes file contains a series of key value pairs that define the account-specific attributes for a customer. When collected together, these attributes can be used to associate a customer account with the appropriate rate plan and rate options, as well as characteristics like budget billing, solar, and peak time rebates.	Utility
Rate Prices File The Rate Prices file specifies the cost to apply to each component of a rate plan with which a customer is associated. The Rate Prices file includes fields for the rate plan, rate components, charges or credits, start and end dates, and other fields as applicable.	Utility
Holidays and Seasons File The Holidays and Seasons file defines the different holidays (such as local or national holidays) and seasons (such as summer and winter) throughout the year when the price of energy varies. Your Delivery Team will work with you to determine whether this file is applicable to your situation.	Utility
Rate Tier Definitions File The Rate Tier Definitions file defines the lower and upper bounds of each tier in a tiered rate plan. Your Delivery Team will create the file and work with you to gather the necessary inputs.	Oracle, with utility input
Rate Period Definitions File The Rate Period Definitions file defines the different "periods" or times of day, season, or year when the cost of energy varies. It is commonly used for time of use (TOU) rate plans. Your Delivery Team will create the file and work with you to gather the necessary inputs.	Oracle, with utility input
Peak Time Rebates Files The Peak Time Rebate files contain information about peak events when weather is more extreme than usual and the cost of energy is higher. These files allow Oracle Utilities to calculate how much money a customer spent during a peak event. • Baselines File • Peak Time Rebates File	Utility
Baselines File The baselines file contains any energy use baselines that should be used to determine the pricing for tiered and peak time rebate rate plans. You Delivery Team will work with you to determine whether this file is applicable to your situation.	Utility

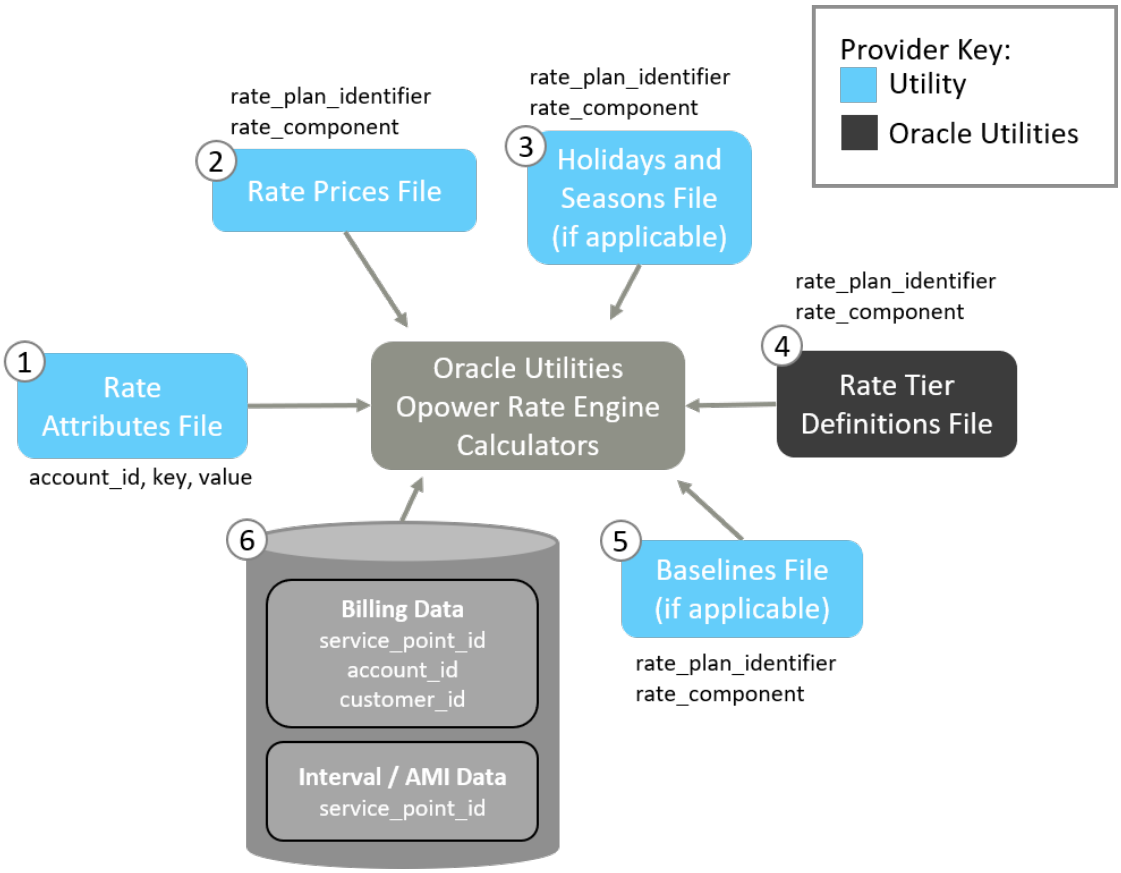
Example Scenarios

The files required for your program will vary slightly depending on the rate plans that must be modeled. Use the example scenarios below to better understand which files you will need to send, and contact your Delivery Team if you have any questions.

Tiered Rate Plan

In a tiered rate plan, customers are charged based on how much energy they use within certain tiers. For example, residential electricity customers could be charged one rate for the first 100 kWhs they consume in a bill period, a higher rate for the next 30 kWhs, and a higher rate for the next 70 kWhs. The graphic below shows the files and data fields involved in a

tiered rate plan, along with who must provide each file. The `rate_plan_identifier` and `rate_component` fields form the common link between most of the files.



1. Rate Attributes File

The Rate Attributes file identifies the `account_id` of customers and the rate plan attributes that each account is associated with. For example, as shown below, a key could be `RATE_PLAN_CODE` and a value could be `E-RES` to associate a customer account with a residential electricity rate plan. Additionally, other key-value pairs can be used to associate a customer's account with different rates based on their geographic location, such as whether they live inside or outside city limits. See Rate Attributes File for more information and additional examples, including for solar customers.

account_id	key	value	start_date	end_date	is_utility_account_attribute_deletion
902101	CITY_LOC	IN	20191208		false
902101	RATE_PLAN_CODE	E-RES	20191208		false
204011	CITY_LOC	IN	20191208		false
204011	RATE_PLAN_CODE	E-RES	20191208		false
309022	CITY_LOC	IN	20191208		false
309022	RATE_PLAN_CODE	E-RES	20191208		false
309023	CITY_LOC	OUT	20200315		false

account_id	key	value	start_date	end_date	is_utility_account_attribute_deletion
309023	RATE_PLAN_CODE	E-RES	20200315		false
204336	RATE_PLAN_CODE	E-RES	20200315		false
204336	BUDGET_BILLING	1	20200315		false

2. Rate Prices File

The Rate Prices file defines the price of energy for each "rate component" of a rate plan—including the price of each "tier" of the rate plan. A rate component is a line item or charge in the rate plan. In the example below, there are five incrementing charges for each tier in the plan, as well as other charges for components like regulatory fees and power supply adjustments. Note that the price of a tier can start and end on a specific date. This can be done when a utility needs to update the tier and replace it with a different price. See Rate Prices File for more information and additional examples.

rate_plan_identifier	rate_component	rate_group	season	price_type	tier	price	effective_start_date	effective_end_date
E-RES/IN-CITY	CBC-CAP			CHARGE		0.00144	20191101	
E-RES/IN-CITY	CBC-EEP			CHARGE		0.00336	20191101	
E-RES/IN-CITY	CBC-STL			CHARGE		0.00123	20191101	
E-RES/IN-CITY	ENERGY			CHARGE	1	0.02901	20180101	
E-RES/IN-CITY	ENERGY			CHARGE	2	0.05833	20180101	
E-RES/IN-CITY	ENERGY			CHARGE	3	0.07815	20180101	
E-RES/IN-CITY	ENERGY			CHARGE	4	0.09315	20180101	20200401
E-RES/IN-CITY	ENERGY			CHARGE	4	0.08315	20200401	
E-RES/IN-CITY	ENERGY			CHARGE	5	0.10815	20180101	
E-RES/IN-CITY	PSA		WINTER	CHARGE		0.02894	20191201	
E-RES/IN-CITY	PSA		SUMMER	CHARGE		0.02894	20191201	
E-RES/IN-CITY	REG			CHARGE		0.01343	20191201	

3. Holidays and Seasons File

The Holidays and Seasons file defines the different holidays (such as local or national holidays) and seasons (such as summer and winter) throughout the year when the price of energy varies. In the example below, the winter and summer seasons are defined, along with a

national holiday in July. This file may not be applicable to your situation if your utility does not change the cost of energy based on seasons or holidays. See Holidays and Seasons File file for more information.

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_date	start_time	event_date	effective_start_date
E-RES/IN-CITY	ENERGY	WINTER				MONTH	8	1001	0		20200101
E-RES/IN-CITY	ENERGY	SUMMER				MONTH	4	601	0		20200101
E-RES/IN-CITY	ENERGY		HOLIDAY			DAY	1		0	20200703	

4. Rate Tier Definitions File

The Rate Tier Definitions file defines the lower and upper bound of each tier in the plan. In the example below, there are five tiers, and each tier has an upper and lower bound. The price of energy increments when a customer's energy use moves from one tier to the next. See Rate Tier Definitions File for more information and additional examples.

rate_plan_identifier	rate_component	tier	lower_bound	upper_bound	effective_start_date	effective_end_date
E-RES/IN-CITY	ENERGY	1	0	501	20200101	
E-RES/IN-CITY	ENERGY	2	501	1001	20200101	
E-RES/IN-CITY	ENERGY	3	1001	1501	20200101	
E-RES/IN-CITY	ENERGY	4	1501	2501	20200101	
E-RES/IN-CITY	ENERGY	5	2501		20200101	

5. Baselines File

The baselines file contains any energy use baselines that should be used to determine the pricing for a tiered rate plan. For example, your utility could specify an energy use baseline of 500 kWh for a group of customers, and then charge those customers a specific rate for any energy that is consumed within that baseline. If customers exceed 100% of the baseline, they would then be moved to the next tier and charged a higher rater. The baselines file can specify multiple baselines for different customer demographics or zip codes. For example, one set of baselines could apply to low-income customers, and another set could apply to customers within a specific zip code. This file may not be applicable if your utility does not use energy use baselines to determine tiered rates. See Baselines File for more information.

The asterisk (*) in this example is a wildcard indicating that the usage baseline is applicable to all rate plan identifiers.

account_id	season	usage_baseline	resolution	rate_plan_identifier	rate_component	start_date	end_date	premise_id
902101	SUMMER	0.5900000	DAY	*	TIERED	20180401		9671204723
204011	SUMMER	26.338000	DAY	*	TIERED	20200507		3274021769
309022	SUMMER	7.7000000	DAY	*	TIERED	20190501		1422197854
309023	WINTER	29.400000	DAY	*	TIERED	20190101		1492198469
309024	WINTER	1.9200000	DAY	*	TIERED	20181101		8989765498
309116	SUMMER	16.500000	DAY	*	TIERED	20190501		13576452167

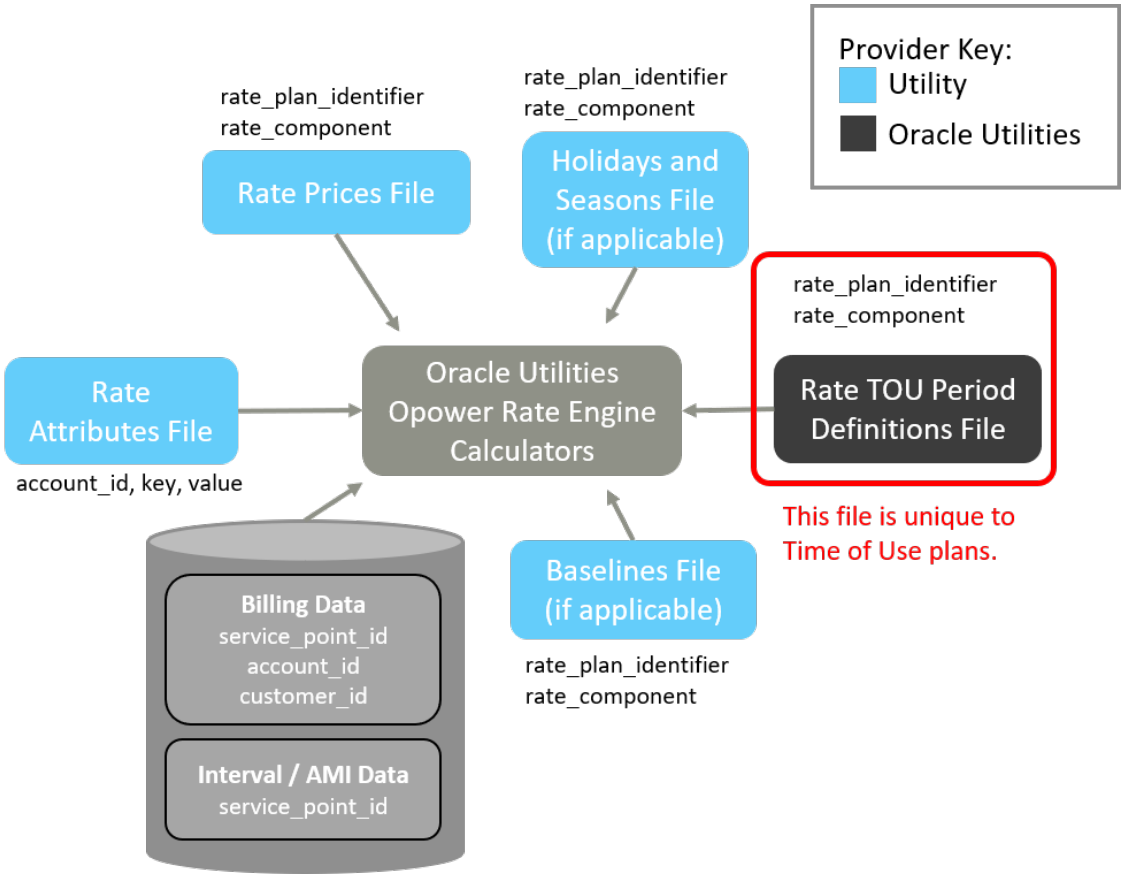
6. Billing and Interval Data

The billing data store includes the fields necessary to associate a `customer_id`, `account_id`, and `service_point_id` with the tiered rate plan. The interval data store provides the `service_point_id` field to associate a customer's service point with the tiered rate plan.

Time of Use Rate Plan

In a time of use (TOU) rate plan, customers are charged based on the time of day when energy is used. TOU insights can be shown in various Oracle Utilities Opower products and features such as the High Bill Alert AMI communication and the Load Shifting Time of Use 101 module.

For this type of rate plan, most of the same files as in a tiered rate plan are applicable. See the tiered rate plan example above for descriptions and examples of each file. However, a TOU plan has one primary difference: a TOU period definitions file is required instead of a tier definitions file.



The Rate TOU Period Definitions file specifies each time period in the day when the cost of energy varies. For example, as shown in the table below, the file can specify the peak, partial peak, and off-peak time periods for different seasons of the year, days of the week, and even holidays. See Rate Period Definitions File for more information and additional examples.

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_time	effective_start_date
E-RES/IN-CITY	ENERGY	WINTER	WEEKDAY	PEAK		HOUR	3	1500	20190101
E-RES/IN-CITY	ENERGY	WINTER	WEEKDAY	PART_PEAK	1	HOUR	8	700	20190101
E-RES/IN-CITY	ENERGY	WINTER	WEEKDAY	PART_PEAK	2	HOUR	4	1800	20190101
E-RES/IN-CITY	ENERGY	WINTER	WEEKDAY	OFF_PEAK		HOUR	9	2200	20190101
E-RES/IN-CITY	ENERGY	WINTER	WEEKDAY	OFF_PEAK		HOUR	24	0	20190101
E-RES/IN-CITY	ENERGY	WINTER	HOLIDAY	OFF_PEAK		HOUR	24	0	20190101

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_time	effective_start_date
E-RES/IN-CITY	ENERGY	SUMMER	WEEKDAY	PEAK		HOUR	3	1500	20190101
E-RES/IN-CITY	ENERGY	SUMMER	WEEKDAY	PART_PEAK	1	HOUR	8	700	20190101
E-RES/IN-CITY	ENERGY	SUMMER	WEEKDAY	PART_PEAK	2	HOUR	4	1800	20190101
E-RES/IN-CITY	ENERGY	SUMMER	WEEKDAY	OFF_PEAK		HOUR	9	2200	20190101
E-RES/IN-CITY	ENERGY	SUMMER	WEEKDAY	OFF_PEAK		HOUR	24	0	20190101
E-RES/IN-CITY	ENERGY	SUMMER	HOLIDAY	OFF_PEAK		HOUR	24	0	20190101

Rate Attributes File

The rate attributes file contains a series of key value pairs that provide all the account-specific attributes for a customer. When collected together, these attributes can be used to associate an account with the appropriate rate plan and rate options used when calculating the costs for each customer's bill. The rate attributes file can also be used to identify customers who have solar. This data is required and must be sent on an ongoing basis as customers are introduced into the system or as customer data changes. The data must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

> **Note**
>
> The rate attributes file cannot be loaded into the system until after the utility account has been created by your Delivery Team. Utilities should not send the file until the Delivery Team has received the utility account information in the customer/billing file as defined in Legacy Billing Data File Specifications. The common data element that must match across the datasets is the `account_id`. The `account_id` is a common value for applying rate attributes to a particular customer's account.

Field	Description
account_id	Customer account identifier. The account entity should be established following the rules and requirements in Legacy Billing Data File Specifications. **Type**: VARCHAR **Can Be Empty?**: No.

Field	Description
key	Key identifying the attribute. See Examples of Rate Attribute Keys below for common use cases, such as for rate plans, solar customers, and low-income customers. **Type**: VARCHAR **Can Be Empty?**: No.
value	Value for the attribute. See Examples of Rate Attribute Keys below for common use cases. **Type**: VARCHAR **Can Be Empty?**: No.
start_date	Effective start date of the specified attribute value. See Start and End Dates for more guidance. **Type**: DATE(YYYYMMDD) **Can Be Empty?**: No.
end_date	Date on which the specified attribute value is no longer effective. This day is not part of the period covered by the rate attribute. This field may be left NULL if the value is effective indefinitely. **Type**: DATE(YYYYMMDD) **Can Be Empty?**: No.
is_utility_account_attribute_deletion	Deletion flag for the rate attribute. Supported values include `true` and `false`. The default value is `false`. **Note**: The rate attribute deletion feature is disabled by default to ensure that rate attributes are not erroneously deleted. If you want to allow deletions, make a request to your Delivery Team. Note that deleting a rate attribute entry results in the deletion of all future values for the `account_id` and `key` columns associated with that rate attribute. **Type**: STRING **Can Be Empty?**: Yes.

Examples of Rate Attribute Keys

Key	Sample Value	Description
rate_plan_code	Rate-1	Rate plan identifier (for example, B-1, A-1). All customers should have at least one rate_plan_code key value pair.
critical_peak_option	CPP	Set to the code specifying the critical peak option selected (for example, ‘CPP’).
low_income_discount	FERA	Indicates that an account has a low income discount.
budget_billing	1	Use a value of 1 to indicate budget billing and 0 to indicate no budget billing.

Key	Sample Value	Description
SOLAR_TRUE_UP	1	Use a value of 1 to indicate that a customer is using solar technology and should therefore receive insights related to solar power. Note that the attribute name (SOLAR_TRUE_UP) is case sensitive, and that 1 is the only supported value. The solar true up indicates the dates (usually annual) on which a customer's net energy metering billing has an anniversary. The anniversary may be when an annual true-up occurs and balances are paid off or reset. Oracle Utilities Opower expects a new row in the data file for each year that a customer has solar. This data can then be used to calculate monthly and yearly net energy metering billing balances, and to send time-based insights to the customer about their solar usage. See the SOLAR_TRUE_UP examples in the Rate Attributes Sample Data section below for an idea of how the data could look. **Note**: Oracle Utilities Opower will exclude any customer bill periods that fall outside of the true-up period's start and end date. For example, any bills that are partially within the true up period will be excluded. To be included, a start date of a bill period must begin after the true-up period's start_date, and the end date of a bill period must fall within the true-up period's end_date.

Rate Attributes Sample Data

account_id	key	value	start_date	end_date	is_utility_account_attribute_deletion
123	rate_plan_code	Rate-1	20080501		false
123	low_income_discount	FERA	20080501		false

account_id	key	value	start_date	end_date	is_utility_account_attribute_deletion
123	critical_peak_option	CPP	20080501		false
456	rate_plan_code	Rate-1	20090701		false
456	critical_peak_option	CPP	20090701		false
789	SOLAR_TRUE_UP	1	20161011	20171011	false
789	SOLAR_TRUE_UP	1	20171011	20181011	false

Rate Attribute Data File Assumptions

Any key listed in the file should be unique over a given period of time. For example, a customer cannot have more than one rate plan code active over the same time period.

Baselines File

The baselines file contains any usage baselines that should be used to determine the pricing for tiered and peak time rebate rate plans. Baselines should be provided for each customer as new customers are introduced for the rate plan, or as baselines for customers change. The data must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

Baseline Data File Assumptions

- Any baseline listed in the file should be unique over a given period of time. For example, a customer cannot have more than one baseline for the same season.
- A baseline value should always be provided in the same unit of measure as the reads for the service point associated with the account. For example, if the service point is electric and reads are in kWh, the baseline values should be in kWhs.

> **Note**
>
> This data file is only required if you have tiered and Peak Time Rebate rate plans. Your Delivery Team will work with you to determine if it is required in your situation.

Column Name	Description
account_id	Account identifier. **Type**: VARCHAR **Can Be Empty?**: No.

Column Name	Description
season	Season for which the baseline is applicable. This may be NULL if the baseline is not season-specific. **Type**: ENUM **Allowed Values**: • WINTER • SPRING • SUMMER • FALL **Can Be Empty?**: Yes.
usage_baseline	Baseline value. **Type**: DECIMAL (12,6). See #unique_240 for details. **Can Be Empty?**: No.
resolution	Resolution at which the baseline value applies. **Type**: ENUM **Allowed Values**: • QUARTER_HOUR • HALF_HOUR • HOUR • DAY • BILLING • MONTH • YEAR **Can Be Empty?**: No.
rate_plan_identifier	The rate plan for which the baseline is applicable. This may be a wildcard (*) if the baseline applies to all rate plans for which the Service Agreement (SA) is eligible. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the rate component to which the baseline applies. **Type**: VARCHAR **Can Be Empty?**: No.
start_date	Effective start date of the specified baseline value. **Type**: DATETIME. See #unique_241 for details. **Can Be Empty?**: Yes.
end_date	Date on which the specified baseline value is no longer effective. (This day is not part of the period covered by the specified baseline value.) This field may be NULL if the value is effective indefinitely. **Type**: DATETIME.See #unique_241 for details. **Can Be Empty?**: Yes.

Baselines Sample Data

The Baselines File contains any usage baselines that should be used to determine the pricing for tiered and peak time rebate rate plans. The table below shows an example of how the baselines data might look.

account_id	season	usage_baseline	resolution	rate_plan_identifier	rate_component	start_date	end_date
123	SUMMER	20.1	DAY	R	TIERED	20080501 1200	20081001 1200
123	WINTER	35.5	DAY	R	TIERED	20081101 1200	20090301 1200

Rate Prices File

In order to properly apply prices for cost calculation, Oracle Utilities must receive the prices to charge for each rate attribute, period, and charge. Typically this file is provided once during setup of a new plan and on a periodic basis when prices for existing rate plans change (typically 2-4 times a year). For real time pricing plans, this file should be provided daily as soon as prices become available. This data is required and must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

Column Name	Description
rate_plan_identifier	A unique rate plan for which the price is applicable. This may be a wildcard (*) if the price is applicable to all rate plans. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the rate component for which the price is applicable. This may be a wildcard (*) if the price is applicable to all rate components in the specified rate plan. **Type**: VARCHAR **Can Be Empty?**: No.
rate_group	Applicable for rates that have different prices for different customer groups. **Type**: VARCHAR **Can Be Empty?**: Yes.
season	The season for which the prices are applicable. This is required for seasonal rates. **Type**: ENUM **Allowed Values**: • WINTER • SPRING • SUMMER • FALL **Can Be Empty?**: Yes.
period	The period for which the prices are applicable. This is required for TOU, TOU-tiered, and PDP components. **Type**: ENUM **Allowed Values**: • ON_PEAK • PART_PEAK • OFF_PEAK • CRITICAL_PEAK • NON_CRITICAL_PEAK **Can Be Empty?**: Yes.

Column Name	Description
price_type	The type of price. CHARGE means something that is taken away from the customer (when positive, the usual case). CREDIT means something that is given back (when negative, the usual case). **Type**: ENUM **Allowed Values**: • CHARGE • CREDIT **Can Be Empty?**: No.
tier	The tier for which the price(s) are applicable. This is required for tiered and TOU-tiered rates. **Type**: INTEGER **Can Be Empty?**: Yes.
start_time	The start time of the applicable price in military time (for example, 0800, 1700). Valid values for hourly prices are 0000 to 2300. Do not specify a timezone. The local timezone of the utility is assumed. **Type**: TIME (HHMM) **Can Be Empty?**: Yes.
price	Price to be applied. The prices should always be provided in dollars, not cents. Credits must be preceded by a minus sign (-) to indicate a negative value. **Type**: DECIMAL (12,6). See Supported Decimal Formats for details. **Can Be Empty?**: No.
effective_start_date	The date on which this specified price becomes effective. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.
effective_end_date	Date on which the specified price is no longer effective. (This day is not part of the period covered by the specified price.) A NULL date means the end date is yet unknown and the price is effective indefinitely until an end date is provided. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.

Rate Prices Sample Data

The rate prices file describes the prices to charge for each rate attribute, period, and charge. The table below shows an example of how the rate prices file might look.

rate_plan_identifier	rate_component	rate_group	season	period	price_type	tier	start_time	price	effective_start_date	effective_end_date
Rate-1	TOU_TIERED		SUMMER	ON_PEAK	CHARGE	1		0.20776	20080501	
Rate-1	TOU_TIERED		SUMMER	ON_PEAK	CHARGE	2		0.22023	20080501	
Rate-1	TOU_TIERED		SUMMER	ON_PEAK	CHARGE	3		0.22023	20080501	

rate_plan_identifier	rate_component	rate_group	season	period	price_type	tier	start_time	price	effective_start_date	effective_end_date
Rate-1	TOU_TIERED		SUMMER	ON_PEAK	CHARGE	4		0.22023	20080501	
Rate-1	TOU_TIERED		SUMMER	ON_PEAK	CHARGE	5		0.22023	20080501	
*	PDP			CRITICAL_PEAK	CHARGE			0.60000	20080501	
Rate-2	TOU		SUMMER	ON_PEAK	CHARGE			0.29108	20080501	
Rate-2	TOU		SUMMER	OFF_PEAK	CHARGE			0.13111	20080501	
Rate-2	TOU		WINTER	PART_PEAK	CHARGE			0.13612	20080501	
Rate-2	TOU		WINTER	OFF_PEAK	CHARGE			0.11289	20080501	
Rate-2	DEMAND/CONNECTED_LOAD		SUMMER		CHARGE			4.73	20080501	
Rate-2	DEMAND/CONNECTED_LOAD		WINTER		CHARGE			0.66	20080501	
Rate-3	FLAT/CUSTOMER				CHARGE			0.47310	20080501	
Rate-4	DEMAND/MAXIMUM		SUMMER		CHARGE			6.45	20080501	
Rate-4	DEMAND/MAXIMUM		WINTER		CHARGE			1.36	20080501	
Rate-4	DEMAND/PRIMARY_VOLTAGE		SUMMER		CREDIT			-0.77	20080501	
Rate-4	DEMAND/PRIMARY_VOLTAGE		WINTER		CREDIT			-0.21	20080501	

Rate Plan Configuration Files

You must send rate plan configuration files in order for your Delivery Team to configure rate plans with the correct components, names, and pricing. These files need to be provided when

a rate plan is first defined and when any of the components change. Since updates to these configurations are expected to be infrequent, these configuration files may be extracted in an automated fashion or compiled manually by the utility to provide to the Delivery Team.

- Rate Tier Definitions File
- Rate Period Definitions File
- Holidays and Seasons File
- Peak Time Rebates File

Rate Tier Definitions File

For tiered rate plans, the rate tier definitions file provides the metadata used to configure the tiers for the plan. The data must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

> **Note**
>
> This data is only required if you have a tiered rate plan. Your Delivery Team will create the file and work with you to gather the necessary inputs.

Column Name	Description
rate_plan_identifier	Unique rate plan for which the tier definition is applicable. This may be a wildcard (*) if the tier definition is applicable to all rate plans. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the rate component for which the tier definition is applicable. This may be a wildcard (*) if tier definition is applicable to all rate components in the specified rate plan. **Type**: VARCHAR **Can Be Empty?**: No.
tier	Tier number (for example, '1'). **Type**: INTEGER **Can Be Empty?**: No.
lower_bound	The lower bound of the tier. The lower bound is inclusive. The unit of measure should be the same as the associated service point's unit of measure. **Type**: DECIMAL (12,6). See Supported Decimal Formats for details. **Can Be Empty?**: No.
upper_bound	The upper bound of the tier. The upper bound is exclusive. This may be NULL for the highest tier. The unit of measure should be the same as the associated service point's unit of measure. **Type**: DECIMAL (12,6). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.
effective_start_date	The date on which this tier configuration becomes effective. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: No.

Column Name	Description
effective_end_date	Date on which this tier configuration is no longer effective. (This day is not part of the period covered by the rate tier definition.) **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.

Rate Tier Sample Data

This example assumes the tier definitions are constant across all rate plans for this particular client.

rate_plan_identifier	rate_component	tier	lower_bound	upper_bound	effective_start_date	effective_end_date
*	*	1	0	100	20090101	
*	*	2	100	130	20090101	
*	*	3	130	200	20090101	
*	*	4	200	300	20090101	
*	*	5	300		20090101	

Rate Period Definitions File

Many pricing plans have periods associated with different prices. For example, in a TOU plan, afternoon hours may be “on peak.” The purpose of the rate period definitions file is to provide rate period configuration information to Oracle Utilities for these types of plans. This file should be provided on a periodic basis at the same frequency at which rate periods change. Note that when defining TOU periods, off-peak periods should not be explicitly defined. On-peak and part-peak periods should be defined, and the system will default any other times that are not covered by on-peak and part-peak as off-peak. The data must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

> **Note**
>
> This data is only required if you have a time of use (TOU) plan or another pricing plan that requires rate periods. Your Delivery Team will create the file and work with you to gather the necessary inputs.

Column Name	Description
rate_plan_identifier	The unique rate plan for which the event is applicable. This may be a wildcard (*) if the event is applicable to all rate plans. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the rate component for which the event is applicable (for example, ‘TOU’). This may be a wildcard (*) if the event is applicable to all rate components in the specified rate plan. **Type**: VARCHAR **Can Be Empty?**: No.

Column Name	Description
season	The season for which the period is being defined. **Type**: ENUM **Allowed Values**: • WINTER • SPRING • SUMMER • FALL **Can Be Empty?**: Yes.
day_type	The type of day for which the period is being defined. **Type**: ENUM **Allowed Values**: • WEEKEND • WEEKDAY • HOLIDAY **Can Be Empty?**: Yes.
period	The period being defined. **Type**: ENUM **Allowed Values**: • ON_PEAK • PART_PEAK • OFF_PEAK • CRITICAL_PEAK • NON_CRITICAL_PEAK **Can Be Empty?**: No.
ordinal	Allows for distinguishing events with same season, day_type, and period. **Type**: INTEGER (1-10) **Can Be Empty?**: Yes.
resolution	Used in combination with duration to indicate the length of time covered by the event. **Type**: ENUM **Allowed Values**: • QUARTER_HOUR • HALF_HOUR • HOUR • DAY • BILLING • MONTH • YEAR **Can Be Empty?**: No.
duration	Used in combination with resolution to indicate the length of time covered by the event. **Type**: INTEGER **Can Be Empty?**: No.
start_date	Not applicable for rate periods. **Type**: DATE (MMDD) **Can Be Empty?**: Yes.

Column Name	Description
start_time	Start time of the event in military time. Do not specify a timezone. The local timezone of the utility is assumed. **Type**: TIME (HHMM) **Can Be Empty?**: No.
event_date	Not applicable for rate periods. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.
effective_start_date	The date on which this event configuration becomes effective. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: No.
effective_end_date	Date on which the configuration is no longer effective, if known. This day is not part of the event period. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.

Rate Period Sample Data

The rate period definitions file provides configuration information for pricing plans that have different prices for different periods of time. The table below shows an example of how the rate prices file might look.

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_date	start_time	event_date	effective_start_date	effective_end_date
Rate-1	*	SUMMER	WEEKDAY	ON_PEAK	1	HOUR	6		1200		20090101	
Rate-1	*	SUMMER	WEEKDAY	PART_PEAK	1	HALF_HOUR	7		0830		20090101	
Rate-1	*	SUMMER	WEEKEND	PARK_PEAK	2	HALF_HOUR	7		1800		20090101	
Rate-1	*	WINTER	WEEKDAY	PART_PEAK	1	HOUR	13		0800		20090101	
*	CPP			CRITICAL_PEAK		HOUR	6		1200		20090101	

Holidays and Seasons File

Many pricing plans have pricing associated with holidays and seasons. For example, in some TOU plans, all weekends and holidays may be considered “off peak.” Additionally, in many rate plans, prices vary between summer and winter. The purpose of this file is to provide holiday and/or season information to Oracle Utilities. The data must be provided in a single file that follows a specific naming convention. See Rates Data File Name Conventions for more information.

Column Name	Description
rate_plan_identifier	The unique rate plan for which the event is applicable. This may be a wildcard (*) if the event is applicable to all rate plans. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the rate component for which the event is applicable (for example, "CPP"). This may be a wildcard (*) if the event is applicable to all rate components in the specified rate plan. **Type**: VARCHAR **Can Be Empty?**: No.
season	The applicable season. This is not applicable for holiday definitions. Only include this if sending season information. **Type**: ENUM **Allowed Values**: • WINTER • SPRING • SUMMER • FALL **Can Be Empty?**: Yes.
day_type	The type of day. This is used to define the holiday. This is not applicable for season definitions. **Type**: ENUM **Allowed Values**: • WEEKEND • WEEKDAY • HOLIDAY **Can Be Empty?**: No.
period	The period being defined. This is not applicable for holiday definitions. **Type**: ENUM **Allowed Values**: • ON_PEAK • PART_PEAK • OFF_PEAK • CRITICAL_PEAK • NON_CRITICAL_PEAK **Can Be Empty?**: Yes.
ordinal	Allows for distinguishing events with same season, day_type, and period. This is not applicable for holiday definitions. **Type**: INTEGER (1-10) **Can Be Empty?**: Yes.

Column Name	Description
resolution	Used in combination with the `duration` to indicate the length of time covered by the event. The value should be 'DAY' for holiday definitions. **Type**: ENUM **Allowed Values**: • QUARTER_HOUR • HALF_HOUR • HOUR • DAY • BILLING • MONTH • YEAR **Can Be Empty?**: No.
duration	Used in combination with the `resolution` to indicate the length of time covered by the event. The value should be 1 for holiday definitions. **Type**: INTEGER **Can Be Empty?**: No.
start_date	The start date of the season. This is not applicable for holiday definitions. **Type**: DATE (MMDD) **Can Be Empty?**: Yes.
start_time	Start time of the event in military time. The value should be 0000 for holiday definitions. Do not specify a timezone. The local timezone of the utility is assumed. **Type**: TIME (HHMM) **Can Be Empty?**: No.
event_date	The date of the holiday. This is not applicable for seasons. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.
effective_start_date	Not applicable for holiday definitions. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.
effective_end_date	Not applicable for holiday definitions. **Type**: DATE (YYYYMMDD) **Can Be Empty?**: Yes.

Holidays and Seasons Sample Data

The holidays and seasons data file specifies different pricing information related to special holidays or seasons of the year, such as winter and summer. The tables below provide examples of how holidays and seasons data might look.

Holidays Sample Data

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_date	start_time	event_date	effective_start_date	effective_end_date
*	*		HOLIDAY			DAY	1		0000	20100704		
*	*		HOLIDAY			DAY	1		0000	20101225		

Seasons Sample Data

rate_plan_identifier	rate_component	season	day_type	period	ordinal	resolution	duration	start_date	start_time	event_date	effective_start_date	effective_end_date
*	*	SUMMER				MONTH	6	0501	0000		20110501	
*	*	WINTER				MONTH	6	1101	0000		20111101	

Peak Time Rebates File

In order to support rate plans with rebate components, the Rate Engine must receive the calculated rebates for each customer. This peak time rebates file should be transmitted after the event date and after the rebates have been calculated by the utility billing system. The file must follow a specific naming convention. See Rates Data File Name Conventions for more information.

Note: This file is only if applicable if your utility needs to provide peak time rebates data in support of certain products. Your Delivery Team will work with you to determine if it is required in your situation.

Column Name	Description
account_id	Account identifier. **Type**: VARCHAR **Can Be Empty?**: No.
rate_plan_identifier	The rate plan for which the rebate is applicable. This may be a wildcard (*) if the rebate applies to all rate plans for which the Service Agreement (SA) is eligible. **Type**: VARCHAR **Can Be Empty?**: No.
rate_component	The unique identifier of the type of rate component to which the rebate applies. **Type**: VARCHAR **Can Be Empty?**: No.
start_date	Effective start date of the specified rebate value. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.

Column Name	Description
end_date	Effective end date of the specified rebate value. This should be exclusive. For example, if the rebate is calculated in hourly increments and ends at 11:00, the `end_date` should be 20120101 1100. The hour from 11:00-12:00 will not be included. **Type**: DATETIME. See Date and Time Handling for details. **Can Be Empty?**: Yes.
performance_value	kWh or Therm value used to calculate the rebate amount for this period. A positive value should be sent for customers who used less than their baseline. A negative value should be sent for customers who used more than their baseline. **Type**: DECIMAL (12,6). See Supported Decimal Formats for details. **Can Be Empty?**: No.
performance_dollars	Rebate earned during this period of time. Prices should always be provided in dollars, not cents. Rebates should be negative or zero amounts. No positive amounts should be provided. **Type**: DECIMAL (12,6). See Supported Decimal Formats for details. **Can Be Empty?**: Yes.

Peak Time Rebates Sample Data

The peak time rebates file specifies pricing information related to the rebates that a customer can receive by participating in a peak day event. The table below provides an example of how the data might look.

account_id	rate_plan_identifier	rate_component	start_date	end_date	performance_value	performance_dollars
123	R_PTR	PEAK_TIME_REBATE	20120101 1300	20120101 1900	5.354	-10.3507471
456	RL_PTR	PEAK_TIME_REBATE	20100609 1200	20100609 1300	1.248	-2.1667152
456	RL_PTR	PEAK_TIME_REBATE	20100609 1300	20100609 1400	0.445	-0.7725868
456	RL_PTR	PEAK_TIME_REBATE	20100609 1400	20100609 1500	0.139	-0.2413249

Guidelines for Common Data Elements

Certain data elements are commonly used across the rate data files, such as account and rate plan identifiers, start dates, and end dates.

Account and Rate Plan Identifiers

The Rate Attributes File cannot be loaded until after a utility account has been created by your Delivery Team. Therefore, you should not send rate attributes files until Oracle Utilities has received your utility account information in the customer and billing data files as defined in Legacy Billing Data File Specifications. However, there are two common data element that must match across the data sets:

- **Rate Plan Identifiers** (`rate_plan_identifier`): Customer information needs to include a unique identifier which can be mapped to a specific rate structure and price. This identifier must match across all of the rate data files in which it is required.
- **Account Identifiers** (`account_id`): Unique account identifiers are used as the common value for applying rate attributes to a particular customer's account.

Start and End Dates

Many of the rates data files require effective start and end dates. For each data feed, you must order data by the effective start date. If the data is not ordered, errors may occur or the data may not be loaded as expected. A few of these scenarios are outlined below:

- If there is an existing record and another record with the same primary keys (for the same customer or rate / rate component combination) and an earlier start date is received, the data cannot be loaded. This causes an overlapping period that will generate an error.
- If there is an existing record and the start date for a new record is during the same period of the first record and has a null end date, this will generate an error.
- If there is a rate period with an effective start and end date and there is another record with a start date during that period and a null end date, an error will occur.
- If there is an existing record and another record with the same start date is received, the new record will overwrite the existing record. Corrections can only be made if the period is exactly the same.
- If there is an existing record with a null end date and another record with a later start date is received, the existing record will be end dated with the later start date and a new record will be inserted with the new start date.

When you are defining the period for an event or attribute using a `start_date`/`end_date` or `effective_start_date`/`effective_end_date`, keep in mind that the period includes the start date but *not* the end date. To avoid a gap between periods, the start date of the next period should be the same as the end date of the previous period.

Rates Data File Format

The following file format conventions apply to all rates data files that you transfer:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files. CSV format is also supported.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- All files should include a header line containing column names.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- All fields should be included in each generated file. A field that is not required may contain an empty value, but it should still have tabs (that is, the number of tabs should be the same in every row of the file).
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.
- Do not remove columns or add columns to the middle of the file. You must coordinate with your Delivery Team before adding columns to the end of a file.

Rates Data File Name Conventions

All files transferred to Oracle Utilities must follow the standard naming convention. Your Delivery Team will work with you to define the appropriate file names for any other files that must be automatically generated and transferred.

File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

> **Note**
>
> Not all files listed below are applicable or required for every utility. Your Delivery Team will work with you to determine which files are required.

Rate Attributes File:

opwr_<utility>_<population token>_<specification version number>_customer_rates_attributes_<yyyyMMddHHmmss>.gz

Baselines File:

opwr_<utility>_<population token>_<specification version number>_peak_time_rebate_baselines_<yyyyMMddHHmmss>.gz

Rate Prices File:

opwr_<utility>_<population token>_<specification version number>_rate_prices_<yyyyMMddHHmmss>.gz

Rate Tier Definitions File:

opwr_<utility>_<population token>_<specification version number>_rate_tier_definition_<yyyyMMddHHmmss>.gz

Rate Period Definition File:

opwr_<utility>_<population token>_<specification version number>_rate_period_definition_<yyyyMMddHHmmss>.gz

Holidays and Seasons File:

opwr_<utility>_<population token>_<specification version number>_rate_holidays_seasons_definition_<yyyyMMddHHmmss>.gz

Peak Time Rebates File:

opwr_<utility>_<population token>_<specification version number>_peak_time_rebates_<yyyyMMddHHmmss>.gz

The naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.

- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Rates Data Version History.
- **Data Type**: The filename should identify the type of rate data the file contains, such as *customer_rates_attributes*, *peak_time_rebate_baselines*, or *rate_prices*.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

Example Names

Suppose the population token is residential, the specification version number of the document is v1-0-0, and your utility is named "The Great Energy Company." Some examples of your utility's filenames would be as follows:

opwr_gec_res_v1-0-0_customer_rates_attributes_20210316131415.gz

opwr_gec_res_v1-0-0_peak_time_rebate_baselines_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility's filename would be as follows:

opwr_gec_res_v1-0-0_customer_rates_attributes_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.customer.rates.attributes.20210316131415.gz

Rates Data Version History

The following table lists updates to the rates data document specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Rates Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update the 'Rate Attributes File' sample table in Rates Data File Specifications Overview to include the `is_utility_account_attribute_deletion` column. Update the 'Rate Prices File' sample table description on the same page to correct the `seasons` column (plural) to `season` (singular).	v1-15-0	07/03/2025
Remove references and topics related to the Peak Pricing Event file, which has been deprecated and is no longer used.	v1-14-0	10/01/2024
Remove redundant file names from various topics. Update Rates Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-13-0	04/11/2022
Clarify that zip files should only contain one file each.	v1-12-0	11/23/2021
Update Introduction to clarify the purpose of rates data. Update Rates Data File Specifications Overview to include detailed examples of how the data files work. Update formatting of tables to be more mobile-friendly.	v1-11-0	06/25/2020
Clarify file extension requirements and expectations. Clarify start and end date rules for the SOLAR_TRUE_UP attribute in the Rate Attributes File file.	v1-10-0	11/12/2019
Remove redundant definition of required files. Update definition and examples of Rate Attributes File file to describe how solar customers can be identified.	v1-9-0	06/14/2019
Update file name convention for peak time rebate baselines file. Edit sections and headings to simplify content and clarify instructions.	v1-8-0	12/21/2018

Update	Specification Version Number	Date
Added description of new column for rate attribute deletion. Clarified that a timezone does not need to be specified for the start_time column. Update performance_dollars column to say it can be empty.	v1-7-0	07/02/2018
Added statement about using a minus sign to indicate negative values for prices. Updated file retention section to reflect regional differences.	v1-6-0	05/08/2018
Added statement about valid end-of-line separators.	v1-5-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-4-0	03/02/2018
Updated the sample value and key for `low_income_discount` data field.	v1-3-0	05/22/2017
Updated the descriptions for the `lower_bound` and `upper_bound` data fields.	v1-2-0	04/13/2017
Added a statement about the utility's data obligations.	v1-1-0	03/31/2017
Added baseline specification version number to the document. Updated explanations and organization. Updated naming conventions to include instructions on how to specify a population token and specification version number in the data file. Added note about periods in data file names. Added document number.	v1-0-0	02/13/2017
Added legal notices. Updated document branding.	Not applicable.	01/11/2017

Rate Engine Plus Data Transfer

The Oracle Utilities Opower platform delivers personalized energy cost insights to customers through printed reports, email messages, alerts, and web applications. The cost insights can be calculated by Oracle Utilities and displayed in user-friendly cost data graphs, rate plan comparisons, and hypothetical cost scenarios that help customers evaluate and switch between rate plans.

Oracle Utilities has a standard Rate Engine, as well as an add-on called Rate Engine Plus. This documentation describes the data components related to Rate Engine Plus. For more information about the standard Rate Engine, see Introduction Rates Data Transfer. For more

information about how the Rate Engine Plus differs from the standard Rate Engine, see the Oracle Utilities Opower Rate Engine Plus Cloud Service Product Overview.

> **Note**
>
> The requirements for transferring customer billed usage information are described separately in Legacy Billing Data Transfer. The billed usage data requirements are not addressed here, except to reference elements that need to be common across the two data sets.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Rate Engine Plus Data Integration Process

Your Delivery Team will use the following process to set up and configure utility-specific rates information for Rate Engine Plus:

1. **Scoping**: A product expert is assigned to review the utility's rate attributes and confirm the applicable rate components for those plans with the support of rate subject matter experts from the utility.
2. **Data Extract Build and Analysis**: The utility works with the Delivery Team to build the required data files. The Delivery Team analyzes this data to confirm the quality standards are met for consumption by the Rate Engine Plus. This process generally consists of two phases:
   - **Phase 1 - Full Historical File Integration**: For historical cost data display and rate analysis, the utility should perform one-time data file transfer for all periodic and historical data files that are applicable. The length of the history will be dictated by each utility's requirements, but it should coincide with the length of history for which customer and billed usage data is being provided from the utility production billing system. This is typically January 1st of the previous calendar year to the present. For example, if your Delivery Team planned to launch your program for the first time on December 1, 2025, the team would need data going back to January 1, 2024.
   - **Phase 2 - Incremental File Integration**: Ongoing data transfer frequency for each of the data elements is dependent on the frequency in which the data changes. Your Delivery Team will work with you to determine the data transfer frequency (weekly, daily) that is required.
3. **Configuration and Quality Assurance**: The client provides the Delivery Team with a set of sample test data for validation of the rate calculations done by the Rate Engine Plus. The Delivery Team tests the configurations using the test data sets provided by the client.
4. **User Acceptance**: The Delivery Team works with the client to conduct user acceptance testing of the calculations being presented by the end product prior to rollout.

Rate Engine Plus Data File Specifications

To set up the Rate Engine Plus tool, you must provide account-specific rate files to Oracle Utilities. See Rate Attributes File for more information about the required data elements. The

account-specific files should be provided on an ongoing basis as customers are introduced into the system or as customer data changes. You must also follow the proper file format requirements and Rate Engine Plus Data File Name Conventions when transferring Rate Engine Plus data to Oracle Utilities.

Rate Engine Plus Rate Eligibility

Customer rate eligibility information is required if Oracle Utilities Opower Rate Analysis features are enabled for the utility. Rate eligibility information is used to determine the rate plans for which customers may be eligible. Eligible rate plans are then displayed in rate analysis features, such as the Rate Comparison. If required, Oracle Utilities will work with the utility to collect eligibility information.

Rate Engine Plus Data File Format

The following file format conventions apply to all Rate Engine Plus data files that you transfer:

- Oracle Utilities prefers to receive all data as tab-separated value (TSV) files. CSV format is also supported.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- All files should include a header line containing column names.
- Rows can use "line feed" and "carriage return / line feed" as valid end-of-line separators.
- All fields should be included in each generated file. A field that is not required may contain an empty value, but it should still have tabs (that is, the number of tabs should be the same in every row of the file).
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file.
- Do not remove columns or add columns to the middle of the file. You must coordinate with your Delivery Team before adding columns to the end of a file.

Rate Engine Plus Data File Name Conventions

All Rate Engine Plus files transferred to Oracle Utilities must follow the standard naming convention. Your Delivery Team will work with you to define the appropriate file names for any other files that must be automatically generated and transferred.

File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Rate attributes file:

opwr_<utility>_<population token>_<specification version number>_customer_rates_attributes_<yyyyMMddHHmmss>.<gz or zip>

The naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation

process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.

- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.
- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Rate Engine Plus Data Version History.
- **Data Type**: The filename should identify the type of rate data the file contains, such as customer_rates_attributes, customer_baselines, or rate_prices.
- **Date**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

Example File Names

Suppose the population token is residential, the specification version number of the document is v1-0-0, and your utility is named "The Great Energy Company.” Some examples of your utility’s filenames would be as follows:

opwr_gec_res_v1-0-0_customer_rates_attributes_20210316131415.gz

Suppose your utility is using PGP-encrypted files. An example of your utility’s filename would be as follows:

opwr_gec_res_v1-0-0_customer_rates_attributes_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.customer.rates.attributes.20210316131415.gz

Rate Engine Plus Data Version History

The following table lists updates to the Rate Engine Plus data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Rate Engine Plus Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update Rate Engine Plus Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-9-0	04/11/2022
Clarify that zip files should only contain one file each.	v1-8-0	11/23/2021
Clarify file extension requirements and expectations.	v1-7-0	11/12/2019
Remove redundant definition of required files. Update definition and examples of Rate Attributes File file to describe how solar data customers can be identified. Add information about Rate Engine Plus Rate Eligibility.	v1-6-0	06/14/2019
Edit sections and headings to simplify content and clarify instructions.	v1-5-0	01/10/2019
Added description of new column for rate attribute deletion.	v1-4-0	07/02/2018
Updated file retention section to reflect regional differences.	v1-3-0	05/08/2018
Added statement about valid end-of-line separators.	v1-2-0	03/19/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-1-0	03/02/2018
First version.	v1-0-0	05/17/2017

Unsubscribe Data Transfer

To reduce the risk of non-compliance with legal regulations regarding customer consent for communications (such as the CAN-SPAM Act and TCPA), as well as to provide a consistent user experience, Oracle Utilities offers a process for regular, automatic synchronization of address-level unsubscribes of digital products between utility systems and the Oracle Utilities Opower system. This syncing process can be used for unsubscribes that apply to the email, IVR (phone), and/or SMS (text messaging) channels.

To ensure that the systems are synchronized, Oracle Utilities offers both data extracts (from Oracle Utilities to the utility system) and data imports (from the utility to the Oracle Utilities system). However, either may be implemented without the other.

> **Note**
>
> Terms used by Oracle Utilities to describe data fields might differ in meaning from your internal system's terminology. Data fields are defined in this documentation to facilitate communication by ensuring use of common definitions. In order to avoid delays in launching your program, it is important that you read this content carefully and quickly contact your Oracle Utilities Delivery Team if you have any questions.

> **Note**
>
> As part of the setup and data transfer process, you must agree to certain data obligations to ensure the quality and integrity of the data. See Utility Data Obligations for more information.

Unsubscribe Data Integration Process

Integration of your unsubscribe data into the Oracle Utilities Opower platform generally involves the following phases: sample historical file integration and incremental file integration. In each phase, your Delivery Team will work with you to integrate, validate, and approve each of these files. The files will contain the same columns, and will follow the same formatting and naming standards. The files will only differ in the number of customers included and the periods of time they cover. See Data Transfer Process Overview for more information about each phase of the process.

Unsubscribe Data File Specifications

This section describes the distinct data elements that you must provide in the unsubscribe data file that you transfer to Oracle Utilities. Each unsubscribe data file that you send to Oracle Utilities should include the data fields described in the tables below. Each table row below is a column in the data file. The header row in each file you generate must contain all the values specified in the **Column Name** column, even if the values in those columns are blank. Additional notes to keep in mind include the following:

- The columns should appear from left to right in the order in which they appear in the table and must be separated by commas.
- All of the data fields described in the sections must be included in a single, channel-specific data file and sent to Oracle Utilities. For example, all data related to email unsubscribe must be included in a single email unsubscribe data file, while all data related to voice unsubscribes must be included in a single IVR unsubscribe data file. See Unsubscribe Data File Name Conventions for more information.

Email Unsubscribe Fields

Column Name	Description
address	The email address that the event applies to. The email address must be encoded in ASCII format. Email addresses that contain characters that are not in the ASCII character set are not supported. **Type**: STRING **Can Be Empty?**: No.

Column Name	Description
occurred_on	The date and time of the event. This should comply with the ISO 8601 date and time standard in the UTC time zone. **Type**: STRING **Can Be Empty?**: No.
email_address_event_type	The event type. This should always be `UNSUBSCRIBE`. **Type**: STRING **Can Be Empty?**: No.

Example Email Unsubscribe File

The following table is an example of an email unsubscribe file.

address	occurred_on	email_address_event_type
Jane.Smith@example.com	2015-07-05T22:16:18+00:00	`UNSUBSCRIBE`
Jane.Smith@example.com	2015-07-08T23:16:18+00:00	`UNSUBSCRIBE`

Interactive Voice Response (IVR) Unsubscribe Fields

Column Name	Description
address	The phone number that the event applies to. This should comply with the International Telecommunication Union (ITU) E.164 standard. **Type**: STRING **Can Be Empty?**: No.
occurred_on	The date and time of the event. This should comply with the ISO 8601 date and time standard in the UTC time zone. **Type**: STRING **Can Be Empty?**: No.
ivr_address_event_type	The event type. This should always be `UNSUBSCRIBE`. **Type**: STRING **Can Be Empty?**: No.

Example IVR Unsubscribe File

The following table is an example of an IVR unsubscribe file.

address	occurred_on	ivr_address_event_type
+12223334444	2015-07-05T22:16:18+00:00	UNSUBSCRIBE
+12223334444	2015-07-08T23:16:18+00:00	UNSUBSCRIBE

SMS Unsubscribe Fields

Column Name	Description
address	The phone number that the event applies to. This should comply with the International Telecommunication Union (ITU) E.164 standard. **Type**: STRING **Can Be Empty?**: No.
occurred_on	The date and time of the event. This should comply with the ISO 8601 date and time standard in the UTC time zone. **Type**: STRING **Can Be Empty?**: No.
sms_address_event_type	The event type. This should always be `UNSUBSCRIBE`. **Type**: STRING **Can Be Empty?**: No.
short_code	The short code that the address receives communications from. This must only consist of digits. The minimum length is 3 characters. The maximum length is 8 characters. **Type**: STRING **Can Be Empty?**: No.

Example SMS Unsubscribe File

The following table is an example of an SMS unsubscribe file.

address	occurred_on	sms_address_event_type	short_code
+12223334444	2015-07-05T22:16:18+00:00	UNSUBSCRIBE	12345
+12223334444	2015-07-08T23:16:18+00:00	UNSUBSCRIBE	12346

Unsubscribe Data Order and File Format

Unsubscribe data files are processed in a particular order and file format.

Data Order

The Oracle Utilities Opower platform processes files according to the order received, then dated. If we receive multiple files on the SFTP server in the same time frame (as could be done for full historical data files), they will be automatically queued by their filename date stamps.

For example, if a file is received on the SFTP server named *opwr_util_full_20140520.txt* and then receive one named *opwr_util_full_20140519.txt* an hour later, *opwr_util_full_20140520.txt* will be processed first despite the fact that its date stamp is later than the *opwr_util_full_20140519.txt*. However, if these files are received at the same time, the file name data stamp will take precedence and *opwr_util_full_20140519.txt* will be processed before *opwr_util_full_20140520.txt*.

To ensure that all files are queued in the correct order, files should be named so that the date stamps are consistent with the intended load order of the files. This ensures that they will be

processed in the right order. See Unsubscribe Data File Name Conventions for more information.

File Format

ll files generated by your extract program need to meet the following standards to load successfully into the Oracle Utilities Opower platform:

- The fields in these files must be comma-delimited and in CSV format.
- No null fields may be included, as all of the fields are required.
- Files must use UTF-8 encoding so that both Latin and non-Latin characters can be supported.
- Some fields must have specific values. For example, `email_address_event_type` must always contain `UNSUBSCRIBE`.
- All files must include a header row containing column names from the tables specified in this document. Header column names must exactly match the names specified in this data specification.
- It is strongly recommended that files be compressed for transfer. Zip and gzip compression are supported, but gzip is strongly preferred. Files can be sent uncompressed, but compressing the files will greatly decrease the transfer time. Zip files must contain only one file each. There should be no directories in the zip file. Compressed gzip or zip files must contain only one text file each.

Unsubscribe Data File Name Conventions

All unsubscribe data files transferred to Oracle Utilities must follow a standard naming convention. File name components enclosed in angle brackets (<>) must be populated. File name components enclosed in square brackets ([]) are optional and can be included or left out. Any components without brackets must be part of the file name and should not be changed.

Email Unsubscribe File:

opwr_<utility>_res_<specification version number>_email-address-event-v1_<tier>_<yyyyMMddHHmmss>.csv

IVR (Voice) Unsubscribe File:

opwr_<utility>_res_<specification version number>_ivr-address-event-v1_<tier>_<yyyyMMddHHmmss>.csv

SMS (Text) Unsubscribe File:

opwr_<utility>_res_<specification version number>_sms-address-event-v1_<tier>_<yyyyMMddHHmmss>.csv

This naming convention is made up of the following components:

- **Prefix**: The *opwr* prefix, which is the standard abbreviation for Oracle Utilities Opower.
- **Utility Identifier**: A three- or four-character code that identifies the utility sending the file. Your Delivery Team will work with you to choose the utility code early in the implementation process. The code needs to be a unique identifier for the utility in the Oracle Utilities Opower system. For example, “The Great Energy Company” might have an identifier of *gec*.
- **Population Token**: An abbreviation that indicates whether the data is applicable to residential (*res*) or non-residential (*nonres*) customer populations. For the non-residential

category, there are additional segmentations available to designate Small and Medium Business populations (*nonres-smb*) or Large Commercial Industrial populations (*nonres-lci*). Consult your Service Delivery Manager to determine which is the most appropriate population token if you need to use additional segmentations.

- **Specification Version Number**: A three-digit version number of the data transfer specification. Each digit must be separated by a hyphen (-) and not a period or an underscore. This must match the latest number shown in the Specification Version Number column in Unsubscribe Data Version History.
- **Data Type**: The filename must include the phrase <*channel*>*-address-event-v1* to indicate the type of data included and the specification version used to generate and interpret the file. For example, if the channel is email, the phrase would be *email-address-event-v1*.
- **Tier**: The deployment environment to which this data corresponds. Can be *stage* for pre-production or *prod* for live production.
- **Date/Time**: The date that the file was generated, in the format *yyyyMMddHHmmss*.
- **File Extension**: All files must be formatted in CSV form. All files should be gzipped and should end with *.gz*, or be zipped and end with *.zip*. If you choose to PGP encrypt your files, then the *.pgp* extension should be used. In the case of PGP encryption, compression is not required since the encryption process includes file compression. Files can be up to one gigabyte compressed, or up to four gigabytes uncompressed.

File Name Examples

Suppose the specification version number of the document is v1-0-0 and your utility is named "The Great Energy Company." An example of your utility's file name would be as follows:

opwr_gec_res_v1-0-0_email-address-event-v1_prod_20210316131415.csv

Suppose your utility is using PGP-encrypted files. An example of your utility's file name would be as follows:

opwr_gec_res_v1-0-0_email-address-event-v1_prod_20210316131415.pgp

Additional Notes

The filename must not exceed 183 characters and must not contain a period (.) unless the period comes right before the file extension. The Oracle Utilities Opower file reception system considers the first period in the file name to be the beginning of a file extension. For example, the following file name is invalid and cannot be processed:

opwr.gec.res.v1.0.0.email-address-event-v1_prod_20210316131415.gz.

Unsubscribe Data Version History

The following table lists updates to the unsubscribe data specification by version number and date. The latest version number should be used in the file name that you send to Oracle Utilities. See Unsubscribe Data File Name Conventions for more information.

Update	Specification Version Number	Date
Update Unsubscribe Data File Name Conventions to be clearer about file name length limit and what file name components are required.	v1-9-0	04/11/2022

Update	Specification Version Number	Date
Clarify that zip files should only contain one file each.	v1-8-0	11/23/2021
Clarify file extension requirements and expectations.	v1-7-0	11/12/2019
Edit sections and headings to simplify content and clarify instructions.	v1-6-0	01/10/2019
Updated file retention section to reflect regional differences.	v1-5-0	05/08/2018
Added statement about valid end-of-line separators.	v1-4-0	03/09/2018
Updated description of file transfer process. Added information about correcting file upload errors and Oracle file retention policy.	v1-3-0	03/02/2018
Added a statement about the utility's data obligations.	v1-2-0	03/31/2017
Added document number and updated branding.	v1-1-0	02/13/2017
Added legal notices. Updated document branding. Added baseline specification version number to the document. Updated naming conventions.	v1-0-0	01/11/2017
Updated document formatting. Updated document name and scope to include unsubscribe data for all digital channels (email, voice, and text), not just the email channel. Added field descriptions and file naming conventions for importing voice and text unsubscribe data.	Not applicable.	05/18/2016