

# How To Offload Your iMessages for Long-Term Storage (Mac Only)

## 1. Access the Messages Folder

To access your iMessages, first navigate to the Messages folder on your Mac:

- Open **Finder**.
- Press **Command + Shift + G** to open the "Go to Folder" dialog.
- Type `~/Library/Messages/` and hit **Enter**.

You'll now see the **Messages** folder. Copy the entire folder to another location, like your Downloads folder or an external hard drive.

## 2. Install Required Tools

### You'll need:

- **SQLite3**: Install via the command `brew install sqlite` if you're on a Mac.
- **DB Browser for SQLite**: Download from [sqlitebrowser.org](http://sqlitebrowser.org).
- **Python**: Ensure you have Python installed.
- **Flask**: in terminal run `'pip install flask'`

## 3. Use Python to Access Messages

Next, use the following Python script to extract and display iMessage conversations. Make sure to update the `chat.db` path to match your setup:

```
import sqlite3
from flask import Flask, render_template, request

app = Flask(__name__, template_folder='C:/Users/user/OneDrive/Desktop/Imessage_Project/templates')

# Replace with your database path
DATABASE = 'C:/Users/user/OneDrive/Desktop/Message Data/chat.db'

def get_db_connection():
    conn = sqlite3.connect(DATABASE)
```

```

conn.row_factory = sqlite3.Row
return conn

# Main route to list conversations
@app.route('/')
def index():
    conn = get_db_connection()
    query = """
SELECT
chat_identifier as message_title
,coalesce(case when display_name = '' then null else display_name end,Group_Members) chat_display_name
,max(datetime(message.date / 1000000000 + 978307200, 'unixepoch', 'localtime')) AS date
, COUNT(message.ROWID) as message_count

FROM
    message
LEFT JOIN chat_message_join
ON message.ROWID = chat_message_join.message_id
LEFT JOIN chat
ON chat.ROWID = chat_message_join.chat_id
LEFT JOIN message_attachment_join
ON message_attachment_join.message_id = message.ROWID
LEFT JOIN attachment
ON message_attachment_join.attachment_id = attachment.ROWID
LEFT JOIN handle
ON message.handle_id = handle.ROWID
--349,450
LEFT JOIN
(
SELECT
chat_id
,GROUP_CONCAT(DISTINCT handle.id) as Group_Members -- ORDER BY handle.id
FROM chat_handle_join
LEFT JOIN handle
ON chat_handle_join.handle_id = handle.ROWID
group by chat_id
) MBRs
on chat.ROWID = mbrs.chat_id
--LEFT JOIN chat_handle_join
--ON chat_handle_join.chat_id = chat.ROWID

where chat_identifier is not null

group by 1,2

ORDER BY message.date desc
"""
    conversations = conn.execute(query).fetchall()
    conn.close()
    return render_template('index.html', conversations=conversations)

# Route to show messages of a specific conversation
@app.route('/conversation/')
def conversation(message_title):
    conn = get_db_connection()
    query = """
SELECT
datetime(message.date / 1000000000 + 978307200, 'unixepoch', 'localtime') AS date
,message.is_from_me
,handle.id
,Group_Members
,CASE
when handle.id = chat_identifier
then 'N'
else 'Y'
end as Group_Chat_Ind
,chat_identifier message_title
,display_name
,coalesce(case when display_name = '' then null else display_name end,Group_Members) chat_display_name
,case
when attachment.filename is null
or attachment.filename = ''
Then 'N'
else 'Y'
end as Attachment_Ind
,replace(replace(attachment.filename,'~/Library/Messages',''+ r"C:\Users\user\OneDrive\Desktop(Message Data)'),'',''+ "\" + \"\"') filename
--,attachment.filename
,message.text

```

```

FROM
  message
LEFT JOIN chat_message_join
ON message.ROWID = chat_message_join.message_id
LEFT JOIN chat
ON chat.ROWID = chat_message_join.chat_id
LEFT JOIN message_attachment_join
ON message_attachment_join.message_id = message.ROWID
LEFT JOIN attachment
ON message_attachment_join.attachment_id = attachment.ROWID
LEFT JOIN handle
ON message.handle_id = handle.ROWID

LEFT JOIN
(
  SELECT
  chat_id
  ,GROUP_CONCAT(DISTINCT handle.id) as Group_Members --ORDER BY handle.id)
FROM chat_handle_join
LEFT JOIN handle
ON chat_handle_join.handle_id = handle.ROWID
group by chat_id
) MBRS
on chat.ROWID = mbrs.chat_id

where chat.chat_identifier = ?

ORDER BY message.date asc
"""
messages = conn.execute(query, (message_title,)).fetchall()
if messages:
    chat_display_name = messages[0]['chat_display_name']
else:
    chat_display_name = "Unknown Chat"
if messages:
    Attachment_Ind = messages[0]['Attachment_Ind']
else:
    Attachment_Ind = "Unknown Chat"
if messages:
    filename = messages[0]['filename']
else:
    filename = "Unknown Chat"
conn.close()
return render_template('conversation.html', messages=messages, message_title=message_title, chat_display_name=chat_display_name, Attachment_Ind=Attachment_Ind, f

if __name__ == '__main__':
    app.run(debug=True)

```

## 4. HTML Template for Conversation Index

Below is the HTML code for the index page where you can view all of your chats to choose from. Save this as index.html.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Conversations</title>
  <style>

```

```
body {
  font-family: Arial, sans-serif;
  background-color: #2f3e47;
  color: white;
  margin: 0;
  padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  width: 300px;
  background-color: #3e4c55;
  border-radius: 20px;
  padding: 20px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  display: flex;
  flex-direction: column;
}

.search-box {
  position: relative;
  margin-bottom: 20px;
}

.search-box input {
  width: 100%;
  padding: 10px;
  border-radius: 20px;
  border: none;
  outline: none;
  background-color: #fff;
  color: #000;
}

.conversation-list {
  list-style-type: none;
  padding: 0;
  margin: 0;
  max-height: 400px; /* Set a fixed height */
  overflow-y: auto; /* Enable vertical scroll */
  overflow-x: hidden; /* Disable horizontal scroll */
}

.conversation-item {
  display: flex;
  align-items: center;
  padding: 10px;
  margin-bottom: 10px;
  background-color: #4b5b66;
  border-radius: 10px;
  cursor: pointer;
  transition: background-color 0.3s ease;
  white-space: nowrap; /* Prevent horizontal overflow */
  overflow: hidden; /* Hide overflow text */
  text-overflow: ellipsis; /* Show ellipsis if text overflows */
}

.conversation-item:hover {
  background-color: #5c6a76;
}

.avatar {
  width: 40px;
  height: 40px;
  border-radius: 50%;
  margin-right: 10px;
  flex-shrink: 0;
}

.conversation-info {
  display: flex;
  flex-direction: column;
  overflow: hidden;
}
```

```

.conversation-title {
    font-weight: bold;
    color: #ffffff;
    text-overflow: ellipsis; /* Ellipsis for long conversation titles */
    overflow: hidden;
}

.conversation-status {
    font-size: 12px;
    color: #cfcfcf;
    text-overflow: ellipsis; /* Ellipsis for long statuses */
    overflow: hidden;
}

.status-dot {
    width: 10px;
    height: 10px;
    border-radius: 50%;
    background-color: green;
    margin-right: 5px;
}

.offline .status-dot {
    background-color: red;
}

/* Add scrollbar styling */
.conversation-list::-webkit-scrollbar {
    width: 8px;
}

.conversation-list::-webkit-scrollbar-thumb {
    background-color: #888;
    border-radius: 10px;
}

.conversation-list::-webkit-scrollbar-thumb:hover {
    background-color: #555;
}

/* Anchor tag to fill the conversation item */
.conversation-link {
    text-decoration: none;
    color: inherit;
    width: 100%;
    display: flex;
}

</style>
</head>
<body>

<div class="container">
  <!-- Search Box -->
  <div class="search-box">
    <input type="text" placeholder="Search...">
  </div>

  <!-- Conversation List -->
  <ul class="conversation-list">
    {% for conversation in conversations %}
    <li class="conversation-item">
      <a href="{{ url_for('conversation', message_title=conversation['message_title']) }}" class="conversation-link">
        
        <div class="conversation-info">
          <div class="conversation-title">{{ conversation['chat_display_name'] }}</div>
          <div class="conversation-status">
            <span class="status-dot {% if conversation['is_online'] == False %} offline {% endif %}"></span>
            Last message: {{ conversation['date'] }}
          </div>
        </div>
      </a>
    </li>
    {% endfor %}
  </ul>
</div>

</body>

```

```
</html>
```

## 5. HTML Template for Conversations

Below is the HTML code for the conversation page where you can view all of your messages with someone or in the group chat. Save this as conversation.html.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{ message_title }} Conversation</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background-color: #f0f4f8;
    margin: 0;
    padding: 0;
  }

  .chat-container {
    width: 100%;
    max-width: 800px;
    margin: 0 auto;
    background-color: #fff;
    border-radius: 20px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    overflow: hidden;
    display: flex;
    flex-direction: column;
    height: 100vh;
  }

  /* Header */
  .chat-header {
    background-color: #2f3e47;
    color: white;
    padding: 20px;
    display: flex;
    justify-content: space-between;
    align-items: center;
  }

  .chat-header img {
    border-radius: 50%;
    margin-right: 10px;
  }

  .chat-header-title {
    font-size: 20px;
    font-weight: bold;
  }

  /* Message List */
  .message-list {
```

```
padding: 20px;
overflow-y: auto;
flex-grow: 1;
background-color: #e6eaea;
display: flex;
flex-direction: column;
}

.message-list::-webkit-scrollbar {
width: 8px;
}

.message-list::-webkit-scrollbar-thumb {
background-color: #888;
border-radius: 10px;
}

/* Message Bubble */
.message {
display: flex;
margin-bottom: 20px;
align-items: flex-end;
}

.message-content {
max-width: 60%;
padding: 12px 16px;
border-radius: 20px;
font-size: 14px;
position: relative;
display: flex;
flex-direction: column;
}

.message-text {
margin: 0;
word-wrap: break-word;
}

/* Default date style */
.message-left .message-date {
font-size: 10px;
color: #888;
margin-top: 5px;
text-align: left;
}

/* Left-aligned (sender) messages */
.message-left {
justify-content: flex-start;
}

.message-left .message-content {
background-color: #f4f4f4;
border-bottom-left-radius: 0;
border-radius: 18px 18px 18px 0;
}

.message-left .sender-info {
/*font-weight: bold;*/
font-size: 9px;
margin-bottom: 5px;
color: #555;
text-align: left;
}

/* Right-aligned (user) messages */
.message-right {
justify-content: flex-end;
}

.message-right .message-content {
background-color: #007aff;
color: white;
border-bottom-right-radius: 0;
border-radius: 18px 18px 0 18px;
text-align: left;
}
```

```

.message-right .message-date {
    font-size: 10px;
    color: #888;
    margin-top: 5px;
    text-align: right;
}

/* White date style for user's messages */
.message-right .message-date {
    color: white;
}

/* Attachments */
.attachment {
    margin-top: 10px;
}

.attachment img, .attachment video {
    max-width: 100%;
    border-radius: 10px;
}

.attachment .filename {
    font-size: 12px;
    color: #555;
    margin-top: 5px;
}

/* Input area */
.chat-input {
    padding: 10px 20px;
    background-color: #2f3e47;
    display: flex;
    align-items: center;
}

.chat-input textarea {
    width: 100%;
    padding: 10px;
    border-radius: 20px;
    border: none;
    outline: none;
    resize: none;
    background-color: #fff;
    color: #000;
}

.back-button {
    margin-left: 10px;
    color: white;
    text-decoration: none;
}
</style>
</head>
<body>

<div class="chat-container">
  <!-- Chat Header -->
  <div class="chat-header">
    <div class="chat-header-title">
      {{ chat_display_name }}
    </div>
    <a href="{{ url_for('index') }}" class="back-button">Back to conversations</a>
  </div>

  <!-- Message List -->
  <div class="message-list" id="messageList">
    {% for message in messages %}
    <div class="message {% if message['is_from_me'] == 1 %}message-right{% else %}message-left{% endif %}">
      <div class="message-content">
        <!-- Only show the contact/sender ID for messages not from me -->
        {% if message['is_from_me'] == 0 %}
        <div class="sender-info">{{ message['id'] }}</div>
        {% endif %}
        <p class="message-text">{{ message['text'] }}</p>
        {% if message['Attachment_Ind'] == 'Y' %}
        <div class="attachment">
          {% if message['filename'].endswith('.jpg') or message['filename'].endswith('.png') or message['filename'].endswith('.heic') or message['filen

```



```


{% elif message['filename'].endswith('.mp4') or message['filename'].endswith('.mov') %}
<video controls>
  <source src="{{ message['filename'] }}" type="video/mp4">
</video>
{% endif %}
<div class="filename">
  <a href="{{ message['filename'] }}" target="_blank">{{ message['filename'] }}</a>
</div>
</div>
{% endif %}
<div class="message-date">{{ message['date'] }}</div>
</div>
</div>
{% endfor %}
</div>

<!-- Chat Input (Optional, not functional in this template) -->
<div class="chat-input">
  <textarea placeholder="Type your message..."></textarea>
</div>
</div>

<script>
  // Scroll to the bottom of the message list on page load
  window.onload = function() {
    var messageList = document.getElementById('messageList');
    messageList.scrollTop = messageList.scrollHeight;
  };
</script>

</body>
</html>

```

## 6. Open Your Conversation Interface

After running the Python script, you can view your conversations by opening your web browser and navigating to `http://127.0.0.1:5000`.

If you'd like to save a copy of the conversation page, you can use your browser's save feature:

- In Chrome, press **Ctrl + S** (or Command + S on Mac) and save the page as an HTML file.

Happy archiving! For more tips, visit our website or follow us on social media.