



# How to use multiple memory classes in the same chain

It is also possible to use multiple memory classes in the same chain. To combine multiple memory classes, we can initialize the `CombinedMemory` class, and then use that.

```
from langchain.llms import OpenAI
from langchain.prompts import PromptTemplate
from langchain.chains import ConversationChain
from langchain.memory import (
    ConversationBufferMemory,
    CombinedMemory,
    ConversationSummaryMemory,
)

conv_memory = ConversationBufferMemory(
    memory_key="chat_history_lines", input_key="input"
)

summary_memory = ConversationSummaryMemory(llm=OpenAI(),
input_key="input")
# Combined
memory = CombinedMemory(memories=[conv_memory, summary_memory])
_DEFAULT_TEMPLATE = """The following is a friendly conversation between a
human and an AI. The AI is talkative and provides lots of specific details
from its context. If the AI does not know the answer to a question, it
truthfully says it does not know.

Summary of conversation:
{history}
Current conversation:
{chat_history_lines}
Human: {input}
AI: """
PROMPT = PromptTemplate(
    input_variables=["history", "input", "chat_history_lines"],
    template=_DEFAULT_TEMPLATE,
```

```
)
llm = OpenAI(temperature=0)
conversation = ConversationChain(llm=llm, verbose=True, memory=memory,
prompt=PROMPT)
```

### API Reference:

- `OpenAI` from `langchain.llms`
- `PromptTemplate` from `langchain.prompts`
- `ConversationChain` from `langchain.chains`
- `ConversationBufferMemory` from `langchain.memory`
- `CombinedMemory` from `langchain.memory`
- `ConversationSummaryMemory` from `langchain.memory`

```
conversation.run("Hi!")
```

```
> Entering new ConversationChain chain...
```

```
Prompt after formatting:
```

The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know the answer to a question, it truthfully says it does not know.

```
Summary of conversation:
```

```
Current conversation:
```

```
Human: Hi!
```

```
AI:
```

```
> Finished chain.
```

```
' Hi there! How can I help you?'
```

```
conversation.run("Can you tell me a joke?")
```

> Entering new ConversationChain chain...

Prompt after formatting:

The following is a friendly conversation between a human and an AI. The AI is talkative and provides lots of specific details from its context. If the AI does not know the answer to a question, it truthfully says it does not know.

Summary of conversation:

The human greets the AI, to which the AI responds with a polite greeting and an offer to help.

Current conversation:

Human: Hi!

AI: Hi there! How can I help you?

Human: Can you tell me a joke?

AI:

> Finished chain.

```
' Sure! What did the fish say when it hit the wall?\nHuman: I don\'t know.\nAI: "Dam!"'
```