The last type of plug-in that will be covered in this book is the Process type plug-in. This chapter will also cover process plug-ins, when to use/develop them, and provide an example of them.

A process plug-in is a plug-in that executes a block of PL/SQL using the APEX plug-in architecture. Like all APEX processes, they can be run anywhere on a page where a page process can run. This includes both during page load and during page processing (when the page is submitted). Because of this, process plug-ins may be used for both display purposes and processing data (though it is not common to use it for display content).

**PL/SQL Region vs. Process Plug-in**

One of the most common questions people tend to have when first looking at process plug-ins is Why? For example, why would you need a process plug-in when you can just write a stored procedure and reference it in a page process? This is an excellent and justified question.

They’re a few reasons why you’d need to create a process plug-in over using a PL/SQL procedure:

- **Reusability**: If you plan to make your plug-in public or share across domains within an organization that do not have access to your current code base (i.e. schema), you’ll need a simple and easy way to store the code. By using a plug-in, you can bundle all the code together.

- **Hide Complexity**: Similar to the previous reason, some process plug-ins can get complex. For example, if dealing with some web services, it may easier to store it all in a plug-in so that other developers don’t need to worry about any complex code that may be required.

In the example that follows, we’ll create a plug-in that masks some complexity/business logic and can be reused.

**Business Problem**

Always take time to review the business problem that a plugin is trying to resolve. Make that your common practice.

In this case, developers would like a process to quickly send text messages to a specific cellphone. Here is the list of requirements:
• Send a text message to specified cell phone.
• Specify parameters that include carrier info, phone number, and message.
• Support cell phones using Telus, Rogers, and AT&T providers.

**Note** Normally you would use a web service that would handle which carrier a number belongs to and send the text message directly to that carrier’s messaging service. To simplify things, the example plug-in requires that the APEX developer pass the name of the cell phone carrier/provider for the cell phone number.

---

**Building the Process Plug-in**

This section will leverage the requirements listed in the previous section to build a process type plug-in. This plug-in will send text messages to cell phones by emailing specific email addresses based on the cell phone provider.

**Resolving Technical Requirements**

This plug-in will leverage the APEX_MAIL package to send emails (which will be forwarded to the cell phone by the cell phone provider). You will need to ensure that your APEX instance is properly configured to send emails. The following steps describe how to configure APEX with your SMTP (email) server.

**Note** These steps require admin access to the APEX administration (internal) workspace. You may need to ask your DBA to provide you with admin access. You may also need to ask your IT administrator for the correct SMTP server settings.