



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Composer Help

Web Service Stubbing

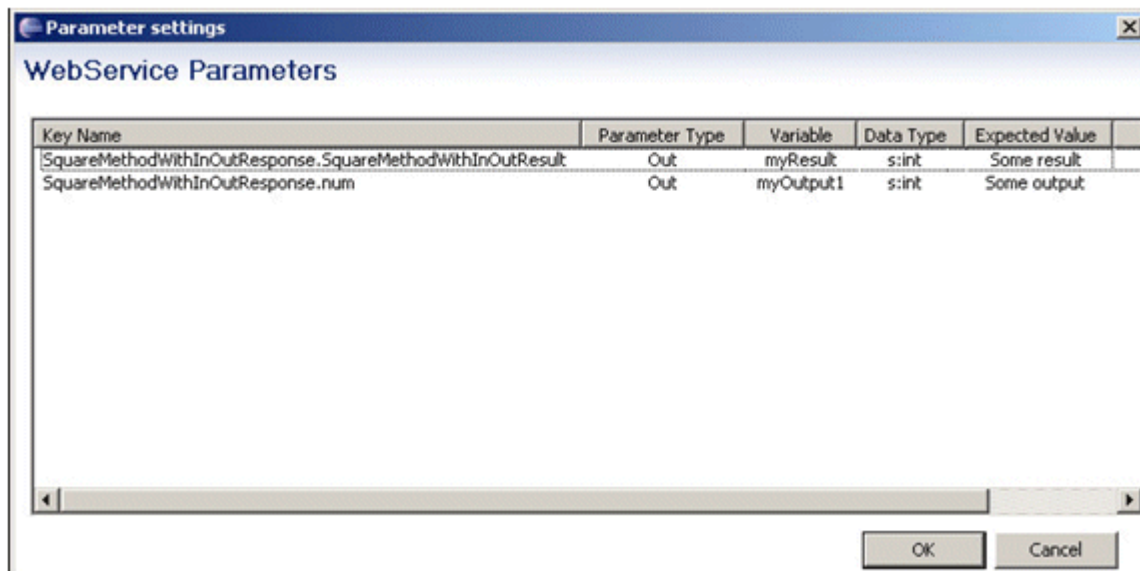
# Web Service Stubbing

Composer's Web Services stubbing feature allows you to work with Web Services in off-line mode when you do not have access to the Web Service itself or if the Web Service is under development. This feature is intended to be used in a test environment. It is not intended for a production environment unless there is need to remove an active Web Service from a callflow for debugging purposes.

## Using Web Services Stubbing

To use Web Services stubbing:

1. To enable stubbing, add the variable `COMPOSER_WSSTUBBING` to your Entry block and set its value to 1 indicating stubbing is turned on (0 = stubbing is turned off). In Composer 8.0.2 and later, this variable is present by default in the **Variables Setting dialog box**, which opens from the Entry object.
2. Create the Web Service block.
3. Place the Web Service Description Language (WSDL) file in your Project. The assumption is that the WSDL file for the Web Service is available at all times.
4. For the **Service URL** property, use a local URL to the WSDL file. When the Web Service is ready to be used, change this local URL to the correct URL.)
5. To specify the expected output value for the Web Service result as well as any output parameters, use the **Output Result** property of the Web Service block. An example is shown below.



above examples:

- If stubbing is on, the `myResult` variable will be assigned the value `Some result` and `myOutput1` will be

assigned the value of Some output.

- If stubbing is off, the value returned by the Web Service will be stored in these two variables.

## Limitation

Web Service stubbing currently does not support auto-synchronization of output parameters in case of Web Services with complex return types.