

# IBM Platform Symphony RFE55194 and RFE12598 Readme File

## **About RFE55194 (Replacing hardcoded information with variables in `client.py`)**

RFE55194 removes hardcoded user names, passwords, and application names in `client.py` and adds additional exceptions for the user. With this feature, you can configure the user name, password, and application name using variables. This feature also provides `JobCancelled` and `UnhandledRemoteError` exceptions.

## **About RFE12598 (Resolving conflicts for STDOUT and STDIN in Python wrapper (`pythonwrapperservice.py`) with SWIG generated C ++ library)**

RFE12598 provides socket communication between service side processes. This feature allows service side standard output and standard input messages.

**Readme file for:** IBM® Platform Symphony

**Product/Component Release:** Symphony 6.1.1

**Update Name:** Replacing hardcoded information with variables in `client.py`, and resolving conflicts for STDOUT and STDIN in Python wrapper (`pythonwrapperservice.py`) with SWIG generated C ++ library

**Fix ID:** sym-6.1.1-build-243057-pimco

**Publication date:** 11 Sep 2014

**Last modified date:** 11 Sep 2014

1 Scope .....	2
2 Configuration .....	2
2.1 Prerequisites .....	2
2.2 Installation files .....	3
2.3 Installation procedure .....	3
2.4 Configuration procedure .....	3
2.5 Verification procedure .....	4
1. How this feature works .....	5
4 Copyright and trademark information.....	6

## 1 Scope

Applicability	
Operating system	Linux2.6-glibc2.3-x86_64
Python version	2.4.5 64 bit, 2.7.2 64 bit
Symphony version	6.1.1
Cluster types	This feature applies to a single grid cluster or DE.
Other	This feature applies to SOAM.
Dependencies	
File system	This feature has no requirement on the file system type.
<Other>	Python 2.4.5 64 bit or Python 2.7.2 64 bit version must be installed on the client host and all compute hosts.
Limitations	
<Limitation>	None.
Known Issues	None.

## 2 Configuration

### 2.1 Prerequisites

Python 2.4.5 64 bit or Python 2.7.2 64-bit must be installed on the client host and all compute hosts.

## 2.2 Installation files

This package includes the following files:

File name	Description
<code>sympython-wrapper-lnx26-lib23-x64-6.1.1_243057.tar.gz</code>	This package contains this new feature for Linux x86_64.

## 2.3 Installation procedure

1. Download the `sympython-wrapper-lnx26-lib23-x64-6.1.1_243057.tar.gz` package from <http://www.ibm.com/eserver/support/fixes>.
2. Decompress the package: `sympython-wrapper-lnx26-lib23-x64-6.1.1_243057.tar.gz`.

For example, run:

```
tar xzvf sympython-wrapper-lnx26-lib23-x64-6.1.1_243057.tar.gz
```

3. Copy the python files to `$GRID_DIR_LOCATION` and replace the existing files:  

```
cp client.py $GRID_DIR_LOCATION/grid/  
cp custom.py $GRID_DIR_LOCATION/grid/  
cp service/pythonwrapperservice.py $GRID_DIR_LOCATION/grid/service/  
cp service/invokeservice.py $GRID_DIR_LOCATION/grid/service/
```

## 2.4 Configuration procedure

### 2.4.1 Configure RFE55194

There are two ways to set the user name, password, and application name. The default application name is "PythonApp". The default user name and password are both "Guest".

#### a. Set user name, password, and application name using environment variables:

Run:

```
export SOAM_APPLICATION_NAME="AppName"  
export SOAM_USERNAME="Guest"  
export SOAM_PASSWORD="Guest"
```

#### b. Set user name, password, and application name using global variables:

- a. Open a user module, such as `test_apply.py`.
- b. Configure the application name.

For example:

```
grid.client.APPLICATION_NAME = "PythonApp"
```

- c. Configure the user name and password.

For example:

```
grid.client.USERNAME = "Guest"  
grid.client.PASSWORD = "Guest"
```

- d. Save the file.

NOTE: Global variables take precedence, then environment variables, and finally, default values. For example, environment variables will overwrite the default values, and global variables will overwrite environment variables.

## 2.4.2 Configure RFE12598

1. Go to the service directory and deploy the service package:
  - a) `cd service`
  - b) `./makepackage.sh`
2. Set the `USE_SOCKET` environment variable to Y in the Service section of the application profile.  
For example:

```
<env name="USE_SOCKET">Y</env>
```

3. Register the application:

```
soamreg ../PythonApp.xml
```

## 2.5 Verification procedure

1. RFE55194: Configure the application name, user name, and password.

### Configure the application name, user name, and password using environment variables:

- a. Set the environment variables:

```
export SOAM_APPLICATION_NAME="AppName"  
export SOAM_USERNAME="Guest"  
export SOAM_PASSWORD="Guest"
```
- b. Register the application:

```
soamreg AppName.xml
```
- c. Run the user module:

```
Python sample/test_apply.py
```

### Configure the application name, user name, and password using global variables:

- a. Open the `sample/test_apply.py` file in the grid location:

```
vi sample/test_apply.py
```
- b. Add the following to the file:

```
grid.client.APPLICATION_NAME = "AppName"  
grid.client.USERNAME = "Guest"  
grid.client.PASSWORD = "Guest"
```

For example:

```
...  
def main_function():  
    grid.client.USE_LOCAL_MAP_APPLY = False  
    grid.client.APPLICATION_NAME = "AppName"  
    grid.client.USERNAME = "Guest"  
    grid.client.PASSWORD = "Guest"
```

```
submitter = grid.client.Client(max_slots=1, import_local_env=False,
debug_level=logging.INFO, debug_stdio=sys.stdout)
```

...

c. Save the file.

d. Register the application:

```
soamreg AppName.xml
```

e. Run the user module:

```
Python sample/test_apply.py
```

2. RFE12598: Echo messages to stdout on the service side:

a. Create test.py under grid location:

```
#!/usr/bin/python
import grid.client
import logging
import sys
def f(x):
    import os
    os.system("echo hello")
    return x

c = grid.client.Client()
for x in c.imap_apply(f, (range(1,3))):
    print 'hello'
    print x
```

b. Run test.py to ensure you can successfully print the output:

```
# python test.py
hello
1
hello
2
# cat $EGO_TOP/soam/work/PythonApp_21760_WorkerService.out
hello
```

## 3 Usage

### 3.1 How the RFE55194 feature works

This feature provides three environment variables and three global variables to configure the application name, user name, and password.

You can configure the application name, user name, and password in two ways: in the user modules as global variables, as follows:

```
grid.client.APPLICATION_NAME = "AppName"
grid.client.USERNAME = "Guest"
grid.client.PASSWORD = "Guest"
```

or as export them as environment variables, as follows:

```
export SOAM_APPLICATION_NAME="AppName"
export SOAM_USERNAME="Guest"
export SOAM_PASSWORD="Guest"
```

The default value of APPLICATION\_NAME is "PythonApp". The default value of USERNAME and PASSWORD are both "Guest".

### **3.2 How the RFE12598 feature works**

This feature provides socket communication between service side processes. Originally, on the service side, STDOUT and STDIN are used to exchange data in a special format; echoing messages to STDOUT breaks that protocol. After enabling this feature, echoing messages to STDOUT or STDIN will not impact workload.

## **4 Copyright and trademark information**

© Copyright IBM Corporation 2014.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM Web site pages might contain other proprietary notices and copyright information that should be observed.