



Pipeline Pilot Interface to *FlexS*

Version 1.30.1.3

User Guide

(for *FlexS* version 1.30.1 and above and Pipeline Pilot version 6.1 and above)

Edgar Derksen, Sally Hindle



Contents

Contents	3
1 Quick Start Steps	5
1.1 Drag, drop and set license	5
1.2 Note about the <i>FlexS</i> installation	5
2 Introduction	7
2.1 About <i>FlexS</i>	7
2.2 <i>FlexS</i> Component in Pipeline Pilot	7
2.2.1 Superpositioning	8
2.2.2 Virtual screening	9
3 The Internal and External <i>FlexS</i> Installations	11
3.1 Using the Internal Installation	11
3.2 Using an External <i>FlexS</i> Installation	11
3.2.1 To Connect to a <i>FlexS</i> Installation on the Pipeline Pilot Server	12
3.2.2 To Connect to a <i>FlexS</i> Installation on a Remote Linux Server	12
3.3 Running <i>FlexS</i> in Parallel in Pipeline Pilot	15
3.3.1 Running in Parallel on the Pipeline Pilot Server	16
3.3.2 Running in Parallel on a Remote Linux Cluster	16
3.3.3 Example Scenarios and Required Settings	19
4 Trouble Shooting	21
4.1 Problems connecting to the external installation	21
4.2 Problems using the <i>ssh</i> Method	23
4.3 Further help	23

5	Tips and Tricks	25
5.1	Other Significant Parameters in the <i>FlexS</i> Components	25
5.1.1	<sshHasSameFS>	25
5.2	Accessing Other Domains within Pipeline Pilot	25

Quick Start Steps

1.1 Drag, drop and set license

1. Unzip the download package and save the folder on your system.
2. Drag and drop the component (the `.xml` file) from the folder onto the component area of your Pipeline Pilot client.
3. Add your license for the executable of *FlexS* in the field for the parameter `<opsFlexSLicense>` in the Implementation tab – if your license is available from a license server, simply type in the name of the server in this format `@servername`, or if you have a license file then you may browse for it using the `...` facility. See below and sections 3.2.1 and 3.2.2 for more details on the *FlexS* installation and license.
4. Build the component into your Pipeline Pilot workflows. See 2.2 for suggested workflows.

1.2 Note about the *FlexS* installation

The *FlexS* component is set by default to use a so-called internal installation of *FlexS*. This works as follows. When you run the *FlexS* component (and the parameter `<UseInternalFlexS>` is set to *True*), it searches for a *FlexS* installation in the directory:

```
<scitegic install directory>/public/bin/BioSolveIT/
```

and uses this installation to run the calculation using the license you gave in the third step described above. If a *FlexS* installation is not found in this directory, Pipeline Pilot will install it itself using information from the component. Don't worry, you do not need Administrator rights to do this as it is actually Pipeline Pilot that carries out the installation and not you as a user.

Alternatively, you can also use an existing external installation of the *FlexS* software. This means you must already have *FlexS* installed on your system outside of Pipeline Pilot. If you have not already done so, visit the download page at BioSolveIT:

<http://www.biosolveit.de/download>

and fetch the download package for your system for the latest *FlexS* package. Follow the instructions in the package to install *FlexS* and receive your licenses.

See sections 3.2.1 and 3.2.2 for more details on the *FlexS* installation.

Introduction

2.1 About *FlexS*

FlexS is a computer program for predicting ligand superpositions. For a given pair of ligands, *FlexS* predicts the conformation and orientation of one ligand relative to the other, which is known as the reference ligand. The reference ligand is assumed to be rigid, thus, it should be given in a conformation which is similar to the bound state. For further information and references please visit <http://www.biosolveit.de/FlexS>. The two main applications of *FlexS* are ligand superpositioning (for example for CoMFA/QSAR) and virtual screening.

1. Ligand superpositioning - If you have a protein and a small molecule binding to it (so called reference ligand) but no structure of the protein, you can take the structure of the reference as a negative fingerprint of the active site. So, in contrast to a docking calculation, *FlexS* determines the structure of a test ligand by flexibly aligning to the reference. The result is scored according to the similarity between the test and the reference structure. To use *FlexS* for superpositioning within Pipeline Pilot see the example in section 2.2.1.
2. Virtual screening - with a reference ligand and a set of compounds to hand, you want to rank the set of compounds according to how similar their structures are to the reference structure. To use *FlexS* for virtual screening within Pipeline Pilot see the example in section 2.2.2 and setup for parallel processing in section 3.3.

2.2 *FlexS* Component in Pipeline Pilot

The component for *FlexS* in Pipeline Pilot is called `flexs-poses-xxx.xml` (where xxx is the version number). Once you have installed the component (See chapter 1 and 3) you can drag and drop the *FlexS* component from your components window to your workflow area.

The reference ligand in *FlexS* in Pipeline Pilot is called the 'Query'. You must set the `<Query>` parameter of the *FlexS* component to specify the reference ligand to be used in the calculation. The query should be a molecule file readable by the standard reader components of Pipeline Pilot. You can also select a multi-molecule file to align the test ligands

against different reference ligands. The test ligands are the incoming molecules from the pipeline. You can find examples of molecule files within this package. Now the *FlexS* component is ready to run.

For each incoming molecule the *FlexS* component runs a calculation using the query which results in a set of poses. These are ranked by the *FlexS* score. Only the best `<nofPoses>` are output from the component. You can change this optional parameter to set the number of poses to keep.

If the calculation for the input molecule fails, the input molecule will be passed to error port.

The following two examples form a first simple tutorial to see how the *FlexS* component works.

2.2.1 Superpositioning



Figure 2.1: Starting example for superpositioning.

1. First you need a source for the input molecules. Drag and drop a molecule reader component like "Mol2 Reader" to your empty pipeline area. Select a MOL2 file in the parameter tab with the parameter `<Source>`. You can try out a MOL2 file in the examples folder of this package.
2. Set the `<Maximum>` parameter to 1 to ensure you start with only one molecule. This will form a quick example as a first test.
3. Drag and drop *FlexS* component to your pipeline area and connect the output port of the molecule reader to the input port of *FlexS* component. Go to *FlexS* component and define a query molecule file with the parameter `<Query>` in the parameter tab.
4. By default 10 different docking poses will be calculated. You can change this value at the parameter `<nofPoses>` (be aware that *FlexS* can create hundreds of poses).
5. To see a simple visual result of the poses with their scores, drag and drop the component "HTML Table Viewer" to your pipeline. Connect the output pass port of *FlexS* component to the input port of "HTML Table Viewer".
6. Your workflow should look like the example in figure 2.1 and is ready to run.

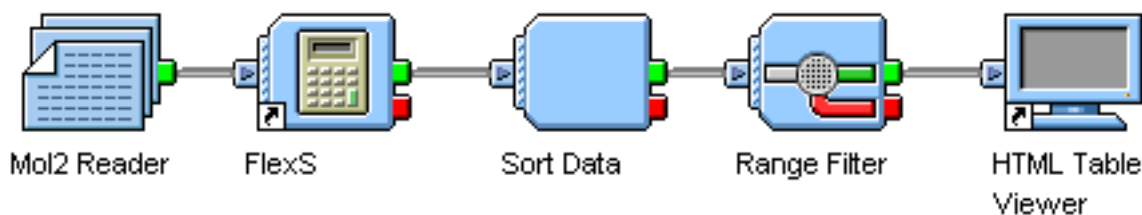


Figure 2.2: Starting example for virtual screening.

2.2.2 Virtual screening

1. Start with a molecule reader as described in 2.2.1. In this example the reader defines the library you want to screen. You can also replace the MOL2 reader with any molecule reader available in the Pipeline Pilot collection
2. Remember to change the `<Maximum>` parameter to read all molecule records from the file.
3. Set up *FlexS* component as described in 2.2.1.
4. Change the `<NofPoses>` parameter to 1. (If you want to take more poses into account you have to do some extended analysis of the results *after* the superposition calculation using the components available to you in Pipeline Pilot- check the Pipeline Pilot documentation for more information.) The result will now be one pose per input molecule.
5. The resulting poses need to be sorted to create a ranked list of your library. Drag and drop a "Sort Data" component and connect it to *FlexS* component. Set the parameter `<SortProperty1>` to value "BSIT_FlexS_Total_Score". The lowest scores are the best in *FlexS* so set the parameter `<SortDirection1>` to "Increasing".
6. To select a subset of the resulting ranked list you can add a filter. In this example we want to filter the 100 best results. Drag and drop the component "Range filter" and connect it to the "Sort Data" component. Set the parameter `<List>` to "1-100".
7. Finally add a viewer or a writer component to view and/or save your results. Your pipeline may look like the example in figure 2.2. Change the workflow to your needs and, for example, try out some more detailed analysis.

The Internal and External *FlexS* Installations

3.1 Using the Internal Installation

As mentioned previously the *FlexS* component is set by default to use a internal installation of *FlexS* (the parameter `<UseInternalFlexS>` set to *True*). This is a default installation made whenever the *FlexS* component is used for the first time. The software is installed automatically by Pipeline Pilot within its own space on the server:

```
<scitegic install directory>/public/bin/BioSolveIT/
```

The installation is carried out once only (for that *FlexS* version) and only once per server not per user. The software is available to all users once it is installed. Note that a normal user cannot edit or delete this installation.

As this installation knows nothing about the licenses you may have for *FlexS*, you have to supply the license information separately. This is supplied using the parameter `<opsFlexSLicense>` as described above in the section 1.

3.2 Using an External *FlexS* Installation

You can also use an existing external installation of the *FlexS* software. This means you also have the opportunity to use settings different to those set by default. To do this, you must already have *FlexS* installed somewhere on your system outside of Pipeline Pilot. To install *FlexS* yourself, visit the download page at BioSolveIT:

```
http://www.biosolveit.de/download
```

and fetch the download package for your system for the latest *FlexS* package. Follow the instructions in the package to install *FlexS* and receive your licenses. Enter the license information for *FlexS* as described in the package **and not using the parameter `<opsFlexSLicense>` as for the internal installation.**

To use an external installation of *FlexS*, you must change the value of the parameter <UseInternalFlexS> in the Implementation tab to *False*. You will then see that the parameter <UseFlexSONPPServer> becomes active and is by default set to *True*.

There are actually two ways to use *FlexS* with an external installation. These are by using *FlexS* installed directly on the Pipeline Pilot server you are using (the *ops* method), or by accessing a remote machine where *FlexS* is installed using *ssh* (the *ssh* method). The method is selected using the parameter <UseFlexSONPPServer>: *True* selects the *ops* method and *False* selects the *ssh* method. Both methods are covered in more detail below.

3.2.1 To Connect to a *FlexS* Installation on the Pipeline Pilot Server

The most common scenario is that you will have an installation of *FlexS* on the Pipeline Pilot server. If you choose this option, you just have to enter the path to the executable and configuration file as parameters in the Implementation tab. Pipeline Pilot will then just start *FlexS* whenever it is required by making a call to the executable that you entered.

This method is given the label *ops* – On Pipeline Pilot Server. You can see an example in figure 3.2.

- Requirement:
 - *FlexS* is installed on the Pipeline Pilot server. You can see which machine is the Pipeline Pilot server by starting your copy of Pipeline Pilot Client on your own workstation and find the name or IP of the server shown at the bottom right of the status bar (see figure 3.1). You must find where the *FlexS* installation is *on that machine*.
- General Steps:
 1. Set <UseInternalFlexS> to *False*
 2. Set <UseFlexSONPPServer> to *True*
 3. Expand <opsFlexSExe>
 4. For the parameter <opsFlexSExe>, enter the path to the *FlexS* installation on the Pipeline Pilotserver. For example:
C:\Programs\BioSolveIT\FlexS\bin\flexs.exe
 5. For the parameter <opsFlexSConfig>, enter the path to the *FlexS* configuration file on the Pipeline Pilotserver. For example:
C:\Programs\BioSolveIT\FlexS\config_ft.dat

You can save these settings in the component.

3.2.2 To Connect to a *FlexS* Installation on a Remote Linux Server

Your existing *FlexS* installation could be on a Linux computer remote from the Pipeline Pilot server – in this case we offer an alternative so you can use the remote installation instead. Here, the calculations carried out by *FlexS* will be done on the remote Linux machine.

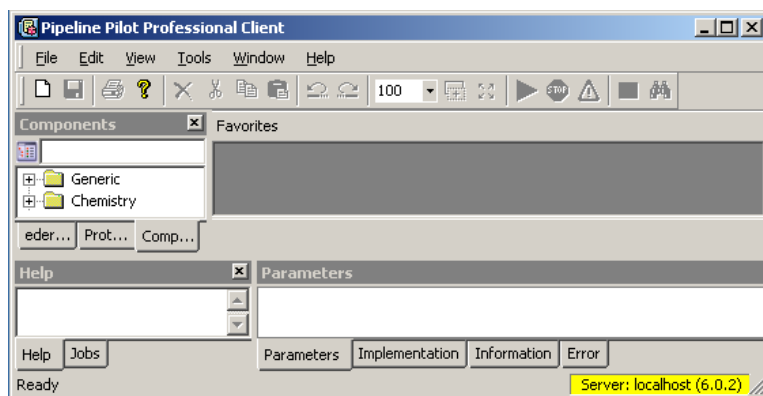


Figure 3.1: See where your Pipeline Pilot Server is installed.

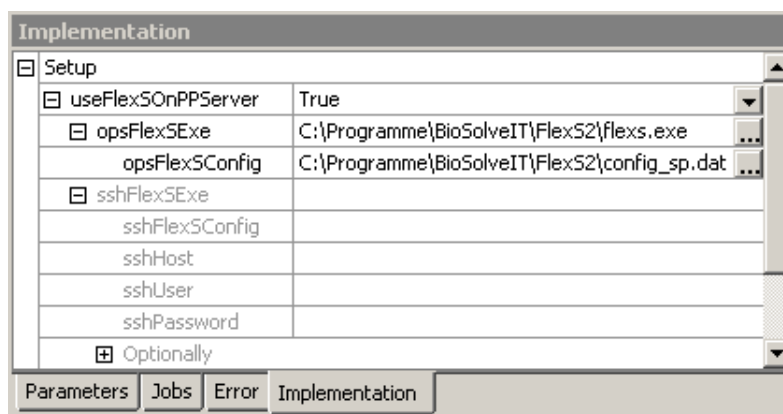


Figure 3.2: *ops*: Connect to an *FlexS* Installation on the Pipeline Pilot Server

Pipeline Pilot logs into the Linux machine using `ssh`, having copied all relevant files to the machine, and will run the calculation there, finally copying back all data it needs to the Pipeline Pilot server to continue with the pipeline.

This method is given the label `ssh`. You can see an example in figure 3.3.

- Requirement:
 - `FlexS` is installed on a Linux machine available to the Pipeline Pilot server via `ssh`.
- General Steps:
 1. Set `<UseInternalFlexS>` to `False`
 2. Set `<UseFlexSOnPPServer>` to `False`
 3. Expand `<sshFlexSExe>`
 4. For the parameter `<sshFlexSExe>`, enter the path to the `FlexS` installation on the Linux machine. For example:
`/software/BioSolveIT/flexs/bin/flexs`
 5. For the parameter `<sshFlexSConfig>`, enter the path to the `FlexS` configuration file. For example:
`/software/BioSolveIT/FlexS/config_sp.dat`
 6. For the parameter `<sshHost>`, enter the Linux machine host name
- User specific steps:
 1. For the parameter `<sshUser>`, enter the user login name for `ssh` on the Linux machine
 2. For the parameter `<sshPassword>`, enter the user password for `ssh` on the Linux machine
 3. There are more advanced options to be found under `<Optionally>` for more specific `ssh` parameters. Note the option `<sshDeleteResults>`, this may be useful for trouble-shooting later.

You can save these settings in the component (be sure not to save your own user specific login details in components available to others!).

All files necessary for the `FlexS` calculation will be transferred via `scp` between the Pipeline Pilot server and `ssh` Linux machine. Files copied and files created the remote server are automatically deleted at the end of the job leaving no trace. However, in case the user would like to leave a copy of the calculation and result files on the Linux machine, or for trouble-shooting as mentioned above, it is possible to set a parameter to tell Pipeline Pilot not to delete these files:

`<Optionally>` : `<sshDeleteResults>` : `False`

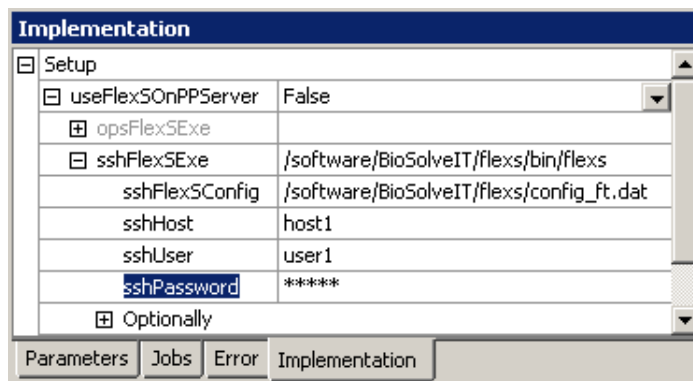


Figure 3.3: *ssh*: Connect to an *FlexS* Installation on a Remote Linux Server via *ssh*

3.3 Running *FlexS* in Parallel in Pipeline Pilot

FlexS in Pipeline Pilot takes advantage of the parallel computing options available in Pipeline Pilot to speed up longer calculations. This section tells you how to adjust the options to your system and needs. You will find the options in the Implementation tab, as in figure 3.4.

The most important limitation to a parallel processing calculation is the number of *FlexS* licenses that you have. If you only have a single license then parallel calculations will not be possible. Further choices in the set-up of the parallel computing calculation depend on the number of Pipeline Pilot licenses you have plus your choice of connection to the external *FlexS* installation (*ops* or *ssh*).

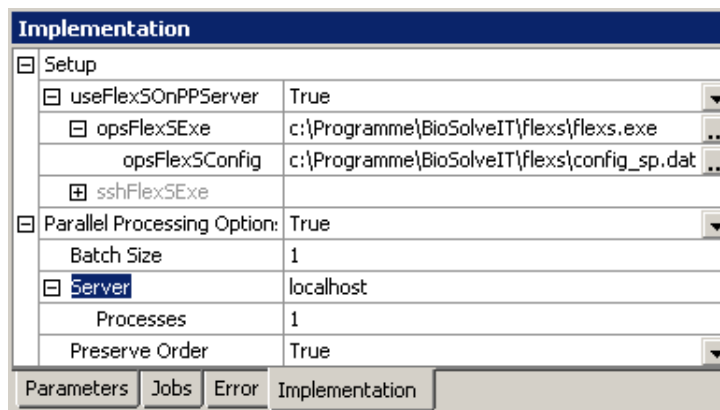


Figure 3.4: The options for tuning parallel processing are found in the Implementation tab

It is important to note that a balance must be achieved between the overhead caused by running several calculations instead of one and the size of the calculation – there is a lot of overhead involved in sending all the essential data to different computers and collecting the results. For the *FlexS* component, however, we still think parallel computing will work also for parallel jobs with batch size 1. Of course, these figures depend on the speed of your machines and network: experiment with your set-up if you intend to carry out large calcu-

lations often. Also we advise you to read the Pipeline Pilot documentation about parallel processing to understand more fully how it works.

Note: to set up large parallel processing jobs, you will need Administrator rights to change one setting.

3.3.1 Running in Parallel on the Pipeline Pilot Server

You may have a multi-processor machine as your Pipeline Pilot server. If you also have the appropriate number of Pipeline Pilot and *FlexS* licenses the simplest way to start a parallel calculation is to raise the number of processes to the number of processors of the machine.

You could also have more than one Pipeline Pilot server available in your network. If so, you can enter a list of the server names at the parameter <Server>. Below that, for the parameter <Processes> enter a list of the number of processes each server should receive. The lists are both comma separated and must be in corresponding order. Remember to adjust also the <Batch Size> accordingly.

Note: the path to the external installation of *FlexS* must be same on all servers!

3.3.2 Running in Parallel on a Remote Linux Cluster

We have developed an implementation in the component whereby a large cluster can be incorporated to run *FlexS* jobs, without them having to be Pipeline Pilot servers. However, it must be a Linux cluster and the component must use the ssh method. You must also have enough *FlexS* licenses available to the cluster.

The settings in the Implementation tab must be made as for running the ssh method, with two important changes. Instead of one <sshHost>, enter a comma separated list of the host names in the Linux cluster. Then, you must choose how many batches in total you want the job to be split into and enter this total in the <Processes> parameter. Remember to change the <Batch Size> to fit with the number of processes.

Figure 3.5 may help clarify how the method works.

The Pipeline Pilot server will run with the number of processes given in <Processes>. However, in this case the processes the server receives are not the *FlexS* calculations themselves but instructions for starting the ssh jobs. The server iterates through its 10 jobs each time spawning an ssh job on a Linux host. Beware that the Pipeline Pilot server does not know how many processors the Linux hosts have so make sure you choose the number of <Processes> to fit the number of Linux hosts and their number of processors respectively – be careful not to overload the Linux hosts.

Figure 3.6 shows the detailed settings required to set up the calculation as in figure 3.5.

As you may already have realised, you could enter more than one Pipeline Pilot server at the <Server> parameter (along with another entry for number of <Processes> as a comma separated list) to execute a doubly parallel calculation!

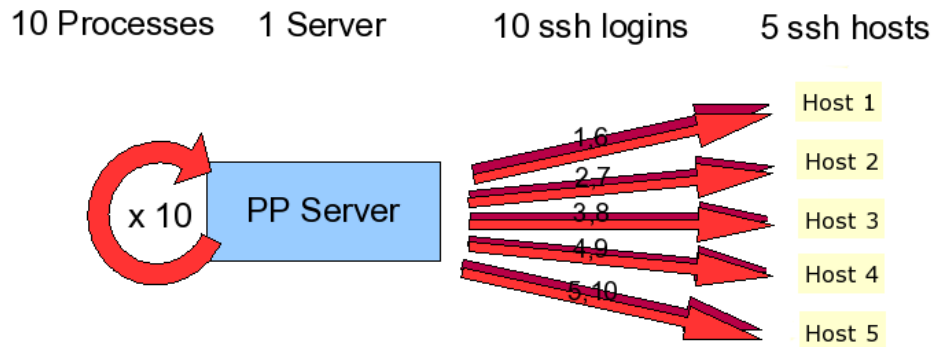


Figure 3.5: The method of parallel processing on Linux clusters in the *FlexS* component. The Pipeline Pilot server is given 10 processes. The 10 processes start an ssh job respectively, distributed amongst the ssh hosts.

Note: to get this method to work you will need to change the maximum number of processes per Pipeline Pilot server. The Pipeline Pilot Client will let you enter any number for the <Processes> parameter and does not warn you if this number is above the maximum.

Changing the maximum number requires Administrator rights! The number of processes per Pipeline Pilot server is usually restricted to the number of processors of the server. You must override this maximum to be able to set the number of processes you want for your parallel calculation. In the above example, the maximum must be set to 10 or more. Take the following steps:

- Go to the 'Scitegic Server Home Page', for example, via the Help menu in your Pipeline Pilot client.
- Click on 'Pipeline Pilot Administration Portal' and log in with the Administrator user name and password.
- In the last field of the table ('Maximum number of simultaneous parallel processing subprotocols allowed') change the value, click 'Save' and log out again.

Implementation	
[-] Setup	
[-] useFlexSOnPPServer	False
+ opsFlexSExe	
[-] sshFlexSExe	/software/BiosolveIT/flexs/bin/flexs
sshFlexSConfig	/software/BioSolveIT/flexs/config_sp.dat
sshHost	host1,host2,host3,host4,host5
sshUser	user1
sshPassword	*****
+ Optionally	
[-] Parallel Processing Options	True
Batch Size	1
[-] Server	localhost
Processes	10
Preserve Order	True

Parameters Jobs Error Implementation

Figure 3.6: The options for parallel processing for the example shown in figure 3.5 would look similar to these

3.3.3 Example Scenarios and Required Settings

ssh/ops	<i>ops</i>
Parallel Processing Options	False
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	-
Behavior	The calculation will run as one complete job on the Pipeline Pilot server
ssh/ops	<i>ssh</i>
Parallel Processing Options	False
Number of PP servers in list	-
Number of ssh Hosts in list	1
Number of Processes	-
Behavior	The calculation will run as one complete job on the ssh host
ssh/ops	<i>ops</i>
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	1
Behavior	(Default) Induces the 'pipeline' effect: the job will run in chunks on the server processor(s)
ssh/ops	<i>ops</i>
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	-
Number of Processes	>1
Behavior	A true parallel effect: the job will be run in chunks on the server processors
ssh/ops	<i>ops</i>
Parallel Processing Options	True
Number of PP servers in list	4 (each with two processors)
Number of ssh Hosts in list	-
Number of Processes	2,2,2,2
Behavior	A true parallel effect: the job will run in chunks in parallel split across 8 processors
ssh/ops	<i>ssh</i>
Parallel Processing Options	True
Number of PP servers in list	1
Number of ssh Hosts in list	5 (each with 2 processors)
Number of Processes	10
Behavior	A true parallel effect: the job will run in chunks in parallel split across 10 processors

Trouble Shooting

4.1 Problems connecting to the external installation

Other most commonly seen problems with *FlexS* in Pipeline Pilot is with the connection to the external *FlexS* installation. For one thing, *FlexS* itself must be correctly installed on the system independently of Pipeline Pilot – it is essential first to make sure this is the case (especially to make sure that *FlexS* can locate the licenses). Once *FlexS* runs fine on your system, the remaining key task is to make sure the paths to the executable and configuration file are correct within the Pipeline Pilot component (pay extra attention to the setting of `<useFlexSONPPServer>` and the component should be fine.

When the error messages pop up, they may contain a relevant part of the start-up message from *FlexS* plus an extra message that may help you identify the problem. The message is easier to see if you click on 'Details' in the error message box, as in figure 4.1.

In fact, the best way to read the complete message is to go to the 'Jobs' tab below the Protocol workspace and check under the last run job for a file called 'Errors.txt' – as in figure 4.2. Clicking on the link brings up the error report in a browser.

A correctly started *FlexS* job outputs the following header: if there is a problem you will see some of this header and the point where the problem occurs:

```
                                F l e x S
Copyright                        Prediction of Ligand Superposition

BioSolveIT GmbH                 Version:   1.20.3      (22.08.06)
An der Ziegelei 79              Modules:  [PVM] [CSUPER]
53757 St. Augustin
Germany                          Authors:  C.Lemmen, M.Rarey, T.Lengauer,
                                C.Hiller, B.Kramer, M.Lilienthal,
                                F.Sonnenburg, M.Zimmermann
www.biosolveit.de               Contact:  flexs@biosolveit.de
```

Additional copyright notes:

Software-basis: (C) 2001 by Fraunhofer Gesellschaft (FhI-SCAI)
Getline library: (C) 1993 by Chris Thewalt

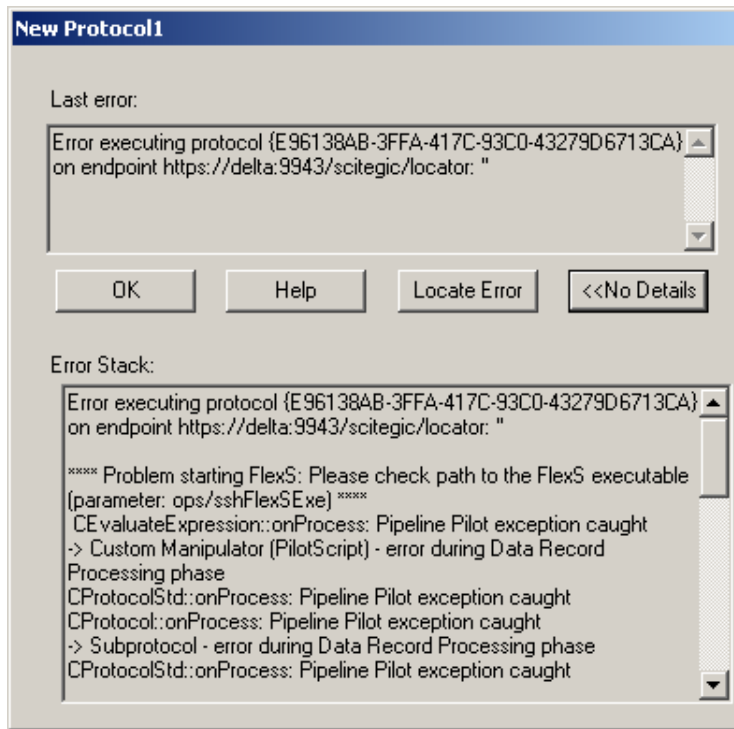


Figure 4.1: An error box reporting that the *FlexS* exe could not be found.

Protocol Name	Status	Start D...
New Protocol1	Not Started	
New Protocol3	Error	Tue Jun/17/...
Errors.txt		
New Protocol4	Error	Mon Jun/16...
New Protocol48	Finished	Fri Jun/13/2...
New Protocol45	Stopped	Fri Jun/13/2...
New Protocol44	Finished	Fri Jun/13/2...
New Protocol21	Finished	Fri Jun/13/2...
New Protocol23	Error	Fri Jun/13/2...
New Protocol4	Finished	Fri Jun/13/2...
New Protocol1	Finished	Fri Jun/13/2...

Figure 4.2: The full error report can be found in the 'Jobs' tab.

```
PVM library: (C) 1997 by University of Tennessee, Knoxville TN
Torsion angle data: (C) by GMD SCAI, CCDC, BASF AG

>> Running on gamma (Linux 2.6.25.11-0.1-pae) with 4 processors.
>> FlexS configuration file '/software/BioSolveIT/flexs/config_sp.dat' loaded.
>> FlexS_base license check (BioSolveIT keys): succeeded.
...
```

4.2 Problems using the *ssh* Method

You may also experience problems using the *ssh* login, for example, the user name is unknown or the host is not found.

4.3 Further help

More complicated errors may arise during the running of *FlexS*. Again though, the errors will be collected and as much information shown as possible. If you are familiar with *FlexS* you may want to take a look at all the output of the job yourself to see if you can recognize the problem. In this case, you can look in the temporary folders Pipeline Pilot sets up internally to find the output, or if you are working with the *ssh* method, set the parameter `<sshDeleteResults>` to *False* so that you may then find the files retained on the *ssh* host: these will be in the directory set under the *ssh* parameter `<sshTempPath>` (see the help text associated with this parameter to find its default value) – essentially a cryptically named folder whose name begins with the date and time of the job!

If you still do not know what is causing the errors, write down as much information as possible relating to your installation scheme and cut and paste the text from the 'Errors.txt' file or, if possible, the output files from the *FlexS* job itself into an email. Send all the information to:

support@biosolveit.de

Tips and Tricks

5.1 Other Significant Parameters in the *FlexS* Components

For detailed documentation of all parameters, refer to the documentation you find in the *Help* area of the Pipeline Pilot window.

We list here particularly interesting parameters: those that greatly influence the workflow or change the outcome of calculations, or those that may help you understand what is happening in the component.

5.1.1 <sshHasSameFS>

*FlexS*Implementation : Setup : UseFlexSOnPPServer : sshFlexSExe : Optionally : <sshHasSameFS>

The name of this parameter stands for "ssh Has Same File System". Normally, for an *ssh* job, Pipeline Pilot must first copy all the data required by *FlexS* to the <sshHost> using scp. This is time consuming. It is possible that the Pipeline Pilot Server and <sshHost> actually share the same file system rendering the scp process unnecessary. Select True if the Pipeline Pilot Server and <sshHost> share the same File System - no copying of data is necessary. Selecting False means Pipeline Pilot copies all the data to and back from the <sshHost> – this is just a little slower but will always still work. Leave the parameter set to False if you are uncertain!

5.2 Accessing Other Domains within Pipeline Pilot

Often in house data or even your own working data are accessible from a windows computer via a domain (a path starting for example 'z:\...' or '\\...\') which you cannot find from within Pipeline Pilot. That means you must first literally transfer the data to the Pipeline Pilot Server itself.

If you are using a Linux Pipeline Pilot server, this hint does not apply.

To get around this problem and make the Pipeline Pilot working environment much more flexible you can allow users access to domains – you need Administrator rights to be able to do this! Also check first that you should change these settings as they may have already been set to fit the current environment.

- Go to the 'Scitegic Server Home Page', for example, via the Help menu in your Pipeline Pilot client.
- Click on 'Pipeline Pilot Administration Portal' and log in with the Administrator user name and password.
- In the Security tab go to Authentication.
- For the 'Authentication Method' choose 'DOMAIN' and a set of parameters will appear.
- Enter the domain name in the field 'Domain' and choose 'Full' for 'Impersonation'
- Choose 'DOMAIN' for 'Retrieve Groups' and leave 'Limit access to listed domains' set to 'No'
- click 'Save' and log out again.

After you have done this you will need to enter your domain login details when you start the Pipeline Pilot Client.