**Parameter file - parameters and variables:**

It is always a best practice in coding that you should never hard code the values. The same applies to Informatica as well. It is always better to pass the values using parameters or variables in place of hard coding them. When you define parameters or variables in the code, you need to pass the values to those parameters and variables. A parameter file serves that purpose. Any value that you hard code can be passed through the Parameter file. You can define the Parameter file at the session level and workflow level.

You must have noticed that the system defined the variable *$PMSourceFileDir\* or *$PMTargetFileDir\*. Similar to that, we can define User Define variables. You can define the variable at both the mapping level and workflow level.

*If the value passed remains constant across the session run, it is called***parameter***, and if the value changes across the session run, it is called***variable***.*

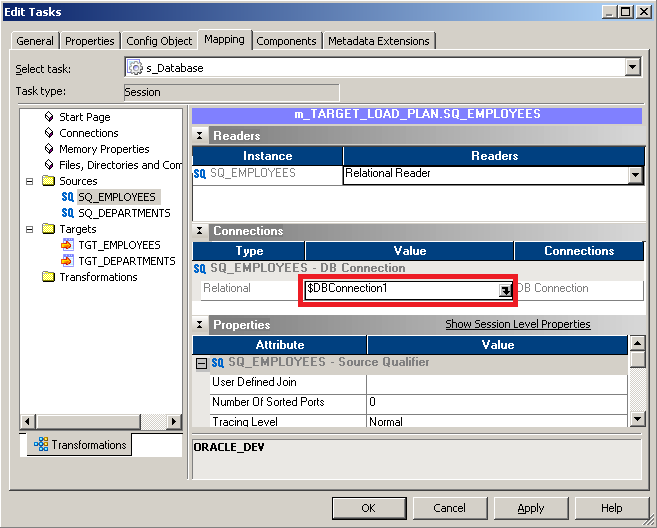
Let's take an example to understand the Parameter file. You are aware that Informatica will have three different repositories to cater to the need of three regions. Let's say we have three repositories REPO\_DEV, REPO\_TEST, REPO\_PROD serving the development, testing, and production regions respectively. Also, corresponding to three regions, we have three oracle databases--ORACLE\_DEV, ORACLE\_TEST, and ORACLE\_PROD respectively. When you start the coding in the development region under REPO\_DEV, you will hardcode the database connection value to ORACLE\_DEV. Your code is working successfully, and when you want to deploy the code to test the region, you will need to replace the database connection value to ORACLE\_TEST manually. Changing the code after testing is not allowed.

The same case applies when you wish to deploy the code from test to production. The solution comes as a Parameter file. Parameter files serve the purpose of passing the value based on the region in which you are running the code. We are defining the Parameter file for passing the value for the source database connection (*$DBCONNECTION1*), target file name (*$TGTFILENAME*), e-mail recipient (*$EMAILUSER*), and a mapping-level variable for the location ($$LOCATION). We are using the example of session-level variable, workflow-level variable, and mapping-level variable so you understand clearly how it works.

**Defining session-level variables**

The variables that are defined under session task are called session-level variables. There are various values that can be passed via variables, such as source, target data base connection value, source/target file name, source/target file path, session log file name/path, and so on. session task does most of the work in Workflow Manager screen and, hence, has been assigned special privilege. To pass the value through a variable, simply replace the hard-coded value with a variable of your choice.

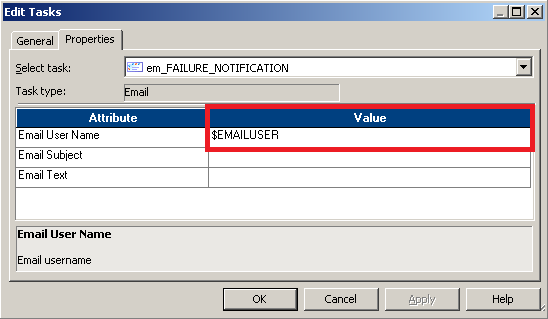
As shown in the following screenshot, we are using $DBCONNECTION1 as the database connection variable:



Similarly, assign the value to the target file name (*$TGTFILENAME*). Since we define the target file name under the session task, it will be referred to as session-level variable.

**Defining workflow-level variables**

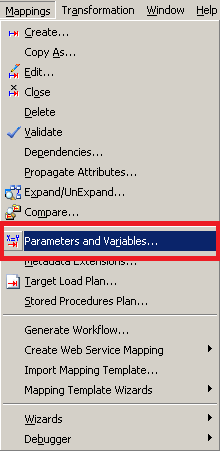
The variables that are defined under various tasks are called workflow-level variables. Note that you can define the session-level variables under workflow also in the Parameter file. In our case, we are passing an e-mail user value as the variable. To assign the value, simply replace the hardcoded value by the variable in the e-mail task as shown in the following screenshot:



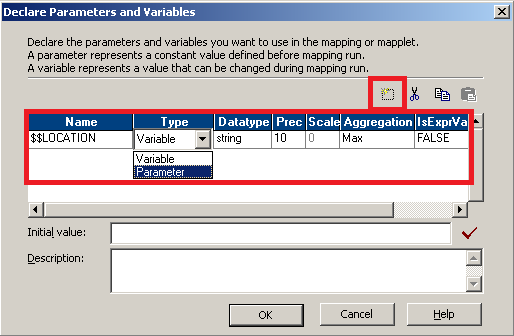
**Defining mapping-level variables**

As mentioned earlier, you can define the variable for the hardcoded values in mapping inside transformations also. You need to define parameters or variables under mapping before you can use them in transformations, else the mapping will become invalid. To add the values, perform the following steps:

1. Open the mapping for which you wish to add variables in the Mapping Designer, go to Mappings | Parameters and Variables:

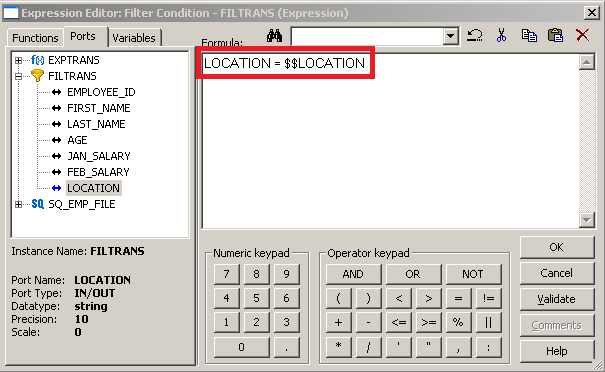


1. In the next screen, click on Add a new variable to this table. Define the variable, and select the type as Parameter and Variable based on your requirement. Click on OK:



As you can notice, mapping-level variables are always defined as $$.

Open the transformation to which you wish to assign the variable or parameter. We are using filter transformation to pass the value:

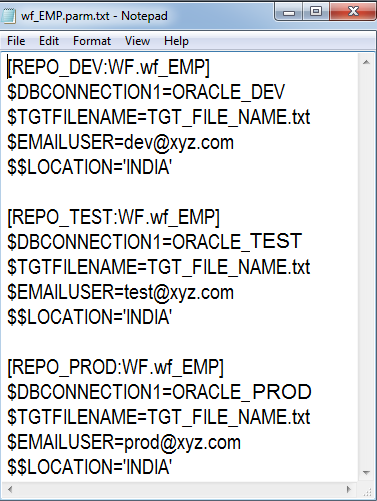


As you can notice, we have assigned $$LOCATION to the filter condition.

With this, we are done with defining the variables or parameters. Next, we will see how to pass the values to these using a parameter file.

**Creating a parameter file**

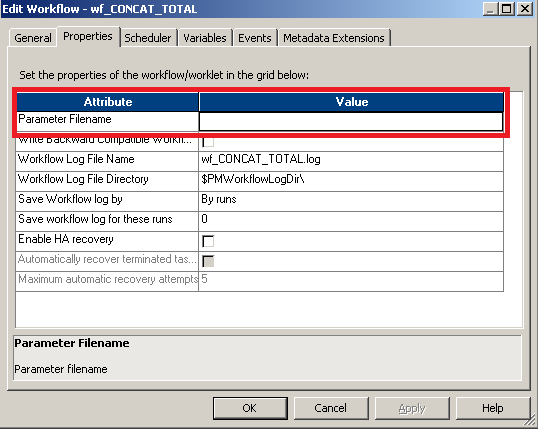
A Parameter file is nothing but a simple .txt file that contains the values of the variable to be passed A sample Parameter file for the variable defined in the previous steps is shown as follows:



As you can see, the Parameter file contains the values of the variables to be replaced in the three regions. Informatica matches the Repository name against the name of Repository defined in the Parameter file and replaces the values of the variables internally before it runs the workflow. wf\_EMP is the name of workflow for which you defined the Parameter file. So, suppose you are running in the production region, Informatica will match the Repository name REPO\_PROD against the same name in the Parameter file, replace the variables with the value internally, and execute the workflow with the replaced values.

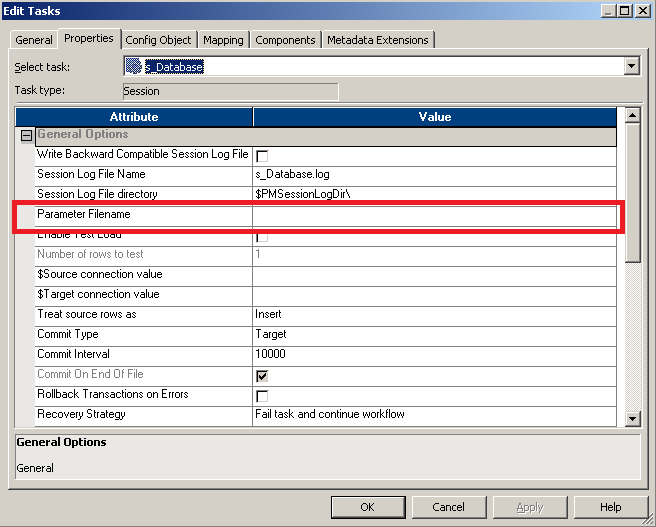
**Mentioning the Parameter file at the workflow level**

To define the Parameter file at the workflow level, open Workflow Manager, go to Workflow | Edit | Properties. Specify the path and name of the parameter file for the attribute as indicated in the following screenshot:



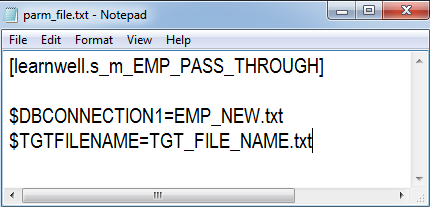
**Mentioning the Parameter file at the session level**

To define the Parameter file at the session level, open the session in the Workflow Manager, double-click on the session task, and click on Properties. Specify the path and name of the Parameter file for the attribute as indicated in the following screenshot:



You have learned how to use parameters and variables, create a parameter file, and define the Parameter file. Another importance of parameter file is that if the value of any variable is changing, you need not modify the code; simply change the value in the Parameter file, and the changed value will take effect the next time you run the code.

Also, note that Parameter files can be used to replace the values for the individual session run. They can also be defined at the folder level as against at repository level as seen earlier. It is not mandatory to define the workflow for all three regions. A sample Parameter file for an individual session run (s\_m\_EMP\_PASS\_THROUGH) defined at the folder level (learnwell) is shown in the following screenshot:



Parameters are different from variables in the fact that:

 The value of a **parameter is fixed** during the run of the mapping

 **Variables can change** in value during run-time   
  
Both parameters and variables can be accessed from any component in the mapping which supports it.

**Changing values of Variables**

To change the value of a variable, one of the following functions can be used within an expression: SETMAXVARIABLE($$Variable, value) , SETMINVARIABLE($$Variable, value), SETVARIABLE($$Variable, value) , SETCOUNTVARIABLE($$Variable), where:

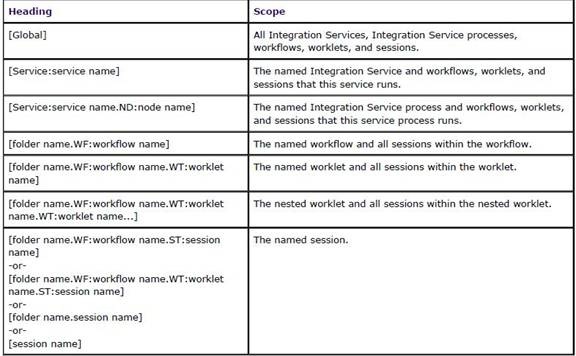
SETVARIABLE sets the variable to a value that you specify (executes only if a row is marked as insert or update). At the end of a successful session, the Integration Service saves either the MAX or MIN of (start value.final value) to the repository, depending on the aggregate type of the variable. Unless overridden, it uses the saved value as the start value of the variable for the next session run.

SETCOUNTVARIABLE - increments a counter variable. If the Row Type is Insert increment +1, if Row Type is Delete increment -1. A value = 0 is used for Update and Reject.

SETMAXVARIABLE - compare current value to value passed into the function. Returns the higher value and sets the current value to the higher value.

SETMINVARIABLE - compare current value to the value passed into the function. Returns the lower value and sets the current value to the lower value.

At the end of a successful session, the values of variables are saved to the repository. The SetVariable function writes the final value of a variable to the repository based on the Aggregation Type selected when the variable was defined.



Sample parameter file:

[Service:IntegrationSvc\_01]

$$SuccessEmail=dwhadmin@etl-tools.info

$$FailureEmail=helpdesk@etl-tools.info

[DWH\_PROJECT.WF:wkf\_daily\_loading]

$$platform=hpux

$$DBC\_ORA=oracle\_dwh

[DWH\_PROJECT.WF:wkf\_daily\_loading.ST:s\_src\_sa\_sapbw]

$$DBC\_SAP=sapbw.etl-tools.info

$$DBC\_ORA=oracle\_sap\_staging

**The Order of Execution of Paramater Values in Informatica:**

1. Paramaeter File

2. Value Assigned in Pre-Session Variable Assignment

3. Value saved in the Repository

4. Initial Value defined for the variable at declaration level

5. The default value defined by Informatica for the Datatype of the defined variable

Informatica Checks for the variable values in the above order and stops and assigns the value to the parameter/variable when it finds a value.

**Note:*In case there are values defined in two parameter files and the Parameter files are assignd at both Session Level and Workflow level, The Paramater file defined at the Workflow level always overrides the Paramater file defined at the Session Level.***

How these variable assignment actually, works :

1. Paramaeter File  
Assign a value to the variable in the parameter file itself

2. Value Assigned in Pre-Session Variable Assignment  
You can assign values for a variable explicitly using the Pre/Post Session variable assignment in the Session Properties.

3. Value saved in the Repository When you declare a variable, you also declare the aggregation type of that variable. It could be Min, mix or count. Based upon this declaration after the session is ended , the Integration Service writes and saves the final value of the variable into the Repository. SO, from the next load onwards, the variable displays the value stored in the Repository.  
Note: You can check the value stored in the Repository by righclicking on the session using that variable and clicking “View persistent values.”

4. Initial Value/Default Value defined for the variable at declaration level You can provide an initial value to the variable while declaration itself.

5. The default value defined by Informatica for the Datatype of the defined variable In case, the Inetgration service doesn’t find any value assigned against the declared Variable, it assigns the Default value for the datatype of which the variable is declared.

"Hi experts, I read through various posts which stated that session parameter file is ignored when workflow parameter file is declared. However in help file there is section where it states that there is a solution via using the $PMMergeSessParamFile ( HELP FILE Session parameter file is ignored when a workflow parameter file is specified while running a PowerCenter session Problem Description A PowerCenter workflow with a session is configured with a parameter file in the session properties and the workflow properties. When running the workflow, the session parameter file is ignored and the workflow parameter file is used. Cause By default, when you define a workflow parameter file and a session parameter file for a session within the workflow, the Integration Service uses the workflow parameter file, and ignores the session parameter file. Solution You can force the workflow to use both the workflow parameter file and session parameter file by adding the following to the workflow section of the workflow parameter file: $PMMergeSessParamFile=TRUE This must be set for the section defined for the workflow in the workflow parameter file. Example [DEV\_FOLDER.wf\_workflow] $PMMergeSessParamFile=TRUE Note If you define the same parameter or variable in both the session and workflow parameter files, the Integration Service sets parameter and variable values according to the following rules: • When a parameter or variable is defined in the same section of the workflow and session parameter files, the Integration Service uses the value in the workflow parameter file. • When a parameter or variable is defined in both the session section of the session parameter file and the workflow section of the workflow parameter file, the Integration Service uses the value in the session parameter file. ) I tried this solution but it didn't work. Do you have any idea of what might be the good solution since i desparately looking for someway to use both Workflow parameter file and session parameter file in a workflow. For example ,Here is my Workflow parameter definition: $$Environment=DevEnvironment email@removed email@removed $PMMergeSessParamFile=TRUE Any help is appreciated,"