

????????? Module Documentation

Contents

TBA, Scan Selection and Pass-off Module Information..... 3

DataTypes.....4

Section header..... 5

 Test Bed Analysis Task User Guide..... 5

TBA, Scan Selection and Pass-off Module Information

Table 1: Module Information

Title	Funtionality XXXX
Module ID	com.rr.perf.tbapassoff
Module Version	1.4
User Authority	Dave Widdowson
Computing Authority	Matt Wood
Technical Authority	Owen Cumpson
Team Leader	Nicolas Balkota

Table 2: Module history.

Date	Version	Description	Phobos Release
15 June 2012	1.4	Improvements resulting from user feedback.	8.6.7
29 June 2012	1.5	Improvements resulting from user feedback.	8.6.8

DataTypes

Section header

Test Bed Analysis Task User Guide

Background

Test Bed Analysis or TBA is used to calculate derived parameters from basic engine parameters (often referred as measured or calibrated values). TBA are used during an engine test and the resulting derived parameters provide information about component performance eg. efficiencies and pressure ratios. Generally a TBA program is used to perform this calculation. In some cases, this program is specific to an engine and in others, the same program will be used for a range of engine types.

Most TBA programs required some data input to tailor their behaviour to a specific engine but the amount of data varies between engines. Examples of data used by a TBA is:

- Cycle Data
- IPNPPN Relationships
- Graphs
- Test Constants
- Manual Input Values

This data is provided from Phobos and passed to the TBA program via the Phoenix interface. The Phobos TBA Task is used to configure the TBA program and TBA data required for an engine.

TBA Program

This is a natively compiled program (e.g. QT65 is written in FORTRAN) supplied with the many forms of data and actually does the thermodynamic calculations. There are also different types of TBA i.e. some are RRAP based and others are bespoke so it is important to be aware of these differences. At the moment the only TBA program validated with Phobos is QT65.

Phobos TBA Task

This provides the User interface to the TBA and effectively defines all the data that is supplied to the TBA program. Due to the many types of TBA program it is important to be aware of the specific data input requirements when using it for a new TBA program.

The TBA Task is a standard Phobos task where profiles are used to configure and record the various inputs required. The TBA task consists of the following sub-tasks, executed in that order:

1. Data Selection Task
2. Let Task 1 - Pre-Rake Profiling
3. Rake Profiling Task
4. Let Task 2 - Post-Rake Profiling
5. TBA Task
6. Lest Task 3 - Post TBA Task

These are shown in the picture below and each task is described in detail in the subsequent sections.

The general task control window allows user to choose how to view or file the results when the profile is run.

1. Data Selection Task

This is a generic sub task that allows the user to specify the input curve that has one or more scans of measured data. Curve can be added and removed using the relevant buttons, which can display tool tips (help note triggered by stopping the mouse cursor over a user interface element).

Phobos will run the TBA program on the selected curve and create an output curve version that contains the derived values.

2. Let Task 1 - Pre-rakeprofiling

Generally used for applying scaling factors or deltas to measured data. The available options are:

Table 3: Let Task Options

Option	Description	Recommendations
LetSet	Specify the name of the LetSet variant that contains the expressions.	Only specify when let expressions are required. The let task will be ignored if not set.
Version Name	The curve version will retain the name of the input curve this will define its version.	Leave blank since the TBA task will define the version name.
Fail on Error	If ticked, a let statement failure (e.g. syntax, missing variables) will stop the profile execution at this stage.	Should be ticked when the let expressions are critical to the subsequent tasks.

This let task allows user to:

- Apply modifiers (e.g. scale or delta) to input measured values (IPNs or MIVs).
- Reset the values of measured values (IPNs or MIVs).
- Create and set new measured values, which can then be used in the Rake Profile / averaging task.
- **Do Not** Create or modify Averaged Values (part of PPNs) at this stage since they will be overwritten by the Rake Profile / averaging task.



Note: The data on the input curve version is not changed by this Let Task, only the data on the output curve-version is affected.

3. Rake Profiling Task (Averaging)

This is a generic sub task that allows both Rake Profiling and Averaging data. For Test Bed Analysis it will generally only be used for Averaging.

Table 4: Rake Profiling Options

Option	Description	Recommendations
Rake Profing Set	Specify the RakeProfilingSet name that contains the IPN/PPN relationships.	
Curve Version Name	The curve version will retain the name of the input curve. This will define its version.	Leave blank since the TBA task will define the version name.
Print Option	Select level of print output for profiling and averaging diagnostics.	Use default of None unless diagnosing problems, can be verbose.

Option	Description	Recommendations
Tolerance Check IPNs	Tick to enable IPN value tolerance checking.	Will only apply the tolerance checks defined in rake profiling set.
Update IPN values	Tick to allow update of rejected IPN values as per the replacement method set in the RakeProfile.	



Warning: If all the IPNs that a PPN is attached to are either zero or not available then the PPN is set to the PPN's default value which in Phobos has a default of -9999.0. This is unlike Alice which uses 0.0 by default. This value should be as expected by the TBA. Hence, it is strongly recommended that the default values for all of the PPNs in the Rake Profiling Set are set to 0.0 if TBA program, such as QT65, expects the ALICE convention.

4. Let Task 2 - Post-rake profiling

Generally used for applying scales & deltas to averaged data. The options are:

- Apply modifiers (e.g. scale or delta) to averaged data.
- Reset the values of averaged data.
- Create and set new averaged data. (which can then be used in the Test Bed Analysis program).



Warning: Do Not: Modify Measured Values at this stage since the associated averaged value will not be altered.

5. Test Bed Analysis Task

This is the key sub-task that is used to identify the TBA program and the necessary data. It then runs test bed analysis program and handles the results according to the options selected in the task control window.

Table 5: TBA Options

Option	Description	Recommendations
Model	A standard performance model referencing a ProgramVersion to define the TBA executable and various data files to define the graphical data.	
Model Changes	Model changes can be entered here	
Test Constants	Used to reference a set of name/value pairs that represent test constants.	Can not be used to set IPN or MIVs values.
CaseSet	Generic place holder to define extra variables or run conditions specific to a TBA program.	Some TBA programs like QT65 do not use a CaseSet. Ignore for those.
NamesList	Defines the derived value names (PPNs) that will be stored on the output curve.	Required or no derived values will be filed.
Version Name	This defines the version name given to each derived curve version. The curve name will be the same as the input curve.	If the TBA profile is run automatically via EPAP, then this value will be overwritten to "BASIC_DERIVED".

Table 6: Advanced TBA Options

Option	Description	Recommendations
Extra MC	Place holder for Extra Model Changes input. Typically used when the profile is run automatically via EPAP (AUTO_MODEL_INPUT).	This file is not stored in the database, only accessible from the profile or the audit trail. It should not be used when running TBA manually, only relevant to EPAP automatic profile.



Note: ISSUE 1:- Observe the guidelines about what type of parameters are specified in each of the 3 let sets and note that parameters with the same name may be overwritten by different parts of the task, this should be avoided. The explanations and advice for each of the lets described in this document should be followed.



Note: ISSUE 2:- The let 1 and 2 output versions do not get stored as output curves, however if the third output let task has a version name this will be the final stored curve version for the whole TBA task.



Note: ISSUE 3:- An improvement for future releases may be to have one final output curve version for the whole task, or to have the intermediate versions also filed.

The TBA sub task will read the inputs to the TBA executable in that order:

1. Model and dependencies from the Cycle Definition File
2. Model Changes
3. Extra Model Changes (from the advanced tab, AUTO_MODEL_INPUT in case of an automatic profile used by EPAP)
4. Test Constants



Note: The last item that contains an input overwrites any input from previous items in the above list.

6. Let Task 3 - Post TBA

This exists to allow creation of extra variables used in the quality plots. Thus, replicating the ALICE functionality used in Derby where the auto profile would perform a Let following the TBA. As this let set is run last during the profile execution, it could be used to alter any existing parameter (IPNs, MIVs or PPNs), but it should only be used to create new parameters as required by the quality checking plots.



Warning: Do Not: Modify Derived parameters as it will override the TBA parameters.

Task and Program Output

All tasks in Phobos use the output window to provide a summary of progress. A new tab will appear in the Phobos Output Window when a task is run to display the logging information as the profile is running.

Most TBA programs will write output to one or more streams/files and the level of output can be controlled by program flags. When Phobos runs the TBA program, it defines the location to store this output, which can be viewed using the output viewer. To see the output either click on the hyperlink that is displayed in the Output Window or **Windows > Open Output Viewer**. In the output viewer, each run of a Phobos profile is identified as a folder that can be expanded to view the various output files logged by the tasks.