Honeywell

Profit Toolkit

FCCU Toolkit
User's Guide

For Open Systems

5/01

Rev 2.0

Honeywell

Profit Toolkit

FCCU Toolkit
User's Guide

For Open Systems

5/01

Rev 2.0

AP13-201

TotalPlant

Copyright, Notices, and Trademarks

Honeywell
Industrial Automation and Control
16404 N. Black Canyon Hwy
Phoenix, AZ 85053

Table of Contents

About This Publication

Statement of Work

The following table describes the audience, purpose, and scope of this book:

Purpose	This book explains how to use the FCCU functions. It describes the inputs, outputs and error messages associated with each function.
Audience	Process and control engineers

Release Information

This is document version 2.0 for FCCU Toolkit R200.00.

RMPCT Course Information

Honeywell offers several courses that explain the math and conceptual underpinnings of RMPCT as well as application implementation of the Advanced Process Control suite of products.

Engineers wanting a more technical exposure to RMPCT can contact:

Honeywell Automation College

2500 W. Union Hills Drive

Phoenix, AZ 85027

602-313-5669

Writing Conventions Used in This Book

The following writing conventions have been used throughout this book and other books in the Profit Suite library.

- Words in double quotation marks " " name sections or subsections in this publication.
- Words in *italics* name book titles, add grammatical emphasis, introduce words that are being referenced or defined, or represent mathematical variables. The context makes the meaning and use clear.
- Words in **bold type** indicate paragraph topics or bring important phrases to your attention.
- Shading brings paragraphs and table entries to your attention.
- Windows pull down menus and their options are separated by an angle bracket >. For example, Under Settings> Communications, set the baud rate.
- Messages and information that you type appear in `Courier` font.
- Acronyms, Scan parameters, point names, file names, and paths appear in UPPERCASE. The context makes the meaning and use clear.
- Command keys appear in UPPERCASE within angle brackets. For example, press <ENTER>.
- TPS user station touch-screen targets appear in rounded boxes. For example, touch MODIFY NODE.
- Graphic buttons appear in UPPERCASE within brackets [ ]. For example, touch [TAG].
- Point-dot-parameter means a point name and one of its parameters. For example, point-dot-SP means the SP parameter for the point.
- Zero as a value and when there is a chance for confusion with the letter O is given as Ø. In all other cases, zero as a numerical place holder is given as 0. For example, 1.0, 10, 101, CVØ1, parameter PØ.
- The terms *screen* and *display* are used inter changeably in discussing the graphical interfaces. The verbs *display* a screen and *call* a screen are also used inter changeably.
- These names, and may be used interchangeably.

Former Name	Product Name
RMPCT	Profit Controller
DQP	Profit Optimizer
APC Development Environment or APCDE	Profit Design Studio
RPID	Profit PID

References

The following comprise the Profit Suite library.

Documentation

Title	Number
General	
Profit Controller (RMPCT) Concepts Reference	RM09-400
Profit Controller (RMPCT) Designer's Guide (Off-Line Design)	RM11-410
Profit Optimizer Designer's Guide (Off-Line Design)	PR11-400
Profit Toolkit Designer's Guide	AP11-400
APC Identifier User's Guide	AP09-200
Profit-PID (RPID)	RM11-100
Profit Sensor User's Guide	PS09-100
Open	
Profit Suite Installation Guide for Open Systems Viewer - Controller - Optimizer - Toolkit	RM20-501
Profit Controller (RMPCT) User's Guide for Open Systems	RM11-401
Profit Optimizer User's Guide for Open Systems	PR11-421
Profit Trender User's Guide	RM11 431
Profit Toolkit User's Guide for Open Systems	AP11-401
Profit Toolkit Function Reference	AP11-410
FCCU Toolkit User's Guide for Open Systems	AP13-201
Fractionator Toolkit User's Guide for Open Systems	AP13-101
Lab Update User's Guide	AP13-111
Wrapper Builder User's Guide	AP11-411
Profit Bridge User's Guide	AP20-401
TPS System	
Profit Controller (RMPCT) Installation Reference for AM, AxM and Open LCN-Side	RM20-400
Profit Controller (RMPCT) Commissioning	RM20-410
Profit Controller (RMPCT) User's Guide for AM, AxM and Open LCN-Side	RM11-400
Profit Optimizer Installation Reference for AM and Open LCN-Side	PR20-400
Profit Optimizer User's Guide for AM and Open LCN-Side	PR11-420
Profit Suite ToolKit TDC Data Converter, Performance Monitor, Simulation BackBuilder Data Collector, RMPCT Cascade, Gain Scheduler Step Test Builder, PV Validation	AP09-300
Fractionator Toolkit (LCN)	AP13-100
FCCU Toolkit (LCN)	AP13-200
Furnace Pass Balance Temperature Control User's Guide	AP13-600
Non-Linear Level Control User's Guide	AP09-700
Embedded Uniformance	
Excel Companion User's Guide (Profit Embedded PHD)	AP20-510
Power Point Companion User's Guide (Profit Embedded PHD)	AP20-520
Process Trend User's Guide (Profit Embedded PHD)	AP20-530

For Technical Assistance

If You Need Assistance	If you need technical assistance, contact your local Honeywell Service Organization, as explained in the following paragraphs.
International Customers	Outside of the United States, contact your local Honeywell Service Organization. If you are not sure of the location or telephone number, call your Honeywell representative for information.
Customers Inside the United States	Within the United States, call the Technical Assistance Center (TAC) at the toll free number 1-800-822-7673.
Arizona Customers	Within Arizona, the local number for TAC is 602-313-5558.
Services Provided	Calls to TAC are answered by a dispatcher from 7:00 A.M. to 5:00 P.M., Mountain Standard Time (6:00 A.M. to 4:00 P.M. when daylight savings time is in effect). Outside of these hours, emergency calls—those which affect your ability to control or view a process—will be received by an answering service, and returned within one hour. TAC maintains its own TPS network, and frequently can duplicate problems on this equipment.
Time Saving Tip	It is a good idea to make specific notes about the problem before making the call. This helps to reduce delays and expedite answers.

Section 1 - FCCU Toolkit Functions

1.1 Overview

In this Guide

This guide describes the Open FCCU Toolkit functions, how to use them, and the possible error codes associated with their execution. The toolkit functions are as follows:

DLLs

The following functions are located in **HSTool_FCCU.DLL**

Functions	Description
Feed_Prop	Consolidates all the properties associated with feeds.
Reactor_Prop	Consolidates all the properties associated with the reactor.
CycleOil_Prop	Consolidates all properties associated with the Cycle Oil product.
Gasoline_Prop	Consolidates all the properties associated with the gasoline product(s).
Measured_Conv	Calculates the conversion and severity based on the process measurements.
Cat_Circ_Pred	Calculates ASTM D86 or the EFV point of a petroleum fraction
Severity_Pred	Calculates the severity and conversion based on the FCCU model.
Product_Pred	Calculates the yield and production of a specified product.
Octane_Pred	Calculates the octane and octane flow of the gasoline product.
Calculation_ Comp	Calculates the bias for the severity and product prediction functions.
Dynamic_Comp	Transfer function for dynamically lagging a variable. See the Profit Toolkit Functions Reference document for a detailed description of dynamic compensation.

General Description of Functions

The FCCU toolkit functions fit into three categories. The first are property functions (“Prop”) which that consolidate characteristics associated with material and equipment in the FCCU.

The second are the main FCCU Toolkit prediction (“Pred”) functions that estimate the variables of interest for control and optimization. The resultant calculated value is filtered to reduce noise, biased and converted to toolkit units. Last good value holding is available for every function.

The third are compensation function (“Comp”) which compensates the bias and correction factors related to the main prediction functions.

Input Values

The user has to supply several inputs to ensure a function works properly. The inputs may come from a variety of places.

- User entered
- Input from a DCS or other external source via the DSS server
- Output from another function

The user may not be required to fill in each input for a given function. Depending on the usage of the function, the user may not have to fill in every parameter. The user can ignore inputs that are not required.

Filtering

Filtering is applied to reduce noise and to ramp a value from its previous value to its current value as smoothly as possible. Currently the filter types available are a first order lag filter and a first order lag filter with deadtime. These filters are invoked by selecting a filter type of 1 or 2 (several additional parameters also have to be filled in). Other filter types will be available in the future. If filtering is not required for a specific function, then a filter type of 0 should be selected. Biasing and conversion are sequenced immediately after the filtering.

For the first order lag filter types, in order to filter from a previous value to the current value, it is required to store the previous values. When the user sets up a function, a storage parameter for the previous value is automatically created. The user does not need to manipulate this parameter, but should be aware of its existence. This parameter is an array named Filter_Shift.

For several functions, when an intermediate calculated value is being biased, the bias itself is filtered. This prevents a bumping the output value, when the user enters a new bias.

Last Good Value Holding

Last good value holding is available for each function. This is invoked by entering a positive value in the number of intervals to hold the value (LGV_Num). Last good value is the last function to be invoked. Thus if a bad value is processed or calculated by the function it will hold the value for the number of intervals specified by the user. The user can turn off last good value holding by setting LGV_Num to 0, or alternately if the user requires the last good value held for an infinite period the user can set LGV_Num to a negative value.

When the user configures a function, storage parameters for last good value holding are automatically created. The user does not need to manipulate these parameters, but should be aware of their existence. This parameter is usually an array named LGV_Values.

Biasing

Most FCCU functions allow biasing of the final output value. The bias is supplied in user units.

Important Note: Make sure the bias is entered in **user units**. Most other input parameters are expected in toolkit units.

Engineering Unit

The FCCU functions conversion factors are required to convert **output**

Conversion

parameters from toolkit units to user units. For the main output parameters the function will calculate the output in both user and toolkit units.

The conversion factors in the FCCU function are **not** used to convert input parameters to toolkit units, as the FCCU functions expect these values to be supplied in toolkit units.

Note: Use the process functions to supply inputs in toolkit units.

Engineering Units

The following table defines the units handled by the functions:

Engineering Units			
Value	**Toolkit Units**	**User Supplied Conversion Factor or Flag**	**Comment**
Pressure	Pounds per square inch – gravity Psig		Gauge Pressure – Most functions require pressure input in these units
Pressure	Pounds per square inch – absolute Psia	Multiplier to convert to Psia	Absolute Pressure – Some functions (like RVP output in absolute pressure units.
Temperature	Degrees Fahrenheit Deg F	Flag indicating whether user value is in Deg F or Deg C	Supports only the following temperature units. 0 indicates Deg F 1 indicates Deg C
Gravity	Specific Gravity Spgr @60DegF/60DegF	Flag indicating whether user value is in specific gravity or API	0 indicates Specific Gravity 1 indicates API Gravity
Mass Flow	Thousand of pounds per hour Mlb/hr	Multiplier to convert to Mlb/hr	
Volumetric Flow	Thousand of barrels per hour Mbbl/day	Multiplier to convert to Mbbl/day	
Watson K	N/a	N/a	
Viscosity	Centistokes	No conversion factor	User **must** supply viscosity in Centistokes
Enthalpy	Thousands of BTU per pound MBTU/lb	Multiplier to convert to MBTU/lb	
Heat	Millions of BTU per hour MMBTU/hr	Multiplier to convert to MBTU/lb	
Mole Fraction	N/a	N/a	

Function Order

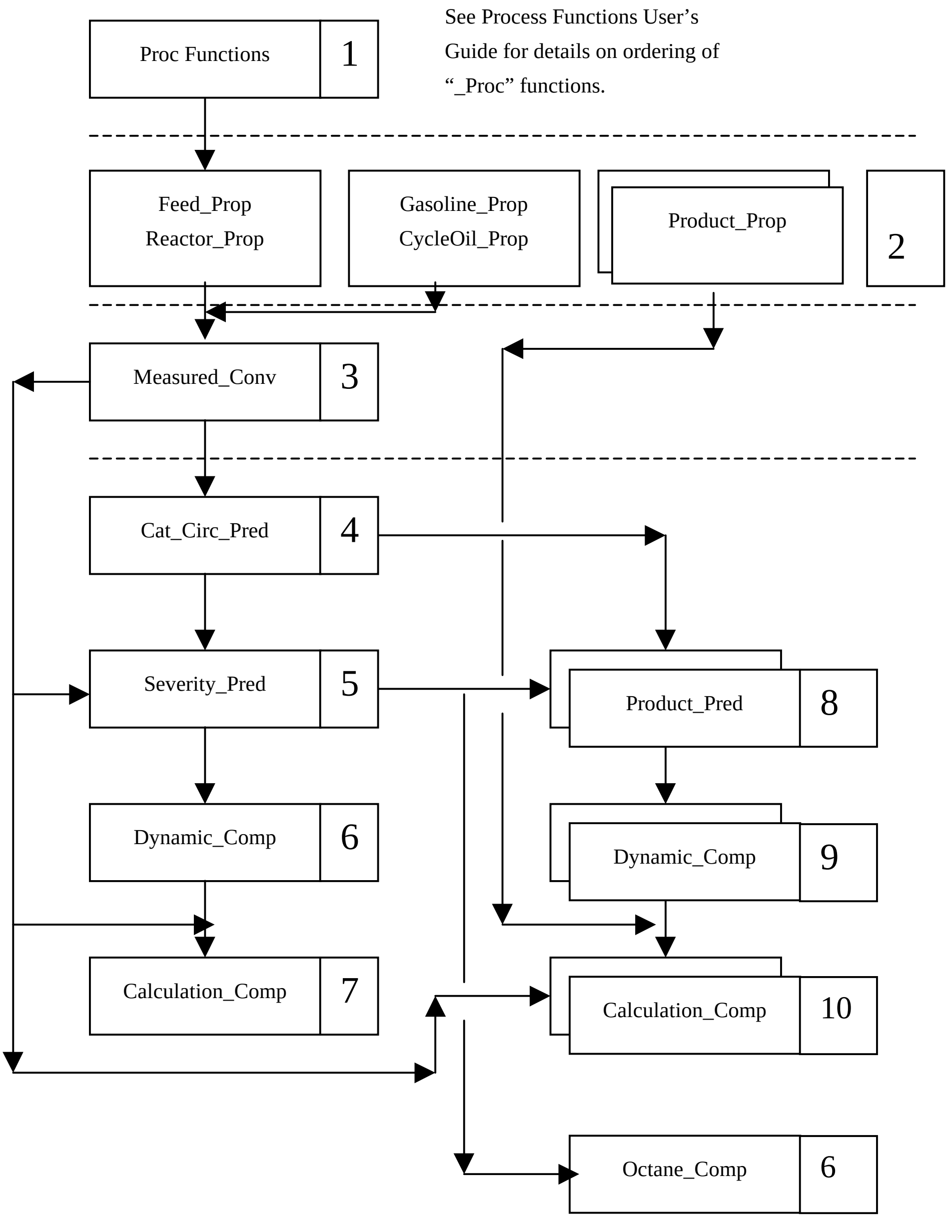
Function calling order is extremely important in the FCCU toolkit. The functions are to be ordered/called according to the number shown to the right of the function boxes. When functions are grouped together in the same row, this means that the order relative to other functions in the same row is not important.

As an example, Gasoline_Prop must be called prior to the "Proc" functions and Measured_Conv functions, but can be called before or after the Feed_Prop, Gasoline_Prop or Reactor_Prop.

As seen in the diagram, functions that have multiple parallel function boxes indicate that several instances of the function are typically required. For example, the Product_Prop and Product_Pred functions typically require one instance of each function per product.

Arrows indicate the main data flows which can be summarized as follows:

1. Use the Process functions (_Proc) to filter and convert the raw process inputs to the desired toolkit units. See Process Functions User's Guide.
2. Use Feed_Prop, Reactor_Proc, CycleOil_Prop, Gasoline_Prop and Product_Prop to calculate all properties related to measured streams.
3. Use Measured_Conv to calculate actual measured conversion and actual measured severity, based on inputs from CycleOil_Prop and Gasoline_Prop functions.
4. Use Cat_Circ_Pred function to calculate catalyst circulation rate and other catalyst calculations. Measured conversion is required as input to this function.
5. Use Severity_Pred to calculate a model based severity.
6. Dynamically compensate the severity using Dynamic_Comp. Octane_Pred can be scheduled since in this slot since it only relies on Severity_Pred.
7. Feed both the measured and dynamically compensated predicted severity into Calculation_Comp to calculate the bias and correction for the severity.
8. Use Product_Pred to predict the product make for each FCCU product.
9. Dynamically compensate the product prediction(s) using Dynamic_Comp.
10. Feed both the measured and dynamically compensated predicted product make into Calculation_Comp to calculate the bias and correction for each of the product(s).

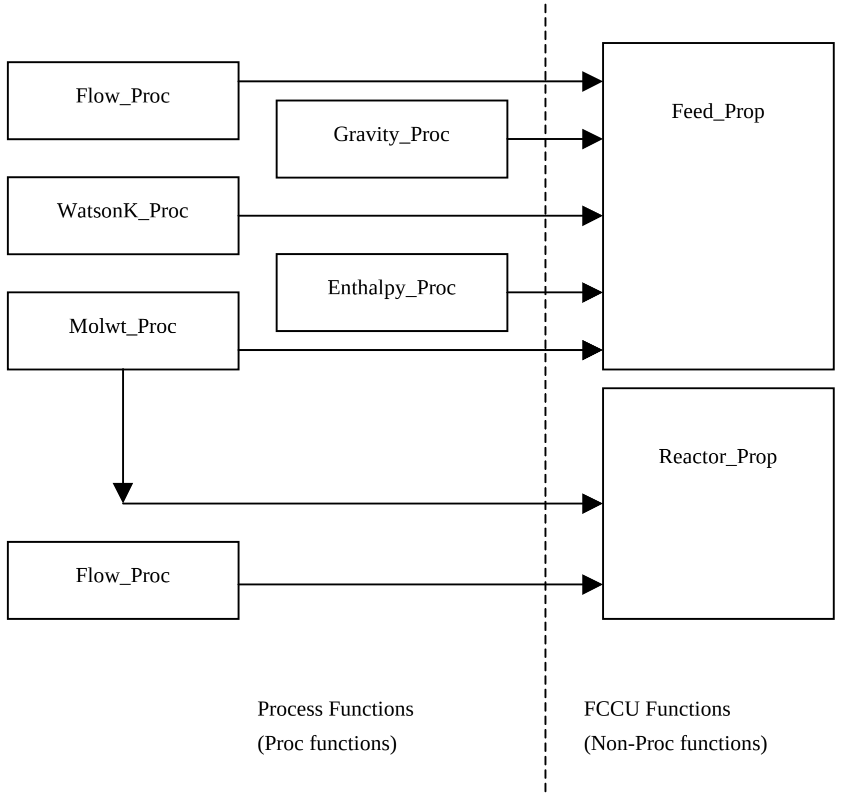

Proc Functions
1
See Process Functions User's Guide for details on ordering of "_Proc" functions.
Feed_Prop
Reactor_Prop
Gasoline_Prop
CycleOil_Prop
Product_Prop
2
Measured_Conv
3
Cat_Circ_Pred
4
Severity_Pred
5
Dynamic_Comp
6
Calculation_Comp
7
Product_Pred
8
Dynamic_Comp
9
Calculation_Comp
10
Octane_Comp
6

Using Outputs from Functions as Input to Other Functions

The functions are designed to connect together. So for example, a user entering the input parameters for the Feed_Prop function should use the output of Flow_Proc function as the input for the Feed_Prop function. This ensures that the input is conditioned properly and in the correct units.

The non-Proc functions (Proc functions are the functions named with "_Proc") are generally designed requiring inputs in toolkit units. So all the user has to do is set up one Proc function for say each temperature, the temperature will then be processed (filtered, biased and converted to toolkit units), and ready to use by the non-Proc functions. Remember to use the Proc output that is in toolkit units as input to the non-Proc function, as both user and toolkit units are usually available.

The output from a Proc function may be used as input to several other functions. For example, the output gravity (in toolkit units) from the Gravity_Proc function can be used as input to the Feed_Prop and also the Gasoline_Prop functions.



The Input Tables below list the function outputs that are recommended for use as input to another function.

Input Dependencies

Input entry may not be required for some of the input parameters. The entry requirements are determined based on the user entry of some key inputs. For example, in the Cat_Circ_Pred function, the user is not required to enter parameters relating to the calculation of enthalpy or reaction, if the user chooses to manually provide the enthalpy.

The Function Input Tables below list the key input parameter dependencies.

Error Status

All functions set an array of error statuses indicating whether error(s) have occurred in a function. Message(s) corresponding to the error(s) are raised and shown in two places:

- The function array parameter – RetMessage – that can be viewed on the Profit® Viewer Input/Output display.
- On the Profit® Viewer Status Messages display.

The errors can also be looked up in this user's guide.

Errors are positive values. A status of zero indicates that no error has occurred. A negative status is a warning. The most common warning is a last good value warning. This indicates that last good value holding is active (i.e. the output from the function is bad and the last good value is being used). There are several other warnings, which can be looked up in the Return Status Tables below.

Utility Function Error Status

Errors can also be raised by Utility functions that are used by the Fractionator functions. In this case, an error message preceded with the Utility function name is displayed. For example

Filter - The filter type specified is out of range. Filter Type must be 0 or 1, 2, 3, 4 or 5.

In the example shown above, the error was raised by the *Filter* function.

Utility errors can be looked up in the Profit Tookit Functions Reference. The list of Utility functions are shown below.

Utility Function Error Status (The status value returned for Toolkit functions is the Status Value plus 100)	
Error Status	**Utility Function**
1 - 4	Press_PC
5 - 8	Temp_PC
9 - 15	Corr_Grav
16 - 40	Flow_Corr
41 - 46	Filter_First_Ord_Lag
47 - 54	Filter_First_Ord_Lag_Deadtime
55 - 60	Filter
61 - 62	Last_Good_Value
63 - 70	WatK_MABP
71 - 78	WatK_Visc
79 - 82	Mol_WT
83 - 92	Enth_HC
93 - 97	Enth_Stm
98 - 105	Enth_Gas
106 - 111	EFV_Temp
112 - 121	EFV_To_D86

Section 2 - Feed Properties (Feed_Prop)

2.1 Overview

In This Section

This function consolidates all the properties associated with feeds. The properties are then used by other FCCU functions.

Usage of Feed_Prop

A separate instance of the Feed_Prop function should be configured for each set of FCCU feeds that the user desires to model separately.

If the user desires to model all the feeds as a single feed, then only one Feed_Prop function needs to be configured.

2.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Feed_Prop to calculate the feed properties:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the total FEED flow in both mass and volume units. FEED_Flow_Wt = Σ FEED_Flow_In[i] FEED_Flow_Vol = Σ (FEED_Flow_In[j] / (FEED_Grav_In[j] * 14.59146))
2.	Calculate the average FEED gravity. FEED_Grav = (FEED_Flow_Wt / (FEED_Flow_Vol * 14.59146f)):
3.	Calculate the average feed Watson K. Feed_WatK = Σ (Feed_WatK_In[j] * Feed_Flow_In[j]) Feed_WatK = Feed_WatK / Feed_Flow_Wt:
4.	Calculate the average feed molecular weight Feed_Molwt = Σ (Feed_Molwt_In[j] * Feed_Flow_In[j]) Feed_Molwt = Feed_Molwt / Feed_Flow_Wt:
5.	Calculate the average feed sulfur. Feed_Sulf = Σ (Feed_Sulf_In[j] * Feed_Flow_In[j]) Feed_Sulf = Feed_Sulf / Feed_Flow_Wt:
6.	Calculate the average feed con carbon. Feed_Con_Carbon = Σ (Feed_Con_Carbon_In[j] * Feed_Flow_In[j]) Feed_Con_Carbon = Feed_Con_Carbon / Feed_Flow_Wt:
7.	Calculate the average enthalpy of all feeds at inputs and average enthalpy of all feeds at riser outlet conditions Total heat of liquid Feed_Liq_Enth = Σ (Feed_Liq_Enth[j] * Feed_Flow_In[j]) Feed_Liq_Enth = Feed_Liq_Enth / FEED_Flow_Wt Total heat of vapor Feed_Vap_Enth = Σ (Feed_Vap_Enth[j] * Feed_Flow_In[j]) Feed_Vap_Enth = Feed_Vap_Enth / FEED_Flow_Wt

Input table The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of feed streams
Feed_Flow_In[ ]	Flow_Proc	Feed_Num > 0	TK units Mlb/hr	Mass flow for each independent feed stream
Feed_Grav_In[ ]	Gravity_Proc	Feed_Num > 0	TK units spgr	Gravity for each independent feed stream
Feed_WatK_In[ ]	WatsonK_ Proc	Feed_Num > 0		Watson K for each independent stream
Feed_Molwt_In[ ]	Molwt_Proc	Feed_Num > 0		Molecular weight for each independent feed stream
Feed_Liq_Enth_In[ ]	Enth_Proc	Feed_Num > 0	TK units MBTU/lb	Enthalpy of liquid for each independent feed stream
Feed_Vap_Enth_In[ ]	Enth_Proc	Feed_Num > 0	TK units MBTU/lb	Enthalpy of vapor for each independent feed stream
Feed_Sulf_In[ ]		Feed_Num > 0	Wt %	Sulfur for each independent feed stream
Feed_Con_ Carbon_In[ ]		Feed_Num > 0	Wt %	Con carbon for each independent feed stream
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.

Feed_Flow_Wt	TK units Mlb/hr	Total feed mass flow
Feed_Flow_Vol	TK units Mbbl/day	Total feed volumetric flow
Feed_Grav	TK units spgr	Average gravity of all feed streams
Feed_WatK		Average Watson K of all feed streams
Feed_Molwt		Average molecular weight of all feed streams
Feed_Sulf	Wt%	Average sulfur of all feed streams as weight percent of feed
Feed_Con_Carbon	Wt%	Average con carbon of all feed streams
Feed_Liq_Enth	TK units MBTU/lb	Heat contribution from all streams at riser inlet conditions
Feed_Vap_Enth	TK units MBTU/lb	Heat contribution from all streams at riser outlet conditions
Stream_in_Error		No longer used.

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred.
1	The feed mass flow is a bad value. Feed number ____ is the feed in error.
2	The feed gravity is a bad value. Feed number ____ is the feed in error.
3	The feed gravity is miniscule. Feed number ____ is the feed in error.
4	The total feed volumetric flow is miniscule.
5	The feed Watson K is a bad value. Feed number ____ is the feed in error.
6	The total feed mass flow is miniscule.
7	The feed molecular weight is a bad value. Feed number ____ is the feed in error.
8	The feed sulphur is a bad value. Feed number ____ is the feed in error.
9	The feed con carbon is a bad value. Feed number ____ is the feed in error.
10	The feed liquid enthalpy is a bad value. Feed number ____ is the feed in error.
11	The feed vapor enthalpy is a bad value. Feed number ____ is the feed in error.

Examples Table The following table shows example for the Feed_Prop function.

Examples Table			
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Comment**
Feed_Num	I	2	
Feed_Flow_In\1	I	100 Mlb/hr	
Feed_Flow_In\2	I	200 Mlb/hr	
Feed_Grav_In\1	I	0.7 spgr	
Feed_Grav_In\2	I	0.8 spgr	
Feed_WatK_In\1	I	12	
Feed_WatK_In\2	I	13	
Feed_Molwt_In\1	I	25	
Feed_Molwt_In\2	I	30	
Feed_Liq_Enth_In\1	I	100 MBTU/lb	
Feed_Liq_Enth_In\2	I	200 MBTU/lb	
Feed_Vap_Enth_In\1	I	100 MBTU/lb	
Feed_Vap_Enth_In\2	I	200 MBTU/lb	
Feed_Sulf_In\1	I	2 %	
Feed_Sulf_In\2	I	3 %	
Feed_Con_Carbon_In\1	I	1 %	
Feed_Con_Carbon_In\2	I	2 %	
Ret_Status	O	0	
Feed_Flow_Wt	O	300 Mlb/hr	
Feed_Flow_Vol	O	26.92377 Mbbl/day	
Feed_Grav	O	0.763636 spgr	
Feed_WatK	O	12.66667	
Feed_Molwt	O	28.33333	
Feed_Sulf	O	2.666667 %	
Feed_Con_Carbon	O	1.666667 %	
Feed_Liq_Enth	O	166.667 MBTU/lb	
Feed_Vap_Enth	O	166.667 MBTU/lb	
Stream_in_Error	O	0	

Section 3 - Reactor Properties (Reactor_Prop)

3.1 Overview

In This Section

This function consolidates all properties associated with the reactor. The properties are then used by other FCCU functions.

Usage of Reactor_Prop

Several functions require input from Reactor_Prop. They include:

- Cat_Circ_Pred
- Product_Pred
- Severity_Pred

3.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Reactor_Prop to calculate the reactor properties:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the total reactor stripping steam flow. Rx_SS_Flow = Σ (Rx_SS_Flow_In[j])
2.	Calculate the total riser steam flow Ris_Stm_Flow Σ (Ris_Stm_Flow_In[j])
3.	Calculate total heat of all riser steams at their respective inlet and outlet conditions At inlet conditions: Ris_Stm_Heat_In = Σ (Ris_Stm_Flow_In[j] * Ris_Stm_Enth_In[j]) At outlet conditions: Ris_Stm_Heat_Out += Σ (Ris_Stm_Flow_In[j] * Ris_Stm_Enth_Out[j])
4.	Calculate the riser volume in toolkit units Ris_Volume = Ris_Volume_In * Ris_Vol_Conv_Fact Calculate the riser level in toolkit units. Ris_Level =Ris_Level_In * Ris_Level_Conv_Fact

Input table

The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Rx_SS_Num				Total number of reactor stripping steam streams
Rx_SS_Flow_In[ ]	Flow_Proc	Rx_SS_Num > 0	TK units Mlb/hr	Steam flow for each independent reactor stripping steam stream
Ris_Stm_Num				Total number of riser steam streams
Ris_Stm_Flow_In[ ]	Flow_Proc	Ris_Stm_Num > 0	TK units Mlb/hr	Steam flow for each independent riser steam stream
Ris_Stm_Enth_In[ ]	Enth_Proc	Ris_Stm_Num > 0	TK units MBTU/lb	Steam enthalpy of each independent riser steam at riser inlet conditions
Ris_Stm_Enth_Out[ ]	Enth_Proc	Ris_Stm_Num > 0	TK units MBTU/lb	Steam enthalpy of each independent riser steam at riser outlet conditions
Ris_Volume_In			User units	Riser volume

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Reactor_Level_In			User units	Reactor catalyst level in user units
Ris_Vol_Conv_Fact				Riser volume conversion factor to go from user units to cubic feet (ft^3)
Reactor_Level_Conv_Fact				Reactor level conversion factor to go from user volume units to inches of H2O
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Rx_SS_Flow	TK units Mlb/hr	Total reactor stripping steam flow
Ris_Stm_Flow	TK units Mlb/hr	Total riser steam flow
Ris_Stm_Heat_In	TK units MMBTU/hr	Total heat of all riser steams at riser inlet conditions
Ris_Stm_Heat_Out	TK units MMBTU/hr	Total heat of all riser steams at riser outlet conditions
Ris_Volume	TK units ft^3	Riser volume
Reactor_Level	in H_2O	Riser level

Return Status Table Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred.
1	The reactor stripping steam flow is a bad value. Stream number ____ is the stream in error.
2	The riser steam flow is a bad value. Stream number ____ is the stream in error.
3	The riser steam enthalpy (in) is a bad value. Stream number ____ is the stream in error.
4	The riser steam enthalpy (out) is a bad value. Stream number ____ is the stream in error.
5	The riser volume is a bad value.
6	The riser volume conversion factor is a bad value.
7	The riser volume conversion factor is miniscule.
8	The reactor catalyst level is a bad value.
9	The reactor level conversion factor is a bad value.
10	The reactor level conversion factor is miniscule.

Examples Table The following table shows example for the Reactor_Prop function.

Examples Table			
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Comment**
Rx_SS_Num	I	2	
Rx_SS_Flow_In\1	I	17.525 Mlb/hr	
Rx_SS_Flow_In\2	I	0	
Ris_Stm_Num	I	5	
Ris_Stm_Flow_In\1	I	29.106 Mlb/hr	
Ris_Stm_Flow_In\2	I	0	
Ris_Stm_Flow_In\3	I	0	
Ris_Stm_Flow_In\4	I	0	
Ris_Stm_Flow_In\5	I	0	
Ris_Stm_Enth_In\1	I	1.26444	
Ris_Stm_Enth_In\2	I	0	
Ris_Stm_Enth_In\3	I	0	
Ris_Stm_Enth_In\4	I	0	
Ris_Stm_Enth_In\5	I	0	

Examples Table			
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Comment**
Ris_Stm_Enth_Out\1	I	1.5151	
Ris_Stm_Enth_Out\2	I	0	
Ris_Stm_Enth_Out\3	I	0	
Ris_Stm_Enth_Out\4	I	0	
Ris_Stm_Enth_Out\5	I	0	
Ris_Volume_In	I	1330 ft^3	
Reactor_Level_In	I	0	
Ris_Vol_Conv_Fact	I	1	
Reactor_Level_Conv_Fact	I	1	
RetNum	I	10	
Ret_Status	O	0	
Rx_SS_Flow	O	17.525 Mlb/hr	
Ris_Stm_Flow	O	29.106 Mlb/hr	
Ris_Stm_Heat_In	O	36.80279	
Ris_Stm_Heat_Out	O	44.0985	
Ris_Volume	O	1330 ft^3	
Reactor_Level	O	0	

Section 4 - Gasoline Properties (Gasoline_Prop)

4.1 Overview

In This Section

This function consolidates all the properties associated with Gasoline. The properties are then used by other FCCU functions. The calculation also calculates the cycle oil that is lost to or gained from the lighter product.

Usage of Gasoline_Prop

The Gasoline_Prop function calculates the total cycle oil flow. This is used by Measured_Conv to calculate the measured conversion and measured severity values.

Gasoline_Prop also provides a correction for the cycle oil lost to or gained from the adjacent lighter product. This correction is based on deviations from the cycle oil design conditions. Measured_Conv uses this correction value to calculate the corrected total cycle oil.

Important Note

The Gasoline_Prop outputs should **only** be used by Measured_Conv. The outputs should not be input to Calculation_Comp.

4.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Gasoline_Prop to calculate the Gasoline properties:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the total Naphtha flow in both mass and volume units. Naphtha_Flow_Wt = Σ Naphtha_Flow_In[i] Naphtha_vol[j] = (Naphtha_Flow_In[j] / (Naphtha_Grav_In[j] * 14.59146)) Naphtha_Flow_Vol = Σ Naphtha_vol[j]
2.	Calculate the average Naphtha gravity. Naphtha_Grav = (Naphtha_Flow_Wt / (Naphtha_Flow_Vol * 14.59146f)):
3.	Calculate the average RVP% point Naphtha_avg_RVP = Σ (Naphtha_RVP_In[j] * Naphtha_vol[j]) Naphtha_RVP = Naphtha_avg_RVP / Naphtha_Flow_Vol
4.	Calculate the average 90% point Naphtha_avg_90 = Σ (Naphtha_90_In[j] * Naphtha_vol[j]) Naphtha_90 = Naphtha_avg_90 / Naphtha_Flow_Vol
5.	Estimate the Naphtha product volume and mass lost to or gained from the lighter product. CycleOil_As_Heavier_Vol = (Naphtha_90_Ref – Naphtha_90) / (Naphtha_90_Slope * 100.0f) * Naphtha_Flow_Vol CO_As_Lighter_Wt= CycleOil_As_Heavier_Vol * Naphtha_Grav * 14.59146

Input table

The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Naphtha_Num				Total number of Naphtha streams
Naphtha_Flow_In[ ]	Flow_Proc	Naphtha_Num > 0	TK units Mlb/hr	Mass flow for each independent Naphtha stream
Naphtha_Grav_In[ ]	Gravity_Proc	Naphtha_Num > 0	TK units spgr	Gravity for each independent Naphtha stream
Naphtha_RVP_In[ ]		Naphtha_Num > 0	TK units psig	Reid Vapor pressure for each independent Naphtha stream
Naphtha_90_In[ ]		Naphtha_Num > 0	TK units Deg F	Naphtha ASTM D86 90% point for each independent Naphtha stream

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Naphtha_90_Ref			TK units Deg F	Naphtha ASTM D86 90% reference value used in off-line design
Naphtha_90_Slope			Deg F / % Distilled	Naphtha slope of the ASTM D86 curve at the 90% point
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Naphtha_Flow_Wt	TK units Mlb/hr	Total Naphtha mass flow
Naphtha_Flow_Vol	TK units Mbbl/day	Total Naphtha volumetric flow
Naphtha_Grav	TK units spgr	Average gravity of all Naphtha streams
Naphtha_90	TK units Deg F	Average Naphtha ASTM D86 90% point
Naphtha_RVP	TK units psig	Average Naphtha Reid Vapor Pressure (RVP)
CycleOil_As_Lighter_Wt	TK units Mlb/hr	Cycle Oil mass flow that is lost to or gained from the Naphtha stream. This value is set to zero when
CycleOil_As_Lighter_Vol	TK units Mbbl/day	Cycle Oil volumetric flow that is lost to or gained from the Naphtha stream.
Stream_in_Error		No longer used.

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred.
1	Unable to allocate space for dynamic array Naphtha_vol.
2	The Naphtha mass flow in is a bad value. Stream number____ is the stream in error.
3	The Naphtha gravity in is a bad value.
4	The Naphtha gravity in is miniscule.
5	The sum of the Naphtha volumetric flow is miniscule.
6	The Naphtha Reid vapor pressure is a bad value. Stream number____ is the stream in error.
7	The Naphtha ASTM D86 90% point is a bad value. Stream number____ is the stream in error.
8	The Naphtha ASTM D86 90% reference is a bad value.
9	The Naphtha slope is a bad value.
10	The Naphtha slope is miniscule.

Examples Table

The following table shows example for the Gasoline_Prop function.

Examples Table			
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Comment**
Naphtha_Num	I	5	
Naphtha_Flow_In\1	I	65.489998	
Naphtha_Flow_In\2	I	10	
Naphtha_Flow_In\3	I	10	
Naphtha_Flow_In\4	I	10	
Naphtha_Flow_In\5	I	10	
Naphtha_Grav_In\1	I	0.9	
Naphtha_Grav_In\2	I	1	
Naphtha_Grav_In\3	I	1	
Naphtha_Grav_In\4	I	1	
Naphtha_Grav_In\5	I	1	
Naphtha_RVP_In\1	I	4	
Naphtha_RVP_In\2	I	0	
Naphtha_RVP_In\3	I	0	

Examples Table			
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Comment**
Naphtha_RVP_In\4	I	0	
Naphtha_RVP_In\5	I	0	
Naphtha_90_In\1	I	359.600006	
Naphtha_90_In\2	I	0	
Naphtha_90_In\3	I	0	
Naphtha_90_In\4	I	0	
Naphtha_90_In\5	I	0	
Naphtha_90_Ref	I	359.6	
Naphtha_90_Slope	I	2	
Ret_Status	O	0	
Naphtha_Flow_Wt	O	105.4899979	
Naphtha_Flow_Vol	O	7.728264332	
Naphtha_Grav	O	0.935471475	
Naphtha_RVP	O	2.581140995	
Naphtha_90	O	232.0445862	
CycleOil_As_Lighter_Wt	O	-67.27909851	
CycleOil_As_Lighter_Vol	O	-4.928909779	

Section 5 - Cycle Oil Properties (CycleOil_Prop)

5.1 Overview

In This Section

This function consolidates all properties associated with the Cycle Oil product. The properties are then used by the measured conversion functions. The function also calculates the cycle oil that is lost to or gained from the heavier product.

Usage of CycleOil_Prop

The CycleOil_Prop function calculates the total cycle oil flow. This is used by Measured_Conv to calculate the measured conversion and measured severity values.

The user has the option of calculating a correction for the cycle oil lost to or gained from the adjacent heavier product. This correction is based on deviations from the cycle oil design conditions. Measured_Conv uses this correction value to calculate the corrected total cycle oil.

Important Note

The CycleOil_Prop outputs should **only** be used by as inputs to Measured_Conv. The outputs should not be input to Calculation_Comp.

5.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by CycleOil_Prop to calculate the Cycle Oil properties:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the total Cycle Oil flow in both mass (Mlb/hr) and volume (Mbbl/day) toolkit units. CycleOil_Flow_Wt = Σ CycleOil_Flow_In[i] CycleOil_vol[j] = (CycleOil_Flow_In[j] / (CycleOil_Grav_In[j] * 14.59146)) CycleOil_Flow_Vol = Σ CycleOil_vol[j]
2.	Calculate the average Cycle Oil gravity. CycleOil_Grav = (CycleOil_Flow_Wt / (CycleOil_Flow_Vol * 14.59146f))
3.	Skip steps 3 and 4 if Heavy_Corr_Flag = 0 Calculate the average 90% point CycleOil_avg_90 = Σ (CycleOil_90_In[j] * CycleOil_vol[j]) CycleOil_90 = CycleOil_avg_90 / CycleOil_Flow_Vol
4.	Estimate the Cycle Oil product volume and mass lost to or gained from the lighter product. CycleOil_As_Heavier_Vol = (CycleOil_90_Ref - CycleOil_90) / (CycleOil_90_Slope * 100.0f) * CycleOil_Flow_Vol CycleOil_As_Heavier_Wt = CycleOil_As_Heavier_Vol * CycleOil_Grav * 14.59146

Input table

The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
CycleOil_Num				Total number of Cycle Oil streams
CycleOil_Flow_In[ ]	Flow_Proc	CycleOil_Num > 0	TK units Mlb/hr	Mass flow for each independent Cycle Oil stream
CycleOil_Grav_In[ ]	Gravity_Proc	CycleOil_Num > 0	TK units spgr	Gravity for each independent Cycle Oil stream
Heavy_Corr_Flag				Flag indicating whether to correct for cycle oil lost to or gained from the heavier product (0=No correction, 1=Do correction)
CycleOil_90_In[ ]		CycleOil_Num > 0	TK units Deg F	ASTM D86 90% point for each

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
		Heavy_Corr_Flag = 1		independent Cycle Oil stream
CycleOil_90_Ref		Heavy_Corr_Flag = 1	TK units Deg F	Cycle Oil ASTM D86 90% reference value used in off-line design
CycleOil_90_Slope		Heavy_Corr_Flag = 1	Deg F / % Distilled	Cycle Oil slope of the ASTM D86 curve at the 90% point
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
CycleOil_Flow_Wt	TK units Mlb/hr	Total Cycle Oil mass flow
CycleOil_Flow_Vol	TK units Mbbl/day	Total Cycle Oil volumetric flow
CycleOil_Grav	TK units spgr	Average gravity of all Cycle Oil streams
CycleOil_90	TK units Deg F	Average Cycle Oil ASTM D86 90% point
CycleOil_As_Heavier_Wt	TK units Mlb/hr	Cycle Oil mass flow that is lost to or gained from the heavier stream. This value is set to zero when Heavy_Corr_Flag is set for no correction.
CycleOil_As_Heavier_Vol	TK units Mbbl/day	Cycle Oil volumetric flow that is lost to or gained from the heavier stream. This value is set to zero when Heavy_Corr_Flag is set for no correction.
Stream_in_Error		No longer used.

Return Status Table Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	Unable to allocate space for dyanamic array CycleOil_vol.
2	The Cycle Oil mass flow in is a bad value. Feed number ____ is the feed in error.
3	The Cycle Oil gravity is a bad value. Feed number ____ is the feed in error.
4	The Cycle Oil gravity is miniscule. Feed number ____ is the feed in error.
5	The Cycle Oil volumetric flow is miniscule.
6	The flag indicating whether to do correction for heavier products must be equal to 0 or 1.
7	The ASTM D86 90% point is a bad value. Feed number ____ is the feed in error.
8	The ASTM D86 90% reference is a bad value.
9	The Cycle Oil slope is a bad value.
10	The Cycle Oil slope is miniscule.

Examples Table The following table shows example for the CycleOil_Prop function.

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
CycleOil_Num	I	5	5	
CycleOil_Flow_In\1	I	71.07	71.07	
CycleOil_Flow_In\2	I	55.450001	55.450001	
CycleOil_Flow_In\3	I	0	0	
CycleOil_Flow_In\4	I	0	0	
CycleOil_Flow_In\5	I	0	0	
CycleOil_Grav_In\1	I	0.9796	0.9796	
CycleOil_Grav_In\2	I	1.1073	1.1073	
CycleOil_Grav_In\3	I	1	1	
CycleOil_Grav_In\4	I	1	1	
CycleOil_Grav_In\5	I	1	1	

Examples Table				
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Heavy_Corr_Flag	I	0	1	Example 2 has heavy correction
CycleOil_90_In\1	I	359.600006	359.600006	
CycleOil_90_In\2	I	359.600006	359.600006	
CycleOil_90_In\3	I	0	0	
CycleOil_90_In\4	I	0	0	
CycleOil_90_In\5	I	0	0	
CycleOil_90_Ref	I	359.600006	359.600006	
CycleOil_90_Slope	I	2	2	
RetNum	O	10	10	
Ret_Status	O	0	0	
CycleOil_Flow_Wt	O	126.5200043	126.5200043	
CycleOil_Flow_Vol	O	8.404010773	8.404010773	
CycleOil_Grav	O	1.031748533	1.031748533	
CycleOil_90	O	--------	359.6000061	Example 2 has heavy correction
CycleOil_As_Heavier_Wt	O	0	0	
CycleOil_As_Heavier_Vol	O	0	0	
Stream_in_Error	O	0	0	

Section 6 - Product Properties (Product_Prop)

6.1 Overview

In This Section

This function calculates the corrected product flows based on deviations from design conditions.

Usage of Product_Prop

A separate instance of Product_Prop needs to be configured for each product.

The corrected measured product flow calculated by Product_Prop is input to Calculation_Comp.

6.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Product_Prop to calculate the compensated product flows:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the total product flow in toolkit mass (Mlb/hr) or volume (Mbbl/day) units. Tot_Prod_Flow_Mass = Σ (Prod_Flow[j]) Tot_Prod_Flow_Vol += Prod_Flow[j] / (Prod_Grav_Molwt[j] * 14.59146)
2.	Calculate the average product specification Prod_Spec = Σ (Prod_Spec_In[j] * Prod_Flow[j]) / Tot_Prod_Flow_Mass
3.	Estimate the product lost to or gained from the adjacent product in both mass (Mlb/hr) and volume units (Mbbl/day) (Product_Xchg_Mass = Σ (Prod_Spec_In[j] – Prod_Spec_Ref) * Numer_Coeff / (Denom_Coeff1 * Denom_Coeff2) * Prod_Flow[j]) Product_Xchg_Vol = Σ (Product_Xchg_Mass / (Prod_Grav_Molwt[j] * 14.59146))
4.	Calculate corrected flow for this product in both mass (Mlb/hr) and volume (Mbbl/day) units: Corr_Prod_Flow_Mass = Tot_Prod_Flow_Mass + Adjacent_Corr_Flow_Mass - Product_Xchg_Mass Corr_Prod_Flow_Vol = Tot_Prod_Flow_Vol + Adjacent_Corr_Flow_Vol - Product_Xchg_Vol

5.	If user desires output in mass flow units then
	Corr_Prod_Flow_User = Corr_Prod_Flow_Mass / Conv_Factor
	Product_Xchg_User = Product_Xchg_Mass / Conv_Factor
	Tot_Prod_Flow_User = Tot_Prod_Flow_Mass / Conv_Factor
	If user desires output in volumetric flow units then
	Corr_Prod_Flow_User = Corr_Prod_Flow_Vol / Conv_Factor
	Product_Xchg_User = Product_Xchg_Vol / Conv_Factor
	Tot_Prod_Flow_User = Tot_Prod_Flow_Vol / Conv_Factor

Input table The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Prod_Num				Total number of Product streams for a single product.
Prod_Flow[ ]	Flow_Proc	Prod_Num > 0	TK units Mlb/hr	Product flow for each independent stream
Prod_Spec_In[ ]		Prod_Num > 0		Product specification measured for each independent stream
Prod_Grav_Molwt[ ]	Gravity_Proc or Molwt_Proc	Prod_Num > 0	TK units spgr orUnit-less if molecular weight	Product gravity or molecular weight for each independent stream.If User_V_or_M is set to Mass, then this value should be set equal to 1. Specific gravity entered for liquid volume flow streams.Molecular weight entered for gas volume flow streams.
Prod_Spec_Ref			Same units as Prod_Spec_In	Product specification reference value used in off-line design
Numer_Coeff				Numerator coefficient for correction. Set equal to 1.0 for zero effect on correction.
Denom_Coeff1				First denominator coefficient for correction. Set equal to 1.0 for zero effect on correction.
Denom_Coeff2				Second denominator coefficient for correction. Set equal to 1.0 for zero effect on correction.
Adjacent_Corr_ Flow_Mass	Flow_Proc		TK units Mlb/hr	Flow correction from an adjacent product. **Note**: To obtain this value in the correct units feed the adjacent corrected flow into a Flow_Proc function and use the Mass_Flow_TK_Units output as input to this parameter.
Adjacent_Corr_ Flow_Vol	Flow_Proc		TK units Mbbl/day	Flow correction from an adjacent product. **Note**: To obtain this value in the correct units feed the adjacent corrected flow into a Flow_Proc function and use the Vol_Flow_TK_Units output as input to this parameter.
User_V_or_M				Certain flow values are output in user units. This Flag indicates whether these outputs should be volumetric or mass flow. (0 = volumetric flow, 1 = mass flow)

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Conv_Factor				Multiplier to go from user flow units to toolkit units (Mlb/hr). Used to output internal liquid and vapor values in user units
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Tot_Prod_Flow_TK	TK units Mlb/hr	Total product flow
Tot_Prod_Flow_User	User units	Total product flow
Corr_Prod_Flow_TK	TK units Mlb/hr	Total compensated (corrected) product flow
Corr_Prod_Flow_User	User units	Total compensated (corrected) product flow
Prod_Spec		Average specification of all product streams. The average is based on mass units.
Product_Xchg_TK	TK units Mlb/hr	Product flow that is lost to or gained from the adjacent product
Product_Xchg_User	User units	Product flow that is lost to or gained from the adjacent product
Stream_in_Error		No longer used.

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred.
1	The product flow is a bad value. Product number____ is the product in error.
2	The product gravity or molecular weight is a bad value. Product number____ is the product in error.
3	The product gravity or molecular weight is less than or equal to zero. Product number____ is the product in error.
4	The product specification is a bad value. Product number____ is the product in error.
5	The total product flow is miniscule.
6	The product specification reference is a bad value.
7	The numerator coefficient is a bad value.
8	The first denominator coefficient is a bad value.
9	The first denominator coefficient is miniscule.
10	The second denominator coefficient is a bad value.
11	The second denominator coefficient is miniscule.
12	The adjacent flow correction in mass units is a bad value.
13	The adjacent flow correction in volumetric units is a bad value.
14	The flag indicating whether to output user values in volumetric or mass flow units must be equal to either zero or one.
15	The conversion factor is a bad value.
16	The conversion factor is a miniscule value.

Examples Table

The following table shows example for the Product_Prop function.

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Prod_Num	I	5		
Prod_Flow\1	I	41.07 Mlb/hr		
Prod_Flow\2	I	25.450001 Mlb/hr		
Prod_Flow\3	I	30 Mlb/hr		
Prod_Flow\4	I	15 Mlb/hr		
Prod_Flow\5	I	15 Mlb/hr		

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Prod_Spec_In\1	I	359.600006		
Prod_Spec_In\2	I	359.600006		
Prod_Spec_In\3	I	359.600006		
Prod_Spec_In\4	I	359.600006		
Prod_Spec_In\5	I	359.600006		
Prod_Grav_Molwt\1	I	1		
Prod_Grav_Molwt\2	I	1		
Prod_Grav_Molwt\3	I	1		
Prod_Grav_Molwt\4	I	1		
Prod_Grav_Molwt\5	I	1		
Prod_Spec_Ref	I	359.600006		
Numer_Coeff	I	1		
Denom_Coeff1	I	1		
Denom_Coeff2	I	1		
Adjacent_Corr_Flow_Mass	I	0		
Adjacent_Corr_Flow_Vol	I	0		
User_V_or_M	I	1		
Conv_Factor	I	1		
RetNum	I	10		
Ret_Status	O	0		
Tot_Prod_Flow_TK	O	126.5200043		
Tot_Prod_Flow_User	O	126.5200043		
Corr_Prod_Flow_TK	O	126.5200043		
Corr_Prod_Flow_User	O	126.5200043		
Prod_Spec	O	359.6000061		
Product_Xchg_TK	O	0		
Product_Xchg_User	O	0		
Stream_in_Error	O	0		

Section 7 - Measured Conversion (Measured_Conv)

7.1 Overview

In This Section

This function calculates the measured conversion and measured severity based on the process measurements. Prior to outputting the conversion and severity the values are filtered. Last good value holding is also available.

Usage of Measured_Conv

The Measured_Conv outputs both measured conversion and measured severity. The measured severity is input to Calculation_Comp, which uses the measured severity to bias, and corrects the predicted severity based on differences with the measured severity.

Since measured conversion is defined as the percent of feed converted to product excluding the Cycle Oil product, the user must input the total feed flow rate and the total cycle oil product into Measured_Conv.

Measured_Conv also takes inputs from the Gasoline_Prop and CycleOil_Prop functions. These functions provide corrections for cycle oil lost to or gained from adjacent products due to deviations from design conditions.

7.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Measured_Conv to calculate the measured conversion and measured severity:

Calculation Algorithm	
Step #	**Description**
1.	Calculate total feed flows in mass and volumetric units: Feed_wt = Σ Feed_Flow[j] Feed_vol = Σ Feed_Flow_Vol [j]
2.	Estimate the measured conversion in volume and weight percent: Meas_Conv_Vol = 100.0 * (Feed_vol - CycleOil_Flow_Vol - CycleOil_As_Lighter_Vol - CycleOil_As_Heavier_Vol) / Feed_vol Meas_Conv_Wt = 100.0 * (Feed_Wt - CycleOil_Flow_Wt - CycleOil_As_Lighter_Wt - CycleOil_As_Heavier_Wt) / Feed_wt
3.	Filter the measured conversion in wt % based on the supplied Filter_Type.
4.	Calculate the measured severity based on volume. Meas_Severity_Vol = Meas_Conv_Vol / (100.0 - Meas_Conv_Vol)
5.	Filter the measured conversion in vol % based on the supplied Filter_Type.
6.	Calculate the measured severity based on weight. Meas_Severity_Wt = Meas_Conv_Wt / (100.0 - Meas_Conv_Wt)
7.	Call Last_Good_Value utility function to provide last good value processing.

Input table

The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of independent feed streams used in off-line design
Feed_Flow[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mlb/hr	Feed mass flow for each independent feed
Feed_Flow_Vol[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mbbl/day	Feed volumetric flow for each independent feed

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
CycleOil_Flow_Wt	Flow_Proc or CycOil_Prop		TK units Mlb/hr	Total Cycle Oil product mass flow
CycleOil_Flow_Vol	Flow_Proc or CycOil_Prop		TK units Mbbl/day	Total Cycle Oil product volumetric flow
CycleOil_As_Lighter_Wt	Gasoline_Prop		TK units Mlb/hr	Cycle Oil as lighter product mass flow
CycleOil_As_Lighter_Vol	Gasoline_Prop		TK units Mbbl/day	Cycle Oil as lighter product volumetric flow
CycleOil_As_Heavier_Wt	CycleOil_Prop		TK units Mlb/hr	Cycle Oil as heavier product mass flow
CycleOil_As_Heavier_Vol	CycleOil_Prop		TK units Mbbl/day	Cycle Oil as heavier product volumetric flow
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Wt[ ]				Previous filtered value storage
Filter_Shift_Vol[ ]				Previous filtered value storage
Etime			Minutes	Application execution time
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
				will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..4]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Meas_Conv_Wt	Wt%	Calculated measured conversion.
Meas_Conv_Vol	Vol%	Calculated measured conversion.
Meas_Severity_Wt		Calculated measured severity based on measured conversion in weight units
Meas_Severity_Vol		Calculated measured severity based on measured conversion in volume units
Stream_in_Error		No longer used.
Add_Bias_Filt_Shift		Previous filtered additive bias storage
Filter_Shift_Wt[ ]		Previous filtered value storage
Filter_Shift_Vol[ ]		Previous filtered value storage
Filter_Shift_Wt[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..4]		Last good value storage array for key output variables

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The feed mass flow is a bad value. Feed number ____ is the feed in error.
2	The feed volumetric flow is a bad value. Feed number ____ is the feed in error.
3	The Cycle Oil volumetric flow is a bad value
4	The Cycle Oil product volume lost to or gained from the lighter product is a bad value.
5	The Cycle Oil product volume lost to or gained from the heavier product is a bad value.
6	The total feed volumetric flow is miniscule.
7	The calculated measured conversion based on volume is less than or equal to zero.
8	The calculated measured conversion based on volume is greater than or equal to 100 percent.
9	The Cycle Oil mass flow is a bad value
10	The Cycle Oil product mass lost to or gained from the lighter product is a bad value.
11	The Cycle Oil product mass lost to or gained from the heavier product is a bad value.
12	The total feed mass flow is miniscule.
13	The calculated measured conversion based on mass is less than or equal to zero.
14	The calculated measured conversion based on mass is greater than or equal to 100 percent.
15	The measured conversion based on volume = 100 percent, can't calculate severity as this will result in divide by 0
16	The measured conversion based on weight = 100 percent, can't calculate severity as this will result in divide by 0
17	Errors filtering measured conversion (vol%). See associated message from Filter function
18	Errors filtering measured conversion (wt%). See associated message from Filter function
141 – 146	An error occurred in the Filter_First_Ord_Lag function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 – 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 – 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
Negative errors	Last good value holding is active. The error that is causing the bad value can be determined by looking up the positive of the error value

Examples Table The following table shows example for the Measured_Conv function. The example assumes no filtering and no last good value holding.

Examples Table			
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Comment**
Feed_Num	I	1	
Feed_Flow\1	I	448.9499 Mlb/hr	
Feed_Flow_Vol\1	I	33.2115 Mbbl/day	
CycleOil_Flow_Wt	I	126.517 Mlb/hr	
CycleOil_Flow_Vol	I	8.399 Mbbl/day	
CycleOil_As_Lighter_Wt	I	0 Mlb/hr	
CycleOil_As_Lighter_Vol	I	0 Mbbl/day	
CycleOil_As_Heavier_Wt	I	0 Mlb/hr	
CycleOil_As_Heavier_Vol	I	0 Mbbl/day	
Ret_Status	O	0	
Meas_Conv_Wt	O	71.819 Wt %	
Meas_Conv_Vol	O	74.7105 Vol %	
Meas_Severity_Wt	O	2.54854	
Meas_Severity_Vol	O	2.954217911	
Stream_in_Error	O	0	

Section 8 - Catalyst Circulation Prediction(Cat_Circ_Pred)

8.1 Overview

In This Section

This function calculates catalyst-based values necessary in other functions. The main outputs calculated are the catalyst circulation rate, the cat to oil ratio and the reactor stripping steam. Prior to outputting the catalyst circulation rate the value is filtered, biased and converted to user units. Last good value holding is also available.

Usage of Cat_Circ_Pred

Two functions utilize the outputs from the Cat_Circ_Pred function.

Severity_Pred requires the catalyst to oil ratio as input, and Product_Pred uses the reactor stripping steam to catalyst ratio.

8.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Cat_Circ_Pred to calculate the catalyst circulation rate:

Calculation Algorithm	
Step #	**Description**
1.	Filter the additive bias. Use the first order lag filter. Filter the multiplicative bias. Use the first order lag filter.
2.	Calculate total moles of the feed: Tot_feed_moles = Σ (Feed_Flow[i] / Feed_Molwt[i])
3.	Calculate total feed flow as sum of all the flows: Tot_feed_flow = Σ Feed_Flow[i]
4.	Calculate heat of feed into riser: Feed_heat_in = Σ (Feed_Liq_Enth[i] * Feed_Flow[i])
5.	Calculate heat of feed out of riser: Feed_heat_out = Σ (Feed_Vap_Enth[i] * Feed_Flow[i])
6.	Calculate heat into system by adding the feed in heat and the riser steam in heat: Heat_In = Feed_heat_in + Riser_Steam_Heat_In
7.	Calculate heat out of system by adding the feed out heat and the riser steam out heat: Heat_Out = Feed_heat_out + Riser_Steam_Heat_Out
8.	Determine whether to calculate enthalpy of reaction or to use user provided value. User provided value: If Enth_Rxn_Type = 0 Loc_Enth_Rxn = Enth_Rxn Calculate the enthalpy of reaction: If Enth_Rxn_Type = 1 Loc_Enth_Rxn = ƒ(Heat_Rxn_Coef, Heat_Rxn_Riser_Coef, Heat_Rxn_Regen_Coef, Heat_Rxn_Conv_Coef, Riser_Temp, Riser_Temp_Ref, Regen_Temp, Regen_Temp_Ref, Meas_Conv_Wt, Meas_Conv_Wt_Ref)
9.	Calculate heat of reaction from enthalpy of reaction: Heat_Rxn = ƒ(Loc_Enth_Rxn, Meas_Conv_Wt, Tot_feed_moles)
10.	Calculate the specific heat of the catalyst: Cat_Specific_Heat = ƒ(Regen_Temp, Riser_Temp)
11.	Calculate the catalyst circulation rate in toolkit units using a heat balance around the riser: Value_TK_Units = ƒ (Heat_Out, Heat_In, Heat_Rxn, Heat_Loss, Cat_Specific_Heat, Regen_Temp, Riser_Temp)

Calculation Algorithm	
Step #	**Description**
12.	Filter the catalyst circulation rate based on the supplied Filter_Type.
13.	Convert the filtered catalyst circulation rate in toolkit units Mlb/hr to mass flow in user units by dividing by the conversion factor. Value_User_Units = Value_TK_Units / Conv_Fact
14.	Bias the filtered catalyst circulation rate user value by using a multiplicative and additive adding filtered bias. Value_User_Units = Value_User_Units * Mult_Bias + Add_Bias
15.	Convert the biased catalyst circulation rate user value back to toolkit units. Value_TK_Units = Value_User_Units * Conv_Fact
16.	Calculate the catalyst to oil ratio: Cat_To_Oil = Value_TK_Units / Tot_feed_flow
17.	Calculate the reactor stripping steam to catalyst ratio: Rx_SS_To_Cat = Reactor_Strip_Steam / Value_TK_Units
18.	Call Last_Good_Value utility function to provide last good value processing.

Input table

This function requires inputs from the following functions:

- Flow_Proc
- Molwt_Proc
- Enth_Proc
- Temperature_Proc

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of independent feed streams used in off-line design
Feed_Flow[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mlb/hr	Feed mass flow for each independent feed
Feed_Molwt[ ]	Molwt_Proc or Feed_Prop	Feed_Num > 0		Feed molecular weight for each independent feed
Feed_Liq_Enth[ ]	Enth_Proc or Feed_Prop	Feed_Num > 0	TK units MBTU/lb	Feed enthalpy at riser inlet conditions for each independent feed
Feed_Vap_Enth[	Enth_Proc or	Feed_Num > 0	TK units	Feed enthalpy at riser output conditions

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
]	Feed_Prop		MBTU/lb	for each independent feed
Enth_Rxn_Type				Flag indicating whether enthalpy of reaction is to be supplied by user or calculated (0=User supplied, 1=Calculated)
Enth_Rxn		Enth_Rxn_Type = 0	TK units MBTU/lb	User supplied enthalpy of reaction
Rx_SS_Flow	Flow_Proc or Reactor_Prop		TK units Mlb/hr	Total stripping steam flow to reactor
Riser_Steam_Flow	Flow_Proc or Reactor_Prop		TK units Mlb/hr	Total steam flow to riser
Riser_Steam_Heat_In	Enth_Proc or Reactor_Prop		TK units MMBTU/hr	Riser steam heat at riser inlet conditions
Riser_Steam_Heat_Out	Enth_Proc or Reactor_Prop		TK units MMBTU/hr	Riser steam heat at riser outlet conditions
Riser_Temp	Temperature_Proc		TK units Deg F	Riser outlet temperature
Riser_Temp_Ref		Enth_Rxn_Type = 1	TK units Deg F	Riser outlet temperature reference
Regen_Temp	Temperature_Proc		TK units Deg F	Regenerator bed temperature
Regen_Temp_Ref		Enth_Rxn_Type = 1	TK units Deg F	Regenerator bed temperature reference
Meas_Conv_Wt	Measured_Conv		Wt %	Measured conversion
Meas_Conv_Wt_Ref		Enth_Rxn_Type = 1	Wt %	Measured conversion reference
Heat_Rxn_Coef	Offline design	Enth_Rxn_Type = 1		Heat of reaction coefficient
Heat_Rxn_Riser_Coef	Offline design	Enth_Rxn_Type = 1		Heat of reaction riser coefficient
Heat_Rxn_Regen_Coef	Offline design	Enth_Rxn_Type = 1		Heat of reaction regenerator coefficient

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Heat_Rxn_Conv_Coef	Offline design	Enth_Rxn_Type = 1		Heat of reaction conversion coefficient
Heat_Loss			TK units MBTU/hr	Heat loss from the riser
Conv_Fact				Conversion factor to go from catalyst circulation in user mass flow units to TK mass flow units
Mult_Bias				Multiplicative bias for catalyst circulation
Mult_Bias_Filt_Time				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual
Mult_Bias_Filt_Shift				Previous filtered catalyst circulation multiplicative bias storage
Add_Bias			User units	Additive bias for catalyst circulation
Add_Bias_Filt_Time			Minutes	Filter time for the additive bias
Add_Bias_Filt_Shift				Previous filtered catalyst circulation additive bias storage
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual
Filter_Shift[ ]				Previous filtered value storage
Etime			Minutes	Application execution time

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..4]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Value_TK_Units	TK units Mlb/hr	Calculated catalyst circulation mass flow rate
Value_User_Units	User units	Calculated catalyst circulation mass flow rate. **Note:** This value **must** be in mass units.
Cat_To_Oil		Calculated catalyst to oil ratio
Rx_SS_To_Cat		Reactor stripping steam to catalyst ratio

Cat_Specific_Heat	BTU/lb/Deg F	Specific heat of catalyst
Heat_In	TK units MMBTU/hr	Total heat into the riser
Heat_Out	TK units MMBTU/hr	Total heat out of the riser
Calcd_Enth_Rxn	MBTU/lb-mole	Calculated enthalpy of reaction. Only calculated if Enth_Rxn_Type is set to 1.
Heat_Rxn	TK units MMBTU/hr	Heat of reaction
Stream_in_Error		No longer used.
Mult_Bias_Filt_Shift		Previous filtered multiplicative bias storage
Add_Bias_Filt_Shift		Previous filtered additive bias storage
Filter_Shift[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..4]		Last good value storage array for key output variables

Return Status Table Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The additive bias is a bad value.
2	The multiplicative bias is a bad value.
3	The multiplicative bias is less than or equal to zero
4	The feed flow is a bad value. Feed number ____ is the feed in error.
5	The feed molecular weight is a bad value. Feed number ____ is the feed in error.
6	The feed molecular weight is less than or equal to zero. Feed number ____ is the feed in error.
7	The feed liquid enthalpy is a bad value. Feed number ____ is the feed in error.
8	The feed vapor enthalpy is a bad value. Feed number ____ is the feed in error.
9	The riser steam heat in is a bad value
10	The riser steam heat out is a bad value
11	The enthalpy of reaction flag is a bad value
12	The enthalpy of reaction flag must equal to zero or one
13	The user provided enthalpy of reaction is a bad value.
14	The heat of reaction coefficient is a bad value.
15	The riser outlet temperature is a bad value.
16	The riser outlet temperature reference is a bad value.

Return Status	
Status value	**Description**
17	The heat of reaction riser coefficient is a bad value.
18	The regenerator bed temperature is a bad value.
19	The regenerator bed temperature reference is a bad value.
20	The heat of reaction regenerator coefficient is a bad value.
21	The measured conversion is a bad value.
22	The measured conversion reference is a bad value.
23	The heat of reaction conversion coefficient is a bad value.
24	For future use.
25	The riser outlet temperature is a bad value.
26	For future use.
27	The heat loss from the riser is a bad value.
28	The calculated specific heat of the catalyst is equal to zero. Unable to calculate catalyst circulation.
29	The regenerator temperature minus the riser temperature is equal to zero. Unable to calculate catalyst circulation.
30	The conversion factor is a bad value.
31	The conversion factor is a miniscule value.
32	The total feed flow is a miniscule value
33	The total stripping stream flow to the reactor is a bad value.
34	Calculated catalyst circulation rate is a miniscule value. Can't calculate reactor stripping stream to catalyst ration ration as this will result in divide by 0.
35	Errors filtering the additive bias. See associated message from Filter function
36	Errors filtering the multiplicative bias. See associated message from Filter function
37	Errors filtering the catalyst circulation rate. See associated message from Filter function
141 - 146	An error occurred in the Filter_First_Ord_Lag function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 - 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 - 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
Negative errors	Last good value holding is active. The error that is causing the bad value can be determined by looking up the positive of the error value

Examples Table The following table shows example for the Cat_Circ_Pred function. The example assumes no filtering and no last good value holding.

Examples Table			
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Comment**
Feed_Num	I	1	
Feed_Flow[ ]	I	448.95 Mlb/hr	
Feed_Molwt[ ]	I	485.02	
Feed_Liq_Enth[ ]	I	0.2662 MBTU/lb	
Feed_Vap_Enth[ ]	I	0.7574 MBTU/lb	
Enth_Rxn_Type	I	1	Example 1 – Calculated enthalpy of reaction.
Enth_Rxn	I		
Reactor_SS_Flow	I	0 Mlb/hr	
Riser_Steam_Flow	I	29.106 Mlb/hr	
Riser_Steam_Heat_In	I	36.8029 MMBTU/hr	
Riser_Steam_Heat_Out	I	44.1 MMBTU/hr	
Riser_Temp	I	966.20 Deg F	
Riser_Temp_Ref	I	966.00 Deg F	
Regen_Temp	I	1335.2 Deg F	
Regen_Temp_Ref	I	1335.0 Deg F	
Meas_Conv_Wt	I	71.820 wt %	
Meas_Conv_Wt_Ref	I	71.800 wt %	
Heat_Rxn_Coef	I	89000.	
Heat_Rxn_Riser_Coef	I	0	
Heat_Rxn_Regen_Coef	I	0	
Heat_Rxn_Conv_Coef	I	2000.0	
Heat_Loss	I	4.0 MMBTU/hr	
Conv_Fact	I	1.0000	
Mult_Bias	I	1	
Mult_Bias_Filt_Time	I	0	
Add_Bias	I	0	
Add_Bias_Filt_Time	I	0	
Filter_Type	I	0	
Ret_Status		0	

Examples Table			
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Comment**
Value_TK_Units	O	2849.7 Mlb/hr	
Value_User_Units	O	2849.7 Mlb/hr	
Cat_To_Oil	O	6.3475	
Rx_SS_To_Cat	O	0	
Cat_Specific_Heat	O	0.2768 BTU/lb/F	
Heat_In	O	156.30 MMBTU/hr	
Heat_Out	O	384.12 MMBTU/hr	
Calcd_Enth_Rxn	O	89.040 MBTU/lb	
Heat_Rxn	O	59.193 MMBTU/hr	
Stream_in_Error	O	0	

Section 9 - Severity Prediction (Severity_Pred)

9.1 Overview

In This Section

This function calculates the severity and conversion based on the FCCU model. Prior to outputting the severity, the value is filtered, and corrected using the catalyst correction factor and then biased. The catalyst correction factor and bias are provided by the Calculation Compensation function. Last good value holding is also available.

Usage of Severity_Pred

The Severity_Pred function is a key function in the FCCU toolkit as it predicts both severity and measured conversion.

The severity calculated by the function is dynamically compensated (using the Dynamic_Comp function) after which the dynamically compensated severity is input to Calculation_Comp, which biases and corrects the severity based on differences with the measured severity.

Severity_Pred requires input from the catalyst circulate rate function – Cat_Circ_Rate – to calculate the predicted severity.

9.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Severity_Pred to calculate the severity:

Calculation Algorithm	
Step #	**Description**
1.	Filter the additive bias. Use the first order lag filter.
2.	Calculate total feed flow as sum of all the flows: Tot_feed_flow = Σ Feed_Flow[i]
3.	Convert riser pressures to absolute units and then to atmospheres: (Riser_Press + Local_Pressure) / 14.696
4.	Scale variables in accordance to off-line design calculations: scale riser temperature Riser_temp_s = ƒ(Riser_Temp, Riser_Temp_Ref) scale regenerator temperature Regen_temp_s = ƒ(Regen_Temp, Regen_Temp_Ref) Scale catalyst surface area Cat_surf_area_s = ƒ(Cat_Surf_Area) For j = 1 to Feed_Num Convert all gravity values to API units prior to scaling gravity. Feed_grav_s[j] = ƒ(Feed_Grav[j], Feed_Grav_Ref[j]) Feed_watK_s[j] = ƒ(Feed_WatK[j], Feed_WatK_Ref[j]) Calculate the severity factors. Severity_K2_Factor = ƒ(Severity, Severity_K2_Coef) Severity_K3_Factor = ƒ(Severity, Severity_K3_Coef) Calculate the riser reaction factors. Riser Pressure Factor Riser_Press_Factor = ƒ(Riser_press_atm, Riser_Press_Coef) Riser Temperature Factor Riser_Temp_Factor = ƒ(Riser_temp_s, Riser_Temp_Coef) Reactor stripping steam to catalyst factor Rx_SS_To_Cat_Factor = ƒ(Rx_SS_To_Cat, Rx_SS_To_Cat_Coef) Riser Steam Flow Factor Riser_Steam_Flow_Factor = ƒ(Riser_Steam_Flow, Riser_Steam_Flow_Coef)

Calculation Algorithm	
Step #	**Description**
	Calculate the regenerator reaction factors. Regen_Temp_Factor = ƒ(Regen_temp_s, Regen_Temp_Coef)
5.	Calculate the feed characterization factors for each independent feed and sum them together For j = 1 to Feed_Num Feed_Grav_Factor[j] = ƒ(Feed_Grav_s[j], Feed_Grav_Coef[j]) Feed_WatK_Factor[j] = ƒ(Feed_WatK_s[j], Feed_WatK_Coef[j]) Feed_Sulf_Factor[j] = ƒ(Feed_Sulf[j], Feed_Sulf_Coef[j]) Tot_feed_grav_factor = Σ[(Feed_Grav_Factor[j] * Feed_Flow[j]] / Tot_feed_flow Tot_feed_watk_factor = Σ[(Feed_Watk_Factor[j] * Feed_Flow[j]] / Tot_feed_flow
6.	Calculate catalyst additive factor for each additive. For j = 1 to Cat_Add_Num Cat_Add_Factor = ƒ(Cat_Add[j], Cat_Add_Coef[j])
7.	Calculate other catalyst factors. Catalyst micro-activity factor Cat_Micro_Act_Factor = ƒ(Cat_Micro_Act, Cat_Micro_Act_Coef) Catalyst surface area factor Cat_Surf_Area_Factor = ƒ(Cat_Surf_Area, Cat_Surf_Area_Coef)
8.	Carbon on feed For j = 1 to Feed_Num Con_Carbon_Factor[j] = ƒ(Feed_Con_Carbon[j], Feed_Con_Carbon_Coef[j]) Tot_con_carbon_factor = Σ(Con_Carbon_Factor[j]
9.	Calculate the product yield. Product_Yield = ƒ(at_Bal_Coef * (exp(K1_Coef + Severity_K2_Factor + Riser_Temp_Factor + Tot_feed_grav_factor + Cat_Add_Factor + Regen_Temp_Factor + Tot_feed_sulf_factor) * Severity_K3_Factor * Tot_feed_watk_factor *Cat_Micro_Act_Factor * Cat_Surf_Area_Factor * Riser_Press_Factor * Riser_Steam_Flow_Factor + Tot_con_carbon_factor + Rx_SS_To_Cat_Factor
10.	Filter the calculated product yield based on the supplied Filter_Type.
11.	Calculate the product yield to weight fraction of feed. Product_Wt_Fraction = Product_Yield * 0.01
12.	Calculate the uncompensated product as a function of total feed. Product_Raw = Product_Wt_Fraction * Tot_feed_flow

Calculation Algorithm	
Step #	**Description**
13.	Apply catalyst correction factor to the calculated product. This correction factor comes from the severity compensation factor. Product_Corr = Product_Raw * Cat_Corr_Fact
14.	Apply bias between dynamically compensated corrected product and measured product. This bias comes from the product compensation function for this product. Product_TK_Units = Product_Corr + Add_Bias
15.	Convert product to user units Product_User_Units = Product_TK_Units / (Conv_Factor * Product_Grav_Molwt)

Input table

The input table provides a description of the function inputs.

Input table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of independent feed streams used in off-line design
Feed_Flow[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mlb/hr	Feed mass flow for each independent feed
Feed_Molwt[ ]	Molwt_Proc or Feed_Prop	Feed_Num > 0		Feed molecular weight for each independent feed
Feed_Sulf[ ]	Feed_Prop	Feed_Num > 0	Wt %	Percent sulfur content in each independent feed
Feed_Sulf_Coef[ ]	Offline design	Feed_Num > 0		Feed sulfur coefficient for each independent feed from off-line design
Feed_Grav[ ]	Gravity_Proc or Feed_Prop	Feed_Num > 0	TK units spgr	Feed gravity for each independent feed
Feed_Grav_Ref[ ]		Feed_Num > 0	TK units spgr	Feed gravity reference for each independent feed
Feed_Grav_Coef[ ]	Offline design	Feed_Num > 0		Feed gravity coefficient for each independent feed from off-line design
Feed_WatK[ ]	WatsonK_Proc or Feed_Prop	Feed_Num > 0		Feed Watson K for each independent feed
Feed_WatK_Ref[ ]		Feed_Num > 0		Feed Watson K reference for each independent feed
Feed_WatK_Coef[ ]	Offline design	Feed_Num > 0		Feed Watson K coefficient for each independent feed from off-line design

Input table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Reactor_Level	Reactor_Prop		Inches H2O	Reactor catalyst level
Reactor_Level_ Coef	Offline design			Reactor level coefficient from off-line design
Riser_Steam_ Flow	Flow_Proc or Reactor_Prop		TK units Mlb/hr	Total steam flow to riser
Riser_Temp	Temperature_ Proc		TK units Deg F	Riser outlet temperature
Riser_Temp_Ref			TK units Deg F	Riser outlet temperature reference
Riser_Temp_Coef	Offline design			Riser temperature coefficient from off-line design
Riser_Press	Pressure_Pro c		TK units psig	Riser pressure
Riser_Press_Coef	Offline design			Riser pressure coefficient from off-line design
Riser_Volume	Reactor_Prop		Ft^3	Riser volume
Riser_Temp_Avg_ Coef	Offline design			Average riser temperature coefficient from off-line design
Riser_Feed_Exp_C oef	Offline design			Riser feed expansion coefficient from off-line design
Regen_Temp	Temperature_ Proc		TK units Deg F	Regenerator bed temperature
Regen_Temp_Ref			TK units Deg F	Regenerator bed temperature reference
Regen_Press	Pressure_Pro c		TK units psig	Regenerator pressure
Regen_O2			%	Regenerator percent oxygen in the flue gas
Meas_Conversion	Meas_Conver sion			Measured conversion in units from off-line design
Cat_To_Oil	Cat_Circ_Pre d			Catalyst to oil ratio
Cat_To_Oil_Coef	Offline design			Catalyst to oil ratio coefficient from off-line design
Cat_Surf_Area			M^2/g	Catalyst surface area in off-line design units
Cat_Surf_Area_Co ef	Offline design			Catalyst surface area coefficient from off-line design
Cat_Micro_Act			%	Catalyst micro activity
Cat_Micro_Act_ Coef	Offline design			Catalyst micro activity from off-line design

Input table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Cat_Add_Num				Number of catalyst additives included in off-line design
Cat_Add[ ]				Catalyst additive factor for each additive used in off-line design
Cat_Add_Coef[ ]	Offline design			Catalyst additive factor coefficient for each additive used in off-line design
Cat_Slip_Factor				Catalyst slip factor
Carb_Cat_Temp_Coef	Offline design			Carbon on catalyst temperature coefficient from off-line design
Carb_Cat_Press_Coef	Offline design			Carbon on catalyst pressure coefficient from off-line design
Carb_Cat_O2_ Coef	Offline design			Carbon on catalyst oxygen coefficient from off-line design
Carb_Cat_Comp	Offline design			Carbon on catalyst compensation (called gain in CL)
Carb_Cat_Bias				Carbon on catalyst bias
Carb_Cat_Coef	Offline design			Carbon on catalyst coefficient from off-line design
Geometry_Coef	Offline design			Geometry coefficient from off-line design
Vorhes_Coef	Offline design			Vorhes coefficient from off-line design
Local_Pressure			TK units psi	Local atmospheric pressure
Cat_Corr_Fact	Calculation_Comp			Catalyst correction factor. The user determines whether to use the current catalyst correction factor or the catalyst correction factor that has been held since last update. Calculation_Comp calculates both these values.
Add_Bias	Calculation_Comp			Additive bias for yield. The user determines whether to use the current bias or the bias that has been held since last update. Calculation_Comp calculates both these values. If the held value is used for the Catalyst correction factor then the held value must be used for the additive bias. Similarly, if the current value is used for the Catalyst correction factor then the current value must be used for the additive bias.

Input table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Add_Bias_Filt_ Time			Minutes	Filter time for the additive bias
Add_Bias_Filt_ Shift				Previous filtered severity additive bias storage
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift[ ]				Previous filtered value storage
Etime			Minutes	Application execution time
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..3]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Severity		Calculated severity corrected and biased
Severity_Raw		Raw unbiased and uncorrected calculated severity
Severity_Corr		Calculated severity corrected with catalyst correction factor (unbiased)
Conversion		Calculated conversion in percent
Geometry_Factor		Calculated geometry factor
Feed_Char_Factor[ ]		Calculated feed characterization factor for each independent feed
Feed_Sulf_Factor[ ]		Calculated feed sulfur factor for each independent feed
Carbon_On_Cat	Wt%	Calculated carbon on catalyst
Cat_Char_Factor		Calculated catalyst characterization factor
Rxn_Temp_Factor		Calculated reaction temperature factor
Avg_Rxn_Temp_Factor		Average reaction temperature factor
Feed_Exp_Factor		Calculated feed expansion factor
Reactor_Level_Factor		Calculated reactor level factor
Riser_Press_Factor		Calculated riser pressure factor
Cat_Add_Factor		Calculated combined catalyst additive factor
Denom		Calculated result of all terms in the denominator for raw severity calculation.
Numer		Calculated result of all terms in the numerator for raw severity calculation
Stream_in_Error		No longer used.
Add_Bias_Filt_Shift		Previous filtered additive bias storage
Filter_Shift[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..3]		Last good value storage array for key output variables

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The additive bias is a bad value
2	The feed flow is a bad value. Feed number ____ is the feed in error.
3	The feed molecular weight is a bad value. Feed number ____ is the feed in error.
4	The feed molecular weight is a miniscule value.
5	The regenerator pressure is a bad value.
6	The local atmospheric pressure is a bad value.
7	The riser pressure is a bad value.
8	The riser pressure is less than or equal to zero.
9	The riser outlet temperature is a bad value.
10	The riser outlet temperature reference is a bad value.
11	The riser outlet temperature reference is a miniscule value.
12	The regenerator bed temperature is a bad value.
13	The regenerator bed temperature reference is a bad value.
14	The regenerator bed temperature reference is a miniscule value.
15	The catalyst surface area is a bad value.
16	The catalyst surface area is less than zero.
17	The reactor catalyst level is a bad value
18	The feed gravity is a bad value. Feed number ____ is the feed in error.
19	The feed gravity is less than or equal to zero. Feed number ____ is the feed in error.
20	The feed gravity reference is a bad value. Feed number ____ is the feed in error.
21	The feed gravity reference is less than or equal to zero. Feed number ____ is the feed in error.
22	The feed Watson K is a bad value. Feed number ____ is the feed in error.
23	The feed Watson K is less than or equal to zero. Feed number ____ is the feed in error.
24	The feed Watson K Reference is a bad value. Feed number ____ is the feed in error.
25	The feed Watson K Reference is less than or equal to zero. Feed number ____ is the feed in error.
26	The geometry coefficient is a bad value.
27	The feed gravity coefficient is a bad value. Feed number ____ is the feed in error.
28	The feed Watson K coefficient is a bad value. Feed number ____ is the feed in error.
29	The feed Watson K is less than the feed Watson K reference. Feed number ____ is the feed in error.

Return Status	
Status value	**Description**
30	The feed percent sulfur is a bad value. Feed number ____ is the feed in error.
31	The feed percent sulfur is less than zero. Feed number ____ is the feed in error.
32	The feed percent sulfur coefficient is a bad value. Feed number ____ is the feed in error.
33	The total feed flow is a miniscule value.
34	The carbon on catalyst temperature coefficient is a bad value.
35	The carbon on catalyst oxygen coefficient is a bad value.
36	The regenerator percent oxygen in the flue gas is a bad value.
37	The carbon on catalyst compensation is a bad value.
38	The carbon on catalyst bias is a bad value.
39	The carbon on catalyst pressure coefficient is a bad value.
40	The carbon on catalyst coefficient is a bad value.
41	The catalyst surface area coefficient is a bad value.
42	The catalyst micro activity is a bad value.
43	The catalyst micro activity is less than zero.
44	The catalyst micro activity coefficient is a bad value.
45	The riser temperature coefficient is a bad value.
46	The average riser temperature coefficient is a bad value.
47	The riser outlet temperature is a bad value.
48	The regenerator bed temperature is a bad value.
49	The riser feed expansion coefficient is a bad value.
50	The measured conversion is a bad value.
51	The reactor catalyst level coefficient is a bad value.
52	The catalyst additive factor is a bad value. Feed number ____ is the feed in error.
53	The catalyst additive factor coefficient is a bad value. Feed number ____ is the feed in error.
54	The riser pressure coefficient is a bad value.
55	The riser volume is a bad value.
56	Either the riser volume or riser pressure in Atmosphere units is zero.
57	The total steam flow to riser is a bad value.
58	The catalyst slip factor is a bad value.
59	The catalyst to oil ratio is a bad value.
60	The catalyst to oil ratio is less than or equal to zero.
61	The catalyst to oil ratio coefficient is a bad value.

Return Status	
Status value	**Description**
62	The Vorhes coefficient is a bad value.
63	The denominator in the severity calculation is less than or equal to zero.
64	The catalyst correction factor is a bad value.
65	The Severity equals negative one (-1). Unable to calculate Conversion.
66	Errors filtering the additive bias. See associated message from Filter function
67	Errors filtering the raw severity. See associated message from Filter function
141 - 146	An error occurred in the Filter_First_Ord_Lag function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 - 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 - 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
Negative errors	Last good value holding is active. The error that is causing the bad value can be determined by looking up the positive of the error value

Examples Table

The following table shows example for the Severity_Pred function. The example assumes no filtering and no last good value holding.

Examples Table				
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Feed_Num	I	1		
Feed_Flow[ ]	I	448.95 Mlb/hr		
Feed_Molwt[ ]	I	485.02		
Feed_Sulf[ ]	I	2.05 wt%		
Feed_Sulf_Coef[ ]	I	0		
Feed_Grav[ ]	I	0.9259 spgr		
Feed_Grav_Ref[ ]	I	0.946488		
Feed_Grav_Coef[ ]	I	-0.01483		
Feed_WatK[ ]	I	12.03463		
Feed_WatK_Ref[ ]	I	10		
Feed_WatK_Coef[ ]	I	1.275086		
Reactor_Level	I	0		
Reactor_Level_Coef	I	0		

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Riser_Steam_Flow	I	29.106 Mlb/hr		
Riser_Temp	I	966.2 Deg F		
Riser_Temp_Ref	I	966.0 Deg F		
Riser_Temp_Coef	I	-8.68912		
Riser_Press	I	18.415 psig		
Riser_Press_Coef	I	-0.78322		
Riser_Volume	I	1330 ft3		
Riser_Temp_Avg_Coef	I	1		
Riser_Feed_Exp_Coef	I	3.0		
Regen_Temp	I	1335.2 Deg F		
Regen_Temp_Ref	I	1335.0 Deg F		
Regen_Press	I	15.23 psig		
Regen_O2	I	4.0%		
Meas_Conversion	I	71.8194 wt%		
Cat_To_Oil	I	6.34741		
Cat_To_Oil_Coef	I	1.033275		
Cat_Surf_Area	I	140 m^2/g		
Cat_Surf_Area_Coef	I	0		
Cat_Micro_Act	I	65 %		
Cat_Micro_Act_Coef	I	1.201719		
Cat_Add_Num	I	1		
Cat_Add[ ]	I	0		
Cat_Add_Coef[ ]	I	0		
Cat_Slip_Factor	I	0.85		
Carb_Cat_Temp_Coef	I	0		
Carb_Cat_Press_Coef	I	0		
Carb_Cat_O2_Coef	I	0		
Carb_Cat_Comp	I	0		
Carb_Cat_Bias	I	0		
Carb_Cat_Coef	I	0		
Geometry_Coef	I	-0.1287		
Vorhes_Coef	I	0.687895		

Examples Table				
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Local_Pressure	I	14.696		
Cat_Corr_Fact	I	1.0		
Add_Bias	I	0		
Add_Bias_Filt_Time	I	0		
Filter_Type	I	0		
LGV_Num	I	0		
Ret_Status	O	0		
Severity	O	2.5485		Severity is biased by -2.27262
Severity_Raw	O	4.8211		
Severity_Corr	O	4.8211		
Conversion	O	71.819 %		
Geometry_Factor	O	0.8792		
Feed_Char_Factor[ ]	O	2.3547		
Feed_Sulf_Factor[ ]	O	1.0		
Carbon_On_Cat	O	1.0		
Cat_Char_Factor	O	150.87		
Rxn_Temp_Factor	O	0.0023		
Avg_Rxn_Temp_Factor	O	966.20		
Feed_Exp_Factor	O	2.0773		
Reactor_Level_Factor	O	1.0		
Riser_Press_Factor	O	0.5293		
Cat_Add_Factor	O	1.0000		
Denom	O	0.9830		
Numer	O	4.7645		
Stream_in_Error	O	0		

Section 10 - Product (Yield) Prediction (Product_Pred)

10.1 Overview

In This Section

This function calculates the yield and production of a specified product. Prior to outputting the production, the value is filtered, and corrected using the catalyst correction factor and then biased. The catalyst correction factor and bias are provided by the Calculation Compensation function. Last good value holding is also available.

Usage of Product_Pred

Configure a separate instance of the Product_Pred function for each distinct product on the FCCU.

Product_Pred requires input from the predicted severity function – Severity_Pred – to calculate the predicted product yield.

10.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Product_Pred to calculate the predicted product yield:

Calculation Algorithm	
Step #	**Description**
1.	Filter the additive bias. Use the first order lag filter.
2.	Calculate total feed flow as sum of all the flows: Tot_feed_flow = Σ Feed_Flow[i]
3.	Convert riser pressures to absolute units and then to atmospheres: (Riser_Press + Local_Pressure) / 14.696
4.	Scale variables in accordance to off-line design calculations: scale riser temperature Riser_temp_s = ƒ(Riser_Temp, Riser_Temp_Ref) scale regenerator temperature Regen_temp_s = ƒ(Regen_Temp, Regen_Temp_Ref) Scale catalyst surface area Cat_surf_area_s = ƒ(Cat_Surf_Area) For j = 1 to Feed_Num Convert all gravity values to API units prior to scaling gravity. Feed_grav_s[j] = ƒ(Feed_Grav[j], Feed_Grav_Ref[j]) Feed_watK_s[j] = ƒ(Feed_WatK[j], Feed_WatK_Ref[j]) Calculate the severity factors. Severity_K2_Factor = ƒ(Severity, Severity_K2_Coef) Severity_K3_Factor = ƒ (Severity, Severity_K3_Coef)

Calculation Algorithm	
Step #	**Description**
	Calculate the riser reaction factors. Riser Pressure Factor Riser_Press_Factor = ƒ(Riser_press_atm, Riser_Press_Coef) Riser Temperature Factor Riser_Temp_Factor = ƒ(Riser_temp_s, Riser_Temp_Coef) Reactor stripping steam to catalyst factor Rx_SS_To_Cat_Factor = ƒ(Rx_SS_To_Cat, Rx_SS_To_Cat_Coef) Riser Steam Flow Factor Riser_Steam_Flow_Factor = ƒ (Riser_Steam_Flow, Riser_Steam_Flow_Coef) Calculate the regenerator reaction factors. Regen_Temp_Factor = ƒ(Regen_temp_s, Regen_Temp_Coef)
5.	Calculate the feed characterization factors for each independent feed and sum them together For j = 1 to Feed_Num Feed_Grav_Factor[j] = ƒ(Feed_Grav_s[j], Feed_Grav_Coef[j]) Feed_WatK_Factor[j] = ƒ (Feed_WatK_s[j], Feed_WatK_Coef[j]) Feed_Sulf_Factor[j] = ƒ(Feed_Sulf[j], Feed_Sulf_Coef[j]) Con_Carbon_Factor[j] = ƒ(Feed_Con_Carbon[j], Feed_Con_Carbon_Coef[j]) Tot_feed_grav_factor = Σ[Feed_Grav_Factor[j] * Feed_Flow[j]] / Tot_feed_flow Tot_feed_watk_factor = Σ[Feed_Watk_Factor[j]) * Feed_Flow[j]] / Tot_feed_flow Tot_feed_sulf_factor = Σ[Feed_Sulf _Factor[j]) * Feed_Flow[j]] / Tot_feed_flow Tot_con_carbon_factor = Σ[Con_Carbon_Factor[j* Feed_Flow[j]] / Tot_feed_flow
6.	Calculate catalyst additive factor for each additive. For j = 1 to Cat_Add_Num Cat_Add_Factor = ƒ(Cat_Add[j], Cat_Add_Coef[j])
7.	Calculate other catalyst factors. Catalyst micro-activity factor Cat_Micro_Act_Factor = ƒ(Cat_Micro_Act, Cat_Micro_Act_Coef) Catalyst surface area factor Cat_Surf_Area_Factor = ƒ(Cat_Surf_Area, Cat_Surf_Area_Coef)

Calculation Algorithm	
Step #	**Description**
8.	Calculate the product yield. Product_Yield = ƒ(Mat_Bal_Coef , K1_Coef , Severity_K2_Factor , Riser_Temp_Factor , Tot_feed_grav_factor , Cat_Add_Factor , Regen_Temp_Factor , Tot_feed_sulf_factor , Severity_K3_Factor , Tot_feed_watk_factor ,Cat_Micro_Act_Factor , Cat_Surf_Area_Factor , Riser_Press_Factor , Riser_Steam_Flow_Factor , Tot_con_carbon_factor , Rx_SS_To_Cat_Factor)
9.	Filter the calculated product yield based on the supplied Filter_Type.
10.	Calculate the product yield to weight fraction of feed. Product_Wt_Fraction = Product_Yield * 0.01
11.	Calculate the uncompensated product as a function of total feed. Product_Raw = Product_Wt_Fraction * Tot_feed_flow
12.	Apply catalyst correction factor to the calculated product. This correction factor comes from the severity compensation factor. Product_Corr = Product_Raw * Cat_Corr_Fact
13.	Apply bias between dynamically compensated corrected product and measured product. This bias comes from the product compensation function for this product. Product_TK_Units = Product_Corr + Add_Bias
14.	Convert product to user units Product_User_Units = Product_TK_Units / (Conv_Factor * Product_Grav_Molwt)
15.	Call Last_Good_Value utility function to provide last good value processing.

Input table

This function requires inputs from the following functions:

- Flow_Proc
- Molwt_Proc
- Enth_Proc
- Temperature_Proc

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of independent feed streams used in off-line design
Feed_Flow[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mlb/hr	Feed mass flow for each independent feed
Feed_Sulf[ ]	Feed_Prop	Feed_Num > 0	Wt %	Percent sulfur content in each independent feed
Feed_Sulf_Coef[ ]	Offline design	Feed_Num > 0		Feed sulfur coefficient for each independent feed from off-line design

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Grav[ ]	Gravity_Proc or Feed_Prop	Feed_Num > 0	TK units spgr	Feed gravity for each independent feed
Feed_Grav_Ref[ ]		Feed_Num > 0	TK units spgr	Feed specific gravity reference for each independent feed
Feed_Grav_Coef[ ]	Offline design	Feed_Num > 0		Feed gravity coefficient for each independent feed from off-line design
Feed_WatK[ ]	WatsonK_Proc or Feed_Prop	Feed_Num > 0		Feed Watson K for each independent feed
Feed_WatK_Ref[ ]		Feed_Num > 0		Feed Watson K reference for each independent feed
Feed_WatK_Coef[ ]	Offline design	Feed_Num > 0		Feed Watson K coefficient for each independent feed from off-line design
Feed_Con_Carbon[ ]	Feed_Prop	Feed_Num > 0	Wt %	Feed con carbon for each independent feed from off-line design
Feed_Con_Carbon_Coef[ ]	Offline design	Feed_Num > 0		Feed con carbon coefficient for each independent feed from off-line design
Riser_Steam_Flow	Flow_Proc or Reactor_Prop		TK units Mlb/hr	Total steam flow to riser
Riser_Steam_Flow_Coef	Offline design			Riser steam flow coefficient from off-line design
Riser_Temp	Temperature_Proc		TK units Deg F	Riser outlet temperature
Riser_Temp_Ref			TK units Deg F	Riser outlet temperature reference
Riser_Temp_Coef	Offline design			Riser temperature coefficient from off-line design
Riser_Press	Pressure_Proc		TK units psig	Riser pressure
Riser_Press_Coef	Offline design			Riser pressure coefficient from off-line design
Rx_SS_To_Cat	Cat_Circ_Pred			Reactor stripping steam to catalyst ratio
Rx_SS_To_Cat_Coef	Offline design			Reactor stripping steam to catalyst ratio coefficient from off-line design
Regen_Temp	Temperature_Proc		TK units Deg F	Regenerator bed temperature
Regen_Temp_Ref			TK units Deg F	Regenerator bed temperature reference

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Regen_Temp_Coef	Offline design			Regenerator temperature coefficient from off-line design
Severity	Severity_Pred		Wt%	Calculated/Predicted severity based on weight percent
Severity_K2_Coef	Offline design			First severity coefficient from off-line design
Severity_K3_Coef	Offline design			Second severity coefficient from off-line design
Cat_Surf_Area			M^2/g	Catalyst surface area in off-line design units
Cat_Surf_Area_Coef	Offline design			Catalyst surface area coefficient from off-line design
Cat_Micro_Act			%	Catalyst micro activity
Cat_Micro_Act_Coef	Offline design			Catalyst micro activity from off-line design
Cat_Add_Num				Number of catalyst additives included in off-line design
Cat_Add[ ]				Catalyst additive factor for each additive used in off-line design
Cat_Add_Coef[ ]	Offline design			Catalyst additive factor coefficient for each additive used in off-line design
K1_Coef	Offline design			Described as a constant in the CL
Mat_Bal_Coef	Offline design			Material balance coefficient from the off-line design
Product_Grav_Molwt	Gravity_Proc or Molwt_Proc		TK units spgr or no units if molecular weight.	Used to convert product in toolkit units to product in user units. Input specific gravity if the desired user units are volumetric flow for a liquid. Input molecular weight if the desired user units are volumetric flow for gas. Input one (1) if the desired user units are mass flow.
Local_Pressure			TK units psi	Local atmospheric pressure
Conv_Factor				Conversion factor to go from user units to TK units (don't include gravity or molecular weight)

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Cat_Corr_Fact	Calculation_Comp			Catalyst correction factor. The user determines whether to use the current catalyst correction factor or the catalyst correction factor that has been held since last update. Calculation_Comp calculates both these values.
Add_Bias	Calculation_Comp			Additive bias for product yield. The user determines whether to use the current bias or the bias that has been held since last update. Calculation_Comp calculates both these values. If the held value is used for the Catalyst correction factor then the held value must be used for the additive bias. Similarly, if the current value is used for the Catalyst correction factor then the current value must be used for the additive bias.
Add_Bias_Filt_Time			Minutes	Filter time for the additive bias
Add_Bias_Filt_Shift				Previous filtered catalyst circulation additive bias storage
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift[ ]				Previous filtered value storage
Etime			Minutes	Application execution time

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..6]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Product_Yield	Wt %	Calculated uncorrected product yield
Product_Wt_Fraction		Calculate uncorrected product weight fraction
Product_Raw	TK units Mlb/hr	Raw unbiased and uncorrected calculated product
Product_Corr	TK units Mlb/hr	Calculated product corrected with catalyst correction factor (unbiased)
Product_TK_Units	TK units Mlb/hr	Calculated product corrected and biased
Product_User_Units	User units	Calculated product corrected and biased
Severity_K2_Factor		Severity factor based on K2 coefficient
Severity_K3_Factor		Severity factor based on K3 coefficient

Output Table		
Output	**Eng. Units**	**Description**
Riser_Press_Factor		Riser pressure factor
Riser_Temp_Factor		Riser temperature factor
Rx_SS_To_Cat_Factor		Reactor stripping steam to catalyst factor
Riser_Steam_Flow_Factor		Riser steam factor
Regen_Temp_Factor		Regenerator temperature factor
Feed_Grav_Factor[ ]		Feed gravity factor for each independent feed
Feed_WatK_Factor[ ]		Feed Watson K factor for each independent feed
Feed_Sulf_Factor[ ]		Feed Sulfur factor for each independent feed
Cat_Add_Factor		Combined catalyst additive factor
Cat_Surf_Area_Factor		Catalyst surface area factor
Cat_Micro_Act_Factor		Catalyst micro activity factor
Con_Carbon_Factor[ ]		Feed con carbon factor for each independent feed
Stream_in_Error		No longer used.
Add_Bias_Filt_Shift		Previous filtered additive bias storage
Filter_Shift[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..6]		Last good value storage array for key output variables

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The additive bias is a bad value
2	The feed flow is a bad value. Feed number ____ is the feed in error.
3	The riser pressure is a bad value.
4	The riser pressure is less than or equal to zero.
5	The local pressure is a bad value.
6	The riser outlet temperature is a bad value.
7	The riser outlet temperature reference is a bad value.
8	The riser outlet temperature reference is miniscule.
9	The regenerator bed temperature is a bad value.

Return Status	
Status value	**Description**
10	The regenerator bed temperature reference is a bad value.
11	The regenerator bed temperature reference is miniscule.
12	The feed gravity is a bad value. Feed number ____ is the feed in error.
13	The feed gravity is less than or equal to zero. Feed number ____ is the feed in error.
14	The feed gravity reference is a bad value. Feed number ____ is the feed in error.
15	The feed gravity reference is less than or equal to zero. Feed number ____ is the feed in error.
16	The feed Watson K is a bad value. Feed number ____ is the feed in error.
17	The feed Watson K is less than or equal to zero. Feed number ____ is the feed in error.
18	The feed Watson K Reference is a bad value. Feed number ____ is the feed in error.
19	The feed Watson K Reference is less than or equal to zero. Feed number ____ is the feed in error.
20	The severity is a bad value.
21	The severity is less than or equal to zero.
22	The first severity (K2) coefficient is a bad value.
23	The second severity (K3) coefficient is a bad value.
24	The riser pressure coefficient is a bad value.
25	The riser temperature coefficient is a bad value.
26	The reactor stripping steam to catalyst ratio is a bad value.
27	The reactor stripping steam to catalyst ratio coefficient is a bad value.
28	The riser steam flow is a bad value.
29	The riser steam flow is less than or equal to zero.
30	The riser steam flow coefficient is a bad value.
31	The regenerator temperature coefficient is a bad value.
32	The feed gravity coefficient is a bad value. Feed number ____ is the feed in error.
33	The feed Watson K coefficient is a bad value. Feed number ____ is the feed in error.
34	The feed Watson K value is less than the feed Watson K reference. Feed number ____ is the feed in error.
35	The feed sulfur is a bad value. Feed number ____ is the feed in error.
36	The feed sulfur coefficient is a bad value. Feed number ____ is the feed in error.
37	The feed con carbon is a bad value. Feed number ____ is the feed in error.
38	The feed con carbon coefficient is a bad value. Feed number ____ is the feed in error.
39	The total feed flow is a miniscule value.
40	The catalyst additive is a bad value. Feed number ____ is the feed in error.
41	The catalyst additive coefficient is a bad value. Feed number ____ is the feed in error.

Return Status	
Status value	**Description**
42	The catalyst micro activity is a bad value.
43	The catalyst micro activity less than zero.
44	The catalyst micro activity coefficient is a bad value.
45	The catalyst surface area is a bad value.
46	The catalyst surface area is less than zero.
47	The catalyst surface area coefficient is a bad value.
48	The material balance coefficient is a bad value.
49	The K1 coefficient is a bad value.
50	The catalyst correction factor is a bad value.
51	The conversion factor is a bad value.
52	The conversion factor is miniscule.
53	The product gravity or molecular weight is a bad value.
54	The product gravity or molecular weight is less than or equal to zero.
55	Errors filtering the additive bias. See associated message from Filter function
56	Errors filtering the product yield. See associated message from Filter function
141 - 146	An error occurred in the function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 - 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 - 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
Negative errors	Last good value holding is active. The error that is causing the bad value can be determined by looking up the positive of the error value

Examples Table The following table shows example for the Product_Pred function. The example assumes no filtering and no last good value holding.

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Feed_Num	I	1		
Feed_Flow\1	I	448.9499291 Mlb/hr		
Feed_Sulf\1	I	2.05 %		
Feed_Sulf_Coef\1	I	0		
Feed_Grav\1	I	0.9259 spgr		
Feed_Grav_Ref\1	I	0.946488 spgr		
Feed_Grav_Coef\1	I	0		
Feed_WatK\1	I	12.03462739		
Feed_WatK_Ref\1	I	10		
Feed_WatK_Coef\1	I	0		
Feed_Con_Carb\1	I	3.47 wt%		
Feed_Con_Carb_Coef\1	I	0		
Riser_Steam_Flow	I	29.106 Mlb/hr		
Riser_Steam_Flow_Coef	I	0		
Riser_Temp	I	966.2 Deg F		
Riser_Temp_Ref	I	966 Deg F		
Riser_Temp_Coef	I	8.046256714		
Riser_Press	I	18.415 psig		
Riser_Press_Coef	I	0		
Regen_Temp	I	1335.2 Deg F		
Regen_Temp_Ref	I	1250.0 Deg F		
Regen_Temp_Coef	I	-1.373034182		
Rx_SS_To_Cat	I	2.794220209		
Rx_SS_To_Cat_Coef	I	0		
Severity	I	2.548540071		
Severity_K2_Coef	I	0		
Severity_K3_Coef	I	0.20174191		
Cat_Surf_Area	I	140 m^2/g		
Cat_Surf_Area_Coef	I	0		

Examples Table				
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Cat_Micro_Act	I	65 %		
Cat_Micro_Act_Coef	I	0		
Cat_Add_Num	I	1		
Cat_Add\1	I	0		
Cat_Add_Coef\1	I	0		
K1_Coef	I	9.713201364		
Mat_Bal_Coef	I	1		
Product_Grav_Molwt	I	20.688		Molecular Weight
Local_Pressure	I	14.696 psi		
Conv_Factor	I	1.0		
Cat_Corr_Fact	I	1.0		
Add_Bias	I	0		
Add_Bias_Filt_Time	I	0		
Filter_Type	I	0		
LGV_Num	I	0		
Ret_Status	O	0		
Product_Yield	O	18215.		
Product_Wt_Fraction	O	182.15 wt%		
Product_Raw	O	81777 Mlb/hr		
Product_Corr	O	81777 Mlb/hr		
Product_TK_Units	O	81777 Mlb/hr		
Product_User_Units	O	3952.881592		
Severity_K2_Factor	O	0.0		
Severity_K3_Factor	O	1.2077		
Riser_Press_Factor	O	1.0		
Riser_Temp_Factor	O	0.0017		
Rx_SS_To_Cat_Factor	O	0.0		
Riser_Steam_Flow_Factor	O	1.0		
Regen_Temp_Factor	O	-0.0936		
Feed_Gravity_Factor\1	O	0.0		
Feed_WatK_Factor\1	O	1.0		
Feed_Sulf_Factor\1	O	0.0		

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Cat_Add_Factor	O	0.0		
Cat_Surf_Area_Factor	O	1.0		
Cat_Micro_Act_Factor	O	1.0		
Con_Carbon_Factor\1	O	0.0		
Stream_in_Error	O	0.0		

Section 11 - Octane Prediction (Octane_Pred)

11.1 Overview

In This Section

This function calculates the octane and octane flow of the gasoline product. Prior to outputting the octane volume the value is filtered, converted to user units and then biased. Last good value holding is also available.

Usage of Octane_Pred

Configure a separate instance of the Octane_Pred function for each distinct product that requires an Octane prediction.

Octane_Pred requires input from the predicted severity function – Severity_Pred – to calculate the predicted Octane.

11.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Octane_Pred to calculate the Octane in the gasoline product:

Calculation Algorithm	
Step #	**Description**
1.	Filter the additive bias. Use the first order lag filter. Filter the multiplicative bias. Use the first order lag filter.
2.	Scale variables in accordance to off-line design calculations: scale riser temperature Riser_temp_s = ƒ (Riser_Temp, Riser_Temp_Ref) For j = 1 to Feed_Num Convert all gravity values to API units prior to scaling gravity. Feed_grav_s[j] = ƒ(Feed_Grav[j], Feed_Grav_Ref[j]) Feed_watK_s[j] = ƒ(Feed_WatK[j], Feed_WatK_Ref[j]) Calculate the riser reaction factors. Riser Temperature Factor Riser_Temp_Factor = ƒ(Riser_temp_s, Riser_Temp_Coef)
3.	Calculate the feed characterization factors for each independent feed and sum them together For j = 1 to Feed_Num Feed_Grav_Factor[j] = ƒ(Feed_Grav_s[j], Feed_Grav_Coef[j]) Feed_WatK_Factor[j] = ƒ(Feed_WatK_s[j], Feed_WatK_Coef[j]) Tot_feed_grav_factor = Σ [Feed_Grav_Factor[j] * Feed_Flow[j]] / Tot_feed_flow Tot_feed_watk_factor = Σ[Feed_Watk_Factor[j]) * Feed_Flow[j]] / Tot_feed_flow
4.	Calculate the Naphtha factors. Naphtha_90_Factor = ƒ(Naphtha_90, Naphtha_90_Coef) Naphtha_RVP_Factor = ƒ(Naphtha_RVP, Naphtha_RVP_Coef) Calculate the Severity factor. Severity_Factor = ƒ(Severity, Severity_Coef)

Calculation Algorithm	
Step #	**Description**
5.	Calculate catalyst additive factor for each additive. For j = 1 to Cat_Add_Num Cat_Add_Factor = ƒ(Cat_Add[j], Cat_Add_Coef[j]) Cat_Add_Factor = ƒ(Cat_Add_Factor)
6.	Calculate the octane: Octane_Raw = ƒ((K1_Coef, Riser_Temp_Factor, Tot_feed_grav_factor, Tot_feed_watk_factor, Naphtha_RVP_Factor, Naphtha_90_Factor, Severity_Factor, Cat_Add_Factor)
7.	Filter the calculated octane based on the supplied Filter_Type.
8.	Apply multiplicative and additive biases to the calculated octane.
9.	Calculate the octane volume in toolkit units where volume is in MBPD. Octane_Vol_TK_Units = Octane * Naphtha_Flow / (Naphtha_Grav * 14.59146)
10.	Convert octane volume to user units Octane_Vol_User_Units = Octane_Vol_TK_Units / Conv_Factor
11.	Call Last_Good_Value utility function to provide last good value processing.

Input table The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_Num				Total number of independent feed streams used in off-line design
Feed_Flow[ ]	Flow_Proc or Feed_Prop	Feed_Num > 0	TK units Mlb/hr	Feed mass flow for each independent feed
Feed_Grav[ ]	Gravity_Proc or Feed_Prop	Feed_Num > 0	TK units spgr	Feed gravity for each independent feed
Feed_Grav_Ref[ ]		Feed_Num > 0	TK units spgr	Feed gravity reference for each independent feed
Feed_Grav_Coef[ ]	Offline design	Feed_Num > 0		Feed gravity coefficient for each independent feed from off-line design
Feed_WatK[ ]	WatsonK_Proc or Feed_Prop	Feed_Num > 0		Feed Watson K for each independent feed
Feed_WatK_Ref[ ]		Feed_Num > 0		Feed Watson K reference for each independent feed

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Feed_WatK_Coef[ ]	Offline design	Feed_Num > 0		Feed Watson K coefficient for each independent feed from off-line design
Riser_Temp	Temperature_Proc		TK units Deg F	Riser outlet temperature
Riser_Temp_Ref			TK units Deg F	Riser outlet temperature reference
Riser_Temp_Coef	Offline design			Riser temperature coefficient from off-line design
Severity	Severity_Pred		Wt%	Calculated/Predicted severity based on wt %
Severity_Coef	Offline design			Severity coefficient from off-line design
Cat_Add_Num				Number of catalyst additives included in off-line design
Cat_Add[ ]				Catalyst additive factor for each additive used in off-line design
Cat_Add_Coef[ ]	Offline design			Catalyst additive factor coefficient for each additive used in off-line design
Naphtha_Flow	Flow_Proc or Gasoline_Prop		TK units Mlb/hr	Total Naphtha product mass flow
Naphtha_Grav	Gravity_Proc or Gasoline_Prop		TK units spgr	Naphtha specific gravity
Naphtha_RVP	Reid_Vap_Press or Gasoline_Prop		TK units Psig	Naphtha RVP
Naphtha_RVP_Coef	Offline design			Naphtha RVP coefficient from off-line design
Naphtha_90	Gasoline_Prop		TK units DegF	Naphtha ASTM D86 90% point
Naphtha_90_Coef	Offline design			Naphtha ASTM D86 90% point coefficient from the off-line design
K1_Coef	Offline design			K1 coefficient from off-line design
Conv_Factor				Conversion factor to go from volume in user units to TK units (MBPD)

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Mult_Bias				Multiplicative bias for Octane
Mult_Bias_Filt_ Time				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Mult_Bias_Filt_ Shift				Previous filtered Octane multiplicative bias storage
Add_Bias				Additive bias for Octane
Add_Bias_Filt_ Time			Minutes	Filter time for the additive bias
Add_Bias_Filt_ Shift				Previous filtered Octane additive bias storage
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift[ ]				Previous filtered value storage
Etime			Minutes	Application execution time
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..4]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Octane_Raw		Calculated uncorrected octane
Octane		Calculate octane corrected with multiplicative and additive biases
Octane_Vol_TK_ Units	TK units Octane - Mbbl/day	Octane volume
Octane_Vol_User_Units	User units	Octane volume
Riser_Temp_ Factor		Riser temperature factor
Feed_Gravity_ Factor[ ]		Feed gravity factor for each independent feed
Feed_WatK_ Factor[ ]		Feed Watson K factor for each independent feed
Naphtha_90_ Factor		Naphtha ASTM D86 90% point factor
Naphtha_RVP_ Factor		Naphtha RVP factor
Severity_Factor		Severity factor based on K2 coefficient

Output Table		
Output	**Eng. Units**	**Description**
Cat_Add_Factor		Combined catalyst additive factor
Stream_in_Error		No longer used.
Mult_Bias_Filt_ Shift		Previous filtered multiplicative bias storage
Add_Bias_Filt_ Shift		Previous filtered additive bias storage
Filter_Shift[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..4]		Last good value storage array for key output variables

Return Status Table Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The additive bias is a bad value.
2	The multiplicative bias is a bad value.
3	The multiplicative bias is less than or equal to zero.
4	The riser outlet temperature is a bad value.
5	The riser outlet temperature reference is a bad value.
6	The feed flow is a bad value. Feed number ____ is the feed in error.
7	The feed gravity is a bad value. Feed number ____ is the feed in error.
8	The feed gravity is less than or equal to zero. Feed number ____ is the feed in error.
9	The feed gravity reference is a bad value. Feed number ____ is the feed in error.
10	The feed gravity reference is less than or equal to zero. Feed number ____ is the feed in error.
11	The feed Watson K is a bad value. Feed number ____ is the feed in error.
12	The feed Watson K is less than or equal to zero. Feed number ____ is the feed in error.
13	The feed Watson K reference is a bad value. Feed number ____ is the feed in error.
14	The feed Watson K reference is less than or equal to zero. Feed number ____ is the feed in error.
15	The riser temperature coefficient is a bad value.
16	The riser temperature is less than the riser reference temperature.
17	The feed gravity coefficient is a bad value. Feed number ____ is the feed in error.
18	The feed Watson K coefficient is a bad value. Feed number ____ is the feed in error.

Return Status	
Status value	**Description**
19	The feed Watson K value is less than the feed Watson K reference. Feed number ____ is the feed in error.
20	The total feed flow is a miniscule value
21	The Naphtha ASTM D86 90% point is a bad value.
22	The Naphtha ASTM D86 90% point is less than or equal to zero.
23	The Naphtha ASTM D86 90% point coefficient is a bad value.
24	The Naphtha Reid Vapor Pressure (RVP) is a bad value.
25	The Naphtha Reid Vapor Pressure (RVP) is less than or equal to zero.
26	The Naphtha Reid Vapor Pressure (RVP) coefficient is a bad value.
27	The severity is a bad value.
28	The severity is less than or equal to zero.
29	The severity coefficient is a bad value.
30	The catalyst additive is a bad value.
31	The catalyst additive coefficient is a bad value.
32	The K1 coefficient is a bad value.
33	The Naphtha product flow is a bad value.
34	The Naphtha specific gravity is a bad value
35	The Naphtha specific gravity is less than or equal to zero.
36	The conversion factor is a bad value.
37	The conversion factor is miniscule.
38	Errors filtering the multiplicative bias. See associated message from Filter function
39	Errors filtering the additive bias. See associated message from Filter function
40	Errors filtering the raw Octane value. See associated message from Filter function
141 - 146	An error occurred in the Filter_First_Ord_Lag function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 - 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 - 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
Negative errors	Last good value holding is active. The error that is causing the bad value can be determined by looking up the positive of the error value

Examples Table

The following table shows example for the Octane_Pred function. The example assumes no filtering and no last good value holding.

Examples Table				
Parameter Name [ ] indicates array	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Feed_Num	I	1		
Feed_Flow\1	I	448.950012 Mlb/hr		
Feed_Grav\1	I	0.9259 spgr		
Feed_Grav_Ref\1	I	0.9465 spgr		
Feed_Grav_Coef\1	I	-0.0713		
Feed_WatK\1	I	12.035		
Feed_WatK_Ref\1	I	10.000		
Feed_WatK_Coef\1	I	1.4302		
Riser_Temp	I	966.20 Deg F		
Riser_Temp_Ref	I	900.00 Deg F		
Riser_Temp_Coef	I	0.0335		
Severity	I	2.55		
Severity_Coef	I	0.0071		
Cat_Add_Num	I	1		
Cat_Add\1	I	0		
Cat_Add_Coef\1	I	0		
Naphtha_Flow	I	100 Mlb/hr		
Naphtha_Grav	I	0.9259 spgr		
Naphtha_RVP	I	4.0 psig		
Naphtha_RVP_Coef	I	0		
Naphtha_90	I	359.60 Deg F		
Naphtha_90_Coef	I	0		
K1_Coef	I	3.6003		
Conv_Factor	I	1		
Mult_Bias	I	1		
Mult_Bias_Filt_Time	I	0		
Add_Bias	I	0		
Add_Bias_Filt_Time	I	0		
Ret_Status	O	0		
Octane_Raw	O	92.403		
Octane	O	92.403		
Octane_Vol_TK_Units	O	684.11 Octane –Mbbl/day		

Examples Table				
Parameter Name **[ ] indicates array**	**Input / Output**	**Example 1 –**	**Example 2 –**	**Comment**
Octane_Vol_User_Units	O	684.11 Octane – Mbbl/day		
Riser_Temp_Factor	O	1.150816798		
Feed_Gravity_Factor\1	O	0.788812399		
Feed_WatK_Factor\1	O	2.761801004		
Naphtha_90_Factor	O	1		
Naphtha_RVP_Factor	O	1		
Severity_Factor	O	1.006712675		
Cat_Add_Factor	O	1		
Stream_in_Error	O	0		

Section 12 - Calculation Compensation (Calculation_Comp)

12.1 Overview

In This Section

This function calculates the catalyst correction factor and bias for the severity and product prediction functions. The bias is based on the difference between the measured and predicted inputs.

Usage of Calculation_Comp

Calculation_Comp is used to bias and correct the severity prediction and product prediction functions. As such the user is required to create a separate instance of the Calculation_Comp function to correct the predicted severity. In addition to this, a separate instance of Calculation_Comp is required for each product (i.e. one Calculation_Comp per Product_Pred function).

If Calculation_Comp is correcting severity, then it requires the dynamically compensated predicted severity (from Dynamic_Comp) and the measured severity (from Measured_Conv) as input. It then calculates the bias and catalyst correction factor, which are both fed-back to Severity_Pred routine.

If Calculation_Comp is correcting product, then it requires the dynamically compensated predicted product (from Dynamic_Comp) and the measured product (from Product_Prop) as input. It then calculates the bias and catalyst correction factor, which are both fed-back to Product_Pred routine.

12.2 Detail Description

Calculation Algorithm

The following table shows the algorithm used by Calculation_Comp to calculate the catalyst correction factor and biases:

Calculation Algorithm	
Step #	**Description**
1.	Calculate the current catalyst correction factor. Cat_Corr_Fact = Measured_Input / Dyn_Comp_Input
2.	Filter the calculated correction factor. Cat_Corr_Fact = Filter(Cat_Corr_Fact)
3.	Update the catalyst correction factor that has been held since the user last updated it. if (Cat_Corr_Update == 1) Cat_Corr_Fact_Held = Cat_Corr_Fact
4.	Calculate the additive bias types required by other calculations. Bias between the measured input and uncorrected dynamically compensated calculated input Added_Bias = Measured_Input - Dyn_Comp_Input Bias between measured input and dynamically compensated calculated input corrected with the held catalyst correction factor Added_Bias_Corr_Held = Measured_Input - (Dyn_Comp_Input * Cat_Corr_Fact_Held) Bias between measured input and dynamically compensated calculated input corrected with the current catalyst correction factor Added_Bias_Corr = Measured_Input - (Dyn_Comp_Input * Cat_Corr_Fact)
5.	Call Last_Good_Value utility function to provide last good value processing.

Input table

The input table provides a description of the function inputs.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Dyn_Comp_Input	Dynamic_Comp		TK units Mlb/hr for Products. Unitless for Severity	Dynamically compensated uncorrected calculated Input. This input is intended for either the output severity or the product.
Dyn_Comp_Initialized	Dynamic_Comp			Flag indicating whether the associated dynamic compensation function has completed its initialization. 0 = not completed, 1 = completed.

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
Measured_Input	Measured_Conv or Product_Prop		Same as Dyn_Comp_Input	Measured input. This input is intended for either the measured severity or the measured product (Wt%). **Important Note:** When compensating products, do not use the output from CycleOil_Prop and/or Gasoline_Prop as the measured input. Input should be taken from Product_Prop.
Cat_Corr_Update				Flag indicating whether to update Cat_Corr_Fact_Held (0 = no update, 1 = update)
Cat_Corr_Fact_Held				Value of catalyst correction factor that has been held since the last time the Cat_Corr_Update flag was set to update
Filter_Type				Filter to be used (0 = none, 1 = First order lag filter, 2 to 5 not yet supported). Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff_Num				Number of filter coefficients being passed. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Coeff[ ]				Filter coefficients being passed in. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift_Num				Number of past values necessary for filter. Not required if Filter Type = 0. Details of how to use the filter functions can be found in Toolkit Utilities User Manual.
Filter_Shift[ ]				Previous filtered value storage
Etime			Minutes	Application execution time

Input Table				
Input	**Source**	**Only Req'd if**	**Eng. Units**	**Description**
LGV_Num				The number of intervals to hold the last good value. For more details see Utility Function User's Guide. If LGV_Num > 0, this indicates the number of intervals the last good value will be held. If LGV_Num = 0, then last good value processing is turned off. If LGV_Num < 0, then the last good value will be held indefinitely.
LGV_Count				Counter of number of intervals last good value held
LGV_Values[1..3]				Last good value storage array for key output variables
RetNum				The maximum number of simultaneous error messages that can be raised for this function.

Output table The output table provides a description of the function outputs.

Output Table		
Output	**Eng. Units**	**Description**
Ret_Status		Number of errors/warnings. Ret_Status is positive when the highest priority message is an error. Ret_Status is negative when the highest priority message is a warning. Ret_Status is zero, when there are no warnings or errors.
RetStatus[RetNum]		Array of current function return statuses. See Return Status Table below for details.
RetMessage[RetNum]		Array of current Error/Warning Messages.
RetFloat[RetNum]		Array of float arguments for error messages. For internal use only.
RetString[RetNum]		Array of string arguments for error messages. For internal use only.
Add_Bias		Bias between the measured input and uncorrected dynamically compensated calculated input
Add_Bias_Corr_Held		Bias between measured input and dynamically compensated calculated input corrected with the held catalyst correction factor
Add_Bias_Corr		Bias between measured input and dynamically compensated calculated INPUT corrected with current catalyst correction factor
Cat_Corr_Fact		Current value of catalyst correction factor

Output Table		
Output	**Eng. Units**	**Description**
Cat_Corr_Fact_Held		Value of catalyst correction factor that has been held since the last time the Cat_Corr_Update flag was set to update
Filter_Shift[ ]		Previous filtered value storage
LGV_Count		Counter of number of intervals last good value held
LGV_Values[1..3]		Last good value storage array for key output variables

Return Status Table

Shows the error and warning messages associated with the function.

Return Status	
Status value	**Description**
0	No error occurred
1	The measured input is a bad value
2	The dynamic compensation input is a bad value.
3	The dynamic compensation input is miniscule.
4	The catalyst correction update flag must be either 1 or 0.
5	Errors filtering the catalyst correction factor. See associated message from Filter function
141 - 146	An error occurred in the Filter_First_Ord_Lag function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
147 - 154	An error occurred in the Filter_First_Ord_Lag_Deadtime function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference
155 - 160	An error occurred in the Filter function. Details of the error can be found by looking up the resulting status value in the Profit Toolkit Functions Reference