

# FTFitter Object

---

The FTFitter object provides general purpose least-squares fitting of spectral lineshapes and continuum models to FTS spectra, particularly SPIRE, FTS-2, and lab instruments.

## Superclasses

IDL\_Object

## Creation

See [FTFitter::Init](#).

## Properties

Objects of this class have the following properties. See [FTFitter Properties](#) for details on individual properties.

<a href="#">COVAR</a>	<a href="#">HIDE_PEAKS</a>	<a href="#">N_LINES</a>
<a href="#">DEBUG</a>	<a href="#">HIDE_RESIDUAL</a>	<a href="#">QUIET</a>
<a href="#">ERROR</a>	<a href="#">HIDE_SPC</a>	<a href="#">RESIDUAL</a>
<a href="#">FILENAME</a>	<a href="#">INSTRUMENT</a>	<a href="#">RESOLUTION</a>
<a href="#">FIT_RESULT</a>	<a href="#">LINE_PARMS</a>	<a href="#">SPECTRUM</a>
<a href="#">FREQUENCY</a>	<a href="#">LINETYPE</a>	<a href="#">STATUS</a>
<a href="#">FRINGE_PARMS</a>	<a href="#">LINEWIDTH</a>	<a href="#">STRUCT</a>
<a href="#">FRINGE_TYPE</a>	<a href="#">MAX_ITER</a>	<a href="#">THRESHOLD</a>
<a href="#">FTOL</a>	<a href="#">MESSAGE_PROC</a>	<a href="#">WINDOW</a>
<a href="#">HIDE_ERR</a>	<a href="#">MESSAGE_TLB</a>	<a href="#">XTOL</a>
<a href="#">HIDE_FIT</a>	<a href="#">NAMES</a>	<a href="#">XUNITS</a>
<a href="#">HIDE_LINES</a>	<a href="#">N_ITER</a>	<a href="#">YUNITS</a>

In addition, objects of this class inherit the properties of all superclasses of this class.

## Methods

This class has the following methods:

<a href="#">FTFitter::Add_Line</a>	<a href="#">FTFitter::Fit_lines</a>	<a href="#">FTFitter::Print_lines</a>
<a href="#">FTFitter::Apodize</a>	<a href="#">FTFitter::Get_model</a>	<a href="#">FTFitter::Remove_fringes</a>
<a href="#">FTFitter::Calc_Model</a>	<a href="#">FTFitter::GetProperty</a>	<a href="#">FTFitter::SetProperty</a>
<a href="#">FTFitter::Calc_Residual</a>	<a href="#">FTFitter::Init</a>	<a href="#">FTFitter::Status</a>
<a href="#">FTFitter::Cleanup</a>	<a href="#">FTFitter::Load_config</a>	<a href="#">FTFitter::Write_lines_XML</a>
<a href="#">FTFitter::Delete_Line</a>	<a href="#">FTFitter::Load_lines</a>	<a href="#">FTFitter::Write_model</a>
<a href="#">FTFitter::Find_lines</a>	<a href="#">FTFitter::Load_SPC</a>	<a href="#">FTFitter::Write_residual</a>

In addition, this class inherits the methods of its superclasses (if any).

## Documentation Version History

1.0	Dec 2010	Introduced.
1.1	Jul 19 2011	Minor typo corrections.
1.2	Aug 4 2011	Updated documentation
1.3	Aug 25 2011	Added new properties
1.4	Jan 13 2015	Added new properties

## FTFitter::Init

---

The FTFitter::Init function method initializes the FTFitter object.

**Note:** Init methods are special *lifecycle methods*, and as such cannot be called outside the context of object creation. This means that in most cases, you cannot call the Init method directly. There is one exception to this rule: if you write your own subclass of this class, you can call the Init method from within the Init method of the subclass.

### Syntax

*Obj* = OBJ\_NEW('FTFitter' [, PROPERTY=value])

or

*Result* = *Obj*->[FTFitter::]Init([, PROPERTY=value]) (In a lifecycle method only.)

### Return Value

When this method is called indirectly, as part of the call to the OBJ\_NEW function, the return value is an object reference to the newly-created object.

When called directly within a subclass Init method, the return value is 1 if initialization was successful, or zero otherwise.

### Arguments

None

### Keywords

Any property listed under [FTFitter Properties](#) that contains the word “Yes” in the “Init” column of the properties table can be initialized during object creation using this method. To initialize the value of a property, specify the property name as a keyword set equal to the appropriate property value.

All other keywords are passed to the superclass of this object.

## FTFitter::Add\_Line

---

Adds a single line or array of line parameter structures to the end of the line list.

### Syntax

*Obj->*[\[FTFitter::\]](#)*Add\_Line, Line\_Parm, Index*

### Arguments

#### Line\_Parm

The line parameter structure or array of structures to add to the list.

#### Index

The optional index at which to insert the line(s). If omitted, the line is added to the end of the list.

### Keywords

None.

# FTFitter::Apodize

---

Smooths the spectrum loaded into the SPECTRUM property to the given resolution, and apodizes it using the Norton-Beer apodizing function with FWHM of 1.6 to reduce sidelobes even further. The result is stored internally for use in the peak detection algorithm.

## Syntax

*Obj->*[[FTFitter::](#)]Apodize [, FWHM] [, /RESIDUAL]

## Arguments

### FWHM

The resolution to which to smooth the spectrum before applying the apodization.

## Keywords

### RESIDUAL

Set this keyword to apodize the residual spectrum instead of the raw spectrum, in the case where you want to find peaks in the residual spectrum.

## FTFitter::Calc\_Model

---

Calculates the model from the current parameters and updates the plot object. This calls the FTFitter::Iterproc method to replicate the steps used during the fitting loop. This only needs to be called if the FTFitter::Fit\_Lines method hasn't already been called. To just return the model without updating the plot, use the FTFitter::Get\_Model method.

### Syntax

*Obj->[\[FTFitter::\]](#)Calc\_Model*

### Arguments

None

### Keywords

None

## FTFitter::Calc\_Residual

---

Updates the self.residual pointer with the difference between the raw spectrum and the fit model.

### Syntax

*Obj->[\[FTFitter::\]](#)Calc\_Residual*

### Arguments

None

### Keywords

None

## FTFitter::Cleanup

---

The FTFitter::Cleanup procedure method performs all cleanup on the object.

### Syntax

*Obj->Cleanup*

or

*Obj->*[\[FTFitter::\]Cleanup](#)

or

*OBJ\_DESTROY, Obj*

### Arguments

None

### Keywords

None



## FTFitter::Delete\_Line

---

Adds a single line or array of line parameter structures to the end of the line list.

### Syntax

*Obj->*[\[FTFitter::\]](#)*Delete\_Line, Index, [NAME=value], [/ALL]*

### Arguments

#### Index

The index of the line to delete. If the NAME keyword is given, the index is ignored.

### Keywords

#### NAME

Set to a string or array of strings of the names to delete.

#### ALL

Set this keyword to delete all the lines in the list.

## FTFitter::Find\_lines

---

The FTFitter::Find\_lines function method autodetects lines in the residual (loaded spectrum - model) with the linewidth set in the main parameters. Be sure to call FTFitter::Calc\_residual first. The procedure does the following:

- apodize to remove ILS ringing
- highpass filter to remove continuum
- find peaks above threshold
- add lines with given lineshapes

### Syntax

```
Result = Obj->[FTFitter::]Find_lines( [LINESHAPE=string{'Gaussian' | 'Lorentzian' | 'Unresolved' | 'Voigt'}]  
[,/FILTER] [,/ADD] [,MAX_LINES=value] [,CENTRE_DRIFT=value] [,WIDTH_DRIFT=value]  
[,LO_FREQ=value] [,HI_FREQ=value] [,EMISSION] [,/ABSORPTION] )
```

### Return Value

Returns a list of {FTFITTER\_LINE\_PARMS} line parameter structures.

### Arguments

None

### Keywords

#### LINESHAPE

Set to label the found lines as something other than the default Gaussian. Must be a valid lineshape (Gaussian, Lorentzian, Unresolved, Voigt).

#### FILTER

Set to high-pass filter the spectrum. Note that if we are dealing with the residual of a spectrum that has already been filtered, then we don't want to filter again.

#### ADD

Set to load the found lines into the object line list.

#### MAX\_LINES

Set to the maximum number of lines to find.

#### CENTRE\_DRIFT

Set to the fraction +/- that the centre of found lines can vary. If zero, then the centre is unbounded.

**WIDTH\_DRIFT**

Set to the fraction +/- that the width of found lines can vary from the Linewidth parameter. If zero, the width is fixed.

**LO\_FREQ**

The lowest frequency to consider.

**HI\_FREQ**

The highest frequency to consider.

**EMISSION**

Set to find emission lines (default), set to 0 to disable

**ABSORPTION**

Set to find absorption lines, set to 0 to disable (default)

## FTFitter::Fit\_lines

---

Fit the loaded or detected lines using the `blended_ft_lines` routine. The `self.resolution` parameter defines the length of window to apply to the interferogram. This results in Sinc lineshapes with FWHM of  $1.207 * \text{self.resolution}$ .

### Syntax

*Obj->[FTFitter::]Fit\_lines, [/APODIZED] [,LO\_FREQ=value] [,HI\_FREQ=value]*

### Arguments

None

### Keywords

#### APODIZED

Set to use the apodized spectrum instead of the raw spectrum (e.g. when peak finding)

#### LO\_FREQ

The minimum frequency to consider

#### HI\_FREQ

The maximum frequency to consider

## FTFitter::Get\_model

---

The FTFitter::Get\_model function method returns the fit model array.

### Syntax

*Result = Obj->*[FTFitter::](#)*Get\_model()*

### Return Value

Returns the spectrum array generated by the fit.

### Arguments

None

### Keywords

None

## FTFitter::GetProperty

---

The FTFitter::GetProperty procedure method retrieves the value of a property or group of properties for the object.

### Syntax

*Obj->[FTFitter::](#)GetProperty[, PROPERTY=variable]*

### Arguments

None

### Keywords

Any property listed under [FTFitter Properties](#) that contains the word “Yes” in the “Get” column of the properties table can be retrieved using this method. To retrieve the value of a property, specify the property name as a keyword set equal to a named variable that will contain the value of the property.

Any keyword not recognized is passed to this object’s superclass.

## FTFitter::Load\_config

---

Load an XML settings file.

### Syntax

*Obj->*[[FTFitter::](#)]Load\_config, *Filename*

### Arguments

#### Filename

The filename for the XML settings file to load.

### Keywords

None.

## FTFitter::Load\_lines

---

Load line parameters from an XML file.

### Syntax

*Result = Obj->*[FTFitter::](#)*Load\_lines( Filename [, ERR=err] )*

### Return Value

Returns 1 on success, 0 on failure.

### Arguments

#### Filename

The filename for the XML line parameter file to load. The parameter file must be consistent with the `ftfitter_line_parameters.xsd` schema.

### Keywords

ERR

The error flag from the XML parsing object.



## FTFitter::Load\_SPC

---

Load a spectrum from a GRAMS compatible .SPC file.

### Syntax

*Obj->*[[FTFitter::](#)]Load\_SPC, *Filename*

### Arguments

#### Filename

The filename for the .SPC file to load. If Filename is a null string or not provided, a file selection dialog box will appear.

### Keywords

None.

## FTFitter::Print\_lines

---

Print the current line parameters to the console, or to a text file if the filename keyword is set. Note that the object overloads the print operator, so you can also display the line parameters just by passing the object to the Print command: *Print, Obj*

### Syntax

*Obj->*[[FTFitter::](#)]Print\_lines, [/SORT\_AMPLITUDE] [,/SORT\_FREQ] [,/ERRORS] [,FILENAME=string]

### Arguments

None.

### Keywords

#### **SORT\_AMPLITUDE**

Set this keyword to print the lines in order of increasing amplitude.

#### **SORT\_FREQ**

Set this keyword to print the lines in order of increasing centre frequency.

#### **ERRORS**

Set this keyword to include the fit uncertainties in the printout.

#### **FILENAME**

If this keyword is set to a string filename, the output will be written to a text file instead of to the console.

## FTFitter::Remove\_fringes

---

This method removes channel fringes from the spectrum.

**Note:** This method is currently not implemented.

### Syntax

*Obj->*[\[FTFitter::\]Remove\\_fringes](#)

### Arguments

None.

### Keywords

None.

## FTFitter::SetProperty

---

The FTFitter::SetProperty procedure method sets the value of a property or group of properties for the object.

### Syntax

*Obj->*[[FTFitter::](#)]SetProperty[, PROPERTY=value]

### Arguments

None

### Keywords

Any property listed under [FTFitter Properties](#) that contains the word “Yes” in the “Set” column of the properties table can be set using this method. To set the value of a property, specify the property name as a keyword set equal to a the appropriate property value.

## FTFitter::Status

---

Outputs a diagnostic message. If the [message\\_proc](#) property is defined, then the string is sent to the specified procedure. Otherwise, the text is sent to the console as an informational message. If the [message\\_proc](#) procedure is part of a GUI application, then the [message\\_tlb](#) property should be set to the TLB of the application. This TLB identifier will be passed to the [message\\_proc](#) procedure in the TLB keyword.

### Syntax

*Obj->[\[FTFitter::\]Status](#), Text*

### Arguments

Text

The message to send to the status display or console.

### Keywords

None.

## FTFitter::Write\_Lines\_XML

---

The FTFitter::Write\_Lines\_XML procedure method writes the current line parameters to an XML file.

### Syntax

```
Obj->\[FTFitter::\]Write_Lines_XML, Filename [,COMMENT=string] [,/SORT_AMPLITUDE]  
[,/SORT_FREQ]
```

### Arguments

#### Filename

The filename for the XML file to write.

### Keywords

#### COMMENT

A string comment to add to the XML file header.

#### SORT\_AMPLITUDE

Set this keyword to sort the output in order of increasing amplitude.

#### SORT\_FREQ

Set this keyword to sort the output in order of increasing centre frequency.

## FTFitter::Write\_model

---

The FTFitter::Write\_model procedure method writes the current model spectrum to a file.

**Note:** this method is currently not implemented

### Syntax

*Obj->[\[FTFitter::\]Write\\_model](#), Filename*

### Arguments

#### Filename

The filename for the file to write.

### Keywords

None.

## FTFitter::Write\_residual

---

The FTFitter::Write\_model procedure method writes the current residual spectrum to a file.

**Note:** this method is currently not implemented

### Syntax

*Obj->*[[FTFitter::](#)]Write\_residual, Filename

### Arguments

#### Filename

The filename for the file to write.

### Keywords

None.



## FTFitter Properties

---

FTFitter objects have the following properties in addition to properties inherited from any superclasses. Properties with the word “Yes” in the “Get” column of the property table can be retrieved via [FTFitter::GetProperty](#). Properties with the word “Yes” in the “Init” column of the property table can be set via [FTFitter::Init](#). Properties with the word “Yes” in the “Set” column in the property table can be set via [FTFitter::SetProperty](#).

Objects of this class have the following properties:

<a href="#">COVAR</a>	<a href="#">HIDE_PEAKS</a>	<a href="#">RESIDUAL</a>
<a href="#">DEBUG</a>	<a href="#">HIDE_RESIDUAL</a>	<a href="#">RESOLUTION</a>
<a href="#">ERROR</a>	<a href="#">HIDE_SPC</a>	<a href="#">SPECTRUM</a>
<a href="#">FILENAME</a>	<a href="#">INSTRUMENT</a>	<a href="#">STATUS</a>
<a href="#">FIT_RESULT</a>	<a href="#">LINE_PARMS</a>	<a href="#">STRUCT</a>
<a href="#">FREQUENCY</a>	<a href="#">LINETYPE</a>	<a href="#">THRESHOLD</a>
<a href="#">FRINGE_PARMS</a>	<a href="#">LINEWIDTH</a>	<a href="#">WINDOW</a>
<a href="#">FRINGE_TYPE</a>	<a href="#">MAX_ITER</a>	<a href="#">XTOL</a>
<a href="#">FTOL</a>	<a href="#">NAMES</a>	<a href="#">XUNITS</a>
<a href="#">HIDE_ERR</a>	<a href="#">N_ITER</a>	<a href="#">YUNITS</a>
<a href="#">HIDE_FIT</a>	<a href="#">N_LINES</a>	
<a href="#">HIDE_LINES</a>	<a href="#">QUIET</a>	

In addition, objects of this class inherit:

- [IDL\\_Object](#)

### COVAR

The covariance matrix for the set of parameters returned by MPFIT. The matrix is NxN where N is the number of parameters. Note that each line contains 5 parameters, ordered as [Linetype, Amplitude, Center, FWHM, transition rate].

The square root of the diagonal elements gives the formal 1-sigma statistical errors on the parameters IF errors were treated “properly”. To compute the correlation matrix, PCOR, use this:

```
IDL> PCOR = COV * 0
```

```
IDL> FOR i = 0, n-1 DO FOR j = 0, n-1 DO PCOR(i,j) = COV(i,j)/sqrt(COV(i,i)*COV(j,j))
```

If the fitting terminates abnormally, then COVAR points to a scalar with value !VALUES.D\_NAN.

Property Type	Pointer to Double Vector		
<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> No	<b>Registered:</b> No

### DEBUG

An integer value that determines if debugging information will be displayed:

Value	Description
0	No debug information displayed (Default)
1	Display debug messages

<b>Property Type</b>	Integer		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## ERROR

The array of input error values.

<b>Property Type</b>	Double vector		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## FILENAME

The filename string for the spectrum being fit.

<b>Property Type</b>	String		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## FIT\_RESULT

The last reported fitting status from MPFIT.

<b>Property Type</b>	Long		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## FREQUENCY

The array of input frequency abscissa values. Must match the length of the spectrum array.

<b>Property Type</b>	Double vector		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## FRINGE\_PARMS

TBD

## FRINGE\_TYPE

TBD

## FTOL

The convergence tolerance for MPFIT (relative error desired in the sum of squares).

<b>Property Type</b>	Double		
----------------------	--------	--	--

<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No
-----------------	-----------------	------------------	-----------------------

## HIDE\_ERR

A flag to prevent the display of the error spectrum during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## HIDE\_FIT

A flag to prevent the display of the fit spectrum during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## HIDE\_LINES

A flag to prevent the display of the lineshape spectra during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## HIDE\_PEAKS

A flag to prevent the display of the line peak positions during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## HIDE\_RESIDUAL

A flag to prevent the display of the residual spectrum during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## HIDE\_SPC

A flag to prevent the display of the spectrum during plot updates.

<b>Property Type</b>	Boolean		
<b>Get:</b> No	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## INSTRUMENT

The name of the instrument being used.

<b>Property Type</b>	String		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## LINE\_PARMS

An array of line parameter structures to use in the fitting. Getting this array will return the loaded and found line parameters. Setting this array will replace the loaded list of line parameters. The line parameter structure ({FTFITTER\_LINE\_PARMS}) has the following tags:

Field Name	Description																
NAME	Contains the optional name string for the line. This string can be used to identify lines for later analysis.																
VERSION	Version number. Currently 1.0																
LINESHAPE	Contains the name of the lineshape or continuum model. Must be one of the following strings: Gaussian, Lorentzian, Voigt, Unresolved, Planck, Polynomial.																
AMPLITUDE	Contains the first lineshape parameter structure. The purpose of this parameter depends on the lineshape: <table border="1"> <thead> <tr> <th>Lineshape</th><th>Parameter</th></tr> </thead> <tbody> <tr> <td>Polynomial</td><td>Offset</td></tr> <tr> <td>Poly2</td><td>Quartic Coefficient</td></tr> <tr> <td>Gaussian</td><td>Amplitude</td></tr> <tr> <td>Lorentzian</td><td>Amplitude</td></tr> <tr> <td>Voigt</td><td>Scale Factor</td></tr> <tr> <td>Unresolved</td><td>Integrated Area</td></tr> <tr> <td>Planck</td><td>Emissivity</td></tr> </tbody> </table>	Lineshape	Parameter	Polynomial	Offset	Poly2	Quartic Coefficient	Gaussian	Amplitude	Lorentzian	Amplitude	Voigt	Scale Factor	Unresolved	Integrated Area	Planck	Emissivity
Lineshape	Parameter																
Polynomial	Offset																
Poly2	Quartic Coefficient																
Gaussian	Amplitude																
Lorentzian	Amplitude																
Voigt	Scale Factor																
Unresolved	Integrated Area																
Planck	Emissivity																
CENTRE	Contains the second lineshape parameter structure. The purpose of this parameter depends on the lineshape: <table border="1"> <thead> <tr> <th>Lineshape</th><th>Parameter</th></tr> </thead> <tbody> <tr> <td>Polynomial</td><td>Linear Coefficient</td></tr> <tr> <td>Poly2</td><td>Quintic Coefficient</td></tr> <tr> <td>Gaussian</td><td>Centre Frequency</td></tr> <tr> <td>Lorentzian</td><td>Centre Frequency</td></tr> <tr> <td>Voigt</td><td>Centre Frequency</td></tr> <tr> <td>Unresolved</td><td>Centre Frequency</td></tr> <tr> <td>Planck</td><td>Temperature (K)</td></tr> </tbody> </table>	Lineshape	Parameter	Polynomial	Linear Coefficient	Poly2	Quintic Coefficient	Gaussian	Centre Frequency	Lorentzian	Centre Frequency	Voigt	Centre Frequency	Unresolved	Centre Frequency	Planck	Temperature (K)
Lineshape	Parameter																
Polynomial	Linear Coefficient																
Poly2	Quintic Coefficient																
Gaussian	Centre Frequency																
Lorentzian	Centre Frequency																
Voigt	Centre Frequency																
Unresolved	Centre Frequency																
Planck	Temperature (K)																
WIDTH	Contains the third lineshape parameter structure. The purpose of this parameter depends on the lineshape: <table border="1"> <thead> <tr> <th>Lineshape</th><th>Parameter</th></tr> </thead> <tbody> <tr> <td>Polynomial</td><td>Quadratic Coefficient</td></tr> <tr> <td>Poly2</td><td>Sextic Coefficient</td></tr> <tr> <td>Gaussian</td><td>FWHM</td></tr> <tr> <td>Lorentzian</td><td>FWHM</td></tr> <tr> <td>Voigt</td><td>Width</td></tr> </tbody> </table>	Lineshape	Parameter	Polynomial	Quadratic Coefficient	Poly2	Sextic Coefficient	Gaussian	FWHM	Lorentzian	FWHM	Voigt	Width				
Lineshape	Parameter																
Polynomial	Quadratic Coefficient																
Poly2	Sextic Coefficient																
Gaussian	FWHM																
Lorentzian	FWHM																
Voigt	Width																

		Unresolved	N/A
		Planck	N/A
TRANS_RATE	Contains the fourth lineshape parameter structure. The purpose of this parameter depends on the lineshape:		
	<b>Lineshape</b>	<b>Parameter</b>	
	Polynomial	Cubic Coefficient	
	Poly2	Septic Coefficient	
	Gaussian	N/A	
	Lorentzian	N/A	
	Voigt	Transition Rate	
	Unresolved	N/A	
	Planck	N/A	

Each parameter (Amplitude, Centre, Width, Trans\_rate) is a named structure ({FTFITTER\_FIT\_PARMS}) with the following form:

Field Name	Description
VALUE	Contains the value of the parameter.
ERR	Contains the fit uncertainty for the parameter
FIXED	Set to 1 to fix the parameter, set to 0 to allow the parameter to be fit.
LOWER_BOUNDED	Set to 0 for no lower bound on the parameter range. Set to 1 to limit the range to be greater than LOWER.
UPPER_BOUNDED	Set to 0 for no upper bound on the parameter range. Set to 1 to limit the range to be less than UPPER.
LOWER	The lower bound for the value range, if LOWER_BOUNDED is set.
UPPER	The upper bound for the value range, if UPPER_BOUNDED is set.

## LINETYPE

A flag that determines how the automatic peak finder identifies lines. :

Value	Description
0	Find emission lines only
1	Find absorption lines only
2	Find both emission and absorption lines

<b>Property Type</b>	Integer		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## LINEWIDTH

The FWHM to which the spectrum is apodized before peak finding.

<b>Property Type</b>	Floating		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## MAX\_ITER

The maximum number of iterations allowed in the minimization sequence.

<b>Property Type</b>	Long		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## MESSAGE\_PROC

The name of a procedure to handle status messages produced by FTFilter::Status. If not set, then status messages are sent to the console.

<b>Property Type</b>	Floating		
<b>Get:</b> No	<b>Set:</b> No	<b>Init:</b> Yes	<b>Registered:</b> No

## MESSAGE\_TLB

The TLB widget ID to send to the MESSAGE\_PROC procedure. This ID will be passed to the TLB keyword of the procedure.

<b>Property Type</b>	Floating		
<b>Get:</b> No	<b>Set:</b> No	<b>Init:</b> Yes	<b>Registered:</b> No

## NAMES

The array of line names currently loaded into the object.

<b>Property Type</b>	String Array		
<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> No	<b>Registered:</b> No

## N\_ITER

The number iterations actually performed.

<b>Property Type</b>	Long		
----------------------	------	--	--

<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> No	<b>Registered:</b> No
-----------------	----------------	-----------------	-----------------------

## N\_LINES

The number of lines currently loaded into the object.

<b>Property Type</b>	Long		
<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> No	<b>Registered:</b> No

## QUIET

A flag to disable printing of diagnostic messages.

<b>Property Type</b>	Integer		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## RESIDUAL

The residual spectrum array after the fit has been subtracted from the spectrum.

<b>Property Type</b>	Double Vector		
<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> Yes	<b>Registered:</b> No

## RESOLUTION

Set this parameter to the true resolution of the input spectrum, using the relation:  $\text{resolution} = 1/(2 \cdot \text{OPDmax})$ , where OPDmax is the maximum length of the interferogram, not including any zero-padding. This defines the Sinc width of the ILS convolution used for all lines. The Sinc FWHM =  $1.207 \cdot \text{resolution}$ . The width between the first zero-crossings of the Sinc is  $2 \cdot \text{resolution}$ .

<b>Property Type</b>	Double		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## STATUS

The last status message string that was displayed.

<b>Property Type</b>	String		
<b>Get:</b> Yes	<b>Set:</b> No	<b>Init:</b> Yes	<b>Registered:</b> No

## SPECTRUM

The double-precision spectrum array to be fit.

<b>Property Type</b>	Double vector		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## STRUCT

A named structure ({FTFITTER\_PARMS}) containing all the FTFitter parameters. The structure contains the following fields:

Field Name	Description
INSTRUMENT	The string name of the instrument being used.
FILENAME	The filename of the data file being fit.
LINETYPE	The linetype for the peak finder. 0=emission, 1=absorption, 2=both.
THRESHOLD	The threshold for the peak finder, in units of standard deviations from the mean.
LINEWIDTH	The FWHM to which the spectrum is apodized before peak detection.
RESOLUTION	The true resolution of the spectrum. Used to define the ILS width for the convolved lineshapes. See the RESOLUTION parameter documentation for details.
FRINGE_TYPE	TBD
FRINGE_PARMS	TBD
MAX_ITER	Maximum number of iterations in the minimization routine.
XTOL	Relative error tolerance for MPFIT between consecutive iterations.
FTOL	FTOL measures the relative error desired in the sum of squares. Default: 1D-10.
XUNITS	String to use for the X axis units.
YUNITS	String to use for the Y axis units.

## THRESHOLD

The threshold for the peak finder, in units of standard deviations from the mean.

<b>Property Type</b>	Double		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## WINDOW

The double-precision ILS window function array to use for the convolution. The default is a tophat function with width determined by the resolution parameter.

<b>Property Type</b>	Double vector		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## XTOL

The convergence tolerance for MPFIT (relative error desired in the approximate solution. Default: 1D-10.

<b>Property Type</b>	Double
----------------------	--------



<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No
-----------------	-----------------	------------------	-----------------------

## XUNITS

The string to use for the X axis units.

<b>Property Type</b>	String		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No

## YUNITS

The string to use for the Y axis units.

<b>Property Type</b>	String		
<b>Get:</b> Yes	<b>Set:</b> Yes	<b>Init:</b> Yes	<b>Registered:</b> No