* [Forecasting Bandwidth Utilization (Time Series)](http://127.0.0.1:54857/help/topic/com.ibm.spss.modeler.tutorial/clementine/example_broadband_createmodels_container.htm)

# Forecasting with the Time Series Node

An analyst for a national broadband provider is required to produce forecasts of user subscriptions in order to predict utilization of bandwidth. Forecasts are needed for each of the local markets that make up the national subscriber base. You will use time series modeling to produce forecasts for the next three months for a number of local markets. A second example shows how you can convert source data if it is not in the correct format for input to the Time Series node.

These examples use the stream named broadband\_create\_models.str, which references the data file named broadband\_1.sav. These files are available from the Demos folder of any IBM® SPSS® Modeler installation. This can be accessed from the IBM SPSS Modeler program group on the Windows Start menu. The broadband\_create\_models.str file is in the streams folder.

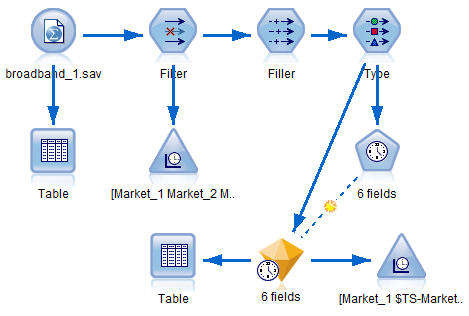
The last example demonstrates how to apply the saved models to an updated dataset in order to extend the forecasts by another three months.

In IBM SPSS Modeler, you can produce multiple time series models in a single operation. The source file you'll be using has time series data for 85 different markets, although for the sake of simplicity you will only model five of these markets, plus the total for all markets.

The broadband\_1.sav data file has monthly usage data for each of 85 local markets. For the purposes of this example, only the first five series will be used; a separate model will be created for each of these five series, plus a total.

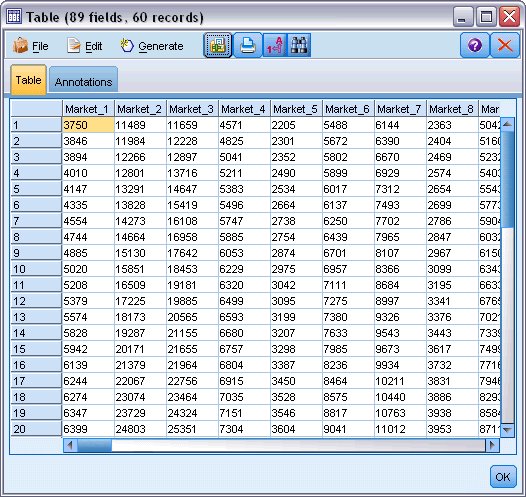
The file also includes a date field that indicates the month and year for each record. This field will be used to label records. The date field reads into IBM SPSS Modeler as a string, but in order to use the field in IBM SPSS Modeler you will convert the storage type to numeric Date format using a Filler node.

*Figure 1. Sample stream to show Time Series modeling*



The Time Series node requires that each series be in a separate column, with a row for each interval. IBM SPSS Modeler provides methods for transforming data to match this format if necessary.

*Figure 2. Monthly subscription data for broadband local markets*



**Creating the Stream**

1. Create a new stream and add a Statistics File source node pointing to *broadband\_1.sav*.
2. Use a Filter node to filter out the *Market\_6* to *Market\_85* fields and the *MONTH\_* and *YEAR\_* fields to simplify the model.

*Tip*: To select multiple adjacent fields in a single operation, click the *Market\_6* field, hold down the left mouse button and drag the mouse down to the *Market\_85* field. Selected fields are highlighted in blue. To add the other fields, hold down the Ctrl key and click the *MONTH\_* and *YEAR\_* fields.

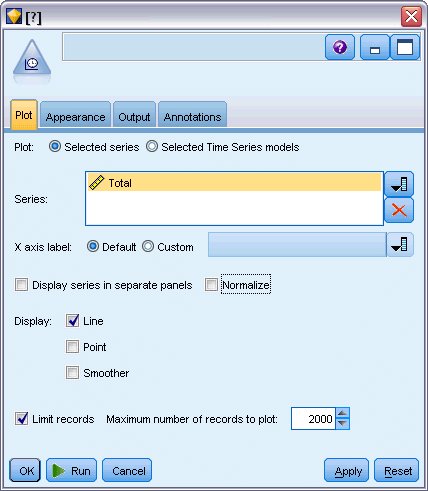
*Figure 1. Simplifying the model*



**Examining the Data**

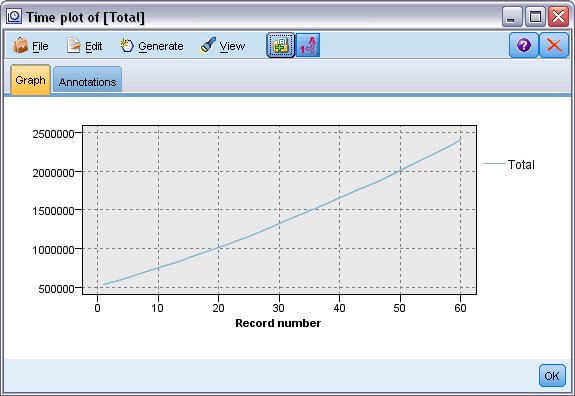
It is always a good idea to have a feel for the nature of your data before building a model. Do the data exhibit seasonal variations? Although the Expert Modeler can automatically find the best seasonal or nonseasonal model for each series, you can often obtain faster results by limiting the search to nonseasonal models when seasonality is not present in your data. Without examining the data for each of the local markets, we can get a rough picture of the presence or absence of seasonality by plotting the total number of subscribers over all five markets.

*Figure 1. Plotting the total number of subscribers*



1. From the Graphs palette, attach a Time Plot node to the Filter node.
2. Add the *Total* field to the Series list.
3. Deselect the **Display series in separate panels** and **Normalize** check boxes.
4. Click **Run**.

*Figure 2. Time plot of Total field*

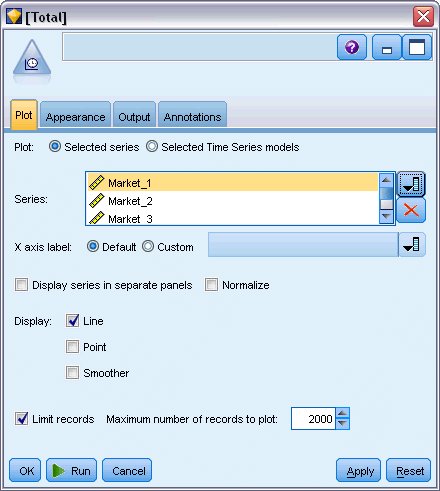


The series exhibits a very smooth upward trend with no hint of seasonal variations. There might be individual series with seasonality, but it appears that seasonality is not a prominent feature of the data in general.

Of course you should inspect each of the series before ruling out seasonal models. You can then separate out series exhibiting seasonality and model them separately.

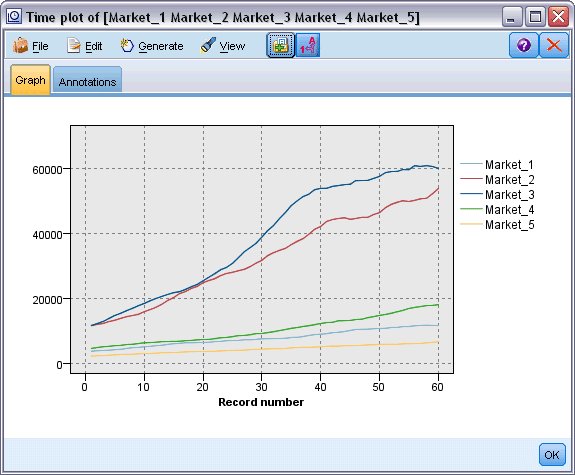
IBM® SPSS® Modeler makes it easy to plot multiple series together.

*Figure 3. Plotting multiple time series*



1. Reopen the Time Plot node.
2. Remove the *Total* field from the Series list (select it and click the red X button).
3. Add the *Market\_1* through *Market\_5* fields to the list.
4. Click **Run**.

*Figure 4. Time plot of multiple fields*



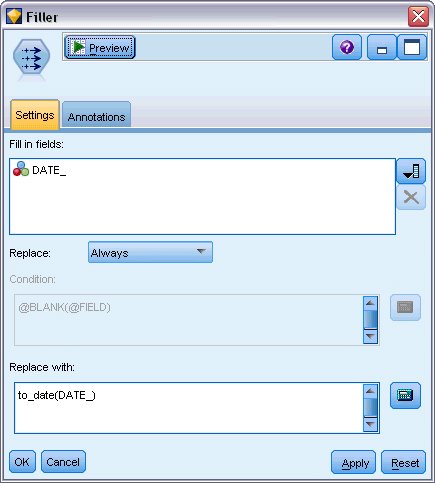
Inspection of each of the markets reveals a steady upward trend in each case. Although some markets are a little more erratic than others, there is no evidence of seasonality to be seen.

**Defining the Dates**

Now you need to change the storage type of the *DATE\_* field to Date format.

1. Attach a Filler node to the Filter node.
2. Open the Filler node and click the field selector button.
3. Select **DATE\_** to add it to **Fill in fields**.
4. Set the **Replace** condition to **Always**.
5. Set the value of **Replace with** to **to\_date(DATE\_)**.

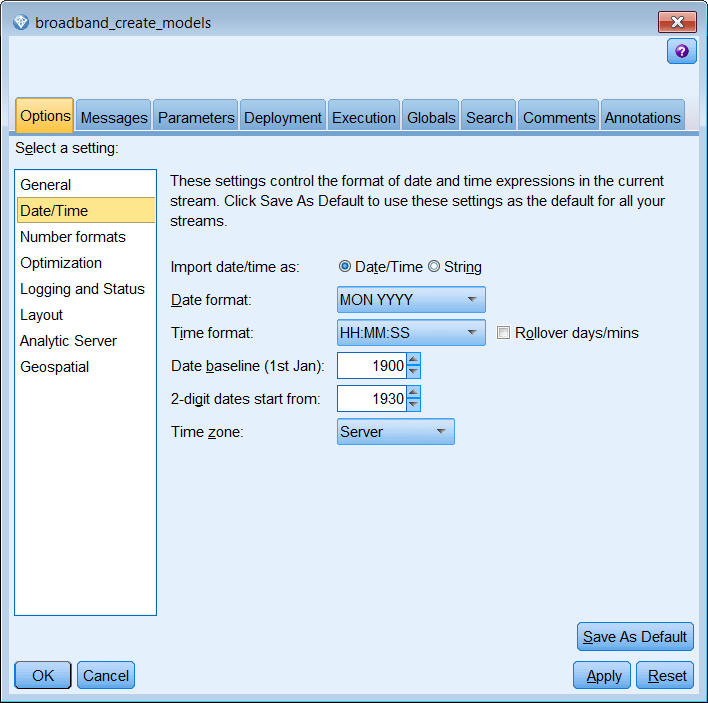
*Figure 1. Setting the date storage type*



Change the default date format to match the format of the Date field. This is necessary for the conversion of the Date field to work as expected.

1. On the menu, choose **Tools > Stream Properties > Options** to display the Stream Options dialog box.
2. Select the **Date/Time** pane and set the default **Date format** to **MON YYYY** .

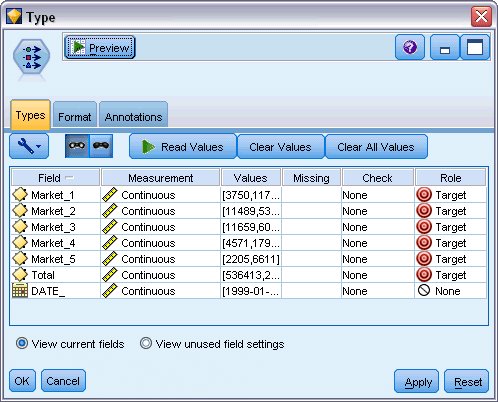
*Figure 2. Setting the date format*



**Defining the Targets**

1. Add a Type node and set the role to **None** for the *DATE\_*field. Set the role to **Target** for all others (the *Market\_n* fields plus the *Total* field).
2. Click the **Read Values** button to populate the Values column.

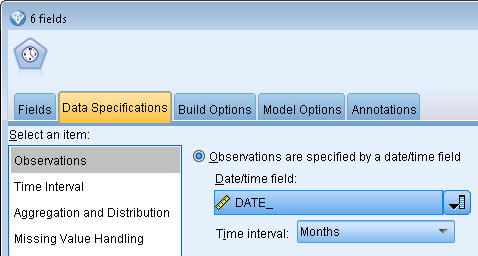
*Figure 1. Setting the role for multiple fields*



**Setting the Time Intervals**

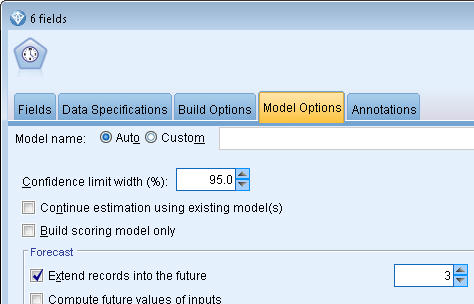
1. From the Modeling palette, add a Time Series node to the stream and attach it to the Type node.
2. On the Data Specifications tab, in the Observations pane, select DATE\_ as the **Date/time field**.
3. Select Months as the **Time interval**.

*Figure 1. Setting the time interval*



1. On the Model Options tab, select the **Extend records into the future** check box.
2. Set the value to **3**.

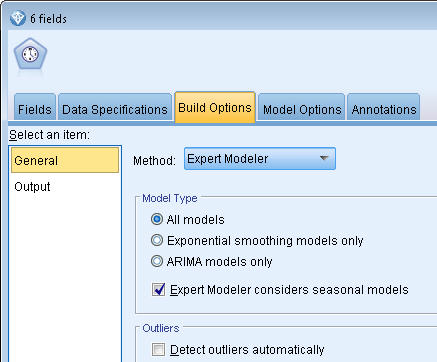
*Figure 2. Setting the forecast period*



**Creating the Model**

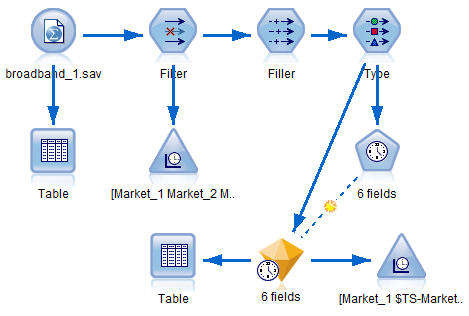
1. On the Time Series node, choose the Fields tab. In the **Fields** list, select all 5 of the markets and copy them to both the **Targets** and **Candidate inputs** lists. In addition, select and copy the Total field to the **Targets** list.
2. Choose the Build Options tab and, on the General pane, ensure the Expert Modeler **Method** is selected using all default settings. Doing so enables the Expert Modeler to decide the most appropriate model to use for each time series. Click **Run**.

*Figure 1. Choosing the Expert Modeler for Time Series*



1. Attach the Time Series model nugget to the Time Series node.
2. Attach a Table node to the Time Series model nugget and click **Run**.

*Figure 2. Sample stream to show Time Series modeling*



There are now three new rows (61 through 63) appended to the original data. These are the rows for the forecast period, in this case January to March 2004.

Several new columns are also present now; the *$TS-* columns are added by the Time Series node. The columns indicate the following for each row (that is, for each interval in the time series data):

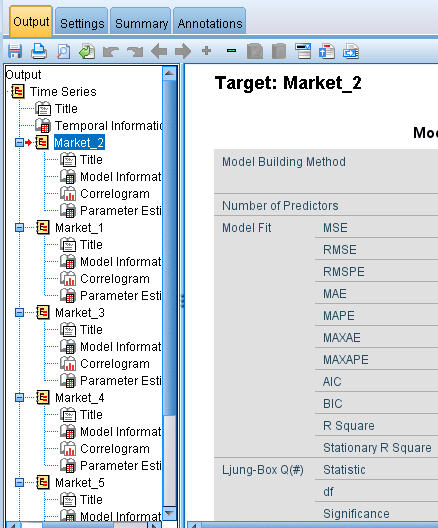
| **Column** | **Description** |
| --- | --- |
| $TS-*colname* | The generated model data for each column of the original data. |
| $TSLCI-*colname* | The lower confidence interval value for each column of the generated model data. |
| $TSUCI-*colname* | The upper confidence interval value for each column of the generated model data. |
| $TS-Total | The total of the $TS-*colname* values for this row. |
| $TSLCI-Total | The total of the $TSLCI-*colname* values for this row. |
| $TSUCI-Total | The total of the $TSUCI-*colname* values for this row. |

The most significant columns for the forecast operation are the *$TS-Market\_n*, *$TSLCI-Market\_n*, and *$TSUCI-Market\_n* columns. In particular, these columns in rows 61 through 63 contain the user subscription forecast data and confidence intervals for each of the local markets.

**Examining the Model**

1. Double-click the Time Series model nugget, and select the Output tab to display data about the models generated for each of the markets.

*Figure 1. Time Series models generated for the markets*



In the left Output column, select the **Model Information** for any of the Markets. The **Number of Predictors** line shows how many fields were used as predictors for each target; in this case, none.

The remaining lines in the **Model Information** tables show various goodness-of-fit measures for each model. The **Stationary R Square** value provides an estimate of the proportion of the total variation in the series that is explained by the model. The higher the value (to a maximum of 1.0), the better the fit of the model.

The **Q(#) Statistic**, **df**, and **Significance** lines relate to the Ljung-Box statistic, a test of the randomness of the residual errors in the model; the more random the errors, the better the model is likely to be. **Q(#)** is the Ljung-Box statistic itself, while **df** (degrees of freedom) indicates the number of model parameters that are free to vary when estimating a particular target.

The **Significance** line gives the significance value of the Ljung-Box statistic, providing another indication of whether the model is correctly specified. A significance value less than 0.05 indicates that the residual errors are not random, implying that there is structure in the observed series that is not accounted for by the model.

Taking both the **Stationary R Square** and **Significance** values into account, the models that the Expert Modeler has chosen for *Market\_3*, and *Market\_4* are quite acceptable. The **Significance** values for *Market\_1*, *Market\_2*, and *Market\_5* are all less than 0.05, indicating that some experimentation with better-fitting models for these markets might be necessary.

The display shows a number of additional goodness-of-fit measures. The **R Square** value gives an estimation of the total variation in the time series that can be explained by the model. As the maximum value for this statistic is 1.0, our models are fine in this respect.

**RMSE** is the root mean square error, a measure of how much the actual values of a series differ from the values predicted by the model, and is expressed in the same units as those used for the series itself. As this is a measurement of an error, we want this value to be as low as possible. At first sight it appears that the models for *Market\_2* and *Market\_3*, while still acceptable according to the statistics we have seen so far, are less successful than those for the other three markets.

These additional goodness-of-fit measures include the mean absolute percentage errors (**MAPE**) and its maximum value (**MAXAPE**). Absolute percentage error is a measure of how much a target series varies from its model-predicted level, expressed as a percentage value. By examining the mean and maximum across all models, you can get an indication of the uncertainty in your predictions.

The MAPE value shows that all models display a mean uncertainty of around 1%, which is very low. The MAXAPE value displays the maximum absolute percentage error and is useful for imagining a worst-case scenario for your forecasts. It shows that the largest percentage error for most of the models falls in the range of roughly 1.8 to 3.7%, again a very low set of figures, with only *Market\_4* being higher at nearer 7%.

The **MAE** (mean absolute error) value shows the mean of the absolute values of the forecast errors. Like the RMSE value, this is expressed in the same units as those used for the series itself. **MAXAE** shows the largest forecast error in the same units and indicates worst-case scenario for the forecasts.

Interesting though these absolute values are, it is the values of the percentage errors (MAPE and MAXAPE) that are more useful in this case, as the target series represent subscriber numbers for markets of varying sizes.

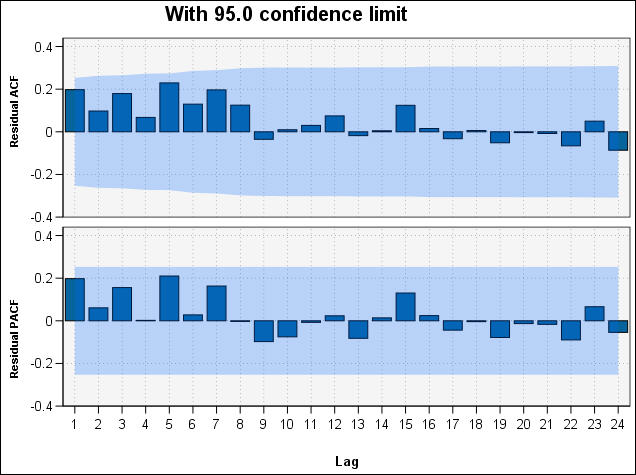
Do the MAPE and MAXAPE values represent an acceptable amount of uncertainty with the models? They are certainly very low. This is a situation in which business sense comes into play, because acceptable risk will change from problem to problem. We'll assume that the goodness-of-fit statistics fall within acceptable bounds and go on to look at the residual errors.

Examining the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the model residuals provides more quantitative insight into the models than simply viewing goodness-of-fit statistics.

A well-specified time series model will capture all of the nonrandom variation, including seasonality, trend, and cyclic and other factors that are important. If this is the case, any error should not be correlated with itself (autocorrelated) over time. A significant structure in either of the autocorrelation functions would imply that the underlying model is incomplete.

1. For the fourth market, in the left column, click **Correlogram** to display the values of the autocorrelation function (ACF) and partial autocorrelation function (PACF) for the residual errors in the model.

*Figure 2. ACF and PACF values for the fourth market*



In these plots, the original values of the error variable have been lagged by up to 24 time periods and compared with the original value to see if there is any correlation over time. For the model to be acceptable, none of the bars in the upper (ACF) plot should extend outside the shaded area, in either a positive (up) or negative (down) direction.

Should this occur, you would need to check the lower (PACF) plot to see whether the structure is confirmed there. The PACF plot looks at correlations after controlling for the series values at the intervening time points.

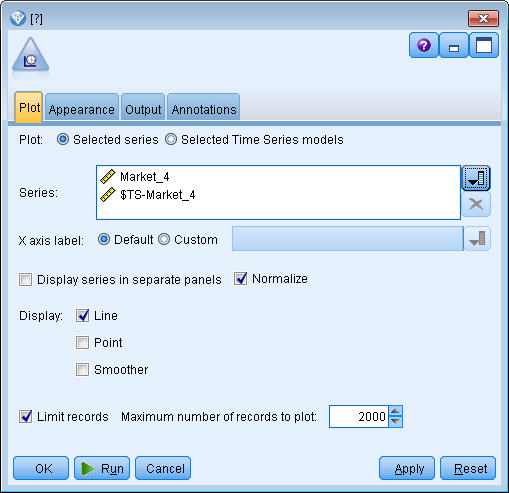
The values for *Market\_4* are all within the shaded area, so we can continue and check the values for the other markets.

1. Click the **Correlogram** for each of the other markets and the totals.

The values for the other markets all show some values outside the shaded area, confirming what we suspected earlier from their **Significance** values. We'll need to experiment with some different models for those markets at some point to see if we can get a better fit, but for the rest of this example, we'll concentrate on what else we can learn from the *Market\_4* model.

1. From the Graphs palette, attach a Time Plot node to the Time Series model nugget.
2. On the Plot tab, clear the **Display series in separate panels** check box.
3. At the **Series** list, click the field selector button, select the *Market\_4* and *$TS-Market\_4* fields, and click **OK** to add them to the list.
4. Click **Run** to display a line graph of the actual and forecast data for the first of the local markets.

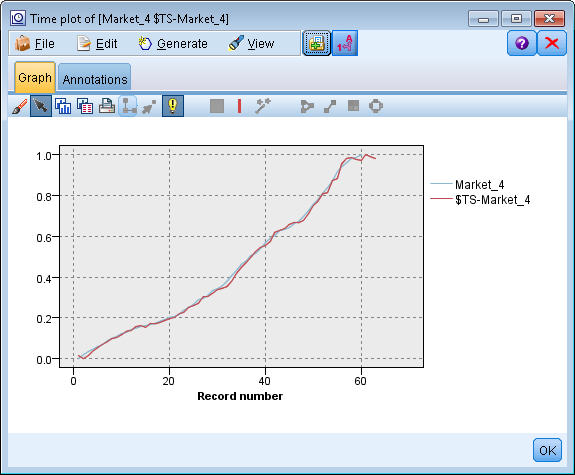
*Figure 3. Selecting the fields to plot*



Notice how the forecast (*$TS-Market\_4*) line extends past the end of the actual data. You now have a forecast of expected demand for the next three months in this market.

The lines for actual and forecast data over the entire time series are very close together on the graph, indicating that this is a reliable model for this particular time series.

*Figure 4. Time Plot of actual and forecast data for Market\_4*



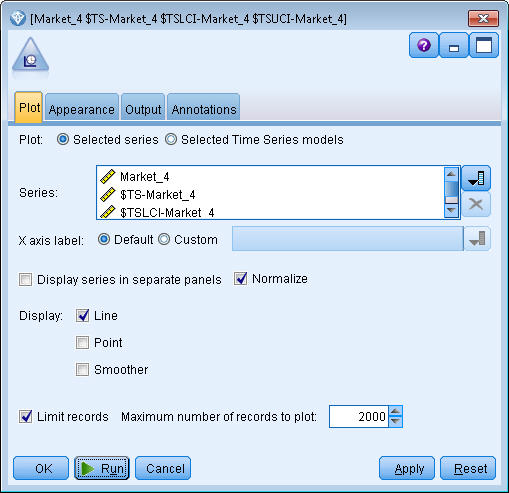
Save the model in a file for use in a future example:

1. Click **OK** to close the current graph.
2. Open the Time Series model nugget.
3. Choose **File > Save Node** and specify the file location.
4. Click **Save**.

You have a reliable model for this particular market, but what margin of error does the forecast have? You can get an indication of this by examining the confidence interval.

1. Double-click the last Time Plot node in the stream (the one labeled **Market\_4 $TS-Market\_4**) to open its dialog box again.
2. Click the field selector button and add the *$TSLCI-Market\_4* and *$TSUCI-Market\_4* fields to the **Series** list.
3. Click **Run**.

*Figure 5. Adding more fields to plot*

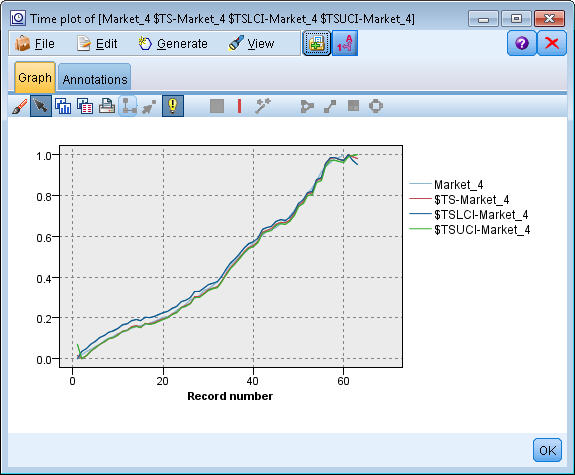


Now you have the same graph as before, but with the upper (*$TSUCI*) and lower (*$TSLCI*) limits of the confidence interval added.

Notice how the boundaries of the confidence interval diverge over the forecast period, indicating increasing uncertainty as you forecast further into the future.

However, as each time period goes by, you will have another (in this case) month's worth of actual usage data on which to base your forecast. You can read the new data into the stream and reapply your model now that you know it is reliable. See the topic [Reapplying a Time Series Model](http://127.0.0.1:54857/help/topic/com.ibm.spss.modeler.tutorial/clementine/example_broadband_applymodels.htm#example_broadband_applymodels) for more information.

*Figure 6. Time Plot with confidence interval added*



# Summary

You have learned how to use the Expert Modeler to produce forecasts for multiple time series, and you have saved the resulting models to an external file.

In the next example, you will see how to transform nonstandard time series data into a format suitable for input to a Time Series node.