About Previewer 2.0

What is Previewer?

Previewer is a software tool which produces demonstration data for presentation in an OLAP (On-line Analytical Processing) environment. It was originally designed to fabricate OLAP cubes based solely on business knowledge. Recent versions of Previewer however can also re-engineer existing OLAP cubes and/or import OLAP data from production sources. Previewer is able to generate cubes in many formats including CSV, Star Schema, and XML.

With Previewer's easy-to-use user interface, a user can quickly and easily define what a cube should look like in terms of dimensions, measures and overall percentages at each level in the dimension hierarchy. Previewer uses the resulting model to generate data points which can then be converted to OLAP cubes. Cubes created by Previewer are produced with random point generation techniques that constrain the generated points to fit into user-specified percentages. In many cases, cubes created with Previewer are virtually indistinguishable from cubes based on actual production data.

What is OLAP?

OLAP refers to a class of Business Intelligence (BI) products, which use cubes to organize data about a business, normally extracted from or resident in a relational database, for analysis and presentation. The cube is a simple model of data organization.

Those fields in a database whose contents appear in a cube are divided into two classes, dimensions and measures. A dimension is a field like "District Office", "Month sold", "Salesperson", or "Product type". It serves to label the record in which it is found. The possible values in that field are known as categories.

A measure is a field like "Total sales", "Expenditures", or "Units sold". It contains a numeric quantity, and one for which totals and subtotals are meaningful. Usually, you will also be allowed to treat a field containing a percentage as a measure, and replace totals with weighted averages.

In addition to actual values of the field specified as a dimension, other values are also accepted as categories. The categories for a dimension form a tree, and only the lowest-level nodes of that tree are the actual field values. Also, each category has a full name as well as an internal code. Thus, a cube contains summarized information that would be stored in multiple tables in a normalized, non-OLAP production database.

How Does Previewer Work?

Whereas someone uses OLAP to enhance their knowledge of a business, someone can use Previewer and his knowledge of the business to create fabricated OLAP cubes that look realistic.

The principle behind Previewer is that you set up the structure of each dimension and measure and then you enter the relative weightings at the various levels in the dimension hierarchy. This weighting information, called "percentages", is used by Previewer in the actual generation of cubes. By specifying the percentages you can influence how the cube will be generated. The only rule is that at each level the total weighting has to add up to 100%. For example, the years dimension, 100%, might be composed of the following: 1999 (25%), 2000 (25%), and 2001 (50%). You could further split each year into quarters with their own respective percentages reflecting the overall trends of the business model.

You also specify for each measure a total value. For example, the Sales measure might be $2,545,000 and the Cost measure might be $1,914,000. Previewer uses these amounts to randomly generates points that are constrained by the percentage values associated with each level in the hierarchy.

What Can Previewer be Used For?

Previewer can be used to demonstrate the operation of OLAP technology on data customized to an individual customer's operation without the need for the lengthy preparations to connect to actual live databases. This provides pre-sales reps of OLAP products the ability to quickly show the technology based solely on prior knowledge of the business.

Another use of Previewer is in creating industry vertical demos. OLAP portals and sales kits based on Previewer cubes or star-schemas can be constructed efficiently and without the problems connected with the sourcing of proprietary data. Previewer also has the ability to reverse engineer cubes so that existing OLAP data can be leveraged into the sales cycle.

In many situations Previewer can also be used by end-users of OLAP to prototype new cubes or adjust existing cubes before resources are expended in applying the changes in the source data mart or data warehouse.

Currently Supported Features of Previewer

The following are features currently supported by Previewer version 2.0:

- Windows interface for maintaining Previewer models.
- Support for maintaining the Dimensions/Measures hierarchy.
- Percentage bars at each dimension level for each measure.
- Independent support for positive and negative percentage values.
- Scaling of percentage bars when handing very small values.
- Percentage bar configurators for quickly setting common patterns.
- Percentage bar locking for fine tuning of percentage values.
- Ability to control irregularity at the cube and dimension levels.
- Support for independent and dependant measures.
- Ability to import dimensions from other cubes.
- Ability to reverse engineer existing cubes.
- Facility for creating star schemas in any ODBC database.
- Import of dimensions from existing cubes.

- Import of dimension and category information from CSV file

Installing the Previewer User Interface

Before You Install Previewer

Previewer requires the following minimum configuration:

Operating System:	Windows 95
Ram:	32 MB
Disk Space for Install:	40 MB
Disk Space for Run time:	20 MB
Video:	640 x 480 Color (8 bits)
Database Connectivity:	ODBC drivers for star schema support

The following is recommended:

Operating System:	Windows NT or 2000
Ram:	128 MB
Video:	1024 x 768 Color (16 bits)

How to Install Previewer

The full installation of Previewer is a two-step process and involves the installation of the Previewer user interface followed by the installation and registration of the Previewer point generation engine. It is possible to install just the Previewer user interface. This will allow you to make full use of Previewer's OLAP modelling capabilities but to create actual cubes or star-schemas you will need to complete the installation and registration of the point generation engine.

Dowloading and Installing Previewer

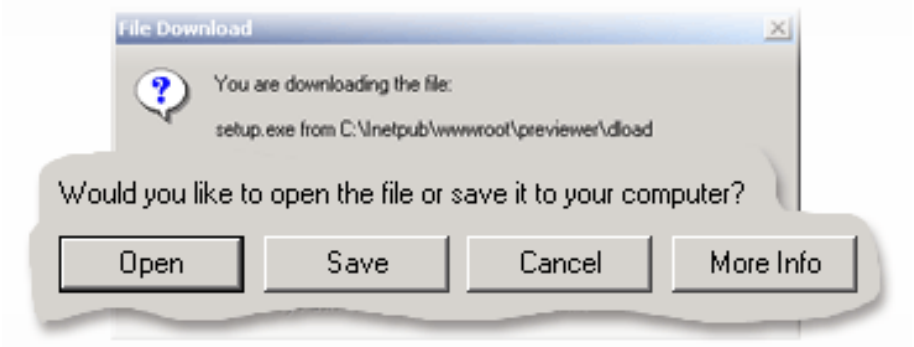
The Previewer Install files can be downloaded from the Seward Consulting web site at:

http://www.sewardconsulting.com/previewer/

...click on download button under download section and click on Previewer 2.0

You can 'Save or Run' the previewer file from the Previewer download page. Click on the save or run button.

You will be prompted to save or open the file.



Open: To download and run this setup click the Open button. The InstallShield Wizard will then guide you through the setup.

Save: To save this setup to disk for installation at another time:

- Click the Save button.
- Save the file named "setup.exe" to your hard drive.
- When you're ready to install, locate setup.exe on your hard drive and double-click the file to begin the installation.
- The InstallShield Wizard will then guide you through the setup.

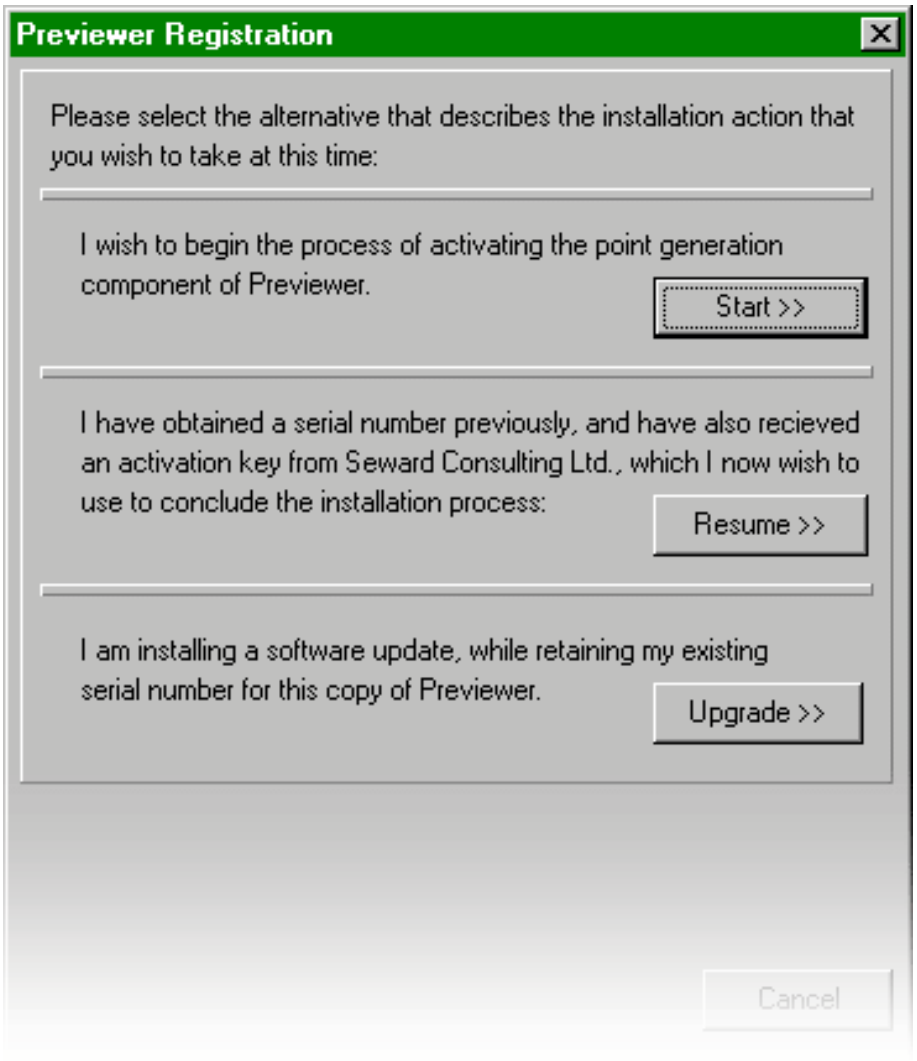
At this point the Previewer User Interface will have been installed on your computer. The user interface will allow you to make full use of Previewer's OLAP modelling capabilities but to create actual cubes or star-schemas you will need to complete the installation and registration of the point generation engine.The procedure for doing this is described in the next section titled Installation (Engine).

Installing the Previewer Engine

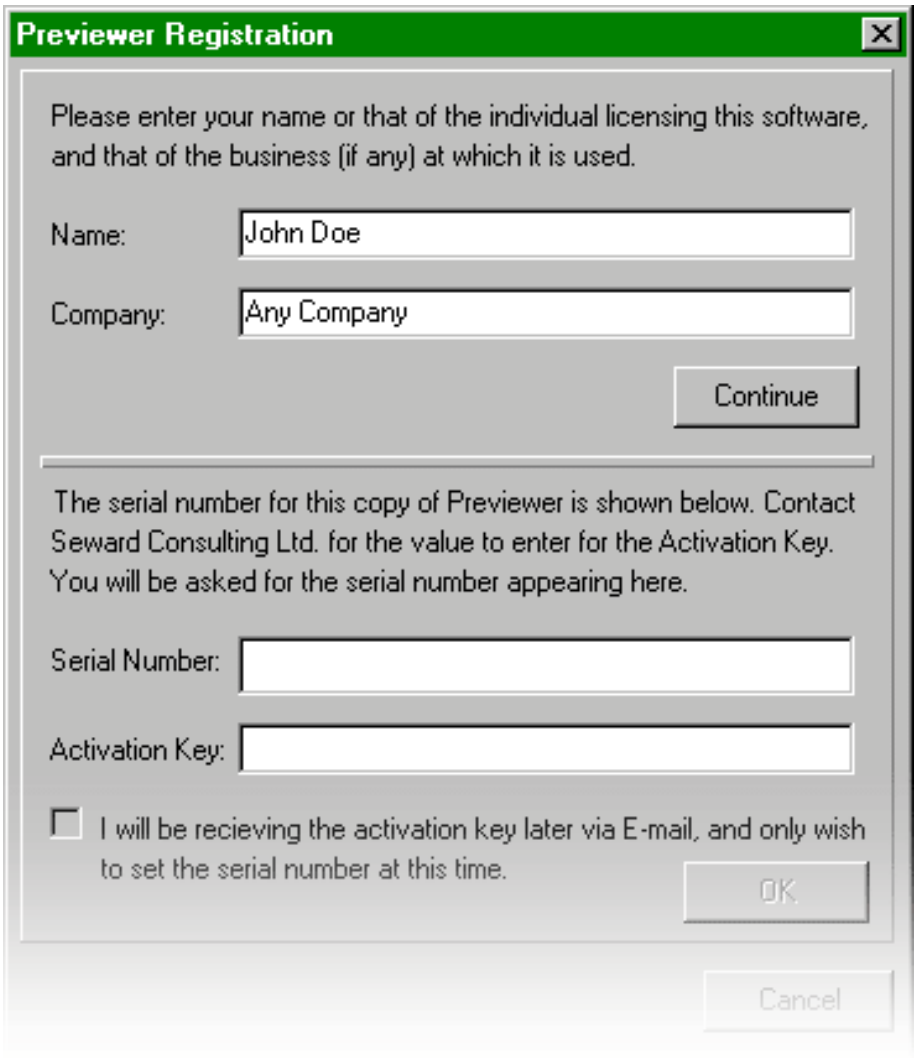
Registering Perviewer

When you have installed Previewer, the Previewer user interface is available to you for designing models. To actually generate data points however to produce cubes or star schemas, a further step is required which involves registering the Previewer engine.

Select the "Install Engine" option from the File menu. The following screen will be displayed:



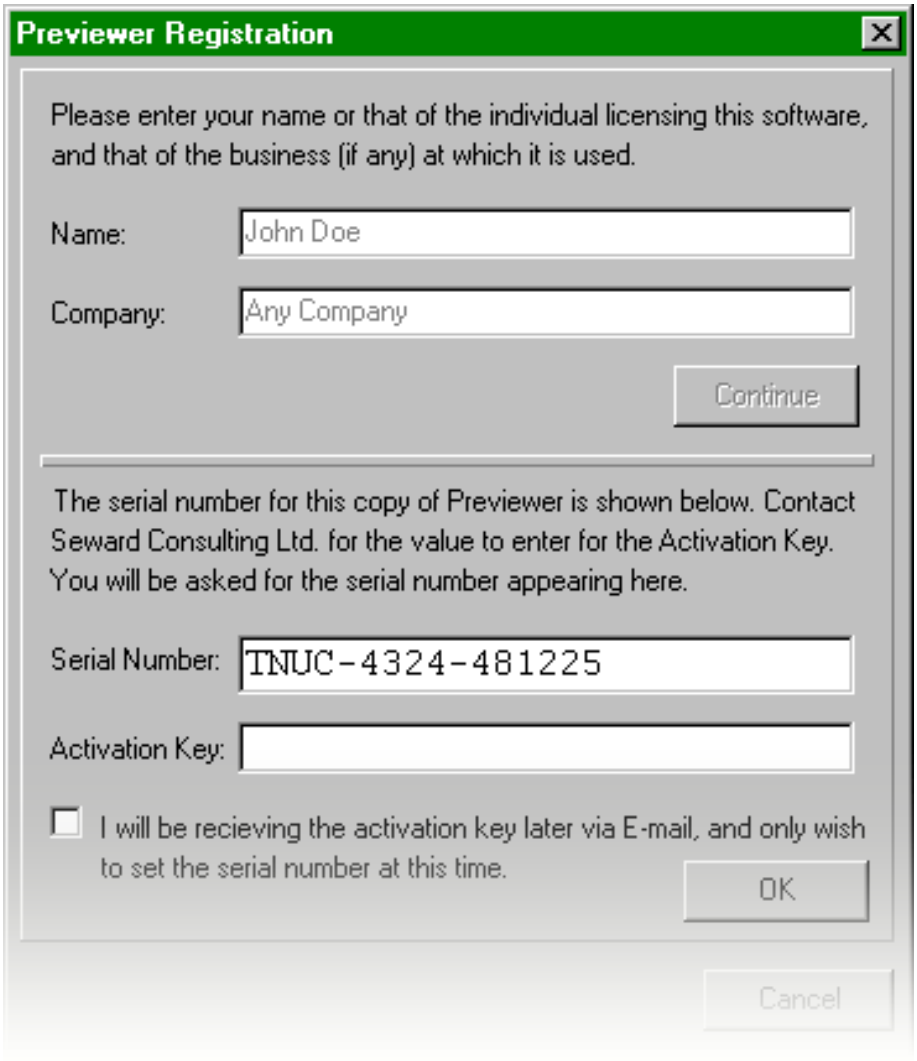
This registration dialogue is designed to handle several situations involving the registration of the Previewer engine. For now, let's begin by describing how to install Previewer for the first time. To do this, press the Start button. The dialogue box will take on the following appearance:



The top half of the panel is enabled when this display first appears.
As shown in the illustration, you can enter your name and that of your company.
Once that is done, press the Continue button to proceed to the next step.

The result will be a screen that will look like the following:

At this point, you have two choices available for continuing the installation. You can telephone Seward Consulting Ltd., and obtain an activation key immediately, which you can then enter in the box below the serial number. Or,



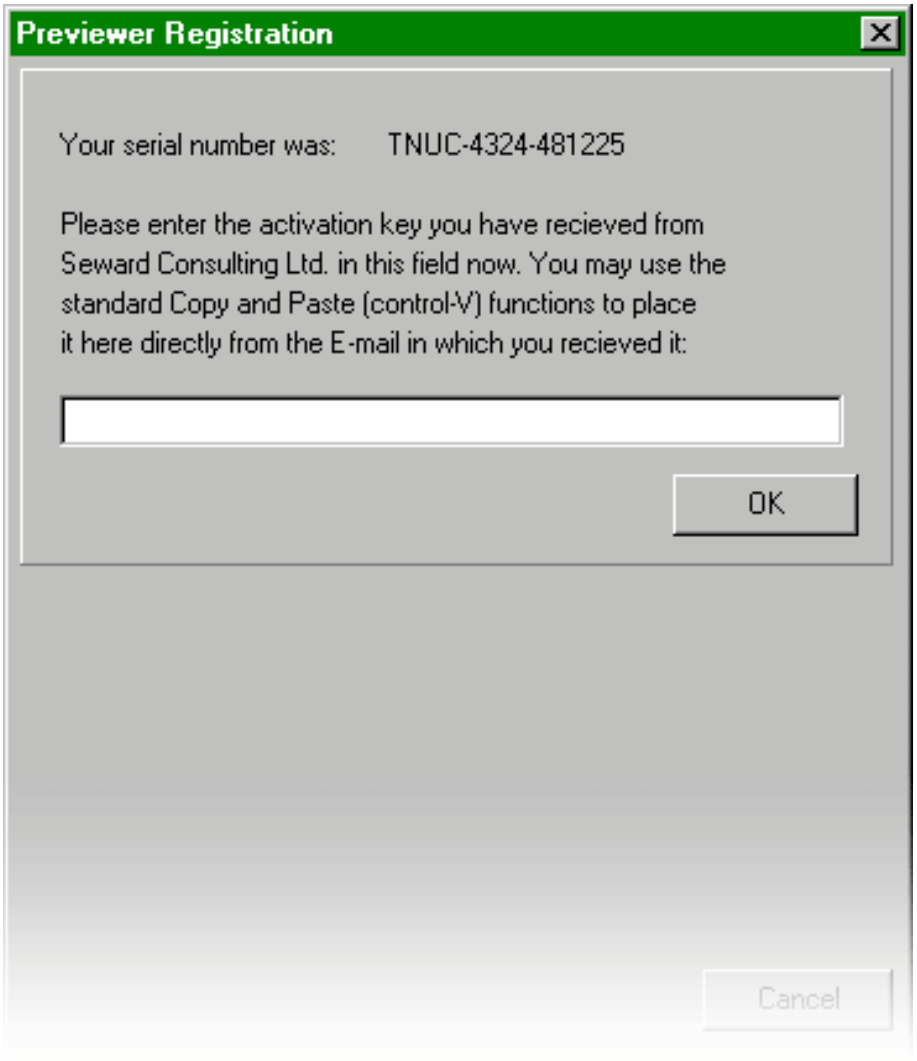
you can check the check box to indicate you will enter the activation key later. (Pressing Cancel instead of pressing OK will mean that installation does not proceed. In this case the serial number shown will irrelevant and you will have to start over again and a different serial number will be generated.)

In either case, you will need to inform Seward Consulting Ltd. of the serial number the program has displayed; this will allow us to provide you with the activation key you require, which must match your serial number.

In the second case, you can contact Seward Consulting by e-mail. After checking the box and pressing OK, you can exit from Previewer and even turn off or reset your computer, and then continue the installation later. This is the only circumstance in which a serial number is retained if its corresponding activation key has not been entered.

When you later receive the activation key from Seward Consulting Ltd., and run Previewer again, you must then select the Resume option instead of the Start option on the initial installation screen in order to use the previously generated serial number for which you have received the activation key. Pressing the Start button will lead to the generation of a completely different serial number, for which the activation key you have received will not work.

Pressing the Resume button leads to the dialogue box taking this form:



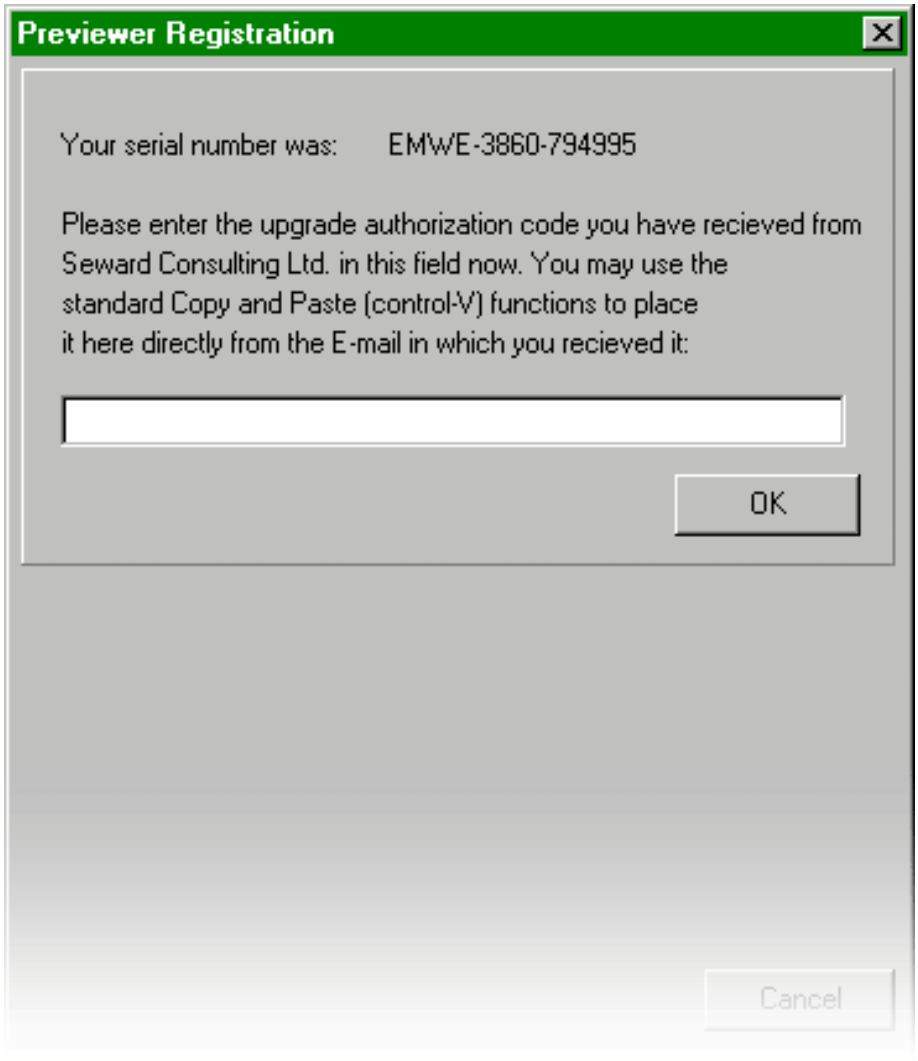
This provides a field where your activation key may be entered.

As noted, this only works when you have previously specified that you would provide the activation key after a delay. When Previewer is fully installed and you wish to modify your licensing options for the Previewer engine or to install an updated version of the Previewer software, select the third option, Upgrade, on the initial installation screen.

This will give you the following screen:

This is similar to the one for delayed entry of the activation key during installation.

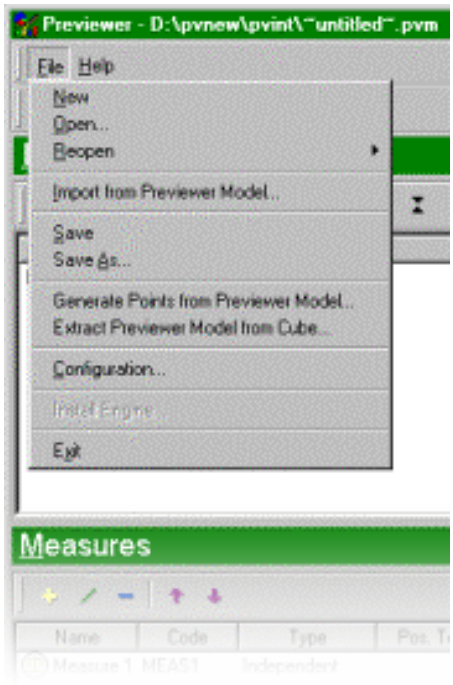
In this case, however, the upgrade



installation code you enter only associates your license information with your copy (new or existing) of the Previewer engine.

Previewer must still have been fully installed with the normal activation key for the serial number of the installation to function.

This step is required either when there has been a change in your licensing information, or when the Previewer engine is updated. Your upgrade authorization code will not change for a given serial number if there has been no change in your licensing information, but it is completely different from your activation key.



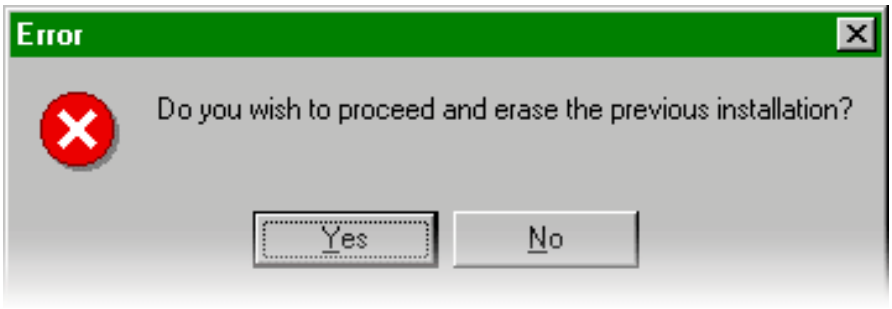
After you finish installing the serial number that enables the point generation function of Previewer, the next time you run Previewer, the File menu of the Previewer interface will offer you the additional options "Generate Points from Previewer Model" and "Extract Previewer Model from Cube", as shown here.

Overwriting an existing installation

In the event that it becomes necessary to install the point generation capability again with a new serial number, if the "Install Engine" option is disabled, as shown in the illustration on the previous page, you can manually initiate the installation process directly by running the program remove.exe in the directory containing Previewer. (At the present time, this option is not normally disabled after the engine is installed, so it is important not to select this option once the engine is successfully installed by mistake).

The files remove.dat and remove.exe are required for installing the point generation ability of Previewer, but are not used afterwards during normal operation. If you have removed them, therefore, you would first re-install Previewer itself from the distribution files. After this, however, the "Install Engine" option may still be disabled, as the initial installation and serial number installation are independent. But the files remove.exe and remove.dat will definitely be present in your chosen installation directory once again.

When you run the installation program, it is possible that you will see a warning message like this:



...if you are certain that your copy of Previewer is not able to generate points for you with the existing serial number, you can click on the button marked "Yes" at this point and proceed.

Also, of course, if you are installing a software update or changing licensing information, you should ignore this message and proceed, since you will not be altering your serial number, but it is important to remember to choose the third option, Upgrade, on the next screen.

Interface

Using Previewer

Previewer is a program developed by Seward Consulting Ltd. that generates files that can be read by Transformer, one of the tools in the suite of OLAP products offered by COGNOS. At present, .def and .dat files are generated, because they allow a complete model to be transmitted to Transformer without having to enter some parts of the model by hand within Transformer as well.

Previewer now also allows the generation of SQL files to produce databases of the star schema type as well, and it can automatically run a copy of Transformer on your computer to generate the .mdc cube directly.

Power Play is the COGNOS product that allows a manager to effectively view corporate data. To use Power Play, it is necessary to convert that data to the form it understands. And Transformer is the tool which performs that conversion.

Previewer, based on percentages entered by the user, will generate fictitious corporate data which can be immediately read by Transformer, allowing prospective Power Play users to see how their corporation's data might look with Power Play, and what they can do with it.

In Previewer, you are provided with an interface that allows you to define the following:

- Measures, and their global totals.
- Dimensions.
- The categories in each dimension, and their hierarchical organization.
- For each category, and each measure, the percentage that category receives of that measure. Percentages are defined locally, within their own node of the hierarchy.

Using this information, a .DEF file and a .DAT file are generated.

The .DEF file lists the measures, dimensions, and categories used in the model. Their names, and codes (which must not contain embedded spaces) are given.

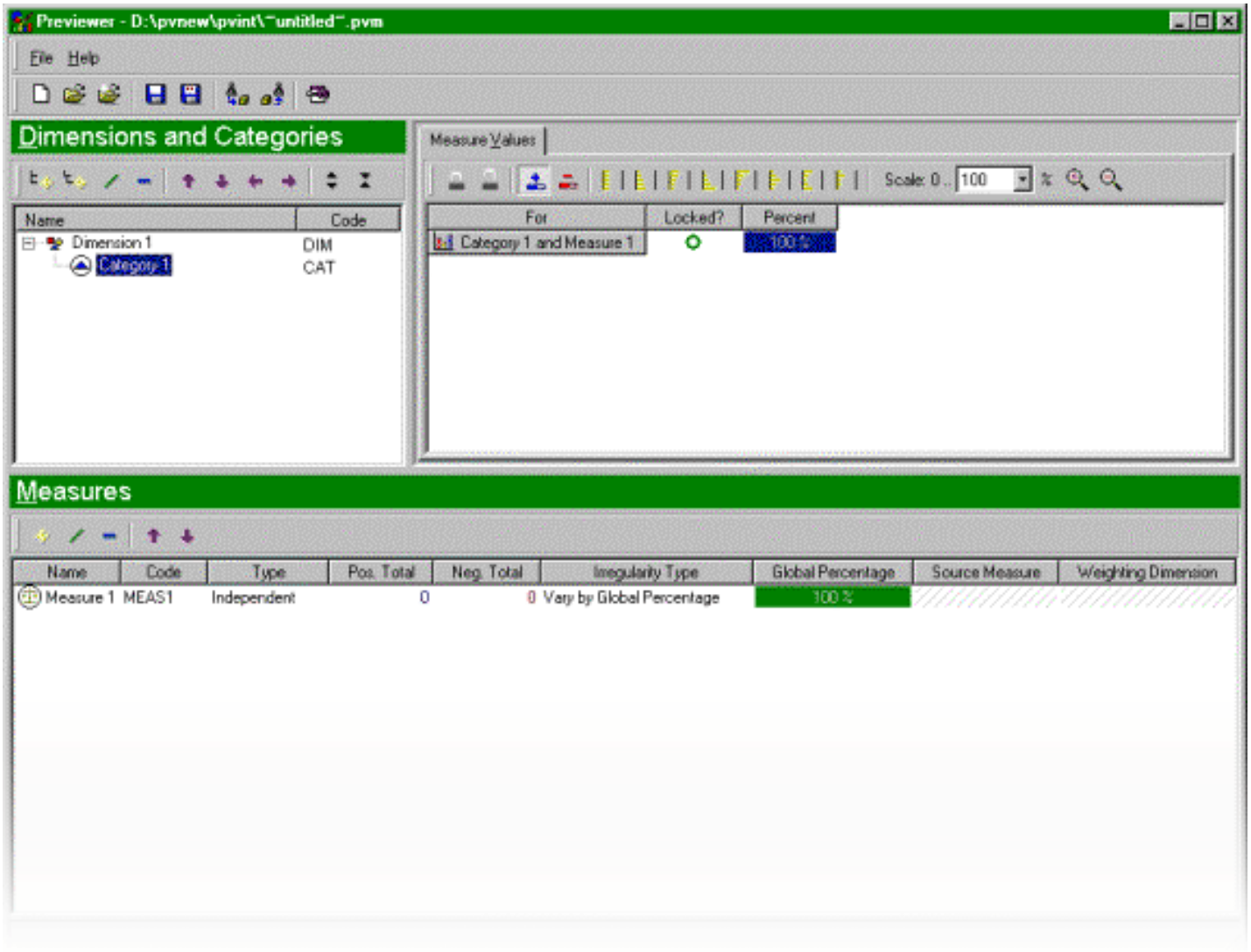
The .DAT file is a list of transactions or totals that correspond to business transactions. Each record in that file consists of a sequence of lowest-level categories, one from each dimension, followed by a sequence of numeric values, one for each measure.

The Interface

The Previewer interface allows you to design a model and add information about the distribution of the quantities associated with the measures defined in the model among the categories of the various dimensions.

To create an actual cube from the model you need to save the model as .pvm file. You can then choose the menu option "Generate Points from Previewer Model". This produces an OLAP cube based on generated data corresponding to the distributions you specified for the measures.

The Previewer interface looks like this:



As you can see, it is divided into three panes, each of which has an important function in Previewer:

1. **Dimensions and Categories**
2. **Measures**
3. **The categories in each dimension, and their hierarchical organization.**
4. **Measure Values (Percentage**

Dimensions and Categories

The pane on the upper right is labeled "Dimensions and Categories". It allows you to define the dimensions and the categories in the model.

The buttons at the top of this pane have the following functions:


Add a new item to the tree which is at the same level as the item on which the cursor is currently located.


Add a new item to the tree which is a subcategory of the current item.


Insert a new item into the tree before the current item at the same level.


Delete the current item


Move the current item up one place in the list.


Move the current item down one place in the list.


Raise the current item one level in the hierarchy of items.


Lower the current item one level in the hierarchy of items.


Expand the tree so that all items are visible.


Collapse the tree so that only the items at the highest level, the dimensions, are visible.

In addition to names, the dimensions and categories have codes, which can be entered from the interface. Both names and codes must be unique.

Measures

The pane at the bottom portion of the window is where measures are defined. Measures may belong to one of two types: Independent or Weighted. At least one measure in a model must be independent.

Independent measures

For independent measures, you specify the overall total for the quantity with which the measure is concerned and you define percentages for each category. These percentages govern how the measure is distributed within each dimension.

You can also specify an irregularity for the measure which is used to influence the randomness of the data. With a cube generated by Previewer, as opposed to a cube produced from live data, the distribution of a measure is specified independently within each dimension. Thus, if you specify that 5% of revenue comes from sales of TV sets, and 3% of revenue comes from sales of typewriters, this will not only be true of the overall total, it will be true individually in each location from New York to Peoria. The irregularity parameter tells Previewer to maintain the overall total and the overall distribution that was specified, but to cause variation in the distribution when constraints are imposed on other dimensions.

The original Previewer required you to generate at least as many points in your cube as was required to represent every possible combination of one lowest-level category from each dimension. This version of Previewer allows you to generate cubes with considerably fewer points. When you take advantage of this, and use Previewer to generate a cube with a smaller number of points, specifying a low irregularity doesn't guarantee that the irregularity in the resulting cube will be low. In general, what will happen, particularly if the cube has many dimensions, is that the specified irregularity will be seen in a high-level view of the cube, where the numbers being examined are for specified categories in two dimensions, and are otherwise overall totals, but when categories are specified for more dimensions, the irregularity will be higher, and essentially independent of the amount requested.

Weighted Measures

A weighted measure is a measure that is based on another Independent measure. Instead of specifying percentages for each lowest-level category as you would for Independent measures, you specify a set of ratios that the Previewer engine will use when it is generating points.

This is accomplished by specifying for the lowest-level categories of one dimension (weighting dimension) ratios between the value of the weighted measure and the value of the independent measure it depends on (source measure). The ratios you are required to specify in order to produce realistic results are average ratio, minimum, and maximum.

An obvious application of weighted measures is to allow the ratio of "Gross sales" to "Units sold" to depend on the categories in the "Product type" dimension. This would represent the typical price of the specified Product type.

The available buttons in this section resemble those of the categories pane:


Add a new measure to the list, after the current measure.


Add a new measure to the list which precedes the current measure.


Delete the current item.


Move the current item up one place in the list.


Move the current item down one place in the list.

For the measures, there are several items to enter.

Name	The name of the measure.
Code	The code used to identify the measure.
Type	The type of the measure, either independent or weighted.
Pos. Total	(Independent measures only) the total for the measure, or the total of its positive values if it has both positive and negative values.
Neg. Total	(Independent measures only) the total of the negative values of the measure.
Irregularity Type	(Independent measures only) Usually 'Vary by Global Percentage', one can choose here to specify the irregularity in a more detailed fashion. The other choices all involve specifying irregularity individually for each dimension, optionally with irregularities assigned to two or three families as well.

Global Percentage	(Independent measures only) This only applies if the irregularity type is 'Vary by Global Percentage'.
Source Measure	(Weighted measures only) This is the related independent measure that the Weighted measure is dependent upon.
Weighting Dimension	(Weighted measures only) This is the dimension for which the ratios are specified.

Measure Values (Percentages)

When measures have been specified, you can then enter percentages for each measure associated with the various lowest-level categories in each dimension. If a measure is weighted, rather than independent, then the Measure Values pane is used to specify the weighting ratios.

The button bar for this section has the following appearance:

The first two buttons are used to lock or unlock all the measure values displayed. With locking you can freeze one or more bars and thereby control precisely how the percentages are distributed as you increase or decrease the value of a given bar.

Since Previewer works by distributing quantities from a starting total, if a measure can take on both positive and negative values, a separate positive and negative total are required, each of which is distributed by means of its own set of percentages, and there is therefore also a pair of buttons to allow you to switch between positive and negative values.

The last series of yellow push buttons provide a quick way to select a distribution from a set of pre-canned distribution. The distributions that you can select from are: uniform, increasing, decreasing, exponential increase, exponential decrease, normal distribution, low in the middle, and random.

Other Features

From the File menu, you can select "Generate Points from Previewer Model", which is how cubes are created from the models you design and modify within the Previewer interface.

At present, it has the limitation that the current model you are working on must be saved to disk as a .pvm file before the engine will make use of it.

A new feature allows Previewer to read in an existing cube, and produce a model that corresponds to it. This works with cubes saved locally on your computer as .MDC files, and it requires you to have a copy of COGNOS PowerPlay installed, which Previewer will run in order to read the cube.

The .pvm file that results from this "reverse-cube-engineering" process will always only have independent measures. Also, although Previewer has many sophisticated features, it only supports cubes where:

- Every dimension and every category is applicable to each measure.
- All measures are quantities that can be, and are, totaled between categories, not averaged (that is, the measures are quantities that are extensive and not intensive)

This is a result of the way Previewer is designed to generate points (i.e. by allocating quantities from a total and doing so for a point for which a lowest-level category is specified for each dimension in the model).

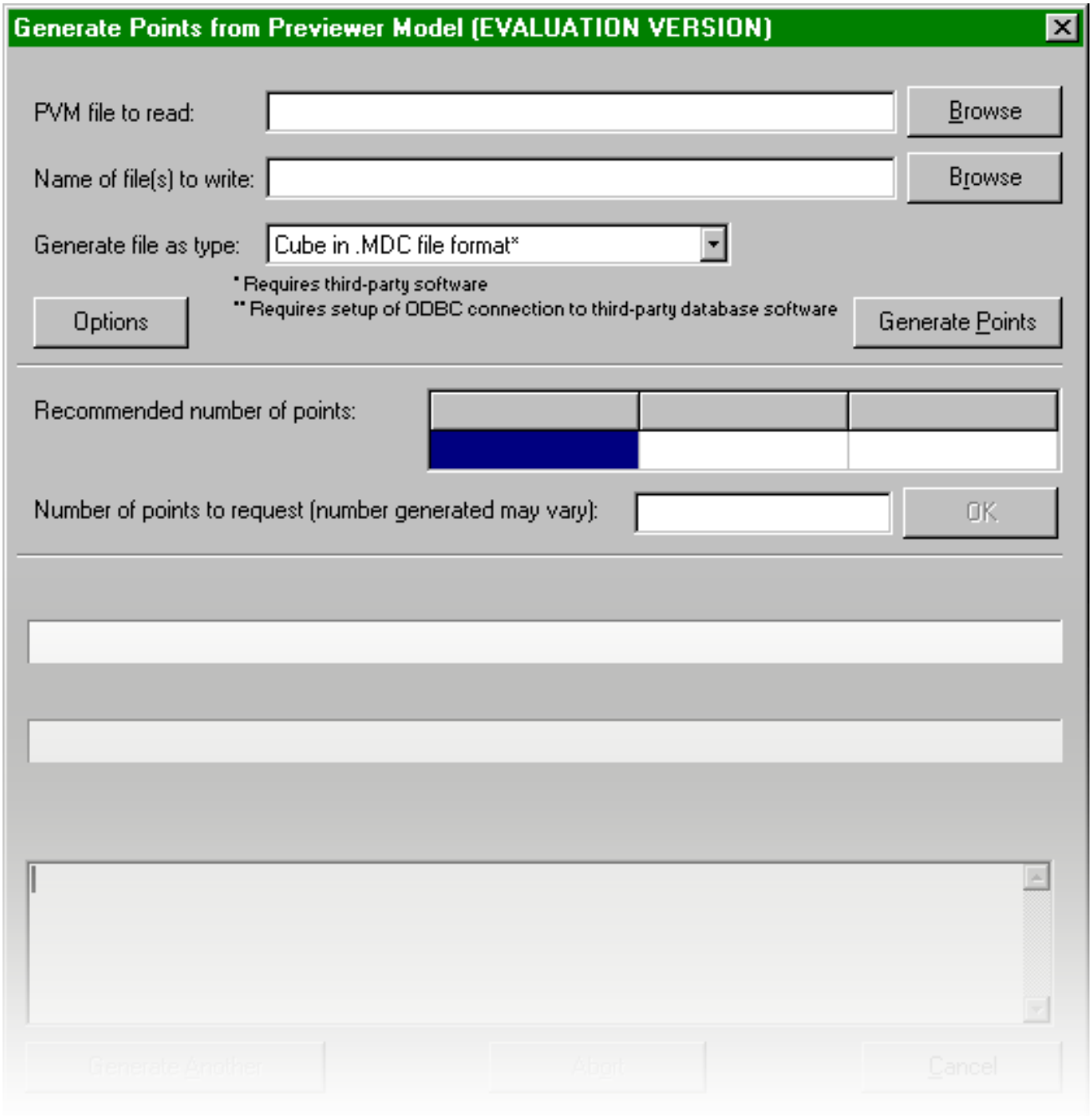
The Previewer interface does not allow you to create a model that does not conform to this criteria. Since Previewer now allows you to read an existing cube from an .MDC file, and produce a .pvm file from it, and there are existing cubes which do use the features Previewer does not support, it must be noted that attempting to do this for cubes not corresponding to this description will produce invalid results. In some cases, the results will still be usable, and in nearly all cases, deleting the measures or dimensions involved from the model, or entering new values or totals will allow you to proceed.

Difficulties have also been noted during development with creation of .pvm files from cubes containing negative numbers or fractional quantities. While many of these difficulties have been corrected, one should be alert to the possibility of problems when dealing with cubes containing measures with these attributes.

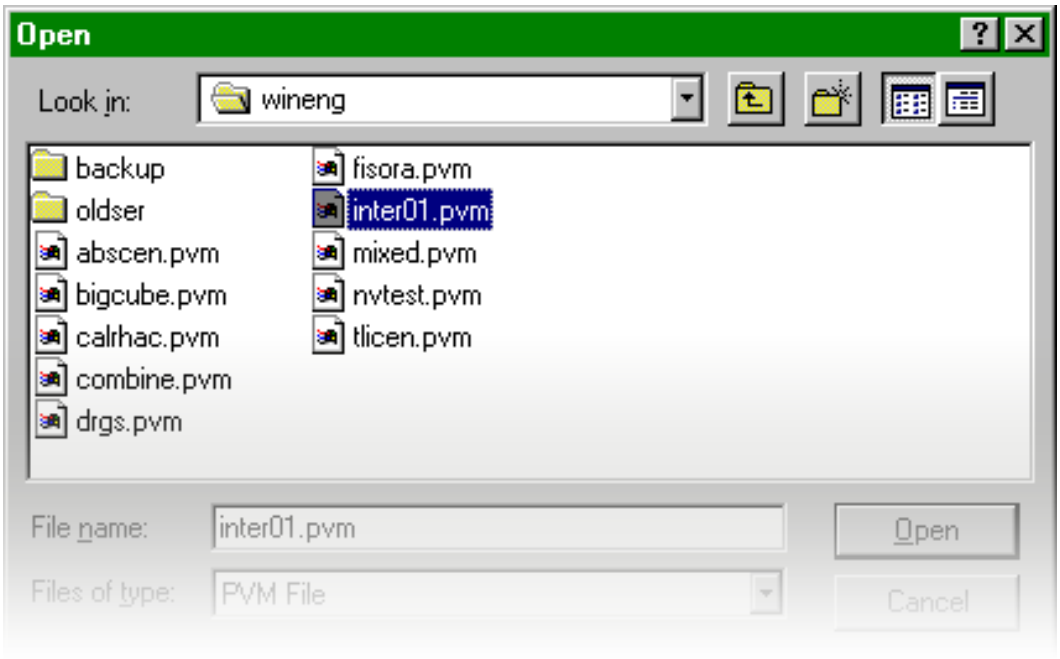
Create Cubes

Generating a Cube

Once you have finished designing a Previewer model, the next step is to use that model to create a cube, using data generated by Previewer. To do this, you will select the option "Generate Cube from Model" in the File menu, and Previewer will then present you with a dialogue box that looks like this:

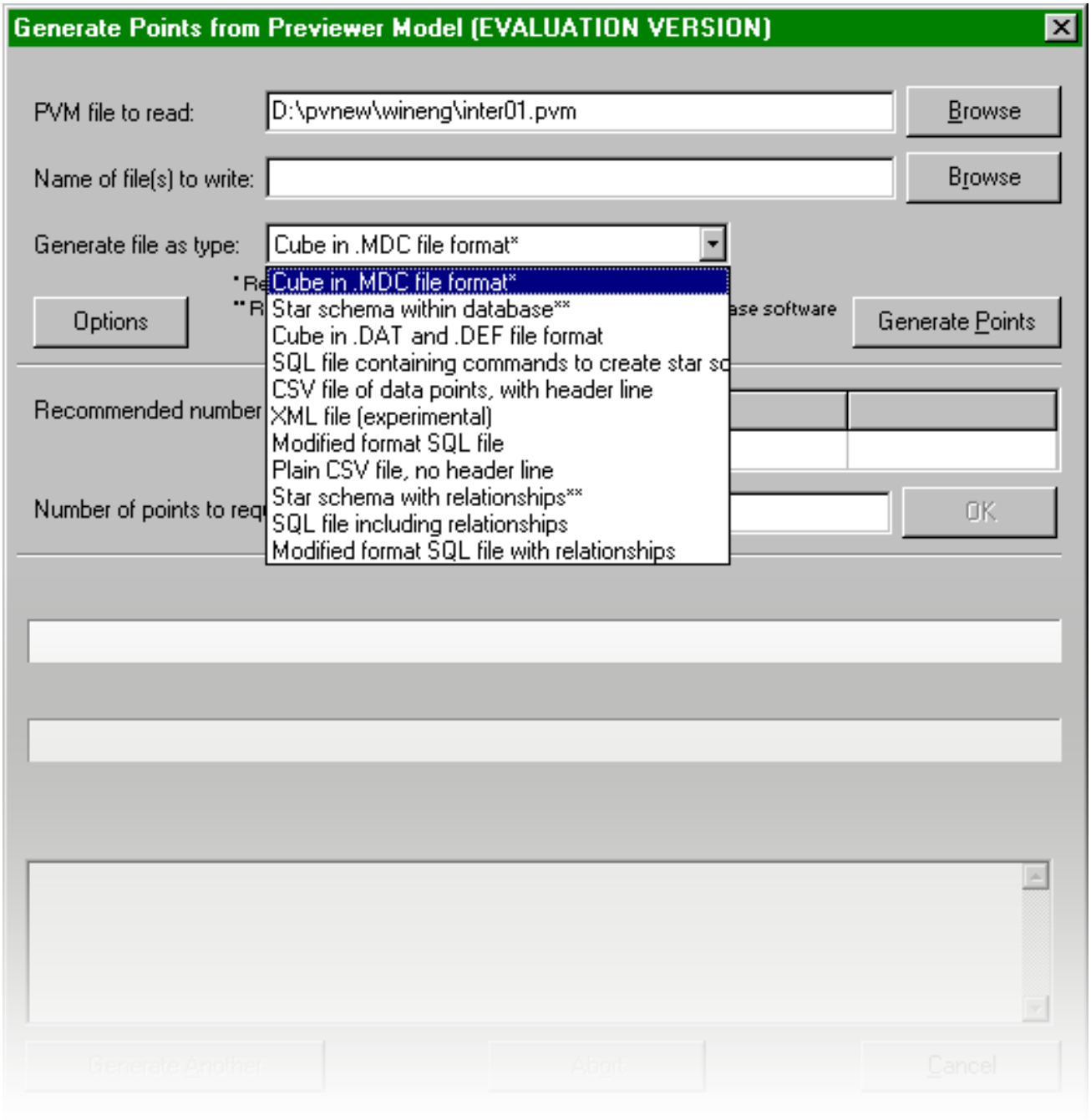


From this dialogue box, you begin the process of generating a cube by specifying the .pvm file containing the Previewer model to work from, either by typing its name in the field provided, or by using the Browse button adjacent to that field to obtain a standard file-selection dialogue box. Such a dialogue box will look like this:



This provides you with a convenient way to locate and specify the file you want by drive and directory.

The next step is to select the format of output that you would like to have generated.



Previewer 2.0 offers several possible formats for output.

- **Cube in .DEF and .DAT Format**: Previewer 1.0 could only be used to generate cubes that could be viewed within Cognos PowerPlay. It did this by generating sets of .DAT and .DEF files, which could be submitted to Cognos Transformer to produce actual cubes in .MDC file format.

- **Cube in .MDC Format**: Previewer 2.0 offers the option of creating a cube in .MDC file format directly. This is done by running a copy of Transformer after the .DAT and .DEF files have automatically been written. Thus, if you choose to create a cube in .MDC format, you still need to specify the filename for the .DAT and .DEF files. Also, a copy of Transformer must be installed on your computer. If you experience problems running Transformer, you can of course run Transformer after selecting "Cube in .DEF and .DAT Format".

- **CSV File**: You may output the data points in the form of a file of comma-separated values that can be easily read by a BASIC program or by a number of stand-alone database packages.

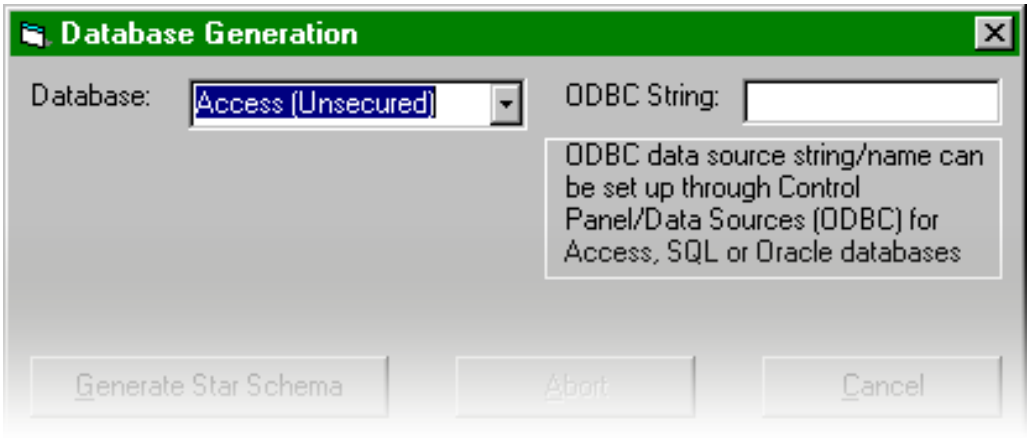
Optionally, this file may contain a header line naming the fields.

- **XML file**: The points can be output in an XML-style format. However, the output at present is incomplete, including only the XML code likely to be common to whichever XML-based database products may be supported at some time in the future.

- **SQL and Star Schema**: The program can also generate data in SQL (Structured Query Language) that contains commands to create and which can populate a database which contains the cube in a popular star schema format. In addition to creating an SQL file, you can choose to create the actual database. This works in the same way that creating an .MDC format works: the SQL file is written out, and then a database program is used, this time via an ODBC connection, to execute the SQL commands to create the database.

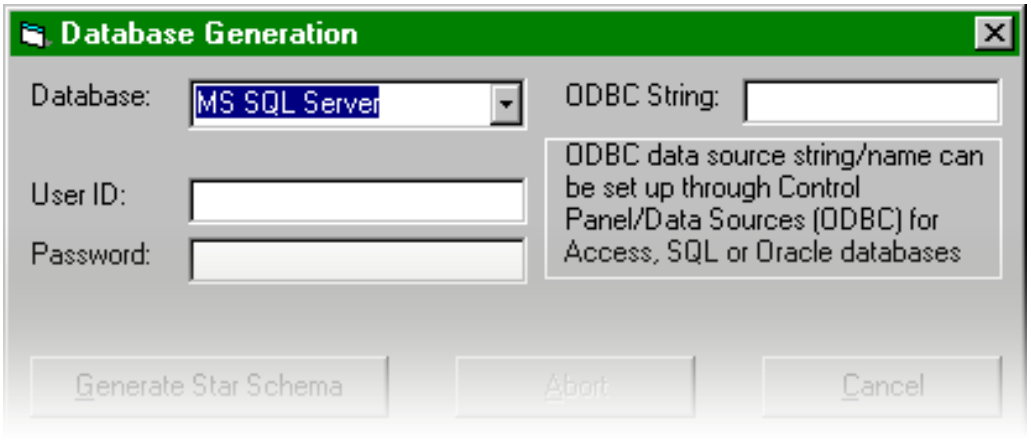
You will notice in the list there are options which refer to a "Modified format SQL file". When a database is created, the temporary file with the SQL commands is written out in a special format, where each command is on a single line, and the terminating semicolon is omitted. As well, there are extra comment lines which are used to facilitate actions such as optionally deleting pre-existing tables with the same name as those to be created, if that is authorized by the user. These options, therefore, allowed output of the temporary SQL file without executing its commands for test purposes.

When you choose to create a database, after the points are generated and the temporary SQL file is written, you will be presented with this screen:



You need to select the type of database, which can be a database within Microsoft Access, Microsoft SQL Server, or Oracle. The connection to the database is set up using the "Data Sources (ODBC)" icon in the ODBC settings in the Control Panel in Windows, and the name of the data source is entered in the field labeled "ODBC String:" on this form.

If you select an Oracle or SQL Server database, or an Access database with a password, this dialogue box will take on the following appearance:

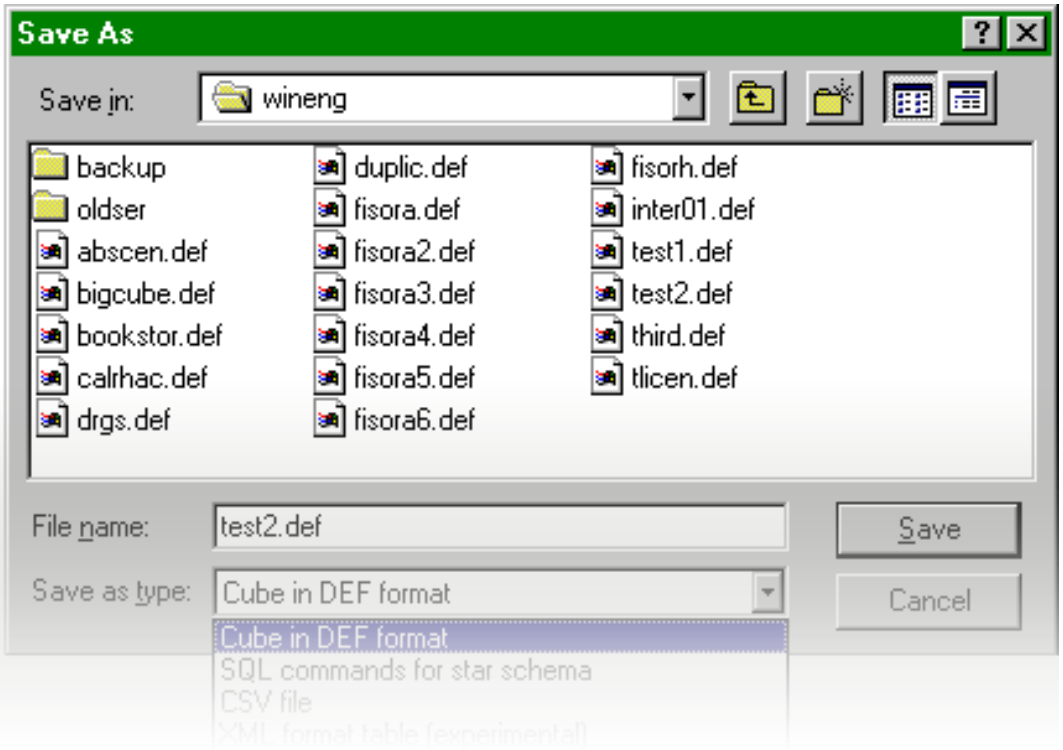


This will give you two additional fields in which to enter your user identification and your password.

Once you have filled in all the fields, Pressing the button marked "Generate Star Schema", Previewer will read the temporary SQL file and cause the corresponding database to be created. Pressing "Cancel" will return you to the Previewer engine.

It is possible, after a database is created, to choose another database format, and fill in the fields again, and create another database containing the same cube before returning.

The file(s) to be written may be specified by a standard Windows dialogue box. This dialogue box displays existing files, and an option is available to choose the type of existing files to see.



By selecting the output format before selecting the output file, this dialogue box will begin by displaying files with the appropriate extension by default.

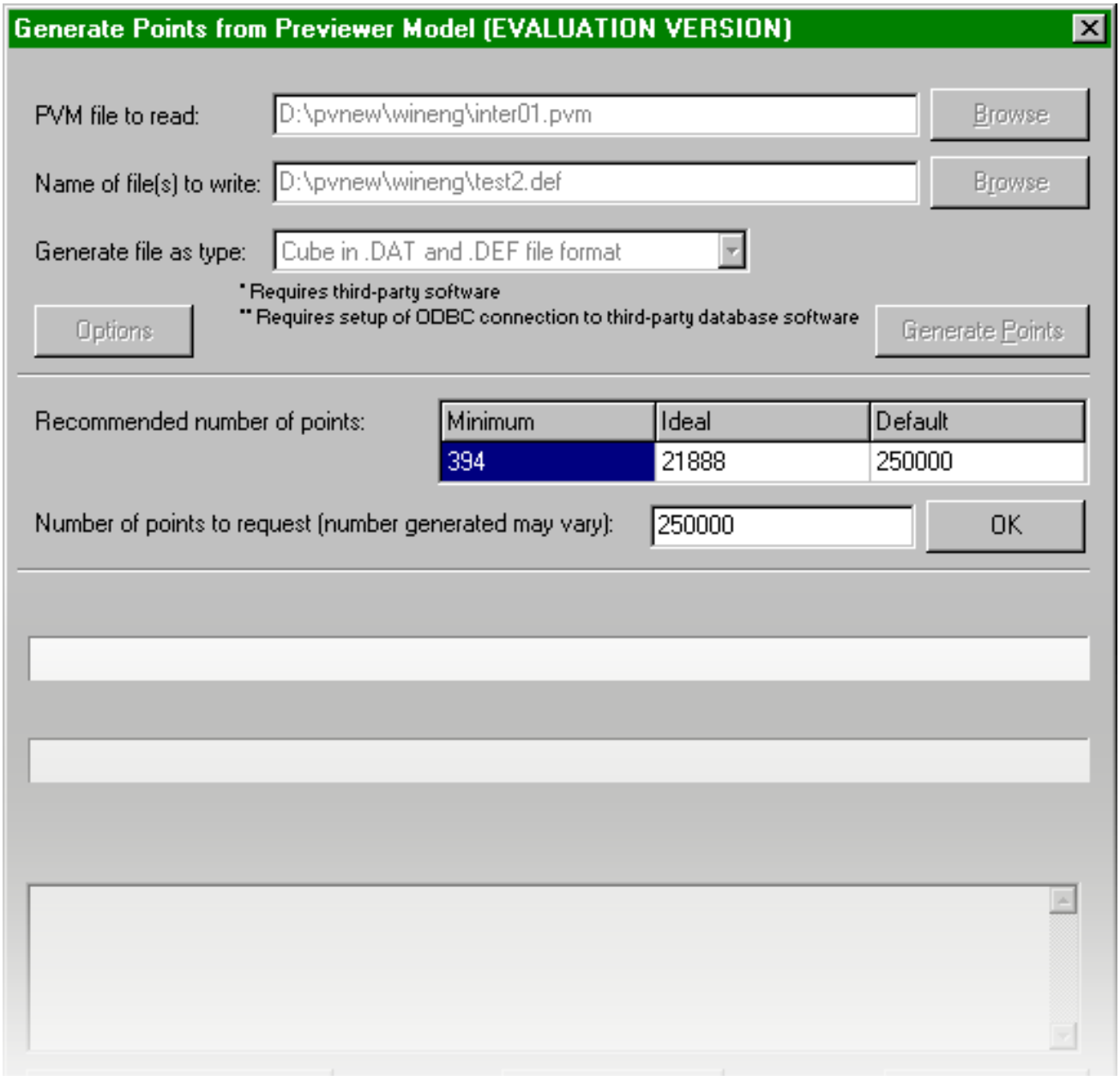
Whatever file name may appear in the field for specifying the output file, the extension of the output file

will be altered, if necessary, to the extension appropriate for the format selected. The Options button in the main dialogue box allows you to set a number of advanced options which may be useful in the operation of Previewer. Once the files to produce have been selected, you can then presses the "Generate Points" button on the main point generation dialogue box to proceed.

If the output file specified happens to be one that already exists, a dialogue box like the following will appear:

Selecting "Yes" lets you specify another file. Selecting "No" will let the existing file be overwritten. Selecting "Cancel" returns you to the main dialogue box, as if "Generate Points" had not been pushed.

Once you have committed to generating points, the main dialogue box will appear as follows:

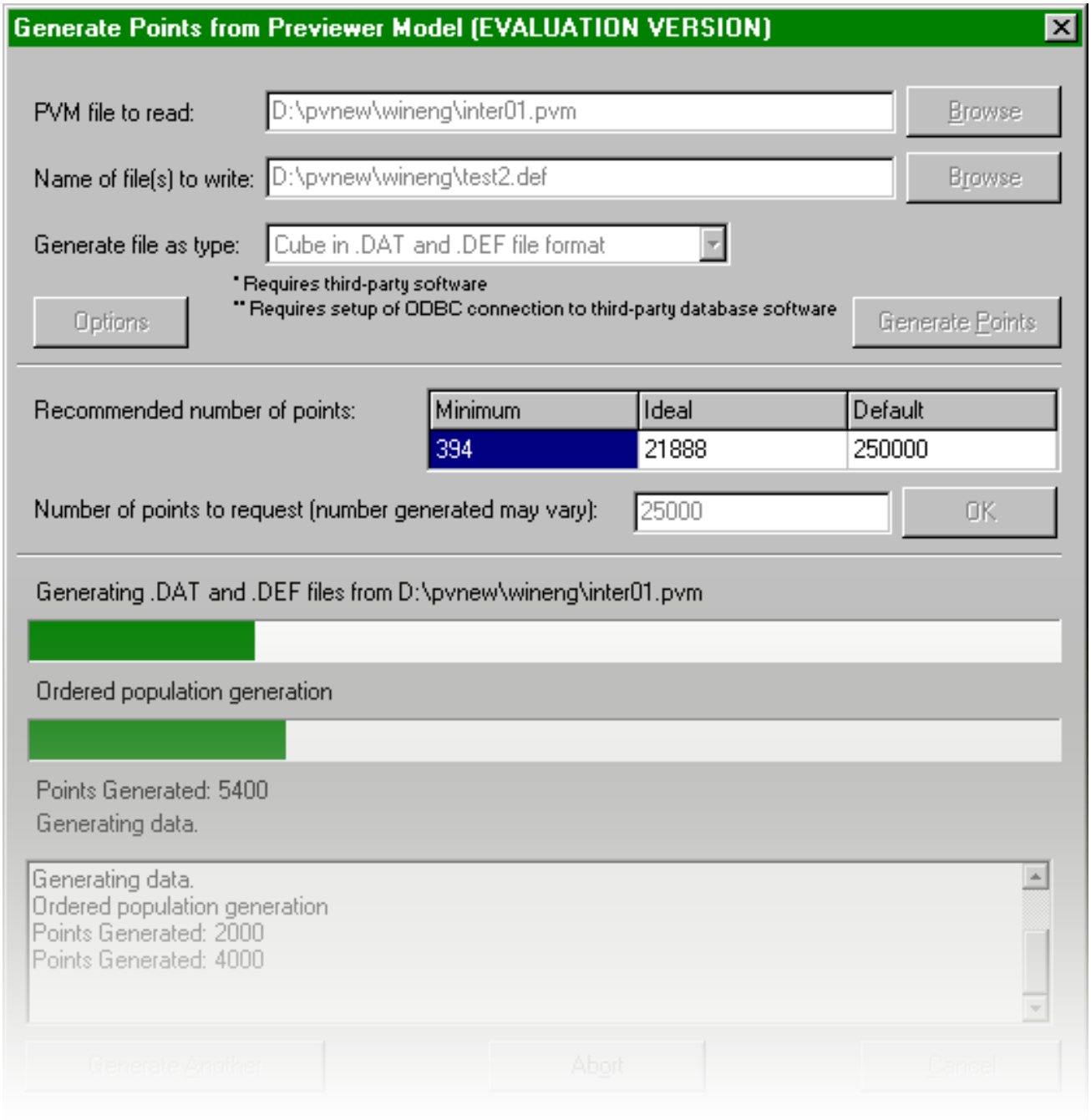


At this stage, a field is enabled in which you can enter the number of points you would like Previewer to generate. Information is displayed to help you decide how many points you would like:

- The minimum required number of points.
- A number of points that gives Previewer flexibility in generating points for the model.
- The default value selected by Previewer.

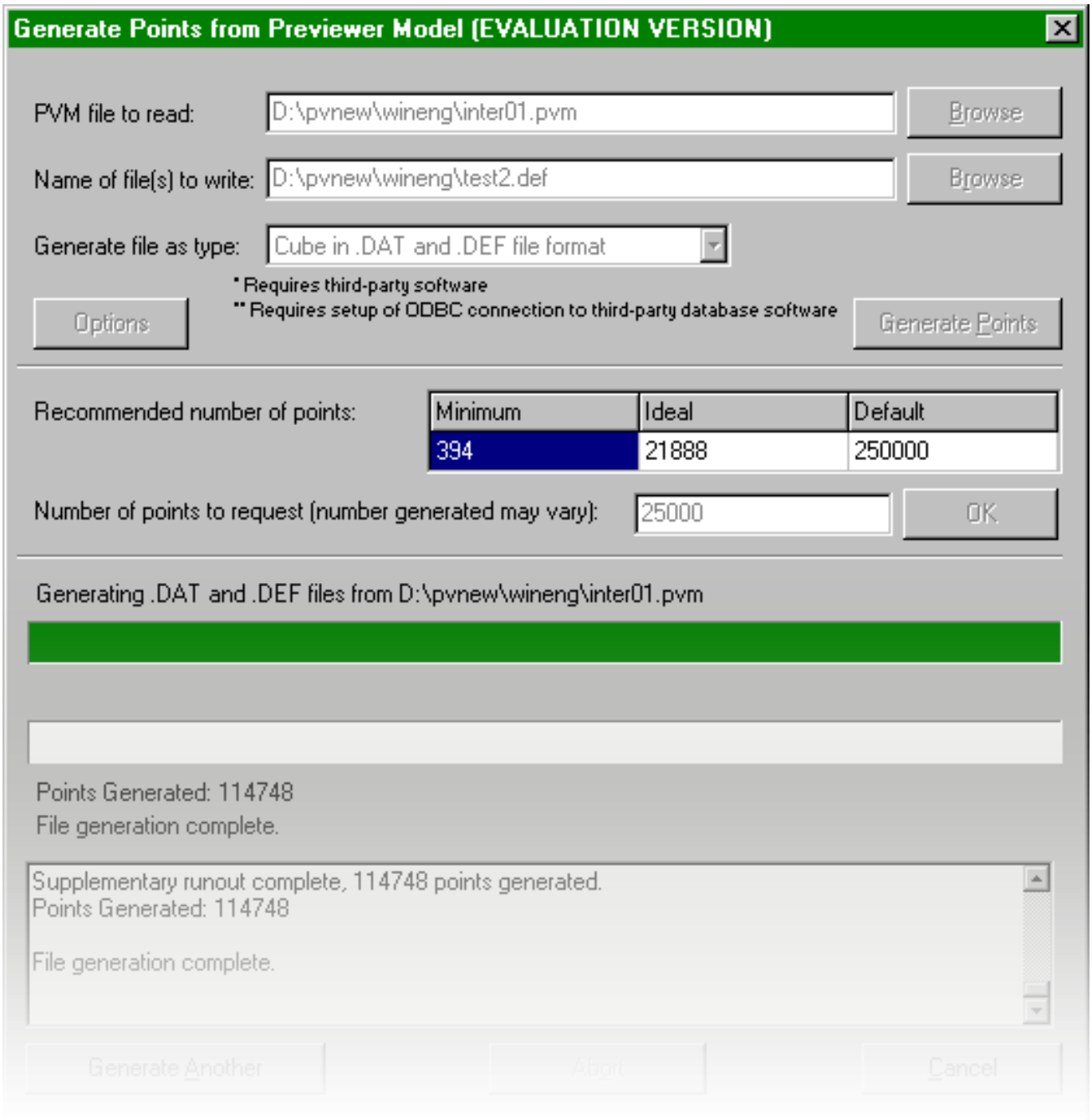
Note that the Cancel button remains active, so you can exit without generating a cube.

If you want to change the number of points to generate, type the number you want in the field provided. Press the OK button to start the point generation process.



There are two progress bars that will allow you to view the progress of point generation. A scrolling text box also contains messages output from the point generation process, and other messages indicating what the program is doing appear above and below the progress bars.

An Abort button is available which allows you to cut short the generation of points.



In this example, although 25,000 points were initially selected, as you can see from this screenshot, 114,748 points were actually generated. Some degree of variation from the number of points requested is common (but it is not usually as large as this example). Also, the variation can be in either direction. It is possible to receive less than a third of the points requested as well as more than three times as many. Usually, the variation will be less; it depends on the number of dimensions in the model, the number of categories in each dimension, and how uniformly the totals for the measures are distributed.

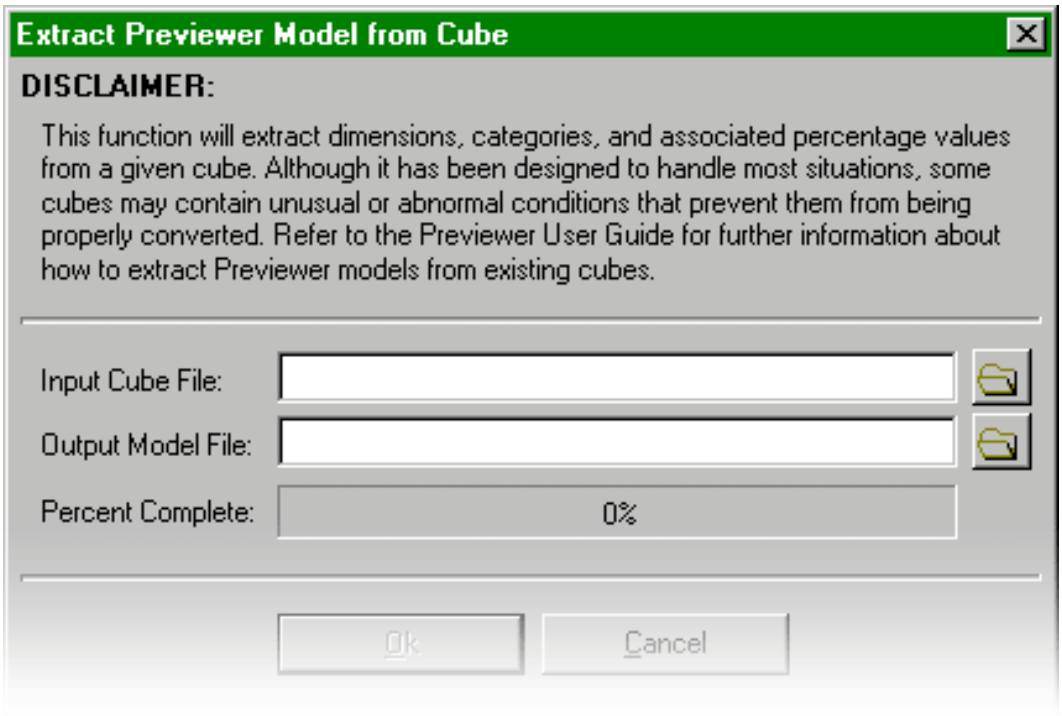
If there are several measures, each one sparsely distributed, and each one distributed in a different way, a large number of points is particularly likely to be forced in order for the totals for the measures to be generated.

Note that now a "Generate Another" button is enabled. This can be used to return to the top of the dialogue box to select another .pvm file. Otherwise, the Cancel button will return you to the main Previewer interface.

Model

Obtaining a Model from a Cube

Selecting the option "Extract Previewer Model from Cube..." causes the following dialogue box to appear:



This dialogue box allows the user to select the .mdc file to read, and to specify the name of the .pvm file to generate, using standard file dialogue boxes.

This dialogue box allows the user to select the .mdc file to read, and to specify the name of the .pvm file to generate, using standard file dialogue boxes. The progress of model extraction will be shown on the percent bar visible on this dialogue.

Because users of PowerPlay may use many advanced features which make their cubes unique, and the model extraction procedure is a straightforward procedure which does not attempt to account for all the possible variations in a PowerPlay cube, a disclaimer appears on the dialogue box.

Limitations of cube conversion

Although Previewer can generate .pvm files from many existing cubes in .mdc format, it does have a number of limitations.

Note that this function requires a copy of COGNOS PowerPlay installed on the user's computer, and a valid cognos.ini file in the user's Windows directory (Previewer does not attempt to read .mdc files directly, and it requires the cognos.ini file to locate the user's copy of PowerPlay).

- Some cubes have the name MEASURES replaced by DATA or another keyword. An attempt was made to make the program able to respond automatically to this through use of PowerPlay's DimensionLine object, but this was not successful, nor was it possible to edit those cubes to make them more conventional.
- Some cubes, including the Great Outdoors example provided with COGNOS PowerPlay, give some measures 'Not Applicable' status for selected dimensions. The cube conversion method provided with Previewer has not been designed to handle this. The result, however, is usually a model where the percentages are all zero for the measure and dimension where this has happened, and therefore the model may be edited to allow one to proceed.
- It is also assumed that all measures are quantities for which totals are meaningful. If a measure consists of percentages, inaccurate or meaningless results may be produced by Previewer when a cube is generated based on a model reverse-engineered from the cube containing such a measure. This can be ignored, or the measure can be deleted within Previewer afterwards by the user.
- While the Previewer point generation engine is designed to support fractional and negative numeric values, occasional problems have been encountered, particularly for negative values that are sparsely distributed. There is the possibility that the point generation engine will quit early in such a case, but it is designed to properly close the file with the points it has generated when this happens.
- The Previewer cube conversion process does not recognize calculated measures; the values appearing for these measures in reports will be read, and an independent measure will be constructed in the .pvm file for the measure, but no knowledge of the underlying calculation will be obtained by this process.
- More generally, cube conversion does not automatically make use of all of Previewer's advanced features nor does it support any of the more advanced features of COGNOS PowerPlay. As well, no attempt is made to analyze an existing cube to determine what level of irregularity should be used: instead, 10% global irregularity is used for every measure, and all measures are created as independent measures.

Category Trees from Databases

Although Previewer does not, at this time, have the ability to accept a complete model from data stored in star schema format in a relational database, it is able to use comma-delimited files generated from such a model to produce a tree of categories for one dimension.

To make use of this feature, it is necessary to give the comma-delimited file generated from your database a .PDC extension, and to select that extension in the file dialogue box in the Open command on the file menu. This will cause a complete model, with one measure and random percentages, to be created, which can either be used as a starting point for creating a model, or saved and read in using the Import command.

The two supported formats of category tree data are as follows:

1. With three fields:

category code, category name, category code of parent category

The following is an example of this format:

```
"P1", "First Header", ""
"P2", "Subheading 1", "P1"
"C1", "First LLC", "P2"
"C2", "LLC under FH", "P1"
"P2a", "Subheading 2", "P1"
"P3", "SubSubheading 1", "P2a"
"C3", "Third LLC", "P3"
"C4", "Fourth LLC", "P3"
"C5", "LLC under S2", "P2a"
"P4", "Second Header", ""
"P5", "Subheading 3", "P4"
"C6", "Sixth LLC", "P5"
```

2. With an even number of fields: each pair of fields contains a category code and a category name. The first pair of fields contains the code and name of a lowest-level category. All parent categories, if any, are in subsequent pairs of fields: the topmost parent category is in the last pair of fields, its child category, if not the lowest-level category for the row, is in the pair of fields immediately preceding, and so on. Blank pairs of fields, if any, start with the second pair of fields, and continue contiguously.

The following is an example of this format:

```
"C1", "First LLC",    "",   "",                "P2",  "Subheading 1", "P1", "First
Header"
"C2", "LLC under FH", "",   "",                "",    "",             "P1", "First
Header"
"C3", "Third LLC",    "P3", "SubSubheading 1", "P2a", "Subheading 2", "P1", "First
Header"
"C4", "Fourth LLC",   "P3", "SubSubheading 1", "P2a", "Subheading 2", "P1", "First
Header"
"C5", "LLC under S2", "",   "",                "P2a", "Subheading 2", "P1", "First
Header"
"C6", "Sixth LLC",    "",   "",                "P5",  "Subheading 3", "P4", "Second
Header"
```

Both of these methods will produce the same tree structure of categories:

```
First Header P1
- Subheading 1 P2
- - First LLC C1
- LLC under FH C2
- Subheading 2 P2a
- - SubSubheading 1 P3
- - - Third LLC C3
- - - Fourth LLC C4
- - LLC under S2 C5
Second Header P4
- Subheading 3 P5
- - Sixth LLC C6
```

Features

Advanced Features

The original version of Previewer (version 1.0) provided a basic tool for creating fabricated OLAP cubes. With version 2.0 the following advanced features were added:

- **The ability to generate a model with fewer points.**
- **The ability to control the irregularity of a model.**
- **The ability to create a measure that is dependent on another measure.**

Generating a model with fewer points

In the Previewer 1.0, at least one point was generated for every possible combination of lowest-level categories from each dimension. The points generation engine in Previewer 2.0 operates on different principles, and can, if desired, generate far fewer points to create a cube. This ability may be useful when there are a large number of dimensions in a model, and the number of combinations of categories would prove to be unwieldy.

Controlling the irregularity of a model

To understand irregularity, consider the following examples. From one year to the next, the proportion of sales between different regions may not change much. But it would still be unrealistic not to have it change at all. Or, if a company has several products, which it sells in several different locations, the proportion each product contributes to sales in any one location should be similar to the proportion it contributes to total sales, but it would be unrealistic for the proportions to be identical.

One approach to accommodate this would be to provide a way for Previewer users to maintain combinations of categories from different dimensions. However, with Previewer 2.0, another method is used which permits forcing the percentages of a measure, when viewed between categories in one dimension, to vary when a category in another dimension is selected.

The percentages for a measure are divided up between up to ten populations. For each dimension, its percentages are randomly varied between populations in such a way that the variations cancel out. But these variations will correlate with variations found in other dimensions, causing irregularity to appear in the final model.

With large models, however, if it is selected to produce a small number of points, the randomness involved in generating individual points will itself produce irregularity. This was accentuated with the new method of generating points, since each category in each dimension is chosen at random with each point. Therefore, to allow low levels of irregularity to be achieved when desired, in the Previewer 2.0 engine, two new point generating loops were added which loop through all categories in all dimensions

as originally done in Previewer 1.0. (They will only be employed, however, if the user has requested the generation of enough points).

Previewer 2.0 was originally designed, therefore, to ask the user for one additional parameter for each measure besides its total: its irregularity level, which internally is a number between 0 and 1. Further enhancements to this have recently been made. Now, for each combination of a measure and a dimension, an irregularity level can be specified.

As well, a common irregularity, and an orientation for the previous irregularity level, can be specified. The irregularities for a measure can be specified to belong to two (which forces nine populations) or three (which forces eight populations) classes. In that case, each dimension within that measure can have its irregularity specified as belonging to one of those classes or to none of them.

What does this do?

An independent measure in a Previewer model begins with an overall total. Within each dimension, percentages of that total are assigned to the individual categories within the dimension. When a cube is generated from the model, you can drill down into that cube with PowerPlay, for example. This lets you see graphs of how the measure is divided up between the categories in a dimension. For example, how sales are divided up between product lines. This is true not only for the overall total, but also for the part of the cube that corresponds to only those data points which belong to a single category in another dimension. For example, the distribution of sales in one location.

As we've seen, irregularity is concerned with ensuring that those partial distributions, instead of all being identical to the distribution of the overall total, vary somewhat. The degree of this variation is controlled by the irregularity. One can just assign a single value to the irregularity for an entire measure, or give each dimension its own irregularity. When each dimension has a different irregularity, then the degree of variation you see when looking at the distribution along the categories in one dimension for the part of the cube corresponding to one category in a second dimension is controlled by the smaller of the two irregularities of the two dimensions involved.

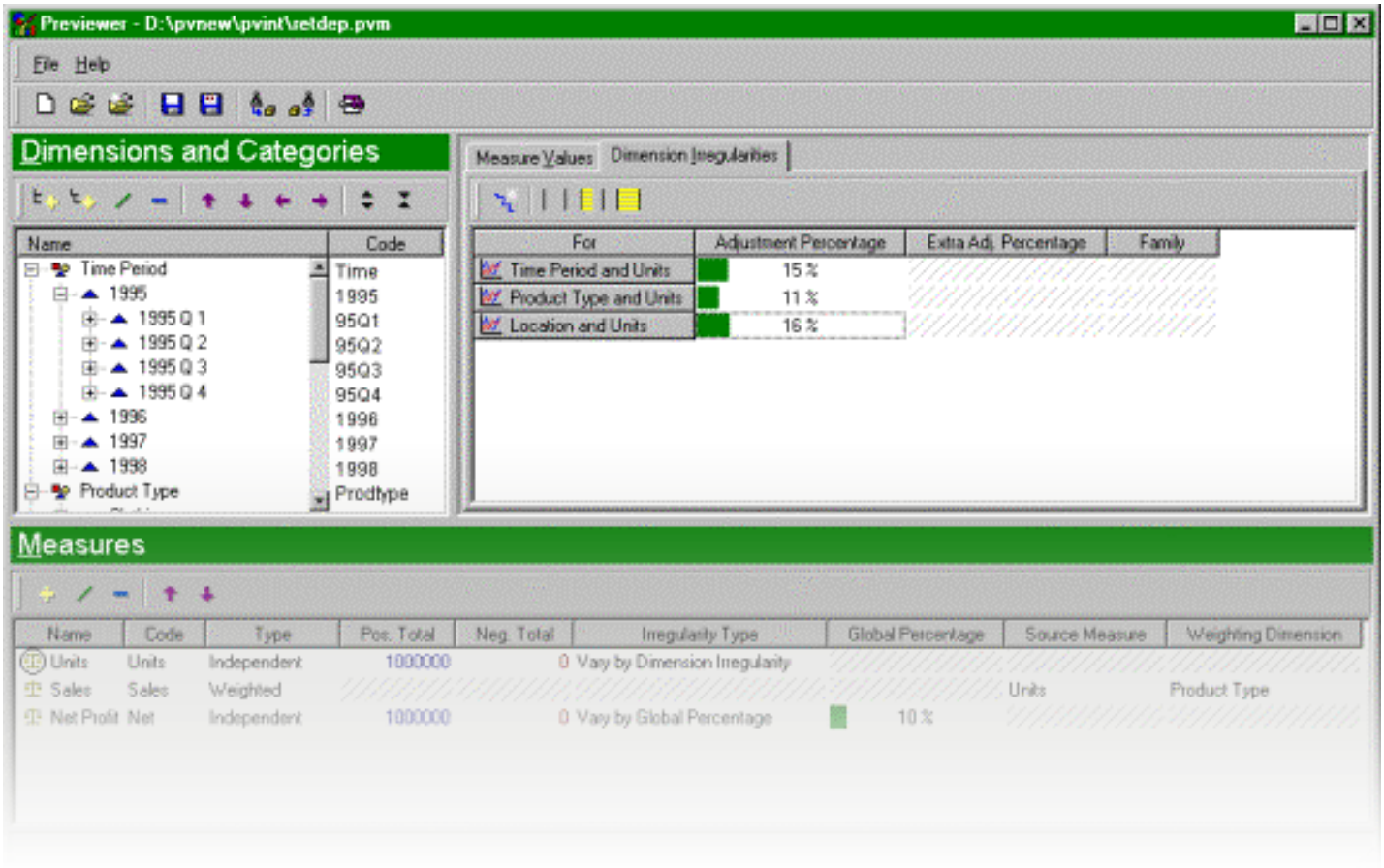
When irregularities are specified as belonging to classes (irregularity families), you are able to exert more control over how the dimensions behave in relation to each other. The common irregularity follows the previous rule: its contribution to the variation seen where two dimensions are involved is based on the lesser of the common irregularities of those two dimensions. The irregularity that is associated with one of the two or three classes, called irregularity families, will make an additional contribution to the irregularity you see only if both dimensions involved belong to the same class. If that is the case, the additional contribution is again based on the lesser of those two irregularities.

What is this feature used for?

Consider a cube reflecting a retail sales situation. In addition to the typical dimensions of product type, location, and time, another dimension, using data from customer surveys is included: customer income level. The distribution of sales between different types of product is likely to change only slightly from one year or one city to another, but considerably more variation in the product mix is expected to be seen for different customer income levels.

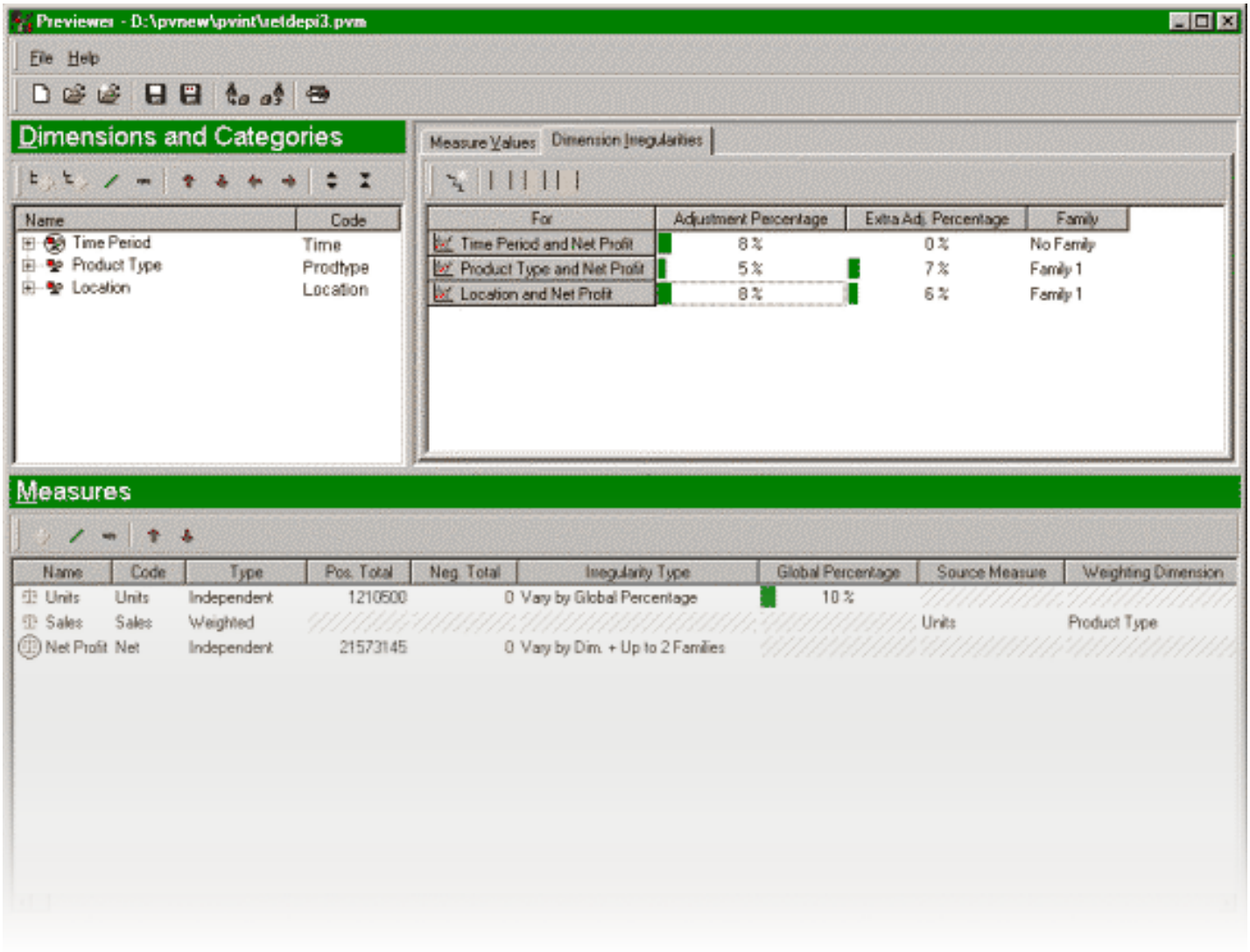
This can be handled by causing customer income level and product type to belong to the same family, and assigning the other dimensions to another family, so that irregularity specified for the two dimensions of customer income level and product type is not "seen" by the other dimensions.

In the Previewer interface, when you choose to control irregularity in detail by selecting "Vary by Dimension Irregularity" instead of the default "Vary by Global Percentage" for a measure, the screen may look something like this:



In this example, you can specify the irregularity for each dimension for the measure Units. Other choices for Irregularity Type allow the other two columns, "Extra Adj. Percentage" and "Family" to be used. The Adjustment Percentage is the main irregularity for the measure, the Extra Adjustment Percentage is the common irregularity for the measure, and the Family is the orientation for the main irregularity, as discussed above.

In that case, the screen changes its appearance to something like this:



Consider the case where Units Sold and Dollar Value of Sales are two measures. In this case, we would like a higher number of units sold to be accompanied by a higher dollar value of sales wherever it occurs.

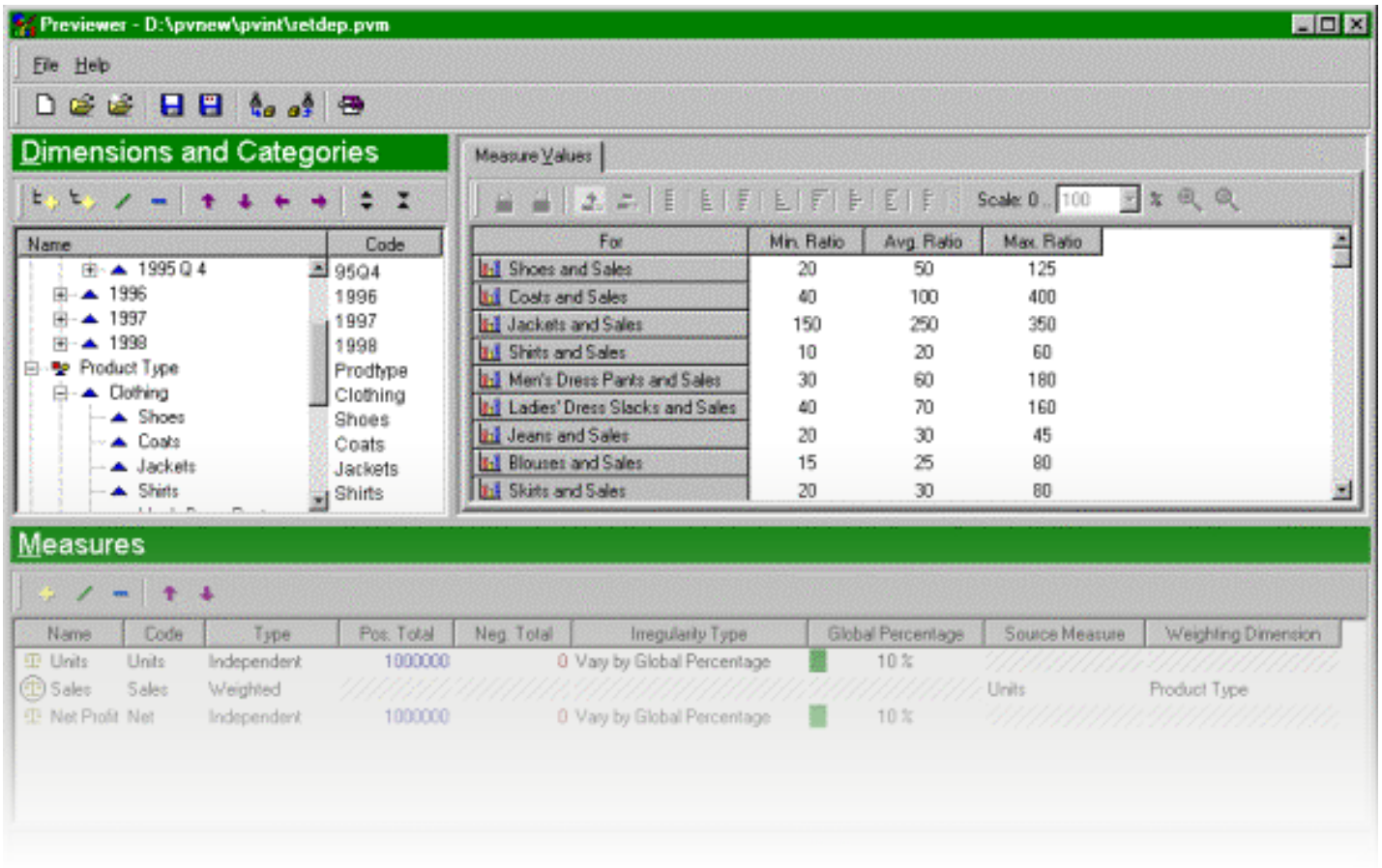
Weighted measures:

Previewer 2.0 allows the user to specify weighted measures in addition to independent measures.

Independent measures are the standard type that we have seen up to now, and which existed in the original Previewer. These measures have a total, percentages for each category in each dimension, and now an irregularity level. Only independent measures have an irregularity level.

Weighted measures allow an additional degree of complexity. A source measure and a total are again indicated. Then, a dimension is specified, and percentages are given for the categories in that dimension. In addition, for each lowest-level category in that dimension a separate minimum and maximum ratio between the weighted measure and the source measure is given. Thus, if the source measure is units sold, and the weighted measure depending on it is dollar volume, one can choose the type of item as the dimension, and the percentages for every category can be based on the average price for each type of item times the percentages of the units measure for each type of item, and then the ratios for the categories can be based on the price ranges.

When a weighted measure is specified for a model, Previewer may look something like this:



Because Previewer starts with all ratios equal to 1, and it validates input to ensure that the average ratio is between the minimum and maximum ratios, if the ratios are all larger than 1, it is necessary to enter the maximum ratio first, followed by the average and the minimum ratios.

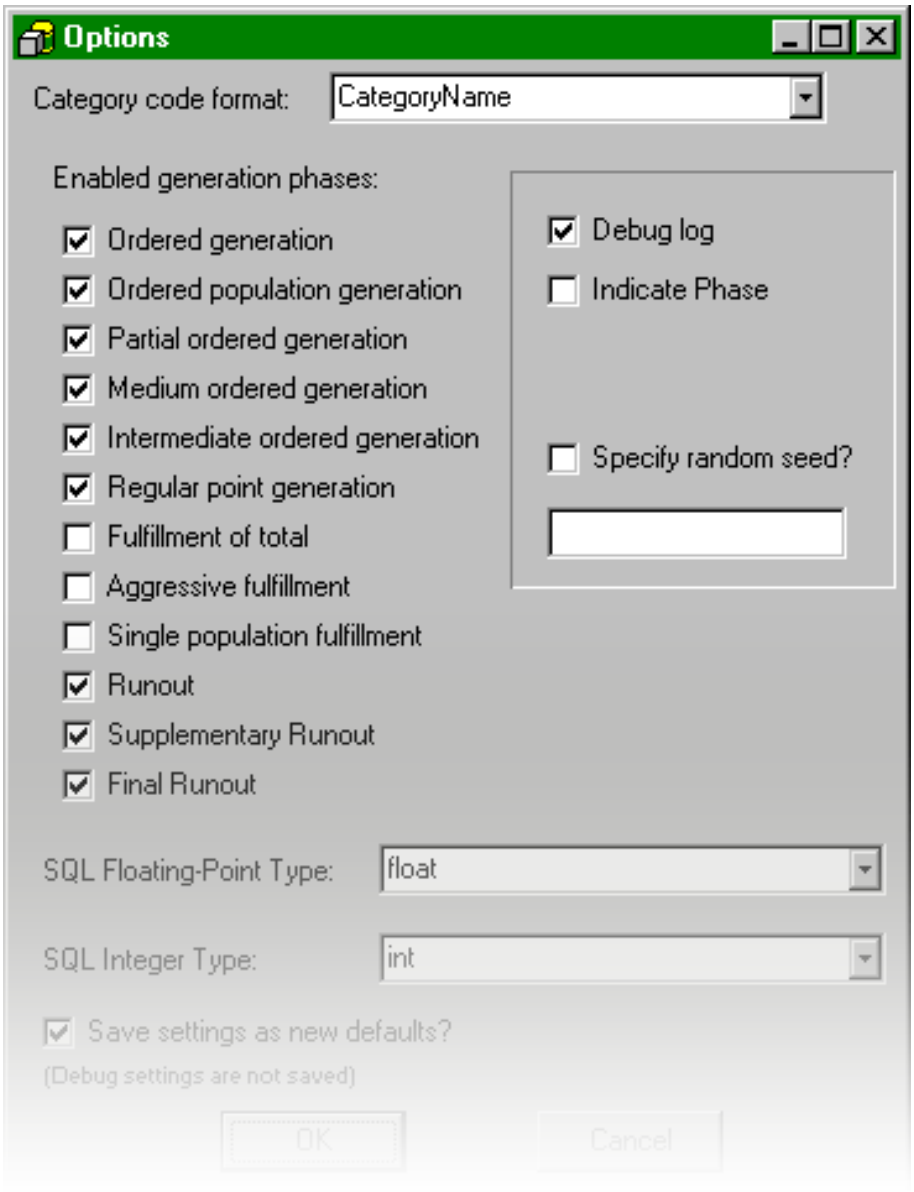
The purpose of a weighted measure is to ensure that when measures that are logically related, such as units sold and sales in dollars, that the quantities in both measures are distributed in the same way.

While irregularity is only specified for independent measures, there is a connection between the concept of irregularity and weighted measures. If it were not for irregularity, either irregularity that is requested directly, or irregularity that results from requesting the generation of fewer points than are called for by the product of the number of lowest-level categories in every dimension, the independent measures would all have a uniform distribution, following the percentages specified for each category. In that case, the discrepancies which weighted measures are designed to prevent would never become noticeable.

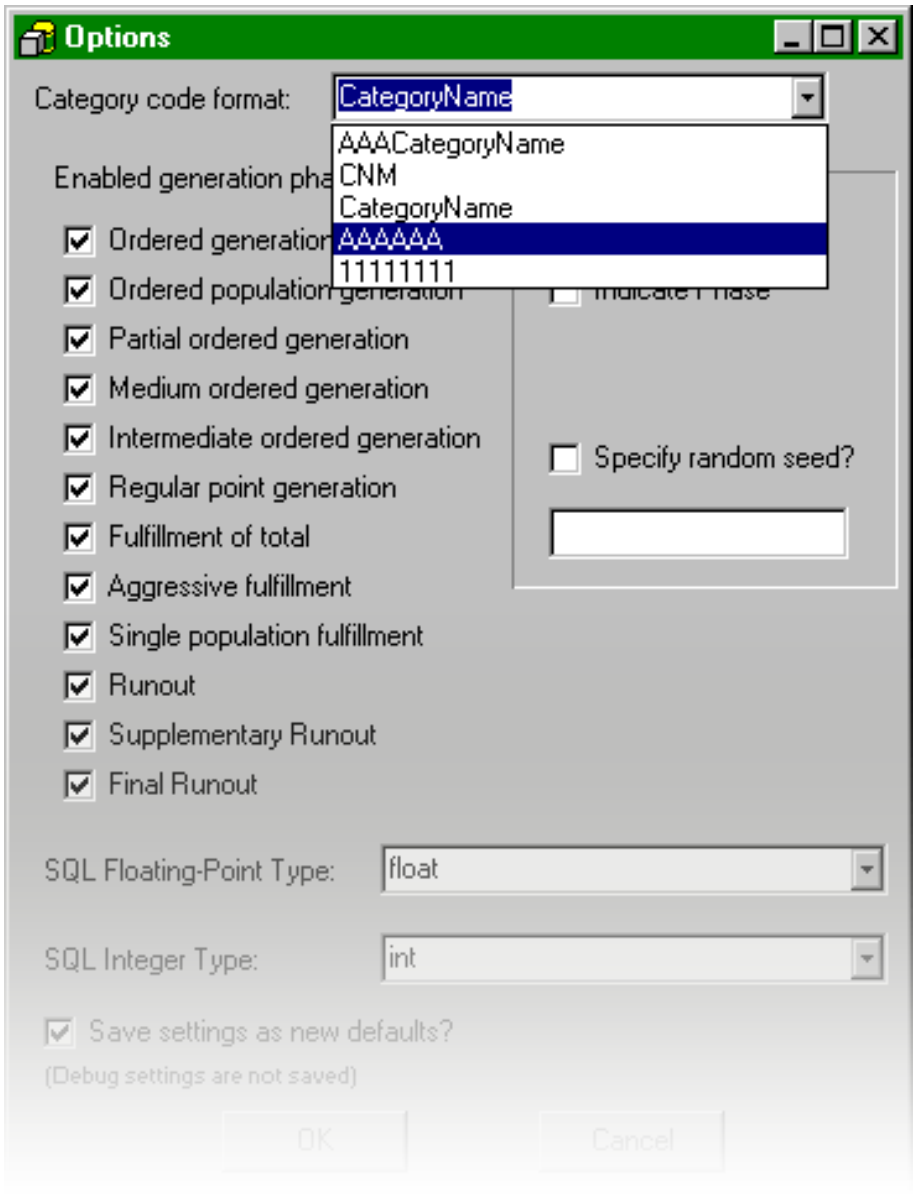
Appendix

Appendix: Point Generation Options

Previewer also offers the choice of setting a number of options on a special dialogue box:



With some troublesome models, it might be necessary to change the format of category codes to ensure that every category has a unique code. The possible formats for category codes are illustrated below:



Because of the complexity of Previewer point generation, the possibility of an error during point generation, such as overflow or division by zero, may possibly arise. However, to ensure reliable operation of Previewer, while it does not continue generating points after such an occurrence, an error-trapping mechanism ensures the program will still continues to run and closes the output file properly. Also, a series of diagnostic messages appears in the message area on the main point generation dialog box. The Debug Log option can be unchecked to save time by avoiding the preparation and buffering of these diagnostic messages.

When such an error occurs, the initial seed used for random number generation is reported; this can be specified here to allow problems to be reproduced.

Note that there is a series of 11 point generation phases which may be individually enabled or disabled, all of which are enabled by default. Although they are usually all enabled, they will not necessarily all be used in the generation of any one cube.

Previous versions of Previewer operated by going through the possible combinations of categories in order, which corresponds to the generation phase called 'Ordered generation'; then, they searched in each dimension for categories which still contained any unallocated quantities in any of the measures and output these so that the totals would come out right, and this is what the term 'Runout' refers to.

The first phase, Ordered Generation, operates by cycling through the categories of each dimension in order, and producing, in each point, a value for each measure based on the product of the percentages in the applicable categories.

This will invariably produce a cube with a high degree of regularity. A cube is regular when the distribution of a measure between the categories of one dimension remains the same when one is looking at either the overall distribution, or the distribution subject to a constraint on the categories permitted in other dimensions.

In order to produce a cube with a more realistic appearance, without manipulating internal arrays which are two-dimensional, involving the categories in more than one dimension at a time, we have chosen to use the technique of dividing the totals for the measures between populations, each of which has an altered set of percentages, so that the overall total of all the populations remains the same as the original. Populations are used in all the phases from "Ordered Population Generation" to "Aggressive Fulfillment".

Also, it requires the generation of a point for every possible combination of categories, one for each dimension. For some models, this may be an impractical number of points to generate.

The first point generation method developed for this version of Previewer, designated Regular Point Generation, allows a cube to be generated for a model with a considerably smaller number of points. This will make it impossible for the resulting cube to exhibit full regularity when all dimensions are constrained to a single category, since many combinations of categories will not have points generated for them, and will hence receive no part of the total for any measure, but the cube could still be monotonously regular when viewed under broader constraints.

To allow the highest possible regularity to be achieved when the number of points to be generated is not enough to allow the Ordered Generation and Ordered Population Generation phases to be used, the phases Partial Ordered Generation, Medium Ordered Generation, and Intermediate Ordered Generation were added.

Partial Ordered Generation runs through categories in order, like Ordered and Ordered Population, but it skips a fraction of the possible combinations at random, and thus is used when the number of points allowed falls slightly short of that required for Ordered Generation. Medium Ordered Generation runs through the categories of two dimensions at a time, choosing the categories from other dimensions for a point at random, for every possible pair of dimensions; Intermediate Ordered Generation does the same for only the two dimensions with the largest number of categories.

The last five phases in this list, Fulfillment of Total, Aggressive Fulfillment, Single Population Fulfillment, Runout, and Supplementary Runout are aimed at achieving correct totals in an efficient manner. The original Previewer, when performing runout, simply started with the first category in each dimension, and proceeded to whichever category contained a leftover quantity for a given measure; in this version, additional complications are introduced to reduce the systematic bias in the output cube that this might introduce.
