Independent Statistics & Analysis
U.S. Energy Information Administration

November 2015

Energy Information Administration Standard 2015-1 Model Documentation and Archiving

Energy Information Administration Standard

Purpose: To ensure that the procedures, equations, and assumptions that define EIA models are publicly available.

Applicability: All models developed and maintained by EIA or its contractors. (In a large model or system, separable components or modules may be considered as individual models for documentation purposes.)

Section 2015-1.1. Model documentation requirements

1. Model documentation should correspond to a specific archived version of the model.
2. The documentation should include the required components listed in the Model Documentation Checklist and additional information as the Office of Energy Analysis (OEA) deems appropriate.
3. To the extent possible under the existing resource constraints, the documentation should be prepared in accordance with the "Guidelines for Mathematical Specification in Model Documentation."
4. The documentation will be available in the standard format determined by OEA. A Word template is available from OEA or the Office of Communications (OC).
5. All documentation should undergo a quality assurance review by OEA prior to dissemination. In addition, new or significantly revised model documentation may undergo reviews by other EIA offices and/or independent expert reviewers from outside EIA when appropriate.
6. When a product using model outputs is sent to the Administrator for approval, the responsible Office Director should specify what documentation is available. If up-to-date documentation is not available, the responsible program office should provide a schedule whereby it will complete the documentation within 90 days of the product's release. If the documentation cannot be completed within this time frame, the program office may request an exception.
7. The most current model documentation should be available on EIA's Website. Documentation related to previous versions of a model may also be available as part of a model archive package (see Section 2015-1.3).

Guidelines for mathematical specifications in model documentation

1. Text of the mathematical specification

a) The mathematical specification of the model should be an unambiguous formal statement of the modeler's concept of the methodology and structure to represent real world phenomena. Given the description of the model's methodology and structure along with knowledge of the model's inputs and transformations, the reader should be able to form an intuitive

understanding of the model rationale, the specific interrelationships and assumptions represented, and the essential model outputs.

b) Portions of the mathematical specifications can be included in appendices; the text and the appendices jointly should provide a complete specification of the model. In practice, an interested person may investigate the documentation and computer code as well as discuss the model with EIA staff to more fully understand the model's intricacies.

c) When standard, commonly used algorithms are applied, references to published literature may be provided. Standard methods used in the code need not be described in complete detail, but the documentation should convey the intuitive rationale underlying these methods.

d) Documentation should be written in the indicative mood (e.g., "the subroutine calculates the parameters"). Occasional lapses into the imperative mood (e.g., "calculate the parameters") give the reader the impression that computer programming specifications have been pasted into the documentation without proper adaptation.

2. Documenting optimization methods employing iterative techniques

a) In cases where an optimization problem is stated, or an iterative process is used to find a solution of some mathematical problem, it is important to state precisely what optimization problem is being solved, or to state precisely, in mathematical terms, the solution to be obtained when the iterative process is complete.

b) Where iterative processes are used, the basic solution methodology should be described. If there are key parameters, such as a damping parameter, that are involved in the iterative process, then these should be described and the relevant equations containing these parameters given.

c) The mathematical specification in the documentation need not deal rigorously with the possible problems of non-convergence or multiple solutions. However, if an iterative procedure in the code has a default condition which allows the computation of the program to continue even if convergence is not attained, then this condition should be stated.

d) If a linear program problem is of such a size and complexity that it is impractical to include the complete specification, the documentation should provide a description of the structure of the problem in as much detail as practical.

e) The documentation should provide all information needed by an expert to access electronically any desired coefficient or constant in the problem, to determine the dimensions of the objective function vector and constraint matrix, to identify which constraints involve strict or non-strict inequality, and to obtain all information necessary to generate the results of the optimization problem.

3. Presenting regression diagnostics

a) In cases where parameters are estimated by statistical regression (ordinary least squares or a similar method), the parameter estimates, R2 statistic, *t*-statistics, and other key diagnostics may be presented in one of two ways:
   i. A discussion may be included in the documentation, or
   ii. An output listing from a statistical or econometric package may be presented. The output should be clearly labeled and modified, if necessary, to identify the regression model and the particular parameters for which the diagnostics are provided. Explanatory lines within the package output should be added, as needed, for clarity.

b) When the regression is based on time series data, a unit root test statistic (e.g., Dickey-Fuller) and/or residual autocorrelation test statistic (e.g., Durbin-Watson) should be reported, along with an associated *p*-value.

4. Mathematical equations

a) To the extent possible, each equation should appear immediately before or after its own explanatory text.

b) When a variable that has been defined is used again in an equation that appears more than two pages after the definition, the definition should be repeated or a page number or equation number for the definition should be given (e.g., "where x is as defined on page n" or "where x is as defined in equation n"). Alternatively, the documentation can include a single comprehensive, alphabetized list of variables with definitions.

c) When numbers (e.g., estimated parameter values, unit conversion factors) are inserted into an equation, their meaning should be explained in the text.

d) All the fonts, including subscript fonts, used in the equations should be large enough to be easily read, even if this means breaking some equations into multiple lines. (Font sizes below 6 pt. should be avoided.) Ideally, the font used for full-sized symbols in the equations should be at least as large as the font used in the text. Second and subsequent lines of one equation should be indented to indicate that the equations are continued from a previous line.

5. Subscripts and indices

a) Subscripts or indices should be used to indicate all index variables on which each variable depends, e.g., month or year for time series variables, region for geographically defined variables.

b) A subscript may be suppressed in an equation only if

   i. it indexes all variables in the equation and

   ii. the text clearly states that the subscript has been suppressed to provide a cleaner notation.

   For example, when the documentation describes a long series of computations performed within a loop over certain variables (e.g., year, region, industry), suppressing the subscripts that define the loop is often convenient and improves readability. Before the first equation in which subscripts are to be suppressed, the text should state, "In equations x through y, the subscripts a,…, z have been suppressed for a cleaner notation," where x and y are the numbers of the first

and last equations, respectively, in which the suppression occurs, and a,..,z are the suppressed subscripts. After equation y, the text should remind the reader that the subscripts have been suppressed in the previous equations and state that, unless otherwise indicated, the suppression will not occur in subsequent equations.

c) If the expression on the right-hand side of an equation explicitly depends on certain indices, the indices must also appear on the left-hand side, unless they are being aggregated out. For example, we may have

$$a_i = \sum_{j=1}^{n} a_{ij}, \quad (1)$$

but we cannot have

$$a_i = b_{ij} c_k. \quad (2)$$

Rather, (2) must be written

$$a_{ijk} = b_{ij} c_k. \quad (3)$$

d) The subscripts that appear on a variable in its definition should be consistent with those that appear on the same variable in the equation in which the variable is used. The variable definition should include the definitions of the subscripts (e.g., $S_{t,r,y}$ = average stock of vehicles of type t in region r during year y).

6. Relating the mathematical specification to the computer code

a) Often a single mathematical equation in the documentation is equivalent to several computer code statements. Certain "temporary variables" that are used in the code to temporarily store values need not be discussed in the documentation. In some cases, words rather than a mathematical expression can be used to specify simple computations, but only if it is clear how the stated ideas would be expressed mathematically.

b) When parallel computations are performed on several variables, one equation in the documentation can describe several lines of computer code. In such cases, the documentation should clearly describe the several instances to which this general equation applies, and relevant subscripts or arguments for representing the various categories should be clearly defined.

c) Variable names used in the computer code are often too long to be convenient for use in the documentation. When a variable name in the computer code differs from the name used in the documentation, the documentation should provide a cross reference list. (Computer code variables that have no exact counterparts in the documentation, such as temporary variables, need not be cross referenced.) Cross referencing can be accomplished in one of two ways:

i. A cross-reference table may be included in the documentation, or

ii. When there is a declaration in the computer code of a variable which is the equivalent of a documentation variable, a comment in the code may indicate the name of the equivalent variable in the documentation, in the distinctive form {vn}, where vn represents the variable name in the documentation. The distinctive form makes it easy to locate the names electronically. (If another distinctive form is used, then the documentation should state the form.) Comments may also be placed next to the variable names (e.g., a FORTRAN COMMON block) or in another location in the code.

For a given model, the responsible program office should choose either (i) or (ii) above. (When a program office decides to switch options, a combination of (i) and (ii) may be used for a limited time during the transition period. This situation should be explained in a note accompanying the variable cross reference list.)

In some cases, one documentation variable name might correspond to several computer variable names. In some circumstances, for example, the computer code might use two different variables that have the same definition but are in different units. If one variable name is used in the documentation to represent multiple variables in the computer code, an explanation should be included in the variable cross reference list.

Section 2015-1.2. Model documentation components checklist

Elements are required unless "optional" is specified. Materials need not be presented in the order discussed here. Not all of the information included in a single item below need appear together as a unit; the information may appear in different sections, if this arrangement improves the flow of the text.

a) A reference to the appropriate archive package.

b) A high level description of changes from the previous archived version. (This may be omitted in cases of one-time reports or in the case of substantial changes from the previous archived version.)

c) Model overview: A concise description of the model, its purposes and uses, how it generates forecasts, critical assumptions, and a discussion of any significant departures from accepted theory or practice.

d) Process flow diagram (optional): A flowchart showing the sequencing of the data inputs, calculations (processes), and outputs of the model.

e) Mathematical specifications: The equations representing the computations performed in the model. The relevant equations include those used to transform input data and parameters into model data and parameters, as well as equations that characterize the solutions of algorithms. For linear programming models, for example, the relevant equations include the objective technology matrix and constraint matrix. For additional guidelines, see the "Guidelines for Mathematical Specifications in Model Documentation."

f) Variable and parameter definitions: The following should be included for all variables and parameters used in the documentation:

   i. Clear definitions, including data sources for the input variables and parameters.
   ii. Units of measurement. The reader should be able to see (though it may take some investigation in some cases) that the units on the right-hand side of an equation are the same as those on the left-hand side.

   iii. Whether each variable is an input, output, or only used in intermediate calculations. In many cases, this will be clear from the text of the documentation.
   iv. Whether data elements are direct inputs to the model or only used in preliminary calculations, e.g., to estimate fixed parameters for input to the model. If data were transformed or manipulated prior to use, an explanation should be provided.

   Computer-code variables that have no counterparts in the model documentation, such as temporary computer-code variables or variables used only in debug statements, need not be defined in the documentation. Their definitions should be included as comments in the computer code.

g) Model estimation procedures: The methods and data sources used to estimate parameters and other quantities in the model should be identified. Enough information about the estimation techniques should be given to allow an expert to exactly reproduce the estimation results. This includes a precise citation of data sources, the data series used, and an exact description of which portions of the data series are used in each calculation.

h) Existence and uniqueness of solutions (optional): For iterative or optimization problems, in cases where the existence or uniqueness of a solution has been demonstrated by analytical means, a description may be presented. Tests from different initial value conditions should be conducted to provide evidence of the uniqueness of solutions.

i) Sensitivity analysis (optional): Tests should be performed to determine whether changes in key model inputs cause key model outputs to respond in a sensible fashion. If a sensitivity analysis is performed, the results of the most recent analysis should be included in the documentation or a reference should be given to an information product that provides these results.

Section 2015-1.3. Model archiving

Purpose: To ensure that EIA model calculations are reproducible.
Applicability: All models used by EIA.

Required actions:

1. A model archive package should be prepared by the program office when model outputs are used in an EIA product that is publicly disseminated.
2. A model archive package should contain the following:

   a. All source code and program control files needed to compile, link, and execute the reference or base case scenario of the model. If alternate source code versions were

used to run the model for other scenarios cited in the same product, these versions should also be provided unless a scenario-specific archive has been created.

b. Input data files used by the model for the reference or base case cited in the model, along with other file versions that are needed to run the model for other scenarios cited in the same product. The data values should be provided in an accompanying computer file or files (e.g., ASCII, spreadsheet, or database files may be used). Each row and column of the data file should be labeled.

c. Primary output (as opposed to debugging or trace files) from the reference or base case scenario used in the product. It is not necessary to provide all output files for all scenarios cited in the product, as long as the outputs not provided can be regenerated from runs of the archived model and the primary outputs from the model can be verified against disseminated results.

d. Instructions for compiling and running the model and comparing the results to disseminated results. A description of changes needed to run the alternative scenarios published in the report or to create scenario-specific archives should be included.

e. The source of the proprietary data and software, along with instructions for obtaining these, should be included in the archive instructions. However, an archive package should exclude proprietary data and software used in the model. (In some cases, the program office may choose to offer an alternate version of the model, modified to exclude proprietary data and software. The alternate version will have more limited functionality than the full model used internally by EIA.)

3. The program office should create and verify an archive package within 60 days of disseminating an EIA information product utilizing model outputs.

4. The archive package must be retained until no longer needed for current business. The program offices should consult with the EIA Office of Resource and Technology Management (ORTM) regarding records retention requirements.

5. (Optional) The program office may develop a policy for public dissemination of the archive that addresses such topics as:

    a. The model's transportability.

    b. Additional software required (such as proprietary embedded models) that is not provided by EIA with the model.

    c. EIA's expectations for how a public user will identify the model if the model has been modified by a user from outside EIA.

    d. Limits on EIA support.

    e. Any other issues pertinent to outside use.

Section 2015-1.4. Use of proprietary models

Purpose: To permit the use of proprietary models in EIA modeling systems and in conjunction with EIA products.
Applicability: To all models available to EIA through license, purchase, or subscription.

Required Actions:

1. Every agreement for the acquisition or use of a model should provide for the following:
    a. Publicly available documentation of the model's design, theoretical basis, empirical implementation, and objective capabilities.
    b. A means for EIA to replicate model calculations for a period of three years after each application in an EIA product.
2. For an active EIA model,
    a. The model documentation should be available to the public.
    b. The model version and archive of all model inputs and outputs associated with a disseminated information product must be identified so the results can be replicated.
    c. All changes EIA makes to the model should be documented and archived to EIA standards.
    d. A proprietary model should not be embedded in an EIA modeling system unless the model is commercially available.

Approval Date: September 15, 2015.