Uncertainty Metadata Naming Conventions (UNC) Specification

version draft-v0.2

CoMet Toolkit Team

October 27, 2026

Contents

Uncertainty Metadata Naming Conventions (UNC) 1

- Authors 1
  - Lead Authors 1
  - Contributors 1
- Introduction 1
  - Goals 1
  - Relationship to the CF Conventions 1
  - Philosophy 1
  - Terminology 2
  - Format for Examples 3
- Dataset Structure 3
  - Variables 3
  - Dimensions 4
  - Data Types 4
  - Attributes 4
- Uncertainty Variables 5
  - Assigning Uncertainty Variables 5
  - Coverage Intervals 6
    - Coverage Probability 8
    - Coverage Factor 8
    - Units and Standard Names 9
    - Relative Differences 9
  - Uncertainty Components 10
  - Uncertainty Evaluation 11
  - Compression by Gathering 12
- Probability Distribution 12
- Error Correlation 13
  - Form 1: Error-Correlation Matrices 14
  - Form 2: Parameterised Error-Correlation Structures 16
  - Form 3: Error-Correlation Descriptions 17
  - Comments on Error Correlations 17
- Appendix A: Controlled Vocabularies 17
  - Probability Distributions 18
  - Error-Correlation Structures 18
  - Coverage Probabilities 20
- Appendix B: Changes from v0.1 20

Uncertainty Metadata Naming Conventions (UNC)

Authors

Lead Authors

- Sam Hunt, NPL
- Pieter De Vis, NPL
- Owen Embury, NCEO / University of Reading
- David Hassell, NCAS / University of Reading

Contributors

This version of the specification was developed in collaboration with the Climate and Forecast (CF) conventions community, as part of a proposal to include these conventions within CF (see Relationship to the CF Conventions).

Introduction

Goals

Measurement datasets are becoming larger, more complex, and are increasingly used to support critical applications such as manufacturing, health, and environmental monitoring. Reliable interpretation of these measurements requires accompanying uncertainty and error-covariance information - however, this is often overlooked. Where available, such information lacks standardisation and could, in principle, be highly complex and large.

The goal of this Uncertainty metadata Naming Conventions (UNC) specification is to provide a standardised metadata format for storing the accompanying uncertainty information with measurement datasets. This includes the magnitude of the uncertainty, the probability distribution it describes, and the structure of the error-correlation between the elements of a dataset. This format is intended to support fully capturing the content of the error-covariance matrices associated with measurement data in a compact structure, by parameterising error-correlation with a simple set of metadata.

Relationship to the CF Conventions

Draft v0.2 is a major revision of v0.1. It was developed in collaboration with the CF community, and has been proposed for inclusion in the CF conventions as a new chapter, “Uncertainty” (see CF pull request #661, and the associated discussion in CF issue #660). This version of the specification is aligned with that proposal, and will be kept aligned with it as it progresses through the CF review process.

This specification can be read on its own, but it builds on the wider CF conventions - for example, the definitions of the `units`, `units_metadata` and `standard_name` attributes. Where this specification depends on other parts of the CF conventions, links are given to the relevant sections of the CF conventions document.

Within CF, the uncertainty of a data variable has previously been expressed using an ancillary variable with the `standard_error` standard name modifier (see the Ancillary Data section of the CF conventions). The CF proposal deprecates this approach in favour of the *uncertainty variables* defined in this specification.

The changes from v0.1 are summarised in Appendix B: Changes from v0.1.

Philosophy

This specification is intended to contribute to and build upon an existing ecosystem of standards and best practices. In particular the following are adhered to, to the extent possible:

- The understanding of uncertainty concepts defined in the JCGM GUM (Guide to the expression of uncertainty in measurement) suite of documents.

- The definition of uncertainty-related terminology defined in the JCGM VIM (International Vocabulary of Metrology).
- The NetCDF data model for creating self-describing, array-oriented scientific datasets.
- The Climate and Forecast (CF) conventions on metadata for weather and climate data, with which this specification is designed to be fully consistent.

The work builds on previous work on the standardisation of uncertainty information for climate data developed within the H2020 FIDUCEO project.

Within this context, this specification also attempts to adhere to the following principles:

- **Scalability of Complexity**

  The complexity of the metadata should align with the use case:

  - *Simple use cases* should be achievable with a *simple implementation*.
  - *Complex use cases* should be *possible without unnecessary restrictions*.
- **Minimisation of Redundancy**

  The metadata specification should *avoid duplication* of information to prevent potential inconsistencies.
- **Human and Machine Readability**

  Metadata must be:

  - *Comprehensible* for humans.
  - *Parsable* by machines.

Terminology

The provision of uncertainty metadata in this specification is designed to be consistent, in both scientific content and terminology, with the GUM and the VIM, both published by the Joint Committee for Guides in Metrology (JCGM), which should be considered the canonical references for this specification. This includes the following important terms:

- Measurand
- Measurement error
- Measurement uncertainty
- Standard uncertainty
- Expanded uncertainty
- Coverage interval
- Coverage probability
- Coverage factor
- Type A evaluation and Type B evaluation of measurement uncertainty

Definitions of the following terms are adopted from work undertaken in the H2020 FIDUCEO:

- Error-correlation
- Error-covariance

Measurement, in the context of uncertainty, is the process of obtaining a numerical value that represents a physical quantity (i.e. the measurand). A measured value is typically derived from a numerical simulation (e.g. climate model output), a direct observation (e.g. sensor data), or a remote sensing platform (e.g. satellite retrievals). The error of a measurement is the measured value minus the true, unknowable value of the measurand. The uncertainty of a measurement is a measure of the dispersion of the values that could reasonably be attributed to the measurand.

Format for Examples

The plain text (ASCII) format used to describe the contents of a NetCDF dataset is called CDL (NetCDF Common Data Language). Snippets of CDL are used to present examples in this specification. A minimal example of a measurement dataset in CDL is given below. Here the dataset contains a single `temperature` variable with accompanying metadata (`units` and `long_name`), which is defined along `time`, `lat` and `lon` dimensions (each of which is length 2). Data are omitted in specification examples for brevity.

```
netcdf short_example {
  dimensions:
    time = 2 ;
    lat = 2 ;
    lon = 2 ;

  variables:
    float temperature(time, lat, lon) ;
      temperature:units = "K" ;
      temperature:long_name = "measured temperature" ;

  data:
    temperature =
      290.1, 291.2,
      292.3, 293.4,
      294.5, 295.6,
      296.7, 297.8 ;
}
```

The examples follow the CF conventions, and so include attributes defined by CF (such as `standard_name`, `units` and `units_metadata`) as well as those defined by this specification. In some examples, long attribute values are split across multiple lines to improve readability, whereas the actual encoded value would be a single string with no embedded newlines - such pseudo-CDL will not parse directly.

Dataset Structure

As mentioned above, the NetCDF data model for creating self-describing, array-oriented scientific datasets is adopted. The components of NetCDF datasets are described in Section 2 of the NUG (NetCDF Users Guide). In this section, we introduce the core components of this data model relevant to this specification.

Variables

Datasets are composed of variables, which are multidimensional data arrays.

This specification defines the following categories of variables:

- **Data Variables**

  Data variables represent a multidimensional array of measured values. (These were called *observation variables* in v0.1.)

- **Uncertainty Variables**

  Uncertainty variables characterise a component of the uncertainty of a *data variable*, by defining a coverage interval for its values. A *data variable* may have multiple *uncertainty variables* associated with it, and an *uncertainty variable* may be associated with more than one *data variable*. See Uncertainty Variables.

- **Distribution Parameter Variables**

  Distribution parameter variables provide the values of a parameter (such as skewness) of the probability distribution from which an *uncertainty variable*'s coverage interval is derived. See Probability Distribution.

- **Error-Correlation Variables**

  Error-correlation variables contain an explicit matrix of error-correlation coefficients along one or more dimensions of a *data variable*. See Error Correlation.

- **Error-Correlation Parameter Variables**

  Error-correlation parameter variables are scalar variables that provide the value of a parameter used to configure a parameterised error-correlation structure. See Error Correlation.

A dataset may also contain variables that do not fall into any of these categories.

Dimensions

A variable may have any number of named dimensions, including zero – e.g., `"x"`, `"y"`, `"time"`. Dimensions may be of any size, including 1.

Uncertainty variables may have all or a subset of the dimensions of their *data variable*, and some also have an additional size-two *interval dimension*, used to store the limits of an asymmetric coverage interval (see Coverage Intervals).

Data Types

Data variables and *uncertainty variables* must have a numeric data type.

Note: these variables may be encoded as e.g. integers for efficient storage on disc (see the Packed Data section of the CF conventions).

Attributes

Dataset attributes provide metadata about the dataset, its variables, and dimensions. Global attributes describe the entire dataset (e.g., title, institution, history). Variable attributes define specific properties of the variable (e.g., units, valid ranges). These attributes ensure data is interpretable, support automated processing, and facilitate sharing by following standardised conventions.

This specification defines a set of variable attributes to:

- link *data variables* with their associated *uncertainty variables*
- define how the values of an *uncertainty variable* characterise the uncertainty of its *data variable*
- define the error-correlation properties of a given *uncertainty variable* in a compact way.

These attributes are summarised in the table below. A dataset may also contain other unrelated attributes.

Summary of attributes

Attribute	Type	Used on	Requirement	Description
`uncertainty_variables`	string	data variable	Optional	Identifies the *uncertainty variables* of the *data variable*. See Uncertainty Variables.
`coverage_interval`	string	uncertainty variable	Required	Specifies how the *uncertainty variable* array defines the coverage interval. See Coverage Intervals.
`coverage_probability`	numeric	uncertainty variable	Required	Contains the probability that the coverage interval contains the true *data variable* value. See Coverage Probability.

`uncertainty_representation`	string	uncertainty variable	Optional (default `difference`)	Specifies whether the *uncertainty variable* array represents absolute or relative differences to the *data variable* values. See Relative Differences.
`probability_distribution`	string	uncertainty variable	Required if `coverage_interval` is `standard_deviation`, otherwise optional	Describes the mathematical form of the probability distribution from which the coverage interval is derived. See Probability Distribution.
`coverage_factor`	numeric	uncertainty variable	Optional (default 1), only if `coverage_interval` is `standard_deviation`	Gives a positive numerical coverage factor for the coverage interval. See Coverage Factor.
`uncertainty_evaluation`	string	uncertainty variable	Optional	Describes the method used to evaluate the uncertainty. See Uncertainty Evaluation.
`uncertainty_component`	string	uncertainty variable	Optional	Describes the component of the uncertainty represented by the *uncertainty variable*. See Uncertainty Components.
`error_correlation`	string	uncertainty variable	Optional	Describes the correlations of errors across the domain. See Error Correlation.

Uncertainty Variables

An *uncertainty variable* characterises a component of the uncertainty of a *data variable*, by defining a coverage interval for the *data variable*'s values. Its attributes define how its array values characterise the uncertainty. Two of these attributes, `coverage_interval` and `coverage_probability`, must be set on all *uncertainty variables*, with the others being optional, or conditionally optional depending on the value of another attribute (see Attributes). The `probability_distribution` and `error_correlation` attributes, which can describe the uncertainty in more detail, are described in Probability Distribution and Error Correlation.

Assigning Uncertainty Variables

When a variable provides metadata about the uncertainty in the measurement of a quantity described by a *data variable*, this association is expressed by providing a link between the variables. The attribute `uncertainty_variables` is used to express these types of relationships. It is a string attribute whose value is a blank separated list of variable names, for which the order is not significant. The nature of the relationship between a *data variable* and its *uncertainty variables* (i.e. those named by its `uncertainty_variables` attribute) is determined by the *uncertainty variables'* attributes. See Example 1.

Example 1. Assigning uncertainty variables

```
dimensions:
  time = 30 ;

variables:
  // Data variable
```

```
float q(time) ;
  q:standard_name = "specific_humidity" ;
  q:units = "g/g" ;
  q:uncertainty_variables = "u_noise u_calibration" ;
// Uncertainty variables
float u_noise(time) ;
  u_noise:standard_name = "specific_humidity" ;
  u_noise:units = "g/g" ;
  u_noise:coverage_interval = "standard_deviation" ;
  u_noise:coverage_probability = 0.6827 ;
  u_noise:probability_distribution = "gaussian" ;
float u_calibration ;
  u_calibration:standard_name = "specific_humidity" ;
  u_calibration:units = "g/g" ;
  u_calibration:coverage_interval = "half_width" ;
  u_calibration:coverage_probability = 0.95 ;
```

The *data variable* `q` is linked to two *uncertainty variables*, `u_noise` and `u_calibration`, by its `uncertainty_variables` attribute (the order of the names is not significant). Each *uncertainty variable* characterises a different component of the uncertainty in q, as defined by its own attributes. `u_noise` gives the standard uncertainty of each element of `q` along the `time` dimension, whereas `u_calibration` gives the half-width of a coverage interval with a coverage probability of 0.95, as a single value that applies to every element of q (see Coverage Intervals).

Coverage Intervals

Uncertainty is expressed as a statistical coverage interval (hereafter referred to as a “coverage interval”) derived from the probability distribution of the measurand (hereafter referred to as the “probability distribution”) that contains the true value with a given probability, i.e. the range of values within which the true value is expected to lie. For instance, for a measured value of 50 there could be a probability of 0.95 that the true value lies within the coverage interval [40, 60]. This concept is applicable to any probability distribution, including Gaussian, non-Gaussian, and asymmetric distributions. For example, when a coverage interval is defined for a skewed probability distribution, the interval limits would typically be asymmetric about the measured value.

The coverage interval, which may vary across the *data variable*’s domain, is defined by the array of an *uncertainty variable* with a numeric data type; and the probability that the true value lies inside the coverage interval is stored by the *uncertainty variable*’s `coverage_probability` attribute (see Coverage Probability).

In general, an *uncertainty variable* must have all or a subset of the dimensions of the *data variable* to which it is related, with the addition of a size-two *interval dimension* for defining the coverage interval limits (but the interval dimension may be omitted in the common case of the coverage interval being symmetrical about the *data variable* values, as described below). These are stored as difference offsets to the *data variable* values (but relative differences may also be provided, see Relative Differences). The coverage interval limits are reconstructed by adding the two offsets to the *data variable* values. The interval dimension must be the most rapidly varying dimension (the last dimension in CDL order), with the first element representing the coverage interval’s lower limit and the second representing the coverage interval’s upper limit. The order of other dimensions is not restricted, and does not have to be the same as for the parent *data variable*. For instance, for a *data variable* containing a measured value of 30 there could be a probability of 0.9 that the true value lies within the coverage interval [25, 40]. The coverage interval offsets could be stored in an *uncertainty variable* array as (–5, 10), and adding the *data variable* value of 30 to these values reconstitutes the actual coverage interval of [25, 40].

When the coverage interval is symmetrical about the *data variable* values (i.e. the *data variable* values lie at the midpoint of the coverage interval), the coverage interval may be fully defined by a single non-negative value. This value is equal to the absolute difference between the *data variable* value and the coverage interval’s lower or upper limit. The coverage interval limits are reconstructed by subtracting the half-width from, and adding it to, the *data variable* value. This case is indicated by specific values of the `coverage_interval` attribute (such as `standard_deviation` and `half_width`), for which the size-two interval dimension is not required and must be omitted. For instance, for a *data variable* containing a measured value of 20 there could be a probability of 0.95 that the true value lies within the coverage interval [15, 25]. This coverage interval is symmetrical around the *data*

variable value, so the half-width could be stored in an *uncertainty variable* array as 5, and subtracting this from, and adding it to, the *data variable* value of 20 reconstitutes the actual coverage interval of [15, 25].

The `coverage_interval` attribute specifies how the *uncertainty variable* array defines the coverage interval. The attribute must be set, and takes one of the following values:

`standard_deviation`
: The array has no interval dimension and contains the coverage interval's non-negative half-width that is equal to the standard deviation of the probability distribution defined by the `probability_distribution` attribute, which must also be set. One standard deviation of the probability distribution is also known as the "standard uncertainty". However, if the `coverage_factor` attribute has a value other than 1, then the values in the *uncertainty variable* array are instead one standard deviation multiplied by this amount (see Coverage Factor).

`half_width`
: The array has no interval dimension and contains the coverage interval's non-negative half-width.

`offsets`
: The array has a size-two interval dimension that contains the offsets which define the lower and upper limits of the coverage interval.

Example 2. Standard uncertainty

```
dimensions:
  time = 30 ;

variables:
  // Data variable
  float q(time) ;
    q:standard_name = "specific_humidity" ;
    q:units = "g/g" ;
    q:uncertainty_variables = "q_uncertainty" ;
  // Uncertainty variable
  float q_uncertainty(time) ;
    q_uncertainty:standard_name = "specific_humidity" ;
    q_uncertainty:units = "g/g" ;
    q_uncertainty:coverage_interval = "standard_deviation" ;
    q_uncertainty:coverage_probability = 0.6827 ;
    q_uncertainty:probability_distribution = "gaussian" ;

data:
  q = 0.024, 0.021, 0.046, ... ;
  q_uncertainty = 0.003, 0.001, 0.002, ... ;
```

The *uncertainty variable* `q_uncertainty` contains the standard deviation (σ, the standard uncertainty) of the Gaussian distribution that characterises the errors in the *data variable* q. The probability that the true values of q are within σ of its actual array values is 0.6827. For instance, for the first element of the `time` dimension the measured value of specific humidity is 0.024, and there is a 68.27% chance that its true value lies in the range [0.021, 0.027].

Example 3. Asymmetric uncertainty coverage interval

```
dimensions:
  time = 30 ;
  interval_offset = 2 ;

variables:
  // Data variable
  float q(time) ;
```

```
    q:standard_name = "specific_humidity" ;
    q:units = "g/g" ;
    q:uncertainty_variables = "q_uncertainty" ;
  // Uncertainty variable
  float q_uncertainty(time, interval_offset) ;
    q_uncertainty:standard_name = "specific_humidity" ;
    q_uncertainty:units = "g/g" ;
    q_uncertainty:coverage_interval = "offsets" ;
    q_uncertainty:coverage_probability = 0.9 ;

data:
  q = 0.024, 0.021, ... ;
  q_uncertainty = -0.006, 0.003,
                  -0.002, 0.004,
                  ... ;
```

The *uncertainty variable* `q_uncertainty` contains a coverage interval defined by lower and upper limits. The probability that the true values of the *data variable* q are within the coverage interval limits is 0.9. For instance, for the first element of the `time` dimension the measured value of specific humidity is 0.024, and there is a 90% chance that its true value lies in the range [0.018, 0.027]. The probability distribution of the errors is unspecified.

Coverage Probability

The `coverage_probability` attribute contains the probability, in the range 0 to 1, that the coverage interval contains the true *data variable* value. The attribute must be set. For convenience, Table A.5 and Table A.6 provide the coverage probabilities for coverage intervals based on the standard uncertainty (i.e. one standard deviation either side of the mean of the distribution), and selected expanded uncertainties, of the Gaussian and uniform distributions.

Coverage Factor

When the `coverage_interval` attribute is `standard_deviation`, the `coverage_factor` attribute gives a positive numerical “coverage factor” for the coverage interval. For values other than 1, the *uncertainty variable* represents an “expanded uncertainty”, i.e. one for which the values in the *uncertainty variable* array are in fact the standard deviation of the probability distribution multiplied by the coverage factor. The standard deviation of the probability distribution (i.e. the standard uncertainty) may be calculated by dividing the *uncertainty variable* array values by the coverage factor. The attribute is optional with a default value of 1, and must not be set if the `coverage_interval` attribute is not `standard_deviation`.

Example 4. Expanded uncertainty

```
dimensions:
  time = 30 ;

variables:
  // Data variable
  float q(time) ;
    q:standard_name = "specific_humidity" ;
    q:units = "g/g" ;
    q:uncertainty_variables = "q_uncertainty" ;
  // Uncertainty variable
  float q_uncertainty(time) ;
    q_uncertainty:standard_name = "specific_humidity" ;
    q_uncertainty:units = "g/g" ;
    q_uncertainty:coverage_interval = "standard_deviation" ;
    q_uncertainty:probability_distribution = "gaussian" ;
    q_uncertainty:coverage_probability = 0.9545 ;
```

```
    q_uncertainty:coverage_factor = 2 ;

 data:
   q = 0.024, 0.021, 0.106, ... ;
   q_uncertainty = 0.006, 0.002, 0.004, ... ;
```

The *uncertainty variable* `q_uncertainty` contains two standard deviations (2σ) of the Gaussian distribution that characterises the errors in the *data variable* `q`. The probability that the true values of `q` are within 2σ of its actual array values is 0.9545. For instance, for the first element of the `time` dimension the *data variable* value is 0.024, and there is a 95.45% chance that its true value lies in the range [0.018, 0.030].

Units and Standard Names

An *uncertainty variable* containing difference offsets represents the same physical quantity as the parent *data variable*, and so its units must be physically equivalent to the *data variable* units (not necessarily identical, see the Units section of the CF conventions). If the *uncertainty variable* has a `standard_name` attribute then it must be identical to the `standard_name` attribute of the parent *data variable*. However, if the *uncertainty variable* has no standard name then it may be associated with any appropriate parent *data variable* with physically equivalent units. Note that there are different rules regarding the `units` and `standard_name` attributes of an *uncertainty variable* that contains relative differences (see Relative Differences).

Relative Differences

The coverage interval limits may alternatively be defined as relative differences (i.e. fractional or percentage differences, as opposed to difference offsets) to the *data variable* values. This is specified by the `uncertainty_representation` attribute, which takes one of the following values:

`difference`

This is the default. The array contains values that are differences to the *data variable* values.

`relative_difference`

The array contains values that are fractional or percentage relative differences to the *data variable* values.

For a symmetric coverage interval defined by a single non-negative number (i.e. when the `coverage_interval` attribute is `standard_deviation` or `half_width`), the limits are reconstituted by multiplying the *data variable* value by the factors `1 - f` and `1 + f`, where `f` is the relative offset uncertainty array value expressed as a fraction. If the uncertainty is provided as a percentage, `f` is found by dividing the uncertainty array value by 100. For instance, for a *data variable* containing a measured value of 20 there could be a probability of 0.95 that the true value lies within the symmetrical coverage interval [15, 25], and the half-width could be stored in an *uncertainty variable* array as the relative values 25% or 0.25.

For a coverage interval defined by two offsets stored in an interval dimension (i.e. when the `coverage_interval` attribute is `offsets`), the limits are reconstituted by multiplying the *data variable* value by the factors `1 + f` and `1 + g`, where `f` and `g` are the two relative offset uncertainty array values expressed as fractions. For instance, for a *data variable* containing a measured value of 20 there could be a probability of 0.9 that the true value lies within the asymmetric coverage interval [15, 30], in which case the coverage interval offsets could be stored in an *uncertainty variable* array as the relative values (–25%, 50%) or (–0.25, 0.5).

An *uncertainty variable* containing relative difference offsets does not represent the same physical quantity as the parent *data variable*, and must not have a `standard_name` attribute. It must have dimensionless units or units of percent, for fractional and percentage relative differences respectively.

The *uncertainty variable* containing relative difference offsets may be associated with any number of parent *data variables*, but it is not recommended to use relative differences for a *data variable* whose array contains values of zero, or has on-scale units with a shifted origin (such as on-scale Celsius, see the Temperature Units section of the CF conventions), because there is a possibility that the coverage intervals reconstructed by the data reader are not those intended by the data writer. In the case of a *data variable* value of zero, relative difference offsets always result in a zero-width coverage interval; and in the case of on-scale units with a shifted origin, the calculated coverage interval depends on whether the given *data variable* units are used in the calculation, or if equivalent units with the origin unshifted are used instead.

Example 5. Uncertainty defined as relative differences

```
dimensions:
  time = 30 ;

variables:
  // Data variables
  float skin(time) ;
    skin:standard_name = "sea_surface_skin_temperature" ;
    skin:units = "K" ;
    skin:units_metadata = "temperature: on_scale" ;
    skin:uncertainty_variables = "uncertainty" ;
  float subskin(time) ;
    subskin:standard_name = "sea_surface_subskin_temperature" ;
    subskin:units = "K" ;
    subskin:units_metadata = "temperature: on_scale" ;
    subskin:uncertainty_variables = "uncertainty" ;
  // Uncertainty variable
  float uncertainty ;
    uncertainty:units = "%" ;
    uncertainty:uncertainty_representation = "relative_difference" ;
    uncertainty:coverage_interval = "half_width" ;
    uncertainty:coverage_probability = 0.9 ;

data:
  skin = 283.15, 284.15, 282.9, ... ;
  subskin = 284.5, 283.9, 283.0, ... ;
  uncertainty = 1.0 ;
```

The *uncertainty variable* `uncertainty` contains the half-width of the coverage interval that characterises the errors in both the *data variables* `skin` and `subskin`, expressed as a relative percentage difference. For each element of the `time` dimension there is a 90% chance that the true value lies within 1% of its *data variable* value. For instance, the first element of the `skin` variable is 283.15 and has a coverage interval of [280.3185, 285.9815]; and the first element of the `subskin` variable is 284.5 and has a coverage interval of [281.655, 287.345].

Uncertainty Components

When multiple *uncertainty variables* are given, each stores a component of uncertainty, and the definition of the coverage interval and coverage probability may differ across the components.

The `uncertainty_component` attribute describes the component of the uncertainty represented by the *uncertainty variable*. In general, the description is given as non-standardised text (e.g. `Uncertainty correlated over synoptic scales`). The special value of `combined_uncertainty` indicates that the *uncertainty variable* represents the combined uncertainty (i.e. the total uncertainty in the *data variable* arising from all uncertainty components), and at most one of a *data variable*'s *uncertainty variables* can have this attribute value. The attribute is optional and there is no default value.

When there is no `combined_uncertainty` component, it is assumed that the uncertainty components are all independent of each other (i.e. there is no inter-component correlation). When a combined uncertainty component is present, there is no assumption about the independence of the other components.

Example 6. Uncertainty components

```
dimensions:
  time = 30 ;
  lon = 360 ;
  lat = 180 ;
```

```
variables:
  // Data variable
  float sst(time, lat, lon) ;
    sst:standard_name = "sea_surface_temperature" ;
    sst:units = "K" ;
    sst:units_metadata = "temperature: on_scale" ;
    sst:uncertainty_variables = "u_total u_random u_systematic" ;
  // Uncertainty variables
  float u_total(time, lat, lon) ;
    u_total:standard_name = "sea_surface_temperature" ;
    u_total:units = "K" ;
    u_total:units_metadata = "temperature: on_scale" ;
    u_total:coverage_interval = "standard_deviation" ;
    u_total:coverage_probability = 0.6827 ;
    u_total:probability_distribution = "gaussian" ;
    u_total:uncertainty_component = "combined_uncertainty" ;
  float u_random(time, lat, lon) ;
    u_random:standard_name = "sea_surface_temperature" ;
    u_random:units = "K" ;
    u_random:units_metadata = "temperature: on_scale" ;
    u_random:coverage_interval = "standard_deviation" ;
    u_random:coverage_factor = 2 ;
    u_random:coverage_probability = 0.9545 ;
    u_random:probability_distribution = "gaussian" ;
    u_random:uncertainty_component = "Uncorrelated" ;
  float u_systematic(lat, lon) ;
    u_systematic:standard_name = "sea_surface_temperature" ;
    u_systematic:units = "K" ;
    u_systematic:units_metadata = "temperature: on_scale" ;
    u_systematic:coverage_interval = "standard_deviation" ;
    u_systematic:coverage_probability = 0.6827 ;
    u_systematic:probability_distribution = "gaussian" ;
    u_systematic:uncertainty_component = "Correlated over synoptic scales" ;
```

Multiple uncertainty components, including the combined uncertainty, described by their `uncertainty_component` attributes. Note that the definition of the coverage interval and coverage probability may differ between uncertainty components. Note that not all *uncertainty variables* need to have the same set of dimensions.

Uncertainty Evaluation

The `uncertainty_evaluation` attribute describes the method used to evaluate the uncertainty, one of:

`statistical`
: Based on statistical analysis, i.e. obtained from a probability density function derived from an observed frequency distribution, such as a coverage interval derived from the standard error of the mean of an original sample. This is also known as a “Type A evaluation” (GUM).

`scientific_judgement`
: Based on scientific judgement, i.e. obtained from an assumed probability density function, such as a coverage interval derived from an instrument manufacturer’s accuracy specification. This is also known as a “Type B evaluation” (GUM).

`mixed`
: Based on a combination of statistical analysis and scientific judgement, i.e. a combination of the `statistical` and `scientific_judgement` methods.

The `uncertainty_evaluation` attribute is optional, and there is no default value. Note that an evaluation method does not imply anything about the nature of the uncertainty, for instance systematic and random uncertainty can both be evaluated using either the `statistical` or `scientific_judgement` methods.

Compression by Gathering

If an *uncertainty variable* of a *data variable* that has been compressed by gathering (see the Compression by Gathering section of the CF conventions) does not span the compressed dimension, then the subset of the parent *data variable* dimensions is restricted to the *data variable*'s uncompressed dimensions, i.e. it may have any of the dimensions listed by the `compress` attribute of the compressed coordinate variable, as well as any of the other dimensions of the *data variable* (those apart from the compressed dimension).

Probability Distribution

The `probability_distribution` attribute of an *uncertainty variable* describes the mathematical form of the probability distribution from which the coverage interval is derived (i.e. the probability distribution that characterises the dispersion of the values that could reasonably be attributed to the *data variable*). In general the attribute is optional, and there is no default value. However, the attribute must be set if the `coverage_interval` attribute is `standard_deviation`.

The `probability_distribution` attribute is a string attribute of the form: `distribution [(parameter: value [parameter: value ...])]`, where square brackets indicate optional content. The *distribution* keyword identifies the mathematical form of the probability distribution, and the optional *parameter: value* pairs provide parameters that define the distribution.

The *distribution* must be one of the probability distribution names defined in Table A.1 (such as `gaussian`). See Example 2.

A distribution parameter may be defined by a *parameter: value* pair. The type of parameter is named by the *parameter* keyword, and must be one of the distribution parameter names defined in Table A.2 (such as `skew`). The *value* names a *distribution parameter variable* whose data array contains the distribution parameter values, and which may have attributes to describe it (such as `units` or `standard_name`). See Example 7.

A *distribution parameter variable* must have all or a subset of the dimensions of the *uncertainty variable* to which it is related. The order of dimensions of a *distribution parameter variable* is not restricted; it does not have to be the same as for the parent *uncertainty variable*. If a *distribution parameter variable* of an *uncertainty variable* that has been compressed by gathering does not span the compressed dimension (see the Compression by Gathering section of the CF conventions), then the subset of the parent *uncertainty variable* dimensions is restricted to the *uncertainty variable*'s uncompressed dimensions, i.e. it may have any of the dimensions listed by the `compress` attribute of the compressed coordinate variable, as well as any of the other dimensions of the *uncertainty variable* (those apart from the compressed dimension).

Example 7. Uncertainty probability distribution

```
dimensions:
  time = 30 ;
  interval_offset = 2 ;

variables:
  // Data variable
  float q(time) ;
    q:standard_name = "specific_humidity" ;
    q:units = "g/g" ;
    q:uncertainty_variables = "q_uncertainty" ;
  // Uncertainty variable
  float q_uncertainty(time, interval_offset) ;
    q_uncertainty:standard_name = "specific_humidity" ;
    q_uncertainty:units = "g/g" ;
    q_uncertainty:coverage_interval = "offsets" ;
    q_uncertainty:coverage_probability = 0.9 ;
    q_uncertainty:probability_distribution = "skewed_gaussian
```

```
                                                    (skew: skew kurtosis: kurtosis)" ;
  // Distribution parameter variables
  float skew(time) ;
    skew:units = "1" ;
  float kurtosis(time) ;
    kurtosis:units = "1" ;
```

The *uncertainty variable* `q_uncertainty` contains a coverage interval defined by lower and upper limits. The probability that the true values of the *data variable* q are within the coverage interval limits is 0.9. The errors are characterised by a skew-normal distribution, which is further defined at each point by its skewness and kurtosis (i.e. the third and fourth standardised statistical moments), which are provided in separate *distribution parameter variables*.

Error Correlation

The *uncertainty variable* array provides a measure of the magnitude of the error in the *data variable* at each domain location. The error-correlations describe the pattern, independently of the magnitude of the errors themselves, of statistical relationships between the unknown actual errors at different domain locations. For instance, this information can be used for assessing the uncertainty of derived products (such as those formed by averaging, or by subtraction of a statistical norm), and for the weighting of observations when updating numerical models via data assimilation systems.

Error correlation information may be defined by the *uncertainty variable*'s `error_correlation` attribute, which relates *data variable* dimensions to matrices of error-correlation coefficients (possibly in parametric form), or to a non-standardised description of the error-correlation structure, or both. In particular, the parametric form of an error-correlation matrix always names the structural type of the correlations (such as `independent` or `common`), providing physical insight into the error structure. The allowed parametric forms are defined in Appendix A: Controlled Vocabularies.

Error-correlation information may be given for any dimensions of the *data variable*, including any that are not spanned by the *uncertainty variable*. This is because, even though the probability distribution characterised by the *uncertainty variable* does not depend on such dimensions, the error correlations along them may still have different characteristics to those of other dimensions.

The `error_correlation` attribute may be used to describe the correlations of errors across the domain. Error correlations are defined as correlation coefficients which must be in the range –1 (perfect negative correlation) to +1 (perfect positive correlation), and 0 indicates no correlation. The attribute is optional and there is no default value.

In general, error correlations are given in the form of an error-correlation matrix that provides the error-correlation coefficients between all pairs of locations along one or more dimensions of the *data variable* (see form 1, below). However, when the correlation structure is parameterisable, the parameters that fully define the error-correlation matrix may be provided instead, which can not only significantly reduce the dataset size, but also provides physical insight regarding the error-correlation structure (see form 2, below). Non-standardised text describing the error-correlation structure may be used instead of an error-correlation matrix or its parameterised structure (see form 3, below), and also with forms 1 and 2.

Error-correlations are provided instead of any other joint-dispersion measure (e.g. error-covariance) because they allow the pattern of error connectivity to be completely de-coupled from the uncertainty magnitudes. This de-coupling allows the error-correlation structure to be parameterised (see form 2, below), and also allows the same structure to be re-used across multiple *uncertainty variables*. A corresponding joint-dispersion measure that combines the error pattern with the error magnitude may be reconstructed by pre- and post-multiplying the dimensionless error-correlation matrix (which by definition has ones on its main diagonal) with a diagonal matrix of uncertainty magnitudes. For instance, this will be the error-covariance when the uncertainty values are standard deviations of the probability distribution.

The `error_correlation` attribute is a string attribute comprising a list of one or more blank-separated elements, the order of which is not significant. Each element has one of the following forms, where square brackets indicate optional content, and different elements may have different forms:

- **Form 1**. Error-correlation coefficients defined explicitly in an error-correlation matrix:

- **1a**. `dim: error_correlation_var [([comment:] comment)]`
- **1b**. `dim: dim: [dim: ...] error_correlation_var [([comment:] comment)]`

- **Form 2**. Error-correlation coefficients defined implicitly by a parametric structure:
  - **2a**. `dim: [dim: ...] error_correlation_structure [(comment: comment)]`
  - **2b**. `dim: [dim: ...] error_correlation_structure (parameter: value [parameter: value ...] [comment: comment])`
- **Form 3**. Error-correlation coefficients characterised only by non-standardised text:
  - **3a**. `dim: [dim: ...] ([comment:] comment)`

In all forms, *dim* names a dimension of the parent *data variable* to which the error correlation information applies. A dimension can appear at most once anywhere in the `error_correlation` attribute, and not all parent *data variable* dimensions need to be referenced.

Form 1: Error-Correlation Matrices

In **form 1a**, *error_correlation_var* names an *error-correlation variable* that contains a square matrix of numeric correlation coefficients between the errors for every pair of cells indexed by the given dimension, and can be used when the errors for the given dimension are independent of those for all other dimensions. The *error-correlation variable* has the given dimension, and a different dimension of the same size that is not one of the parent *data variable* dimensions. The correlation between the errors at the i-th and j-th positions of the given dimension is stored at the intersection of the i-th row and j-th column of the matrix. Each error-correlation coefficient must be in the range –1 to 1. The matrix must be symmetric and have ones on the main diagonal. An *error-correlation variable* must have dimensionless units. For instance, if the given dimension is `time` with size 360, then `time: varname` means that the *error-correlation variable* `varname` contains a 360 by 360 error-correlation matrix for the `time` dimension. See Example 8.

Example 8. One-dimensional error correlations

```
dimensions:
  time = 30 ;
  lon = 360 ;
  lat = 180 ;
  lat1 = 180 ;

variables:
  // Data variable
  float sst(time, lat, lon) ;
    sst:standard_name = "sea_surface_temperature" ;
    sst:units = "K" ;
    sst:units_metadata = "temperature: on_scale" ;
    sst:uncertainty_variables = "unc" ;
  // Uncertainty variable
  float unc(lat, lon) ;
    unc:standard_name = "sea_surface_temperature" ;
    unc:units = "K" ;
    unc:units_metadata = "temperature: on_scale" ;
    unc:coverage_interval = "standard_deviation" ;
    unc:coverage_probability = 0.6827 ;
    unc:probability_distribution = "gaussian" ;
    unc:error_correlation = "lat: correlations
                            time: independent
                            lon: triangular (localization_radius: 15)" ;
  // Error-correlation variable
  float correlations(lat, lat1) ;
```

Error correlations described by error-correlation matrices that demonstrate forms 1a, 2a, and 2b of the `error_correlation` attribute. For the `lat` dimension there is an *error-correlation variable* called

`correlations` giving the actual error-correlation coefficients (form 1a). For the `time` dimension, that is not spanned by the *uncertainty variable*, the error-correlation coefficients are parameterised as `independent` (form 2a). For the `lon` dimension the error-correlation coefficients are parameterised as the `triangular` error-correlation structural type, with the configuration parameter `localization_radius` given as an integer number of grid cells within the `error_correlation` attribute (form 2b).

Form 1b is similar to form 1a, but now the *error-correlation variable* provides error correlations for every pair of cells indexed by two or more dimensions simultaneously. The error-correlation matrix is square and indexes the given dimensions after they have been flattened in row-major order, i.e. the first of the dimensions given in the `error_correlation` attribute varies the slowest, and the last varies the fastest. The number of rows or columns in the matrix is therefore equal to the product of the sizes of the given dimensions. For instance, if the dimension `lat` has size 20 and dimension `lon` has size 30, then `lat: lon: varname` means that the *error-correlation variable* `varname` contains a 600 by 600 error-correlation matrix for the `lat` and `lon` dimensions that is composed of 400 non-overlapping square blocks, each of size 30 by 30. Each block along the main diagonal represents the error-correlations between pairs of cells on the `lon` dimension at a specific `lat` dimension location, and the off-diagonal blocks represent the correlations between pairs of cells on the `lon` dimension that have different `lat` dimension locations. See Example 9.

To save space in the dataset, form 1b should only be used when the error-correlation structure is intrinsically coupled across the dimensions, and cannot be calculated using the Kronecker product (KRONECKER) of error-correlation matrices that each apply to a single dimension, which are defined by form 1a.

Example 9. Multi-dimensional error correlations

```
dimensions:
  time = 30 ;
  lon = 360 ;
  lat = 180 ;
  lat_lon = 64800 ;
  lat_lon1 = 64800 ;

variables:
  // Data variable
  float sst(time, lat, lon) ;
    sst:standard_name = "sea_surface_temperature" ;
    sst:units = "K" ;
    sst:units_metadata = "temperature: on_scale" ;
    sst:uncertainty_variables = "unc" ;
  // Uncertainty variable
  float unc(time, lat, lon) ;
    unc:standard_name = "sea_surface_temperature" ;
    unc:units = "K" ;
    unc:units_metadata = "temperature: on_scale" ;
    unc:coverage_interval = "standard_deviation" ;
    unc:coverage_probability = 0.6827 ;
    unc:probability_distribution = "gaussian" ;
    unc:error_correlation = "lat: lon: correlations" ;
  // Error-correlation variable
  float correlations(lat_lon, lat_lon1) ;
```

Error correlations described by an error-correlation matrix that demonstrates form 1b of the `error_correlation` attribute. For the `lat` and `lon` dimensions combined there is a single *error-correlation variable* called `correlations`. This variable's data is a 64800 by 64800 matrix, where 64800 = 180 x 360, i.e. the product of the sizes of the `lat` and `lon` dimensions. Note that error-correlation information does not need to be provided for every *data variable* dimension, as is the case here.

Form 2: Parameterised Error-Correlation Structures

Form 2a is a parameterised alternative to forms 1a and 1b, for which *error_correlation_structure* names an error-correlation structural type which implies the error-correlation coefficient matrix for every pair of cells indexed by the given dimension or dimensions. The structural type must be one of those defined in Table A.3. For instance, when the errors are independent (i.e. uncorrelated) the error-correlation matrix is always the identity matrix, which can be parameterised by the structural type `independent`. Each structural type must be associated with a number of dimensions that is in the range allowed by the structural type definition in Table A.3, and the parameterisation applies to all of those dimensions simultaneously, rather than to each named dimension independently. If the data writer expects *error_correlation_structure* to be interpreted as one of the structural type strings defined in Table A.3, then none of the variables in the dataset should be given a name identical to that of the string, because the encoding of an *error-correlation variable* takes precedence. See Example 8.

Form 2b is similar to form 2a, but with the addition of standardised parameters for configuring the structural type. Parameters for configuring the structural type are given by the *parameter: value* pairs that identify a configuration parameter and provide its value. Each parameter must be one of those defined in Table A.4, which also specifies the structural types that the parameter may be paired with. In general, *value* names a scalar *error-correlation parameter variable* that contains the parameter value in its data array, and may have attributes to describe it (such as `units` or a `standard_name`). However, in the special case that the parameter value is a dimensionless integer, then the *value* may alternatively be set within the `error_correlation` string as the decimal representation of that integer. If the data writer expects *value* to be interpreted as an integer, then none of the variables in the dataset should be given a name identical to that of the string of digits, because the encoding of an *error-correlation parameter variable* takes precedence. See Example 8 and Example 10.

Example 10. One-dimensional error correlations with an error-correlation parameter variable

```
dimensions:
  time = 30 ;
  lon = 360 ;
  lat = 180 ;
  lat1 = 180 ;

variables:
  // Data variable
  float sst(time, lat, lon) ;
    sst:standard_name = "sea_surface_temperature" ;
    sst:units = "K" ;
    sst:units_metadata = "temperature: on_scale" ;
    sst:uncertainty_variables = "unc" ;
  // Uncertainty variable
  float unc(lat, lon) ;
    unc:standard_name = "sea_surface_temperature" ;
    unc:units = "K" ;
    unc:units_metadata = "temperature: on_scale" ;
    unc:coverage_interval = "standard_deviation" ;
    unc:coverage_probability = 0.6827 ;
    unc:probability_distribution = "gaussian" ;
    unc:error_correlation = "lat: correlations
                            time: independent
                            lon: triangular (localization_radius: r)" ;
  // Error-correlation variable
  float correlations(lat, lat1) ;
  // Error-correlation parameter variable
  float r ;
    r:units = "m" ;
```

As for Example 8, but now for the `lon` dimension, the form 2b configuration parameter `localization_radius` is instead provided by the *error-correlation parameter variable* `r`, which gives the parameter value in units of metres.

Form 3: Error-Correlation Descriptions

In **form 3a**, the actual error-correlation coefficients for the given dimensions are unknown, but nonetheless their structure may be described by non-standardised text given by the *comment*. For instance, `time: (time scale = 1 day)` could be used to indicate the temporal scale over which the errors are correlated along the `time` dimension. The non-standardised information may optionally be preceded by the string `comment:` (e.g. `time: (comment: time scale = 1 day)`), but it is recommended to not do this. When more than one dimension is given, the comment applies to those dimensions simultaneously, rather than to each named dimension independently. For instance, `lat: lon: (length scale = 100 km)` may imply a different error correlation structure to `lat: (length scale = 100 km) lon: (length scale = 100 km)`. See Example 11.

Example 11. Error correlations described by comments

```
dimensions:
  time = 30 ;
  lon = 360 ;
  lat = 180 ;

variables:
  // Data variable
  float sst(time, lat, lon) ;
    sst:standard_name = "sea_surface_temperature" ;
    sst:units = "K" ;
    sst:units_metadata = "temperature: on_scale" ;
    sst:uncertainty_variables = "unc" ;
  // Uncertainty variable
  float unc(time, lat, lon) ;
    unc:standard_name = "sea_surface_temperature" ;
    unc:units = "K" ;
    unc:units_metadata = "temperature: on_scale" ;
    unc:coverage_interval = "standard_deviation" ;
    unc:coverage_probability = 0.6827 ;
    unc:probability_distribution = "gaussian" ;
    unc:error_correlation = "time: (time scale = 1 day)
                            lat: lon: (length scale = 100 km)" ;
```

Error correlations described by non-standardised text that demonstrates form 3a of the `error_correlation` attribute.

Comments on Error Correlations

It is also possible to provide an optional non-standardised comment that gives extra information about the explicit or parameterised error-correlations of forms 1 and 2. In form 1, the comment is placed in round brackets immediately after the *error-correlation variable* name (e.g. `lat: varname (length scale = 100 km)`). The comment text may optionally be preceded by the string `comment:`, but it is recommended to not do this. In form 2a, the comment text is placed in round brackets immediately after the error-correlation structural type name, and must be preceded by the string `comment:` in order to distinguish the comment from any possible standardised parameters (e.g. `lat: independent (comment: Grid spacing exceeds the error correlation length)`). In form 2b, the comment text is placed in the round brackets immediately after the error-correlation structural type name, must be preceded by the string `comment:`, and must be placed after all of the standardised parameter definitions (e.g. `lat: triangular (localization_radius: 50 comment: length scale = 100 km)`).

Appendix A: Controlled Vocabularies

This appendix defines the controlled vocabularies required by some components of the following *uncertainty variable* attributes:

- `probability_distribution` (see Probability Distribution)

- `error_correlation` (see Error Correlation)

Probability Distributions

Table A.1. Distributions allowed by the *distribution* component of the `probability_distribution` attribute

Distribution	Description
`gaussian`	Normal distribution.
`uniform`	A continuous probability distribution in which all values within a specified interval occur with equal likelihood, producing a flat probability density across the range.
`skewed_gaussian`	A continuous probability distribution that generalises the normal distribution to allow for non-zero skewness.
`log_normal`	A continuous probability distribution of a random variable whose logarithm is normally distributed.
`chi_squared`	The distribution of a sum of the squares of *k* independent standard normal random variables.

Table A.2. Distribution parameters allowed by the *parameter* component of the `probability_distribution` attribute

Distribution parameter	Description
`skew`	A measure of the asymmetry of the probability distribution, given by the third standardised moment.
`kurtosis`	A measure of the degree of “tailedness” in the probability distribution, given by the fourth standardised moment.
`degrees_of_freedom`	The number of values in the calculation of a statistic that are free to vary. For instance, a `chi_squared` distribution with *k* degrees of freedom is the distribution of a sum of the squares of *k* independent standard normal random variables.

Error-Correlation Structures

Table A.3. Error-correlation structural types allowed by the *error_correlation_structure* component of the `error_correlation` attribute

Structural type	Description	Number of dimensions (N)	Implied error-correlation matrix structure	Optional configuration parameters (see Table A.4)
`independent`	No error-correlation.	$N \geq 1$	The identity matrix. The error-correlation structure arising from two or more dimensions simultaneously is the same as that resulting from the structural type applied to those dimensions individually.	None

`common`	Perfect error-correlation.	N ≥ 1	A full matrix of ones. The error-correlation structure arising from two or more dimensions simultaneously is the same as that resulting from the structural type applied to those dimensions individually.	None
`triangular`	Error-correlation drops linearly along each dimension.	N ≥ 1	A banded diagonal matrix, with values dropping off linearly from the diagonal. The error-correlation structure arising from two or more dimensions simultaneously may be different to that resulting from the structural type applied to those dimensions individually.	`localization_radius`
`exponential`	Error-correlation drops exponentially along each dimension.	N ≥ 1	A banded diagonal matrix, with values dropping off exponentially from the diagonal. The error-correlation structure arising from two or more dimensions simultaneously may be different to that resulting from the structural type applied to those dimensions individually.	`localization_radius`, `e_folding_length`

Table A.4. Error-correlation structure parameters allowed by the *parameter* component of the `error_correlation` attribute

Structural type configuration parameter	Description	Applicable structural types (see Table A.3)
`localization_radius`	The distance along a dimension at which error-correlation reduces to zero. An *error-correlation parameter variable* may have any units that are physically meaningful for the dimension (e.g. units of `metres` or `degrees_north` could be used for a “latitude” dimension; `degrees` could be used when both “latitude” and “longitude” dimensions are indicated). A dimensionless integer is assumed to be a number of cells.	`triangular`, `exponential`
`e_folding_length`	The distance along a dimension at which error-correlation reduces by a factor of *e*. An *error-correlation parameter variable* may have any units that are physically meaningful for the dimension (e.g. units of `metres` or `degrees_north` could be used for a “latitude” dimension; `degrees` could be used when both “latitude” and “longitude” dimensions are indicated). A dimensionless integer is assumed to be a number of cells.	`exponential`

Coverage Probabilities

For convenience, Table A.5 and Table A.6 show a selection of commonly used coverage probabilities that may be used when the coverage interval is defined as a multiple (i.e. the coverage factor) of one standard deviation either side of the mean of a Gaussian or uniform probability distribution (see Coverage Factor). These values are only relevant when the `coverage_interval` attribute is `standard_deviation` and the `probability_distribution` attribute is `gaussian` or `uniform`; the coverage factor is defined by the `coverage_factor` attribute.

Table A.5. Selected coverage probabilities for the standard and expanded uncertainties of the Gaussian distribution

Coverage factor	Coverage probability
1	0.6827
1.645	0.9
1.96	0.95
2	0.9545
2.576	0.99
3	0.9973

Table A.6. Selected coverage probabilities for the standard and expanded uncertainties of the uniform distribution

Coverage factor	Coverage probability
1	0.5774
1.559	0.9
1.645	0.95
1.715	0.99
1.732	1.0

Appendix B: Changes from v0.1

Draft v0.2 is a major revision of v0.1, made to align the specification with the CF conventions (see Relationship to the CF Conventions). Datasets written following v0.1 are not compatible with draft v0.2. The main changes are summarised in the table below.

Summary of changes from v0.1

v0.1	Draft v0.2	Notes
Observation variable	*Data variable*	Terminology aligned with CF.
`unc_comps` attribute (list of names)	`uncertainty_variables` attribute (blank-separated string of names)	
Uncertainty variables must have the same dimensions as their *observation variable*	*Uncertainty variables* may have all or a subset of the dimensions of their *data variable*, plus an optional interval dimension	See Coverage Intervals.
Meaning of the *uncertainty variable* values not defined	`coverage_interval` and `coverage_probability` attributes (both required), and `coverage_factor` attribute	Supports standard uncertainties, expanded uncertainties, and symmetric or asymmetric coverage intervals.

Uncertainty variable with no `units` is a fractional uncertainty	`uncertainty_representation = "relative_difference"`, with units of `"1"` or `"%"`	See Relative Differences.
`pdf_shape` attribute (default `"gaussian"`)	`probability_distribution` attribute (no default, but required when `coverage_interval` is `standard_deviation`)	Distributions may now be parameterised with *distribution parameter variables*.
`pdf_shape = "rectangular"`	`probability_distribution = "uniform"`	See Table A.1.
-	`uncertainty_evaluation` and `uncertainty_component` attributes	Including the identification of a combined uncertainty.
`err_corr_dimi_name`, `err_corr_dimi_form`, `err_corr_dimi_params` and `err_corr_dimi_units` attributes	Single `error_correlation` string attribute	See Error Correlation.
Error-correlation form `random`	Error-correlation structural type `independent`	See Table A.3.
Error-correlation form `systematic`	Error-correlation structural type `common`	See Table A.3.
Error-correlation form `err_corr_matrix`, with the matrix variable name given in `err_corr_dimi_params`	Form 1a or 1b of the `error_correlation` attribute, naming an *error-correlation variable*	The *error-correlation variable* must not use the same dimension twice.
Parameter units given in `err_corr_dimi_units`	Parameter values given in *error-correlation parameter variables*, with their own `units` attribute	Dimensionless integer parameters may be given directly in the `error_correlation` attribute.
-	Error-correlation structural types `triangular` and `exponential`, and descriptions of error-correlation by non-standardised text (form 3a)	See Table A.3.

The example below shows the final example dataset from v0.1 updated to follow draft v0.2. In v0.1, the uncertainty values were implicitly standard uncertainties, and `u_noise` was a fractional uncertainty because it had no `units` attribute.

Example 12. v0.1 example updated to follow draft v0.2

v0.1:

```
variables:
  float temperature(time, lat, lon);
    temperature:unc_comps=["u_calibration", "u_noise"];
    temperature:units="K"
  float u_calibration(time, lat, lon);
    u_calibration:units="K";
    u_calibration:pdf_shape="rectangular";
    u_calibration:err_corr_dim1_name=["lat", "lon"];
    u_calibration:err_corr_dim1_form="systematic";
    u_calibration:err_corr_dim2_name="time";
    u_calibration:err_corr_dim2_form="err_corr_matrix";
    u_calibration:err_corr_dim2_params=["err_corr_calibration_time"];
  float u_noise(time, lat, lon);
```

```
    u_noise:err_corr_dim1_name=["time", "lat", "lon"];
    u_noise:err_corr_dim1_form="random";
  float err_corr_calibration_time(time, time);
```

Draft v0.2:

```
variables:
  // Data variable
  float temperature(time, lat, lon) ;
    temperature:units = "K" ;
    temperature:uncertainty_variables = "u_calibration u_noise" ;
  // Uncertainty variables
  float u_calibration(time, lat, lon) ;
    u_calibration:units = "K" ;
    u_calibration:coverage_interval = "standard_deviation" ;
    u_calibration:coverage_probability = 0.5774 ;
    u_calibration:probability_distribution = "uniform" ;
    u_calibration:error_correlation = "lat: lon: common
                                      time: err_corr_calibration_time" ;
  float u_noise(time, lat, lon) ;
    u_noise:units = "1" ;
    u_noise:uncertainty_representation = "relative_difference" ;
    u_noise:coverage_interval = "standard_deviation" ;
    u_noise:coverage_probability = 0.6827 ;
    u_noise:probability_distribution = "gaussian" ;
    u_noise:error_correlation = "time: lat: lon: independent" ;
  // Error-correlation variable
  float err_corr_calibration_time(time, time1) ;
    err_corr_calibration_time:units = "1" ;
```

The `u_calibration` standard uncertainty has a uniform distribution, so its coverage probability is 0.5774 (see Table A.6). Its errors are fully correlated in the `lat` and `lon` dimensions (form 2a), and correlated in the `time` dimension as defined by the *error-correlation variable* `err_corr_calibration_time` (form 1a), which now uses a separate dimension, `time1`, of the same size as `time`. The `u_noise` standard uncertainty is given as a fractional relative difference, with uncorrelated errors in all dimensions.