HadIOD.1.2.0.0 User Guide
v1.2

Chris Atkinson (chris.atkinson@metoffice.gov.uk)

Version	Description	Created
1.0	First version of document	27/07/20
1.1	Updated to include use of EN.4.2.2 from March 2021 onwards. Updated figures showing basic data statistics (now generated using latest plotting code).	02/11/21
1.2	Updated to include use of ICOADS.3.0.2 from April 2022 onwards.	26/09/22

1   Quick Start Guide

HadIOD is a database created by the Met Office Hadley Centre of in situ surface and subsurface ocean temperature and salinity observations supplemented with metadata including bias corrections, measurement uncertainties and quality flags. It is essential that users of ocean data correct for known biases (systematic errors) and explore the sensitivity of their applications to observational uncertainties and HadIOD seeks to facilitate this. Target users of HadIOD include ocean modellers and anyone wanting to simply 'pick up and use' ocean data appropriately (it is perhaps less suited to observation specialists looking for comprehensive metadata e.g. detailed metadata about observation provenance or other coincident measured oceanographic and/or meteorological variables). The database coverage is global from 1850-present though coverage is often sparse; in general the oceans are better sampled in space (horizontally) and depth as time progresses (see Section 2.8 for some basic plots). Monthly updates are made on a best endeavours basis with approximately a few months delay.

HadIOD is versioned and this document describes HadIOD.1.2.0.0.

What data are available?

Data are extracted from the HadIOD.1.2.0.0 database into HadIOD public release files which are described in this document. Files are in daily CF-compliant NetCDF4 format. The bulk of the data are provided in a single 'main' data file for each day. Separate files for each day containing additional sets of bias corrections for ships and XBTs/MBTs are also available.

HadIOD brings together observations and metadata from a variety of sources and as such relies on the work of many other people. The sources of observations in the HadIOD.1.2.0.0 public files are: ICOADS.2.5.1 [Woodruff et al., 2011], EN4 [Good et al., 2013] and some CMEMS (http://marine.copernicus.eu/) drifting buoy data.   ICOADS.3.0.2 [Freeman et al., 2017; https://icoads.noaa.gov/] was substituted for ICOADS.2.5.1 from April 2022 onwards. Further details of the sources of data in HadIOD are given in FAQs Section 2.7.5 and Section 2.7.7 and in Atkinson et al. [2014]. Only a subset of the data and fields in the HadIOD database have been extracted into the HadIOD public files. We have provided the information we expect the average user to need.

Generated using E.U. Copernicus Marine Service Information.

How do I obtain the data?

HadIOD.1.2.0.0 public files can be downloaded from:

- https://www.metoffice.gov.uk/hadobs/hadiod/download-hadiod1-2-0-0.html

PLEASE READ FAQs Section 2.7.17 for dataset terms and conditions.

If you use HadIOD data we would be grateful if you could let us know for our records.

Dataset announcements (e.g. any issues reported by users) will be made via the HadIOD webpages:

- https://www.metoffice.gov.uk/hadobs/hadiod/

How do I read the data?

NetCDF is a platform-independent, self-describing binary format and there are a number of common tools (Section 3.4) that can be used to access the data. Some basic Python code is provided in Section 4 showing an example of reading the data and performing some simple calculations and processing.

Files are CF1.7 compliant, this means some variable attributes and the way the data are stored follow conventions that enable the exchange of climate data, see http://cfconventions.org.

How to cite the data

When using the data, please quote the version number (for example HadIOD.1.2.0.0), the date the data were downloaded and cite the following:

> *Atkinson, C. P., N. A. Rayner, J. J. Kennedy, and S. A. Good (2014), An Integrated Database of Ocean Temperature and Salinity Observations, J. Geophys. Res. Oceans, 119, 7139-7163, https://doi.org/10.1002/2014JC010053.*

PLEASE READ FAQs Section 2.7.16 for further information about citing the data.

Further information and contact

For further help please read the rest of the document. In particular see Section 2 for information about using the HadIOD data including working with the observation bias corrections and uncertainties and Section 3 for details of the file format and contents. Atkinson et al. [2014] (free access) is recommended for technical details. For further enquiries please contact us.

Contents

2 Getting started with HadIOD.1.2.0.0

2.1 Introduction

HadIOD (Met Office Hadley Centre Integrated Ocean Database) is a database of global historical in situ ocean temperature and salinity observations. It brings together observations made by surface-only observing platforms (like ships and buoys) and observations from sub-surface ocean profiling platforms (like profiling floats and bathythermographs) and supplements these with metadata including quality flags, bias corrections and estimates of measurement uncertainty.

The term 'Integrated' in 'Integrated Ocean Database' refers to the merging of data from multiple sources concerned with different components of the observing system (e.g. the surface or subsurface ocean) into a single data store, and the inclusion of appropriate metadata (such as bias corrections and duplicate flags) to help enable greater internal consistency.

A primary target use for HadIOD is ocean modelling, in particular ocean analyses that assimilate both surface and sub-surface data. HadIOD should also be helpful to users wanting to simply 'pick up and use' ocean data appropriately, with the focus on providing metadata such as bias corrections, measurement uncertainties and overall quality flags, rather than detailed information more suited perhaps to observation specialists e.g. metadata about data provenance or other coincident measured oceanographic and/or meteorological variables. See also Section 2.1.1.

At its core HadIOD is a relational database hosted at the Met Office that is not visible to users. Data can be extracted from this database in various ways and in various formats and provided to users. A summary of the various datasets extracted from HadIOD is given in FAQs Section 2.7.15. This user guide describes the public NetCDF files created from version 1.2.0.0 of the HadIOD relational database and served via https://www.metoffice.gov.uk/hadobs/hadiod. The public files described here are considered the primary source of HadIOD.1.2.0.0 data for users.

In this user guide, where it is necessary for clarity to distinguish the HadIOD.1.2.0.0 database from the HadIOD.1.2.0.0 public files we will do so explicitly, though in general just 'HadIOD' (the thing) is sufficient. Unless specified otherwise we are referring to version 1.2.0.0 of HadIOD.

The HadIOD.1.2.0.0 public files are available as daily files from 1850 to the present in CF-compliant NetCDF4 format. There is a single file for each day that contains the bulk of the data which we refer to as the 'main' files. Separate daily files containing additional sets of bias corrections for ships and XBTs/MBTs are also available and can be used to explore the bias correction uncertainty. The format and contents of the files is described in Section 3.

Note that only a subset of the data and fields in the HadIOD database have been extracted into the HadIOD public files. We have provided the information we expect the average user to need. If you do not see what you need then please contact us and we will try to help.

It is emphasised that HadIOD brings together observations and metadata from a variety of sources and as such relies on the work of many other people. The sources of observations in the HadIOD.1.2.0.0 public files are: ICOADS.2.5.1 [Woodruff et al., 2011], EN4 [Good et al., 2013] and some CMEMS (http://marine.copernicus.eu/) drifting buoy data. ICOADS.3.0.2 [Freeman et al., 2017; https://icoads.noaa.gov/] was substituted for ICOADS.2.5.1 from April 2022 onwards. Further details of the sources of data in HadIOD are given in FAQs Section 2.7.5 and Section 2.7.7.

The published reference for HadIOD.1.2.0.0 is Atkinson et al. [2014] (free access) (note that this paper describes HadIOD.1.0.0.0; the differences between HadIOD versions is described in FAQs Section

2.7.14). We would recommend reading this for technical information about HadIOD in addition to this user guide.

For help with citing the HadIOD public files see FAQs Section 2.7.16.

2.1.1   What are HadIOD's distinguishing features?

The main distinguishing features of HadIOD are as follows:

- **Merging of data from a wide variety of different observation types into a unified format**. This will benefit users wanting all available ocean data. For example, ocean analyses where observations from the sub-surface ocean and near the ocean-atmosphere interface are needed for assimilation. Data for each day are provided together in a single file with straightforward format.

- **Provision of observational bias corrections and uncertainties in line with current understanding**. It is essential that users correct for known biases in the data and allow properly for the sensitivity of their applications to observational uncertainties. By combining the observations, bias corrections and uncertainties together in a single data set, users are able to conveniently utilize our current understanding of observation error in their applications. With applications such as data assimilation in mind, we have tried to ensure that at minimum an estimate of uncertainty, a depth and an overall quality flag is provided for every observation in HadIOD.

- **Global historical coverage of data from 1850–present**. Updates with the latest data are made monthly with approximately a few months delay from the end of a month on a best endeavours basis.   The database is designed with research rather than operational applications in mind. If you would benefit from more timely updates please contact us, we would be interested to hear from potential users who may have this requirement.

- **Providing the core information required by non-observational specialists for their research** (i.e. users who want largely to just pick up and use the observations, such as ocean modellers). HadIOD is a database of observations supplemented with a variety of metadata.   HadIOD carries some, but not all, of the wide range of supplementary metadata sometimes available from its data sources. Observational specialists interested in detailed metadata may find other products more appropriate for their needs[1]; see Table 1 for a list of some other datasets. HadIOD uses observation IDs to retain traceability to its source data, please see FAQs Section 2.7.13 for more information about this.

2.2   How do I get the data?

Data are available from:

- https://www.metoffice.gov.uk/hadobs/hadiod/download-hadiod1-2-0-0.html

---

[1] *Note however that only a subset of the data stored in the HadIOD database has been extracted into the HadIOD public files described here. If you don't see what you want in the files then please* contact us *as it might be possible to extract supplementary data. New data could also potentially be added into the database.*

Daily data files can be downloaded from 1850-present in yearly batches or in their entirety. The latest daily data are made available in monthly batches with approximately a few months delay from the end of the month. Updates are made on a best endeavours basis, sometimes there may be a longer delay.

PLEASE READ FAQs Section 2.7.17 for dataset terms and conditions.

If you use HadIOD data we would be grateful if you could let us know for our records.

Dataset announcements (e.g. any issues reported by users) will be made via the HadIOD webpages:

- https://www.metoffice.gov.uk/hadobs/hadiod/

2.3   How do I use the data?

The data are in NetCDF4 format, which is a standard data format for climate data. Furthermore, the data files are CF compliant (http://cfconventions.org) at 1.7 standard (with one exception, see Section 3.2.1) which means that the metadata in the files are in a standardised format. The format of the data files is described in more detail in Section 3, some tools for working with NetCDF files are listed in Section 3.4. Some example code for reading and processing the data is found in Section 4.

2.3.1   Do's and Don'ts

Do – engage with the errors in the data. It is essential that users of ocean data correct for known biases (systematic errors) and explore the sensitivity of their applications to observational uncertainties.

Do – apply the bias corrections provided with the data. We don't apply these automatically so that the user can investigate their impact if needed. Check the global attribute *usage* in the files to ensure you apply the corrections appropriately (see also Section 2.5).

Don't – ignore the observation uncertainties. These help to understand the relative reliability of the data which can differ widely dependent on observation type, time and location. For some observation types multiple sets of bias corrections are provided to help with exploring the uncertainty.

Do – send us feedback when you use the data. We are interested to know how we could improve what we provide.

2.4   What are all the bias correction and uncertainty variables for?

Observed values are not perfect. They can suffer from both persistent systematic errors which are correlated amongst observations (we refer to these as biases) and random errors which are uncorrelated/independent from one observation to the next. These errors obscure the true value we seek to observe. Our estimate of the reliability of an observed value due to some source of error is quantified by an uncertainty estimate which describes the spread of a probability density function from which the (unknown) error is drawn (often a standard deviation). Where errors resulting from systematic effects are understood we facilitate correction of these by providing bias corrections. These corrections are themselves not perfectly known and our estimate of the reliability of a bias correction is described by a bias correction uncertainty. A bias correction uncertainty describes the unknown residual bias in an observation after the correction has been applied. There are generally several 'sources of uncertainty' that jointly contribute to the uncertainty associated with an observed value. An overall probability distribution describing the final (unknown) error of an observed (possibly corrected) value is built up from the probability distributions associated with all the different sources

of uncertainty. The *Guide to the Expression of Uncertainty in Measurement* ('the GUM', Joint Committee for Guides in Metrology [2008]) provides authoritative guidance on how to determine, combine and express uncertainty.

In HadIOD we try to incorporate the latest knowledge of observational errors so users can benefit through correcting for known biases and exploring the impact of observation uncertainty on their application. Help on how to do this is given in Sections 2.5 and 2.6. The error model currently used for observations in HadIOD is based on that used for HadSST.3.1.1.0 [Kennedy et al., 2011a, 2011b] (hereon referred to as HadSST3), which is a Met Office Hadley Centre dataset of gridded in situ sea surface temperature (SST) anomalies that incorporates a detailed treatment of observation biases and uncertainties. In HadIOD, each observed value (Ob) is considered to consist of the true value (True) plus a bias associated with the type of platform or instrument (BT, e.g. observations made by a bucket deployed from a ship or a particular expendable-bathythermograph probe type), a bias specific to an individual platform or instrument (BP, e.g. a particular ship or probe), and a random measurement error (M):

$$Ob = True + BT + BP + M \qquad (2.1)$$

In the HadIOD public files (see also Section 3.2), where GV=Geophysical Variable and GV is one of '*temp*' (temperature) or '*sal*' (salinity):

- Bias corrections for BT are stored in the variable '*[GV]_type_corr*'. The uncertainties for these bias corrections are in the variable '*[GV]_type_corr_unc*'. Systematic errors (residual biases) associated with the uncertainties from '*[GV]_type_corr_unc*' can be correlated amongst observations from the same platform type or instrument type.
- Bias corrections for BP are in variables named '*[GV]_plat_corr*'. The uncertainties for these bias corrections are in the variable '*[GV]_plat_corr_unc*'. Systematic errors (residual biases) associated with the uncertainties from '*[GV]_plat_corr_unc*' can be correlated amongst observations from the same individual platform or instrument.
- The uncertainty due to M is given in variables named '*[GV]_rand_unc*'. Random errors associated with this uncertainty are not correlated with errors in other observations. The effect of random measurement error differs for each observed value and cannot be corrected for.

The variables named '*[GV]_comb_unc*' give the uncertainty obtained by combining all other uncertainty terms in quadrature (i.e. treating them as uncorrelated with each other). This gives the final uncertainty in each observed value once any corrections have been applied. Uncertainties are provided as standard deviations (standard uncertainties). When doing calculations which combine multiple observed values the different uncertainty terms must be handled appropriately. Section 2.6 provides help on how to use this uncertainty information.

The nature of the errors and the extent to which they are known can be quite variable for different platform and instrument types. Consequently, the extent to which this information can be provided in HadIOD varies amongst observations. The bias correction and uncertainty variables contain many missing values as a placeholder for possible future knowledge. A summary of the information provided in the main HadIOD.1.2.0.0 files is given in Table 2 for temperature and Table 3 for salinity. At minimum we try to provide some measure of uncertainty for every observation but in a minority of cases this may not be possible (see FAQs Section 2.7.12).

As our knowledge of observation errors gradually improves, this will be incorporated into future versions of HadIOD.

2.5   How do I use the bias corrections?

The bias corrections in the main files should be applied as follows:

- corrected temperature = *temp* + *temp_type_corr* + *temp_plat_corr*
- corrected salinity = *sal* + *sal_type_corr* + *sal_plat_corr*
- corrected depth = *depth* + *depth_corr*

To help users explore uncertainties the additional bias correction files (see Section 2.6) provide alternative sets of corrections for *temp_type_corr* and *depth_corr* for some observation types and these should be applied in the same way.

For some observations the value of a correction may be zero (where we consider an observation effectively unbiased) or missing (where the bias is not known). For the purpose of applying the corrections, missing values should be treated as zero.

The bias corrections (and uncertainties) provided could differ in sets of files created from another version of the HadIOD database (e.g. where knowledge of biases may have changed) or in other sets of files created from the same version of the HadIOD database (e.g. if a different set of corrections is used). When writing code it is best to apply any corrections anew each time a new set of files is used, do not assume that correction values have been allocated as for other sets of files (e.g. don't presume that corrections will or won't be provided for certain observations based on using other files). By making fewer assumptions about the data it will be easier to switch in and out different sets of HadIOD data without having to modify code

In HadIOD.1.2.0.0 we provide corrections for some types of temperature observation. We do not currently provide any salinity corrections. Table 2 and Table 3 summarise the corrections and uncertainties provided in the main HadIOD.1.2.0.0 files.

Depth corrections are provided for observations where there may be an error in the reported depth or where the temperature corrections are made relative to a reference depth. Depth corrections are only available for a few platform types (and not for individual platforms) so currently only one depth correction variable (without uncertainty) is provided to save file space.

The NetCDF data files include the global attributes *source* and *usage*. For some observation types in HadIOD there are multiple versions of the bias corrections available which are used to explore the uncertainty in the bias corrections (see Section 2.6). The *source* variable is used to describe the version of HadIOD that is used to provide the data (e.g. HadIOD.1.2.0.0) as well as which bias corrections are provided in the file (for example which bias correction scheme has been selected for expendable-bathythermograph observations). The *usage* variable gives any rules that must be followed when using the data in a file and this may include rules for applying the bias corrections (for example, if a temperature correction is made relative to some reference depth then a depth correction will be provided that adjusts observed depth to the reference depth; the rule would state that this depth correction *must* be applied in tandem with the temperature correction).

When using a set of files (where set refers to files of the same sort from the same version of HadIOD, e.g. a set of main files from HadIOD.1.2.0.0 or a set of additional ship correction files from HadIOD.1.2.0.0) a user would not normally need to check the *usage* for every file. The *usage* rules for each file in a set will be the same provided the *source* information is the same. When a user downloads a set of data from its download webpage (Section 2.2) the *source* will be the same in all files downloaded. For example, all main HadIOD.1.2.0.0 files have the following characteristics:

- *source*: HadIOD version 1.2.0.0:

    Gouretski and Reseghetti [2010] MBT/XBT correction scheme and ship Mean Ensemble Member correction scheme.

- *usage*: file-specific guidance:

    temp_type_corr and *depth_corr* must be applied together (and not independently).

    For help working with correction and uncertainty variables read the product user guide (see global attribute: references).

Do not presume the *usage* is the same in different sets of files, users must ensure they follow the *usage* for each set as appropriate.

2.5.1   Potential temperature corrections

Potential temperature is available in EN4 and is included in the HadIOD main files (EN4 is the main source of sub-surface profile information in HadIOD.1.2.0.0). Potential temperature is not available in HadIOD for surface-only (non-profile) observations but will be very close to the in situ temperature. The potential temperature has been calculated for uncorrected observations (i.e. without any temperature or depth corrections applied). The corrections could be applied to the potential temperatures as follows:

- corrected potential temperature = *potemp* + *temp_type_corr* + *temp_plat_corr*
- corrected depth (for use with potential temperature) = *depth* + *depth_corr*

However, users should be aware that calculating potential temperature and then applying corrections (as above) will not give the same result as calculating potential temperatures from the corrected in situ observations (i.e. applying corrections first). An error will be introduced in the former.

The magnitude of this error will vary dependent on the water mass and the values of the corrections. In HadIOD.1.2.0.0 the observation types affected are XBTs and MBTs (as these are the profile observation types for which corrections are provided). For these observation types we might expect potential temperature errors to be small (order 0.01°C or less) relative to measurement error though somewhat systematic (like the corrections). Users should consider recalculating potential temperatures for affected observations if required.

2.6   How do I use the uncertainty estimates?

There are three uncertainty components in HadIOD and the errors associated with each of these have different degrees of correlation between individual observations (see Section 2.4 for the HadIOD error model). The components are:

1. **Random measurement error component**. This represents uncertainties arising from uncorrelated measurement errors. At minimum we try to provide an estimate of the uncertainty due to random measurement error for every observation (both temperature and salinity and for all platform and instrument types).

2. **Individual platform bias component**. This represents uncertainties arising from the imperfect knowledge of biases in individual platforms such as individual ships. In HadIOD.1.2.0.0 we treat the errors (residual biases) associated with these uncertainties as completely correlated for all temperature or salinity observations from the same platform, where observations from the same platform are identified by having the same platform ID (stored in the variable *plat_id*) and platform type (stored in the variable *plat_type*). This uncertainty component is only currently provided for surface temperature observations made by ships, drifting buoys

and moored buoys (Table 2). For these observations this uncertainty is associated with a bias correction of zero, i.e. we don't correct for bias in individual ships and buoys but we can describe the range of the probable value of the (uncorrected) bias as an uncertainty.

3. **Platform-type or instrument-type bias component**. This represents uncertainties arising from the imperfect knowledge of biases in platform types such as ships or bathythermographs. The correlation of the errors (residual biases) associated with these uncertainties can be complex, difficult to describe and difficult to propagate analytically. HadIOD handles this by providing multiple sets of bias corrections that explore the uncertainty space. By applying each of these in turn a user can explore the sensitivity of their application to this uncertainty. In HadIOD.1.2.0.0 we provide multiple sets of bias corrections for surface temperature observations made by ships (from 1850-present) and temperature profile observations made by mechanical bathythermographs (MBTs) and expendable bathythermographs (XBTs) (see FAQs Section 2.7.7 for more information). **The main HadIOD files include one set of XBT/MBT and ship corrections. Further sets of bias corrections for ships and XBTs/MBTs are provided in separate daily files (see Section 3).** This component of uncertainty is increasingly important at larger time and space scales, for example if concerned with decadal trends.

For some observations the value of an uncertainty component (see Table 2 and Table 3) may be zero (where we consider an observation effectively certain) or missing (where the uncertainty is not known). For the purpose of propagating the uncertainties, missing values should be treated as zero.

In a minority of cases where no observation uncertainty information is provided (and *temp_comb_unc* or *sal_comb_unc* is missing), users may wish to exclude these observations from their analysis or assign their own uncertainty values, see FAQs Section 2.7.12.

How exactly these uncertainties are handled by users will depend on the application and the limitations it imposes. The information in the subsequent sections tries to give a basic guide to working with the uncertainty components provided with the HadIOD data files. We begin in Section 2.6.1 with some mathematical background about the law of propagation of uncertainty before describing its application to the three uncertainty components in HadIOD (Sections 2.6.2, 2.6.3 and 2.6.4), culminating in a few imagined use cases in Section 2.6.5. Section 2.6.3.1 describes a Monte Carlo approach for assessing the sensitivity of an application to the uncertainty which may be appropriate where it is difficult to propagate the uncertainty analytically (a similar approach is used in Section 2.6.4 for the platform type bias component). Some important caveats are given in Section 2.6.6. Section 4 shows a worked example in Python, illustrating some of the different approaches for working with the uncertainties.

This guide cannot be all things to all people but hopefully it is a useful start. Users will need to carefully consider how to propagate the uncertainty information in their own application. Keep in mind that HadIOD only provides information about **measurement uncertainties**. Users will need to consider whether other sources of error may be present in their application (e.g. sampling errors may arise if an observation does not exactly represent the quantity of interest, e.g. if a point observation is used to represent an area average). Please do contact us if there is anything you are not sure about or if you need advice and we will try to help.

2.6.1   Uncertainty propagation

For a function of the form:

$$Y = f(X_1, X_2, X_3, X_i, \dots, X_n) \qquad (2.2)$$

the law of propagation of uncertainty can be written as:

$$u_c^2(y) = \sum_{i=1}^{n} c_i^2 u^2(x_i) + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_i c_j\, u(x_i, x_j) \qquad (2.3)$$

where an estimate x_i of quantity X_i has an associated uncertainty $u(x_i)$ and y is an estimate of quantity Y calculated using the estimated quantities x_i. The $u(x_i)$ term is the standard uncertainty (the standard deviation of the probability distribution from which the unknown error in x_i is drawn). The term $u_c^2(y)$ is the squared standard uncertainty (the variance) associated with the value y which comes from a combination of the uncertainties associated with the different estimated quantities, x_i. The c_i terms are the 'sensitivity coefficients':

$$c_i = \frac{\partial f}{\partial x_i} = \frac{\partial Y}{\partial X_i} \qquad (2.4)$$

which describe how sensitive y is to an uncertainty associated with x_i. The first term on the RHS of Equation (2.3) is the sum of the squares of the standard uncertainties $u(x_i)$ multiplied by the square of their sensitivity coefficients, known as 'adding in quadrature'. The second term on the RHS of Equation (2.3) deals with the error covariance of correlated quantities $u(x_i, x_j)$ for all combinations of pairs of quantities.

A simple example of a calculation described by Equation (2.2) would be the non-weighted average of a set of observations:

$$Y = \frac{\sum_{i=1}^{n} X_i}{n} = \left(\frac{X_1}{n} + \frac{X_2}{n} + \frac{X_i}{n} + \dots + \frac{X_n}{n} \right) \qquad (2.5)$$

for which $c_i = 1/n$.

Noting that $c_i^2 u^2(x_i) = c_i c_j u(x_i, x_i)$ then Equation (2.3) can also be written as:

$$u_c^2(y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j\, u(x_i, x_j) \qquad (2.6)$$

In matrix form this is:

$$u_c^2(y) = \mathbf{C}_y \mathbf{U}_x \mathbf{C}_y^{\mathrm{T}}, \qquad (2.7)$$

where $\mathbf{U}_x$ is the error covariance matrix:

$$\mathbf{U}_x = \begin{bmatrix} u^2(x_1) & \cdots & u(x_1, x_n) \\ \vdots & \ddots & \vdots \\ u(x_n, x_1) & \cdots & u^2(x_n) \end{bmatrix} \qquad (2.8)$$

(the diagonal terms are the squared standard uncertainties and the off-diagonals are the error covariances between pairs of quantities) and $\mathbf{C}_y$ is a row vector of sensitivity coefficients:

$$\mathbf{C}_y = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right] = [c_1, c_2, \ldots, c_n] \tag{2.9}$$

The matrix form of the propagation of uncertainty equation can be a helpful way of applying the law of propagation of uncertainty when dealing with many quantities and correlations. A possible downside is that for calculations involving a large number of observations this matrix can get quite large (for example, for n=10,000 observations, a 10,000 x 10,000 matrix of 32 bit floating point numbers = 3.2 GB of memory). If matrix size is an issue then iterating through the summations of Equation (2.6) is an alternative.

An easy way to handle the three separate uncertainty components in HadIOD is to propagate each one individually through any calculation and then to combine these at the end to get an overall estimate of calculation uncertainty:

$$u_c^2(overall) = u_c^2(component1) + u_c^2(component2) + u_c^2(component3) \tag{2.10}$$

In the following sections we describe how each of the uncertainty components in HadIOD can be propagated to get the terms on the RHS of Equation (2.10).

2.6.2   Component 1: Propagating uncertainty associated with uncorrelated errors

For uncorrelated errors the error covariance between quantities x_i is zero. The second term on the RHS of Equation (2.3) thus disappears and it becomes:

$$u_c^2(y) = \sum_{i=1}^{n} c_i^2 u^2(x_i) \tag{2.11}$$

If taking a matrix approach, then the off-diagonal terms of the error covariance matrix shown in Equation (2.8) would be zero. All that remains is to determine c_i (Equation (2.4)) and then the uncertainty associated with the random measurement error component of a calculation can be determined.

Note that in some cases it may be possible to simplify Equation (2.11) further. For example, in the case where we are calculating the non-weighted average of a set of observations and where each observation has the same random measurement error uncertainty value then:

$$u_c^2(y) = \frac{u^2(x_i)}{n} \tag{2.12}$$

This simplification could be made if, for example calculating the average of a set of observations of the same type in HadIOD.1.2.0.0 which have been allocated the same random measurement error uncertainty (e.g. ships or profiling floats, see Table 2 and Table 3). Such simplified approaches to propagating uncertainty are not recommended however as the way uncertainties are allocated could change in future HadIOD data releases. By making fewer assumptions about the data it will be easier to switch in and out different sets of HadIOD data without having to modify code.

2.6.3 Component 2: Propagating uncertainty associated with simply-correlated errors

Unlike the uncorrelated errors in Section 2.6.2, when there is a correlation between errors in estimated quantities, we need to know the error covariances to propagate uncertainty correctly (i.e. the second term on the RHS of Equation (2.3) or the off-diagonal elements in the error covariance matrix in Equation (2.8)). The error covariance between quantities x_i and x_j is:

$$u(x_i, x_j) = u(x_i) \times u(x_j) \times r(x_i, x_j) \tag{2.13}$$

where r is the correlation coefficient, describing the correlation of the error between quantities x_i and x_j. To propagate the uncertainty we thus need to know r for all pairs of quantities. In HadIOD.1.2.0.0, for the individual platform bias component, the errors are assumed completely correlated (i.e. $r = 1$) for observations from the same individual platform (thus the error covariance between quantities x_i and x_j becomes $u(x_i)$ multiplied by $u(x_j)$). For observations from different individual platforms the errors are assumed entirely uncorrelated (i.e. $r = 0$) and the error covariance between quantities x_i and x_j is zero.

All that remains is to determine c_i (Equation (2.4)) and then the uncertainty associated with the individual platform bias component of a calculation can be determined using Equation (2.3) or in matrix form using Equation (2.7).

The rule for identifying observations from the same individual platform in the data files is that these should have the same platform ID (stored in the variable *plat_id*) and the same platform type (stored in the variable *plat_type*). Because of deficiencies in the metadata, sometimes observations from the same platform do not share the same ID and sometimes observations from different platforms may share the same generic ID. This will result in the uncertainty being underestimated or overestimated respectively (as decreasing/increasing the error covariance between quantities decreases/increases the overall uncertainty). This is a known issue for historical ship data (where e.g. use of generic IDs such as 'SHIP' is widespread) and is something that the observations community are trying to improve, see also FAQs Section 2.7.9. As improvements are made, we will try to incorporate them into future versions of HadIOD.

In HadIOD.1.2.0.0 observations from the same individual platform are always allocated the same individual platform bias uncertainty value and the associated errors (as described above) are treated as completely correlated amongst observations from the same platform and uncorrelated between platforms. Given this knowledge in some cases it may be possible to simplify Equation (2.3) further. For example, in the case where we are calculating the non-weighted average of a set of observations then:

$$u_c^2(y) = \frac{1}{n^2} \sum_{i=1}^{m} n_i^2 u^2(x_i) \tag{2.14}$$

where m is the total number of individual platforms and n_i is the number of observations made by a particular individual platform. In the special case where $m=1$, the overall uncertainty is simply $u^2(x_i)$, i.e. this component of uncertainty cannot be reduced without observations from multiple individual platforms.

This simplification could be made when calculating the average of a set of observations in HadIOD.1.2.0.0 given knowledge of how the uncertainties are currently allocated. Such simplified

approaches to propagating uncertainty are not recommended however as the way these uncertainties are described could change in future HadIOD data releases (for example both the uncertainty values and the correlations of their associated errors could change). By making fewer assumptions about the data it will be easier to switch in and out different sets of HadIOD data without having to modify code.

2.6.3.1   *Uncertainty propagation by generating multiple samples of observation error*

Although uncertainty components 1 and 2 can be straightforward to propagate analytically, an alternative approach is to generate an 'ensemble' of observations. This is done by adding noise to the observations generated from the error model given in Equation (2.1) following the rules for HadIOD.1.2.0.0 (e.g. how correlation of error is treated). These samples can then be used to explore the sensitivity of an application to the uncertainty. It also works in situations where it is difficult to propagate the uncertainty analytically. The process for generating an ensemble of samples for HadIOD.1.2.0.0 is straightforward:

1. Apply the bias corrections in the main file to the observations to get a set of corrected observations (see Section 2.5).
2. For each observation take a single random draw from a normal distribution, multiply this value by *temp_rand_unc* or *sal_rand_unc* dependent on observation type (these provide $u(x_i)$ for uncertainty component 1, see Section 2.4) and add it to the observation.
3. For each individual platform (identified as per Section 2.6.3) first take a single random draw from a normal distribution, then for each observation from that platform multiply this value by *temp_plat_corr_unc* or *sal_plat_corr_unc* dependent on observation type (these provide $u(x_i)$ for uncertainty component 2, see Section 2.4) and add it to the observation.
4. Repeat Steps 1-3 to generate multiple samples.

The uncertainty in any quantity derived from the observations can be assessed by calculating the derived quantity for each of the samples. The spread of the set of derived quantities gives an estimate of the propagated uncertainties for uncertainty components 1 and 2. The number of samples needed will depend on the application.

A similar approach is used in Section 2.6.4 for the platform type bias component. This could be incorporated here to generate an ensemble of observations that includes all three components of uncertainty. One way to do this would be to create multiple samples (as above) for each set of platform type bias corrections. To do this, for each set of platform type bias corrections, apply these in place of the bias corrections used in Step 1. See Section 4.6 for an example.

2.6.4   Component 3: Propagating uncertainty using multiple bias corrections

The correlation of the errors associated with the platform type bias component uncertainties are difficult to describe mathematically and therefore difficult to propagate analytically using the equations of Section 2.6.1. Instead we provide multiple sets of bias corrections that the user can apply in turn to explore the sensitivity of their application to this component of uncertainty. From a user point of view this is straightforward to do, simply apply one set of bias corrections before calculating the quantity derived from the observations (e.g. the average of a set of observations), then apply a different set of bias corrections and repeat until all bias corrections have been tried. The spread of the set of derived quantities gives an estimate of the uncertainty due to this component.

In practice how exactly this will work will depend on the user application and the corrections used. In HadIOD.1.2.0.0 there are four sets of XBT corrections, two sets of MBT corrections, and 100 realisations of the ship corrections, see FAQs Section 2.7.7 for further information. We recognise that in some applications (such as data assimilation and ocean modelling) the cost of each run may be

expensive and this may limit a user's ability to explore multiple sets of bias corrections. The main HadIOD files include one set of ship and MBT/XBT corrections and applying these will help to remove biases in the data, even if the uncertainty cannot be explored. Additional corrections are provided in separate files (Section 3).

If a user is not including XBT, MBT or ship data in their analysis then they do not need to worry about this component.

The nature of the XBT and MBT corrections are slightly different to the ship corrections. For XBTs and MBTs each set of corrections is taken from a different published correction scheme i.e. each scheme uses a different approach to modelling and estimating the bias corrections (this is known as structural uncertainty). For ships only one correction scheme is used and the multiple sets of ship corrections explore the uncertainty in the input parameters used by the correction scheme.

The single set of ship corrections provided in the main files has been designed to lie somewhere near the centre of the range of possible ship corrections over time (we call this the ship 'Mean Ensemble Member' (MEM), see Atkinson et al. [2014] for more information). In the main files, for each ship correction based on the MEM, we also provide an uncertainty (in *temp_type_corr_unc*) which is the standard deviation of the correction values of the 100 other ship correction realisations. This provides an estimate of the uncertainty in the MEM bias correction value for an individual ship observation and may be helpful for users unable to explore multiple sets of ship corrections. Values for *temp_type_corr_unc* can also be combined with other sources of uncertainty to get an estimate of the overall reliability of each individual corrected ship observation (in *temp_comb_unc*). We emphasise that the correlation of the errors between ship observations for this component of uncertainty, and thus the full structure of the uncertainty, cannot be explored unless applying multiple corrections in turn, and particularly at larger space and time scales. An example of different ways of working with the ship uncertainties is shown in Section 4.

After performing any calculations the user will need to decide the best way to present the results. Because the XBT, MBT and ship bias corrections have complex structures in space and time, it may be the best method is to show the outcome for each set of bias corrections individually, rather than trying to quantify the spread and combine this with the other uncertainty components as per Equation (2.10). That is, if the spread cannot be adequately described by a metric like the standard deviation (for example, imagine if the result of applying the four XBT corrections produced three values clustered together with one outlier) then using the approach shown in Equation (2.10) would misrepresent the overall uncertainty.

2.6.5   Use cases: simple averaging, matchup analysis and data assimilation

We now illustrate the above guidance with a few simple examples. See Section 4 for a simple worked example in code form.

Consider three temperature observations made by surface-only platforms described by variables with the following properties:

plat_type	(7, 7, 5)
plat_id	('12345', '12345', 'SHIP-A')
temp	(12.0, 13.0, 14.2)
temp_type_corr	(0.0, 0.0, -0.2)
temp_plat_corr	(0.0, 0.0, 0.0)
temp_type_corr_unc	(0.0, 0.0, 0.02)
temp_plat_corr_unc	(0.29, 0.29, 0.71)

temp_rand_unc	(0.26, 0.26, 0.74)
temp_comb_unc	(0.39, 0.39, 1.03)
temp_qc	(0, 0, 0)

Here we have two drifting buoy (*plat_type* 7) temperature observations and one ship (*plat_type* 5) temperature observation that passed quality control (*temp_qc* = 0). Note that the two drifting buoy observations are from the same buoy (i.e. they have the same value for *plat_id*). The correction and uncertainty values are typical of what might be found in HadIOD.1.2.0.0 for drifting buoy and ship platform types.

Simple Average:

The corrected temperature is: *temp* + *temp_type_corr* + *temp_plat_corr* = (12.0, 13.0, 14.0)

The average corrected temperature = 13.0 °C

We now calculate the uncertainty of the average due to observation errors. We do this by separately calculating the three components of uncertainty as described above and then combining them using Equation (2.10).

First we calculate the sensitivity coefficients for our simple averaging function. Following Equation (2.5) c_i = 1/3.

We then calculate **component 1** of the uncertainty by substituting values into Equation (2.11) as:

$$u_c^2(y) = \left(\frac{1}{3}\right)^2 0.26^2 + \left(\frac{1}{3}\right)^2 0.26^2 + \left(\frac{1}{3}\right)^2 0.74^2$$

or in matrix form using Equation (2.7):

$$u_c^2(y) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.26^2 & 0.0 & 0.0 \\ 0.0 & 0.26^2 & 0.0 \\ 0.0 & 0.0 & 0.74^2 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Which gives u_c = 0.275 °C.

We calculate **component 2** of the uncertainty by substituting values into Equation (2.3) (the error covariance of paired observations is given by Equation (2.13)) as:

$$u_c^2(y) = \left(\frac{1}{3}\right)^2 0.29^2 + \left(\frac{1}{3}\right)^2 0.29^2 + \left(\frac{1}{3}\right)^2 0.71^2 + 2 \times \left(\frac{1}{3} \times \frac{1}{3} \times 0.29 \times 0.29 \times 1.0\right)$$
$$+ 2 \times \left(\frac{1}{3} \times \frac{1}{3} \times 0.29 \times 0.71 \times 0.0\right) + 2 \times \left(\frac{1}{3} \times \frac{1}{3} \times 0.29 \times 0.71 \times 0.0\right)$$

The full expansion of Equation (2.3) is shown above; note that the last two error covariance terms on the RHS are zero because in HadIOD.1.2.0.0 the correlation of the errors (and thus error covariance) is zero between pairs of observations not from the same platform for this component of uncertainty (see Section 2.6.3). In matrix form using Equation (2.7):

$$u_c^2(y) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.29^2 & 0.29^2 & 0.0 \\ 0.29^2 & 0.29^2 & 0.0 \\ 0.0 & 0.0 & 0.71^2 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Which gives u_c = 0.306 °C.

Component 3 of the uncertainty can be calculated following the approach described in Section 2.6.4, i.e. by applying multiple sets of ship corrections in turn to our ship observation, calculating the average in each case, and finally calculating the spread of the averages. However, because in this simple example case there is only one ship observation (and therefore we are not concerned about error correlation with other ship observations), the uncertainty in our ship observation due to this effect can simply be obtained from *temp_type_corr_unc*. By substituting values into Equation (2.3) (noting that there is no correlation of the errors, thus error covariance is zero between pairs of observations in this case):

$$u_c^2(y) = \left(\frac{1}{3}\right)^2 0.0^2 + \left(\frac{1}{3}\right)^2 0.0^2 + \left(\frac{1}{3}\right)^2 0.02^2$$

or in matrix form using Equation (2.7):

$$u_c^2(y) = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.02^2 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

Which gives u_c = 0.007 °C.

Combining the three components of uncertainty using Equation (2.10) gives an overall uncertainty for the average corrected temperature = 13.0 ± 0.411 °C.

Matchup Analysis:

When comparing one set of observations against another, one approach is to 'match' together pairs of observations that are close in space and time. We might then calculate the average of these 'matchup pairs' to see if there is any average difference between the two observation sets. For a set of *n* matchups we can write this as:

$$Y = \frac{\sum_{i=1}^{n}(X_i - Z_i)}{n}$$

where the set of observations X_i are matched to the set of observations Z_i. This could be rewritten as:

$$Y = \frac{\sum_{i=1}^{n}(X_i)}{n} - \frac{\sum_{i=1}^{n}(Z_i)}{n}$$

We can then calculate the overall uncertainty of each term on the RHS of this equation following the approach taken above for a simple average. These two components can then be combined in quadrature to get the overall measurement uncertainty in our matchup average. If one of the sets of observations are taken from a different dataset (for example if using HadIOD in situ observations to ground truth satellite observations) then the user will need to consider the uncertainty information provided by each dataset to get the overall matchup uncertainty.

Model Data Assimilation:

In data assimilation an observation error covariance matrix $\mathbf{U}_x$ (Equation (2.8)) is normally required[2]. This matrix gives the overall observation error uncertainties (as variances) in the diagonal elements and the covariances of the errors in the observation pairs in the off-diagonal elements. To get the

---

[2] In addition to observation measurement error this matrix may also incorporate observation representativity error, but here we will only deal with the measurement error information that HadIOD provides.

matrix we need to combine the components of uncertainty given above in matrix form. In the case of our simple example this would look like:

$$\mathbf{U}_x = \begin{bmatrix} 0.26^2 + 0.29^2 & 0.29^2 & 0.0 \\ 0.29^2 & 0.26^2 + 0.29^2 & 0.0 \\ 0.0 & 0.0 & 0.74^2 + 0.71^2 + 0.02^2 \end{bmatrix}$$

where the overall uncertainty for each observation (the lead diagonal) is the sum of the variance of each component of uncertainty, and the error covariance for each observation pair (the off-diagonal elements) is the sum of the covariance of each component of uncertainty.

A simplifying assumption is often made in data assimilation that the input observations are unbiased and that any uncertainty is due to random measurement errors (i.e. errors that are uncorrelated from one observation to the next). In this case the decomposition of the error into separate components (some of which are correlated, Equation (2.1)) would be ignored and the matrix would become:

$$\mathbf{U}_x = \begin{bmatrix} 0.26^2 + 0.29^2 & 0.0 & 0.0 \\ 0.0 & 0.26^2 + 0.29^2 & 0.0 \\ 0.0 & 0.0 & 0.74^2 + 0.71^2 + 0.02^2 \end{bmatrix}$$

with only the lead diagonal elements prescribed. For convenience these elements could be populated using the square of values from *temp_comb_unc* which provides the overall uncertainty in individual observations. This matrix would no longer fully describe the nature of the errors, but this form may be necessary to enable data assimilation activities; how the errors are ultimately handled will depend on the activity. Although the third component of the overall ship uncertainty (the 0.02^2 term) appears relatively small for individual observations, this component of uncertainty can become increasingly important on longer space and time scales (e.g. if assessing decadal trends) and should be explored through applying different sets of bias corrections (Section 2.6.4).

2.6.6 Caveats

Although HadIOD provides estimates of observational uncertainty, these do not capture the structural uncertainty arising from the fundamental choices made in the dataset construction process. Consequently, the uncertainty in any analysis based on HadIOD may be underestimated. Some sources of structural uncertainty might include:

- Using an alternative historical SST bias correction scheme to the HadSST.3.1.1.0 scheme chosen by HadIOD (these can differ in method and structure through time, see e.g. Kent et al. [2017]).
- Using an alternative XBT/MBT correction scheme not already included in HadIOD.
- Using a different observation error model and/or populating values in the observation error model differently.
- Using a different approach to recovering missing metadata (for example missing instrument/probe types, needed when assigning bias corrections and uncertainties)
- Using different data collections which may include: different observations and coverage (for example because of newly recovered historical observations, use of different real time data streams or differences in duplicate elimination); different quality control checks and decisions; differences arising from the back-end processing system etc.

This list is by no means comprehensive. Where possible, we would recommend that users consider repeating their analysis using different datasets to explore this structural uncertainty. This might mean trying an entirely different dataset to HadIOD or it could mean switching in and out alternative

dataset components to those in HadIOD, e.g. a user might blend HadIOD surface data with a different source of profile data or an alternative source of historical SST bias corrections. As HadIOD evolves we may further explore this structural uncertainty (e.g. by incorporating more sets of bias corrections or multiple sets of QC flags).

Be aware, as with any uncertainty analysis, there likely remain unknown sources of uncertainty (not explored by the structural uncertainty). If you have feedback on the uncertainty information or how it is provided, please let us know.

2.7   FAQs

2.7.1   Where can I find further information about HadIOD?

We would recommend reading Atkinson et al. [2014] (free access) for further information about how HadIOD is created. Please also contact us if you cannot find what you need.

2.7.2   What geophysical variables are included in HadIOD?

Temperature and salinity on depth below surface. At present, salinity is only available for profile observations. Temperatures are provided for both profile and surface-only observation types.

2.7.3   Is HadIOD.1.2.0.0 the dataset for me?

There are several providers of collections of in situ ocean observational data. The datasets provided have various features and advantages; the main features of HadIOD are given in Section 2.1.1. A brief comparison of the HadIOD public files with some other in situ marine datasets is given in the table below. Like HadIOD these are datasets created with research purposes in mind (e.g. reanalysis). Users with a requirement for near real time updates for operational applications such as forecasting might consider alternate data sources.

	Coverage and updates
	Observation source
	Observation types
	Geophysical variables
	Bias corrections
	Uncertainty information
	Quality Control
	Notes
	Reference
HadIOD.1.2.0.0 (public files)	Global, 1850-ongoing, monthly updates from obs. source (based on GTS/Argo) with approx. a few months lag.
	EN.4.2.[0,1,2], ICOADS.2.5.1 (ICOADS.3.0.2 from April 2022) and CMEMS (drifting buoys)
	Surface-only and profile data
	Temperature, salinity, depth
	XBT, MBT, ship, Argo-adjusted (via EN4)
	Measurement uncertainty for each observation; structural uncertainty via multiple XBT/MBT corrections and ship correction ensemble
	Separate Met Office QC suites for profile and surface observations
	-
	https://www.metoffice.gov.uk/hadobs/hadiod/, Atkinson et al. [2014]
CORA-GLOBAL	Global, 1950-ongoing, biannual updates with approx. 1 year lag.

(Coriolis Ocean database for ReAnalysis) Easy CORA	Snapshot of CORIOLIS database (real time and delayed mode ingestion of various data including main international projects: Argo, WOD, GOSUD, OceanSITES, GTSPP, ICES etc.; UDASH; merchant and research ships; European partners e.g. SeaDataNet; data from French projects) merged with EN4 profiles
	Profile data and some surface-only (buoys and ship TSG)
	Temperature, salinity, depth, pressure
	Argo (drift/offset adjustments from GDAC), XBT (Hammon et al., 2012)
	-
	Comprehensive Coriolis/CORA QC suite, manual review of doubtful profiles
	Easy CORA is an extraction of CORA-GLOBAL providing the best quality measurements with vertical and temporal sub setting. Objective analyses of CORA T and S profiles are also available
	http://www.coriolis.eu.org/Data-Products/Products/CORA; data and docs via CMEMS http://marine.copernicus.eu/services-portfolio/access-to-products product name: INSITU_GLO_TS_REP_OBSERVATIONS_013_001_b; Cabanes et al. [2013]
EN4	Global, 1900-ongoing, monthly updates from GTSPP/Argo, rebuilt roughly annually incorporating new historic and delayed mode data (latest version at 02/11/21 is EN.4.2.2), preliminary and final monthly updates (approx. 1.5 month lag for final)
	WOD, GTSPP, Argo GDAC, ASBO
	Profile data
	Temperature, salinity, depth
	XBT, MBT, Argo (drift/offset adjustments from GDAC)
	Structural uncertainty via multiple XBT/MBT corrections
	Comprehensive Met Office QC suite
	Some vertical and temporal subsetting of profiles with high sampling frequency. EN4 also provides objective analyses of T and S.
	https://www.metoffice.gov.uk/hadobs/en4/, Good et al. [2013]
WOD18 (World Ocean Database 2018)	Global, 1772-ongoing (pre-20th century data is limited but archived if available, earliest is currently 1772), NCEI maintain and update an archive of historical ocean measurements. A full QC'd WOD release is made every few years (e.g. WOD18 = 2018 release), preliminary quarterly updates with new or amended data.
	Blend of numerous historical and recent data sources, new additions as data is recovered or updated
	Profile data and specific surface-only
	temperature, salinity, pressure, depth, biological and chemical
	XBT, MBT
	Structural uncertainty via multiple XBT/MBT corrections
	Comprehensive WOD/WOA QC suite checking data formatting and reported values (automatic and subjective, extra checks for standard level data), originators QC flags for some data, errors corrected when possible in collaboration with originators and users. Only limited checks for some chemical/biological data.
	Provides comprehensive observation/source metadata, data available on observed and (interpolated) standard levels, climatologies from standard level data (World Ocean Atlas 2018)
	https://www.nodc.noaa.gov/OC5/WOD/pr_wod.html, Boyer et al. [2018]

IQuOD (International Quality controlled Ocean Database) preliminary version 0.1	Global, 1772-2018 (pre-20th century data is limited).
	WOD
	As for WOD
	Temperature, salinity, pressure, depth, oxygen, nutrients, ocean tracers, optics and biology (chlorophyll, plankton)
	XBT (Cheng et al., 2014), other corrections via WODselect.
	Measurement uncertainty for observations (temperature and some salinity and depth); structural uncertainty if using multiple XBT/MBT corrections via WODselect.
	Currently WOD QC, eventually automated QC from the subset of tests identified as giving the most accurate results from multiple other QC suites (including WOD and EN4). Expert QC will also be included in a later version.
	Intelligent metadata are provided for some XBTs missing probe type or manufacturer, currently based on Palmer et al. [2018], but with machine learning methods being actively explored in this area.
	http://www.iquod.org/
ICOADS (International Comprehensive Ocean-Atmosphere Data Set)	Global, 1662-ongoing (pre 20th century data is limited but archived if available, earliest is currently 1662), a new version of ICOADS is built every few years, extending coverage and incorporating new historical and delayed mode data and other improvements, the current version is ICOADS.3.1.0 (as of 21/01/20). Monthly updates using GTS data with 2-6 week lag.
	Blend of numerous historical and recent data sources, new additions as data is recovered or updated
	Surface-only
	Various surface marine and surface meteorological variables, depth for some obs via supplementary attachment
	Support for incorporating externally-developed adjustments via *Ivad* attachment (R3.0 includes prototype adjustments for visually-estimated winds and ship-based atmospheric temperature).
	Support for incorporating externally-developed uncertainty estimates via *Ivad* attachment.
	ICOADS basic QC checks. Support for linking alternate QC flags via *Ivad* attachment.
	Provides comprehensive observation/source metadata, monthly summaries from 1800 onwards.
	https://icoads.noaa.gov/, Freeman et al. [2017] (for ICOADS R3.0), Woodruff et al. [2011] (for ICOADS R2.5, used in HadIOD.1.2.0.0)

Table 1. A comparison of HadIOD with some other in situ marine datasets useful for research (i.e. not real time operational) purposes.

HadIOD gets its quality controlled ocean profile data from EN4 and as such there are significant similarities between these datasets. The chief difference is that HadIOD also includes quality controlled data from surface-only platforms, which will be helpful in certain applications such as satellite validation or ocean analyses where observations from the sub-surface ocean and near the ocean-atmosphere interface are needed for assimilation. HadIOD also includes a more thorough treatment of observations errors and uses its own optimal ragged array data format.

The locations of the observations in HadIOD vary in space and time. Users interested in area averages and timeseries or in spatially complete fields may find a gridded dataset more appropriate. The Met Office Hadley Centre provides various gridded ocean datasets that can be accessed through https://www.metoffice.gov.uk/hadobs/. They include:

- [HadSST3](#) and [HadSST4](#): gridded, uninterpolated and bias adjusted sea surface temperatures with uncertainties.
- [HadISST](#): globally complete sea-ice and sea-surface temperatures.
- [EN4](#): objective analyses with uncertainties formed from quality controlled and bias adjusted temperature and salinity profiles.

Many other institutions create their own gridded ocean datasets. These may be based on in situ observations, satellite observations, or a combination of both. Many datasets infill gaps in the data using statistical techniques. Spatially complete data are also provided by reanalyses, which assimilate data into a physical model.

2.7.4   When is HadIOD updated?

HadIOD is presently updated each month on a best endeavours basis. Updates are made a few months after the end of the target month (for example the January 2020 update would be made around late-March 2020). We plan to periodically rebuild HadIOD to incorporate the latest data and metadata (such as bias corrections and uncertainties or QC flags). There is no timescale for this but we would expect a few years at least between releases. Each new release of HadIOD will receive its own version number with the latest being HadIOD.1.2.0.0 (and the previous being HadIOD.1.1.0.0). The rightmost digits of the version number are reserved for tracking minor changes to the current release if necessary.

2.7.5   What are the sources of the observations?

Observations from several different sources have been ingested into the HadIOD database. Details of these are given below. Only the sources with an asterix have been included in the public release files described in this document. Some of the remaining sources have been included in other project-specific data extractions.

- *ICOADS.2.5.1 - [Woodruff et al., 2011]. The primary source of surface ocean measurements in HadIOD. These data are included from 1850-present and provide sea surface temperature measurements (not salinity) to HadIOD. See also Table 1 for information about ICOADS.

  Acknowledgement: Research Data Archive/Computational and Information Systems Laboratory/National Center for Atmospheric Research/University Corporation for Atmospheric Research, et al. 1984, updated monthly. International Comprehensive Ocean-Atmosphere Data Set (ICOADS) Release 2.5, Individual Observations. Research Data Archive at the National Center for Atmospheric Research, Computational and Information Systems Laboratory. https://doi.org/10.5065/D6H70CSV. Accessed 2011 onwards.

- *ICOADS.3.0.2. Used as a substitute for ICOADS.2.5.1 from April 2022 onwards (the last monthly update of ICOADS.2.5.1 was expected in March 2022 [pers. comms. ICOADS team]). ICOADS.3.0.2 (2015-present) extends the coverage of ICOADS.3.0 (1662-2014) [Freeman et al., 2017]. The ICOADS.3.0.2 near real time observations are based on blended marine observations in the TAC and BUFR formats from NOAA's National Centers for Environmental Information (NCEI) GTS collections. The ICOADS.2.5.1 near real time observations are based on GTS observations from NOAA's National Centers for Environmental Prediction (NCEP). Users should be vigilant for any impacts caused by the transition from ICOADS.2.5.1 to ICOADS.3.0.2 e.g. we note an increase in the number of ship observations across the transition.

- ***EN.4.2.0** / ***EN.4.2.1** / ***EN.4.2.2** - [Good et al., 2013]. The primary source of ocean profile data and quality flags in HadIOD. These data are included from 1900-present and provide temperature and salinity measurements on depth (also potential temperature) to HadIOD. From November 2017 the switch was made to using EN.4.2.1 as EN.4.2.0 was retired. From the perspective of HadIOD this switch from EN.4.2.0 to EN.4.2.1 should be seamless with no discernible impact on the data. From March 2021 the switch was made to using EN.4.2.2 ahead of EN.4.2.1 being retired. From the perspective of HadIOD this switch from EN.4.2.1 to EN.4.2.2 should be relatively seamless though there may be an impact on the corrections provided for XBT observations (see FAQs Section 2.7.7). See also Table 1 for information about EN4.

  EN4 acknowledges the datasets contributing to its own construction (for full details see the EN4 webpages) and which HadIOD benefits from. These are:

  - **World Ocean Database 2013** [Boyer et al., 2013; http://www.nodc.noaa.gov/OC5/WOD13/]. Note that EN.4.2.2 uses data from World Ocean Database 2018 (WOD18) but this WOD18 data is not included in HadIOD.1.2.0.0 because of the use of EN.4.2.2 only from March 2021 onwards.
  - **Global Temperature and Salinity Profile Project** [U.S. National Oceanographic Data Center: Global Temperature–Salinity Profile Programme. June 2006. U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Oceanographic Data Center, Silver Spring, Maryland, 20910, https://www.ncei.noaa.gov/products/global-temperature-and-salinity-profile-programme].
  - **Argo** [Argo data were collected and made freely available by the International Argo Program and the national programs that contribute to it. (https://argo.ucsd.edu, https://www.ocean-ops.org). The Argo Program is part of the Global Ocean Observing System].

- o **Arctic Synoptic Basin-wide Observations project** [Takamasa Tsubouchi at National Oceanography Centre, Southampton, U.K.].

- *E.U. Copernicus Marine Environmental Monitoring Service (CMEMS) drifting buoys* - product INSITU_GLO_NRT_OBSERVATIONS_013_030 (Global Ocean In Situ Near Real Time Observations), http://marine.copernicus.eu/. Daily updates of global NRT observations from the main global networks from 2010-present provided by Coriolis (France). CMEMS drifting buoy sea surface temperature data are included in HadIOD from February 2015 to present. In the public files CMEMS is used as a replacement for ICOADS.2.5.1 drifting buoy data from June 2016 onwards following a drop in the number of drifting buoy observations in ICOADS[3]. CMEMS drifting buoy data are also included in HadIOD for October 2014 and used in the public files from 22nd-23rd October 2014 to fill a gap where there are no ICOADS.2.5.1 data. The ICOADS.3.0.2 product (used in HadIOD from April 2022 onwards) has fixed this issue but we continue to use CMEMS as the source of drifting buoy data.

  Acknowledgement: HadIOD is generated using E.U. Copernicus Marine Service Information.

- **Met Office MetDB** - in November 2016, surface buoy data (drifting and moored) were absent from the ICOADS.2.5.1 Real Time updates. GTS buoy data from the Met Office Meteorological Database system were used in HadIOD as a substitute. MetDB data are not included in the public files which use CMEMS drifting buoys and EN4 moored buoys.

- **NOAA Pacific Marine Environmental Laboratory (PMEL) GTMBA** - http://www.pmel.noaa.gov/tao/. Used to provide high temporal sampling resolution sea surface temperature from the tropical moored buoy arrays to HadIOD. A download of this archive was made on 21/09/16 providing data from the 1980s to this date. This provides an alternate to ICOADS.2.5.1 which thins data to hourly sampling. Initially added for satellite validation activities.

- **NOAA Observing System Monitoring Center drifting buoys** - accessed via the ERDDAP data server (http://osmc.noaa.gov/erddap/index.html). From January 2018-March 2022 drifting buoy sea surface temperatures are ingested into the HadIOD processing system for QC to mitigate falling ICOADS.2.5.1 buoy numbers. These are not made available to users. This is not required from April 2022 onwards when ICOADS.3.0.2 is used by HadIOD.

2.7.6   How do I get multiple sets of the bias corrections?

The main HadIOD data files and files containing additional ship, XBT and MBT corrections are available from https://www.metoffice.gov.uk/hadobs/hadiod/download-hadiod1-2-0-0.html. See Section 3 for information on file format and contents.

2.7.7   Where do the bias corrections and uncertainty estimates come from?

The XBT and MBT corrections are from:

- Gouretski and Reseghetti [2010]. XBT and MBT corrections.
- Levitus et al. [2009]. XBT and MBT corrections.
- Gouretski [2012]. XBT corrections.

---

[3] This was caused by a switch in GTS encoding from TAC to BUFR form. ICOADS are addressing this for later ICOADS releases, see https://icoads.noaa.gov/ for the latest news and updates.

- Cowley et al. [2013]. XBT corrections.

We provide Gouretski and Reseghetti [2010] XBT and MBT corrections as standard in the main files. Corrections for the other schemes are provided in a separate set of files (Section 3.3.3). For the interested reader a helpful summary of XBT science is provided by Cheng et al. [2016]. Listings of XBT and MBT correction schemes and any updates (e.g. to correction coefficients) can be found at:

- https://www.nodc.noaa.gov/OC5/XBT_BIAS/xbt_bias.html
- https://www.nodc.noaa.gov/OC5/mbt-bias/

XBT and MBT corrections are calculated based on those used in EN4. In March 2021 HadIOD switched to using EN.4.2.2 (see FAQs 2.7.5) which does not provide Gouretski [2012] XBT corrections, thus February 2021 is the last month these corrections are provided. For the remaining XBT correction schemes EN.4.2.2 incorporates some updates to the correction scheme parameters, how the corrections are applied and the XBT probe-type decisions. It is expected that the transition to using the EN.4.2.2 corrections should be fairly seamless but users should be aware of this change (it is also noted that the amount of XBT data in 2021 is relatively small). More detailed information can be found in the EN4 user guide (see Table 1 for links to EN4).

The ship corrections are based on those used for HadSST.3.1.1.0. The HadSST3 bias correction scheme is described in Kennedy et al. [2011b]. A 100-member ensemble of bias corrections is provided by HadSST3 which explores uncertainty in the ship correction parameters. A single set of ship corrections is provided as standard in the main files which we refer to as the 'Mean Ensemble Member'; this is intended to lie somewhere near the centre of the ensemble but is **not** a best estimate (all ensemble members are equally valid). The parameters chosen to create the 'Mean Ensemble Member' are described in Atkinson et al. [2014]. Bias corrections for the 100-member ensemble are provided in a separate set of files (Section 3.3.2).

Measurement uncertainties are allocated to different platform/probe types based on values found in the literature. For surface-only temperature measurements we chiefly relied on Kennedy et al. [2011a]. For sub-surface profile measurements we chiefly relied on Abraham et al. [2013] and World Ocean Database documentation [Boyer et al., 2009; Boyer et al., 2013].

The bias correction and uncertainty values allocated to the observations are summarised in Table 2 and Table 3. For full details please see Atkinson et al. [2014].

2.7.8 How are the derived variables calculated?

The potential temperatures and some depths in HadIOD are derived from other variables. These derived variables are calculated by the EN4 system. More information can be found in the EN4 user guide (see Table 1 for links to EN4). Please contact us if further information is required.

2.7.9 How do I find observations from the same platform?

An identifier for a particular platform is stored in the variable *plat_id* (see also Section 3.2.2). All observations from the same platform should have the same *plat_id* and the same platform type (stored in the variable *plat_type*). In practice this may not always be the case as *plat_id* identifiers may be missing or unusable (e.g. due to data transcription errors).

In particular there are some known issues with the ICOADS.2.5.1 ship identifiers that users should be aware of:

1. From December 2007 onwards, for security reasons most ship callsigns in ICOADS.2.5.1 are masked with the generic ID 'MASKSTID'. To help users track ships, HadIOD unmasks instances of 'MASKSTID' where possible using data kept by the Met Office. These unmasked IDs cannot be released externally, thus a unique masked ID of form 'MKnnnnnnn' (where nnnnnnn is a 7-digit zero-padded integer) has been mapped to each unique Met Office unmasked ID for use in the public files. These are stored in the *plat_id* variable and can be used to track observations from the same ship. Prior to December 2007 the original ICOADS callsigns are provided in the *plat_id* variable. Users will not be able to track ships across the transition from using ICOADS IDs to using HadIOD masked IDs. From April 2022 onwards HadIOD uses ICOADS.3.0.2 which does not mask ship callsigns, however the decision was made to continue using HadIOD masked IDs to facilitate tracking across the transition from ICOADS.2.5.1 to ICOADS.3.0.2.

2. Some ship observations report callsigns that are missing or unusable or fragment ship voyages into chunks. At some times there is very little callsign information (e.g. the 1860s). More information and a method for recovering useful callsign information can be found in e.g. Carella et al. [2017]. This method or similar may be considered for a future version of HadIOD. Where observations from different ships report under the same generic callsign (e.g. 'SHIP') this prevents the identification of individual ships. Some known generic callsigns to look out for are: SHIP, SHIPX, 0120, PLAT, RIGG, 1, 58, 7, MASKST, MASKSTID, (missing).

All observations within a vertical profile share the same profile metadata including *plat*_id. The *rowSize* variable can be used to gather together observations into profiles (Section 3.2.1). Each profile in a file receives its own unique ID (stored in the *profile* variable).

2.7.10 How is the observation quality flag generated?

The observations in HadIOD have passed through different QC suites dependent on the observation source. The QC flag is set if an observation fails any of the checks within a QC suite. The QC checks include checks on metadata (e.g. checks for valid position, time and depth) as well as the observed value, if the QC flag for an observation is set it could indicate a problem with the metadata, the observed value, or both. For profile data, if a profile as a whole fails QC (e.g. due to an invalid position or a problem with the instrument) then all observations within that profile will have their QC flag set.

Flags for each individual check in a suite are retained in the HadIOD database but for simplicity are not included in the public release files. If flags for individual checks would be helpful to you then please contact us.

QC flags for the profile observations are taken from EN4 [Good et al., 2013]. QC flags for the surface temperature observations are created using the Met Office Hadley Centre MDS QC suite (https://github.com/ET-NCMP/MarineQC). In addition, ICOADS.2.5.1 drifting buoy observations from 1986-2015 are assessed using an updated version of the drifting buoy QC checks described in Atkinson et al. [2013].

2.7.11 What about duplicate observations?

ICOADS and EN4 both carry out comprehensive duplicate elimination when they bring together observations from different data sources. HadIOD carries out an additional duplicate elimination step when bringing together ICOADS and EN4 (1900-onwards), as near-surface ICOADS observations may be duplicates of the surface-most observations in the EN4 profiles. Potential ICOADS duplicates are flagged based on an observation's type, data source and platform ID (note that EN4 observations are preferentially retained so as to keep profiles intact and also because they provide a depth). The

ICOADS data that are flagged are GTMBA moored buoys and near-surface profile data taken from WOD. Other moored and drifting buoys may also be flagged if their ID is found in EN4 for the same month. CMEMS drifting buoy data (which substitute for ICOADS drifting buoy data from June 2016) are not flagged as no duplication is expected with EN4. Further details of this are given in Atkinson et al. [2014]. Observations flagged as duplicates are not included in the public files.

2.7.12 Where does the observation metadata come from?

The basic observation metadata (e.g. time, position, platform, depth, observed value) are largely taken directly from the source datasets. Value added metadata such as bias corrections, uncertainties and QC flags are added by HadIOD. In some cases, basic metadata may be modified or inferred by HadIOD:

- For ICOADS.2.5.1: HadIOD unmasks ship callsigns from December 2007 onwards (Section 2.7.9). Some missing platform types have been inferred based on deck (see Atkinson et al. [2014]) and depths have been inferred based on platform type (see Table 2). Note that in the 1850s a large proportion of ICOADS observations do not provide a time (only a date) and thus fail QC, users not requiring a time might consider using these observations. For a very small number of ICOADS.3.0.2 observations (used in HadIOD from April 2022 onwards) a platform type has not been inferred and these receive a platform type code of 20 (unknown). HadIOD does not assign depths, uncertainties or corrections to these observations and users will need to decide how to handle them (e.g. exclude them from analysis).
- For EN4, various platform type information is provided (more information can be found in the EN4 user guide, see Table 1 for links to EN4). This has been synthesised into a single platform type code (see Section 3.2.3.1). In some cases the information provided by EN4 is not sufficient for a platform type to be inferred by HadIOD and these observations receive a platform type code of 20 (unknown). This mainly occurs from 1990 onwards and generally accounts for fewer than a few percent of the observations in a month, but can approach nearer 5 percent (see Figure 1 where observations are plotted with '*missing*' platform type). HadIOD does not presently assign bias corrections and uncertainties to these observations and users will need to decide how to handle them (e.g. exclude them from analysis, assign them the largest profile observation measurement uncertainty possible at the time using Table 2 or Table 3, etc.). Better handling of these observations is something HadIOD could improve on in the future.

The HadIOD public files include the basic metadata we would expect most users to need. More metadata is available in the HadIOD database (e.g. individual QC flags, decisions on ship measurement type for assigning bias corrections, observation provenance information etc.). If you do not see what you need in the HadIOD public files then please contact us, we may be able to extract extra metadata from the HadIOD database.

2.7.13 What is the HadIOD database identifier?

An integer ID is assigned (by us) to each observation in the HadIOD database. When combined with an observation's year and month this ID uniquely identifies an observation within the version of the database used (e.g. HadIOD.1.2.0.0). In the HadIOD public files these integer IDs are stored in the *temp_id* and *sal_id* data variables. They are helpful if users want to provide feedback about the observations (e.g. if an observation is found to be bad, it could be identified using its ID, year and month). We would recommend users propagate this ID through their application to enable feedback if required. The observation IDs are also used to link data in the main files with data in the files containing additional bias corrections.

The HadIOD database retains traceability to its source data by using source observation IDs. Where a source provides a (ideally unique) observation ID this is included in the HadIOD database to enable traceability (see also Atkinson et al. [2014] for further information). In the HadIOD database observations have both a 'source observation ID' and a 'integer ID' described above. To minimise storage the 'integer id' is used in the HadIOD public files as the 'source observation ID' can be potentially a long alphanumeric string. The link between the 'integer ID' and the 'source observation ID' is preserved in the HadIOD database. If this is of interest then please contact us.

2.7.14 How does HadIOD.1.2.0.0 differ from earlier versions?

Current version

HadIOD.1.2.0.0

HadIOD.1.2.0.0 is considered the first official release of HadIOD. It was created as part of the ERA-CLIM2 project (European Reanalysis of the Global Climate System, http://www.era-clim2.eu/) with contributions from the ESA SST CCI II project (European Space Agency Sea Surface Temperature Climate Change Initiative Phase 2, http://www.esa-sst-cci.org/) and is currently maintained as part of C3S (European Union Copernicus Climate Change Service, https://www.copernicus.eu/en, C3S_312a Lot 3). It is supported by the Met Office Hadley Centre Climate Programme funded by BEIS and Defra. HadIOD.1.2.0.0 is similar to its predecessor HadIOD.1.1.0.0. The main differences (in no particular order) are as follows:

- Fresh run of sea surface temperature QC.
- Addition of unmasked ship callsigns (December 2007 onwards) enabling ship tracking.
- Addition of an ensemble of ship corrections from HadSST.3.1.1.0.
- Use of EN.4.2.0, EN.4.2.1 and EN.4.2.2 (see FAQs Section 2.7.5) plus bug-fixed observation unique IDs and thinning of glider data.
- 4 XBT correction schemes and 2 MBT correction schemes applied to monthly updates.
- Addition of PMEL hi-res GTMBA.
- Addition of CMEMS drifters and MetDB data.
- Improvements to buoy tracking QC (tracks the quality of drifting buoy data over time).
- Extraction of data into CF compliant NetCDF4 format for public release via webpages.
- Improved output verification tools and general bug fixes and system improvements.

Previous versions

HadIOD.1.1.0.0

HadIOD.1.1.0.0 was created as part of the ERA-CLIM2 project, the follow up to ERA-CLIM (http://www.era-clim.eu/). It is similar to its predecessor HadIOD.1.0.0.0. The main differences (in no particular order) are as follows:

- Use of EN.4.1.0 as the source of sub-surface profile observations.
- Surface-only observation temporal coverage extended back to 1850 from 1900.
- 4 XBT correction schemes and 2 MBT correction schemes (see FAQs Section 2.7.7).
- Implementation of monthly updates.
- Tool to extract data into NEMO NetCDF feedback file format for assimilation in ocean models.
- General bug fixes and system improvements.

Data were made available to project participants in ASCII and NEMO NetCDF feedback format (see Section 2.7.15).

HadIOD.1.0.0.0

HadIOD.1.0.0.0 was the first version of HadIOD. It was created as a prototpye for the ERA-CLIM project and is described in Atkinson et al. [2014]. Data were made available to project participants in ASCII and ODB formats (see Section 2.7.15). It covered 1900-2010.

2.7.15 Is HadIOD available in other file formats? Haven't I seen HadIOD elsewhere?

Prototype data from HadIOD.1.0.0.0 were made available within the ERA-CLIM (European Reanalysis of the Global Climate System, http://www.era-clim.eu/) project in ASCII format and European Centre for Medium-Range Weather Forecasts (ECMWF) ODB (Observation Data Base) format. The ODB data was archived on MARS (Meteorological Archival and Retrieval System) at ECMWF and some ASCII data were available as a pre-release to Met Office colleagues. Some near-surface data were also extracted in ASCII format for use in the ESA SST CCI II project (European Space Agency Sea Surface Temperature Climate Change Initiative Phase 2, http://www.esa-sst-cci.org/).

Data from HadIOD.1.1.0.0 were made available within the ERA-CLIM2 (http://www.era-clim2.eu/) project in ASCII and NEMO NetCDF feedback format (hereafter 'NetCDF feedback format', a format used for model data assimilation that has some similarities to the EN4 data format, see https://forge.ipsl.jussieu.fr/nemo/wiki/Documentation: NEMO-OCE pdf). Some NetCDF-feedback and ASCII data were available as a pre-release to Met Office colleagues.

Data from HadIOD.1.2.0.0 were made available within the ERA-CLIM2 project in ASCII and NetCDF feedback format. Some NetCDF-feedback and ASCII were available as a pre-release to Met Office colleagues. The NetCDF-feedback format has been through a few minor iterations as a result of Met Office user feedback (each of these has a separate version number). Some near-surface data were extracted in NetCDF format for use in the ESA SST CCI II project and continue (as of 2020) to be extracted in this format for C3S (European Union Copernicus Climate Change Service, https://www.copernicus.eu/en, C3S_312a Lot 3). Data for public release (described in this document) have also been extracted into CF-compliant NetCDF4 format.

Note that different data may have been extracted from the HadIOD database into the various output datasets dependent upon requirements. For example, the ESA SST CCI II GTMBA data is a blend of ICOADS and high-resolution NOAA PMEL data, whereas the public release GTMBA data excludes the NOAA PMEL data and favours EN4 over ICOADS in the duplicate elimination process. Keep in mind that HadIOD is just a relational database of observations and metadata and these can be sliced and extracted in different ways into different formats.

The variety of past output formats reflects the development of HadIOD from prototype to first public release. Going forwards users will be directed towards the public CF-compliant NetCDF files served via https://www.metoffice.gov.uk/hadobs/hadiod/ as the primary source of HadIOD data (though production of C3S NetCDF data will be maintained for use within that project and NetCDF feedback format data will remain available internally within the Met Office).

2.7.16 How do I cite the HadIOD public data files?

If you use HadIOD data we would be grateful if you could let us know for our records.

When publishing work using the data, please quote the product version number, the date the data were downloaded and cite the following:

Atkinson, C. P., N. A. Rayner, J. J. Kennedy, and S. A. Good (2014), An Integrated Database of Ocean Temperature and Salinity Observations, J. Geophys. Res. Oceans, 119, 7139-7163, https://doi.org/10.1002/2014JC010053.

For example:

'The observations used are taken from HadIOD [Atkinson et al., 2014] version 1.2.0.0; downloaded from https://www.metoffice.gov.uk/hadobs/hadiod on [date downloaded].'

Please also read the dataset terms and conditions (FAQs Section 2.7.17) for further licencing requirements when citing the data.

If applicable, for traceability, please describe which HadIOD bias corrections were used. This information can be found in the NetCDF file global attribute '*source*'. This is particularly important if any additional corrections were used. See FAQs Section 2.7.6 and Section 2.7.7 for more information and references. For example:

'Bias corrections provided by HadIOD using the Gouretski and Reseghetti [2010] MBT/XBT correction scheme and ship 'Mean Ensemble Member' correction scheme were applied. An alternate set of XBT bias corrections were also applied provided by HadIOD using the Cowley et al. [2013] correction scheme.'

Gouretski, V., and F. Reseghetti (2010), On depth and temperature biases in bathythermographic data: Development of a new correction scheme based on analysis of global ocean databases, Deep Sea Res., Part I, 57, 812-833, https://doi.org/10.1016/j.dsr.2010.03.011.

Cowley, R. W., S. Wijffels, L. J. Cheng, T. P. Boyer, and S. Kizu (2013), Biases in expendable bathythermograph data: A new view based on historical side-by-side comparison, J. Atmos. and Ocean. Tech., 30, 1195-1225, https://doi.org/10.1175/JTECH-D-12-00127.1.

2.7.17 What are the HadIOD public data files' Terms and Conditions?

Data are made available without charge for Non-Commercial purposes under a Non-Commercial Government Licence.

Please read the full dataset terms and conditions here:

- *https://www.metoffice.gov.uk/hadobs/hadiod/terms_and_conditions.html*

2.8   Basic data statistics

The following figures give a basic overview of the observations in HadIOD.1.2.0.0 and how they change over time. In general, the oceans become better sampled in space (horizontally) and depth (vertically) as time progresses. For further description we would recommend reading Atkinson et al. [2014].

Figure 1 shows the number of observations in the HadIOD.1.2.0.0 files broken down into broad observation types. Figure 2 and Figure 3 give a feel for the changing coverage of the observation network for temperature and salinity respectively.

Figure 4 illustrates the platform type biases allocated to ship observations in HadIOD.1.2.0.0.
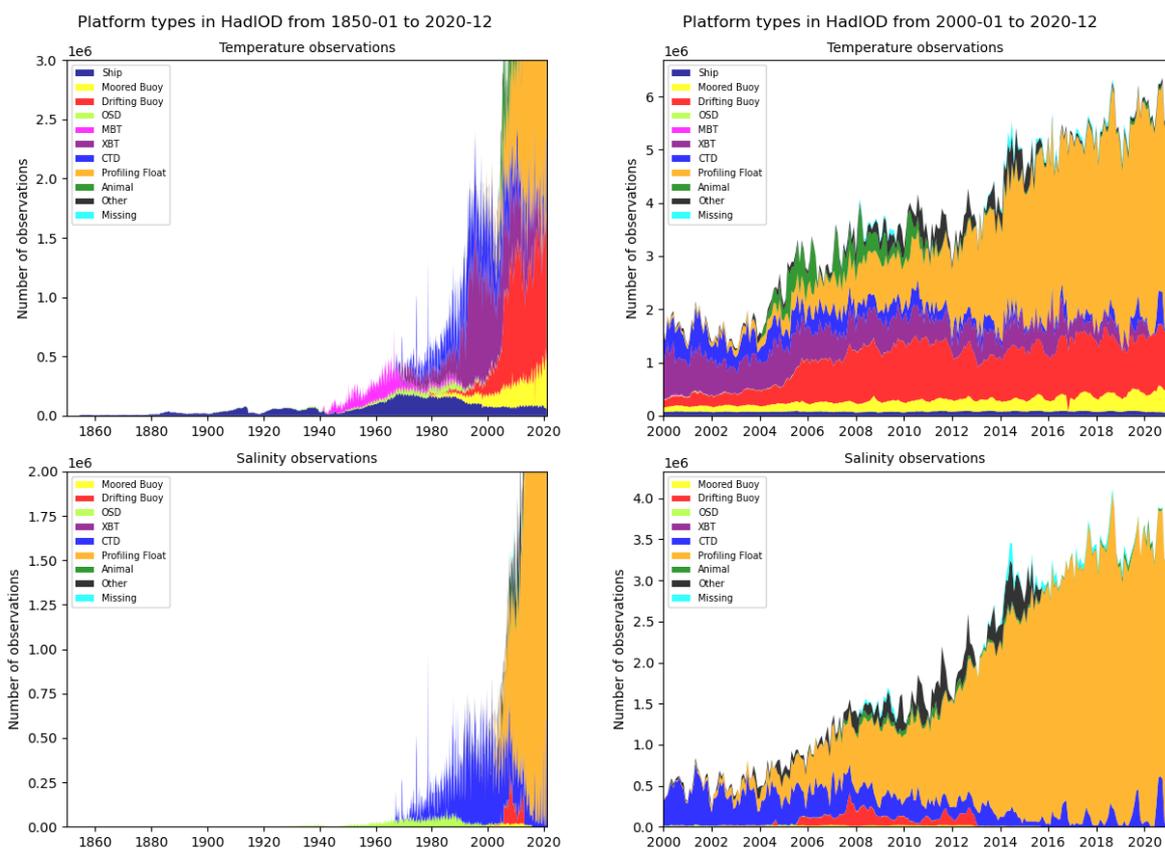
Figure 1. Monthly numbers of (top) temperature and (bottom) salinity observations in HadIOD.1.2.0.0 files (right column shows January 2000 onwards only) passing quality control and duplicate checks. Observations are grouped into categories based on the ICOADS platform ID scheme. Coloured bars are stacked from bottom to top in the same order as the line labels. Ship includes data from ocean station vessels; Ocean Station Data (OSD) includes low vertical resolution data from both bottle and Conductivity Temperature Depth (CTD) profiles; MBT=Mechanical BathyThermograph data; XBT=expendable BathyThermograph data; Profiling Float is chiefly data from the Argo programme; CTD contains high vertical resolution profile data from CTDs and expendable CTDs; Other is an amalgamation of platform types contributing a smaller number of observations and includes data from the U.S Coastal-Marine Automated Network, ocean drilling rigs/platforms and other small entities, undulating oceanographic recorders and gliders; Missing are observations for which platform type metadata is absent.
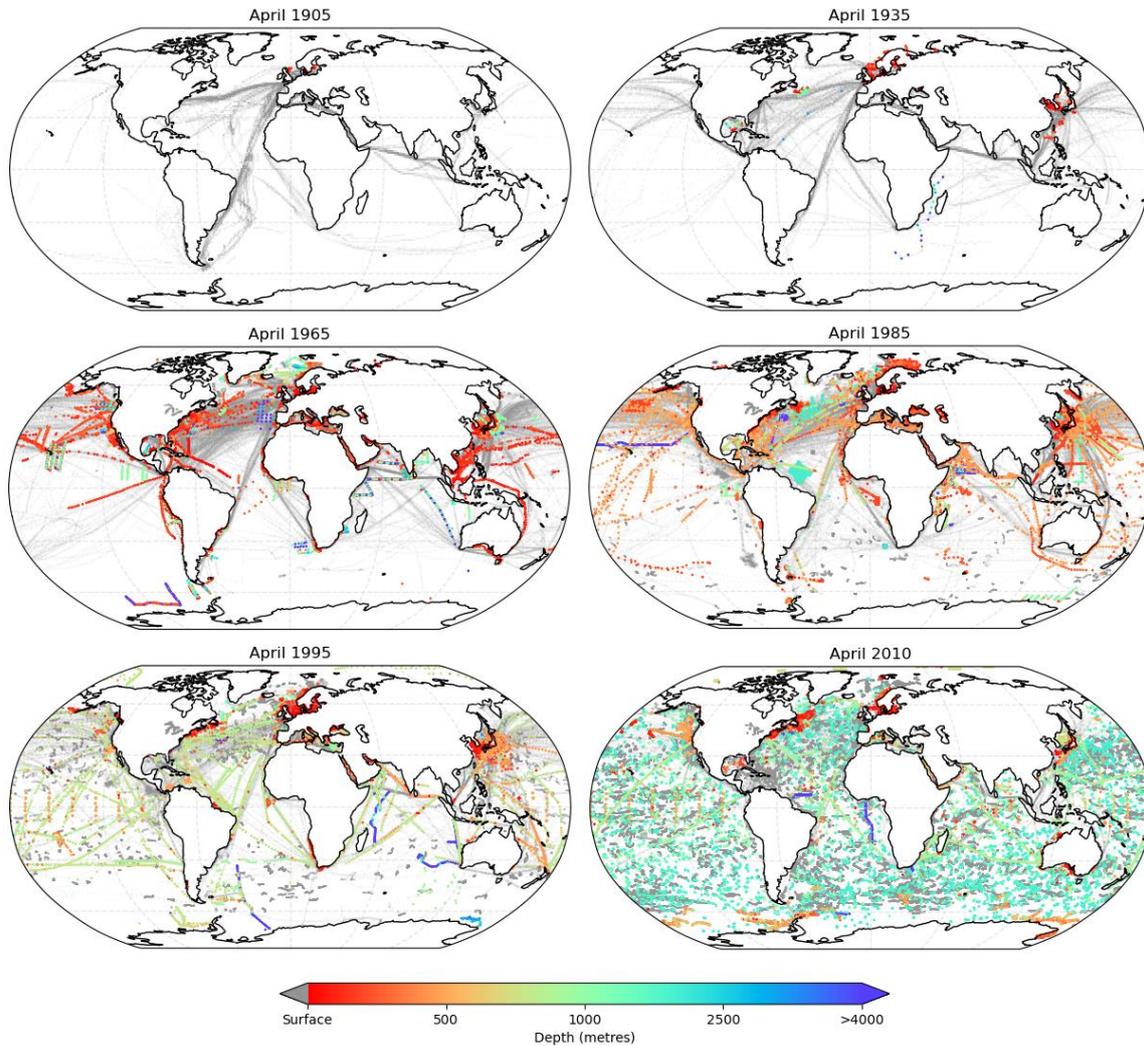
Figure 2. Maps showing maximum observed depths (metres) of temperature observations in HadIOD for April 1905, 1935, 1965, 1985, 1995 and 2010. Only observations that pass quality control checks and duplicate checks are shown. For ocean profile observations (coloured crosses), each of which comprises a series of measurements made at multiple depths at some location and time, maximum observed depth is the depth of the deepest measurement in a profile. For surface-only observations (grey dots), each of which comprises a measurement made at a single (near-surface) depth at some location and time, maximum observed depth is simply the depth of the measurement.
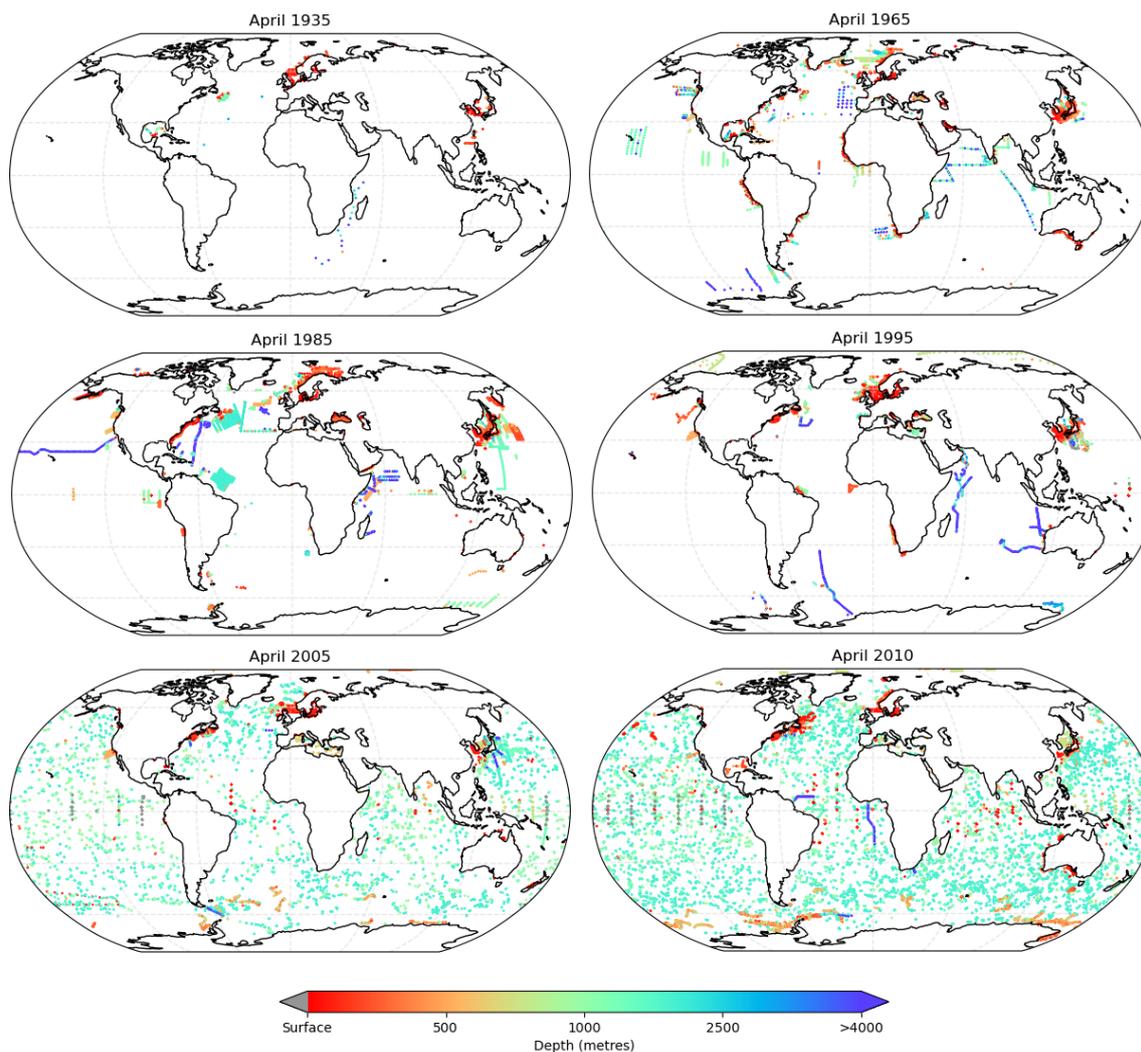
Figure 3. Maps showing maximum observed depths (metres) of salinity observations in HadIOD for April 1935, 1965, 1985, 1995, 2005 and 2010. Only observations that pass quality control checks are shown. Coloured crosses correspond to the deepest measurement depths in ocean profile observations, each of which comprises a series of measurements made at multiple depths at some location and time. At present no salinity observations are provided by surface-only platforms in HadIOD.1.2.0.0.
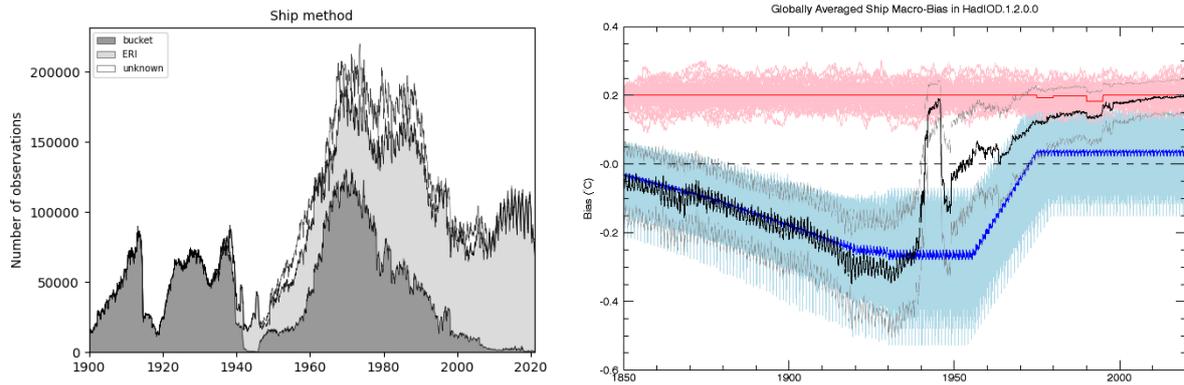
Figure 4. (left panel) Monthly numbers of SST observations from ships 1900-2020 in HadIOD.1.2.0.0 for different measurement methods following metadata recovery. Buckets are shown in shaded dark grey, engine room intakes in shaded light grey, and unknowns in white (note measurements were all from buckets prior to 1900), based on Figure 1 of Kennedy et al. [2011b]; (right panel) Monthly globally averaged SST platform type bias in ship observations in HadIOD.1.2.0.0 (⬚C) (note that <u>biases</u> are shown here, in the files we provide <u>bias corrections</u> that remove these biases). 100 realisations of HadSST.3.1.1.0 bucket (light blue) and engine room intake (light red) biases are shown, with the mean ensemble members used for allocating biases in the HadIOD.1.2.0.0 main files overlaid (dark blue and dark red lines). Platform type biases are allocated to individual ship observations based on their inferred measurement method (left panel). Monthly averages of the ship platform type biases (black) and ship platform type bias uncertainties (grey, denoting +/- 2 std.dev.) in the HadIOD.1.2.0.0 main files are shown. The 100 realisations can be used to explore uncertainty in the ship platform type bias, noting that the mean ensemble member used in the main files is simply semi-central to the ensemble and is not a best estimate. See Sections 2.4, 2.5 and 2.6 for more information.

2.9   Contact us

See https://www.metoffice.gov.uk/hadobs/hadiod/contact.html for latest contact details.

3   Public data files

3.1   File names

The bulk of the data is provided in the 'main' files. Additional sets of bias corrections for ship and XBT/MBT temperature observations are provided in supplementary files. Files are daily and named as follows (YYYY=year, MM=month, DD=day):

- Main files: hadiod1200_YYYYMMDD.nc,
  e.g. hadiod1200_20190630.nc
- Ship correction files: hadiod1200_shipcorrs_YYYYMMDD.nc,
  e.g. hadiod1200_shipcorrs_20190630.nc
- XBT/MBT correction files: hadiod1200_btcorrs_YYYYMMDD.nc,
  e.g. hadiod1200_btcorrs_20190630.nc

The daily data files are stored in zip files. Each zip file contains a year of data (except monthly updates which contain one month of data). Zip files can be unpacked using standard utilities such as unzip on Linux systems or 7-Zip on Windows systems. Data are available from: https://www.metoffice.gov.uk/hadobs/hadiod/download-hadiod1-2-0-0.html.

3.2   Main file contents

3.2.1   File format

The data are made available in NetCDF file format. NetCDF files are a platform-independent, self-describing, binary format and there are a number of common tools (Section 3.3) that can be used to access the data. Section 4 includes a worked example in Python. We use the NetCDF-4 format, making use of per variable compression. To represent the data we adopt version 1.7 of the NetCDF CF metadata conventions (hereafter CF1.7), which is a community standard for the exchange of climate data. In practice this means that some variable attributes (such as 'standard_name') and the way the data are stored follow certain conventions such that the data are fully described using an agreed standard. Detailed information can be found at http://cfconventions.org.

HadIOD.1.2.0.0 provides in situ observations of two observed geophysical variables: temperature and salinity. At present salinity and also potential temperature are only available for sub-surface profile data. Surface and sub-surface observed data are stored together in the same files. Each file contains a single day of data. Files contain observed values, metadata, bias corrections, bias correction and measurement uncertainties, overall quality control decisions and observation unique identifiers. There are several different bias correction and uncertainty fields (Section 2.4). It is not always possible to specify complete correction and uncertainty information for each measurement but at a minimum we try to provide some estimate of measurement uncertainty (in a minority of cases this may not be possible, see FAQs Section 2.7.12); unspecified values are set to the missing data value. We provide a quality flag for each observation.

A full description of the file contents is given in Section 3.2.2 and code tables used by certain variables are given in Section 3.2.3. For reference a sample file header can be seen at the end of this document.

For the HadIOD public files we have adopted the CF1.7 **contiguous-ragged-array representation** which is appropriate for collections of ocean profile data[4] where the number of vertical levels for each profile varies, and one can control the order of writing. In this representation the instances of the individual profiles are stacked sequentially along the same array dimension such that each profile is provided with the minimum amount of space that it requires to optimise storage. A helpful visualisation and example of the contiguous-ragged-array format is given in Sections 9.3.3 and H.3.4 of the CF-1.7-documentation.

A feature of HadIOD is to merge together observations of the surface and sub-surface ocean into a single dataset. To enable surface-only (single sampling depth) and profile (multiple sampling depths) data types to be stored together we treat the surface-only data as single-depth profiles. CF1.7 considers surface-only observations to be 'point' data. The mixing of different data types within a file (in this case 'point' and 'profile') is not at present CF compliant **thus our files are CF1.7 compliant with this single exception**. Allowing different data types to be combined is something under consideration for future CF versions.

Each file contains a collection of profiles. The profile dimension specifies the number of profiles in the collection. Variables having the profile dimension are profile variables and provide metadata about individual profiles (for example space and time co-ordinates of a profile or the ID of the platform that made a profile). Each profile is composed of a series of data elements (observed vertical levels). The obs dimension specifies the total number of data elements for all profiles in a collection. Variables having the obs dimension we refer to as data variables. The data variables provide information about profile data elements (for example: observed temperature or salinity; observation depth; any metadata potentially unique to an observation such as a bias correction). The dimensions, profile variables, data variables and global attributes present in the files are listed in Section 3.2.2.

All data are stored in single-dimension arrays. For the data variables, data for each profile (consisting of observations made at one or more vertical levels) are concatenated head-to-tail into one-dimensional arrays. Different geophysical variables (in this case temperature and salinity) measured by the same profile are stored in separate single dimension arrays. So, for example, if the first profile stored had measured both temperature and salinity at three depths, then elements 1-3 of '*temp*' would correspond to the profile's temperature measurements and elements 1-3 of '*sal*' would correspond to the profile's salinity measurements. If only one geophysical variable were observed (normally temperature) the corresponding elements of the unobserved geophysical variable would be stored as missing data.

The **rowSize** variable is a special profile variable required by the contiguous-ragged-array format; it gives a count of the number of elements (observed levels) in each profile. Surface-only observations (which are treated as single-depth profiles) receive a rowSize of 1. The rowSize variable enables the one-dimensional data arrays to be partitioned into their individual constituent profiles and then mapped to their profile metadata stored in the profile variables.

To get data for profile N, the (N-1) rowSize counts are summed, and the pointer moved to this element position in each data array. The next rowSize(N) elements in each data array are the data values for profile N. The corresponding profile metadata is stored in element N of each profile variable.

---

[4] *A profile is a series of connected observations along a vertical line, like an atmospheric or ocean sounding. For each profile, there is a single time, latitude and longitude.*

For example:

- Imagine that the first three values of rowSize are 3, 1 and 5. This tells us that the first three profiles stored head-to-tail in our one-dimensional data arrays comprise 3, 1 and 5 data elements (observed levels). Thus data for the first, second and third profiles are stored in elements 1-3, 4 and 5-9 of the data arrays respectively. The profile metadata for these profiles are stored in the first three elements of the profile variables i.e. elements 1, 2 and 3 of each profile variable (e.g. time or longitude) correspond respectively to elements 1-3, 4 and 5-9 of each data variable (e.g. temp or depth). An illustration is given below.
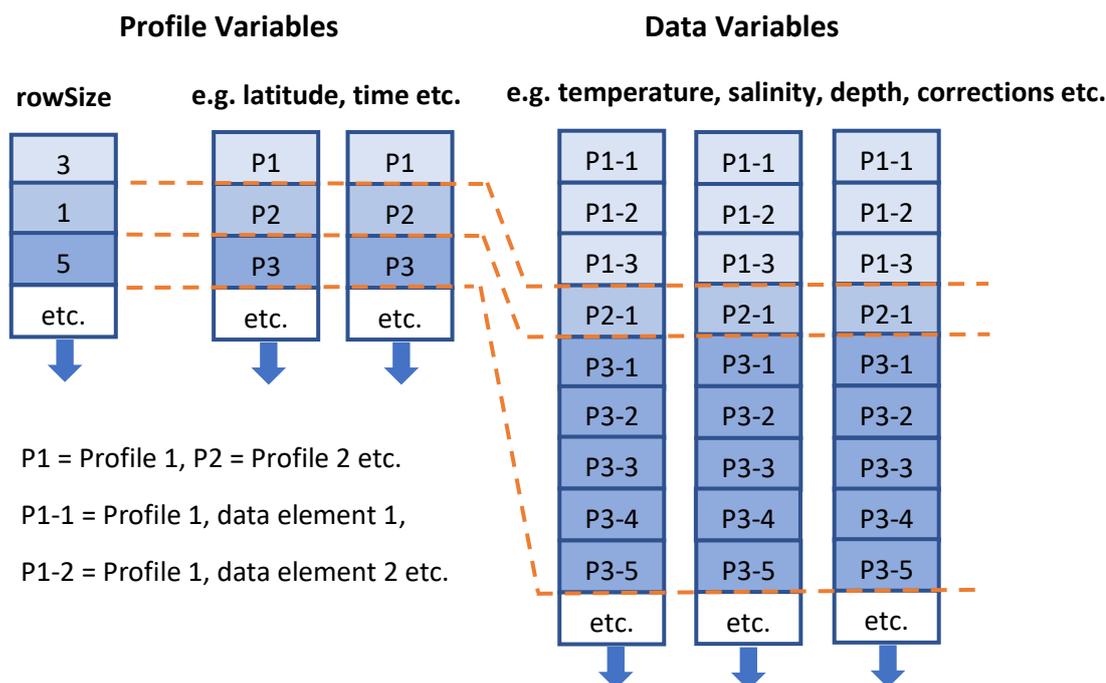
3.2.2 File variables

The contents of each file are listed below. For reference a sample file header can be seen at the end of this document. To save space in the table below some data variable names use the prefix [temp/sal]. This indicates there are separate named instances of the variable in the files for use with either the temperature or salinity variables (e.g. [temp/sal]_qc indicates there is a variable named *temp_qc* for use with the *temp* variable and a variable named *sal_qc* for use with the *sal* variable).

Further information about each variable (e.g. standard name, units, missing data values etc.) can be found in the file header variable attributes. Standard name meanings can be found in the CF Standard Names Table (http://cfconventions.org/). A full explanation of the formatting can be found at http://cfconventions.org/Data/cf-conventions/cf-conventions-1.7/cf-conventions.html. For more information about the bias correction and uncertainty variables see Sections 2.4, 2.5 and 2.6.

Some variables are classed as ancillary variables of other variables (i.e. they provide metadata about the individual values of another data variable). For example the *temp_qc* variable is an ancillary variable of the *temp* variable. The ancillary variables of a variable are listed in its ancillary_variables attribute.

NetCDF variable	Description
Dimensions:	
obs	Length of data variables (i.e. total number of observations in the file).
profile	Length of profile variables (i.e. total number of profiles in the file, each of which may comprise one or more observed levels/depths, the total of which equals '*obs*').
stringid	Maximum length in characters of the '*plat_id*' identifying a platform.
Profile Variables:	
profile	Unique and monotonically increasing integer ID assigned to each profile.
rowSize	Count of number of observed levels/depths in each profile.
time	Time of profile (days since reference date and time). Set to missing if no date and/or time is available.
lat	Latitude of profile.
lon	Longitude of profile.
plat_id	Character string to identify an individual platform, profiles made by the same platform (e.g. a ship or float) should share the same identifier (though in practice may not always, see Section 2.7.9).
plat_type	Code to identify the type of platform that made a profile (see Section 3.2.3 for code tables).
col	Code to identify the source of the data in HadIOD (see Section 3.2.3 for code tables).
Data Variables:	
depth	Depth of an observation in a profile.
depth_corr	Depth bias correction. Added to depth to help remove biases in reported depth.
temp	Observed sea water temperature.
potemp	Sea water potential temperature (referenced to 0 dbar).
sal	Observed sea water salinity.
[temp/sal]_type_corr	Temperature/Salinity correction for observations made by the given platform type (e.g. ship or XBT), see *plat_type*. Added to observed temperature/salinity to help remove biases in the observed quantity.
[temp/sal]_plat_corr	Temperature/Salinity correction for observations made by an individual platform (e.g. an individual ship or float), see *plat_id*. Added to observed temperature/salinity to help remove biases in the observed quantity.
[temp/sal]_type_corr_unc	The uncertainty in the value of *[temp/sal]_type_corr* (std. dev.). Once the correction is applied, this describes the spread of a probability density function from which the (unknown) residual bias is drawn. The error (residual bias) associated with this uncertainty can be correlated with errors in other observations of the same platform type.
[temp/sal]_plat_corr_unc	The uncertainty in the value of *[temp/sal]_plat_corr* (std. dev.). Once the correction is applied, this describes the spread of a

	probability density function from which the (unknown) residual bias is drawn. The error (residual bias) associated with this uncertainty can be correlated with errors in other observations from the same individual platform.
[temp/sal]_rand_unc	The random measurement error uncertainty (std. dev.) associated with the observed temperature/salinity. The error associated with this uncertainty is not correlated with errors in other observations
[temp/sal]_comb_unc	The uncertainty obtained by combining all other temperature/salinity uncertainty terms in quadrature (std. dev.). Valid when corrections have been applied
[temp/sal]_qc	A quality flag indicating whether an observation passed quality checks. A value of 0=pass, a value of 1=fail.
[temp/sal]_id	HadIOD database identifier for a temperature/salinity observation. This is an integer that uniquely identifies an observation within its year and month. When combined with year and month the identifier is unique within the version of HadIOD used (e.g. HadIOD.1.2.0.0). This can be used to link to data in the additional bias correction files. See FAQs Section 2.7.13.
Global Attributes:	
title	Describes the contents of the file.
featureType	Describes the type of data stored in the file (profile).
Conventions	Describes with which conventions the file complies (e.g. CF1.7).
comment	Further comments on file contents.
institution	Describes the institution that produced the file.
source	Gives more details about the source of the data in the file, e.g. HadIOD version number, which bias correction schemes were used etc.
usage	Gives any rules that must be followed when using the data in this file (e.g. how bias corrections should be applied).
fileVersion	Gives the file version (and associated script). This allows us to trace different versions of the same named file (e.g. if the file was regenerated with changes to its format or content).
licence	Terms and conditions when using the data.
references	Primary references that describe the file contents (websites, publications etc.).
history	This describes when the file was created and should also be used to record any subsequent modifications to the file.

3.2.3   Code tables

Some variables make use of code tables to describe their data. These are given below.

3.2.3.1   'plat_type' variable

This code table is a hybrid of the ICOADS platform type (see ICOADS doc R2.5-imma.pdf p53-54, https://icoads.noaa.gov/) and World Ocean Database probe type (see s_29_probe_type table, https://www.nodc.noaa.gov/OC5/WOD/pr_wod.html). ICOADS platform types of 0-19 are used, as are WOD probe types of 0-16 (but with 20 added to each value, e.g. in WOD 2=XBT, below 2+20=22 for XBT). Where ICOADS and WOD code tables overlap (e.g. both contain a code for drifting buoys) one or other is chosen.

Value	Platform Type
0	US Navy or "deck" log, or unknown
1	merchant ship or foreign military
2	ocean station vessel-off station or station proximity unknown
3	ocean station vessel-on station
4	lightship
5	ship
6	moored buoy
7	drifting buoy
8	ice buoy [note: currently unused]
9	ice station (manned, including ships overwintering in ice)
13	Coastal-Marine Automated Network (C-MAN) (NDBC operated)
14	other coastal/island station
15	fixed ocean platform (plat, rig)
16	tide gauge
20	unknown
21	Mechanical Bathythermograph (MBT)
22	Expendable Bathythermograph (XBT)
23	Digital BathyThermograph (DBT)
24	Conductivity-Temperature-Depth data (CTD)
25	Salinity-Temperature-Depth data (STD)
26	Expendable CTD (XCTD)
27	bottle/rossette/net
29	profiling float
32	towed CTD
33	animal mounted
35	glider
36	microBT

3.2.3.2   'col' variable

This code table gives a complete list of the source IDs currently used by the HadIOD.1.2.0.0 database. Only sources with an Asterix (*) are used in the public files described in this document. Descriptions of the sources of data in HadIOD are given in FAQs Section 2.7.5.

Value	Source of Data Collection
1*	ICOADS.2.5.1
2*	EN.4.2.0
3	MetDB
4*	EN.4.2.1
5	PMEL GTMBA
6	NOAA OSMC
7*	CMEMS
8*	EN.4.2.2
9*	ICOADS.3.0.2

3.3 Additional bias correction files

3.3.1 File format

The bulk of the HadIOD public data is provided in the main files. The bias correction files provide additional sets of platform type bias corrections for some temperature observations that can be used in place of those in the main files to explore the bias correction uncertainty (Section 2.6.4). The files are provided in CF1.7 compliant NetCDF-4 file format.

The ship bias corrections provide 100 alternative sets of *temp_type_corr* data for ship observations. Each set is a different realisation of the same ship correction scheme (see FAQs Section 2.7.7).

The XBT/MBT bias corrections provide 3 alternative sets of *temp_type_corr* and *depth_corr* data for XBT and MBT observations. Each set corresponds to a different bias correction scheme: Levitus et al. [2009], Gouretski [2012] and Cowley et al. [2013] (see FAQs Section 2.7.7). The Levitus et al. [2009] scheme provides corrections for MBTs and XBTs, the others provide corrections for XBTs only. Where MBT corrections are not provided those values will be set to missing. If a user would like to mix and match MBT and XBT corrections from different schemes then MBT and XBT observations can be identified using the *plat_type* variable in the main files (Section 3.2.3.1).

Each set of bias corrections can be linked to the main files by matching the values in *temp_id*. An example of working with the ship correction data is shown in Section 4.

The contents of the additional bias correction files are given in Sections 3.3.2 and 3.3.3. For reference sample file headers can be seen at the end of this document.

3.3.2 Ship bias corrections file variables

NetCDF variable	Description
Dimensions:	
obs	Total number of ship corrections for each set of corrections.
corrs	Number of sets of ship corrections provided.
Data Variables:	
temp_type_corr	Temperature corrections for observations made by ships. Added to observed temperature to help remove biases.
temp_id	HadIOD database identifier for a temperature observation. This is an integer that uniquely identifies an observation within its year and month. It is used to link to data in the main files. See FAQs Section 2.7.13.
Global Attributes:	
title	Describes the contents of the file.
Conventions	Describes with which conventions the file complies (e.g. CF1.7).
comment	Further comments on file contents.
institution	Describes the institution that produced the file.
source	Gives more details about the source of the data in the file, e.g. HadIOD version number, which bias correction schemes were used etc.

usage	Gives any rules that must be followed when using the data in this file (e.g. how bias corrections should be applied).
fileVersion	Gives the file version (and associated script). This allows us to trace different versions of the same named file (e.g. if the file was regenerated with changes to its format or content).
licence	Terms and conditions when using the data.
references	Primary references that describe the file contents (websites, publications etc.).
history	This describes when the file was created and should also be used to record any subsequent modifications to the file.

3.3.3 XBT and MBT bias corrections file variables

NetCDF variable	Description
Dimensions:	
obs	Total number of XBT/MBT corrections for each set of corrections.
Data Variables:	
l09_depth_corr	Levitus et al. [2009] depth bias correction. Added to reported depth to help remove biases. Note, the Levitus et al. [2009] scheme does not provide a depth correction so values are set to missing.
l09_temp_type_corr	Levitus et al. [2009] temperature correction. Added to observed temperature to help remove biases.
g12_depth_corr	Gouretski [2012] depth bias correction. Added to reported depth to help remove biases.
g12_temp_type_corr	Gouretski [2012] temperature correction. Added to observed temperature to help remove biases.
c13_depth_corr	Cowley et al. [2013] depth bias correction. Added to reported depth to help remove biases.
c13_temp_type_corr	Cowley et al. [2013] temperature correction. Added to observed temperature to help remove biases.
temp_id	HadIOD database identifier for a temperature observation. This is an integer that uniquely identifies an observation within its year and month. It is used to link to data in the main files. See FAQs Section 2.7.13.
Global Attributes:	
title	Describes the contents of the file.
Conventions	Describes with which conventions the file complies (e.g. CF1.7).
comment	Further comments on file contents.
institution	Describes the institution that produced the file.
source	Gives more details about the source of the data in the file, e.g. HadIOD version number, which bias correction schemes were used etc.
usage	Gives any rules that must be followed when using the data in this file (e.g. how bias corrections should be applied).

fileVersion	Gives the file version (and associated script). This allows us to trace different versions of the same named file (e.g. if the file was regenerated with changes to its format or content).
licence	Terms and conditions when using the data.
references	Primary references that describe the file contents (websites, publications etc.).
history	This describes when the file was created and should also be used to record any subsequent modifications to the file.

3.4 Tools for working with NetCDF files

A list of software tools that work with NetCDF files is maintained by UCAR: (https://www.unidata.ucar.edu/software/netcdf/software.html).

Some simple tools for viewing and manipulating NetCDF files in Linux include:

- ncdump: provided with the NetCDF library, produces a text rendering of a NetCDF file (Unidata at UCAR).
- Climate Data Operators (CDO)s: a set of command line utilities for performing operations on NetCDF files including concatenation, editing and mathematics: (https://code.mpimet.mpg.de/projects/cdo).
- ncview: a program to produce graphical displays of the contents of NetCDF files. More information can be found here. A more complete list can be found here.

In addition, packages are available in most commonly-used scientific programming languages for reading and working withNetCDF files. For example, in Python there are numerous packages including:

- netCDF4 - this basic package provides functionality to read NetCDF files and extract metadata.
- Iris - developed by the Met Office, iris provides functionality to read, write and process files in a variety of formats including NetCDF.

There are a number of packages in R that can be used to process NetCDF files:

- ncdf4: https://cran.r-project.org/web/packages/ncdf4/index.html
- raster: https://cran.r-project.org/web/packages/raster/index.html
- rcdo: https://github.com/r4ecology/rcdo
- RNetCDF: https://cran.r-project.org/web/packages/RNetCDF/index.html
- CM SAF R tools: https://www.mdpi.com/2220-9964/8/3/109

4   Worked example (using Python3)

Here we demonstrate working with the data using a simple example. We will calculate an average of near-surface temperature observations with bias corrections applied using drifters, ships and Argo floats for a user specified day/time and longitude/latitude box. We will also calculate the uncertainty in the average arising from measurement errors. We calculate the average and uncertainty a few different ways, to illustrate some of the different approaches for working with the uncertainties described in Section 2.6. The results of these calculations are shown at the end of Section 4.7.

In Section 4.1 we read in and unpack a main data file and in Section 4.2 we read in and unpack a supplementary ship corrections file. Input files are presumed downloaded in the run directory. The main data file includes bias corrections and uncertainties but, for ships, only one 'median' set of bias corrections is provided. The supplementary ship corrections file provides multiple sets of ship bias corrections.

In Section 4.3 we subset the data required for calculations (for Argo profiles we extract near-surface observations).

The average and its uncertainty are calculated using a few different approaches:

(i)   Using the main data file only: the three components of uncertainty are calculated using a matrix method (described in Section 2.6 and shown in Section 4.4) and combined to give the overall uncertainty in the average;

(ii)   Using the main data file only: the three components of uncertainty are calculated using a summation method (described in Section 2.6 and shown in Section 4.4) and combined to give the overall uncertainty in the average;

(iii)   Using the main data file and its supplementary file: the average and the platform type error component of uncertainty are calculated by applying different sets of ship corrections in turn to the data (described in Section 2.6.4 and shown in Section 4.5), the platform type error component of uncertainty is combined with the remaining components of uncertainty calculated in approach (ii) to give the overall uncertainty in the average;

(iv)   Using the main data file and its supplementary file: the average and its overall uncertainty are calculated by generating multiple samples of observation error (described in Section 2.6.3.1 and shown in Section 4.6).

For the time and longitude/latitude limits chosen in Section 4.1, all approaches give the same values for the average and its overall uncertainty (to thousandth degree Celsius precision). If averaging over larger time and space scales the platform type error component of uncertainty will become increasingly complex and important and approaches (iii) and (iv) will give the correct value for this term (and thus overall uncertainty).

We treat the errors for ships with generic IDs (see FAQs Section 2.7.9) as uncorrelated when calculating the individual platform error component of uncertainty; this is partly to show the handling of generic IDs and partly for simplicity, but a different method could be used here. For approaches (i) and (ii) we calculate the platform type error component of uncertainty by treating the errors described by the platform type correction uncertainties as entirely correlated for observations of the same type (e.g. ship) which is reasonable over a limited time and space.

For approach (iii) the platform type error component of uncertainty is calculated with and without 'trimming'. Trimming enables a more robust estimate of variance to be made by excluding outlying values which can be caused by extreme sets of ship bias correction parameters. A similar trimming is employed by HadIOD to calculate the platform type correction uncertainties (in the *temp_type_corr_unc* variable) which are used by approaches (i) and (ii). In this example, after trimming, approaches (i) and (ii) give the same value as approach (iii) (to 4 d.p.) for the platform type error component of uncertainty.

In Section 4.7 the results of the calculations are printed to screen.

4.1   Read in and unpack the main data

```
import os

import sys

from netCDF4 import Dataset, num2date, chartostring

import numpy as np

from datetime import datetime


#---Set Input Parameters---#

YEAR=2010

MONTH=6

DAY=10

HOUR_LIMITS=(12,16) # hour limits (24hr)

LAT_LIMITS=(0,30) # grid cell latitude limits

LON_LIMITS=(-170,-130) # grid cell longitude limits

GENERIC_SHIPID=['SHIP','SHIPX','0120','PLAT','RIGG','1','58','7','MASKST','MASKSTID','']


#---Read in Main HadIOD Data---#

#read in main input file

datestring='{0:4d}{1:02d}{2:02d}'.format(YEAR,MONTH,DAY)

filename='hadiod1200_'+datestring+'.nc'
```

```python
if not os.path.exists(filename):

    print(filename+' not found')

    sys.exit()

ncfile=Dataset(filename,'r')

ncfile.set_auto_mask(False) # read data into regular (not masked) numpy arrays


#read in dimensions

nobs=ncfile.dimensions['obs'].size

nprofile=ncfile.dimensions['profile'].size

stridlen=ncfile.dimensions['stringid'].size


#read in profile variables

rowSize=ncfile.variables['rowSize'][:]

profile_provar=ncfile.variables['profile'][:]

time_provar=num2date(ncfile.variables['time'][:],units=ncfile.variables['time'].units,calendar=ncfile.variables['time'].calendar)

lat_provar=ncfile.variables['lat'][:]

lon_provar=ncfile.variables['lon'][:]

plat_id_provar=chartostring(ncfile.variables['plat_id'][:,:])

plat_type_provar=ncfile.variables['plat_type'][:]


#unpack profile variables into empty data arrays using rowSize variable

cumrowSize=np.concatenate([[0],np.cumsum(rowSize)])

profile=np.zeros(nobs,dtype=profile_provar.dtype)

time=np.zeros(nobs,dtype=time_provar.dtype); time[:]=datetime(3000,1,1)

lat=np.zeros(nobs,dtype=lat_provar.dtype)+ncfile.variables['lat']._FillValue
```

```python
lon=np.zeros(nobs,dtype=lon_provar.dtype)+ncfile.variables['lon']._FillValue

plat_id=np.zeros(nobs,dtype=plat_id_provar.dtype); plat_id[:]=' '*stridlen

plat_type=np.zeros(nobs,dtype=plat_type_provar.dtype)+ncfile.variables['plat_type']._FillValue

for indx in range(nprofile):

    indxlo=cumrowSize[indx]

    indxhi=cumrowSize[indx+1]

    profile[indxlo:indxhi]=profile_provar[indx]

    time[indxlo:indxhi]=time_provar[indx]

    lat[indxlo:indxhi]=lat_provar[indx]

    lon[indxlo:indxhi]=lon_provar[indx]

    plat_id[indxlo:indxhi]=plat_id_provar[indx]

    plat_type[indxlo:indxhi]=plat_type_provar[indx]


#read in data variables and fill values
temp=ncfile.variables['temp'][:]

depth=ncfile.variables['depth'][:]

depth_corr=ncfile.variables['depth_corr'][:]

temp_type_corr=ncfile.variables['temp_type_corr'][:]

temp_plat_corr=ncfile.variables['temp_plat_corr'][:]

temp_type_corr_unc=ncfile.variables['temp_type_corr_unc'][:]

temp_plat_corr_unc=ncfile.variables['temp_plat_corr_unc'][:]

temp_rand_unc=ncfile.variables['temp_rand_unc'][:]

temp_qc=ncfile.variables['temp_qc'][:]

temp_id=ncfile.variables['temp_id'][:]
```

```
temp_fill=ncfile.variables['temp']._FillValue

depth_fill=ncfile.variables['depth']._FillValue

depth_corr_fill=ncfile.variables['depth_corr']._FillValue

temp_type_corr_fill=ncfile.variables['temp_type_corr']._FillValue

temp_plat_corr_fill=ncfile.variables['temp_plat_corr']._FillValue

temp_type_corr_unc_fill=ncfile.variables['temp_type_corr_unc']._FillValue

temp_plat_corr_unc_fill=ncfile.variables['temp_plat_corr_unc']._FillValue

temp_rand_unc_fill=ncfile.variables['temp_rand_unc']._FillValue


#set missing corrections and uncertainties to zero

depth_corr[depth_corr==depth_corr_fill]=0

temp_type_corr[temp_type_corr==temp_type_corr_fill]=0

temp_plat_corr[temp_plat_corr==temp_plat_corr_fill]=0

temp_type_corr_unc[temp_type_corr_unc==temp_type_corr_unc_fill]=0

temp_plat_corr_unc[temp_plat_corr_unc==temp_plat_corr_unc_fill]=0

temp_rand_unc[temp_rand_unc==temp_rand_unc_fill]=0


#get corrected temperature and depth (preserve missing values)

temp_adj = temp + temp_type_corr + temp_plat_corr

depth_adj = depth + depth_corr

temp_adj[temp==temp_fill]=temp_fill

depth_adj[depth==depth_fill]=depth_fill
```

4.2   Read in additional ship temperature correction data

```
#---Get Multiple Sets of Ship Corrections---#
```

```python
#read in supplementary ship corrections file
datestring='{0:4d}{1:02d}{2:02d}'.format(YEAR,MONTH,DAY)
filename='hadiod1200_shipcorrs_'+datestring+'.nc'
if not os.path.exists(filename):
    print(filename+' not found')
    sys.exit()
ncfile=Dataset(filename,'r')
ncfile.set_auto_mask(False) # read data into regular (not masked) numpy arrays


#read in dimensions, variables and handle missing values
ncorrs=ncfile.dimensions['corrs'].size
ship_temp_type_corr=ncfile.variables['temp_type_corr'][:,:]
ship_temp_type_corr_fill=ncfile.variables['temp_type_corr']._FillValue
ship_temp_type_corr[ship_temp_type_corr==ship_temp_type_corr_fill]=0
ship_temp_id=ncfile.variables['temp_id'][:]
ship_temp_id_fill=ncfile.variables['temp_id']._FillValue


#create array of corrected temperatures with different ship corrections applied
temp_type_corr_ens=np.tile(temp_type_corr,(ncorrs,1))
for i,sid in enumerate(ship_temp_id):
    if sid==ship_temp_id_fill:
        continue
    else:
        fsid=temp_id==sid
        if np.any(fsid):
```

```
            temp_type_corr_ens[np.tile(fsid,(ncorrs,1))]=ship_temp_type_corr[i,:]

temp_adj_ens=np.tile(temp,(ncorrs,1)) + np.tile(temp_plat_corr,(ncorrs,1)) + temp_type_corr_ens
```

4.3   Subset the data

```
#---Subset Data Needed for Calculations---#

#find drifter, ship and Argo (10m or less) obs passing QC, within lon/lat/time limits

drif_subset = (plat_type==7) & (temp_adj!=temp_fill) & (temp_qc==0) & (lon>=LON_LIMITS[0]) & (lon<=LON_LIMITS[1]) & (lat>=LAT_LIMITS[0]) & \
(lat<=LAT_LIMITS[1]) & (time>=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[0])) & (time<=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[1]))

ship_subset = (plat_type==5) & (temp_adj!=temp_fill) & (temp_qc==0) & (lon>=LON_LIMITS[0]) & (lon<=LON_LIMITS[1]) & (lat>=LAT_LIMITS[0]) & \
(lat<=LAT_LIMITS[1]) & (time>=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[0])) & (time<=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[1]))

argo_subset = (plat_type==29) & (temp_adj!=temp_fill) & (depth_adj!=depth_fill) & (temp_qc==0) & (lon>=LON_LIMITS[0]) & (lon<=LON_LIMITS[1]) & \
(lat>=LAT_LIMITS[0]) & (lat<=LAT_LIMITS[1]) & (time>=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[0])) & \
(time<=datetime(YEAR,MONTH,DAY,HOUR_LIMITS[1])) & (depth_adj<=10)


#find the shallowest Argo ob for each valid profile (obs from the same profile are identified using their profile ID)

argo_subsubset = np.zeros(np.count_nonzero(argo_subset),dtype=bool)

if np.any(argo_subset):

    profile_argo_subset=profile[argo_subset]

    depth_adj_argo_subset=depth_adj[argo_subset]

    for profid in np.unique(profile_argo_subset):

        prof_sub=np.where(profile_argo_subset==profid)

        min_depth=10000; keep_indx=np.nan

        for indx in prof_sub[0]:

            if depth_adj_argo_subset[indx]<min_depth:

                keep_indx=indx

                min_depth=depth_adj_argo_subset[indx]

        argo_subsubset[keep_indx]=True
```

```python
    if np.count_nonzero(argo_subsubset)!=np.unique(profile_argo_subset).shape[0]:

        print('warning, problem with Argo subset')

        sys.exit()


#concatenate ob subsets

if np.any(drif_subset) or np.any(ship_subset) or np.any(argo_subset):

    temp_adj_sub=np.concatenate([temp_adj[drif_subset],temp_adj[ship_subset],temp_adj[argo_subset][argo_subsubset]])

    plat_id_sub=np.concatenate([plat_id[drif_subset],plat_id[ship_subset],plat_id[argo_subset][argo_subsubset]])

    plat_type_sub=np.concatenate([plat_type[drif_subset],plat_type[ship_subset],plat_type[argo_subset][argo_subsubset]])

    temp_rand_unc_sub=np.concatenate([temp_rand_unc[drif_subset],temp_rand_unc[ship_subset],temp_rand_unc[argo_subset][argo_subsubset]])

    temp_type_corr_unc_sub=np.concatenate([temp_type_corr_unc[drif_subset],temp_type_corr_unc[ship_subset],\
temp_type_corr_unc[argo_subset][argo_subsubset]])

    temp_plat_corr_unc_sub=np.concatenate([temp_plat_corr_unc[drif_subset],temp_plat_corr_unc[ship_subset],\
temp_plat_corr_unc[argo_subset][argo_subsubset]])

    #this next step creates a subset array of temperatures with different ship corrections applied

    temp_adj_ens_sub=np.concatenate([np.tile(temp_adj[drif_subset],(ncorrs,1)),\
np.reshape(temp_adj_ens[np.tile(ship_subset,(ncorrs,1))],(ncorrs,-1)),\
np.tile(temp_adj[argo_subset][argo_subsubset],(ncorrs,1))],axis=1)

else:

    print('no obs found for specified limits')

    sys.exit()
```

4.4   Calculate the average observed temperature with uncertainty using main data

```python
#---Calculate Mean of Obs with Uncertainty---#

nsub=temp_adj_sub.shape[0]

weights=np.ones(nsub)/nsub
```

```python
#get mean of obs

obmean=np.mean(temp_adj_sub)


#---calculate uncertainty using the matrix method---


#random error (uncorrelated) component of uncertainty

covariance=np.diag(np.square(temp_rand_unc_sub))

var_rand_matrix=np.matmul(np.matmul(weights,covariance),weights) # python handles transposing 1d array


#individual platform error (simply-correlated) component of uncertainty

#(here we treat obs from ships with a generic ID as uncorrelated)

covariance=np.zeros([nsub,nsub])

for m in range(nsub):

    for n in range(nsub):

        if m==n:

            covariance[m,n]=temp_plat_corr_unc_sub[m]**2

        else:

            if (plat_type_sub[m]==plat_type_sub[n]) and (plat_id_sub[m]==plat_id_sub[n]):

                covariance[m,n]=temp_plat_corr_unc_sub[m]*temp_plat_corr_unc_sub[n]

                if plat_type_sub[m]==5 and plat_id_sub[m].strip() in GENERIC_SHIPID:

                    covariance[m,n]=0

var_plat_corr_matrix=np.matmul(np.matmul(weights,covariance),weights) # python handles transposing 1d array


#platform-type error component of uncertainty

#(we assume here that uncertainties from the same platform-type are correlated)
```

```python
covariance=np.zeros([nsub,nsub])

for m in range(nsub):

    for n in range(nsub):

        if m==n:

            covariance[m,n]=temp_type_corr_unc_sub[m]**2

        else:

            if (plat_type_sub[m]==plat_type_sub[n]):

                covariance[m,n]=temp_type_corr_unc_sub[m]*temp_type_corr_unc_sub[n]

var_type_corr_matrix=np.matmul(np.matmul(weights,covariance),weights) # python handles transposing 1d array


#combine the three uncertainty components in quadrature to get overall uncertainty

unc_comb_matrix = np.sqrt(var_rand_matrix + var_plat_corr_matrix + var_type_corr_matrix)


#---calculate uncertainty using the summation method---


#random error (uncorrelated) component of uncertainty

var_rand_sum=0

for n in range(nsub):

    var_rand_sum += weights[n]**2 * temp_rand_unc_sub[n]**2


#individual platform error (simply-correlated) component of uncertainty

#(here we treat obs from ships with a generic ID as uncorrelated)

var_plat_corr_sum=0

for n in range(nsub):

    var_plat_corr_sum += weights[n]**2 * temp_plat_corr_unc_sub[n]**2
```

```
for m in range(nsub-1):

    for n in range(m+1,nsub):

        if (plat_type_sub[m]==plat_type_sub[n]) and (plat_id_sub[m]==plat_id_sub[n]):

            correlation=1.0

            if plat_type_sub[m]==5 and plat_id_sub[m].strip() in GENERIC_SHIPID:

                correlation=0.0

        else:

            correlation=0.0

        var_plat_corr_sum += 2.0*(weights[m]*weights[n]*temp_plat_corr_unc_sub[m]*temp_plat_corr_unc_sub[n]*correlation)


#platform-type error component of uncertainty

#(we assume here that uncertainties from the same platform-type are correlated)

var_type_corr_sum=0

for n in range(nsub):

    var_type_corr_sum += weights[n]**2 * temp_type_corr_unc_sub[n]**2

for m in range(nsub-1):

    for n in range(m+1,nsub):

        if (plat_type_sub[m]==plat_type_sub[n]):

            correlation=1.0

        else:

            correlation=0.0

        var_type_corr_sum += 2.0*(weights[m]*weights[n]*temp_type_corr_unc_sub[m]*temp_type_corr_unc_sub[n]*correlation)


#combine the three uncertainty components in quadrature to get overall uncertainty

unc_comb_sum = np.sqrt(var_rand_sum + var_plat_corr_sum + var_type_corr_sum)
```

4.5 Calculate the average observed temperature with uncertainty using additional ship correction data

```
#---Calculate Mean and Uncertainty using Multiple Ship Corrections---

#get mean and platform-type error component of uncertainty

obmean_ens_arr=np.mean(temp_adj_ens_sub,axis=1)

obmean_ens=np.mean(obmean_ens_arr)

var_type_corr_ens=np.var(obmean_ens_arr)

var_type_corr_ens_trim=np.var(np.sort(obmean_ens_arr)[1:-1])


#combine the three uncertainty components in quadrature to get overall uncertainty
#(the first two uncertainty components are the same as for other methods)

unc_comb_ens = np.sqrt(var_rand_sum + var_plat_corr_sum + var_type_corr_ens)

```

4.6 Calculate the average observed temperature with uncertainty by generating multiple samples of observation error

```
#---Calculate Mean and Uncertainty using Multiple Observation Samples---

#set sample generation parameters

ns=200 # number of samples for each set of ship corrections

ns_tot = ns * ncorrs # total number of samples generated

obmean_samp_arr = np.zeros(ns_tot,dtype=float) # array to hold sample means


#create dictionary of indices of obs for individual platforms

platob_dic={}

for i,platid in enumerate(plat_id_sub):

    platob_dic.setdefault((platid,plat_type_sub[i]), []).append(i)
```

```
#generate samples and store sample means

ct=0

for nc in range(ncorrs):

    for i in range(ns):

        sample=np.array(temp_adj_ens_sub[nc,:]) # obs with set of ship corrections applied

        #add random errors to obs in sample

        rand_draw = np.random.normal(0.0, 1.0, sample.size)

        sample += rand_draw*temp_rand_unc_sub

        #add individual platform errors to obs in sample

        #(errors for generic ships IDs are treated as uncorrelated)

        for key in platob_dic:

            plat_nobs=len(platob_dic[key])

            if key[1]==5 and key[0].strip() in GENERIC_SHIPID:

                plat_draw = np.random.normal(0.0, 1.0, plat_nobs)

            else:

                plat_draw = np.zeros(plat_nobs) + np.random.normal(0.0, 1.0)

            sample[platob_dic[key]] += plat_draw*temp_plat_corr_unc_sub[platob_dic[key]]

        #get mean of sample

        obmean_samp_arr[ct]=np.mean(sample)

        ct+=1


#calculate mean and std.dev. of sample means

obmean_samp=np.mean(obmean_samp_arr)

unc_samp=np.std(obmean_samp_arr)
```

4.7   Print out the results

```
print('average of observations +/- std.dev. = {0:.3f} +/- {1:.3f} (matrix method)'.format(obmean,unc_comb_matrix))

print('average of observations +/- std.dev. = {0:.3f} +/- {1:.3f} (summation method)'.format(obmean,unc_comb_sum))

print('average of observations +/- std.dev. = {0:.3f} +/- {1:.3f} (multiple ship correction method)'.format(obmean_ens,unc_comb_ens))

print('average of observations +/- std.dev. = {0:.3f} +/- {1:.3f} (multiple observation sample method)'.format(obmean_samp,unc_samp))

print('')

print('average of {0} observations:'.format(nsub))

print('{0} observations from {1} drifters'.format(plat_id[drif_subset].shape[0],np.unique(plat_id[drif_subset]).shape[0]))

print('{0} observations from {1} ship IDs (including generic
IDs)'.format(plat_id[ship_subset].shape[0],np.unique(plat_id[ship_subset]).shape[0]))

print('{0} observations from {1} Argo floats'.format(plat_id[argo_subset][argo_subsubset].shape[0],\
np.unique(plat_id[argo_subset][argo_subsubset]).shape[0]))

print('')

print('---uncertainty components by method (std.dev.)---')

print('platform type uncertainty      : matrix={0:.4f}, summation={1:.4f}, multiple ship correction={2:.4f} (trimmed={3:.4f})'.format(\
np.sqrt(var_type_corr_matrix),np.sqrt(var_type_corr_sum),np.sqrt(var_type_corr_ens),np.sqrt(var_type_corr_ens_trim)))

print('individual platform uncertainty: matrix={0:.4f}, summation={1:.4f}'.format(np.sqrt(var_plat_corr_matrix),np.sqrt(var_plat_corr_sum)))

print('random measurement uncertainty : matrix={0:.4f}, summation={1:.4f}'.format(np.sqrt(var_rand_matrix),np.sqrt(var_rand_sum)))


#---RETURNS---#

average of observations +/- std.dev. = 26.492 +/- 0.042 (matrix method)

average of observations +/- std.dev. = 26.492 +/- 0.042 (summation method)

average of observations +/- std.dev. = 26.492 +/- 0.042 (multiple ship correction method)

average of observations +/- std.dev. = 26.492 +/- 0.042 (multiple observation sample method)


average of 362 observations:
```

```
350 observations from 62 drifters

9 observations from 4 ship IDs (including generic IDs)

3 observations from 3 Argo floats


---uncertainty components by method (std.dev.)---

platform type uncertainty      : matrix=0.0005, summation=0.0005, multiple ship correction=0.0006 (trimmed=0.0005)

individual platform uncertainty: matrix=0.0396, summation=0.0396

random measurement uncertainty : matrix=0.0148, summation=0.0148
```

5 References and Acknowledgments

Abraham, J. P., et al. (2013), A review of global ocean temperature observations: Implications for ocean heat content estimates and climate change, *Rev. Geophys.*, *51*, 450–483, https://doi.org/10.1002/rog.20022.

Atkinson, C. P., N. A. Rayner, J. Roberts-Jones, and R. O. Smith (2013), Assessing the quality of sea surface temperature observations from drifting buoys and ships on a platform-by-platform basis, *JGR-Oceans, 118*, 1-23, https://doi.org/10.1002/jgrc.20257.

Atkinson, C. P., N. A. Rayner, J. J. Kennedy, and S. A. Good (2014), An integrated database of ocean temperature and salinity observations, *J. Geophys. Res. Oceans*, *119*, 7139-7163, https://doi.org/10.1002/2014JC010053 (free access).

Boyer, T. P., et al. (2009), World ocean database 2009, in NOAA Atlas NESDIS 66[DVDs], edited by S. Levitus, 216 pp., U. S. Gov. Print. Off., Washington, D. C.

Boyer, T. P., et al. (2013), World Ocean Database 2013, in NOAA Atlas NESDIS 72, edited by S. Levitus and A. Mishonov (technical editor), 209 pp.

Boyer, T. P., O. K. Baranova, C. Coleman, H. E. Garcia, A. Grodsky, R. A. Locarnini, A. V. Mishonov, T. D. O'Brien, C. R. Paver, J. R. Reagan, D. Seidov, I. V. Smolyar, K. Weathers, and M. M. Zweng (2018), World Ocean Database 2018 *(in preparation)*, check here for latest status: https://www.nodc.noaa.gov/OC5/WOD/pr_wod.html).

Cabanes, C., A. Grouazel, K. von Schuckmann, M. Hamon, V. Turpin, C. Coatanoan, F. Paris, S. Guinehut, C. Boone, N. Ferry, C. de Boyer Montégut, T. Carval, G. Reverdin, S. Pouliquen, and P. Y. Le Traon (2013), The CORA dataset: validation and diagnostics of in-situ ocean temperature and salinity measurements, *Ocean Science*, *9*, 1-18, https://doi.org/10.5194/os-9-1-2013.

Cheng, L., J. Abraham, G. Goni, T. Boyer, S. Wijffels, R. Cowley, V. Gouretski, F. Reseghetti, S. Kizu, S. Dong, F. Bringas, M. Goes, L. Houpert, J. Sprintall, and J. Zhu (2016), XBT Science: Assessment of Instrumental Biases and Errors, *Bull. Amer. Meteor. Soc., 97*, 924–933, https://doi.org/10.1175/BAMS-D-15-00031.1.

Cheng, L. J., J. Zhu, R. Cowley, T. Boyer, and S. Wijffels (2014), Time, probe type and temperature variable bias corrections to historical expendable bathythermograph observations, *J. Atmos. Oceanic Technol.*, *31*, 1793–1825, https://doi.org/10.1175/JTECH-D-13-00197.1.

Cowley, R. W., S. Wijffels, L. J. Cheng, T. P. Boyer, and S. Kizu (2013), Biases in expendable bathythermograph data: A new view based on historical side-by-side comparison, *J. Atmos. and Ocean. Tech.*, *30*, 1195-1225, https://doi.org/10.1175/JTECH-D-12-00127.1.

Carella, G., E. C. Kent, and D. I. Berry (2017), A probabilistic approach to ship voyage reconstruction in ICOADS, *Int. J. Climatol., 37,* 2233-2247, https://doi.org/10.1002/joc.4492.

Freeman, E., S. D. Woodruff, S. J. Worley, S. J. Lubker, E. C. Kent, W. E. Angel, D. I . Berry, P. Brohan, R. Eastman, L. Gates, W. Gloeden, Z. Ji, J. Lawrimore, N. A. Rayner, G. Rosenhagen, and S. R. Smith (2017), ICOADS Release 3.0: A major update to the historical marine climate record, *Int. J. Climatol.*, *37*, 2211-2237, https://doi.org/10.1002/joc.4775.

Good, S. A., M. J. Martin, and N. A. Rayner (2013), EN4: Quality controlled ocean temperature and salinity profiles and monthly objective analyses with uncertainty estimates, *J. Geophys. Res. Oceans*, *118*, 6704–6716, https://doi.org/10.1002/2013JC009067.

Gouretski, V., and F. Reseghetti (2010), On depth and temperature biases in bathythermographic data: Development of a new correction scheme based on analysis of global ocean databases, *Deep Sea Res., Part I*, *57,* 812-833, https://doi.org/10.1016/j.dsr.2010.03.011.

Gouretski, V. (2012), Using GBECO digital bathymetry to infer depth biases in the XBT data, *Deep-Sea Res., Part I*, *62*, 40-52, https://doi.org/10.1016/j.dsr.2011.12.012.

Hamon, M., G. Reverdin, and P.-Y. Le Traon (2012), Empirical correction of XBT data, *J. Atmos. Oceanic Technol.*, *29*, 960–973, https://doi.org/10.1175/JTECH-D-11-00129.1.

Joint Committee for Guides in Metrology (JCGM) (2008), Evaluation of measurement data - Guide to the expression of uncertainty in measurement, Report. (Available at: http://www.bipm.org/en/publications/guides/gum.html*.)*

Kennedy J. J., N. A. Rayner, R. O. Smith, M. Saunby, and D. E. Parker (2011a), Reassessing biases and other uncertainties in sea-surface temperature observations since 1850 part 1: measurement and sampling errors, *J. Geophys. Res.*, *116*, D14103, https://doi.org/10.1029/2010JD015218.

Kennedy J. J., N. A. Rayner, R. O. Smith, M. Saunby, and D. E. Parker (2011b), Reassessing biases and other uncertainties in sea-surface temperature observations since 1850 part 2: biases and homogenisation, *J. Geophys. Res.*, *116*, D14104, https://doi.org/10.1029/2010JD015220.

Kent, E. C*., and* A. Kaplan (2006)*,* Toward estimating climatic trends in SST. Part III: Systematic biases*, J. Atmos. Oceanic Technol., 23,* 487–500*,* https://doi.org/10.1175/JTECH1845.1.

Kent, E. C., J. J. Kennedy, T. M. Smith, S. Hirahara, B. Huang, A. Kaplan, D. E. Parker, C. P. Atkinson, D. I. Berry, G. Carella, Y. Fukuda, M. Ishii, P. D. Jones, F. Lindgren, C. J. Merchant, S. Morak-Bozzo, N. A. Rayner, V. Venema, S. Yasui, and H. Zhang (2017), A call for new approaches to quantifying biases in observations of sea surface temperature, *Bull. Amer. Meteor. Soc.*, *98*, 1601-1616, https://doi.org/10.1175/BAMS-D-15-00251.1.

Levitus, S., J. I. Antonov, T. P. Boyer, R. A. Locamini, H. E. Garcia, and A. V. Mishonov (2009), Global ocean heat content 1955-2008 in light of recently revealed instrumentation problems, *Geophys. Res. Lett.*, *36*, L07608, https://doi.org/10.1029/2008GL037155.

Palmer, M. D., T. Boyer, R. Cowley, S. Kizu, F. Reseghetti, T. Suzuki, and A. Thresher (2018), An algorithm for classifying unknown expendable bathythermograph (XBT) instruments based on existing metadata, *J. Atmos. Oceanic Technol., 35(3)*, 429-440, https://doi.org/10.1175/JTECH-D-17-0129.1.

Woodruff, S. D., S. J. Worley, S. J. Lubker, Z. Ji, J. E. Freeman, D. I. Berry, P. Brohan, E. C. Kent, R. W. Reynolds, S. R. Smith, and C. Wilkinson (2011), ICOADS Release 2.5: Extensions and enhancements to the surface marine meteorological archive, *Int. J. Climatol.*, *31*, 951-967, https://doi.org/10.1002/joc.2103.

Table 2. Bias corrections, uncertainty estimates and depths (for surface obs) allocated to temperature (and potential temperature) observations in HadIOD.1.2.0.0 for various platform-types in the ICOADS 2.5.1/ICOADS.3.0.2, EN.4.2.0/EN.4.2.1/EN.4.2.2 and CMEMS datasets. For further explanation of column headers see Section 2.4. Type_corr values for ships, MBTs and XBTs are for corrections provided in the main HadIOD files (see Section 2.6 and FAQs Section 2.7.7). Bias corrections are added to the observed value to reduce systematic errors. ERI=Engine Room Intake, MBT=Mechanical BathyThermograph, XBT=Expendable BathyThermograph, DBT=Digital BathyThermograph, CTD=Conductivity Temperature Depth data, STD=Salinity Temperature Depth data, XCTD=Expendable Conductivity Temperature Depth data, MicroBT=MicroBathyThermograph.

Source	Platform-type	Type_corr (°C)	Type_corr_unc (std.dev. °C)	Plat_corr (°C)	Plat_corr_unc (std.dev.°C)	Rand_unc (std.dev. °C)	Depth (metres)	Depth_corr (metres)
ICOADS	Ship-Bucket[1]	-0.60 to 1.80	0.01 to 0.39[2]	0	0.71	0.74	0.1	0
ICOADS	Ship-ERI[1]	-0.2[3]	~ 0.02[3]	0	0.71	0.74	5	-4.9
ICOADS	Drifting Buoy	0	0	0	0.29	0.26	0.2	0
CMEMS	Drifting Buoy	0	0	0	0.29	0.26	0.2	0
ICOADS	Moored Buoy[4]	-	-	0	0.20	0.30	1	-
ICOADS	Other Surface[5]	-	-	-	-	1	1	-
ICOADS	Sub-Surface[6]	-	-	-	-	_[6]	_[6]	-
EN4	MBT	-0.17 to 0.06	-	-	-	0.3	EN4	~ -26 to 41
EN4	XBT	-0.21 to 0.11	-	-	-	0.15	EN4	~ -60 to 18
EN4	DBT	-	-	-	-	0.05	EN4	0
EN4	CTD	0	-	0	-	0.002	EN4	0
EN4	STD	-	-	-	-	0.002	EN4	0
EN4	XCTD	-	-	-	-	0.02	EN4	-
EN4	Bottle/Rosette/Net	-	-	-	-	0.02	EN4	0
EN4	Profiling Float	0	-	-	-	0.002	EN4	0
EN4	Moored Buoy	0[7]	-	-	-	0.02[7]	EN4	0
EN4	Profiling Drifting Buoy	-	-	-	-	0.01	EN4	0
EN4	Towed CTD	-	-	-	-	0.002	EN4	0
EN4	Animal-Mounted	-	-	-	-	0.005[8]	EN4	0
EN4	Glider	-	-	-	-	0.002	EN4	0
EN4	MicroBT	-	-	-	-	0.002[9]	EN4	0
EN4	Unknown	-	-	-	-	-	EN4	-

[1]Values apply to observations positively identified as buckets (pre-1941) or ERI. For many ships in HadIOD, SST measurement method is unknown or uncertain and is assigned probabilistically, and in these cases type_corr and depth fields are a combination of bucket and ERI properties.

[2]For buckets (pre-1941), as for any ship observation whose type_corr relies on probabilistic metadata, a standard deviation does not adequately represent the uncertainty in the correction (see Atkinson et al. [2014] for further discussion). To better explore the correction uncertainty, the use of multiple sets of bias corrections is encouraged (see Section 2.6).

[3]Except North Atlantic 1970-1994, where Kent and Kaplan [2006] values are used: type_corr = -0.18 to 0.13 °C and type_corr_unc ≈ 0.05 °C.

[4]GTMBA receives values shown for EN4 (excluding depth fields), though these are flagged as duplicates as EN4 is the preferred source of GTMBA data.

[5]C-MAN / Fixed Ocean Platforms / Coastal and Island Stations / Ice Stations.

[6]These observations are from WOD05 (Deck ID=780), in the HadIOD.1.2.0.0 database no values have been provided for these observations on the basis they are flagged as non-preferred duplicates of EN4 WOD13 observations (Section 2.7.11). However, this duplicate flagging only applies from 1900-onwards (when EN4 begins) and prior to 1900 a small number of bottle data (platform type=27) are retained (roughly 0.1% of total observations prior to 1900). In the HadIOD public files we assign these data the following: rand_unc=0.02 °C, depth=1.0 m.

[7]If *not* GTMBA or WOD13, rand_unc is set to 0.36 (the combined uncertainty for surface moored buoys) and type_corr is set to missing.

[8]0.5 °C for MK3 TTDR (Time Temperature Depth Recorder) sensor, 0.1 °C for MK9/MK10.

[9]0.05 °C for instruments manufactured by RBR Ltd.

Table 3. Bias corrections, uncertainty estimates and depths allocated to salinity observations in HadIOD.1.2.0.0 for platform-types in the EN.4.2.0/EN.4.2.1/EN.4.2.2 datasets. For further explanation of column headers see Section 2.4. CTD=Conductivity Temperature Depth data, STD=Salinity Temperature Depth data, XCTD=Expendable Conductivity Temperature Depth data

Source	Platform-type	Type_corr (psu)	Type_corr_unc (std.dev. psu)	Plat_corr (psu)	Plat_corr_unc (std.dev. psu)	Rand_unc (std.dev. psu)	Depth (metres)	Depth_corr (metres)
EN4	CTD	-	-	-	-	0.005	EN4	0
EN4	STD	-	-	-	-	0.005	EN4	0
EN4	XCTD	-	-	-	-	0.05	EN4	-
EN4	Bottle/Rosette/Net	-	-	-	-	0.05	EN4	0
EN4	Profiling Float	-	-	-	-	0.005	EN4	0
EN4	Moored Buoy	-	-	-	-	0.02	EN4	0
EN4	Profiling Drifting Buoy	-	-	-	-	0.01	EN4	0
EN4	Towed CTD	-	-	-	-	0.005	EN4	0
EN4	Animal-Mounted	-	-	-	-	0.005	EN4	0
EN4	Glider	-	-	-	-	0.005	EN4	0
EN4	Unknown	-	-	-	-	-	EN4	-

Sample HadIOD main file header:

```
netcdf hadiod1200_20190610 {
dimensions:
        obs = 211115 ;
        profile = 62840 ;
        stringid = 9 ;
variables:
        int profile(profile) ;
                profile:long_name = "profile id" ;
                profile:units = "1" ;
                profile:cf_role = "profile_id" ;
        int rowSize(profile) ;
                rowSize:long_name = "number of observations for this profile" ;
                rowSize:units = "1" ;
                rowSize:sample_dimension = "obs" ;
        double time(profile) ;
                time:_FillValue = 99999. ;
                time:standard_name = "time" ;
                time:long_name = "time" ;
                time:units = "days since 1850-01-01 00:00:00" ;
                time:calendar = "standard" ;
        double lat(profile) ;
                lat:_FillValue = 99999. ;
                lat:standard_name = "latitude" ;
                lat:long_name = "latitude" ;
                lat:units = "degrees_north" ;
                lat:valid_min = -90. ;
                lat:valid_max = 90. ;
        double lon(profile) ;
                lon:_FillValue = 99999. ;
                lon:standard_name = "longitude" ;
                lon:long_name = "longitude" ;
                lon:units = "degrees_east" ;
                lon:valid_min = -180. ;
                lon:valid_max = 180. ;
        char plat_id(profile, stringid) ;
                plat_id:long_name = "platform identifier" ;
                plat_id:comment = "code to identify an individual platform, observations made by
                                   the same platform should share the same identifier" ;
        byte plat_type(profile) ;
                plat_type:_FillValue = 99b ;
                plat_type:long_name = "platform type" ;
                plat_type:units = "1" ;
                plat_type:comment = "refer to product user guide for code tables
```

```
                                          (see global attribute: references)" ;
        byte col(profile) ;
                col:long_name = "HadIOD data collection code" ;
                col:units = "1" ;
                col:comment = "refer to product user guide for code tables
                                (see global attribute: references)" ;
        float depth(obs) ;
                depth:_FillValue = 99999.f ;
                depth:standard_name = "depth" ;
                depth:long_name = "depth below the surface" ;
                depth:units = "metre" ;
                depth:positive = "down" ;
                depth:axis = "Z" ;
                depth:ancillary_variables = "depth_corr" ;
        float depth_corr(obs) ;
                depth_corr:_FillValue = 99999.f ;
                depth_corr:long_name = "depth correction" ;
                depth_corr:units = "metre" ;
                depth_corr:comment = "add to depth (i.e. depth + depth_corr) to reduce systematic errors (biases)" ;
                depth_corr:usage = "please read global attribute: usage" ;
        float temp(obs) ;
                temp:_FillValue = 99999.f ;
                temp:standard_name = "sea_water_temperature" ;
                temp:long_name = "sea_water_temperature" ;
                temp:units = "Celsius" ;
                temp:coordinates = "time depth lat lon" ;
                temp:ancillary_variables = "temp_type_corr temp_plat_corr temp_type_corr_unc temp_plat_corr_unc
                                            temp_rand_unc temp_comb_unc temp_qc temp_id" ;
        float potemp(obs) ;
                potemp:_FillValue = 99999.f ;
                potemp:standard_name = "sea_water_potential_temperature" ;
                potemp:long_name = "sea_water_potential_temperature" ;
                potemp:units = "Celsius" ;
                potemp:coordinates = "time depth lat lon" ;
                potemp:ancillary_variables = "temp_type_corr temp_plat_corr temp_type_corr_unc temp_plat_corr_unc
                                              temp_rand_unc temp_comb_unc temp_qc temp_id" ;
        float temp_type_corr(obs) ;
                temp_type_corr:_FillValue = 99999.f ;
                temp_type_corr:long_name = "temperature correction (for given platform type)" ;
                temp_type_corr:units = "Celsius" ;
                temp_type_corr:ancillary_variables = "temp_type_corr_unc" ;
                temp_type_corr:comment = "add to temp (i.e. temp + temp_type_corr) to reduce systematic errors (biases)" ;
                temp_type_corr:usage = "please read global attribute: usage" ;
        float temp_plat_corr(obs) ;
                temp_plat_corr:_FillValue = 99999.f ;
                temp_plat_corr:long_name = "temperature correction (for individual platform)" ;
```

```
        temp_plat_corr:units = "Celsius" ;
        temp_plat_corr:ancillary_variables = "temp_plat_corr_unc" ;
        temp_plat_corr:comment = "add to temp (i.e. temp + temp_plat_corr) to reduce systematic errors (biases)" ;
        temp_plat_corr:usage = "please read global attribute: usage" ;
float temp_type_corr_unc(obs) ;
        temp_type_corr_unc:_FillValue = 99999.f ;
        temp_type_corr_unc:long_name = "temperature correction (for given platform type) uncertainty" ;
        temp_type_corr_unc:units = "Celsius" ;
        temp_type_corr_unc:comment = "uncertainty in the value of temp_type_corr (standard deviation),
                                      the error associated with this uncertainty can be correlated with
                                      errors in other observations of the same platform type" ;
        temp_type_corr_unc:usage = "please read global attribute: usage" ;
float temp_plat_corr_unc(obs) ;
        temp_plat_corr_unc:_FillValue = 99999.f ;
        temp_plat_corr_unc:long_name = "temperature correction (for individual platform) uncertainty" ;
        temp_plat_corr_unc:units = "Celsius" ;
        temp_plat_corr_unc:comment = "uncertainty in the value of temp_plat_corr (standard deviation),
                                      the error associated with this uncertainty can be correlated with
                                      errors in other observations from the same individual platform" ;
        temp_plat_corr_unc:usage = "please read global attribute: usage" ;
float temp_rand_unc(obs) ;
        temp_rand_unc:_FillValue = 99999.f ;
        temp_rand_unc:long_name = "temperature random measurement error uncertainty" ;
        temp_rand_unc:units = "Celsius" ;
        temp_rand_unc:comment = "random measurement error uncertainty associated with temp (standard deviation),
                                   the error associated with this uncertainty is not correlated with errors in other observations" ;
        temp_rand_unc:usage = "please read global attribute: usage" ;
float temp_comb_unc(obs) ;
        temp_comb_unc:_FillValue = 99999.f ;
        temp_comb_unc:long_name = "temperature combined uncertainty" ;
        temp_comb_unc:units = "Celsius" ;
        temp_comb_unc:comment = "uncertainty obtained by combining all uncertainty terms in quadrature (standard deviation)" ;
        temp_comb_unc:usage = "valid if temp_type_corr and temp_plat_corr have been applied,
                               please read global attribute: usage" ;
byte temp_qc(obs) ;
        temp_qc:_FillValue = 99b ;
        temp_qc:standard_name = "status_flag" ;
        temp_qc:long_name = "temperature observation quality" ;
        temp_qc:valid_range = 0b, 1b ;
        temp_qc:flag_values = 0b, 1b ;
        temp_qc:flag_meanings = "passed_qc_checks failed_qc_checks" ;
int temp_id(obs) ;
        temp_id:_FillValue = 0 ;
        temp_id:long_name = "HadIOD database identifier" ;
        temp_id:units = "1" ;
        temp_id:comment = "uniquely identifies an observation within its year and month" ;
```

```
        temp_id:usage = "when combined with year and month the observation identifier is unique
                          within the version of HadIOD used as data source (see global attribute: source)" ;
    float sal(obs) ;
        sal:_FillValue = 99999.f ;
        sal:standard_name = "sea_water_salinity" ;
        sal:long_name = "sea_water_salinity" ;
        sal:units = "1e-3" ;
        sal:coordinates = "time depth lat lon" ;
        sal:ancillary_variables = "sal_type_corr sal_plat_corr sal_type_corr_unc sal_plat_corr_unc
                                    sal_rand_unc sal_comb_unc sal_qc sal_id" ;
    float sal_type_corr(obs) ;
        sal_type_corr:_FillValue = 99999.f ;
        sal_type_corr:long_name = "salinity correction (for given platform type)" ;
        sal_type_corr:units = "1e-3" ;
        sal_type_corr:ancillary_variables = "sal_type_corr_unc" ;
        sal_type_corr:comment = "add to sal (i.e. sal + sal_type_corr) to reduce systematic errors (biases)" ;
        sal_type_corr:usage = "please read global attribute: usage" ;
    float sal_plat_corr(obs) ;
        sal_plat_corr:_FillValue = 99999.f ;
        sal_plat_corr:long_name = "salinity correction (for individual platform)" ;
        sal_plat_corr:units = "1e-3" ;
        sal_plat_corr:ancillary_variables = "sal_plat_corr_unc" ;
        sal_plat_corr:comment = "add to sal (i.e. sal + sal_plat_corr) to reduce systematic errors (biases)" ;
        sal_plat_corr:usage = "please read global attribute: usage" ;
    float sal_type_corr_unc(obs) ;
        sal_type_corr_unc:_FillValue = 99999.f ;
        sal_type_corr_unc:long_name = "salinity correction (for given platform type) uncertainty" ;
        sal_type_corr_unc:units = "1e-3" ;
        sal_type_corr_unc:comment = "uncertainty in the value of sal_type_corr (standard deviation),
                                      the error associated with this uncertainty can be correlated with
                                      errors in other observations of the same platform type" ;
        sal_type_corr_unc:usage = "please read global attribute: usage" ;
    float sal_plat_corr_unc(obs) ;
        sal_plat_corr_unc:_FillValue = 99999.f ;
        sal_plat_corr_unc:long_name = "salinity correction (for individual platform) uncertainty" ;
        sal_plat_corr_unc:units = "1e-3" ;
        sal_plat_corr_unc:comment = "uncertainty in the value of sal_plat_corr (standard deviation),
                                      the error associated with this uncertainty can be correlated with
                                      errors in other observations from the same individual platform" ;
        sal_plat_corr_unc:usage = "please read global attribute: usage" ;
    float sal_rand_unc(obs) ;
        sal_rand_unc:_FillValue = 99999.f ;
        sal_rand_unc:long_name = "salinity random measurement error uncertainty" ;
        sal_rand_unc:units = "1e-3" ;
        sal_rand_unc:comment = "random measurement error uncertainty associated with sal (standard deviation),
                                  the error associated with this uncertainty is not correlated with errors in other observations" ;
```

```
                sal_rand_unc:usage = "please read global attribute: usage" ;
        float sal_comb_unc(obs) ;
                sal_comb_unc:_FillValue = 99999.f ;
                sal_comb_unc:long_name = "salinity combined uncertainty" ;
                sal_comb_unc:units = "1e-3" ;
                sal_comb_unc:comment = "uncertainty obtained by combining all uncertainty terms in quadrature (standard deviation)" ;
                sal_comb_unc:usage = "valid if sal_type_corr and sal_plat_corr have been applied,
                                      please read global attribute: usage" ;
        byte sal_qc(obs) ;
                sal_qc:_FillValue = 99b ;
                sal_qc:standard_name = "status_flag" ;
                sal_qc:long_name = "salinity observation quality" ;
                sal_qc:valid_range = 0b, 1b ;
                sal_qc:flag_values = 0b, 1b ;
                sal_qc:flag_meanings = "passed_qc_checks failed_qc_checks" ;
        int sal_id(obs) ;
                sal_id:_FillValue = 0 ;
                sal_id:long_name = "HadIOD database identifier" ;
                sal_id:units = "1" ;
                sal_id:comment = "uniquely identifies an observation within its year and month" ;
                sal_id:usage = "when combined with year and month the observation identifier is unique
                                within the version of HadIOD used as data source (see global attribute: source)" ;

// global attributes:
                :title = "ocean temperature and salinity in situ observations for 10 June 2019
                          taken from the Met Office Hadley Centre HadIOD database" ;
                :featureType = "profile" ;
                :Conventions = "CF-1.7" ;
                :comment = "CF-1.7 contiguous ragged array data representation, compliant with one exception:
                            point observations are stored as single-depth profiles.
                            CF-1.7 does not permit point and profile featureTypes in the same file" ;
                :institution = "Met Office Hadley Centre (UK)" ;
                :source = "HadIOD version 1.2.0.0:
                           Gouretski and Reseghetti (2010) MBT/XBT correction scheme and ship Mean Ensemble Member correction scheme" ;
                :usage = "file-specific guidance:
                          temp_type_corr and depth_corr must be applied together (and not independently).
                          For help working with correction and uncertainty variables read the product user guide
                          (see global attribute: references)." ;
                :fileVersion = "1.0 (create_public_netcdf_file_daily.py r374)" ;
                :licence = "HadIOD is distributed under the Non Commercial Government Licence:
                            http://www.nationalarchives.gov.uk/doc/non-commercial-government-licence/version/2/.
                            The data are available for non-commercial use with attribution to the data providers,
                            please see the product website (see global attribute: references) for terms and conditions" ;
                :references = "website and product user guide: https://www.metoffice.gov.uk/hadobs/hadiod/
                               Atkinson, C.P., N.A. Rayner, J.J. Kennedy, S.A. Good, 2014:
                               An Integrated Database of Ocean Temperature and Salinity Observations
```

```
                              JGR-Oceans, 119, 7139-7163, doi:10.1002/2014JC010053" ;
            :history = "created Fri Jun 19 03:43:49 2020" ;
}
```

Sample HadIOD additional ship corrections file header:

```
netcdf hadiod1200_shipcorrs_20190610 {
dimensions:
        obs = 3610 ;
        corrs = 100 ;
variables:
        float temp_type_corr(obs, corrs) ;
                temp_type_corr:_FillValue = 99999.f ;
                temp_type_corr:long_name = "temperature correction ensemble (for ships)" ;
                temp_type_corr:units = "Celsius" ;
                temp_type_corr:comment = "add to temp (i.e. temp + temp_type_corr) to reduce systematic errors (biases)" ;
                temp_type_corr:usage = "please read global attribute: usage" ;
        int temp_id(obs) ;
                temp_id:_FillValue = 0 ;
                temp_id:long_name = "HadIOD database identifier" ;
                temp_id:units = "1" ;
                temp_id:comment = "uniquely identifies an observation within its year and month" ;
                temp_id:usage = "when combined with year and month the observation identifier is unique
                                    within the version of HadIOD used as data source (see global attribute: source)" ;

// global attributes:
                :title = "ship temperature bias correction ensemble for 10 June 2019
                        taken from the Met Office Hadley Centre HadIOD database" ;
                :Conventions = "CF-1.7" ;
                :comment = "this is a supplementary file intended for use with its main data file" ;
                :institution = "Met Office Hadley Centre (UK)" ;
                :source = "HadIOD version 1.2.0.0:
                         ship bias correction ensemble (100 ensemble members)" ;
                :usage = "file-specific guidance:
                        temp_type_corr and depth_corr (depths available in main file) must be applied together (and not independently).
                        For help working with correction and uncertainty variables read the product user guide
                        (see global attribute: references)." ;
                :fileVersion = "1.0 (create_public_shipcorr_netcdf_file_daily.py r374)" ;
                :licence = "HadIOD is distributed under the Non Commercial Government Licence:
                          http://www.nationalarchives.gov.uk/doc/non-commercial-government-licence/version/2/.
                          The data are available for non-commercial use with attribution to the data providers,
                          please see the product website (see global attribute: references) for terms and conditions" ;
                :references = "website and product user guide: https://www.metoffice.gov.uk/hadobs/hadiod/
                            Atkinson, C.P., N.A. Rayner, J.J. Kennedy, S.A. Good, 2014:
                            An Integrated Database of Ocean Temperature and Salinity Observations
                            JGR-Oceans, 119, 7139-7163, doi:10.1002/2014JC010053" ;
                :history = "created Fri Jun 19 14:29:54 2020" ;
}
```

Sample HadIOD additional XBT/MBT corrections file header:

```
netcdf hadiod1200_btcorrs_20190610 {
dimensions:
        obs = 14510 ;
variables:
        float l09_depth_corr(obs) ;
                l09_depth_corr:_FillValue = 99999.f ;
                l09_depth_corr:long_name = "depth correction (Levitus 2009)" ;
                l09_depth_corr:units = "metre" ;
                l09_depth_corr:comment = "add to depth (i.e. depth + l09_depth_corr) to reduce systematic errors (biases)" ;
                l09_depth_corr:usage = "please read global attribute: usage" ;
        float l09_temp_type_corr(obs) ;
                l09_temp_type_corr:_FillValue = 99999.f ;
                l09_temp_type_corr:long_name = "temperature correction (Levitus 2009)" ;
                l09_temp_type_corr:units = "Celsius" ;
                l09_temp_type_corr:comment = "add to temp (i.e. temp + l09_temp_type_corr) to reduce systematic errors (biases)" ;
                l09_temp_type_corr:usage = "please read global attribute: usage" ;
        float g12_depth_corr(obs) ;
                g12_depth_corr:_FillValue = 99999.f ;
                g12_depth_corr:long_name = "depth correction (Gouretski 2012)" ;
                g12_depth_corr:units = "metre" ;
                g12_depth_corr:comment = "add to depth (i.e. depth + g12_depth_corr) to reduce systematic errors (biases)" ;
                g12_depth_corr:usage = "please read global attribute: usage" ;
        float g12_temp_type_corr(obs) ;
                g12_temp_type_corr:_FillValue = 99999.f ;
                g12_temp_type_corr:long_name = "temperature correction (Gouretski 2012)" ;
                g12_temp_type_corr:units = "Celsius" ;
                g12_temp_type_corr:comment = "add to temp (i.e. temp + g12_temp_type_corr) to reduce systematic errors (biases)" ;
                g12_temp_type_corr:usage = "please read global attribute: usage" ;
        float c13_depth_corr(obs) ;
                c13_depth_corr:_FillValue = 99999.f ;
                c13_depth_corr:long_name = "depth correction (Cowley 2013)" ;
                c13_depth_corr:units = "metre" ;
                c13_depth_corr:comment = "add to depth (i.e. depth + c13_depth_corr) to reduce systematic errors (biases)" ;
                c13_depth_corr:usage = "please read global attribute: usage" ;
        float c13_temp_type_corr(obs) ;
                c13_temp_type_corr:_FillValue = 99999.f ;
                c13_temp_type_corr:long_name = "temperature correction (Cowley 2013)" ;
                c13_temp_type_corr:units = "Celsius" ;
                c13_temp_type_corr:comment = "add to temp (i.e. temp + c13_temp_type_corr) to reduce systematic errors (biases)" ;
                c13_temp_type_corr:usage = "please read global attribute: usage" ;
        int temp_id(obs) ;
                temp_id:_FillValue = 0 ;
                temp_id:long_name = "HadIOD database identifier" ;
```

```
                temp_id:units = "1" ;
                temp_id:comment = "uniquely identifies an observation within its year and month" ;
                temp_id:usage = "when combined with year and month the observation identifier is unique
                                    within the version of HadIOD used as data source (see global attribute: source)" ;

// global attributes:
                :title = "XBT/MBT temperature and depth corrections for 10 June 2019
                        taken from the Met Office Hadley Centre HadIOD database" ;
                :Conventions = "CF-1.7" ;
                :comment = "this is a supplementary file intended for use with its main data file" ;
                :institution = "Met Office Hadley Centre (UK)" ;
                :source = "HadIOD version 1.2.0.0:
                         Levitus et al. (2009) XBT and MBT; Gouretski (2012) XBT; Cowley et al. (2013) XBT" ;
                :usage = "file-specific guidance:
                        temp_type_corr and depth_corr must be applied together (and not independently).
                        For help working with correction and uncertainty variables read the product user guide
                        (see global attribute: references)." ;
                :fileVersion = "1.0 (create_public_xbtmbtcorr_netcdf_file_daily.py r374)" ;
                :licence = "HadIOD is distributed under the Non Commercial Government Licence:
                            http://www.nationalarchives.gov.uk/doc/non-commercial-government-licence/version/2/.
                            The data are available for non-commercial use with attribution to the data providers,
                            please see the product website (see global attribute: references) for terms and conditions" ;
                :references = "website and product user guide: https://www.metoffice.gov.uk/hadobs/hadiod/
                            Atkinson, C.P., N.A. Rayner, J.J. Kennedy, S.A. Good, 2014:
                            An Integrated Database of Ocean Temperature and Salinity Observations
                            JGR-Oceans, 119, 7139-7163, doi:10.1002/2014JC010053" ;
                :history = "created Wed Jul  1 02:35:45 2020" ;
}
```