

UNIFIED MODEL DOCUMENTATION PAPER NO 72

THE SST ANALYSIS SYSTEM

C P Jones, D L Harrison and E B Little

Version 1.0

14 January 1994

(Corresponds to v3.3 of UM)

Numerical Weather Prediction
Meteorological Office
London Road
BRACKNELL
Berkshire
RG12 2SZ
United Kingdom

(c) Crown Copyright 1996

This document has not been published. Permission to quote from it must be obtained from the Head of Numerical Modelling at the above address.

1. INTRODUCTION

This paper describes the programs that are involved in the operational global SST analysis system. The equations involved in the scheme are also included but for a more detailed description of the analysis scheme see Jones (1991) and for a detailed description of the satellite bias correction scheme see Jones (1993).

The SST system composes of the following stages; data extraction (programs STEXT and RALUAT), creation of data records (program QXSSTDATR), quality control (QXSSTQC.PGM), creation of superobs for DRIFTERS and fixed buoys (program QXSST.SUPEROB), creation of ACOBS file (program QXSSTAC.PGM), analysing SST (program QXSST.ANALYSIS), creation of records for the SST observation processing database (program QXSSTOPD), combining the SST analysis with model T_s values over land and seaice (program QXSSTTSTAR), converting data to HDS pp format (QXFIELDCOS), adding the data to a HDS fieldsfile (program FFLLIBM), plotting the SST analysis on Calcomp (program SSTPLOT) and creating ATSR records for merging into the MDB (program RALMRG).

Program QXSSTOPD is not described in this paper, see Jones et al (1993), nor are programs FIELDCOS or FFLLIBM as these are general programs and not unique to the SST system, see UMDP Y8 for documentation on these.

In this paper, each program is described in a different section. Also included is a summary of all the changes made to each program since the program has been operational.

2. SSTEXT

SSTEXT extracts observations from the SDB or MDB. Observations from the SDB consist of ships, fixed bouys, bathythermospheres, TRACKOBS and SATOBS. Drifting bouys, known as DRIFTERS, are extracted from the MDB.

The source code consists of the following members on CFOBS.BLSSTLIB.SSTEXT; SSTEXT (main program), READHK, WRTMDB, WRTSDB, EXDRIFT, EXTRACK, EXSATOB, EXBATHY, EXSHIP and DIFDAY. The job on CFOBS.BLSSTLIB.SSTEXT(@CNTLTOT) creates a single source member on CFOBS.BLSSTLIB.SSTEXT(TOTAL).

CFOBS.BLSSTLIB.SSTEXT(@CNTLLOD) creates a load module of the program. The routines SDB and MDB on SDB.LOADLIB need to be included. A copy of this job can also be found on COP.CORE.BUILD(SSTEXT).

Index to program:

```
PROGRAM SSTEXT
  CALL READHK
  CALL DATIM
  CALL EXSHIP (for buoy data)
  CALL EXSHIP (for ship data)
  CALL EXBATHY
  CALL EXDRIFT
  CALL EXSATOB
  CALL EXTRACK
```

SSTEXT (main program)

```
NAMELIST/USER/LLIST,LDIAGR,P_SHIP,P_BUOY,P_BATHY,P_DRIFT,
      P_SATOB,P_TRACK,MAXLIST
```

```
LLIST    - LOGICAL - set to T to list out observations
              extracted
LDIAGR    - LOGICAL - set to T to monitor progress through
              program
P_SHIP    - LOGICAL - set to T to extract ship data
P_BUOY    - LOGICAL - set to T to extract buoy data
P_BATHY   - LOGICAL - set to T to extract BATHY data
P_DRIFT   - LOGICAL - set to T to extract DRIFTER data
P_SATOB   - LOGICAL - set to T to extract SATOB data
P_TRACK   - LOGICAL - set to T to extract TRACKOB data
MAXLIST   - INTEGER - maximum number of obs to list if LLIST=T
```

The default settings are LLIST=F, LDIAGR=F, P_SHIP=T, P_BUOY=T, P_BATHY=T, P_DRIFT=T, P_SATOB=T, P_TRACK=T, MAXLIST=0

```
READHK    - to read HDS UM housekeeping file
DATIM     - HDS supplied routine to obtain current date
EXSHIP    - to extract ship or buoy data from the SDB
  CALL DIFDAY
  CALL SDB
  CALL WRTSDB
EXBATHY   - to extract BATHY data from the SDB
  CALL DIFDAY
```

```
CALL SDB
CALL WRTSDB
EXDRIFT - to extract DRIFTER data from the MDB
CALL DIFDAY
CALL MDB
CALL WRTMDB
EXSATOB - to extract SATOB data from the MDB
CALL DIFDAY
CALL SDB
CALL WRTSDB
EXTRACK - to extract TRACKOB data from the MDB
CALL DIFDAY
CALL SDB
CALL WRTSDB
```

```
DIFDAY - to calculate yesterday's date
WRTSDB - to write out data extracted from SDB
WRTMDB - to write out data extracted from MDB
```

2.1 Description of Program.

This program extracts the data types described above from the SDB/MDB and writes the data to datasets ready for transfer to the CRAY for further processing.

The data is written out as three data streams, SATOBS, TRACKOBS and SURFACE which consists of all other types.

The namelist USER is read in followed by the housekeeping file, using READHK. If an error occurs in READHK the program aborts and no information is written to the transfer datasets, which will therefore contain the previous day's data, this will be checked for in the data record program. The HDS supplied routine DATIM is called to obtain the current date and time from the system clock.

A header is written out to each of the transfer datasets using an unformatted write consisting of

model id, model hour, model day, model month, model year.

The model id is written as character*4 and the other information is written as integer fullwords.

Each data extraction routine is then called in turn, assuming that the appropriate namelist variable has been set, and the reports written out to the appropriate transfer dataset. The return codes are checked and if an error has been detected a warning message is printed out and a dummy record written out to the appropriate transfer dataset. These dummy records will force the data record program to write out a warning message, but no data will be lost.

2.2 Further details on extraction of data.

2.2.1 General Details.

Data extraction from the SDB is described in Met O (CC) SDB Tech Note 1 (TN1) and data extraction from the MDB is described in Met O (CC3) Tech Note 3 (TN3). The table references and variable names given below are described in these technical notes.

SDB data is extracted as integer halfwords and MDB data is extracted as reals. However, all data is written out to the transfer dataset as integer fullwords. Each record will consist of the record length, SDB type number, data record. Since two IBM fullwords are written out to 1 CRAY fullword, the record is padded with a zero so that the first word read on the CRAY will give the record length. The actual length written out is dependent upon the data type. DRIFTER data does not have a SDB type code so a nominal one is assigned for easier processing, this nominal code being 160.

All available data is extracted but since the transfer datasets are of a finite size, there is a maximum number of observations which can be written out. A warning message is written out if any observations have been ignored. The current limits set are: SDB surface data = 6000, DRIFTER data = 3500, SATOB data = 10000 and TRACKOB data = 30. WARNING: If these limits are increased then changes to array sizes will also be needed in the quality control and SSTOPD programs and possibly the ACOBs program and SST analysis program.

2.2.2 Ship and fixed buoy data - performed by EXSHIP.

SDB types 20 and 15, table 4 in TN1

IDATIM - array of 24 dates/times. First word is previous midnight and subsequent words are each hour to 1Z the previous day.

ISTNS=100 - all reports.

retrieval array is 64 halfwords, IDATA_SDB(2)=120

ICOMB=0, IELEM=2 (section 2 of report required), IOPTN=4, ITIM=0

Length of data record written out = 63

2.2.3 Bathythermograph data - performed by EXBATHY.

SDB type 26, table 26 in TN1

IDATIM - array of 2 dates/times. First word is previous midnight and second word is 12Z the previous day.

ISTNS=100 - all reports.

retrieval array is 388 halfwords, IDATA_SDB(2)=768

ICOMB=0, IELEM=0, IOPTN=10, ITIM=0

Length of data record written out = 63 (first 63 halfwords of extracted data)

2.2.4 SATOB data - performed by EXSATOB

SDB type 148, table 148 in TN1

IDATIM - array of 24 dates/times. First word is previous midnight and subsequent words are each hour to 1Z the previous day.

ISTNS=100 - all reports.

retrieval array is 49 halfwords, IDATA_SDB(2)=94

ICOMB=0, IELEM=4 (section 4 of report required), IOPTN=10, ITIM=0

Length of data record written out = 48

2.2.5 TRACKOB data - performed by EXTRACK.

SDB type 21, table 21 in TN1

IDATIM - array of 2 dates/times. First word is previous midnight and second word is 12Z the previous day.

ISTNS=100 - all reports.

retrieval array is 298 halfwords, IDATA_SDB(2)=592

ICOMB=0, IELEM=0, IOPTN=10, ITIM=0

Length of data record written out = 297

2.2.6 DRIFTER data - performed by EXDRIFT.

Nominal SDB type 160

CSUBT='DRIFTR ', CREQ is used in the yymmdd format and is formed by using internal writes. All data from 0000Z to 2359Z the previous day is retrieved.

Observations are extracted one at a time, NOBS=1, and 26 elements are retrieved, listed in table 1.

To make the data consistent with SDB data, the latitude, longitude and SST value are stored as 10ths. Also the missing data indicator is set to -32768.

The number of observations of each type extracted and any other messages are written to both unit 6 and a message file, COP.SSTANAL.MESSAGE.

2.3 Program I/O

6	OUT	output messages	
10	IN	DRIFTER element dataset	COP.SSTDRIIFT.ELEMENTS
19	IN	housekeeping dataset	COP.IBM.HKGL
20	IN	namelist	COP.CORE.CARDLIB (SSTANAL)
25	OUT	message file	COP.SSTANAL.MESSAGE
30	OUT	surface transfer dataset	COP.SSTANAL.SURFACE
31	OUT	SATOB transfer dataset	COP.SSTANAL.SATOBS
32	OUT	TRACKOB transfer dataset	COP.SSTANAL.TRACKOBS
HOLDREC	IN	SDB.DATE	

1 BUOY/PLATFORM IDENTIFIER
TIME OF RECEIPT
2 YEAR
3 MONTH
4 DAY
5 HOUR
6 MINUTE
7 YEAR
8 MONTH
9 DAY
10 HOUR
11 MINUTE
12 POSITION ACCUARCY FLAG
13 LATITUDE
14 LONGITUDE
15 HOURS SINCE LAST POSITION GIVEN
BUOY DRIFT DATA
16 DIRECTION OF CURRENT
17 SPEED OF CURRENT
BUOY POSITION, 2nd SOLUTION
18 LATITUDE
19 LONGITUDE
20 SEA SURFACE TEMPERATURE
21 QUALITY INFORMATION
22 QUALITY OF LOCATION
ENGINEERING STATUS GROUPS
23 GENERAL ENGINEERING STATUS GROUP
24 GENERAL ENGINEERING STATUS GROUP
25 GENERAL ENGINEERING STATUS GROUP
26 ORIGINATING/GENERATING CENTRE

Table 1 DRIFTER elements extracted

2.4 Description of operational Changes.

V1.1 18 Feb 92. Initial operational version.

V1.2 14 Jul 92. Changes to EXTRACK and EXBATHY to ensure only observations for day required are extracted.

3. RALUAT

RALUAT is the program that extracts ATSR data from the MDB for transfer to the CRAY for further processing. Its actions are very similar to SSTEXT described above.

The source code comprises of the following members of CFOBS.BLSSTLIB.RALUAT; READHK, EXATSR, WRTMDB and RALUAT (main program). The job on CFOBS.BLSSTLIB.RALUAT(@CNTLTOT) is used to combine these routines into one source member, CFOBS.BLSSTLIB.RALUAT(TOTAL).

CFOBS.BLSSTLIB.RALUAT(@CNTLLOD) makes a load module of the program. The routine MDB needs to be included from SDB.LOADLIB. A copy of this job can also be found on COP.CORE.BUILD(RALUAT).

Index to program:

```
PROGRAM RALUAT
  CALL READHK
  CALL DATIM
  CALL EXATSR
```

MAIN

```
NAMELIST /USER/LTEST,LLIST,MAXLIST,LDIAGR
```

```
LTEST - LOGICAL - set to T if test run (only 100 obs are
        extracted)
```

```
LLIST - LOGICAL - set to T to print out extracted
        observations
```

```
LDIAGR - LOGICAL set to T to monitor progress through program
```

```
MAXLIST - INTEGER - maximum number of observations to print out
if LLIST is TRUE (should be set to a value of 100x+1 where x
=1,2,,∞)
```

The default settings are LTEST=F, LLIST=F, LDIAG=F, MAXLIST=0

```
READHK - to read the housekeeping file
```

```
DATIM - HDS supplied routine to obtain current date
```

```
EXATSR - to extract ATSR SST observations from the MDB
```

```
CALL DIFDAY
```

```
CALL MDB
```

```
CALL WRTMDB
```

```
DIFDAY - to calculate yesterday's date
```

```
WRTMDB - to write out extracted observation to transfer
dataset.
```

3.1 Description of Program.

The purpose of this program is to extract ATSR SST reports from the MDB and to write the reports to a dataset ready for transfer to the CRAY for further processing. Since ATSR reports have no SDB type number, a nominal SDB type is assigned to make the subsequent processing on the CRAY easier, this value is 200.

The namelist USER is read in followed by the housekeeping

file, using READHK. If an error occurs in READHK the program aborts and no information is written to the transfer dataset, which will therefore contain the previous day's data, this will be checked for in the data record program. The HDS supplied routine DATIM is called to obtain the current date and time from the system clock. A header is written to the transfer dataset using an unformatted write which consists of

Model Id, model hour ,model day ,model month ,model year,
runtime month, runtime day, runtime hour, runtime minute

The model id is written as character*4 and the other information is written as integer fullwords.

Subroutine EXATSR is called to extract SST reports from the MDB and to write out to the transfer dataset. The reports are extracted and written out in blocks of 100 observations. 22 elements are extracted for each observation which contain all the information necessary for processing on the CRAY, see table 2 for details. The list of elements extracted and their codes are stored on COP.RALUAT.ELEMENTS and COP.RALUAT.CODES respectively. The CREQ variable in the call to the MDB is written out to a dataset COP.RALUAT.EXTTIME which is then read used by the program which merges the model data back into the MDB. This ensures that identical reports are extracted in the two programs.

Each block of 100 reports is written out to a single record in the transfer dataset. The record consists of record length (integer), SDB type (real), number of reports in record (real), and the reports (real), ie.
0,RECORD LENGTH,No. REPORTS,REPORT1,REPORT2,...REPORTN.

Since two HDS words are read into one CRAY word, the record is padded with a zero. This is so that when the record is read on the CRAY, the first CRAY word will be the record length. A running total is kept of the number of reports extracted and a check is made that data volumes have not been exceeded. The extraction program will extract all ATSR reports that are available but the actual number which can be written out to the transfer dataset is restricted to avoid a 'D37' abend. Currently, the maximum number allowed to be transferred is 100000. The transfer dataset is called COP.RALUAT.OBSFILE.

If an error occurs reading the MDB then a dummy record consisting of two missing data indicators is written to the transfer dataset before the program aborts.

Finally, concluding messages are written out which contain among other things the number of observations written out to the transfer dataset and the number, if any, of excess observations which have been ignored. This information is also appended to the message file COP.SSTANAL.MESSAGE used by SSTEXT.

3.2 Return codes and action of program.

If the return code from reading the MDB, IRMDB, is not zero after extracting the ATSR observations then the program aborts.

IRMDB = 1 MDB error

IRMDB = 2 no data available

3.3 Program I/O

6	OUT	output messages	
10	IN	element dataset	COP.RALUAT.ELEMENTS
16	OUT	CREQ variable	COP.RALUAT.EXTTIME
19	IN	housekeeping	COP.IBM.HKGL
20	IN	namelist	COP.CORE.CARDLIB (RALUAT)
25	OUT	messages	COP.SSTANAL.MESSAGE
30	OUT	transfer dataset	COP.RALUAT.OBSFILE

1	SATELLITE IDENTIFIER
2	YEAR
3	MONTH
4	DAY
5	HOUR
6	MINUTE
7	SECOND
8	LATITUDE (COURSE ACCURACY)
9	LONGITUDE (COURSE ACCURACY)
	TIME OF RECEIPT
10	(TOR) YEAR
11	(TOR) MONTH
12	(TOR) DAY
13	(TOR) HOUR
14	(TOR) MINUTE
15	SATELLITE INSTRUMENT
	SEA SURFACE TEMPERATURE
16	NUMBER IN AVERAGE
17	SEA SURFACE TEMPERATURE
18	STANDARD DEV OF SST
19	MULTI/SINGLE VIEW DIFFERENCE
20	UAT PRODUCT CONFIDENCE
21	UAT ACROSS-TRACK BAND NUMBER
22	UAT QUALITY SUMMARY

Table 2 ATSR elements extracted

3.4 Description of operational changes.

V1.1 15 Oct 91. Initial operational version.

V1.2 26 Nov 91. Variable 'CREQ' increased to *200 and 'DATA CURRENT' added to CREQ variable to ensure correct data is extracted. Changes made to checks on MDB return codes.

4. SSTDATR

Four data streams are extracted on the HDS from either the SDB or MDB. These data streams are SURFACE (ships, fixed buoys, bathythermosphers, DRIFTERS), TRACKOBS, SATOBS and ATSR. Each data type is in a different format and therefore before any processing can be performed each data type must be reformatted into a universal format.

The format chosen is known as the data record format. Each observation is written to a different data record record and this record contains all the information required for the quality control, analysis and archiving of the data. A full description of the data record format is given in section 4.4.

Program SSTDATR reformats the four data streams into data records. It is designed to be flexible so that code changes are kept to a minimum when any new data sources become available.

The source code consists of the following members of CFOBS.BLSSTLIB.PREPRO. Routines: SSTDATR (main program), ATSR, BACK@IN, CHECK@OB, CLIM@IN, HEAD@IN, INTERP, MDB@IN, MISDAT, READ@HK, SATOB, SDB@IN, SET@DDS, SET@DS, SET@EVK, SET@INF, SET@OBK, SET@RQS, STAL@AT, STAL@IN, STAL@SB, STAL@SF, STAL@TR, SURFACE, TRACKOB, WRTDAT. Comdecks: COMDAT, CSSTCTRL, CTRL, FIELD, FILES@IO, GRID, HEAD, SDBMDB, STAL. The following members of the Unified Model master library are also used: C@MDI, C@PI, CHKLOO1A, CLOOKADD, IOERRO1A, POSERR1A, PRFIXH1A, PRIFLD1A, PRFLD1A, READFL1A, READHE1A, T@INT1A, P21BIT1A, NEWPAC1A, OPEN1A, CLOSE1A, BUFIN1A, which are not described in this document. The program also uses members PR@LOOK and PP@TITL from MS12.CJSSTLIB.ANALYSIS. (These routines are also UM routines but the latest versions are not used as this would tie SST changes to UM changes. This applies to all the SST programs which read/write ancillary fields although sometimes changes are enforced because of other UM changes).

The source is maintained using the CRAY nupdate facility. Changes to the source code are made by creating a MODECK which is then used to amend a program library from which an executable is then made.

There are several scripts on CFOBS.BLSSTLIB.SCRIPTS which are used to process these routines.

DTCREX - creates an executable from a program library and creates a full source listing on the CRAY.

DTCRGO - runs an executable

DTCRMD - to create a MODECK.

DTCRPL - creates a program library from the source code.

DTCRPLU - to create a new program library by modifying an old library using a MODECK.

Index to Comdecks:

COMDAT - holds data records

CSSTCTRL - holds variables for reading in ancillary fields

datasets
 CTRL - holds control variables and housekeeping information
 FIELD - holds SST fields
 FILES@IO - holds unit numbers for I/O
 GRID - holds grid details
 HEAD - holds headers of ancillary field datasets
 SDBMDB - holds data transferred from SDB and MDB
 STAL - holds station lists

Index to program:

```

PROGRAM SSTDATR
  CALL OPEN (3 times)
  CALL READHK
  CALL BACK_IN
  CALL CLIM_IN
  CALL STAL_IN (called for each data type being processed)
  CALL ATSR (if ATSR data is being processed)
  CALL SURFACE (if surface data is being processed)
  CALL SATOB (if SATOB data is being processed)
  CALL TRACKOB (if TRACKOB data is being processed)
  CALL CLOSE (3 times)
  
```

SSTDATR (main program)

NAMELIST/INPUT/LDIAGR, LDATAP, LSDBP, LDIAGS, P_SURFACE, P_SATOB, P_TRACKOB, P_ATSR, MAX_PRT_DAT, MAX_PRT_SDB, ISST_VER_NO

LDIAGR - LOGICAL - set to T to monitor passage through routines
 LDATAP - LOGICAL - set to T to list obs in data record format
 LSDBP - LOGICAL - set to T to list data transferred from SDB/MDB
 LDIAGS - LOGICAL - set to T to list station lists
 P_SURFACE - LOGICAL - set to T to process surface data
 P_SATOB - LOGICAL - set to T to process SATOB data
 P_TRACKOB - LOGICAL - set to T to process TRACKOB data
 P_ATSR - LOGICAL - set to T to process ATSR data
 MAX_PRT_DAT - INTEGER - number of obs to list in data record format (for each data type being processed)
 MAX_PRT_SDB - INTEGER - number of obs to list in SDB/MDB format (for each data type being processed)
 ISST_VER_NO - INTEGER - SST analysis version number

No default values are set in the program. The values read in operationally are;

LDIAGR=F, LDATAP=F, LSDBP=F, LDIAGS=F, P_SURFACE=T,
 P_SATOB=T, P_TRACKOB=T, P_ATSR=T, MAX_PRT_DAT=1, MAX_PRT_SDB=1,
 ISST_VER_NO=1

READ_HK - to read in housekeeping file
 BACK_IN - to read in background SST field and background error field
 CALL READHEAD

CALL READFLDS
 CLIM_IN - to create the day's climatological SST field by interpolating in time to today's date from adjacent months.
 CALL READHEAD
 CALL READFLDS
 CALL PR_LOOK
 CALL PR_RFLD
 CALL T_INT
 STAL_IN - to read in a station list.
 ATSR - to process ATSR SST reports
 SURFACE - to process surface SST reports
 SATOB - to process SATOB SST reports
 TRACKOB - to process TRACKOB SST reports
 These four routines (if called) are similar in their operation.
 CALL HEAD_IN
 CALL SDB_IN or MDB_IN
 CALL CHECK_OB
 CALL MISDAT
 CALL SET_OBK
 CALL SET_EVK
 CALL SET_INF
 CALL SET_RQS
 CALL SET_DDS
 CALL SET_DS
 CALL STAL_SF or STAL_SB or STAL_TR or STAL_AT
 CALL WRDAT
 HEAD_IN - to read in and convert to CRAY numbers the header of the transfer dataset
 CALL USICTC (CRAY routine to convert integers)
 CALL USCCTC (CRAY routine to convert characters)
 SDB_IN - to read in and convert SDB data
 CALL USICTC
 CALL USCCTC
 MDB_IN - to read in and convert MDB data
 CALL USSCTC (CRAY routine to convert reals)
 CHECK_OB - to check that extracted observations are within the correct model and time window.
 MISDAT - to check observation for missing data; ie. the latitude outside the bounds of 90 to -90 or a longitude outside the bounds of 180 to -180 or the SST value set to missing data. Also observations that have reported a location of 0°lat and long are also rejected.
 SET_OBK - to set the observation key
 SET_EVK - to set the event key
 SET_INF - to set the information block
 SET_RQS - to set the report quality section
 SET_DDS - to set the data description section
 SET_DS - to set the data section
 CALL INTERP - to interpolate a model field to the observation location
 STAL_SF/STAL_SB/STAL_TR/STAL_AT - to set values from the station list in the data record for Surface/SatOB/Trackob/ATsr data
 WRDAT - to write out data record to external file

4.1 Description of Program.

SSTDATR creates the data records for all data types which are then passed on to the quality control stage.

The climatological SST file, background field and background error field datasets are all opened and then the control namelist is read. If an error occurs whilst opening the datasets then a message is printed and the program aborts. The climatological field is created for the current date by interpolating between adjacent months.

The subroutines which control the processing of the various data types are called SURFACE, SATOB, TRACKOB and ATSR. Data transferred from the HDS has a header containing the model ID, the housekeeping date and in the case of ATSR data the runtime of the extraction job. This header is read in and converted and a check is made to ensure that the dates correspond with the CRAY housekeeping file. There are only minor differences in the actions of the four controlling routines and as can be seen from above there exists a number of routines used by all data types. In these common routines, the data type is identified by means of the SDB type number, this is the reason for assigning a nominal SDB type number to ATSR data and DRIFTER data.

Variables which are used for only one data type end with either '_SF', '_SB', '_TR' or '_AT' depending upon the data type. Variables which are common to all data types, either in comdecks or in common data processing routines have no such ending. Data transferred from the HDS is stored either as integers (SDB data) or as reals (ATSR data). However, the common routines need both and therefore, the real array for SDB data and the integer array for ATSR data is created and filled with missing data indicators. The data record is stored as both real and integer but is set as REAL. Bit setting is accessed by integer and surface and trackob call signs are stored as characters.

SURFACE - handles ships, fixed buoys, drifting bouys and bathythermographs. Observations are read in and processed one at a time. Pointers are set according to the SDB type number which identify the location of lat/long/sst/hour of report in the HDS data. The various sections of the data record are then set in turn. Observations which fail the checks in CHECK_OB are discarded. Once all observations have been processed. summary messages are printed out and control is passed back to the main program.

SATOB - handles SATOB SST reports and follows the same actions as SURFACE.

TRACKOB - handles TRACKOB SST reports.

TRACKOB SST reports are SST observations taken along a ship's track. Therefore, in a single TRACKOB report there may be up to 24 individual SST reports. The subroutine TRACKOB

reads in each TRACKOB report transferred from the HDS in turn but then processes each individual SST observation within the TRACKOB report as a separate ship observation, each with its own data record. At least one data record is produced for each TRACKOB report and the processing for each data record is the same as for SURFACE data.

ATSR - handles ATSR SST data.

ATSR data is transferred from the HDS in blocks of 100 observations. Each ATSR block is read in one at a time and for each block one data record is produced for each individual SST report. SST reports which fail the checks in CHECK_OB are not discarded as this would upset the final merge step, instead they are simply flagged as unusable. A test is made as whether the report is flagged as a daytime report. A temporary SDB type of 201 (day) or 202 (night) is created for use in routine SET_EVK so that the correct model observation type can be set.

All data types have a corresponding station list. The station list is used to set the values in the data record which are required in the quality control stage. For each type there is a default list but there is also the facility to set values for individual identifiers. This facility enables particular call signs to be blacklisted by setting the 'NOT USE' and 'DEACTIVATE' flags to 1. The format of the station list is described in section 4.5.

4,2 Return codes and action of program.

(All return codes are 0 if no problem encountered. If a problem is encountered then an explanatory message is printed out).

From OPEN routines, if return code is not zero the program aborts and the return code from the OPEN routine is printed.

IR_READ_HK = 1 problem opening housekeeping file
 = 2 problem reading housekeeping file
 = 3 problem closing housekeeping file

if IR_READ_HK not zero then the program aborts.

IR_BACK_IN = 1 problem reading in background SST dataset, program aborts.

IR_CLIM_IN = 1 problem with climatological SST dataset, program aborts.

IR_READ_SL = 1 too many records on station list, program aborts.

IR_SURFACE, IR_SATOB, IR_TRACKOB, IR_ATSR, = 1, error in processing that data type, warning messages are produced as no data records will be produced for that data type but program execution continues.

IR_DAT_IN - reached end of input dataset transferred from HDS, informative only.

4.3 Program I/O.

6	OUT	output messages	
10	IN	control namelist	£IN/qisstdat.nlist
11	IN	housekeeping file	\$DATAW/qwgl.hkfile
15	IN	SF station list	£DATAR/qrsstsf.slist
16	IN	SB station list	£DATAR/qrsstsb.slist
17	IN	TR station list	£DATAR/qrssttr.slist
18	IN	AT station list	£DATAR/qrsstat.slist
21	IN	SF data from HDS	£DATAT/qtsstsf.obsibm
22	IN	SB data from HDS	£DATAT/qtsstsb.obsibm
23	IN	TR data from HDS	£DATAT/qtssttr.obsibm
24	IN	AT data from HDS	£DATAT/qtsstat.obsibm
31	OUT	SF data records	£DATAT/qtsstdata.surf
32	OUT	SB data records	£DATAT/qtsstdata.stob
33	OUT	TR data records	£DATAT/qtsstdata.trck
34	OUT	AT data records	£DATAT/qtsstdata.atrsr

The ancillary fields datasets are assigned using environment variables. All are opened as read only.

SSTINSURF=£DATAW/qwgl.daily.sst (background SST field, unit 40)
ERRINSURF=£DATAW/qwbgerr.glsst (background error field, unit 41)
SSTCLIMSURF=£DATAW/qwgl.clim.sst (climatological SST dataset, unit 42)

4.4 DATA RECORD DESCRIPTION

Each SST report extracted from the SDB/MDB is reformatted into a data record, the data record has the same format for all observation types. The data record contains all the information required for the observation processing, analysis and archiving into the SSTOPD in the case of SDB observations or merging back into the MDB for ATSR reports. The format is intended to be flexible enough to meet all requirements and will remain the same at all stages of processing. In all important aspects the format is the same as BUFR, the only differences being the use of SSTOPD-like header blocks (observation key, event key and information block) and in the detailed specification of the data descriptors. Most of the syntax of BUFR applies; eg the concepts of location classes and data association. In most cases each observation will be stored in a separate record, as in the case of the SSTOPD, but where necessary it is possible to store several observations within one record to save space. The record consists of six parts:

- Observation Key (OK)
- Event Key (EK)
- Information Block (IB)
- Report Quality Section (RQS)
- Data Descriptor Section (DDS)
- Data Section (DS)

The observation key, event key and information block are similar to the SSTOPD (Jones et al 1991) and contain

amongst other things the observation time and position, and information relating specifically to the observation. To conform to the standards of BUFR this information may also be stored in the data section, and this will indeed will be necessary where there is more than one observation within the data record.

The report quality section contains flags and other information on quality relating to all the data within the data section. It consists of:

1. Length of section (words)
2. Report flags/indicators

The data descriptor section (DDS) and the data section (DS) map directly onto their BUFR equivalents. The DDS consists of:

1. Length of section
2. Number of data subsets
3. Data type and format
4. Collection of Replication Operators (RO) and Sequence Descriptors (SD)

Following the BUFR convention a RO is as follows:

1. Descriptor identifier (=1)
2. Scope of operator (number of SD and RO being replicated)
3. Number of replications

A SD uses bits to indicate the presence of parameters and in this respect differs from BUFR which uses table entries. each SD consists of:

1. Descriptor identifier (=3)
2. Length of Coordinate Presence Indicator (CPI) in words
3. Length of Element Presence Indicator (EPI) in words
4. Length of Data Presence Indicator (DPI) in words
5. Length of element flags in words
6. CPI
7. EPI
8. DPI

The DS consists of:

1. Length of section in words
2. Data (as indicator by the data descriptor section)

As in the SSTOPD and allowed for in BUFR, each observed value may have data associated with it (eg difference from background, flags etc). The bits of the EPI define which observed elements are present in the record, and the bits of the DPI define what associated data follow each value. Data which fall into the coordinate class (eg position, time) are defined by the bits of the CPI but have no associated data. Since the lengths of the EPI/DPI/CPI/flags are included in the data record there is flexibility to cope with new data types and flags. The arrangement of the data in the data section follows the rules of the current SSTOPD; each observed element and its associated data are stored consecutively in the order

given by the DPI, and the data for relating to each element are stored in the order given by the DPI, and the data for relating to each element are stored in the order given by the EPI. As in BUFR, coordinate values remain valid until redefined.

Observation Key

The observation key is set as follows.

word	contents
1	length of key
2	64* SDB type
3-6	identifier
7	latitude (-90 to 90)
8	longitude (180 to -180)
9	date of observation (= 512*year+32*month+day) (year is figures)
10	time of observation (= 256*hour+minute)
11	date of receipt (as for 9)
12	time of receipt (as for 10)
13	collecting centre (SDB data only)
14	...not used...
15	...not used...
16	...not used...

Event Key

The event key is set as follows.

word	contents
1	length of key
2	SST analysis version number
3	hours since last SST analysis
4	model ID
5	validity time of analysis (256*year+month)
6	validity time of analysis (256*day+hour)
7	...not used...
8	256*model observation type

Information Block

The information block is set as follows.

word	contents
1	length of information block
2	...not used...
3	...not used...
4	report sequence number (TRACKOB only)
5	...not used...
6	No. of reports (superobs only)
7	standard deviation of reports (superobs only)
8	range of reports (superobs only)

Report Quality Section

The report quality section is set as follows.

word 1: length of block
word 2: bits set as follows:
bit description

0	SDB quality control performed
1	SDB movement check performed
2	...not used...
3	SDB position flag raised
4	SDB movement flag raised
5	...not used...
6	...not used...
7	duplicate report
8	...not used...
9	unusable report
10	do not use in analysis or buddy check
11	deactivate report
12	...not used...
13	...not used...
14-31	...not used...

word 3: ...not used...

word 4: a priori probability of error in location

word 5: intermediate probability of error in location

word 6: final probability of error in location

Coordinate Presence Indicator (CPI)

bit	description
0	pressure level 1
1-31	...not used...

Element Presence Indicator (EPI)

bit	description
0-7	...not used...
8	skin temperature (for ATSR)
9-11	...not used...
12	sea surface temperature
13-63	...not used...

Data Presence Indicator

bit	description
0	observed value
1	difference from background
2	difference from background used in retrieval
3	difference from analysis
4	quality control flags
5	observation error
6	correction
7	background error
8	observation error/background error
9	a priori probability of gross error
10	probability of gross error after background check
11	probability of gross error after buddy check
12	difference from climatology
13-15...	not used...

a bit set in this specification indicates that at least one datum is present somewhere in the record; the levels without the data present should have missing data indicators set.

Quality Control bits

bit	description
0-2	...not used...
3	secondary rejection
4	secondary correction
5	permanent rejection
6	permanent correction
7	checked mode 1
8	checked mode 2
9	used as buddy
10	...not used...
11	background flag
12	buddy check flag
13	final flag
14	SDB suspect flag
15	superob failed range check
16	superob failed standard deviation check
17	superob failed location check
18	superob failed earlier flags check
19-31	...not used...

RECORD FORMAT AT THE END OF THE FINAL CONVERSION STAGE

Lengths are in words:

	SF	SB	TR	AT

Observation key :				
length :	16	16	16	16
words set :	1-13	1-2, 6-13	1-13	1-2, 6-13
Event key:				
length :	8	8	8	8
words set :	1-6, 8	1-6, 8	1-6, 8	1-6, 8
Information block :				
length :	8	8	8	8
words set :	1	1	1,4	1
Rpt quality section :				
length :	6	6	6	6
words set :	1-2, 4-6	1-2, 4-6	1-2, 4-6	1-2, 4-6
word 2 bits used :	1,3, 4	9-11	1,3, 4	9-11
	9-11		9-11	
data description section				
length of section :	12	12	12	12
No. of subsets :	1	1	1	1
Data format value :	128	128	128	128
SD1:				
DI :	3	3	3	3
length CPI :	1	1	1	1
length EPI :	2	2	2	2
length DPI :	1	1	1	1
length flgs :	1	1	1	1
CPI bits :	0	0	0	0
EPI bits :	12	12	12	8
DPI bits :	0,1,4,5,6,7,9,12 (for all data types)			
data section				
length of section :	11	11	11	11
	---	---	---	---
TOTAL length :	61	61	61	61

Notes of identifiers;

SurFace and TRackob data have character identifiers stored as integers in words 3 to 6 of the observation key. SatoB and ATsr data have a satellite identifying number stored in word 6 of the observation key.

Current satellite identifiers are:

For SATOBS, 3/4/5 = Meteosat 3/4/5
 104 = Himiwari satellite
 236 = NOAA-11

For ATSR, 1 = ERS-1

Notes on type numbers.

The following model type numbers are used

	SDB type	Model ob type
Ships	20	12
DRIFTER	160	17
Fixed buoys	15	13
BATHY	26	26
TRACKOB	21	25
SATOB (Meteosat)	148	28
SATOB (NOAA)	148	27
SATOB (JAPSAT)	148	29
ATSR	200	30
ATSR day	201	301
ATSR night	202	302
bogus SST	170	32
DRIFTER superob	160	35
Fixed buoy superob	15	36

The AC ob type for SST data is 702.

4.5 STATION LIST

The station list is used to add information into the data records about the observations which is then used in the subsequent processing stages. The information provided includes the prior probabilities of gross error and the observation error. There exists two types of records, an information record and a rejection record. The information record contains default values for each observation type although values for specific identifiers can also be specified. The rejection records are used to flag callsigns that are blacklisted. The blacklist is updated monthly, see below.

At present correction records cannot be processed.

1 Information record

- 1, record length
2. record type (2 = information)
3. SDB type
4. length of WMO identifier
5. WMO identifier
6. observation error
7. probabilities of gross error
8. data use flags
9. data use/correction parameters

If a station is on the list then all the values given above are expected. If a station is not on the list then default values of these parameters will be assigned. The defaults for each SDB type will appear at the beginning of the station list and will be identified by a record type of 0. Flags in the record are activated by setting a value of 1 and deactivated by setting a value of 0.

The information record is the same for all SDB types and contains the following values.

- 1.1 record length
- 1.2 record type
- 2.1 SDB type
- 2.2 length of WMO identifier
- 2.3 WMO identifier
- 3.0 length of observation error section
- 3.1 observation error for SST
- 4.0 length of probability section
- 4.1 a priori PGE for SST
- 4.2 a priori PGE for location
- 5.0 length of flag section
- 5.1 do not use in analysis or buddy check (bit 10 of RQS)
- 5.2 deactivate report (bit 11 of RQS)
- 5.3 use as buddy (bit 9 in Q/C bits)
- 6.0 length of correction section

4.6 Criteria for blacklisted callsigns.

A callsign will be added to the blacklist if:

i) the number of observations is greater than 10 and either the rms of O-A is greater than 2.5K or the percentage of gross errors, O-A greater than 5.0K, is greater than 25. In making these assessments only the previous month is considered.

Once a callsign is on the blacklist it will only be removed if during the previous three months there has been no observations received from the callsign or the rms of O-A is less than 2.0K and the percentage of gross errors, O-A greater than 5.0K, is less than 25.

4.7 Procedure for Operational Changes.

The procedure to make changes to the program is as follows.

- 1) Edit the source code of the routine(s) that need to be changed. Note that the main program should always be changed by incrementing the version number and date implemented operationally and a brief description of the change.
- 2) Create a MODECK using the CRAY nupdate and nmodex utilities. A MODECK contains the changes between the current source code and new source code. (script DTCRMD)
- 3) Create a new program library using the MODECK. (script DTCRPLU).
- 4) Create a new executable. (script DTCREX)
- 5) Test the executable.

4.8 Description of operational changes.

V1.1 15 Oct 91. First operational version.
Program library 1

23 Oct 91. Correct how character call signs are written to data record.
Modset CJ221091 Program library 2

26 Nov 91. Add model observation type for BATHYS.
Modset CJ111191 Program library 3

V1.2 10 Dec 91. Code to handle DRIBU data is replaced by code to handle DRIFTER data. DRIBU data no longer extracted.
Modset CJ211191 Program Library 4

7 Jan 92. Change to surface data volume.
Modest CJ111292 Program library 5

V1.3 18 Feb 92. Remove code that is dependent upon knowing volume of data expected. Now any volume of data can be handled.
Modset CJ220192 Program library 6

V1.4 31 Mar 92. Correct bug in CHECK_OB (lat/long divided by 100 not 10), increase max number of station lists to 3000 and set

final flag for ATSR single views so that they are rejected by the quality control.

Modest CJ240392 Program library 7

V1.5 5 May 92. Corrections made to station lists routines.

Modest CJ080492 Program library 8

V1.6 1 Dec 92. Additions to MISDAT to reject observations reporting at 0°lat 0°long. Test for ATSR day/night so that separate model observation types can be set.

Modest DH171192 Program library 9

V1.7 23 Feb 93. Update program to comply with version 3.0 of UM. This includes changing to portable I/O routines and updating all UM routines except PR_LOOK and PP_TITL.

Modest DH200193 Program library a

V1.8 06 Jul 93. A number of changes to correct potentially unsafe code that causes problems when the program is built and run under Unicos 7. The equivalencing of the input data has been removed and INTERP has been changed as some observations near the coast were being rejected as being over land when in fact one of the surrounding grid points was sea. Also, executable now has to be built without any optimization or vectorization.

Modest CJ010793 Program library b

Program library remade are renumbered to be number 1 from next version of program.

V1.9 19 Oct 93. Update program to comply with UM version 3.2.

Program library 1

5. QXSSTQC.PGM

After the data records have been created, quality control is performed by program qxsstqc.pgm. All data streams are quality controlled by the same executable although there are separate routines for each data type. The program has been adapted from the atmospheric quality control program described by Ingleby and Parrett (1991) in UMDP P52.

The source code consists of the following members of MS12.CJSSTLIB.QCONTROL. Routines: QCMAIN (main program), QCREAD, QCBKSB, QCBKSF, QCBKTR, QCBKAT, QCUDDS, QCINIT, QCFINL, QCLISL, QCWRIT, QCSTAT2. Comdecks: COMSFSST, COMSBSST, COMTRSST, COMATSST, COMIO, CNTRQC, CNTRBD, C@MDI, C@PI.

There are several scripts on MS12.CJSSTLIB.SSTCNTL which are used to process these routines.

QCIBMFRT - is used to preprocess the source code on the HDS. The data type(s) to be processed by the resulting program is specified by stating SATB, TRCK, SURF or ATSR. The resulting program, with all comdecks in place, is written to MS12.CJSSTLIB.SRCEGO(IQCNTL).

QCCREXP - is used to run program from source

QCCREX - creates an executable

QCCRGO - runs the executable

Index to Comdecks;

COMIO - holds unit numbers for I/O

CNTRQC - holds information required for processing

COMSSTSF - holds surface SST data

COMSSTSB - holds SATOB SST data

COMSSTTR - holds TRACKOB SST data

COMSSTAT - holds ATSR SST data

CNTRBD - holds information for buddy checking (not yet implemented)

C_MDI - missing data indicators

C_PI - holds constant, π

Index to program:

Operationally all preprocessing options are enabled and it is this program that is described here.

PROGRAM QCMAIN

CALL QCINIT

CALL QCBKSB

CALL QCBKSF

CALL QCBKTR

CALL QCFINL (three times, once each for SF, SB and TR)

CALL QCBKAT (repeated for each block of observations)

CALL QCFINL

QCMAIN (main program)

QCINIT - to read in namelist and initialise variables in CNTRQC
NAMELIST/NAMLQC/TOPHAT_SF, TOPHAT_SB, TOPHAT_SF, TOPHAT_AT, NO_PRT
, NO_WORDS_PER_BLK, L_GLOBAL, L_BUD_CHK, L_BLOCK, L_SB, L_STATS

TOPHAT_SF - REAL - 'k' for surface in QC equations
 TOPHAT_SB - REAL - 'k' for SATOBS in QC equations
 TOPHAT_TR - REAL - 'k' for TRACKOBS in QC equations
 TOPHAT_AT - REAL - 'k' for ATSR data in QC equations
 NO_PRT - INTEGER - No. of each data type to print
 NO_WORDS_PER_BLK - INTEGER - block length (in words) for blocked output
 L_GLOBAL - LOGICAL - set to T for global analysis
 L_BUDCHK - LOGICAL - set to T to implement buddy checking
 L_BLOCK - LOGICAL - set to T to block output data
 L_SB - LOGICAL - set to T to quality control SATOB data
 L_STATS - LOGICAL - set to T to generate statistics

The default values are; TOPHAT_SF=0.15, TOPHAT_SB=0.15,
 TOPHAT_TR=0.15, TOPHAT_AT=0.15, NO_PRT=20, NO_WORDS_PER_BLK=50000,
 L_GLOBAL=T, L_BUDCHK=F, L_BLOCK=F, L_SB=T, LSTATS=T.
 Currently no namelist is read into the program and the variables are left to the default values.

QCBKSB/QCBKSF/QCBKTR/QWCBKAT (SatoB/SurFace/TRackob/ATsr) - to set PGE values. The initial PGE values set from the station list (PGE0) are modified from the SDB checks to give PGE1 (for SATOBS there are no SDB checks so that PGE1=PGE0). Currently, flags raised on ATSR data by the data producers are not used so that again PGE1=PGE0. The background check is performed to give PGE2 and finally PGE2 is copied to PGE3.

CALL QCREAD - to read in data records
 CALL QCUDDS - to unpack data description section of first observation
 QCFINL - to set final flag in data record.
 CALL QCUDDS
 CALL QCWRIT - to write out data records
 CALL QCLISL - to list out observations
 CALL QCSTAT - to produce summary statistics

5.1 Description of program.

The control variables in CNTRQC are set by calling QCINIT and then the background check is performed for each data type in turn by calling QCBKSB, QCBKSF, QCBKTR and QCBKAT. Since there is too much ATSR data to process at the same time, the observations are read in blocks of 10000 observations. QCBKAT is called as many times as there are still observations to process.

The actions of QCBKSB, QCBKSF, QCBKTR and QCBKAT are identical for each data type. The observations are read into the appropriate array by calling QCREAD and the data description section is unpacked for the first observation. This is then used to set pointers various elements of the data record. Since all the observations are read into one 1-d array all data records must be the same length as the pointers are incremented by the length of the record obtained from the first observation. Then the observations are processed in turn and

the probability that the observation is in gross error is calculated using the equations given below.

After all the background checks have been performed, QCMAIN calls QCFINL in turn for each data type. QCFINL first calls QCUDDS to unpack the data description section so that pointers are set and then each observation is looped through and a final flag set according to the value of PGE3 calculated. Once all observations have been processed, QCWRIT is called to write out the data records ready to make an ACOBS file.

The program has a limit of how many observations of each type can be processed. The limits are set to be the same as those used by the extraction programs so there is no danger of these limits being exceeded operationally.

5.2 Return codes and action of program.

If there is a problem in the processing of data an error message is written out and the program aborts.

5.3 Program I/O

```
6  OUT output messages
15 IN  namelist (not used operationally)
31 IN  SF data records      £DATAT/qtsstdat.surf
32 IN  SB data records      £DATAT/qtsstdat.stob
33 IN  TR data records      £DATAT/qtsstdat.trck
34 IN  AT data records      £DATAT/qtsstdat.atrsr
41 OUT SF data records after QC £DATAT/qtsstqc.surf
42 OUT SB data records after QC £DATAT/qtsstqc.stob
43 OUT TR data records after QC £DATAT/qtsstqc.trck
44 OUT AT data records after QC £DATAT/qtsstqc.atrsr
55 OUT summary statistics    £DATAT/qtsst.qcstats
```

5.4 Description of operational changes.

```
V1.0 26 Feb 91. First operational version.
V1.1  1 Dec 91. Changes to storage allowed in comdecks.
V1.2  2 Jan 92. Routine for ATSR data added.
V2.2  8 Jun 93. Rebuilt under Unicos 6.
V2.3 14 Sep 93. Correction to background checking routines to
enable them to handle empty data record datasets.
```

5.5 Current Equations Used in QC Algorithms.

Constants used in the equations are listed at the end of this section.

Each observation is assigned a value for the probability of error in location and the probability of error in the SST value. For ships, buoys, DRIFTERS and TRACKOBS a test is made of whether the observation has failed the SDB location checks. If it has then the probability of error in location is increased otherwise, assuming the check has been performed, the probability is decreased.

The variance of the observation minus background is calculated using

$$V = E_o^2 + E_b^2 \quad (5.1)$$

where,

V is the variance, E_o^2 is the observation error variance and E_b^2 is the error variance of the background field.

In the following equations, $P(\)$ denotes probability, O is the event that an observation has a particular value, G denotes a gross error and the subscript l denotes location error. An overbar represents 'not' and \cup denotes 'or'.

Expressions such as $P(O|\bar{G})$ denote the probability of the observed value already knowing that there is no gross error.

Observations are assumed to have either Gaussian observational errors or gross errors. The probability of the observed departure from background, x , for those observations with no gross errors is

$$P(O|\bar{G}) = N(x, V) = \exp(-x^2/2V) / \sqrt{(2\pi V)} \quad (5.2)$$

Observations with gross errors contain useful information and are assumed to have a uniform probability function of width $1/k$. For these observations the probability of the observed departure from background is

$$P(O|G) = k \quad (5.3)$$

The probability of the observed minus background for all observations is given by

$$P(O) = kP(G) + N(x, V) (1 - P(G)) \quad (5.4)$$

Using Bayes theorem, the final probability of the observation being in gross error is given by

$$P(G|O) = P(O|G) P(G) / P(O)$$

$$\therefore P(G|O) = kP(G) / (k(P(G) + N(x, V) (1 - P(G)))) \quad (5.5)$$

If the observation is assumed to have a location error of probability $P(G_1)$, which in practice is all observations except SATOBS, then equation 5.4 gives $P(\bar{O}|G_1)$. Equation 5.5 is replaced by the following equation.

$$P(G \cup \bar{G}_1 | O) = P(G_1 | O) + kP(G) P(\bar{G}_1) / (P(O|G_1) P(G_1) + P(O|\bar{G}_1) P(\bar{G}_1))$$

$$P(G \cup \bar{G}_1 | O) = P(G_1 | O) + kP(G) P(\bar{G}_1) / (P(O|G_1) P(G_1) + P(O|\bar{G}_1) P(\bar{G}_1))$$

$$\therefore P(G \cup \bar{G}_1 | O) = (P(O|G_1) P(G_1) + kP(G) P(\bar{G}_1)) / (P(O|G_1) P(G_1) + P(O|\bar{G}_1) P(\bar{G}_1))$$

$$\therefore P(G \cup \bar{G}_1 | O) = (kP(G_1) + kP(G) P(\bar{G}_1)) / (kP(G_1) + P(\bar{G}_1) (kP(G) + N(x, V) P(\bar{G}))) \quad (5.6)$$

It can be seen that if $P(G_1)$ is zero then equation 5.6 reduces to equation 5.5.

The following table gives the values used for the quality control system.

	E_o^2	$P(G_1)^a$	$P(G_1)^b$	$P(G)^c$	$P(G)^d$	k
<u>Ships</u>	1.2	0.07	0.75	0.04	0.8	0.15
<u>Fixed buoys</u> ¹	0.4	0.07	0.75	0.01	0.8	0.15
<u>Drifters</u> ¹	0.4	0.04	0.75	0.04	0.8	0.15
<u>Bathys</u>	0.9	0.07	0.75	0.04	0.8	0.15
<u>Trackobs</u>	1.2	0.07	0.75	0.04	0.8	0.15
<u>SATOBS</u> ²	0.6	0.00	N/A	0.02	0.8	0.15
<u>ATSR</u> ³	0.5	0.00	N/A	0.02	0.8	0.15

Superscripts denote the following.

a - prior value of probability of location error

- b - probability of location error if SDB movement/position flag raised
- c - prior value of probability of gross error
- d - probability of gross error if SDB suspect flag raised

- 1 - the value used in the assimilation is calculated using the superobbing equations, see section 6.
- 2 - includes AVHRR, Meteosat and Himiwari satellites.
- 3 - includes both day and nighttime data.

E_b^2 is set to 0.8K everywhere.

6. QXSST.SUPEROB

Fixed buoys and drifting buoys (DRIFTERS) will tend to report many times during one day from the same location, up to 50 times in some cases. In the past this has led to the analysis becoming biased towards the buoy data, especially in areas where the buoys are the only data source. To avoid this 'superobs' are created, there being one superob per buoy that has reported during the day.

The program also creates bogus SST observations in the areas of ocean that are covered by sea-ice.

The source code consists of members from library MS12.CJSSTLIB.QCONTROL: SSTDUP (main program), CDUPS, READHK, C@MDI, COMIO, COMSFSST, CNTRQC, QCREAD, QCUDDS, QCWRIT and ICEOBS and members IN@FLDS and CANCEINT from MS12.CJSSTLIB.ANCIL. The following members of the Unified Model master library are also used: BUFIN1A, BUFOUT1A, CLOOKADD, READHE1A, READFLD1, PRFIXH1A, PRIFLD1A, PRRFLD1A, POSERR1A, CHKLOO1A, IOERR1A, P21BIT1A, NEWPAC1A, OPEN1A and CLOSE1A. In common with other SST programs, members PR@LOOK and PR@TITL from MS12.CJSSTLIB.ANALYSIS are also used.

The source is maintained using the CRAY nupdate facility. Changes to the source code are made by creating a MODECK which is then used to amend a program library from which the executable is then made.

There are several scripts on MS12.CJSSTLIB.SSTCNTL that are used to process these routines.

SOCREX - creates an executable from a program library and creates a full source listing on the CRAY.

SOCRGO - runs an executable

SOCRMD - creates a MODECK

SOCRPL - creates a program library from the source code.

SOCRPLU - to create a new program library by modifying an old library using a MODECK.

Index to Comdecks:

C_MDI - missing data indicators
CANCEINT - variables required for reading ancillary fields
CNTRQC - holds information required for processing
COMIO - holds unit numbers for I/O
COMSFSST - holds SST surface data

Index to program:

PROGRAM SSTDUP
CALL READHK
CALL QCREAD
CALL OPEN
CALL IN_FLDS
CALL CDUPS (twice)
CALL ICEOBS
CALL QCWRIT
CALL CLOSE

SSTDUP (main program)

NAMELIST/CTRL/LDIAG, LDATAP1, LDATAP2, FREQ_ICE, ICE_ERR,
MAX_NO_CALL, MAX_REP_CALL, QC_SD, QC_POS, QC_RANGE, EBASE,
MAX_NO_ICE

LDIAG - LOGICAL - set to T for diagnostic output
LDATAP1 - LOGICAL - set to T to write out data records in CDUPS
LDATAP2 - LOGICAL - set to T to write out data records in
ICEOBS
FREQ_ICE - INTEGER - frequency at which to set bogus SST
observations under seaice
ICE_ERR - REAL - observation error to assign to bogus SST
under seaice observations.
MAX_NO_CALL - INTEGER - maximum number of callsigns that can
be processed
MAX_REP_CALL - INTEGER - maximum number of reports that one
callsign can have
QC_SD - REAL - quality control limit for standard
deviation check
QC_POS - REAL - quality control limit for position check
QC_RANGE - REAL - quality control limit for range check
EBASE - REAL - EBASE value in superob observation error
equation (see later)
MAX_NO_ICE - INTEGER - Maximum number of bogus SST
observations to create.

The default values are:

LDAIG=F, LDATAP1=F, LDATAP2=F, FREQ_ICE=5, ICE_ERR=0.1,
MAX_NO_CALL=1000, MAX_REP_CALL=50, QC_SD=0.5, QC_RANGE=1.0,
QC_POS=0.5, EBASE=0.7, MAX_NO_ICE=10000

READ_HK - to read housekeeping file
QCREAD - to read in surface data records after quality control
stage
OPEN - to open daily sea-ice file
IN_FLDS - to read in sea-ice concentration field
CDUPS - to create superobs
CALL QCUDDS - to unpack data description section of first
observation
CALL QCWRIT
ICEOBS - to create bogus SST observations
CALL QCWRIT
QCWRIT - to write out data records
CLOSE - to close daily sea-ice file

6.1 Description of Program.

Program SSTDUP reads in the housekeeping file and opens and then reads the daily sea-ice dataset. It then reads the surface data records that have been previously created from the quality control stage. CDUPS is called twice, once for fixed buoys and once for DRIFTERS, and superobs created.

In CDUPS all the counting variables and arrays are initialised. Then the DDS of the first section is unpacked to obtain the pointers to the data contained in the data records.

Then, all the observations are searched and the observation number of each required type is stored in array INDEX_TYPE and a count of how many of the required type has been found. (Here type is referring to the model observation type).

Next, all observations from the required type are searched to match observations from the same callsign. The callsign from the current observation is compared to all callsigns already found and if it matches then the arrays for that callsigns are updated, otherwise the callsign is added to the list. For each report for each callsign the SST values, latitude, longitude, difference from background, difference from climatology and a check of flags raised in either the report quality section or quaility control seaction are stored. If the number of reports per callsign is exceeded, the remaining reports of that callsign are ignored and similarly if the number of callsigns reaches the maximum then all remaining callsigns are ignored. Messages are written if either of these situations occur.

The mean SST, O-B and O-C of the reports for each callsign are calculated together with the range of the reported SST values. The standard deviation is also calculated. Some callsigns do not report SST and these are ignored and also if a callsign only reports SST occassionally then only the SST values it reports are considered.

Data records for all superobs are the created and the mean SST, O-B and O-C written to the record as well as the standard deviation and range of SSTs from the callsign. For callsigns that have only reported one SST report, the superob created is simply the original observation with the model observation type changed. The location of he superob is the last location reported by the callsign. The report quality section of the superob is copied from the first report of the callsign. This ensures that SDB or blacklist flags are captured and implies that if the first report from the callsign has been flagged unusable (eg over land) then the superob will also be flagged as unusable.

Next, the observation error is calculated as a function of the number of reports received from the callsign and the standard deviation of those reports.

The quality control flags of the superob are then set. The superob will have a flagged raised if any one of the following conditions is met. (The permitted limits are generally set by namelist variables)

- 1) the range of the reported SST values exceeds the permitted limit.
- 2) the standard deviation of the reported SST values exceed the permitted limit.
- 3) the callsign has deviated by more than the permitted amount in either the latitude or longitude direction.
- 4) the last report from this callsign is over land.
- 5) more than 25% of the individual reports have failed the earlier background check.

The created superob data records are written out to a new dataset and summary totals are listed.

Bogus SST observations under seaice are created in routine ICEOBS. The sea-ice concentration field is looped through and at every `FREQ_ICE` point a bogus SST observation data record is created to a maximum `MAX_NO_ICE` observations. The created data records are written out to the new dataset, after the superob data records.

Before the program completed, the original data records are written out to the new dataset and the sea-ice dataset is closed.

If a fatal error occurs then a message is written out detailing the problem before the program aborts.

6.2 Program I/O

6	OUT	output messages	
10	IN	control namelist	£IN/qisst.superob
19	IN	housekeeping file	£DATAW/qwgl.hkfile
31	IN	data records	£DATAT/datat/qtsstqc.surf
41	OUT	data records	£DATAT/qtsstqc.surf

The daily sea-ice file is read in on unit 15 using an environment variable and the file is opened as read only.
`ICEINSURF=£DATAW/qwgl.daily.ice.`

6.3 Description of operational changes

V1.0 8 Jun 1993 First operational version.
(Program library 1)

V1.1 6 Jul 1993 Minor corrections to CDUPS.
Modest CJ280693 program library 2)

Program library remake are renumbered to be number 1 from next version of program.

V1.3 19 Oct 93 Update program to comply with UM version 3.2
Program library 1

6.4 Current Equations Used.

The observation error, E_o^2 , of a superob may be thought of as the sum of three types of error:

$$E_o^2 = E_{INST}^2 + E_{REP}^2 + E_{DIURNAL}^2 \quad (6.1)$$

where,

E_o^2 = observation error

E_{INST}^2 = instrumental error

E_{REP}^2 = error of representativeness

$E_{DIURNAL}^2$ = error due to inconsistency of reports

($E_{INST}^2 + E_{REP}^2$) will be constant for all buoys and is set as 0.4K in the observation processing stage. $E_{DIURNAL}^2$ is modelled by the equation,

$$E_{DIURNAL}^2 = (1 - \exp(-\sigma)) + 0.7N^{-1/2} \quad (6.2)$$

where,

σ = standard deviation of individual reports

N = number of individual reports

For buoys that have only reported once E_o^2 set to 0.7K.

The current quality control limits set are:

The superob is rejected if any one of the following conditions is met (these limits are set by namelist variables);

standard deviation of reported values exceeds 0.5°C

the range of reported values exceeds 0.5°C

either the latitude or longitude alters by more than 0.5 degrees during the reports.

7. QXOB.ACOBS

This program is the general program used by all observation processing to create an ACOBS file for use in the analysis system. The program is described in UMDP P3 and so only those aspects that apply to SST are described here.

7.1 Format version 1

There is a twenty word header as described in UMDP P3 and the data values contained in the file are;

- 1) latitude of observation
- 2) longitude of observation
- 3) time of observation (in minutes relative to time of obs file)
- 4) not used
- 5) not used
- 6) model observation type
- 7) SST value
- 8) E_o/E_B

(The first six values are standard to all ACOBS files)

The SST ACOB type is 702 and the model observation types were given in section 3.1

7.2 Format version 2

(First used operationally 19 Oct 93)

The format of model dumps is now followed and the data section contains the following information (the first 5 values being common to all ACOBS files)

- 1) latitude of observation
- 2) longitude of observation
- 3) time of observation (in minutes relative to time of obs file)
- 6) model observation type
- 5) identifiers (not yet used)
- 6) SST value
- 7) E_o/E_B

There is only one lookup table for ACOB type 702. The environment variable FILE15 is set to /u/opfc/op2/op/dataw/qwgl.daily.sst

In the new format, data is packed using the CRAY 32 bit method.

8. QXSST.ANALYSIS

This program performs the analysis of SST using the observations in the ACOBS files and creates the daily SST analysis and anomalies from climatology fields.

There are options when the executable is made to create an executable for the global model, the mesoscale model, the limited area model (code in place but not yet tested), the global model but only using observations that lie within mesoscale model or a mesoscale analysis with extra diagnostics. When any executable is run there are options to ignore all observations and therefore create the climatological SST field for any particular date, to exclude a particular observation type(s) from the analysis, to verify the analysis against observations that have not been analysed or to use the satellite bias correction scheme.

The source code consists of the following members of MS12.CJSSTLIB.ANALYSIS. Routines: MAIN (main program), BACK@IN, CLIM@IN, CLIM@INC, COR@FUNC, DIAG@O, GRID@V, FINAL@F, ICE@IN, INC@GRID, INC@OBS, N@FACTOR, OBS@IN, OBS@INT, OBS@INTC, OBS@PRT, OB@DEN, OB@GRID, READHK, READOBS, SST@AC, VERIFY, VERIFY@P, W@FIELDS, READACOB. Comdecks: ACOB, CONST, CSSTCTRL, CTRL, FIELDS, GRID, HEAD, OBS, PARM. The following members of the Unified Model master library are also used: BUFIN1A, BUFOU1A, CHKLOO1A, CLOOKADD, CLOSE1A, C@MDI, C@PI, IOERRO1A, LLTOEQ1A, NEWPAC1A, OPEN1A, P21BIT1A, POSERR1A, PRFIXH1A, PRIFLD1A, PRRFLD1A, READFL1A, READHE1A, TIMER, WRITHE1A, WRITFL1A, T@INT1A, WHENIMDI. Members PR@LOOK and PP@TITL from MS12.CJSSTLIB.ANALYSIS are also used for the same reasons as given for the other programs.

The source is maintained using the CRAY nupdate facility in the same fashion as the other programs. The scripts to process these routines are in MS12.CJSSTLIB.SSTCNTL are are called.
AYCREX - creates an executable from a program library and creates a full source code listing on the CRAY.
AYCRGO - to run an executable
AYCRMD - to create a MODECK
AYCRPL - to create a program library from the source code.
AYCRPLU - to create a new program library by modifying an old library using a MODECK.

Index to Comdecks:

ACOB	- holds variables required for reading old style ACOB file
CONST	- holds constants
CSSTCTRL	- holds variables required for reading/writing ancillary files
CTRL	- holds control information
FIELDS	- holds data fields arrays
GRID	- holds details of the analysis grid
HEAD	- holds headers of ancillary files
OBS	- holds observations
PARM	- holds analysis parameters

Index to program:

```
PROGRAM MAIN
  CALL TIMER
  CALL OPEN (6 times)
  CALL READHK
  CALL ICE_IN
  CALL CLIM_IN
  CALL OBS_IN/READACOB
  CALL OBS_PRT
  CALL BACK_IN
  CALL SST_AC (twice if satellite correction scheme
implemented)
  CALL FINAL_F
  CALL W_FIELDS
  CALL VERIFY
  CALL CLOSE (6 times)
  CALL TIMER
```

(NB. Some of these routines are only called if switches are enabled)

MAIN program

```
NAMelist/DIAG/LOBPRT, LOBPRT_USED, LDIAGG, LDIAGT, LDIAGR, LDIAGO,
LDIAGS, LDIAGZ, FIRST_TIME, USE_NO_OBS, NOBS_DIAG, IROW_DIAGO, ICOL_
DIAGO
```

```
LOBPRT      - LOGICAL - set to T to list out observations read
                in
LOBPRT_USED - LOGICAL - set to T to only list those observations
                actually assimilated
LDIAGG      - LOGICAL - set to T to print out values in a chosen
                area
LDIAGT      - LOGICAL - set to T to obtain statistics for each
                iteration
LDAIGR      - LOGICAL - set to T for monitoring statements
LDIAGS      - LOGICAL - set to T to obtain selected grid point
                values at each iteration
LDIAGO      - LOGICAL - set to T if observation diagnostics are
                required
LDIAGZ      - LOGICAL - set to T to obtain timings
FIRST_TIME  - LOGICAL - set to T to initialise the scheme
USE_NO_OBS  - LOGICAL - set to T if observations are not to be
                assimilated
NOBS_DIAGO  - INTEGER - number of observations for diagnostic
                output
IROW_DIAGO  - INTEGER - first row printed out if LDIAGG
                activated
ICOL_DIAGO  - INTEGER - first column printed out if LDIAGG
                activated
```

(FIRST_TIME=T may be used if the background fields become lost or corrupted, if USE_NO_OBS=T as well then the output field is the climatological field for the given date. IF LDIAGG=T then the area to print is determined by IROW_DIAGO and ICOL_DIAGO. 40 rows and 15 columns (every other column in a span of 30

columns) are printed but the area is truncated if it crosses Greenwich or exceeds the boundaries of a limited area grid. Various fields are listed including increment fields and the SST field after each iteration)

The default values are

LOBPRT=F, LOBPRT_USED=F, LDIAGR=F, LDIAGG=F, LDIAGO=F, LDIAGT=F,
LDIAGS=F, LDIAGZ=T, FIRST_TIME=F, USE_NO_OBS=F, NOBS_DIAG=0,
IROW_DIAG=13, LCOL_DIAG=257

NAMELIST/PARMS/CRIT_DISTANCE, RELAX_TO_CLIM, RELAX_FACTOR,
SCALE_FACTOR, NO_OF_ITERATIONS, OBTYP_LIST, LBIAS_CHK, LEXCLUDE, L
TYPE, LVERIFY, IEXCLUDE_LIST, IVERIFY_TYPE, RVERIFY_LAT, RVERIFY_LO
N, ADD_CLIM, BG_ERR, IOB_FORMAT

CRIT_DISTANCE - REAL ARRAY - maximum radius of influence of
observations

RELAX_TO_CLIM - REAL - Timescale over which to relax to
climatology

RELAX_FACTOR - REAL - relaxation factor

SCALE_FACTOR - REAL ARRAY - scale factors

NO_OF_ITERATIONS - INTEGER - number of iterations

OBTYP_LIST - INTEGER ARRAY - observation types to bias
correct

LBIAS_CHK - LOGICAL - set to T to activate bias correction scheme

LEXCLUDE - LOGICAL - set to T to activate exclude option

LVERIFY - LOGICAL - set to T to activate verify option

LTYPE - LOGICAL - set to T to verify an observation type

IEXCLUDE_LIST - INTEGER ARRAY - observation types to exclude

IVERIFY_TYPE - INTEGER - observation type to be used for
verification if LTYPE=T otherwise it is the interval to use all
observations for verification

RVERIFY_LAT - REAL - latitude resolution for verification
grid

RVERIFY_LON - REAL - longitude resolution for verification
grid

IOB_FORMAT - INTEGER - =1 for old ACOBS format, =2 for new
ACOBS format

CRIT_DISTANCE is an array dimensioned to be 500 and the element
numbers refer to the model observation types. SCALE_FACTOR is
dimensioned to be (500,2), the 2 dimension for the starting and
finishing scale factor values.

If LBIAS_CHK=T then OBTYP_LIST is the observation types that are
to be bias corrected, the first type in the list must be 12.

If LEXCLUDE=T then the observation types listed in IEXCLUDE_LIST
are excluded from the analysis.

If LVERIFY=T then if LTYPE=T the observation type given in
IVERIFY_TYPE is used to verify the analysis. If LTYPE=F then
every IVERIFY_TYPE observation is used for verifying the
analysis. Observations used for verifying the analysis are not
analysed themselves and observations that have been excluded
cannot be verified.

The default values are

NO_OF_ITERATIONS=12, RELAX_TO_CLIM=14.0, RELAX_FACTOR=0.05, LBIAS

_CHK=.FALSE.,LEXCLUDE=.FALSE.,LVERIFY=.FALSE.,LTYPE=.FALSE.,AD
D_CLIM=.TRUE.,RVERIFY_LAT=10.,RVERIFY_LON=10.,BG_ERR=0.8.
Default values for CRIT_DISTANCE are all set to 1000000.0 in
words 13,17,25,25,26,27,28,29,30,32,35,36,301,302 and missing
data for all other words.
Default values for SCALE_FACTOR are all set to 500000.0 and
100000.0 for the same model observation types and missing data
otherwise.

```
READHK - to read hoousekeeping file
ICE_IN - to read in daily seaice analysis file
  CALL READHEAD
  CALL READFLDS
CLIM_IN - to read in climatological SST files and create
          climatology field for the current date
  CALL READHEAD
  CALL READFLDS
  CALL T_INT
IF(IOB_FORMAT.EQ.1)
  CALL OBS_IN - to read in old style ACOBS file
  CALL READOBS
ELSEIF(IOB_FORMAT.EQ.2)
  CALL READACOB - to read new style ACOBS file
  CALL READFLH
  CALL GETDIM
  CALL READACOB1
  CALL READHEAD
  CALL READFLDS
ENDIF
CALL BACK_IN - to read in background SST and error files
  CALL READHEAD
  CALL READFLDS
CALL SST_AC - the control routine for the analysis
              (further details given below)
CALL FINAL_F - to create the final analysis and anomaly fields
CALL W_FIELDS - to write out new SST and error filesd
  CALL WRITHEAD
  CALL WRITFLD
```

SST_AC

This routine controls the analysis of the SST observations.
The precise actions are controlled by the argument IMODE which
is set to 1 for a normal analysis and 2 to invoke the satellite
bias correction scheme. In either mode the actions of the
routine are very similar.

```
CALL CLIM_INC - if IMODE=1 adds climatological increment to
               background field to create first guess field
               - if IMODE=2 copies background field ready for
               satellite bias correction scheme
CALL OBS_INTC - calculate interpolation coefficients to enable
               model fields to be interpolated to observation locations
```

LOOP OVER ITERATIONS

FIRST LOOP OVER OBSERVATIONS

CALL OB_GRID - calculate grid point numbers influenced by observation

CALL COR_FUNC - calculate correlation function

CALL OB_DEN - calculate observation density on model grid

END OF FIRST LOOP OVER OBSERVATIONS

CALL OBS_INT - to interpolate observation density field to observation locations

CALL WHENIMDI

CALL ORDERS

CALL N_FACTOR - to calculate normalisation factors at observation locations

CALL INC_OBS - calculate increments at observation locations

SECOND LOOP OVER OBSERVATIONS

CALL OB_GRID

CALL COR_FUNC

CALL INC_GRID - to calculate increments on model grid

END OF SECOND LOOP OVER OBSERVATIONS

END OF LOOP OVER ITERATIONS

The number of iterations is determined by IMODE. If IMODE=2 only one iteration is performed and if IMODE=1 then the number of iterations is determined by the namelist variable NO_OF_ITERATIONS, default value 12. The scale factor and maximum radius of influence are a function of the observation type.

If IMODE=2 then the background field used each analysis is simply the previous analysis. Separate analyses are performed for *in situ* data and each satellite type, using the first scale factor and the relaxation factor set to 1. The difference of each satellite analysis from the *in situ* analysis is calculated and then the difference field is interpolated to the satellite observation locations and added to the original satellite observations.

If IMODE=1 then the background field is the previous analysis plus the climatological increment, provided that ADD_CLIM=T. The full analysis is then performed, with the corrected satellite observations if the bias correction scheme has been invoked.

There are also facilities to obtain diagnostic information, including the increment field after each location and observation increments each iterations.

8.1 Return codes

If a problem is encountered reading either the housekeeping file or any of the ancillary files then the program will abort.

The program will also abort if the dates in the background SST file and the background error file do not correspond. If the dates of the sea-ice file and the housekeeping file do not match then a warning message is written out and the program continues.

If an old format ACOBS file is being used then there are no return codes and no checks are made in the contents of the file.

If a new format ACOBS file is being used then there are more checks made. Since the file is in basically UM dump format, if an error occurs in reading the file then the program will abort. If the housekeeping date does not correspond with the date in the ACOBS file header the program will abort. Also, checks are made that the contents are SST observations and that 7 data values are in the file, if a check fails then the program will abort.

If the ACOBS file, of either format, is empty of observations then a warning message is written out and the program continues and USE_NO_OBS is automatically set to T.

8.2 Program I/O

```
6  OUT output messages
10 IN  ACOBS file (old format)  £DATAW/qwsstgl.acobs
11 IN  housekeeping file       £DATAW/qwgl.hkfile
12 IN  control namelist       £IN/qisst.nlist
13 IN  parameter namelist     £IN/qisst.parms
40* OUT verification statistics
50* OUT difference fields in satellite bias correction scheme
56* OUT corrections applied to satellite observations
60* OUT observation density fields
```

* Optional, see below for details, and not used operationally.

Portable I/O is used for the ancillary files and the new format ACOBS file. The files are defined by the following environment variables

```
SSTCLIMSURF=£DATAW/qwgl.daily.sst (unit 30)
SSTINSURF=£DATAW/qwgl.daily.sst (unit 31)
ERRINSURF=£DATAW/qwbgerr.glsst (unit 32)
ICEINSURF=£DATAW/qwgl.daily.ice (unit 33)
SSTOUTSURF=£DATAT/qtsst.new.anal (unit 20)
ERROUTSURF=£DATAT/qtsst.new.anal (unit 21)
ACOBSIN=£DATAW/qwsstgl.acobs (unit 10)
```

8.3 Options when building executable

There are sections of code enclosed within *DEF blocks which may be selected when the executable is built using the nupdate - d option.

GLOBAL - for the global analysis on a 288x217 grid

LAM - for an analysis on the limited area grid

MUM - for an analysis on the mesoscale model grid

(one of these must be chosen)

TEST - to write out fields formed in bias correction scheme. The files written out are the observation density fields, first field on unit 50, the next on 51 etc, the calculated difference fields, the first on unit 60, the next on unit 61 etc and the corrections applied to the satellite observations on unit 56.

GLOBMUM - to run the global analysis but only using those observations that lie within mesoscale model area (for checking that mesoscale model analysis is working correctly)

TESTMUM - prints out a 'Z' map of observation distributions in mesoscale analysis

DIAG - for extra diagnostic information. (Can only be used if autotasking is not used)

8.4 Description of operational changes.

V1.0 26 Feb 91. First operational version.
(includes MODEST CJ190291)

Sets headers of fields for addition to HDS fieldsfiles.
Modest CJ130591

V1.1 30 Jul 91. Include code to check for a rerun.
Modset CJ110791

V1.2 13 Aug 91. No coding changes but all SATOB data removed from analysis due to effects from Mount Pinatubo.

V2.0 24 Jan 92. Satellite bias correction scheme mark 1 (Bell 1991). (Never switched on operationally)
Modest CJ240192

V2.1 20 Mar 92. Option include to exclude observation types from analysis and to exclude observations from being analysed so that they can be used to verify the analysis. Tidy up sections of code.
Modset CJ200392

V2.2 6 May 92. Use dynamic allocation more effectively.
Modest CJ060592

V3.0 24 Sep 92. Satellite bias correction scheme mark 2 (Jones 1993), (code in place but not implemented) and remove code for Bell's scheme. Also introduce code to create bogus observations under sea-ice cover but not implemented.
Modset CJ240992 Program library 6

01 Dec 92. Introduce code for limited area and mesoscale model grids with *DEF blocks. Move hardwiring of initial background error value to a namelist variable. Switch on satellite bias correction code mark 2.
Modset CJ191192 Program library 7

V3.1 12 Jan 93. Change equation for increment to climatology.
Modset CJ070193 Program library 8

Remove calls to PR_LOOK and PR_RFLD from ICE_IN.
Modest CJ300193.

V3.2 23 Feb 93. Update all UM routines to vn3.0 of UM and change to packing indicators. Use portable I/O code for reading/writing ancillary fields.
Modset CJ080293 Program library 9

Change to OBS_IN to correct error in assigning observations in mesoscale analysis.
Modest CJ250293 Program library a

V3.3 22 Apr 93. Introduce code to cope with new superob observation types. Remove code to create bogus observations under sea-ice cover. Introduce code for observation type dependent correlation scales. Scale correlation function.
Modest CJ220493 Program library b

Program library remade are renumbered to be number 1 from next version of program.

V3.4 19 Oct 93. Update program to comply with UM version 3.2.
Program library 1

8.5 Current Equations used.

The first step is to increment the background field, simply the previous analysis, towards the climatological field for the current date. The reason for this is to ensure that there is temporal variation in areas void of observations. The increment, I_c is calculated as follows,

$$I_c = (T_{AN}^{t-1})/d + (T_c^t - T_c^{t-1}) \quad (8.1)$$

where,

T_{AN} is the anomaly from climatology field

T_c is the climatology field

d is the timescale over which the field relaxes to climatology

and superscript t is the current day

The current operational values of d and all other constants will be given at the end of the section.

The current climatological field is obtained by temporal interpolation between adjacent months. The current one used is

the third historical SST climatology created by UKMO, MOHSST3, based on ship observations for the years 1951 to 1980, Folland and Parker (1990).

The analysis uses the method of analysis correction (AC) and is described fully in Lorenc et al (1991).

The basic iterative equation is

$$T[u+1] = T[u] + WQ'_i (T_o - K(T[u])) \quad (8.2)$$

where,

T is the temperature field

W is a matrix of weighting factors

Q'_i is a matrix of normalisation factors calculated at observation locations

T_o is the SST observation

K is the interpolation function

u is the iteration number

The following approximation of equation 8.2 which relates increments at grid points to increments at observation locations is used,

$$\Delta T_k = \lambda \sum_i \mu_{ki} Q'_i \Delta T_i \quad (8.3)$$

The summation is over all observations, i, which influence a given grid point k.

Q'_i is a normalization factor which can be written as,

$$Q'_i = \{ \epsilon_1^2 (1 + K_i(D_k)) \}^{-1} \quad (8.4)$$

where ϵ_1^2 is the ratio of observational to first guess error (E_o^2/E_b^2), where E_o^2 and E_b^2 are as defined in the quality control section. The term D_k is defined as the data density at grid point k. It is calculated first at grid points, to avoid expensive searching of observations around other observations, and interpolated to observation locations by means of the bi-linear interpolation operator, K_i . The expression for D_k is,

$$D_k = \sum_i \mu_{ki} \epsilon_i^{-2} \quad (8.5)$$

μ_{ki} is a second-order auto regressive correlation function of the great circle distance between observation i and grid point k. It is calculated as

$$\mu_{ki} = (1 + r_{ki}/S) \exp(-r_{ki}/S) \quad (8.6)$$

where S is a scale factor which can be varied for each observation type if desired.

At version 3.3 μ_{ki} was replaced in the calculations by μ'_{ki}

$$\mu'_{ki} = (\mu_{ki} - \mu_c) / (1 - \mu_c) \quad (8.7)$$

where μ_c is the value of μ_{ki} at the maximum influence of radius of the observation. This scaling ensures that function equals zero at the maximum influence of radius of the observation.

The great circle distance, r_{ki} , is calculated as follows. If the two points have lat-long coordinates of (ϕ_1, λ_1) and (ϕ_2, λ_2) then

$$r_{ki} = a\psi \quad (8.8)$$

where a is the earth's radius and ψ is given by

$$\cos(\psi) = \cos(\phi_1)\cos(\phi_2)\cos(\lambda_2-\lambda_1) + \sin(\phi_1)\sin(\phi_2) \quad (8.9)$$

The radius of rows within the influence area is calculated using

$$N_{row} = r_c / (111000\phi_r) + 1 \quad (8.10)$$

where,

N_{row} is the number of rows within radius
 r_c is the radius of influence (m)
 ϕ_r is the latitudinal resolution of the grid (degrees)
and the 111000 is the degrees to m conversion factor.

The number of columns in range along each row is calculated as

$$\lambda_2 - \lambda_1 = \cos^{-1}\{(\cos(r_c/a) - \sin^2\phi) / \cos^2\phi\} \quad (8.11)$$

$$N_{col} = (\lambda_2 - \lambda_1) / \lambda_r + 1 \quad (8.12)$$

where,

N_{col} is the number of columns within radius of current row
 λ_r is the longitudinal resolution of the grid (degrees)
 ϕ is the latitude of the current row (degrees)

The number of columns in the range is restricted near the poles for computational reasons.

The following table gives the values of the various parameters used in the scheme.

Number of iterations, $N = 12$
Scale factor at $u=1$, $S_1 = 500000m$
Scale factor at $u=N$, $S_N = 100000m$
Scale factor at $u=n$, $S_n = S_1 - ((S_1 - S_N)(n-1)) / (N-1)$
Relaxation factor, $\lambda = 0.05$
Radius of influence of observation, $r_c = 1 \times 10^6$ m
Time scale to relax to climatology, $d=28$

Although the facility exists to specify S_1 , S_N and r_c separately for each observation type, all observation types currently use the same values as given above.

In the satellite bias correction scheme and analysis is performed for *in situ* data and each satellite type. In this analysis there is no increment to climatology, only 1 iteration is performed, the value of S is set to be the value of S_1 and the relaxation factor λ is set to 1.

9. QXSSTTSTAR

This program runs after the analysis to create a combined SST/model T* data field which is then transferred to the HDS for addition to the HDS fieldsfile.

The source code consists of the following members of MS12.CJSSTLIB.ANALYSIS. Main program SSTTSTAR and comdecks; CSSTCTRL, HEAD, GRID, and FIELDS as well as PR@LOOK and PP@TITL as already explained. The following members of the Unified Model master library are also used: CLOOKADD, READHE1A, READFL1A, PRFIXH1A, PRIFLD1A, PRRFLD1A, POSERR1A, CHKLOO1A, IOERRO1A, WRITFL1A, WRITHE1A, P21BIT1A, NEWPAC1A, OPEN1A, CLOSE1A, BUFIN1A and BUFOU1A.

The source is maintained using the CRAY nupdate facility. Changes to the source code are made by creating a MODECK which is then used to amend a program library from which an executable is made

There are several scripts on MS12.CJSSTLIB.SSTCNTL which are used to process these routines.

STCREX - creates an executable and creates a full source listing
STCREXP - to run the program using a modeck
STCRGO - runs an executable
STCRPL - creates a program library from the source code
STCRPLU - to create a new program library by modifying an old library using a modeck

Index to Comdecks:

CSSTCTRL - holds variables for reading in ancillary field datasets
HEAD - holds headers of ancillary field datasets
FIELDS - holds SST fields
GRID - holds grid details

Index to program:

PROGRAM SSTTSTAR
CALL OPEN
CALL READHEAD
CALL READFLDS
CALL FFREAD
CALL WRITHEAD
CALL WRITFLDS
CALL CLOSE

9.1 Description of Program.

The program simply reads in the SST analysis, SST anomaly from climatology field and model T* fields, overwrites the land and sea-ice points in the SST analysis field with T* values and writes the SST analysis and anomaly fields to a dataset ready for transfer to the HDS.

The header of the SST daily dataset is read using READHEAD followed by the SST analysis field and SST anomaly from climatology field, read by separate calls to READFLDS1. The model T* field is read using the general fieldsfile access routine, FFREAD, The required field is on projection 800, and has Met O (CF) field code 18 on level 9999.

A loop is made around all model grid points and at land and sea-ice points, $SST \leq 271.35K$, the model T value is written to the SST analysis array. Finally, the headers of the dataset are written out using WRITHEAD followed by the amended fields in turn using WRITFLDS.

9.2 Return codes and action of program.

If any of the routines called by the main program return a return code not equal to zero then a message is printed out and the program aborts. If this is the case then no fields will be added to the HDS fieldsfile and no contour maps will be plotted.

Program I/O

Unit 6 OUT output messages

Unit 10 IN SST analysis field £DATAW/qtgl.daily.sst (SSTINSURF)

Unit 11 IN model PP fields £DATAW5/qtgl.pp0

Unit 20 OUT SST/T* fields £DATAT/qtgl.daily.sst (SSTTSOUTSURF)

The SST and SST/T* ancillary fields are accessed by the environment variables given in brackets.

10 QXSST.RALMRG and STRFLD

10.1 STRFLD (not yet operational)

Although not strictly part of the SST suite, program STRFLD is important as it extracts and stores background fields of wind mixing energy and latent, sensible, solar and IR heat fluxes ready for use by program RALMRG. It runs in the QG00, QG06, QG12 and QG18 model runs within the MDB script.

The source code consists of the following members of MCF5.DDSSTLIB.SRCE; STRFLD (main program), COMDECK, RDIFFHD, FFVSET, READHC, READHK, RDIFFLE and WRTFLD.

The script ANCILTST in MCF5.DDSSTLIB.SSTCNTL will create a new executable and run the program.

Index to Comdesck:

CTRL - holds control variables and housekeeping information
CONS - holds constants
FFILE - holds fieldsfiles details
FFHED - holds fieldsfile header details
NAMFFV - holds namelist details for fieldsfile
NAMIN - holds namelist details for control file

Index to Program:

PROGRAM STRFLD
CALL RDIFFHD
CALL FFVSET
CALL READHC
CALL READHK
CALL RDIFFLE
CALL WRTFLD

STRFLD (main program)

NAMELIST /FFINS/ITYPESF,JPROJS
NAMELIST /LIMS/BNORTH,BSOUTH,BEAST,BWEST
NAMELIST /CTRL/IRUNTP,ICUT,ITWAF,ITWBE,IMODID,ODIAGS,MODTYP,
MODVER

ITYPESF - INTEGER - array of field types for surface fluxes
JPROJS - INTEGER - projection numbers for surface flux fields
BNORTH - REAL - northern boundary of model area
BSOUTH - REAL - southern boundary of model area
BEAST - REAL - eastern boundary of model area
BWEST - REAL - western boundary of model area
IRUNTP - INTEGER - model run type
ICUT - INTEGER - cut off time (HHMM)
ITWAF - INTEGER - time window boundary (hours after analysis time)
ITWBE - INTEGER - time window boundary (hours before analysis time)
IMODID - INTEGER - model identifier

ODIAGS - LOGICAL - T if diagnostics are to be printed
 MODTYP - INTEGER - model observation type numbers
 MODVER - INTEGER - model version numbers

 RDIFFHD - to read fieldsfile header information
 FFVSET - to set variables ready for FFREAD call
 READHC - to read housekeeping and control files
 CALL READHK - to read housekeeping dataset
 RDIFFLE - to read required fields from fieldsfile
 WRNFLD - to write out required fields to a storage dataset

10.1.1 Description of Program:

Fields of wind mixing energy and surface heat fluxes are read using FFREAD and the fields are written out to a storage dataset. There are 5 such storage datasets (see I/O dataset details) and the dataset updated will correspond to the model run time. During the QG00 run the contents of the QG00 storage dataset are copied to the 'old' QG00 dataset.

10.1.2 Return codes and action of program:

IRCFHD - return code from RDIFFHD
 =1 error reading fixed length header
 =2 integer constant array not big enough
 =3 error reading integer constants
 =4 real constant array not big enough
 =5 error reading real constants
 =6 level dependent constant array not big enough
 =7 error reading level dependent constants
 IOSNME - return code from FFVSET
 =1 problem reading FFINS namelist
 IRCFHC - return code from READHC
 =1 problem opening control file
 =2 problem reading LIMS namelist
 =3 problem reading CTRL namelist
 =4 problem closing control file
 =7 problem opening housekeeping file
 =8 problem reading housekeeping file
 =9 problem closing housekeeping file
 IRCFFLE - return code from RDIFFLE
 =1 no field read

If no fields have been successfully read then a missing data indicator only is written to the storage dataset.

10.1.3 Program I/O:

6 OUT output messages
 18 IN namelist input
 19 IN housekeeping file
 20 IN control file
 25 IN important message file
 31 IN background fieldsfile
 32 OUT stored fields

10.2 QXSST.RALMRG (The operational version does not yet include the fields obtained from STRFLD.)

After the analysis has been performed, records of ATSR data containing relevant model data are created which are then transferred to the HDS for archiving in the MDB. The program that creates these records is called RALMRG.

The source code consists of the following: members RALMRG (main program), FILES@IO, READDAT, INTERP2, DATETIME, INTTIME, SELEVN, HKMATCH and CENTDAY from CFOBS.BLSSTLIB.RALUAT, members ANAL@IN, DATA@PRT, INTERP, READHK, UDDS, CSSTCTRL, FIELDS, GRID and HEAD from CFOBS.BLSSTLIB.OPD and the following members of the Unified Model master library: CLOOKADD, READFLD1, READHED1, PR@FIXH1, PR@RFLD1, POSERRO1, CHKLOOK1, IOERROR1, C@PI and C@MDI. Again, members PP@TITL and PR@LOOK from MS12.CJSSTLIB.ANALYSIS are also used in common with many other programs described in this document.

There are several scripts on CFOBS.BLSSTLIB.SCRIPTS which are used to process these routines.
MGCRESX - creates an executable from a program library and creates a full source listing on the CRAY.
MGCRCGO - runs an executable.
MGCRCPL - creates a program library from the source code
MGCRCPLU - to create a new program library by modifying an old library using a modeck.

Index to Comdecks:

CSSTCTRL - holds variables for reading in ancillary fields datasets
CTRL - holds control variables and housekeeping information
FIELD - holds analysed SST field
FILES_IO - holds unit numbers for I/O
GRID - holds grid details
HEAD - holds grid details

Index to program:

PROGRAM RALMRG
CALL READHK
CALL ANAL_IN
CALL ANCIL_IN
CALL TIME
CALL INTIME
CALL INTERP2
CALL SELEVN
CALL READDAT
CALL UDDS
CALL INTERP

RALMRG (main program)
NAMELIST/INPUT/LDIAGR, LDATAP, LMDBP, MAX_PRT_DAT, MAX_PRT_ARR
LDIAGR - LOGICAL - set to T to monitor passage through routines
LDATAP - LOGICAL - set to T to print out input data records
LMDBP - LOGICAL - set to T to print out MDB records
MAX_PRT_DAT - INTEGER - maximum number of data records to print out
MAX_PRT_ARR - INTEGER - maximum number of MDB records to print out

READHK - to read in housekeeping file
ANAL_IN - to read in analysed SST field
CALL READHEAD
CALL READFLDS
ANCIL_IN - to read in stored fields created by STRFLD
CALL HKMATCH - to check housekeeping times on stored fields are correct
READDAT - to read in data records in blocks of 100 observations
UDDS - to unpack data descriptor section
DATETIME - to convert date/time to year, month, day, hour and minute
INTTIME - to establish the validity times of the two surface flux fields required for interpolation
INTERP2 - to interpolate the analysed heat flux fields to the report location
SELEVN - to calculate solar elevation from date/time
CALL CENTDAY - to calculate the century day (ie days since 1/1/1900)
INTERP - to interpolate the model analysed SST field to the report location

10.2.1 Description of program:

The control namelist, housekeeping file and analysed SST field are all read in and a message is written out. Subroutine READDAT is then called which reads in observations one at a time but creates blocks of 100 observations for processing. The DDS of the first observation is unpacked, which gives the pointers to the required information in the data record, and the header of the MDB transfer data is formed. The header is written out to the transfer dataset using the CRAY routine, WRITIBM (NOTE that header is currently not used).

Each ATSR observation is processed and the MDB array for subsequent transfer to the HDS is created. The analysed SST value is interpolated to the observation location using INTERP and the analysed SST value at the observation location is used to determine the surface type. If the observation is unusable, ie missing lat/long/sst/over land then the value of the analysed SST value and the surface type are set to missing data. The quality control flags are checked to create the UKMO quality indicator flag in the MDB record. The blocks of observations are then written out to the transfer dataset, again using WRITIBM, and the process is repeated until all observations have been processed. The format of the output

dataset is element 1 for all observations, element 2 for all observations etc.

10.2.2 Return codes and action of program:

```
IR_READ_HK = 1 problem opening housekeeping file
            = 2 problem reading housekeeping file
            = 3 problem closing housekeeping file
if IR_READ_HK $ 0 then the program aborts
IR_ANAL_HK = 1 problem reading analysed SST field, program
aborts.
IR_READDAT = -1 more data to process
            = 1 all data processed
            = 2 actual length read in is different to
expected length, program aborts
```

Before the program aborts due to one of the return codes above, a header of zeros is written out to the transfer data indicating to the HDS program that there is no data to merge.

10.2.3 Program I/O

```
Unit 6 OUT output messages
Unit 10 IN SST analysis field      £DATAW/qwgl.daily.sst
Unit 11 IN housekeeping file      £DATAW/qwgl.hkfile
Unit 12 IN control namelist       £IN/qisstral.nlist
Unit 15 IN ATSR data records      £DATAT/qtsstqc.atsr
Unit 20 OUT ATSR records for transfer £DATAT/qtsstral.merge
Unit ?? IN stored fields from STRFLD
```

The dataset containing the records for merging with the MDB, qtsstral.merge, is transferred to the HDS into a dataset called COP.RALUAT.MERGE. The merging is performed by a Met O (CC) owned program, ERSMERGE in job QG06AOSG.

10.2.4 Format of Transfer Dataset

The dataset containing the data ready for merging with the MDB has the following format.

A real header of 3 words, consisting of zeros if no further data is available (creation program failed) or the following if additional data is available. (Currently the header is not used)

RHEAD(1) = 1 (number of descriptors/data to follow)
RHEAD(2) = descriptor = 1048
RHEAD(3) = age of background field (hours)

The header is followed by the data for the observations in blocks of 100 observations. For each observation, the following will be supplied, if a parameter is not available then it will be set to a missing data indicator (-32768.0).
(Note that this is now a non-standard MDI)

Latitude (for checking against the data extracted from MDB)
10m wind
Latent heat flux
Sensible heat flux
Solar heat flux
IR heat flux
UKMO data quality indicator (see note a)
Climatological SST
Surface type (see note b)
SST analysis value
SST background value

Notes:

a: UKMO data quality indicator, bits set are
1 not used in analysis (final flag, includes all unusable reports)
2 buddy check failed
3 buddy check performed
4 background check failed
5 background check performed
6 climatological check failed
7 climatological check performed
8 CFO rejection
9 Priori rejection (if failed RAL checks)

b: Surface type indicator

1 = sea (excluding sea ice)
4 = land (precise meaning not defined)
6 = sea -ice (ie analysed SST \leq 271.35K)

An indicator will be set in the MDB denoting the climatological dataset used.

1 = MOHSST3
2 = MOHSST4

11 ADDITION OF FIELD TO FIELDSFILE

The SST/T* file is converted to a HDS sequential PP file by using the program FIELDCOS. The data is copied to the HDS dataset COP.LINKSST. The data is added to the update fieldsfile, COP.FFCLGU, by job QG06AFFS.

12 MAP PLOTTING

The only 'PRODUCT' job in the SST suite is job QG06ACCS which plots contour maps of the SST field and SST anomaly field on Calcomp which is then used in CFO. Two maps of each field are plotted, a global map and a map for the North Atlantic, including the Caribbean.

Program SSTPLOT
Source: COP.PRODUCT.SOURCE(SSTPLOT)
Job to create load module: COP.PRODUCT.BUILD(SSTPLOT)
Loadmodule: COP.PRODUCT.LOADLIB(SSTPLOT)

Index to program:

PROGRAM MAIN

Initialise common blocks
Open Calcomp and Fieldsfile
Read in Namelist
Read in land sea mask
Read in fields (CALL SEARCH)
Convert analysis field from K to C
Set up grid on which charts are to plotted
Interpolate data to new grid
Plot global maps
Plot North Atlantic Maps
Calculate statistics of fields
Close Calcomp and fields file
END

NAMELIST/IN/LANDM

LOGICAL LANDM ! =T then plot analysis field masking out land points

! =F Combined SST/T field is plotted

(T values are always plotted over sea-ice areas)

Operationally, LANDM is set as .TRUE..

If LANDM is set to .TRUE. then the model land sea mask is read into an INTEGER array, JJMASK(288*217) using a formatted read of (200I1,88I1), it is not possible to write (288I1) under the current compiler. The land sea mask is kept in a dataset called COP.SURFACE.GL. Missing data indicators are then inserted in the analysis field at all land points.

The global maps are plotted on a regular 2° grid to reduce plotting time, little detail is lost in the interpolation to this coarse grid. The North Atlantic maps are plotted on a polarstereographic projection.

The analysis maps have stippled shading indicating sub-zero areas, which in effect show the areas covered with sea-ice. Light line shading is used to show areas warmer than 27.5 C and a darker line shading is used to show areas warmer than 30 C.

The anomaly maps have unequally spaced contours and stippled shading is used for negative anomalies and line shading used for positive anomalies. Light shading indicates anomalies between 0.1 C and 2.0 C and dark shading indicates anomalies greater than 2.0 C. The zero contour is not drawn.

13 OPERATIONAL DETAILS

QG06AOEX

QG06AOEX performs the extraction of SST reports from the SDB. The first step of the job extracts observations for the atmospheric model and is of no concern here. Then there follows two steps which perform the extraction of SST observations, SSTEXT and RALUAT which execute programs of the same name.

The relevant JCL is:

```
//* IN NET QG06N0: SUBMITTED BY QG06NET0: RELEASED BY QG06AOIN
//* -- ATMOSPHERIC EXTRACTION STEP -----
//*
//SSTEXT EXEC PGM=SSTEXT,COND=EVEN
//STEPLIB DD DSN=COP.CORE.LOADLIB,DISP=SHR
//FT06F001 DD SYSOUT=*
//FT10F001 DD DSN=COP.SSTDRIFT.ELEMENTS,DISP=SHR
//FT19F001 DD DSN=COP.IBM.HKGL,DISP=SHR
//FT20F001 DD DSN=COP.CORE.CARDLIB(SSTANAL),DISP=SHR
//FT25F001 DD DSN=COP.SSTANAL.MESSAGE,DISP=OLD
//FT30F001 DD DSN=COP.SSTANAL.SURFACE,DISP=OLD
//FT31F001 DD DSN=COP.SSTANAL.SATOBS,DISP=OLD
//FT32F001 DD DSN=COP.SSTANAL.TRACKOBS,DISP=OLD
//HOLDREC DD DSN=SDB.DATE,DISP=SHR
//*
//RALUAT EXEC PGM=RALUAT,COND=EVEN
//STEPLIB DD DSN=COP.CORE.LOADLIB,DISP=SHR
//FT06F001 DD SYSOUT=*
//FT10F001 DD DSN=COP.RALUAT.ELEMENTS,DISP=SHR
//FT16F001 DD DSN=COP.RALUAT.EXTTIME,DISP=SHR
//FT19F001 DD DSN=COP.IBM.HKGL,DISP=SHR
//FT20F001 DD DSN=COP.CORE.CARDLIB(RALUAT),DISP=SHR
//FT25F001 DD DSN=COP.SSTANAL.MESSAGE,DISP=MOD
//FT30F001 DD DSN=COP.RALUAT.OBSFILE,DISP=OLD
```

SURF

All the CRAY SST programs are executed by script SURF. The actions of script SURF are listed below.

```
ICEDAY      - execute qxice.daily (described in UMDP 71)
              - copy character map to HDS
SSTOBS      - copy observation datasets from HDS
              - copy COP.SSTANAL.SURFACE
              - copy COP.SSTANAL.SATOBS
              - copy COP.SSTANAL.TRACKOBS
              - copy COP.RALUAT.OBSFILE
SSTDAT      - execute qxsstdatr to create data records
SSTQC       - execute qxsstqc.pgm to quality control the data
SUPEROB     - execute qxsst.superob to create superobs of
              drifters and buoys and to create bogus SST
              observations.
SSTAC       - execute qxsstac.pgm to create ACOBS file (not for
              much longer)
SSTANAL     - execute qxsst.analysis to analyse SST
SSTOPD      - execute qxsstopd to create SSTOPD records
              - copy data to COP.SSTOPD.NONDAY
              - copy statistics to COP.SSTOPD.STATS
SSTTSTAR    - execute qxssttstar to create fields for HDS
fieldsfile
FIELDCOS    - execute qxfieldcos to convert data into pp format
              - copy data to COP.LINKSST
RALMRG      - execute qxsst.ralmrgr to create records for MDB
              archiving
              - copy data to COP.RALUAT.MERGE
```

Each individual task, except for a couple of exceptions, is tested for an error and should there be an error then the script terminates at that point. The scripts getibm and putibm are used for transfers to and from the HDS.

QG06AFFS

The SST data transferred from the CRAY into dataset COP.LINKSST is in sequential pp format. Job QG06AFFS adds this dataset onto the update fieldsfile, COP.FFCLGU. The JCL is:

```
//* IN NET QG06NA: SUBMITTED BY QG06NETA: RELEASED BY QG06AFFA
//*
// EXEC CFDAFF, LINK=LINKSST, FF=FLCLGU, CARDS=FFGL0,
// BLOCK=FFBLOCKW
      CFDAFF is the general procedure for adding data to
      fieldsfiles and is on COP.CORE.PROCLIB.
```

QG06ACCS

After the data has been added to the fieldsfile, then contour maps are plotted on Calcomp for use in CFO. This task is performed by job QG06ACCS, a 'PRODUCT' task in the SST suite. The JCL is:

```
//* IN NET QG06PS: SUBMITTED BY QG06NEPS: RELEASED BY NONE
//*
//PLOT EXEC CFPP2G,GOPGM=SSTPLOT,LOADLIB='COP.PRODUCT.LOADLIB'
//FT05F001 DD DSN=COP.PRODUCT.CARDLIB(SSTPLOT),DISP=SHR
//FT08F001 DD SYSOUT=7,DCB=(LRECL=80,BLKSIZE=1680,RECFM=FB)
//FT10F001 DD DSN=COP.FFCLGU,DISP=SHR
//FT20F001 DD DSN=COP.SURFACE.GL,DISP=SHR
//
```

QG06AOSG

Job QG06AOSG merges the model fields which correspond to the RALUAT observations back into the MDB for permanent archiving. The JCL is:

```
//* IN NET QG06NA: SUBMITTED BY QG06NETA: RELEASED BY QG06AFPV
//*
// EXEC PGM=ERSMERGE
//STEPLIB DD DSN=COP.CORE.LAODLIB,DISP=SHR
//FT01F001 DD DSN=COP.RALUAT.TABLE,DISP=SHR
//FT02F001 DD DSN=COP.RALUAT.MERGE,DISP=SHR
//FT04F001 DD DSN=COP.RALUAT.EXTTIME,DISP=SHR
//FT10F001 DD DSN=SDB.UATELEM,DISP=SHR
//FT47F001 DD DSN=SDB.ERSUATCF,DISP=SHR
//
```

13.1 JOB RUNNING ORDER

The following is a diagram showing the approximate running order of all jobs involved in the SST suite, including those jobs which are used to submit and release the SST jobs.

13.2 DATASET DETAILS

The following table gives details on the datasets used in the SST suite.

13.2.1 HDS Datasets (all have organisation of PS).

Name	VOLUME	RECFM	LRECL	BLKSIZE	TRACKS
COP.SSTANAL.MESSAGE	OPR004	FB	150	4500	1
COP.SSTANAL.SATOBS	OPR004	VBS	32768	23476	45
COP.SSTANAL.SURFACE	OPR004	VBS	32768	23476	45
COP.SSTANAL.TRACKOBS	OPR004	VBS	32768	23476	1
COP.SSTDRIIFT.CODES	OPR004	FBA	80	4240	1
COP.SSTDRIIFT.ELEMENTS	OPR004	FBA	800	32000	1
COP.RALUAT.CODES	OPR004	FBA	80	4240	1
COP.RALUAT.ELEMENTS	OPR004	FBA	800	32000	1
COP.RALUAT.EXTTIME	OPR004	FB	200	200	1
COP.RALUAT.MERGE	OPR004	VBS	32768	23476	75
COP.RALUAT.OBSFILE	OPR004	VBS	32768	23476	210
COP.RALUAT.TABLE	OPR004	FB	80	4240	1

13.2.2 CRAY datasets (sizes are given in sectors)

i) Permanent read only files - /u/opfc/op1/op/datar

£DATAR/qrgl.clim.sst	735
£DATAR/qrsstat.slist	1
£DATAR/qrsstsb.slist	1
£DATAR/qrsstsf.slist	13
£DATAR/qrssttr.slist	1

ii) Permanent read/write files - /u/opfc/op2/op/dataw

£DATAW/qwbgerr.glsst	62
£DATAW/qwgl.daily.sst	123
£DATAW/qwsstgl.acobs	(variable in size but generally about 350)

iii) Input cards - /u/opfc/op1/op/in

£IN/qisst.superob	1
£IN/qisstac.nlist	1
£IN/qisstayl.nlist	1
£IN/qisstayl.parms	1
£IN/qisstdat.nlist	1
£IN/qisstral.nlist	1
£IN/qisstt.nlist	1

iv) Executables - /u/opfc/op1/op/libs

£LIBS/qxsst.analysis	420
£LIBS/qxsst.superob	276
£LIBS/qxsst.ralmrg	280
£LIBS/qxsstac.pgm	269
£LIBS/qxsstdatr	323
£LIBS/qxsstqc.pgm	246
£LIBS/qxssttstar	208

```

v) Source - /u/opfc/op1/op/source
£SOURCE/qcsst.analysis      146
£SOURCE/qcsst.superob       94
£SOURCE/qcsst.ralmrq        83
£SOURCE/qxsstqc.pgm         121
£SOURCE/qcsstdatr           111
£SOURCE/qcsstqc.pgm         38
£SOURCE/qcssttstar          90

```

13.3 CARDS USED IN PROGRAMS

The following are the current operational CARD input to the programs.

HDS Programs.

PROGRAM SSTEXT

```

&USER
  LDIAGR=.FALSE., LLIST=.FALSE., MAXLIST=1,
  P_SHIP=.TRUE., P_BUOY=.TRUE., P_BATHY=.TRUE., P_DRIFT=.TRUE.,
  P_SATOB=.TRUE., P_TRACK=.TRUE.
&END

```

PROGRAM RALUAT

```

&USER
  LDIAGR=.FALSE., LTEST=.FALSE., LLIST=.FALSE., MAXLIST=1
&END

```

PROGRAM SSTPLOT

```

&IN
  LANDM=.TRUE.
&END

```

CRAY Programs

PROGRAM QXSSTAC.PGM

```

&ACBSF
  AC_OBS_TYPES=702, L_ATMOS=F, L_SST=T, LHK=T,
  OBS_FORMAT=2, NLEVELS=1, LDIAG=F, LDIAG2=F, LPACK=T
&END

```

PROGRAM QXSST.ANALYSIS

```

&DIAG
  LOBPRT=F, LDIAGO=F, LDIAGR=F, LDIAGT=F, LDIAGG=F, LDIAGS=F,
  FIRST_TIME=F, USE_NO_OBS=F, NOBS_DIAG=1
&END
&PARMS
  NO_OF_ITERATIONS=12, RELAX_TO_CLIM=28.0, RELAX_FACTOR=0.05,
  OBTYPE_LIST=12, 27, ADD_CLIM=T, LBIAS_CHK=T, IEXCLUDE=T,
  LTYPE=F, LVERIFY=F, IEXCLUDE_LIST=17, 13, 32, 28, 30, 29, 301, 302,
  IVERIFY_TYPE=99
&END

```

```

PROGRAM QXSSTDATR
&INPUT
  LDIAGR=F, LDATAP=F, LSDBP=F, LDIAGS=F, P_SURFACE=T, P_SATOB=T,
  P_TRACKOB=T, P_ATSR=T, MAX_PRT_DAT=1, MAX_PRT_SDB=1,
  ISST_VER_NO=1
&END

PROGRAM QXSST.SUPEROB
&CTRL
  LDIAG=F, LDATAP1=F, LDATAP2=F, FREQ_ICE=10, ICE_ERR=0.1,
  QC_SD=0.5, QC_RANGE=1.0, QC_POS=0.5, EBASE=0.7
&end

PROGRAM QXSST.RALMRG
&INPUT
  LDIAGR=F, LDATAP=F, LMDBP=F,
  MAX_PRT_DAT=1, MAX_PRT_ARR=1
&END

PROGRAM QXFIELDCOS
&PACK UNPACK=.FALSE. &END
&TYPE OPER=.TRUE. &END

```