SUBJECT: **TC History Ingestion Technical Note**

PREPARED BY: M.J. Graham

DOCUMENT No: **SPIRE-ICS-DOC-002314**

ISSUE: 0.2 **Date:** **18th March 2003**

Change Record

ISSUE	DATE	
0.1	7 September 2001	Original version
0.2	17 February 2003	Updated as part of release of code into HCSS

Table of Contents

1. INTRODUCTION

1.1 Purpose and scope of the document

This technical note details the design of the TC history ingestion process and the decisions made regarding its design.

This technical note addresses the TC history ingestion process design for the ILT mission phase (version 0.1.x of the HCSS). This uses a TCP/IP-based interface to request data from SCOS-2000; however, later versions will make use of a CORBA-based interface (to be implemented). This version of this technical note does not describe this.

1.2 References

- **[RD1]** FIRST Ground Segment Design Description (FIRST/FSC/DOC/0146, Issue 1.0, Revision 0, 3rd Nov 2000)
- **[RD2]** FIRST Ground Segment Design Description (FIRST/FSC/DOC/0146, Issue 1.0, Revision 0, 3rd Nov 2000)
- **[RD3]** TC History File ICD
- **[RD4]** SCOS-2000 Command History Data Provision Services ICD (S2K-MCS-ICD-0008-TOS-GCI, Issue 1.2, 7th Aug 2000)
- **[RD5]** Technical Note on Observation and Building Block Identifiers for Herschel (ICC/2001-001, Issue 1.0, 6th June 2001))
- **[RD6]** Trading-off Technical solutions for relating TC History and HCSS commanding (FSCDT/TN/014, Issue 0, Revision 1, 14th Feb 2001)
- **[RD7]** HCSS User Requirements Document (FIRST/FSC/DOC/0115, Issue 2.0, xx May 2001)
- **[RD8]** FIRST/Planck Packet Structure Interface Control Document (SCI-OT-ICD-7527, Issue 1.0, Revision 0, 1st September 2000)
- **[RD9]** Analysis of ILT use-cases (FSCDT/TN-012, Issue 0.5, 20th February 2001)
- **[RD10]** SCOS-2000 Command History Operator User Manual (S2K-MCS-SUM-2360-TOS-GCI, Issue 1.4, 23rd April 2001)
- **[RD11]** FIRST Common Science System: Use Case Definitions (FIRST/FSC/DOC/0158, Issue 1.0, 7th November 2000)

1.3 Document structure

This document consists of the following sections:

- Section 1, this section, explains the purpose and context of this document.
- Section 2 provides an overview of the TC history ingestion process.
- Section 3 presents the package structure. It details the structure of the package which contains the TC history ingestion software and describes the TC ingestion API.
- Section 4 presents the design decisions made when designing the TC history ingestion software.

2. THE TC HISTORY INGESTION PROCESS

2.1 Overview

The TC history ingestion process is responsible for retrieving TC history data from SCOS-2000, converting it into a suitable format for storage in the HCSS archive, and associating each TC history record with its corresponding TC.

2.1.1 Requirements

The following functional requirements are placed on the TC history ingestion process:

- **HCSS-UR-3.1-1111**: The HCSS shall be able to associate the TC in the TC history data to the instrument or S/C commanding requests in the corresponding Observations Schedule [**RD7**].

- **UCF-759**: TC history ingestion involves verifying the TC history against observation executions. Any inconsistencies should be detected and flagged, e.g. failed commands, missing commands and additional commands, not possible to determine observation execution associated with TCs [**RD-11**].

2.1.2 Assumptions

The following assumptions are made:

- The format of the TC history file is an ASCII file [**RD-3**].
- The TC history file will be produced by SCOS-2000.
- Commands in the schedule generated by HCSS will be tagged with a 32 bit sequence identifier (TCID) to uniquely identify them from manual commands. Manual commands will have a null or default TCID.
- The schedule to be uplinked to the instrument will also contain commands for setting the Observation and Building Block Identifiers (OBSID's and BBID's) as described in [**RD-5**]. These identifiers can be used to relate the TC history with individual observations for commands with null/default SEQIDs.
- The time in the TC history will be UT (Universal Time), whereas the time in the schedule will be TAI (Temps Atomique Internationale). In ILT, only relative times are used in the schedule instead of absolute times as in operations. This implies that time conversion will be necessary to link the schedule times with the TC history times.
- Event-driven TC packets (on-board generated) are outside the scope of this functionality [**RD8**].
- The BBIDs for all instruments are present in an observing schedule.
- TCs are the lowest level class in command generation.
- TCs are made persistent when the schedule is exported to the MOC.
- The TC history file only contains values for editable command parameters.

2.1.3 Structure

The TC history ingestion process consists of 2 components:

- A TC history file (THF) server to handle requests for TC history
- A TC history ingestor to process TC history

The components communicate with each other via TCP/IP. Note that this is separate to the TCP/IP protocol used to communicate with SCOS-2000 and will not replaced in later versions of the ingestion process.

2.1.3.1 THF server

The THF server component of the TC history ingestion process has the following responsibilities:

- To listen for a request from the ingestor for TC history for a particular time period
- To establish a connection with SCOS-2000
- To inform SCOS-2000 of the TC history data it wishes to receive. This is done by specifying the particular time period for which data is required.
- To receive the filename of the THF from SCOS-2000
- To retrieve the THF from the directory where SCOS-2000 has placed it
- To send the THF to the ingestor

2.1.3.2 TC history ingestor

The TC history ingestor component of the TC history ingestion process has the following responsibilities:

- To establish a connection with the identified database
- To establish a connection with the THF server
- To request the TC history for a particular time period
- To receive a THF from the server containing the requested data
- To validate the THF by:
    - Checking that the first line is as described in [**RD-3**]
    - Checking that the time window covered by the THF is as requested
- To create a TCHistory object for each TC in the THF
- To associate each TCHistory object with its TC in the database (based on the TCID); if the TCHistory object has no corresponding TC in the database (e.g. it represents a manual TC (TCID = 0)), the TCHistory object is stored as is (TBC). Non-manual missing TCs in the database are identified.
- To update the success flags of Observations and Block Executions covered by the THF
- To check that each TC in the database supposed to have a TCHistory (i.e. the TC lies within the time period of the file) does – this identifies missing TC history records.
- To commits the changes to the database

The ingestor runs continuously and requests a new THF after waiting for a specified duration.

2.2 Class diagram

3. PACKAGE STRUCTURE

The package nl.esa.herschel.tchingest contains the TC history ingestion software and consists of the following subpackages:

- api: Contains the interfaces and abstract classes that make up the TC history ingestion API.
- bin: Contains the script for invoking the TC history ingestion process.
- doc: Contains the TC history ingestion documentation.
- server: Contains the class which interfaces with SCOS-2000 to retrieve TC history data in the form of THFs.
- test: Contains the unit level test harnesses for the public tchingest classes.
- util: Contains utility classes for the TC history ingestion process.

The structure of the TcHistory object is fully documented in [**RD-3**] with the code and corresponding javadoc documentation contained in the nl.esa.herschel.ccm.api.tchist package.

3.1 The nl.esa.herschel.tchingest.api API

3.1.1 Interface summary

3.1.1.1 FileIngestion

This interface is used by the file ingestion processes – TC history and OOL data. The key methods are:

- *getFile()* method – retrieve a file.
- *checkFile()* method – verify that the file is of a valid format and contains no data corruptions.
- *processFile()* method – process the file.
- *initStore()* method – initialise the interaction with the archive.
- *commitData()* method – commit the processed data to the archive.

3.1.2 Class summary

3.1.2.1 AbstractFileDetails

This class encapsulates data used to configure both the TC history and OOL data ingestion processes.

3.1.2.2 AbstractFileIngestion

This class extends the FileIngestion interface to provide common implementations for both the TC history and OOL data ingestion processes of the following methods:

- *getFile()* method – retrieve a file covering a specified time interval from the file server
- *initStore()* method – open a new database session
- *commitData()* method – commit the processed data to the database by making each processed object persistent.
- *abort()* method – stop the current ingestion process by clearing the list of processed objects and aborting the current database session (roll-back).
- *wait()* method – wait a specified period of time.

3.1.2.3 AbstractFileServer

This class provides a common implementation for both the TC history and OOL data ingestion processes of the following methods:

- *run()* method – receive a request for a file covering a specified time interval from a client, retrieve the file from its source, e.g. SCOS-2000, and deliver it to the client.
- *getFile()* method – this is implemented by subclasses of this class to handle retrieval of the file from its source, e.g. SCOS-2000.

4. DESIGN DECISIONS

4.1 The TC history ingestion API

The TC history ingestion API is described in section 3 with the code and corresponding javadoc documentation contained in the nl.esa.herschel.tchingest package.

4.2 Interfacing with SCOS-2000

4.2.1 The SCOS-2000 interface

The SCOS-2000 interface is fully documented in [**RD?**].

4.2.2 The THFServer class

This is the class in the server subpackage (see Section 3) which is responsible for communicating with SCOS-2000. This is currently achieved by a TCP/IP connection but will use a CORBA interface in future versions. SCOS-2000 places the requested THF in a prespecified directory (configured from SCOS-2000) where THFServer can retrieve it – this necessitates that THFServer is either running on the same host machine as SCOS-2000 or has access to the directory (e.g. networked disks). The TCHIngestion class connects to this class to retrieve the THF.

4.2.3 SCOS-2000 availibility

Problem:
The TC history ingestion process in unable to connect to SCOS-2000 or the connection to SCOS-2000 goes down.

Solution:
The TC history ingestion process waits for a specified amount of time and then tries to establish or reestablish the connection. Each successful connection, loss of connection and failure to connect is logged.

4.3 Validation of TCs and TC history records

4.3.1 Missing TCs

The TCID is a unique identifier for TCs and is included in the TC history record for a particular TC. The TCHIngestion class has the responsibility of associating TC history records with their corresponding TCs and so can detect missing TCs in the database when processing the THF.
When a missing TC is encountered, a warning message is logged specifying the missing TCID.

4.3.2 Missing Observations/Building Blocks

The header of a TcHistory object also contains the ObsID and BBID of the Observation and Building Block associated with the TC. If the Observation or Building Block cannot be retrieved – it might be missing or the TC might be missing (see 4.3.1), a warning message is logged by the TCHIngestion class

specifying that the TcHistory object with TCID cannot be associated with an Observation or Building Block.

4.3.3 Header information for manual TCs

TCs resulting from manual commands have TCID = 0, which means that the associated Observation and Building Block cannot be retrieved from the TCID reference. An attempt is made, however, by the TCHIngestion class to find the Observation and Building Block most closely associated with the time stamp of a TC history record, i.e. if the time stamp of a TC history record falls within the time period covered by a particular Observation and Building Block then the ObsID and BBID of these are specified in the header of the corresponding TcHistory object.

4.3.4 Missing TC history records

All TCs in the database for the time interval covered by the THF should have an associated TcHistory object. When a missing TC history record is detected by the TCHIngestion class, a warning message is logged specifying which TCID has no history.

4.4 Interfacing with the database

4.4.1 Interaction with the HCSS archive

The nl.esa.herschel.store.api API is used by the TCHIngestion class to interact with the HCSS archive.

4.4.2 Associating TC history records for manual TCs (TCID = 0)

Problem:
TC history records for manual commands (TCID = 0) cannot be associated with a TC (and BB/Observation); with what should be associated in the database? (SCR-0119)

Solution:
The Schedule does not seem the right place to store them as manual commands could, in principle, occur outside the scope of a schedule. A dummy TC could be created, with which to associate the history, but this would only result in pairs of floating objects in the database. Instead such TC history records (TcHistory objects) are simply logged and made persistent in the database, without associations. A TcHistoryExtent class could be added, when required. that retrieves the history for a given time period, to hide the retrieval mechanisms from the client applications. Note that if a suitable container/association can be identified at a later date, the associations could be retrofitted for existing data.

4.4.3 Missing TCs with valid histories

Problem:
The corresponding TC for a TcHistory object with a non-zero TCID cannot be found in the database; with what should it be associated? (SPR-0118)

Solution:
This represents an error condition: the ingestor should make the TcHistory object persistent, ensuring that the data is not lost, and log an error message.

4.5 Successful operations

4.5.1 TC success status

The success status of a TC is determined from its associated TcHistory object. If the TC was not successful, a warning message is logged by the TCHIngestion class specifying the TCID.

4.5.2 Building Block success

If all the TCs for a particular Building Block were successful, the TCHIngestion class sets the uplinked status of the Building Block to successful.

4.5.3 Observation success

If all the TCs for a particular Observation were successful, the TCHIngestion class sets the executed status of the Observation to successful.

4.6 Performance

4.6.1 Size of TC history file

[**RD-3**] specifies that the size of a THF can range from 1 TC to all TCs; however, the latter option if not supported by MOC. Actually a THF can also contain 0 TCs. Typically, it is envisaged that a THF will contain a day's worth of TC history records.

4.6.2 Database usage

It is inefficient to perform a database commit for every TcHistory related transaction. A commit is therefore only performed at the end of processing a THF.

4.6.3 Frequency of ingestion

[**RD-3**] states that it is envisaged that one THF per day will be produced so the frequency of ingestion should be of the same timescale.