Mars Exploration Rover (MER)

Software Interface Specification

Interface Title: **Rover Motion Counter Master File**
Mission: MER Date: August 1, 2004
Module ID: SIS-SCI013-MER
Module Type (REFerence Only or MISsion-specific info included): MIS

Reference Module ID: N/A Date: N/A

Signatures

GDS Generating Elements:

Ops Product Generation Subsystem (OPGS)
Justin Maki

____
Subsystem Engineer Date

GDS Receiving Elements:

MER APSS Surface Subsystem Engineer
Andy Mishkin

____
Subsystem Engineer Date

Concurrance:

MER Imaging Investigation Scientist
Justin Maki

____
Investigation Scientist Date

RSVP Development Team
Brian Cooper

____
Cognizant Engineer Date

SAP Development Team
Jeff Norris

____
Cognizant Engineer Date

SUMMITT Development Team
John Wright

____
Cognizant Engineer Date

MIPL Cognizant Engineer
Doug Alexander

____
Cognizant Engineer Date

Mars Exploration Rover Project

Software Interface Specification (SIS)

Rover Motion Counter (RMC) Master File

Version 1.0

Custodians:
Bob Deen and Doug Alexander

JPL D-22854

August 1, 2004

Jet Propulsion Laboratory
California Institute of Technology

CHANGE LOG

DATE	SECTIONS CHANGED	REASON FOR CHANGE	REVISION
10/2/03			Revision A

OPEN ACTION ITEMS

ITEM	ASSIGNEE

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

(TBD)

LIST OF TABLES

(TBD)

ACRONYMS AND ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
EDR	Experiment Data Record
FSW	Flight Software
GDS	Ground Data System
GSE	Ground Support Equipment
HGA	High Gain Antenna
ICER	Image compression algorithm (not an acronym)
ID	Identification
IDD	Instrument Deployment Device
JPL	Jet Propulsion Laboratory
MER	Mars Exploration Rover
MIPL	Multimission Image Processing Laboratory
NASA	National Aeronautics and Space Administration
ODL	Object Description Language
OPGS	Operations Product Generation Subsystem
OSS	Operations Storage Server
Pancam	Panorama Camera
PDS	Planetary Data System
PMA	Pancam Mast Assembly
RDR	Reduced Data Record
RMC	Rover Motion Counter
RSVP	Rover Sequence Visualization Program
RVF	Rover Vector File
SAP	Science Activity Planner
SCLK	Spacecraft Clock
SFDU	Standard Format Data Unit
SIS	Software Interface Specification
SOAS	Science Operations Analysis Software
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events kernels
SVF	Site Vector File
TBD	To Be Determined
TDS	Telemetry Delivery Subsystem
VICAR	Video Image Communication and Retrieval

GLOSSARY

TERM	DEFINITION

1. INTRODUCTION

1.1 Purpose and Scope

This document describes the file format for "RMC Files", which contain the position and orientation of various coordinate systems indexed by the Rover Motion Counter, or RMC. In addition to generic RMC files, two important specializations of the file are described - Site Vector Files (SVF) and Rover Vector Files (RVF). The concept of "master" and "daily" files is described, and operational scenarios describing typical uses of these files are presented.

The document is oriented towards MER, and often MER-specific terms are used (the names of the RMC slots, for example). However, the concept embodied herein is not MER-specific, and could easily be adapted for use with future missions.

In this document, the two concepts of "position" (where the rover, or coordinate system origin, is in 3-D space, also called "offset") and "orientation" (the orientation of the rover or coordinate system) are important, and are often combined. For convenience, the term "location" is used to represent a combination of both a position and an orientation.

2. DEFINITIONS OF TERMS AND CONCEPTS

2.1 Rover Motion Counter (RMC)

The Rover Motion Counter is a series of numbers that are used to keep track of the location of a rover. Any time an actuator on the rover moves, the RMC is incremented. This allows ground software to keep track of data that was acquired from the same location. See [EDR SIS].

The RMC is broken up into several individual numbers representing the (set of) actuators that moved. For MER, these are, in order:

1. **Site** - Declared by operations personnel (i.e. manually), this is a major coordinate frame from which all activities in a local region are referenced. When the Site is incremented. all other components are set to 0.
2. **Drive** - Incremented by the rover whenever it intentionally moves. When the Drive is incremented, all other components except for Site are set to 0.
3. **IDD** - Incremented by the rover whenever the IDD (Instrument Deployment Device) moves. Other components are NOT set to 0 when this changes.
4. **PMA** - Incremented by the rover whenever the PMA (Pancam Mast Assembly) moves. Other components are NOT set to 0 when this changes.
5. **HGA** - Incremented by the rover whenever the HGA (High Gain Antenna) moves. Other components are NOT set to 0 when this changes.
6. **Tweak** - Never seen in downlink, and unlikely to ever be used, this is reserved to represent an unexpected motion by the rover. For example, it might slip off a rock or be blown by the wind unrelated to an actuator movement. In that case, operations personnel could manually add a "tweak" value if they determine that one occurred. It would be treated just like an IDD, PMA, or HGA movement in software. While software should allow for this eventuality if possible, it is not a requirement, and Tweak will not be specifically discussed further in this document.

There are two basic categories in the above. Site/Drive increments represent cases where the rover is expected to move. Because of this, all lower-priority RMC values are reset to 0. However, when IDD, PMA, or HGA is incremented, the rover is not "supposed" to move. It might do so (e.g. the arm hits a rock and pushes the rover), but it is not the nominal case. This is a very important distinction, and the two categories are treated slightly differently in some cases.

Because the RMC is monotonically increasing, it can be used as a clock of sorts. Clock "ticks" are at very irregular intervals, but given two RMC values, it can be unequivocally determined which came first chronologically. The one that has a lower Site, or if equal, a lower Drive, or if equal, the lower value on any of the remainder, is the first one. Note however that given an RMC value, it is not possible to predict what the next RMC will be, since any of the values could increment.

To illustrate, Table 1.2 shows a possible sequence of RMC values, along with rover activities that could have caused them:

Table 1.2 Example RMC Sequence

Rover Activity	Site	Drive	IDD	PMA	HGA
1) Initial condition	3	5	11	22	3
2) IDD is stowed	3	5	12	22	3
3) Site is declared by Project	4	0	0	0	0
4) Small panorama is taken	4	0	0	1	0
5) Small panorama is taken	4	0	0	2	0
6) Small panorama is taken	4	0	0	3	0
7) HGA pointed toward Earth	4	0	0	3	1
8) Panorama continues	4	0	0	4	1
9) Rover drives forward	4	2	0	0	0
10) IDD deployed	4	2	4	0	0
11) IDD placed on rock	4	2	7	0	0
12) Picture taken of IDD	4	2	7	1	0
13) IDD stowed	4	2	12	1	0
14) Rover backs up	4	3	0	0	0
… etc. ...	… etc. ...	… etc. ...	… etc. ...	… etc. ...	… etc. ...

Note that the value could increment many times with no data being returned. For example, the first Drive consisted of two segments, and the IDD motions required several individual movements.

Thus there is no requirement that the RMC increment only once per incident; this is up to the flight software. Not every possible RMC will be seen on the ground. By the same token, more than one of the minor RMC indices may increment at the same time, if no data was returned between them (e.g. between an IDD move and an HGA move).

It is common to omit trailing values of the RMC if they are all 0. For example, saying "Site 4" is equivalent to "Site (4,0,0,0,0)" and "Drive (5,3)" is the same as "Drive (5,3,0,0,0)".

Note that while the names of the indices are specific to MER, the concept is not. The RMC idea is likely to be reused in future missions. In order to make code multimission (this is not a requirement), the number of indices, names of the indices, and dividing point between intentional and unintentional moves (for MER, that's between Drive and IDD) should be kept flexible.

2.2 Coordinate Frame

There are several types of coordinate frames in use by MER (see [EDR SIS]). For the purposes of this document, the Site and Rover frames are the most important.

Each type of frame has many different instances. For example, every time the rover moves, a different instance of the Rover coordinate frame is created (since the Rover frame is attached to the rover). A different Site frame is created whenever the project declares a Site. The RMC is used to indicate which instance of a coordinate frame is being referred to. Thus it is not sufficient to say e.g. "Site Frame"; one must say "Site 3 Frame", referring to the Site started at RMC (3,0,0,0,0).

A coordinate frame instance is thus completely identified (not specified) by three elements: the frame name, the RMC at which the frame was created, and the solution ID for the frame (see below).

2.3 Reference Frame

Each coordinate frame instance is specified by an orientation and an offset. These orientations and offsets are relative to another coordinate frame instance, referred to as the Reference Frame. The specification of a coordinate frame thus consists of the orientation and offset, as well as the three identifier elements (name, RMC, and solution ID) of the reference frame (sometimes the reference solution ID is omitted; see below).

The offset element is simply the 3-space vector between the origin of the reference frame and the origin of the frame being defined (current = reference + origin).

The orientation element is a quaternion representing the rotation of the current frame with respect to the reference frame (reference = current * orientation). Note that per ground system convention, the scalar part of the quaternion is listed first; this differs from the flight system convention of putting the scalar part last.

For MER, a Site frame will always have an orientation of identity (1,0,0,0); thus all Site frames are assumed to be parallel. However, the orientation quaternion is still stored for completeness, and in order to support future missions where this may not be the case (e.g. long-range rover where planet curvature becomes a factor).

When a Rover frame is defined with a Site as its reference (the normal case), then the orientation quaternion is the orientation of the rover with respect to North/Nadir. If a Rover frame were defined with another Rover frame as reference, this is not the case - the quaternion would represent the rotation between the two (but this is rarely done; the reference should usually be a Site frame).

While a complete reference specification contains the reference's solution ID, there are cases in which the reference's solution ID should be omitted. See the discussion under "Solution", below.

2.4 Solution

The knowledge of rover position and orientation is unfortunately very imprecise. The value returned in telemetry is derived via methods such as counting wheel rotations and integrating positions, which is inherently an imprecise process. Better positions can often be derived via analysis of ground imagery. However, there are many different methods of doing this. Each method, while perfectly valid, may create a different result. To make things worse, many methods require manual input, and thus different results can be generated even using the same method.

Each of these results is called a "Solution", and is valid for a particular purpose. (Some older documents use the term "opinion" instead of "solution".) Usually a "better" (by Project determination) solution will replace an older one, but it's also possible that multiple solutions may be simultaneously valid for different purposes. One may have been chosen to minimize globalerror, while another was chosen to minimize seams in a mosaic taken from two vantage points. Note that the "true" value can never be known; only approximations to it by various means can be determined.

In order to keep track of all these results, each is assigned a Solution ID (attribute "solution_id"). This ID uniquely identifies the solution. Once an ID is assigned to a given solution, that solution may never change - that ID will always refer to the exact same set of numbers. Note though that solution ID's can be re-used in other contexts. For example, an ID of "MER2_003" might be used for three different Sites. And, a given solution could have more than one ID associated with it (for example, solutions are renamed when they go into a Master file, described later). But the combination of Solution ID, Coordinate Frame Name, and RMC value is unique, and will always refer to the same result.

The special solution ID "telemetry" is reserved for values received in telemetry. This represents the rover's idea of where it is. The "telemetry" ID is often omitted; for example, EDR labels do not contain a solution ID. In these cases, "telemetry" is assumed by default.

Naming conventions for Solutions ID's are specified in the sections on Generic RMC Files and Master Files, below.

In general a reference frame identification contains a solution ID. However, there are cases in which the solution ID can (and should) be omitted. By way of illustration, consider a case where the location of a Rover frame is being refined by comparing images taken from that location with a panorama taken at the Site. The Rover frame location thus derived is really referenced to the images from the Site. The actual location of the Site is not relevant. Thus the Rover frame location is actually referenced to the true Site, not to any given solution for where the Site actually is. Since the actual Site location is not included in the Rover frame location computation, the same location applies to any Site solution. Thus, the solution ID for the Reference (the Site) should not be included, since it does not matter (and would actually be misleading if included). If, however, the value of a specific Site solution were in fact included in the Rover frame computation, then the solution used would matter, and a solution ID for the reference would be required.

To summarize, if the actual value of any reference's solution is used to compute the current frame's solution, then the reference's solution ID must be present. If the reference value is not used (so the current frame's solution is entirely relative to the reference), then the reference solution ID should not be present.

2.5 Solution Priorities

An RMC File will often contain more than one solution for a given Frame/RMC. A prime example of this is a Master file, described later. In order to establish an ordering of these solutions, RMC files can contain a priority list. This simply lists all (or some set of) solutions contained within the file, in order of preference (preference being defined by the creator of the file).). The highest priority (or most preferable) solution is listed last. Thus the user can determine which solution for a given Frame/RMC is "best" by looking at this priority list.

2.6 Aliases

When a Site is declared by the project (and this fact is uplinked to the rover), a special case occurs. The RMC is reset in the software to 0's with the Site value incremented. Thus two values of the RMC exist where it is absolutely known that the rover did not move - just before, and just after, the site increment. In order to keep track of this, RMC files may contain an "alias" entry. This simply establishes the equivalence between the before- and after- reset RMC values. This equivalence could be useful if e.g. the Site reference panorama is taken before the Site is declared.

2.7 Derivations

Each entry in an RMC file may optionally contain a derivation tag. Derivations are a special case, used only for Master files (described later).

When an entry is placed in a Master file, the reference for that entry must be the enclosing Site (or immediately prior Site for a Site entry). However, solutions are usually generated by comparing two sets of images, acquired from two different locations. The only reliable information obtained from this process is the relative offset/orientation between those two locations.

One set of images was obviously taken from the location defined by the entry. The other set of images is usually the Site panorama. Thus, the reference is the enclosing Site, and the entry may be placed in the Master file.

However, the reference need not be from the enclosing Site. Perhaps the rover went behind a rock and is not visible from the Site panorama. In these cases, the reference frame of the entry is not the enclosing Site, and thus the entry may not be put directly into the Master file.

It is easy enough to convert the reference to the appropriate Site frame. Location B's reference is location A, which is referenced to Site S. The entry for B is converted to use S as a reference by adding A's location. If the intermediate reference A is subsequently adjusted, then B's location - which is only known relative to A and depends on A's location - also needs to be adjusted.

In order to accomodate this possibility, the original reference (A in this case) and the offset/orientation for it are stored in a Derivation tag within the entry. If A is moved by a subsequent adjustment, a program can optionally scan for additional frames derived using A as a reference, and move them as well (using a different solution ID, of course).

Users of the file should generally ignore the derivation tag. It is only relevant to processes which update entries in the file.

2.8 RMC File

An RMC file - the subject of this document - is nothing more than a collection of coordinate system definitions containing the above-mentioned items. It contains a list of some set of coordinate frames (identified by Frame Name, RMC, and Solution ID), the reference frame for each (similarly identified), and the offset and orientation between the frames.

In addition, an RMC file may contain a solution priority list, and a set of aliases. Additional descriptive items, such as derivations and origination tags, may also appear.

The same physical format is used for a variety of specialized purposes (detailed in the next section).

3. KINDS OF FILES AND THEIR USES

In this section we look at the various types of RMC files, what they are used for, and what rules apply to distinguish them from the generic RMC format. For more information on how the files are used, see the Operational Scenarios section.

3.1 Generic RMC Files

A generic RMC file can contain any kinds of entries allowed by the file format. No specific content requirements are imposed.

Generic files are often used as intermediaries when new solutions are being developed. Analysis programs that determine the rover location will output a generic RMC file containing the solution that they determine. If the user decides to submit his/her solution to the Project for inclusion in the Master file, the generic RMC file is the medium of transfer for that submission.

Outside of a master or daily file, there is no technical requirement for the naming of Solution ID's (other than "telemetry" and the master file names being reserved). However, the MER project recommends the use of several components separated by underscores:

<institution>_<user>_<purpose>_<version>

Institution or user may be omitted if not relevant.

Examples:

a) mipl_rgd_egress-drive-fix_1
b) mipl_daa_Gallery-Pan_1

3.2 Site Vector Files

A Site Vector File (SVF) is an RMC file that contains nothing but Site Frame definitions. Each Site is defined relative to the immediately previous Site, and all Sites back to 0 must be defined (Site 0 has no reference and is not in the file). Thus the file defines a continuous chain of Sites from 0 up to the current Site.

Multiple solutions for each Site may exist (at least in a Master SVF). A priority list must exist, which defines which solutions are the "best". Aliases should exist for each Site (except 0), defining the RMC used before the Site was declared.

Site Vector Files are used to determine the relationship between Site's. Since EDR's do not contain any Site Frame definitions in their labels, and RDR's may contain at most one, an SVF is required any time data from more than one site is being compared or combined.

3.3 Rover Vector Files

Similar to a Site Vector File, a Rover Vector File (RVF) contains nothing but Rover Frame definitions. However, instead of containing _all_ Rover Frame definitions, an RVF file contains only Rover Frame definitions within one Site. Thus different RVF files will exist for each Site.

RVF files will typically <u>not</u> contain an entry for each and every RMC within their Site. There could be a huge number of these, and it becomes unmanageable when new solutions are generated. Instead, an RVF file contains an entry for every RMC at which the rover location changed. This includes of course all Drive increments (for which data was downlinked). If the file contained only telemetry data, that it would be most of what it would contain: Drive increments with 0,0,0 for the lower three components. Telemetry will rarely generate an RVF entry for anything other than a Drive increment, but it is possible (for example, if a new sun-find is performed).

If it is subsequently determined that the rover moved between drives (say, the IDD moved the rover), then a solution is generated for that, and an entry is created in the RVF for the RMC value at which the move occurred.

Thus, the file contains only deltas - RMC values at which the location of the rover changed.

In order to find the location of the Rover frame for any given RMC, the file must be searched for the highest RMC value less than or equal to the given one. In order to make this easier, entries in an RVF must be stored in ascending order of RMC (and with multiple solutions for one RMC, in ascending priority - the last one has the highest priority).

For example, consider the sequence of RMC's in the section on RMC defintion, above. An RVF file for site 4 based on telemetry would have entries for (4,0,0,0,0), (4,2,0,0,0), and (4,3,0,0,0). Ground analysis later determines that the IDD placement at (4,2,7,0,0) moved the rover. This solution would be added to the RVF file in order and is shown in Table 3.3 below:

Table 3.3 Example RMC Sequence

Solution ID	RMC
1) Telemetry	(4,0,0,0,0)
2) Telemetry	(4,2,0,0,0)
3) some_id	(4,2,7,0,0)
4) Telemetry	(4,3,0,0,0)

In order to find the rover location for "Picture taken of IDD" at (4,2,7,1,0), the file would be searched to find the highest RMC less than or equal to (4,2,7,1,0). In this case, that's (4,2,7,0,0). In order to find the rover location for "Small panorama" at (4,0,0,1,0) and following, the same procedure yields

(4,0,0,0,0). The rover location corresponding to the RMC found is extracted from the RVF file and used for the image in question.

3.4 Master Files

Both RVF and SVF files have the concept of a "Master File". This is a central file which contains all Project-approved solutions that have been generated for their respective coordinate frames.

The idea is that a Master file is a central repository where any approved solution can be found. This both provides a historical record, and facilitates reprocessing of old data (using some older solution). Most processing programs will not read a Master file directly; they will instead use Daily files (see below).

There is only one SVF master file for each spacecraft. There is one RVF master file per spacecraft and Site. While multiple versions of a master file may exist for backups and history (the filename convention includes a version number), only the most current version is ever used. Since it maintains all definitions (nothing is ever deleted from a Master file; things are only added), there should be no need to refer to older versions of the file.

A master file naturally contains the telemetry solution for all relevant coordinate frames (all Sites, and all Rover frames where the rover intentionally moved). It additionally contains any subsequent solutions that are generated by ground processing. These subsequent solutions, however, are renamed as they are placed in the master file.

A master file has a specific naming convention for solutions within it. "telemetry" is always used for the telemetry solution. Subsequent solutions are assigned the ID "mission_001", "mission_002", etc. where "mission" is the mission name as defined in the INSTRUMENT_HOST_NAME PDS label (for example, MER1_001, MER1_002, etc.). Note that the numbers are counted independently for each RMC in the file. So for any given RMC, the "MER1_001" slot will be filled up before "MER1_002" is used. For example, some early RMC might be on "MER1_012" (it's unlikely to get so high), but at the same time, the first solution for a new RMC is assigned to "MER1_001".

As a result, the priority list for a master file is constant per mission. In ascending order (best is last):

telemetry
MER1_001
MER1_002
MER1_003
...

Normally the priority list will only contain a few entries (10 should be more than sufficient for the entire mission). Any time a solution is added to the file, however, the priority list must be checked to see if extension is necessary. Extra (unused) slots in the priority list are acceptable (although they may hurt performance somewhat).

A Master file entry (other than telemetry) must contain a <derivation> tag, with the ID of the original solution ID before it was put in the master file. This is used for tracking and historical purposes. The other elements in the <derivation> tag are optional depending on the data (see the section on Derivations).

Master file entries also contain the date/time at which they were added to the Master file. This has nothing to do with the date/time they were generated. The purpose is to allow re-generation of old Daily files as of a certain date.

3.5 Daily Files

Associated with each Master file (both SVF and RVF) is a set of Daily files. As the name implies, a Daily file is normally created each day (Sol), from the corresponding Master file. It contains the <u>single</u> best solution for each Master file entry as of the time it was created.

Thus the Daily file has the "best" Project-approved definitions for the current processing "day". (the Project may wish to generate more than one per day, or fewer, but the name still applies).

Most applications will make use of the Daily files, in order to avoid having to deal with multiple solutions.

The solution ID's in a Daily file match the ID's in the source Master file (they are not renamed). <derivation> tags and the add_date are removed.

Note: Daily files were never generated during MER operations.

4. OPERATIONAL SCENARIOS

This section presents some example scenarios and the steps that would be followed to effect them, and the changes in files that result. The whole section should be considered for illustration and clarification only.

4.1 Declaring a New Site

When the Project decides to declare a Site, that fact must be uploaded to the spacecraft (not covered here). The ground processing is as follows. Note that this operation is initiated manually, since Sites are relatively infrequent and there's no reliable way to determine when one is declared automatically.

The SVF Master File must be updated to reflect the new Site. The “mer_newsite” program will do this. The user provides the following information:

- Old RMC (before site declaration), used to define an alias
- Telemetered value of Site offset vector
- Telemetered value of Site quaternion (theoretically; it's identity for MER)

This information could be provided by supplying an EDR from just before the Site was declared, or by entering it directly.

The “mer_newsite” program then calculates the new Site number (old RMC's Site plus 1), and creates a new <solution> entry in the SVF Master file using a solution ID of "telemetry". It then adds an <alias> entry to the file.

4.2 Pipeline Processing of EDR's

As EDR's are processed in the pipeline, they need to be analyzed for any impacts they might have on the RVF files (note that SVF files are handled manually, as above). This is accomplished by sending all EDR's through the “mer_ingest” program.

This program accepts a list of EDR's, and looks at the label of each one, extracting the RMC. It then searches for an entry in the appropriate RVF master file that matches the Site and Drive indices of the RMC.

If an entry is found, the location values are compared against the EDR as a consistency check, and a warning issued if they do not match within some epsilon.

If an entry is not found, then the EDR represents a new Drive location, so an entry is created and added to the RVF Master file. The IDD, PMA, and HGA components of the RMC in the entry are set to 0 regardless of the actual EDR's RMC value. This ensures against EDR's being presented out of order to “mer_ingest”; the change occurs at the Drive increment. The solution ID for the new entry is "telemetry".

4.3 Generating a New Solution

When a user wants to refine the location of the rover at some point, many methods could be used. Most commonly, images will be analyzed from before and after a move using some program like “xyznet” or “rovernav”. Whatever the method, the final result is a generic RMC file containing the new solution. The solution ID is assigned by the user (presumably according to some Project convention). The solution contains a relative location of the rover with respect to some reference frame (there are no absolute positions or orientations).

The user may iterate to refine the process, try different parameters, or try different methods. Eventually, a satisfactory solution is reached. Once this happens, the user may (optionally) submit it to the Project for approval. Note that the "user" here could be an automated process, or an actual person. It is not required that any given solution be submitted to the Project; solutions may be private to the user or shared locally, if appropriate.

The solution could be for a Site, or for a Rover frame. Either way, the process is the same. The differences are the program names and the master file(s) used.

Once the solution is submitted, someone in the Project checks it and approves it (this process is outside the scope of this document). The approved solution must then be incorporated into the appropriate Master file.

The “mer_svf_append” or “mer_rvf_append” programs are run to add the solution to the master file. These programs both work similarly. They loop through the input RMC file looking for definitions that match the input criteria. For each one, they search for the entries in the master file that match the entry's RMC. For SVF, a match must be found; for RVF, a match is optional (since this might be an IDD move, for example).

The matching master file entries are then scanned to determine the next solution ID to use. If, necessary, the new ID is added to the priority list (this is rare).

The reference of the new entry is then checked to see if it matches the proper reference for master files (enclosing Site for RVF, prior Site for SVF). If it doesn't match, the reference itself is looked up in the master files, and the latest entry is used to "re-parent" the entry to have the appropriate reference. (In this case, the solution ID used for the reference does matter.) The original reference information is added to the <derivation> tag in the new entry.

Finally, a <solution> entry is created for the new entry and inserted into the file at the appropriate (sorted) position.

4.4 Generating a Daily File

Once per processing day (Sol), or more often if the Project decides, Daily files are created from each Master file (both SVF and each Site's RVF). Creation of these files may be initiated by an automated process.

The programs “mer_svf_daily” and “mer_rvf_daily” (or a combination script called “mer_daily”) create a daily file from a master file.

Both of them accept a cutoff date/time. This is usually the date/time at which the program is being run, but an earlier cutoff can be used to re-create previous days' files.

The Daily programs go through the master file looking for each unique RMC value. For each, the highest-priority (latest) available solution is extracted. Items not appropriate to the Daily file are removed from the entry (derivation, add_date tags), and the entry is appended to the Daily file. The Daily file's priority list and alias list (SVF only) are copied from the Master.

Finally, the softlinks to the most recent Daily file are updated to point at the new file.

4.5 Ingesting an Image

In addition to RMC files, coordinate system definitions are also stored in EDR/RDR labels. These labels may be updated from time to time to reflect the current state of the SVF/RVF master files, but in all cases the master files should provide the master definitions.

However, in case the master files are incomplete for some reason, it is possible to augment the in memory "database" of coordinate system definitions with EDR/RDR labels. The algorithm is outlined below.

First, the SVF/RVF files should be read into memory to provide a baseline. If they are not available, it is possible to start with an empty baseline, but obviously any information not contained in the files will be unavailable.

Then, for each file being read in, collect all the coordinate system definitions from that file. They should include the name and RMC values, as well as the solution ID (if not present, default to "telemetry"). Filter out the coordinate systems you're not interested in (for MER, only Site and Rover are relevant here). It is worthwhile checking the reference frame to make sure it follows the rules (referencing the proper Site frame) and tossing out ones that don't.

Once a candidate C.S. is found, then the in-memory database is searched for an exact match of solution ID, and the highest RMC that is less than or equal to the candidate's RMC (for a Site frame, only the Site value is relevant and must match; for a Rover frame, the intentional moves (Site/Drive for MER) must match exactly). If no match is found, add the definition to the in-memory database.

If a match is found, then the actual definition (offset/quaternion values) should be compared. If they match within an epsilon (say, 1e-6) then the definitions are considered to be identical, and the candidate is NOT added to the database. If they don't match, then this is a new definition and the candidate IS added.

This algorithm works best if images are presented in increasing RMC order. If that is not the case, and the images contain defintitions not in the starting database, extra definitions would be generated in the in-memory database. While this is unlikely to cause a problem, one could use a similar search for values higher in RMC than the candidate (that also came from a file) and eliminte those that match.

Note that the solution ID's must match during the comparison. This allows the priority mechanism to still work; solution priorities will apply regardless of whether the solution came from the RMC files or an image file.

4.6 Finding a Rover Coordinate System

In order for a program to find a Rover coordinate system (given an RMC), it needs access to the current SVF and RVF Daily files (Masters could also be used but more searching is required; this is not covered here).

It is advisable for programs to also accept an additional RMC file. This would serve to augment and override the official files. This can be useful for special processing, and to use solutions not submitted to the Project for approval. This is not required, however. If additional RMC files are accepted, the easiest way to deal with them is to read the Daily file into memory, and insert the entries from the additional file into the proper sorted order.

In order to find the matching entry for an RMC, a search must be performed. RVF files do not contain every possible RMC value; only values at which the rover moved are included. Thus, the available entries must be searched for the largest RMC value that is less than or equal to the target RMC. Once that entry is found, the coordinate system information can be extracted and used.

Comparing two RMC values for "less than or equal" is a trivial process. Simply compare each index in order (starting at Site). If at any point one index's value is less than or greater than the other, then the entire RMC is less than or greater than the other.

5. FILE NAMING CONVENTIONS

This section presents file naming conventions for all of the files discussed above, as well as describing softlinks that exist for convenience in finding them.

Master and Daily files all contain a [version] field. Every time the file is updated, the version number is incremented (for backups and traceability). The number of digits varies per product (5 for Masters, 3 for Dailies) but it is always 0-filled.

Only the most recent version is ever used; that is pointed to by a softlink. It is up to the Project to determine how many old versions to maintain. Only the most recent version (without the version number) is archived.

5.1 Generic RMC Files

Generic RMC files can technically be named anything the user wants. However, a suggested convention is to use the name of the solution ID contained within, with ".site" or ".rover" as depending on what is being defined. Do not use ".svf" or ".rvf" as those have specific meanings that generic RMC files usually do not meet.

5.2 Master Site Vector Files

Master SVF files are named:

<*mission*>_Master_<*version*>.svf

where,

mission	=	(string) Mission name (matching INSTRUMENT_HOST_NAME in the PDS label).
version	=	(5 integers) 5-digit version number of the file.

Examples:

a)	MER1_Master_00032.svf	Master Site Vector File for rover MER-1, product version 32.
b)	MER2_Master_00105.svf	Master Site Vector File for rover MER-2, product version 105.

Example softlinks without the <version> always point to the most recent file:

a)	MER1_Master.svf	Softlink to most recent version of Master Site Vector File for rover MER-1.
b)	MER2_Master.svf	Softlink to most recent version of Master Site Vector File for rover MER-2.

5.3 Daily Site Vector Files

Daily SVF files are named:

<*mission*>_Sol_<*day*>_Daily_<*version*>.svf

where,

mission	=	(string) Mission name (matching INSTRUMENT_HOST_NAME in the PDS label).
day	=	(1 to 3 integers) Sol number to which this file applies (NOT 0-padded).
version	=	(3 integers) 3-digit version number of the file.

Examples:

a)	MER1_Sol_0_Daily_001.svf	Daily Site Vector File for rover MER-1, Sol 0, product version 1.
b)	MER1_Sol_1_Daily_001.svf	Daily Site Vector File for rover MER-1, Sol 1, product version 1.
c)	MER1_Sol_1_Daily_001.svf	Daily Site Vector File for rover MER-1, Sol 12, product version 1.
d)	MER2_Sol_12_Daily_002.svf	Daily Site Vector File for rover MER-1, Sol 12, product version 2.

Example softlinks without the <version> always point to the most recent file for the Sol:

a)	MER1_Sol_0_Daily.svf	Softlink to most recent version of Daily Site Vector File for rover MER-1, Sol 0.
b)	MER1_Sol_1_Daily.svf	Softlink to most recent version of Daily Site Vector File for rover MER-1, Sol 1.
c)	MER1_Sol_1_Daily.svf	Softlink to most recent version of Daily Site Vector File for rover MER-1, Sol 12.

Additional example softlinks (one per mission only) without the <day> point at the most recent file for the most recent Sol (this is the filename most users will use):

a)	MER1_Daily.svf	Softlink to most recent version of Daily Site Vector File, and most recent Sol, for rover MER-1.
b)	MER2_Daily.svf	Softlink to most recent version of Daily Site Vector File, an most recent Sol, for rover MER-2.

5.4 Master Rover Vector Files

Master RVF files are named:

<*mission*>_Site_<*site*>_Master_<*version*>.rvf

where,

mission	=	(string) Mission name (matching INSTRUMENT_HOST_NAME in the PDS label).
site	=	(1 to 3 integers) Site number to which this file applies (NOT 0-padded).
version	=	(5 integers) 5-digit version number of the file.

Examples:

a)	MER1_Site_3_Master_00004.rvf	Master Rover Vector File for rover MER-1, Site 3, product version 4.
b)	MER1_Site_4_Master_00001.rvf	Master Rover Vector File for rover MER-1, Site 4, product version 1.
c)	MER2_Site_12_Master_00030.rvf	Master Rover Vector File for rover MER-2, Site 12, product version 30.

Example softlinks without the <version> always point to the most recent file:

a)	MER1_Site_3_Master.rvf	Softlink to most recent version of Master Rover Vector File for rover MER-1, Site 3.
b)	MER1_Site_4_Master.rvf	Softlink to most recent version of Master Rover Vector File for rover MER-1, Site 4.
c)	MER2_Site_12_Master.rvf	Softlink to most recent version of Master Rover Vector File for rover MER-2, Site 12.

5.5 Daily Rover Vector Files

Daily RVF files are named:

<*mission*>_Site_<*site*>_Sol_<*day*>_Daily_<*version*>.rvf

where,

mission	=	(string) Mission name (matching INSTRUMENT_HOST_NAME in the PDS label).
site	=	(1 to 3 integers) Site number to which this file applies (NOT 0-padded).
day	=	(1 to 3 integers) Sol number to which this file applies (NOT 0-padded).
version	=	(3 integers) 3-digit version number of the file.

Examples:

a)	MER1_Site_3_Sol_12_Daily_002.rvf	Daily Rover Vector File for rover MER-1, Site 3, Sol 12, product version 2.
b)	MER1_Site_3_Sol_13_Daily_001.rvf	Daily Rover Vector File for rover MER-1, Site 3, Sol 13, product version 1.
c)	MER2_Site_12_Sol_31_Daily_001.rvf	Daily Rover Vector File for rover MER-2, Site 12, Sol 31, product version 1.

Example softlinks without the <version> always point to the most recent file:

a)	MER1_Site_3_Sol_12_Daily.rvf	Softlink to most recent version of Daily Rover Vector File for rover MER-1, Site 3, Sol 12.
b)	MER1_Site_3_Sol_13_Daily.rvf	Softlink to most recent version of Daily Rover Vector File for rover MER-1, Site 3, Sol 13.
c)	MER2_Site_12_Sol_31_Daily.rvf	Softlink to most recent version of Daily Rover Vector File for rover MER-1, Site 12, Sol 31.

Additional example softlinks (one per mission only) without the <day> point at the most recent file for the most recent Sol (this is the filename most users will use):

a)	MER1_Site_3_Daily.rvf	Softlink to most recent version of Daily Rover Vector File, and most recent Sol, for rover MER-1, Site 3.
b)	MER2_Site_12_Daily.rvf	Softlink to most recent version of Daily Rover Vector File, and most recent Sol, for rover MER-2, Site 12.

6. DETAILED FILE FORMATS

6.1 Generic RMC File

A generic RMC file is an XML file. The full format is described in Appendix J. Generic files used for transport of coordinate frame information will usually contain just a subset of the full file format.

Generic RMC files will not have the "variant" attribute of the <rmc_file> tag.

6.2 Site Vector File

The following constrain the Generic file format turn the file into a Site Vector File:

- Only Site frames are defined.
- All RMC components except Site must be 0 or non-existent (excluding alias old entries).
- All reference frames must be the immediately prior Site frame (Site - 1).
- All Sites, from 1 to the maximum defined Site, must have an entry.
- Site 0 does not have an entry (there is nothing to reference it against).
- A priority list must exist, and be complete.
- Alias entries must exist for every defined Site (except 0).
- Site orientations must be Identity, or absent (for MER - Future missions may allow non-identity orientations).
- All entries are sorted, first by RMC value, and then by Solution ID according to priority (highest priority last).
- The <rmc_file> tag must not have an "index1" attribute.

6.3 Master Site Vector File

A Master Site Vector File conforms to the above definition of a Site Vector File with the following additional constraints:

- Multiple solutions may exist for each entry.
- Solution ID's follow the naming convention: "telemetry" is the first entry, followed by "[mission]_001" where [mission] is the mission name and version number increments by 1.
- Each solution other than telemetry must have a <derivation id=> attribute identifying the solution ID under which the solution was originally generated.
- Original reference frame information will be in <derivation> if other than the prior Site frame.
- Solution ID's will not be skipped; e.g. 001 will be used before 002. Each entry counts independently.
- Each entry must have an add_date attribute.
- The "variant" attribute of the <rmc_file> tag will be "Master_SVF".

6.4 Daily Site Vector File

A Daily Site Vector File conforms to the above definition of a Master Site Vector File with the following additional constraints:

- Only one solution may exist for each entry, and it will be the most recent available at the time the file is generated.
- <derivation> tags shall not exist.
- add_date attributes shall not exist.
- The "variant" attribute of the <rmc_file> tag will be "Daily_SVF".

6.5 Rover Vector File

The following constrain the Generic file format turn the file into a Rover Vector File:

- Only Rover frames are defined.
- All entries must be part of the same Site.
- All reference frames must be the enclosing Site frame.
- All Drive values for which telemetry exists must have an entry.
- A priority list must exist, and be complete.
- All entries are sorted, first by RMC value, and then by Solution ID according to priority (highest priority last).
- The <rmc_file> tag must have an "index1" attribute specifying the Site.

6.6 Master Rover Vector File

A Master Rover Vector File conforms to the above definition of a Rover Vector File with the following additional constraints:

- Multiple solutions may exist for each entry.
- Solution ID's follow the naming convention: "telemetry" is the first entry (if applicable), followed by "[mission]_001" where [mission] is the mission name and the version number increments by 1.
- A "telemetry" solution ID is not required if not appropriate (i.e. for unintentional moves).
- Each solution other than telemetry must have a <derivation id=> attribute identifying the solution ID under which the solution was originally generated.
- Original reference frame information will be in <derivation> if other than the enclosing Site frame.
- Solution ID's will not be skipped; e.g. 001 will be used before 002. Each entry counts independently.
- Each entry must have an add_date attribute.
- The "variant" attribute of the <rmc_file> tag will be "Master_RVF".

6.7 Daily Rover Vector File

A Daily Rover Vector File conforms to the above definition of a Master Rover Vector File with the following additional constraints:

- Only one solution may exist for each entry, and it will be the most recent available at the time the file is generated.
- <derivation> tags shall not exist.
- add_date attributes shall not exist.
- The "variant" attribute of the <rmc_file> tag will be "Daily_RVF".

7. EXAMPLE FILES

Example RMC files in XML format are shown in Appendices A, B, C, D, E, F, G, H and I. The RMC file format is shown in Appendix J and the RMC XML schema is shown in Appendix K.

The example files are based on some actual data from SSTB1 (MER testbed). The EDR's they were created from are:

2F128620574EDR0200P2201L0M1.IMG
2F128625988EDR0206P2201L0M1.IMG
2F128626309EDR0300P2201L0M1.IMG

The scenario by which the RMC files were created follows:

In the first (RMC 2,0) the rover appears to be sitting on the lander. In the second (RMC 2,6), the rover appears to have driven off the lander. The last one (RMC 3,0) is taken from the same vantage point as (2,6) but with a new Site declared.

On Sol 43 the data was acquired. The data from the EDRs were put in the Master files, both SVF and Site 2's RVF. After downlink for the day, the Daily files were generated (see Appendix B). So the Sol_43 dailies have just telemetry in them.

Note that no information was available for the location of Sites 1 and 2, so they are simply zeroed out in the SVF.

On Sol 44, nothing happened (for this example) so the daily files would be identical to Sol 43's dailies (they are omitted here for clarity). In reality, new telemetry entries would likely appear in the Site 3 RVF for Sol 44 and 45's telemetered data.

On Sol 45, an analyst looked at the images and determined that (2,6)'s Z position component was wrong. The analyst created a solution (called, according to a somewhat loosely defined convention from the MER Camera EDR/RDR SIS, "mipl_rgd_egress-drive-fix_1"). This was submitted to the Project and approved, and appears in the Site 2 RVF Master file as ID SSTB1_001 (renamed as per the Master file rules).

About an hour later, the same analyst looked again and came up with a better solution, called mipl_rgd_egress-drive-fix-2. This was submitted and approved, and put in the Site 2 RVF Master file as SSTB1_002.

Shortly thereafter, the analyst realized that (3,0) was the same as (2,6) (according to the aliases), so the same solution was submitted as an update to Site 3. This was approved, and put in the SVF Master file as solution SSTB1_002. Note that the same solution is called SSTB1_002 in the RVF file and SSTB1_001 in the SVF file; that's just a result of following the naming convention.

After this was done, Daily files for Sol 45 were generated (see Appendix C), containing the new solutions.

Note that there is no entry for the edit in the Site 3 RVF. That's because only the position was changed, not the orientation, and the position of Rover (3,0) is (0,0,0) by definition. However, if the orientation of the rover at that spot (2,6/3,0) had also changed, an entry would have appeared in Site 3's RVF for the new orientation at (3,0) (the position would still go in the SVF file).

NOTE: The files are named with SSTB1 as the host name. This is what the instrument host should be. However, at the time of this writing, the actual file labels have an INSTRUMENT_HOST_ID of "MER2". It may be necessary to change the SSTB1's to MER2 in the filenames and <rmc_file mission=> tags in order to use the files properly, unless the labels are updated.

APPENDIX A – Example Master Site Vector File "SSTB1_Master_00059.svf"


```
<rmc_file mission="SSTB1" variant="Master_SVF">

    <priority>
            <entry solution_id="telemetry" />
            <entry solution_id="SSTB1_001" />
            <entry solution_id="SSTB1_002" />
    </priority>

    <solution solution_id="telemetry" name="SITE_FRAME" add_date="2003-03-21T09:33:00Z" index1="1">
            <reference_frame name="SITE_FRAME" index1="0" />
            <offset x="0.0" y="0.0" z="0.0" />
            <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
    </solution>

    <alias>
            <old index1="0" index2="15" index3="0" index4="102" index5="5" />
            <new index1="1" />
    </alias>

    <solution solution_id="telemetry" name="SITE_FRAME" add_date="2003-03-25T23:20:00Z" index1="2">
            <reference_frame name="SITE_FRAME" index1="1" />
            <offset x="0.0" y="0.0" z="0.0" />
            <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
    </solution>

    <alias>
            <old index1="1" index2="9" index3="3" index4="45" index5="2" />
            <new index1="2" />
    </alias>

    <solution solution_id="telemetry" name="SITE_FRAME" add_date="2003-03-26T20:12:00Z" index1="3">
            <reference_frame name="SITE_FRAME" index1="2" />
            <offset x="-1.34588" y="-2.31962" z="0.213165" />
            <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547" />
    </solution>

    <alias>
            <old index1="2" index2="6" index3="0" index4="0" index5="0" />
            <new index1="3" />
    </alias>

    <solution solution_id="SSTB1_001" name="SITE_FRAME" add_date="2003-03-27T14:56:00Z"
                index1="3">
            <reference_frame name="SITE_FRAME" index1="2" />
            <offset x="-1.34588" y="-2.31962" z="0.30" />
            <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547" />
            <derivation id="mipl_rgd_egress-drive-fix_2" />
    </solution>

</rmc_file>
```

APPENDIX B – Example Daily Site Vector File "SSTB1_Sol_43_Daily_001.svf"


```
<rmc_file mission="SSTB1" variant="Daily_SVF">

    <priority>
            <entry solution_id="telemetry" />
            <entry solution_id="SSTB1_001" />
            <entry solution_id="SSTB1_002" />
    </priority>

    <solution solution_id="telemetry" name="SITE_FRAME" index1="1">
            <reference_frame name="SITE_FRAME" index1="0" />
            <offset x="0.0" y="0.0" z="0.0" />
            <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
    </solution>

    <alias>
            <old index1="0" index2="15" index3="0" index4="102" index5="5" />
            <new index1="1" />
    </alias>

    <solution solution_id="telemetry" name="SITE_FRAME" index1="2">
            <reference_frame name="SITE_FRAME" index1="1" />
            <offset x="0.0" y="0.0" z="0.0" />
            <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
    </solution>

    <alias>
            <old index1="1" index2="9" index3="3" index4="45" index5="2" />
            <new index1="2" />
    </alias>

    <solution solution_id="telemetry" name="SITE_FRAME" index1="3">
            <reference_frame name="SITE_FRAME" index1="2" />
            <offset x="-1.34588" y="-2.31962" z="0.213165" />
            <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547" />
    </solution>

    <alias>
            <old index1="2" index2="6" index3="0" index4="0" index5="0" />
            <new index1="3" />
    </alias>

</rmc_file>
```

APPENDIX C – Example Daily Site Vector File "SSTB1_Sol_45_Daily_001.svf"


```
<rmc_file mission="SSTB1" variant="Daily_SVF">

   <priority>
          <entry solution_id="telemetry" />
          <entry solution_id="SSTB1_001" />
          <entry solution_id="SSTB1_002" />
   </priority>

   <solution solution_id="telemetry" name="SITE_FRAME" index1="1">
          <reference_frame name="SITE_FRAME" index1="0" />
          <offset x="0.0" y="0.0" z="0.0" />
          <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
   </solution>

   <alias>
          <old index1="0" index2="15" index3="0" index4="102" index5="5" />
          <new index1="1" />
   </alias>

   <solution solution_id="telemetry" name="SITE_FRAME" index1="2">
          <reference_frame name="SITE_FRAME" index1="1" />
          <offset x="0.0" y="0.0" z="0.0" />
          <orientation s="1.0" v1="0.0" v2="0.0" v3="0.0" />
   </solution>

   <alias>
          <old index1="1" index2="9" index3="3" index4="45" index5="2" />
          <new index1="2" />
   </alias>

   <alias>
          <old index1="2" index2="6" index3="0" index4="0" index5="0" />
          <new index1="3" />
   </alias>

   <solution solution_id="SSTB1_001" name="SITE_FRAME" index1="3">
          <reference_frame name="SITE_FRAME" index1="2" />
          <offset x="-1.34588" y="-2.31962" z="0.30" />
          <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547" />
   </solution>

</rmc_file>
```

APPENDIX D – Example Master Rover Vector File "SSTB1_Site_2_Master_00003.rvf"


```
<rmc_file mission="SSTB1" variant="Master_RVF" index1="2">

  <priority>
      <entry solution_id="telemetry" />
      <entry solution_id="SSTB1_001" />
      <entry solution_id="SSTB1_002" />
  </priority>

  <solution solution_id="telemetry" name="ROVER_FRAME" add_date="2003-03-25T23:20:00Z" index1="2"
          index2="0">
      <reference_frame name="SITE_FRAME" index1="2" />
      <offset x="0.0" y="0.0" z="0.0" />
      <orientation s="0.493547" v1="0.0131355" v2="0.0173344" v3="-0.869447"/>
  </solution>

  <solution solution_id="telemetry" name="ROVER_FRAME" add_date="2003-03-25T23:20:00Z" ndex1="2"
          index2="6">
      <reference_frame name="SITE_FRAME" index1="2" />
      <offset x="-1.34588" y="-2.31962" z="0.213165" />
      <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
  </solution>

  <solution solution_id="SSTB1_001" name="ROVER_FRAME" add_date="2003-03-27T14:15:00Z"
          index1="2" index2="6">
      <reference_frame name="SITE_FRAME" index1="2" />
      <offset x="-1.34588" y="-2.31962" z="0.35" />
      <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
      <derivation id="mipl_rgd_egress-drive-fix_1" />
  </solution>

  <solution solution_id="SSTB1_002" name="ROVER_FRAME" add_date="2003-03-27T14:56:00Z"
          index1="2" index2="6">
      <reference_frame name="SITE_FRAME" index1="2" />
      <offset x="-1.34588" y="-2.31962" z="0.30" />
      <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
      <derivation id="mipl_rgd_egress-drive-fix_2" />
  </solution>

</rmc_file>
```

APPENDIX E – Example Daily Rover Vector File "SSTB1_Site_2_Sol_43_Daily_001.rvf"


```
<rmc_file mission="SSTB1" variant="Daily_RVF" index1="2">

  <priority>
        <entry solution_id="telemetry" />
        <entry solution_id="SSTB1_001" />
        <entry solution_id="SSTB1_002" />
  </priority>

  <solution solution_id="telemetry" name="ROVER_FRAME" index1="2" index2="0">
        <reference_frame name="SITE_FRAME" index1="2" />
        <offset x="0.0" y="0.0" z="0.0" />
        <orientation s="0.493547" v1="0.0131355" v2="0.0173344" v3="-0.869447"/>
  </solution>

  <solution solution_id="telemetry" name="ROVER_FRAME" index1="2" index2="6">
        <reference_frame name="SITE_FRAME" index1="2" />
        <offset x="-1.34588" y="-2.31962" z="0.213165" />
        <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
  </solution>

</rmc_file>
```

APPENDIX F – Example Daily Rover Vector File "SSTB1_Site_2_Sol_45_Daily_001.rvf"


```
<rmc_file mission="SSTB1" variant="Daily_RVF" index1="2">

  <priority>
        <entry solution_id="telemetry" />
        <entry solution_id="SSTB1_001" />
        <entry solution_id="SSTB1_002" />
  </priority>

  <solution solution_id="telemetry" name="ROVER_FRAME" index1="2" index2="0">
        <reference_frame name="SITE_FRAME" index1="2" />
        <offset x="0.0" y="0.0" z="0.0" />
        <orientation s="0.493547" v1="0.0131355" v2="0.0173344" v3="-0.869447"/>
  </solution>

  <solution solution_id="SSTB1_002" name="ROVER_FRAME" index1="2" index2="6">
        <reference_frame name="SITE_FRAME" index1="2" />
        <offset x="-1.34588" y="-2.31962" z="0.30" />
        <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
  </solution>

</rmc_file>
```

APPENDIX G – Example Master Rover Vector File "SSTB1_Site_3_Master_00001.rvf"

```
<rmc_file mission="SSTB1" variant="Master_RVF" index1="3">

   <priority>
         <entry solution_id="telemetry" />
         <entry solution_id="SSTB1_001" />
         <entry solution_id="SSTB1_002" />
   </priority>

   <solution solution_id="telemetry" name="ROVER_FRAME" add_date="2003-03-25T23:20:00Z" index1="3"
               index2="0">
         <reference_frame name="SITE_FRAME" index1="3" />
         <offset x="0.0" y="0.0" z="0.0" />
         <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
   </solution>

</rmc_file>
```

APPENDIX H – Example Daily Rover Vector File "SSTB1_Site_3_Sol_43_Daily_001.rvf"

```
<rmc_file mission="SSTB1" variant="Daily_RVF" index1="3">

  <priority>
        <entry solution_id="telemetry" />
        <entry solution_id="SSTB1_001" />
        <entry solution_id="SSTB1_002" />
  </priority>

  <solution solution_id="telemetry" name="ROVER_FRAME" index1="3" index2="0">
        <reference_frame name="SITE_FRAME" index1="3" />
        <offset x="0.0" y="0.0" z="0.0" />
        <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
  </solution>

</rmc_file>
```

APPENDIX I – Example Daily Rover Vector File "SSTB1_Site_3_Sol_45_Daily_001.rvf"


```
<rmc_file mission="SSTB1" variant="Daily_RVF" index1="3">

   <priority>
         <entry solution_id="telemetry" />
         <entry solution_id="SSTB1_001" />
         <entry solution_id="SSTB1_002" />
   </priority>

   <solution solution_id="telemetry" name="ROVER_FRAME" index1="3" index2="0">
         <reference_frame name="SITE_FRAME" index1="3" />
         <offset x="0.0" y="0.0" z="0.0" />
         <orientation s="0.493609" v1="0.013832" v2="0.00689677" v3="-0.869547"/>
   </solution>

</rmc_file
```

APPENDIX J – RMC File Format

Element	Subelement	Attribute	Type	Count	Examples	PDS Label	Comments
rmc_file				1			
		mission	string	1	MER_1	INSTRUMENT_HOST_ID	
		variant	string	0,1	Master_SVF, Daily_SVF, Master_RVF, Daily_RVF		Specifies if this file conforms to one of the defined RMC variants.
		index1	string	0-1			If present, guarantees all definitions in file have this index1 (site).
	origination			0-n		No PDS label	Match ID with solution. If multiple w/same ID, use last one.
	solution			0-n		COORDINATE_SYSTEM_STATE group	Contain one solution for a single RMC tuple
	alias			0-n		No PDS label	Specifies equivalent RMC's when a Site is declared
	priority			0-1		No PDS label	Priority-ordered list of ID's in this file
solution						COORDINATE_SYSTEM_STATE group	Contain one solution for a single RMC tuple
		id	string	1		SOLUTION_ID	Solution ID.
		name	string	1	ROVER_FRAME	COORDINATE_SYSTEM_NAME	The CS defined by this group.
		index1	int	0-1		COORDINATE_SYSTEM_INDEX(1)	Site for MER.
		index2	int	0-1		COORDINATE_SYSTEM_INDEX(2)	Drive for MER. Requires index1.
		index3	int	0-1		COORDINATE_SYSTEM_INDEX(3)	IDD for MER. Requires index2. May be -1 for wildcard.
		index4	int	0-1		COORDINATE_SYSTEM_INDEX(4)	PMA for MER. Requires index3. May be -1 for wildcard.
		index5	int	0-1		COORDINATE_SYSTEM_INDEX(5)	HGA for MER. Requires index4. May be -1 for wildcard.
		index6	int	0-1		COORDINATE_SYSTEM_INDEX(6)	Reserved for MER (for manual jiggle). Req's index5, may be -1.
		add_date	string	0-1	2003-03-07T01:04:51Z		Date/time (PDS fmt) item added to THIS file. Master files only.
	reference_frame			1			Reference frame in which this solution is expressed.

Element	Subelement	Attribute	Type	Count	Examples	PDS Label	Comments
	offset			0-1			Location of rover w.r.t parent. Optional if identical to parent
	orientation			0-1			Orientation of rover. Optional if identical to parent
	derivation			0-1			For master file only, gives history of how soln was derived.
reference_frame						COORDINATE_SYSTEM_STATE group	Points to a CS; doesn't define one. Same as pointing corr. file
		name	string	1	SITE_FRAME	REFERENCE_COORD_SYSTEM_NAME	Frame name of CS
		index1	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(1)	Site for MER
		index2	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(2)	Drive for MER. Requires index1
		index3	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(3)	IDD for MER. Requires index2
		index4	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(4)	PMA for MER. Requires index3
		index5	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(5)	HGA for MER. Requires index4
		index6	int	0-1		REFERENCE_COORD_SYSTEM_INDEX(6)	Reserved for MER (for manual jiggle). Requires index5
offset						COORDINATE_SYSTEM_STATE group	Contains one XYZ position offset.
		x	float	1		ORIGIN_OFFSET_VECTOR(1)	
		y	float	1		ORIGIN_OFFSET_VECTOR(2)	
		z	float	1		ORIGIN_OFFSET_VECTOR(3)	
orientation						COORDINATE_SYSTEM_STATE group	Contains one orientation RELATIVE to reference frame
		s	float	1		ORIGIN_ROTATION_QUATERNION(1)	
		v1	float	1		ORIGIN_ROTATION_QUATERNION(2)	
		v2	float	1		ORIGIN_ROTATION_QUATERNION(3)	
		v3	float	1		ORIGIN_ROTATION_QUATERNION(4)	
derivation							Original ref CS before going in master, if not same as current ref
		id	string	1			Original solution ID from source file

Element	Subelement	Attribute	Type	Count	Examples	PDS Label	Comments
	reference_frame			0-1			Original reference frame if not the same as the current ref
	offset			0-1			Original location w.r.t. original reference frame
	orientation			0-1			Original orientation w.r.t. original reference frame
origination						No PDS label	Tells where this solution came from. Same as pointing corr file.
		id	string	1			Which solution does this apply to.
		user	string	0-1			Actual attribute set TBD
		institution	string	0-1			Actual attribute set TBD
		program	string	0-1			Actual attribute set TBD
	purpose		text	0-1			Contains optional text description of the solution
alias							Specifies equivalent RMC's when a Site is declared
	old			1		No PDS label	The "old" value (lower numbers)
	new			1		No PDS label	The "new" value (higher numbers)
old						No PDS label	
		index1	int	0-1			Site for MER
		index2	int	0-1			Drive for MER. Requires index1
		index3	int	0-1			IDD for MER. Requires index2
		index4	int	0-1			PMA for MER. Requires index3
		index5	int	0-1			HGA for MER. Requires index4
		index6	int	0-1			Reserved for MER (for manual jiggle). Requires index5
new						No PDS label	
		index1	int	0-1			Site for MER
		index2	int	0-1			Drive for MER. Requires index1. Usually not used.
		index3	int	0-1			IDD for MER. Requires index2. Usually not used.
		index4	int	0-1			PMA for MER. Requires index3. Usually not used.
		index5	int	0-1			HGA for MER. Requires index4. Usually not used.
		index6	int	0-1			Reserved for MER. Requires index5. Usually not used.
priority						No PDS label	Priority-ordered list of ID's in this file

Element	Subelement	Attribute	Type	Count	Examples	PDS Label	Comments
	entry			0-n			An individual ID. The first to apear is the lowest priority
entry						No PDS label	An individual ID. The first to apear is the lowest priority
		id	string	1			The actual ID.

APPENDIX K – RMC File XML Schema


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jeffrey Norris (Jet Propulsion Laboratory) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="rmc_file">
    <xs:annotation>
      <xs:documentation>Contains prioritized solutions and aliases</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="origination">
          <xs:annotation>
            <xs:documentation>Tells where this solution came from.  Same as pointing corr file.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="purpose" type="xs:string" minOccurs="0"/>
            </xs:sequence>
            <xs:attribute name="solution_id" type="xs:string" use="required"/>
            <xs:attribute name="user" type="xs:string" use="optional"/>
            <xs:attribute name="institution" type="xs:string" use="optional"/>
            <xs:attribute name="program" type="xs:string" use="optional"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="alias">
          <xs:annotation>
            <xs:documentation>Specifies equivalent RMC's when a Site is declared</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:sequence>
              <xs:element name="old">
                <xs:complexType>
                  <xs:attribute name="index1" type="xs:integer" use="optional"/>
                  <xs:attribute name="index2" type="xs:integer" use="optional"/>
                  <xs:attribute name="index3" type="xs:integer" use="optional"/>
                  <xs:attribute name="index4" type="xs:integer" use="optional"/>
                  <xs:attribute name="index5" type="xs:integer" use="optional"/>
                  <xs:attribute name="index6" type="xs:integer" use="optional"/>
```

```
                </xs:complexType>
            </xs:element>
            <xs:element name="new">
                <xs:complexType>
                    <xs:attribute name="index1" type="xs:integer" use="optional"/>
                    <xs:attribute name="index2" type="xs:integer" use="optional"/>
                    <xs:attribute name="index3" type="xs:integer" use="optional"/>
                    <xs:attribute name="index4" type="xs:integer" use="optional"/>
                    <xs:attribute name="index5" type="xs:integer" use="optional"/>
                    <xs:attribute name="index6" type="xs:integer" use="optional"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="solution">
    <xs:annotation>
        <xs:documentation>Contain one solution for a single RMC tuple</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="reference_frame" type="reference_frame">
                <xs:annotation>
                    <xs:documentation>Points to a CS; doesn't define one.  Same as pointing corr. file</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="offset" type="offset" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Contains one XYZ position offset.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="orientation" type="orientation" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Contains one orientation RELATIVE to reference frame</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="derivation" minOccurs="0">
                <xs:annotation>
                    <xs:documentation>Original ref CS before going in master, if not same as current ref</xs:documentation>
                </xs:annotation>
```

```
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="reference_frame" type="reference_frame" minOccurs="0"/>
                        <xs:element name="offset" type="offset" minOccurs="0"/>
                        <xs:element name="orientation" type="orientation" minOccurs="0"/>
                    </xs:sequence>
                    <xs:attribute name="solution_id" type="xs:string" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="solution_id" type="xs:string" use="required"/>
        <xs:attribute name="name" type="xs:string" use="required"/>
        <xs:attribute name="index1" type="xs:integer" use="optional"/>
        <xs:attribute name="index2" type="xs:integer" use="optional"/>
        <xs:attribute name="index3" type="xs:integer" use="optional"/>
        <xs:attribute name="index4" type="xs:integer" use="optional"/>
        <xs:attribute name="index5" type="xs:integer" use="optional"/>
        <xs:attribute name="index6" type="xs:integer" use="optional"/>
        <xs:attribute name="add_date" type="xs:string" use="optional"/>
    </xs:complexType>
</xs:element>
<xs:element name="priority">
    <xs:annotation>
        <xs:documentation>Priority-ordered list of ID's in this file</xs:documentation>
    </xs:annotation>
    <xs:complexType>
        <xs:sequence>
            <xs:element name="entry" minOccurs="0" maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>An individual ID.  The first to apear is the highest priority</xs:documentation>
                </xs:annotation>
                <xs:complexType>
                    <xs:attribute name="solution_id" type="xs:string" use="required"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:choice>
<xs:attribute name="mission" type="xs:string" use="required"/>
```

```
        <xs:attribute name="variant" type="xs:string" use="optional"/>
        <xs:attribute name="index1" type="xs:string" use="optional"/>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="offset">
      <xs:annotation>
        <xs:documentation>Contains one XYZ position offset.</xs:documentation>
      </xs:annotation>
      <xs:attribute name="x" type="xs:float" use="required"/>
      <xs:attribute name="y" type="xs:float" use="required"/>
      <xs:attribute name="z" type="xs:float" use="required"/>
    </xs:complexType>
    <xs:complexType name="orientation">
      <xs:annotation>
        <xs:documentation>Contains one orientation RELATIVE to reference frame</xs:documentation>
      </xs:annotation>
      <xs:attribute name="s" type="xs:float" use="required"/>
      <xs:attribute name="v1" type="xs:float" use="required"/>
      <xs:attribute name="v2" type="xs:float" use="required"/>
      <xs:attribute name="v3" type="xs:float" use="required"/>
    </xs:complexType>
    <xs:complexType name="reference_frame">
      <xs:annotation>
        <xs:documentation>Points to a CS; doesn't define one.  Same as pointing corr. file</xs:documentation>
      </xs:annotation>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="index1" type="xs:integer" use="optional"/>
      <xs:attribute name="index2" type="xs:integer" use="optional"/>
      <xs:attribute name="index3" type="xs:integer" use="optional"/>
      <xs:attribute name="index4" type="xs:integer" use="optional"/>
      <xs:attribute name="index5" type="xs:integer" use="optional"/>
      <xs:attribute name="index6" type="xs:integer" use="optional"/>
    </xs:complexType>
  </xs:schema>
```