Annex 13:

Software specification
# Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/03/2004</td>
<td>1.0</td>
<td>Draft #1, Partial (Issuance and Data Model)</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>31/03/2004</td>
<td>1.0</td>
<td>Draft #2, Partial</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>20/04/2004</td>
<td>1.0</td>
<td>Draft #3</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>25/05/2004</td>
<td>1.0</td>
<td>Draft #4</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>06/07/2004</td>
<td>1.0</td>
<td>Draft #5</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>03/09/2004</td>
<td>1.0</td>
<td>Draft #6</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>18/05/2005</td>
<td>1.0</td>
<td>Draft #7</td>
<td>Andrew Howard</td>
</tr>
<tr>
<td>14/10/2005</td>
<td>1.0</td>
<td>Draft #8</td>
<td>Andrew Howard</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** .......................................................................................................................... 1
   1.1 Purpose .................................................................................................................................. 1
   1.2 Intended Audience .................................................................................................................. 1
   1.3 Scope .................................................................................................................................... 1
   1.4 Definitions, Acronyms, Abbreviations, and Terminology ......................................................... 1
   1.5 Derivation Documents ............................................................................................................ 1
2. **Assumptions and Standards** .................................................................................................... 3
   2.1 Assumptions ........................................................................................................................... 3
   2.2 Standards ............................................................................................................................... 3
3. **General Requirements** .......................................................................................................... 4
   3.1 Technical Architecture Overview .......................................................................................... 4
   3.2 Message Exchange and Transaction Verification ..................................................................... 5
   3.3 Communications Hub and Message Queues ......................................................................... 6
   3.3.1 Transaction Queue ............................................................................................................ 6
   3.3.2 STL Account Management Queue ................................................................................... 6
   3.3.3 Reference Information Queue ......................................................................................... 6
   3.3.4 Information Request Queue ............................................................................................. 7
   3.4 Database Model .................................................................................................................... 7
   3.4.1 Registry Submodel ............................................................................................................. 7
   3.4.2 Transaction Process Submodel .......................................................................................... 7
   3.4.3 Reconciliation Submodel .................................................................................................. 8
   3.4.4 System Data Submodel ..................................................................................................... 8
   3.4.5 Projects and Notifications Submodel ............................................................................... 9
   3.4.6 User accounts, roles and privileges .................................................................................. 10
   3.5 General Flow of a Transaction Message ............................................................................... 10
4. **Technical Architecture Specification** .................................................................................... 13
5. **Transaction Processing** ......................................................................................................... 14
   5.1 Scope of Software Specification for Data Exchange Processes .............................................. 14
   5.2 Design Elements ................................................................................................................... 14
   5.2.1 Database Model ................................................................................................................. 14
   5.2.2 UML and Process Flow Diagrams ..................................................................................... 14
   5.2.3 Functions and Objects ....................................................................................................... 15
   5.3 Summary of Transaction Types .............................................................................................. 15
   5.3.1 Issuance (Transaction Type 1) .......................................................................................... 15
   5.3.2 Conversion (Transaction Type 2) ....................................................................................... 16
   5.3.3 External Transfer Process (Transaction Type 3) ................................................................ 16
   5.3.4 Internal Transfers Involving Cancellations and Retirements (Transaction Types 4 and 5) ...... 16
   5.3.5 Internal Transfers Involving Replacement of tCERs and tCERs (Transaction Type 6) .......... 16
   5.3.6 Carry-overs (Transaction Type 7) ..................................................................................... 16
   5.3.7 Expiry Date Change (Transaction Type 8) ......................................................................... 16
   5.3.8 Internal Transfers and Other Supplementary Transactions Routed to STL (Transaction Type 10) ........................................................................................................ 17
5.4 Transaction Message Checks .................................................................................................... 17
   5.4.1 Check Phase ....................................................................................................................... 18
   5.4.2 Version and Authentication Checks ................................................................................... 18
   5.4.3 Message Viability Checks .................................................................................................. 19
   5.4.4 Registry Checks ................................................................................................................ 19
   5.4.5 Data Integrity Checks for Transactions .............................................................................. 20
   5.4.6 Message Sequence Checks for Transactions from Registries ............................................. 21
   5.4.7 Message Sequence Checks for Transactions from STLs .................................................. 23
5.4.8 General Transaction Checks .............................................................. 23
5.4.9 Transaction-specific Checks .......................................................... 24
5.4.10 Registry Messages ................................................................. 32
5.5 Running Totals and Related Limits on Transactions .............................. 32
5.5.1 Unit Type Issuance Limits ......................................................... 33
5.5.2 Unit Type Issuance Levels ......................................................... 33
5.5.3 Project Issuance Level ............................................................... 33
5.5.4 Converted Level ................................................................. 33
5.5.5 Reserve Limit (not a running total) ................................................ 34
5.5.6 Commitment Period Holdings .................................................... 34
5.5.7 Track 2 ERU First-time Transfer Level .......................................... 34
5.5.8 Adjusted Commitment Period Holdings ....................................... 34
5.5.9 Retirement Limit (not a running total) ........................................... 35
5.5.10 Retirement Level ............................................................... 35
5.5.11 Carry-over Limits (not running totals) ........................................... 35
5.5.12 Carry-over Levels ............................................................... 35
5.6 Activity Diagrams ........................................................................ 35
5.6.1 Basic Transaction Activity Diagram ............................................. 36
5.6.2 External Transfer Activity Diagram ............................................. 37
5.7 Transaction Flow Diagrams ............................................................ 39
5.8 Transaction State Diagram ............................................................. 47
6. Reconciliation Process ....................................................................... 49
6.1 Reconciliation Snapshot Data .......................................................... 49
6.2 Reconciliation Message Checks ....................................................... 50
6.2.1 Version and Authentication Checks for Reconciliation .................... 51
6.2.2 Message Viability Checks for Reconciliation ................................ 51
6.2.3 Registry Validation Checks for Reconciliation ............................... 51
6.2.4 Data Integrity Checks for Reconciliation ....................................... 52
6.2.5 Message Sequence Checks for Reconciliation Messages Received from Registries .................................................. 52
6.2.6 Message Sequence Checks for Reconciliation Messages Received from STL .................................................. 53
6.2.7 Other Reconciliation Responses .................................................. 54
6.3 Activity Diagrams ........................................................................ 55
6.4 Reconciliation Processing Flow Diagrams .......................................... 58
6.5 Reconciliation State Diagrams ........................................................ 67
7. Administrative Processes .................................................................... 69
7.1 Transaction Status .......................................................................... 69
7.2 Transaction Clean-up Process .......................................................... 69
7.3 Notifications ................................................................................ 69
7.3.1 Net Source Cancellation .......................................................... 69
7.3.2 Non-compliance Cancellation ................................................... 71
7.3.3 Impending tCER or lCER Expiry ................................................ 71
7.3.4 Reversal of Storage for CDM Project .......................................... 71
7.3.5 Non-submission of Certification Report for CDM Project ............. 72
7.3.6 Excess Issuance for CDM Project .............................................. 72
7.3.7 Commitment Period Reserve Change ......................................... 72
7.3.8 Unit Carry-over ...................................................................... 73
7.3.9 Notification Update .................................................................. 73
7.4 Registry Time Synchronization .......................................................... 73
7.5 General Messages ......................................................................... 74
7.6 Yearly Aggregated Holdings Report .................................................. 74
<table>
<thead>
<tr>
<th>Annex</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>ITL System Interfaces</td>
<td>A-1</td>
</tr>
<tr>
<td>B</td>
<td>Database Entity Relationship Diagrams</td>
<td>B-1</td>
</tr>
<tr>
<td>C</td>
<td>Data Element Definitions</td>
<td>C-1</td>
</tr>
<tr>
<td>D</td>
<td>Lookup Tables and Values</td>
<td>D-1</td>
</tr>
<tr>
<td>E</td>
<td>Functions and Components</td>
<td>E-1</td>
</tr>
<tr>
<td>F</td>
<td>List of Transaction and Reconciliation Checks</td>
<td>F-1</td>
</tr>
<tr>
<td>G</td>
<td>ITL Administrator Application</td>
<td>G-1</td>
</tr>
<tr>
<td>H</td>
<td>UNFCCC Web Site Style Guide</td>
<td>H-1</td>
</tr>
</tbody>
</table>
List of Figures

Figure 3.1: Overview of Technical Architecture ................................................................. 5
Figure 3.2: General Flow of a Transaction Message through ITL Processing .................... 12
Figure 5.1: Key to UML Diagrams ...................................................................................... 15
Figure 5.2: Check Categories ............................................................................................. 17
Figure 5.3: Version and Authentication Checks .................................................................. 19
Figure 5.4: Message Viability Checks ................................................................................. 19
Figure 5.5: Registry Checks ............................................................................................... 20
Figure 5.6: Summary of Data Integrity Checks .................................................................. 20
Figure 5.7: Sequence Checks for Transactions from Registries ........................................ 22
Figure 5.8: Sequence Checks for STL Messages ................................................................. 23
Figure 5.9: General Transaction Checks (Not Applied to Issuance Transactions) ............... 23
Figure 5.10: Transaction-specific Checks .......................................................................... 25
Figure 5.11: Registry Messages ......................................................................................... 32
Figure 5.12: Simple (Non-STL) Transaction Process .......................................................... 36
Figure 5.14: Preliminary Processing ................................................................................... 39
Figure 5.15: Queue Processing Checks ............................................................................... 40
Figure 5.16: Evaluate Transaction ..................................................................................... 41
Figure 5.17: Validate Proposal ........................................................................................... 42
Figure 5.18: Determine Route for Proposal ......................................................................... 43
Figure 5.19: Process Non-external Transaction Notification ............................................... 44
Figure 5.20: Process External Transaction Notification ...................................................... 45
Figure 5.21: Finalise Transaction ....................................................................................... 46
Figure 5.22: Evaluate STL Notification .............................................................................. 47
Figure 5.23: Transaction State Diagram ............................................................................ 48
Figure 6.1: Reconciliation Check Categories ..................................................................... 50
Figure 6.2: Additional Registry Checks for Reconciliation ................................................ 51
Figure 6.3: Summary of Reconciliation Data Integrity Checks ............................................ 52
Figure 6.4: Sequence Checks for Registry Messages .......................................................... 53
Figure 6.5: Sequence Checks for STL Messages ................................................................. 53
Figure 6.6: Other Reconciliation Checks and Messages ...................................................... 54
Figure 6.7: Reconciliation Process Flow Stage 1 - Validate Account Totals ......................... 55
Figure 6.8: Reconciliation Process Flow Stage 2 - Validate Unit Blocks .............................. 56
Figure 6.9: Reconciliation Process Flow Stage 3 - Review Audit Logs ................................ 57
Figure 6.10: Confirm Reconciliation ................................................................................... 58
Figure 6.11: Receive Registry Reconciliation Message ........................................................ 59
Figure 6.12: Check Reconciliation Message ....................................................................... 60
Figure 6.13: Evaluate Registry Reconciliation Message ..................................................... 61
Figure 6.14: Validate Totals ............................................................................................... 62
Figure 6.15: Validate Unit Blocks ....................................................................................... 63
Figure 6.16: Receive Audit Trail ......................................................................................... 64
Figure 6.17: Manual Intervention ....................................................................................... 64
Figure 6.18: Receive STL Reconciliation Notice ................................................................. 65
Figure 6.19: Evaluate STL Reconciliation Notice ............................................................... 66
Figure 6.20: Registry - ITL Reconciliation State ................................................................. 67
Figure 6.21: ITL - STL Reconciliation State ...................................................................... 68
Figure 7.1: Transaction Clean-up Process ........................................................................... 70
1. Introduction

1.1 Purpose

This document contains the software specification for the international transaction log (ITL) to be developed and implemented by the Secretariat to the United Nations Framework Convention on Climate Change (UNFCCC). The purpose of the ITL is to monitor the validity of the transactions conducted by registries established by Parties under the mechanisms defined in Articles 6, 12 and 17 of the Kyoto Protocol and the modalities for the accounting of assigned amounts under Article 7.4 of the Kyoto Protocol.

The design of the ITL provides for the complementary functioning of supplementary transaction logs (STLs) developed by groups of Parties under the Kyoto Protocol. Such STLs are to conduct additional activities in relation to the transactions of those Parties under the Kyoto Protocol and under regional trading schemes. This complementary functioning is designed to avoid the duplication of validity checks and ensure consistent results between transaction logs. It further serves to integrate electronic communications between the relevant registries.

At time of writing, the only STL is the Community Independent Transaction Log (CITL) for the European Union emissions trading scheme. This is has been implemented under Article 20 of EU Directive 2003/87/EC.

1.2 Intended Audience

This document is intended primarily for the technical experts involved in development and implementation of the ITL. It will also be of relevance to experts developing STLs, registries and other data systems which will connect to the ITL.

1.3 Scope

This document details the software specification of the ITL. It does not address the design of STLs or other data systems to be connected to the ITL. However, where applicable, specifications are provided for the ITL side of the data processing, relevant to STLs and the transmission and receipt of electronic communications to and from STLs.

Furthermore, this document does not address the design of registries. It relies upon the technical specifications of the data exchange standards (DES) for details relating to the ITL processing of data relevant to registries and for transmission and receipt of electronic communications to and from registries (see Annex 14).

1.4 Definitions, Acronyms, Abbreviations, and Terminology

See Section 1.4 and Annex A of the Data Exchange Standards for Registry Systems under the Kyoto Protocol: Technical Specification (Version 1.0, Draft # 7, as referred to below).

1.5 Derivation Documents

- Data Exchange Standards for Registry Systems under the Kyoto Protocol: Functional Specifications (Version 1.0)
  → http://unfccc.int/sessions/workshop/281103/documents.html
• Data Exchange Standards for Registry Systems under the Kyoto Protocol: Technical Specification (Version 1.0, Draft #7)
  → http://unfccc.int/meetings/unfccc_calendar/pre-sessional/items/3413.php

• Decisions 15-18/CP.7 on the mechanisms under the Kyoto Protocol
  → Document FCCC/CP/2001/13/Add.2
  → http://unfccc.int/resource/docs/cop7/13a02.pdf

• Decision 19/CP.7 containing general requirements for the ITL and registries and modalities for the accounting of assigned amounts under the Kyoto Protocol
  → Document FCCC/CP/2001/13/Add.2
  → http://unfccc.int/resource/docs/cop7/13a02.pdf

• Decision 24/CP.8 containing general design requirements for the data exchange standards
  → Document FCCC/CP/2002/7/Add.3
  → http://unfccc.int/resource/docs/cop8/07a03.pdf

• Decision 19/CP.9 on the modalities and procedures for afforestation and reforestation project activities under the clean development mechanism in the first commitment period of the Kyoto Protocol
  → Document FCCC/CP/2003/6/Add.2
  → http://unfccc.int/resource/docs/cop9/06a02.pdf
2. **Assumptions and Standards**

2.1 **Assumptions**

This document is based upon the derivation documents specified in Section 1.5. In particular, it is based upon the constraints and specifications contained in the Technical Specification for Data Exchange Standards. This document assumes that the ITL and STLs will operate in a coordinated manner. Critical assumptions from this document include:

- Communication between the ITL and registries will occur over a hardware VPN;

- Communication between the ITL and registries will occur in real time to the extent possible;

- The ITL will have a Communications Hub that, where applicable, routes communications to the relevant STL;

- The ITL will contain information on CDM Projects provided by the CDM Executive Board and on Joint Implementation Projects (JI Projects) provided by the Article 6 Supervisory Committee; and

- The ITL will contain information from the compilation and accounting database maintained by the UNFCCC Secretariat (for example, on allowable Issuance levels and the eligibility of Parties to participate in the mechanisms under the Kyoto Protocol.

Additionally it is assumed that the ITL will be operated in a data centre outside the Secretariat. There will be a fixed private data line between the ITL and the Secretariat premises. It will be used by all data systems in the Secretariat connected to the ITL.

2.2 **Standards**

This document is based upon the following standards:

- SOAP
  
  http://www.w3.org/TR/2000/NOTE-SOAP-20000508

- XML
  
  http://www.w3.org/TR/2000/REC-xml-20001006

- IDEF1X Database Standards
  
  http://www.itl.nist.gov/fipspubs/idef1x.doc

- WSDL
  
  http://www.w3.org/TR/wsdl
3. **General Requirements**

The principal functionality of the ITL is to route and verify transactions that are received from national registries and the CDM Registry. These communications must be secure and processed as real-time transactions. The functional and technical requirements for data exchange standards specify the use of Web services sending encrypted messages. Communications must be protected from modification or interception in transit. Additionally, all messages from data systems (registries, STLs, Secretariat systems) must be authenticated.

Communications can be initiated by either a registry or the ITL with an immediate response expected. These communications are transported through the use of Web services. Some actions executed by registry Web services are asynchronous and processed in real time and others may be processed as jobs. The only immediate action that a Web service must respond to is an acknowledgement that the message was received and passed checks for authentication, and that the XML message format meets specifications. Calls to ITL Web services must be processed as soon as possible so that messages can be passed back to the initiating registry or forwarded on to another registry without delay. While registries must process a request from the ITL and send a responding message back within a 24-hour period, the ITL must process requests in a first-come, first-served fashion. If a message contains transaction information that involves a Party for which an STL has been established, the contents of that message are forwarded to the relevant STL for further processing.

The ITL also contains functionality to ensure the accuracy of data maintained by each registry and to ensure the consistency of transaction and unit information between the ITL and a registry. The process of comparing registry and ITL data and correcting data inconsistencies is called "reconciliation."

The Communications Hub of the ITL may contain additional Web services for supplementary programs. Where these independent Web services are necessary, the ITL Communications Hub will route messages directly to the STL upon receipt. The only checking performed will be to ensure that the message received is well formed and accurately identifies the recipient.

The ITL also has interfaces to data systems in the secretariat (Annex A).

Part of the ITL is the ITL Administrator Application (ITL AA) which is a Browser-based administrative tool that provides the ITL manager with the capability to manage back-end administrative processes (Annex G).

3.1 **Technical Architecture Overview**

To provide this functionality, the ITL requires:

- Web services and Simple Object Access Protocol (SOAP) for the transport and delivery of XML messages;
- Use of a hardware-based Virtual Private Network for IP authentication and decryption of messages; and
- Digital certificates to ensure authentication.
Figure 3.1 provides an overview of how messages are sent from registries to the ITL. The ITL’s VPN checks authentication; thereafter, the Communications Hub Web service receives and logs the incoming message. The ITL processes the transaction and routes the message on to an STL, if appropriate.

**Figure 3.1: Overview of Technical Architecture**

3.2 **Message Exchange and Transaction Verification**

The messages that are transported to and from the ITL must follow the data exchange standards that are the basis for the communication with the national registries of Annex I Parties, the CDM Registry and STLs. These standards provide for the implementation of a common data transfer format and common functionality to ensure accurate, transparent and efficient data transfer between registries and the ITL. The interfaces connecting other data systems to the ITL should also, to the extent possible, be consistent with the data exchange standards.
Transactions originating from Parties for which no STLs are applicable are not routed through an STL. Instead, messages are sent directly back to the registry or to another registry.

For transactions originating from Parties for which an STL is applicable, the ITL conducts its checks and routes messages to the relevant STL. The results of the checks performed by the STL are communicated back to the ITL. If no issues or discrepancies are identified by the STL, the originating registry may proceed with the transaction or the transaction proceeds to the Acquiring Registry. If the STL discovers a discrepancy preventing the transaction from being processed further, the ITL is given this information and it sends a message to the originating Registry that the transaction cannot be completed due to a specific error.

3.3 Communications Hub and Message Queues

The Communications Hub hosts message queues which process all incoming messages. The purpose of the inbound queues is to receive and store messages and to provide sufficient scalability and throughput during peak processing time. The Communications Hub must be able to receive and acknowledge messages even if the ITL application server or database server is not available. When messages are retrieved for processing they are checked for their timestamp of arrival. If this timestamp exceeds 24 hours from the current time, the message is rejected as having expired and the registry is notified that the message is invalid (see also DES Section 3.8).

The Communications Hub also hosts queues for outbound messages. Outbound queues are needed in case a destination data system is temporarily not available. A queue holds a message until it can be delivered to the destination, but a message can only stay in the queues for up to a certain time-limit (24 hours). During this time the Communications Hub resends the message until it receives a response from the destination, but at least 60 seconds have to pass between two attempts (see also DES Section 3.8). The time between attempts and the time-limit must be configurable. Message queues will be administered through the ITL Administrator Application or a 3rd party application (Annex G, Section 2.6).

There are at least four queues servicing incoming messages: a transaction queue, an account management queue, a reference information queue and a request for information queue.

3.3.1 Transaction Queue

The transaction queue manages high priority transaction proposals and subsequent notifications. This queue has the highest priority and is checked first for incoming messages.

3.3.2 STL Account Management Queue

STL Web service messages containing information on account data, including on installations and designated contacts and representatives, are held in a separate account management queue before being directed to the appropriate STL. The ITL does not record or track any information contained in such account management messages and only performs the role of routing the messages onward.

3.3.3 Reference Information Queue

Messages which contain reference information, needed by the ITL for its transaction checks, are held in a reference information queue. These messages will be send to the ITL from data systems in the Secretariat.
3.3.4 Information Request Queue

All other messages, which include reconciliation responses, transaction status requests and other administrative processes, are managed by this lowest priority queue.

3.4 Database Model

Data for the ITL must be maintained in a secure, relational, transactional database containing all relevant tables to hold all the data for supporting logs, registry holdings, transaction history, and reconciliation history. The entity relationship diagrams and data dictionary in Annex B and Annex C describe the logical database model.

The data model is presented in Annex B as five major submodels: Registry, Transaction Process, Reconciliation, Projects and Notifications, and System Data. Each submodel represents key relationships around a set of primary entities. The submodels have dependencies on other submodels.

3.4.1 Registry Submodel

The Registry Submodel contains information pertaining to the operations of each registry. Some elements of this information are input from the compilation and accounting database maintained by the UNFCCC Secretariat. The tables track and record the following information:

- Registry Web service URL and port for all operating environments (this information could be stored in configuration files instead of in the ITL database);
- Operational status and status history;
- Eligibility status of Parties to participate in the mechanisms under the Kyoto Protocol;
- Registry contact information and the type of relationship to, or responsibility a person has, for a registry (this information could be stored in files instead of in the ITL database);
- Allowable Issuance quantities or other unit restrictions; and
- Current unit holdings in each registry, by account type.

See Annex B, Figure B3 for detailed information on the Registry Submodel.

3.4.2 Transaction Process Submodel

Any transaction received by the ITL follows a sequence of processes, which are recorded in tables within the Transaction Process Submodel. These tables track and record the following information:

- Receipt of the message from a registry;
• Storage of incoming message;
• Logging of transaction and unit serial blocks in the transaction;
• Tracking of transaction status as various checks are performed;
• Recording of the appropriate responses to checks as applicable to each unit block;
• Identification of units in an ongoing transaction as "unavailable" until a transaction is finalised;
• Units that are currently in an inconsistent state as identified by a reconciliation process;
• Replacement of tCERs or lCERs;
• Expiry Date Changes for tCERs and lCERs; and
• Routing of transaction to an STL for further processing (if transaction involves party in a supplementary program.

See Annex B, Figure B4 for detailed information on the Transaction Process Submodel.

3.4.3 Reconciliation Submodel

Reconciliation occurs as a scheduled job determined by the ITL Administrator, as negotiated with each registry, or as requested by an STL. The reconciliation tables track and record the following information:

• Each instance in which the ITL requests reconciliation information from a registry;
• The date and time (DateTime) each reconciliation stage occurs, along with the status during each stage of processing;
• Logging of inconsistent unit blocks conflicting with information held in the Registry Unit Holdings table;
• Recording of response codes identifying the errors with the inconsistent blocks.

See Annex B, Figure B5 for detailed information on the Reconciliation Submodel.

3.4.4 System Data Submodel

Other tables record and retrieve system data that support the major processes. These tables track and record the following information:

• All inserts, updates, and deletes for all primary transaction tables (exclusive of the transaction and reconciliation logs);
• Current version number for the ITL Administration Application (ITL AA) and ITL Technical Specification for Data Exchange Standards (DES);

• All checks and the appropriate response codes associated with an error or successful check or similar process;

• List of reminders or other informational messages; and

• ITL system parameters that are used to record system defaults, toggles, and various other parameters.

See Annex B, Figure B6 for detailed information regarding these tables.

3.4.5 Projects and Notifications Submodel

Periodically, it will be necessary for the ITL to notify a registry of actions that need to be taken on units. Examples include Carry-over or Cancellation of a unit at the end of a Commitment Period, or notification from the ITL that a tCER or ICER will expire in 30 days. The ITL is also responsible e.g. for notifying registries regarding necessary actions stemming from CDM Project actions by the CDM Executive Board.

The projects and notification tables track the following information:

• CDM and JI Projects approved under the Kyoto Protocol;

• The content of each notification message;

• The Project, if any, related to the reason for the notification; and

• The registries that receive each notification and the number of units or specific unit blocks, if any, each registry must act upon.

See Annex B, Figure B7 for detailed information regarding these tables.
3.4.6 User accounts, roles and privileges

Every user must receive and use its own personal ITL account and password. System accounts are also needed but should only be used by privileged system users, e.g. the Database Administrator. It must be possible to define password policies regarding format, expiration and re-use which are enforced by the system.

Privileges for accessing data or executing functions must not be granted directly to a user account. Instead privileges must be granted to roles and subsequently roles must be granted to user accounts. It must be possible to assign more than one role to an account.

The creation and deletion of accounts, the changing of passwords and the assignment of roles to accounts must be done easily without requiring code changes. Also the number (granularity) of different privileges and roles and the assignment of privileges to roles should allow for flexibility.

The security system must be designed in a way that it can be transformed into an LDAP based identity management system.

User accounts will be administered through the ITL Administrator Application or a 3rd party application. See Section 1.2, 1.3, 2.1 and 2.5 of Annex G for additional requirements.

3.5 General Flow of a Transaction Message

All messages received by the ITL follow a set of standard processing procedures. Details are defined in the DES. Figure 3.3 shows the general outline of these steps.

1. The incoming message is checked for authentication. These checks are performed by the VPN which first blocks any message from an unidentified sender by examining the digital certificate for authentication. All messages are written to a directory in a file system for monitoring and later examination by the Operator. Files in this directory should be purged after a period of time. These files are different from the files created for archiving later on in the process. Appropriate naming standards must be defined to facilitate the examination.

2. If the authentication check passes, the message must match the WSDL structure as specified by the Web service on the ITL. The message must contain version identifiers which are consistent with the current version.

3. If the Web service accepts the message, it is saved to an incoming message file and is briefly placed in a message queue to be processed as quickly as possible. Each accepted message is retrieved from the message queue in time-received order and checked to determine that it is less than 24 hours old. If the message is valid, it is processed against a set of message validity, registry, data integrity and message sequence checks. These checks determine whether the registries are eligible and operating, and whether the contents of the message meet the minimum requirements to continue further processing. Each check returns a specific response code if an error is identified. These checks and response codes are described in Section 5.4.

4. After these checks have been performed, the name of the message file is recorded to a message log (which keeps a record of every incoming and outgoing message). Multiple XML files that are received can be zipped and stored in master files.
5. If at any point during these checks an error is discovered, a response is sent back to the registry indicating a failure and the reason for the failure.

6. Otherwise, the transaction is evaluated against general transaction checks and checks that are appropriate for the specific transaction type identified in the message. Each check returns a specific response code if an error is identified. These checks and response codes are described in Section 5.4.

7. The ITL evaluates the results of all transaction checks. If all prior checks have been performed without error by the ITL, and if the HTTP SOAP request involves a registry participating in a supplementary program, the contents of the request are forwarded to the appropriate STL for further processing. If the ITL finds discrepancies with the transaction during the transaction checking phases, the message will not be forwarded to the STL, but will be returned to the registry containing the appropriate response codes identifying what checks failed and the reason.

8. Throughout this process, the ITL records the transaction, the transaction status and response codes relating to the transaction. If there is an unhandled exception in the processing of the message, any changes made to the database for this transaction will be rolled back. Prior to the first database change after the message is retrieved from the queue, a new database transaction session will be started. Once the message has been evaluated, all the changes made during the session will be committed at once. If the commitment point is never reached, the changes will be rolled back.
Figure 3.2: General Flow of a Transaction Message through ITL Processing
4. Technical Architecture Specification

The Technical Architecture specification is out of the scope of this document. For requirements related to the Technical Architecture see Annex 11 and Annex 15.
5. **Transaction Processing**

5.1 **Scope of Software Specification for Data Exchange Processes**

This section of the Software Specification addresses the design and development of all functionality necessary to support validation of transactions by the ITL and all other data exchange between registries, the ITL, and an STL.

This section contains software specifications on the following functional processes:

- Issuance;
- Conversion;
- External Transfers;
- Internal Transfers (Cancellation and Retirement);
- Carry-over;
- Replacement; and
- Expiry Date Change.

5.2 **Design Elements**

The design of the data exchange processes for the ITL is presented in this document through the following elements:

- A relational database model;
- A logical data dictionary;
- UML and process flow diagrams; and
- Web service and function definitions.

5.2.1 **Database Model**

An overview of the database model is presented in Section 3 above, in the entity-relationship diagrams in Annex B and in the data dictionary in Annex C. The database is a core design element and is referenced throughout the checks and function descriptions.

5.2.2 **UML and Process Flow Diagrams**

This design documentation utilises the activity and process flow diagrams based on Unified Modelling Language (UML). These diagrams represent the functionality within the ITL which is necessary to review, process, respond to, and store all HTTP SOAP requests. The process flow diagrams are high-level representations which are designed to capture process logic. Within each diagram, specific functions (representing specific programming logic) are identified. These diagrams are annotated with text to help non-technical readers interpret them more easily.

The data process diagrams show business logic and decision structures. Some decisions can be functionally simple or very complex. When the functionality is complex, the box for that component on the behaviour diagram will contain a "ref" symbol to indicate that a detailed sub-diagram is provided. See Figure 5.1, Key to UML Diagrams, for reference.
For the purposes of this technical design document, the UML diagrams in Section 5.5.1 focus on those Web services and functions in the Communications Hub and ITL "swim lanes." References to the functions identified in the registry swim lanes can be found in the Data Exchange Standards for Registry Systems under the Kyoto Protocol.

**Figure 5.1: Key to UML Diagrams**

<table>
<thead>
<tr>
<th>UML Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actors and swim lanes</td>
<td>At the top of each diagram, the participants in the process are represented by a word preceded with a colon (:). Actions involving a participant are presented in the &quot;swim lane&quot; which is directly underneath the box, and represented by a dashed vertical line.</td>
</tr>
<tr>
<td>sd</td>
<td>This symbol indicates that the diagram is a sequence diagram. The symbol is followed by the name of the process.</td>
</tr>
<tr>
<td>ref</td>
<td>This symbol indicates that there is a secondary sub-diagram for the component which provides additional detail of the functionality.</td>
</tr>
</tbody>
</table>

### 5.2.3 Functions and Objects

An object is an entity in which various programming functions can be performed relating to a task. Each function requires specific information to be provided to perform its task ("inputs") and returns specific information after its operations are performed ("outputs"). In all cases, functions generally have objects as inputs and outputs. Although not all objects are identified in these software specifications, several key objects have been specified and are referenced in these functions. See Annex E for details.

These technical design specifications define the inputs, database interactions, and outputs for each process. The document describes in specific data terms the programming logic that should be implemented for all functions required by the ITL.

Technical information for each function, including required inputs and outputs, is included in Annex E.

### 5.3 Summary of Transaction Types

#### 5.3.1 Issuance (Transaction Type 1)

The Issuance of AAUs is undertaken by a Party in an account in its national registry on the basis of its assigned amount (which is in turn calculated on the basis of greenhouse gas emissions during the base year). The Issuance of RMUs is undertaken by a Party in its national registry on
the basis of its net removals of greenhouse gases through LULUCF activities. The Issuance of CERs, tCERs, or ICERs into a pending account is undertaken by the CDM Executive Board, in the CDM Registry, on the basis of certified reductions in greenhouse gas emissions or certified removals of greenhouse gases through a CDM Project activity. The validity of such Issuance is monitored and validated by the ITL.

5.3.2 Conversion (Transaction Type 2)

The Conversion of AAUs and RMUs to ERUs is undertaken by a Party in an account in its national registry. AAUs and RMUs are converted to ERUs on the basis of verified reductions in greenhouse gas emissions and verified removals of greenhouse gases through a JI Project. The validity of such Conversion is monitored and validated by the ITL.

5.3.3 External Transfer Process (Transaction Type 3)

The External transfer of AAUs, RMUs, ERUs, CERs, tCERs and ICERs to another registry is undertaken by a Party, an entity, or the CDM Executive Board, on the basis of the amount proposed by the transferor. The validity of such External transfers is monitored and validated by the ITL.

5.3.4 Internal Transfers Involving Cancellations and Retirements (Transaction Types 4 and 5)

The Internal transfer of AAUs, RMUs, ERUs, CERs, tCERs or ICERs to Voluntary, Net Source, Non-compliance or Excess Issuance Cancellation accounts is undertaken by a Party, an entity, or the CDM Executive Board, on the basis of the amounts proposed by the transferor. The Internal transfer of these units to a Retirement account is undertaken by a Party or an entity, on the basis of the amounts proposed by the transferor. The validity of such Internal transfers is monitored and validated by the ITL.

5.3.5 Internal Transfers Involving Replacement of tCERs and ICERs (Transaction Type 6)

The Replacement of tCERs or ICERs occurs through the internal transfer of AAUs, RMUs, ERUs, CERs, tCERs or ICERs to a Replacement account and is undertaken by a Party or an entity, on the basis of the amounts proposed by the transferor. The validity of such Replacement is monitored and validated by the ITL.

5.3.6 Carry-overs (Transaction Type 7)

The Carry-over of AAUs, ERUs and CERs is undertaken by a Party in an account in its national registry, on the basis of the amount of units in holding accounts (i.e., units that have not been cancelled or retired for that Commitment Period) after expiration of the additional period for fulfilling commitments (the "true-up period"). The units remain in the same account and the serial numbers remain unchanged. The effect of the Carry-over transaction is to give recognition, both within the registry and the ITL, to the validity of the units in the next Commitment Period. Any units in holding accounts that are not carried over in this manner are to be cancelled. The validity of such Carry-over is monitored and validated by the ITL.

5.3.7 Expiry Date Change (Transaction Type 8)

An Expiry Date Change is undertaken by a Party for tCERs and ICERs. For tCERs, this transaction may be necessary to change the expiry date of tCERs issued prior to 2006 if the end of
the second Commitment Period is determined not to be 31 December 2017 (the expiry date with which tCERs would initially be issued). For ICERs, this transaction will occur when the Executive Board approves the extension of ICERs for a Project for an additional period. The ITL ensures that these expiry date updates are consistent with the Project Approvals, and updates the tCER and ICER expiry dates in the ITL database.

5.3.8 Internal Transfers and Other Supplementary Transactions Routed to STL (Transaction Type 10)

The validity of Internal transfers of AAUs, RMUs, ERUs, CERs, tCERs or ICERs among holding accounts is not checked by the ITL, but may be checked by STLs. For these transactions, the ITL conducts general transaction checks necessary to mark the blocks as unavailable due to a pending transaction and splits unit blocks as necessary. The ITL records the results of this basic step and routes them to the relevant STL for further evaluation against STL rules and requirements.

5.4 Transaction Message Checks

As a message is received and processed, it is checked at various levels. Preliminary checks that fail return a failure response immediately back to the registry via a SOAP response. Other checks that may involve interaction with the database take longer and are not returned immediately. The action taken to respond to failure of these checks depends on the point in the process where the message failed. Failures due to transaction checks are returned in the ResponseObject in an HTTP SOAP response initiated by the ITL to the originating registry. See also Annex F. Figure 5.2 below summarizes the check categories, the types of responses, and check actions taken during processing. Additional Reconciliation message checks are listed in Section 6.2.

Figure 5.2: Check Categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Response Code Range</th>
<th>Category Description</th>
<th>Action Upon Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version and Authentication</td>
<td>1000 - 1299</td>
<td>Checks to authenticate sender and to validate version of DES during preliminary processing.</td>
<td>Message returned with response codes or HTTP Soap Error. Message not placed into message queue (unless only a minor version inconsistency is identified).</td>
</tr>
<tr>
<td>Message Viability</td>
<td>1300 - 1399</td>
<td>Checks to determine whether the message is viable when processed from the queue.</td>
<td>Message returned with response codes. Message not logged in the Transaction Log table.</td>
</tr>
</tbody>
</table>

(cont.)
### 5.4.1 Check Phase

The check phases referred to in the following sections indicate the nature and required timing of the check. Phase 1 indicates that the check is required by the data exchange standards, irrespective of the unit types subject to transactions (these checks have been implemented in the CITL for the first phase of the European Union emissions trading scheme, for which Kyoto units are not used); Phase 2 indicates that the check is required for units under the Kyoto Protocol prior to Commitment Period 1; Phase 3 indicates that the check is required for units under the Kyoto Protocol but will not be performed by the ITL prior to Commitment Period 1. Checks under all three phases are required for the ITL.

In the following sections, each check is described along with the appropriate response code number that is returned to the registry if the check fails. The detailed specification for these checks can be found in Annex F: List of Transaction Checks.

### 5.4.2 Version and Authentication Checks

Version and authentication checks are performed within the Communications Hub as preliminary checks upon receipt of the HTTP SOAP request and do not involve any interaction with the ITL database. Failures due to authentication checks and poorly formed XML content are returned as HTTP SOAP errors. All messages are written to a directory in a file system for monitoring and later examination by the Operator in case of problems. If these checks are passed, the message is placed in the message queue for processing. See Figure 5.14, Preliminary Processing, for an activity diagram showing the flow of a message.
Figure 5.3: Version and Authentication Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP error</td>
<td>Certificate Check</td>
<td>Certificate must be recognized.</td>
<td>1</td>
</tr>
<tr>
<td>SOAP error</td>
<td>SOAP Identifier</td>
<td>Initiating Registry must be consistent with sender of SOAP message.</td>
<td>1</td>
</tr>
<tr>
<td>SOAP error</td>
<td>WSDL Check</td>
<td>Message must conform to WSDL.</td>
<td>1</td>
</tr>
<tr>
<td>1031</td>
<td>Major Version</td>
<td>Major Version number in transaction message must match Major Version number for DES.</td>
<td>1</td>
</tr>
<tr>
<td>1032</td>
<td>Minor Version</td>
<td>Minor Version number in transaction message must match Minor Version number for DES.</td>
<td>1</td>
</tr>
</tbody>
</table>

5.4.3 Message Viability Checks

Messages are placed in different queues, depending on the type of message, and are processed on a first-in, first-out basis. The time at which the message is added into the queue becomes the official timestamp in which the ITL acknowledges receipt of the message. Should the ITL database be unavailable for an extended period of time due to hardware failure, messages remain in the queue until they can be processed. Viability checks determine whether the message from the queue should be processed. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.4: Message Viability Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1301</td>
<td>Message Age</td>
<td>Message must be processed within 24 hours of submission.</td>
<td>1</td>
</tr>
</tbody>
</table>

5.4.4 Registry Checks

After the message has been retrieved from the message queue and the location of the message file has been written to the message log, the ITL performs checks to determine if the registries involved in the transaction are identifiable and eligible to participate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.
### Figure 5.5: Registry Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1501</td>
<td>Identify Registry</td>
<td>Initiating Registry must be listed in Registry table.</td>
<td>1</td>
</tr>
<tr>
<td>1503</td>
<td>Initiating Registry Available for Transactions</td>
<td>Initiating Registry status must allow transactions to be proposed.</td>
<td>1</td>
</tr>
<tr>
<td>1504</td>
<td>Acquiring Registry Available for Transactions</td>
<td>Acquiring Registry status must allow transactions to be accepted.</td>
<td>1</td>
</tr>
</tbody>
</table>

### 5.4.5 Data Integrity Checks for Transactions

This category of checks is performed by the Data_Integrity_Checks function to identify incoming messages containing data that fail basic data integrity checks. If any data in a message fail these checks, the message is returned to the sender with an appropriate response code. The message is not logged in the Transaction_Log table and is not processed further. Additionally, all data integrity checks are critical checks in that if they result in failure, no further checks should be processed. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

### Figure 5.6: Summary of Data Integrity Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>2001</td>
<td>Transaction Mask</td>
<td>Transaction ID must be comprised of a valid registry code followed by numeric values.</td>
<td>1</td>
</tr>
<tr>
<td>2002</td>
<td>Transaction Type Code</td>
<td>Transaction Type Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2003</td>
<td>Supplementary Transaction Type Code</td>
<td>Supplementary Transaction Type Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2004</td>
<td>Transaction Status Code</td>
<td>Transaction Status Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2005</td>
<td>Transaction Status DateTime</td>
<td>Transaction Status DateTime must be before the current DateTime and no older than two weeks.</td>
<td>1</td>
</tr>
<tr>
<td>2006</td>
<td>Account Type Code</td>
<td>Account Type Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2007</td>
<td>Initiating Account Identifier</td>
<td>Initiating Account Identifier must be greater than zero.</td>
<td>1</td>
</tr>
<tr>
<td>2008</td>
<td>Acquiring Account Identifier</td>
<td>Acquiring Account Identifier must be greater than zero.</td>
<td>1</td>
</tr>
</tbody>
</table>

(cont.)
Figure 5.6: Summary of Data Integrity Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>Notification Type Code</td>
<td>Notification Type Code must be valid.</td>
<td>2</td>
</tr>
<tr>
<td>2010</td>
<td>Originating Registry</td>
<td>The Originating Registry of all unit blocks must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2011</td>
<td>Unit Type Code</td>
<td>Unit Type Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2012</td>
<td>Supplementary Unit Type Code</td>
<td>Supplementary Unit Type Code must be valid.</td>
<td>1</td>
</tr>
<tr>
<td>2013</td>
<td>Unit Serial Block</td>
<td>Unit Serial block start and Unit Serial block end must be present.</td>
<td>1</td>
</tr>
<tr>
<td>2014</td>
<td>Unit Serial Range</td>
<td>Unit Serial block end must be greater than or equal to the Unit Serial block start.</td>
<td>1</td>
</tr>
<tr>
<td>2015</td>
<td>LULUCF Activity Code</td>
<td>RMUs, ERUs converted from RMUs, tCERs and ICERs must have a valid LULUCF activity code.</td>
<td>1</td>
</tr>
<tr>
<td>2016</td>
<td>No LULUCF Activity Code</td>
<td>AAUs, ERUs converted from AAUs and CERs must not have a LULUCF activity code.</td>
<td>1</td>
</tr>
<tr>
<td>2017</td>
<td>Project ID</td>
<td>ERUs, CERs, tCERs, and ICERs must have a valid Project ID.</td>
<td>1</td>
</tr>
<tr>
<td>2018</td>
<td>No Project ID</td>
<td>AAUs or RMUs must not have a Project ID.</td>
<td>1</td>
</tr>
<tr>
<td>2019</td>
<td>ERU Track Code</td>
<td>ERUs must have a valid track code.</td>
<td>1</td>
</tr>
<tr>
<td>2020</td>
<td>No Track Code</td>
<td>AAUs, RMUs, CERs, tCERs and ICERs must not have a track code.</td>
<td>1</td>
</tr>
<tr>
<td>2021</td>
<td>Expiry Date</td>
<td>tCERs and ICERs must have an Expiry Date.</td>
<td>2</td>
</tr>
<tr>
<td>2022</td>
<td>No Expiry Date</td>
<td>AAUs, RMUs, ERUs and CERs must not have an Expiry Date.</td>
<td>2</td>
</tr>
</tbody>
</table>

5.4.6 Message Sequence Checks for Transactions from Registries

After the data in the message have been checked, the ITL performs checks to ensure that the message received has been submitted in the proper sequence, including whether process status is consistent and appropriate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.
### Figure 5.7: Sequence Checks for Transactions from Registries

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3001</td>
<td>Transaction ID Not Unique</td>
<td>Transaction ID for proposed transactions must not already exist in the ITL.</td>
<td>1</td>
</tr>
<tr>
<td>3002</td>
<td>Prior Record of Transaction ID from Registry</td>
<td>Transaction ID for ongoing transactions must already exist in the ITL.</td>
<td>1</td>
</tr>
<tr>
<td>3003</td>
<td>Transaction Status Out of Sequence for Prior Completed Status</td>
<td>Previous completed transactions cannot be completed again.</td>
<td>1</td>
</tr>
<tr>
<td>3004</td>
<td>Transaction Status Out of Sequence for Prior Rejected Status</td>
<td>Previously rejected transactions cannot be completed.</td>
<td>1</td>
</tr>
<tr>
<td>3005</td>
<td>Transaction Status Out of Sequence for Prior ITL Discrepancy Status</td>
<td>Transactions for which an ITL discrepancy has been previously identified cannot be completed.</td>
<td>1</td>
</tr>
<tr>
<td>3006</td>
<td>Transaction Status Out of Sequence for Prior STL Discrepancy Status</td>
<td>Transactions for which an STL discrepancy has been previously identified cannot be completed.</td>
<td>1</td>
</tr>
<tr>
<td>3007</td>
<td>Transaction Status Out of Sequence for Prior Terminated Status</td>
<td>Previously terminated transactions cannot be completed.</td>
<td>1</td>
</tr>
<tr>
<td>3008</td>
<td>Transaction Status Out of Sequence for Prior Cancelled Status</td>
<td>Previously cancelled transactions cannot be completed.</td>
<td>1</td>
</tr>
<tr>
<td>3009</td>
<td>Transaction Status Out of Sequence for Prior Accepted Status</td>
<td>Previously accepted external transactions cannot be terminated.</td>
<td>1</td>
</tr>
<tr>
<td>3010</td>
<td>Transaction Status Out of Sequence for Accepted or Rejected Status</td>
<td>Transaction status of Accepted or Rejected is not valid for non-external transactions.</td>
<td>1</td>
</tr>
<tr>
<td>3011</td>
<td>Transaction Status Not Compatible with Initiating Registry</td>
<td>Transaction status from Initiating Registry must indicate status of Proposed, Completed, or Terminated.</td>
<td>1</td>
</tr>
</tbody>
</table>

(cont.)
Figure 5.7: Sequence Checks for Transactions from Registries (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3012</td>
<td>Transaction Status Not Compatible with Acquiring Registry</td>
<td>Transaction status from Acquiring Registry must indicate status of Rejected or Accepted.</td>
<td>1</td>
</tr>
</tbody>
</table>

5.4.7 Message Sequence Checks for Transactions from STLs

If a message has been received from an STL, the ITL checks to determine if the order of messages and statuses is consistent and appropriate. See Figure 5.15, Queue Processing Checks, for further details on queue processing.

Figure 5.8: Sequence Checks for STL Messages

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3501</td>
<td>Transaction Status Not Compatible with STL</td>
<td>Transaction status from STL must indicate status of Discrepancy or No Discrepancy.</td>
<td>2</td>
</tr>
<tr>
<td>3502</td>
<td>Prior Record of Transaction ID from STL</td>
<td>Transaction ID for ongoing transactions must exist in ITL.</td>
<td>2</td>
</tr>
</tbody>
</table>

5.4.8 General Transaction Checks

The ITL performs this category of checks for all transaction messages involving existing unit blocks. Issuance transactions do not undergo General Transaction checks. See Figure 5.17, Validate Proposal, for further details on how transactions are checked.

Figure 5.9: General Transaction Checks (Not Applied to Issuance Transactions)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>4001</td>
<td>Applicable Commitment Period</td>
<td>Applicable Commitment Period must correspond to the current or next Commitment Period (including their true-up periods).</td>
<td>1</td>
</tr>
<tr>
<td>4002</td>
<td>Prior Record of Units</td>
<td>Units identified in the transaction must already exist in the ITL.</td>
<td>1</td>
</tr>
<tr>
<td>4003</td>
<td>Registry Holds Units</td>
<td>Units identified in the transaction must be held by Initiating Registry.</td>
<td>1</td>
</tr>
</tbody>
</table>

(cont.)
### Figure 5.9: General Transaction Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>4004</td>
<td>Unit Block Attributes</td>
<td>All attributes of all unit blocks must be consistent with ITL unit block attributes except where attributes are changed by the current transaction.</td>
<td>1</td>
</tr>
<tr>
<td>4005</td>
<td>Single Applicable Commitment Period</td>
<td>All unit blocks in transaction must be for a single Applicable Commitment Period.</td>
<td>1</td>
</tr>
<tr>
<td>4006</td>
<td>Acquiring and Transferring Registry Consistency</td>
<td>For all transactions except for external transfers, the Initiating and Acquiring Registries must be the same.</td>
<td>1</td>
</tr>
<tr>
<td>4007</td>
<td>Acquiring and Transferring Registries for External Transactions</td>
<td>For external transfers, the Initiating and Acquiring Registries must be different.</td>
<td>1</td>
</tr>
<tr>
<td>4008</td>
<td>Units Have ITL Inconsistencies</td>
<td>Units identified in the transaction must not have inconsistencies identified through reconciliation with the ITL.</td>
<td>1</td>
</tr>
<tr>
<td>4009</td>
<td>Units Have STL Inconsistencies</td>
<td>Units identified in the transaction must not have inconsistencies identified through reconciliation with an STL.</td>
<td>1</td>
</tr>
<tr>
<td>4010</td>
<td>Units are Unavailable</td>
<td>Units identified in the transaction must not be involved in another transaction.</td>
<td>1</td>
</tr>
<tr>
<td>4011</td>
<td>Units are Cancelled</td>
<td>Cancelled units must not be subject to further transactions.</td>
<td>1</td>
</tr>
<tr>
<td>4012</td>
<td>Units are Retired</td>
<td>Retired units must not be subject to further transactions.</td>
<td>1</td>
</tr>
<tr>
<td>4013</td>
<td>Units are Expired</td>
<td>Expired tCERs and ICERs must not be subject to further transactions, except internal transfers to a Type 5 cancellation account.</td>
<td>2</td>
</tr>
<tr>
<td>4014</td>
<td>Units Previously Used in Replacement</td>
<td>Units previously used to replace tCERs or ICERs must not be subject to further transactions.</td>
<td>2</td>
</tr>
<tr>
<td>4015</td>
<td>ICER Transaction Ineligibility</td>
<td>ICERs must not be transferred to a holding or retirement account where the CDM Executive Board has notified a replacement requirement for the associated Project.</td>
<td>2</td>
</tr>
</tbody>
</table>

### 5.4.9 Transaction-specific Checks

The ITL performs this category of checks on all Kyoto transactions for the specified transaction types. See Figure 5.17, Validate Proposal, for further details on how transactions are checked.
Figure 5.10: Transaction-specific Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5001</td>
<td>National Registry Issuance</td>
<td>Issuance</td>
<td>AAUs and RMUs must be issued by a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5002</td>
<td>No ERU Issuance</td>
<td>Issuance</td>
<td>ERUs cannot be issued.</td>
<td>2</td>
</tr>
<tr>
<td>5003</td>
<td>RMU Issuance Timing</td>
<td>Issuance</td>
<td>RMUs cannot be issued before the end of the Commitment Period, unless annual Issuance option is selected.</td>
<td>2</td>
</tr>
<tr>
<td>5003</td>
<td>CDM Registry Issuance</td>
<td>Issuance</td>
<td>CERs, tCERs and ICERs must be issued by the CDM Registry.</td>
<td>2</td>
</tr>
<tr>
<td>5004</td>
<td>Single Issuance Unit Type</td>
<td>Issuance</td>
<td>A transaction must not issue more than one Unit Type.</td>
<td>2</td>
</tr>
<tr>
<td>5005</td>
<td>Single Issuance Commitment Period</td>
<td>Issuance</td>
<td>The Original Commitment Period must be the same for all units issued by the transaction.</td>
<td>2</td>
</tr>
<tr>
<td>5006</td>
<td>Consistent Applicable Commitment Period</td>
<td>Issuance</td>
<td>The Applicable Commitment Period must be the same as the Original Commitment Period for all units issued by the transaction.</td>
<td>2</td>
</tr>
<tr>
<td>5007</td>
<td>Issued Serial Numbers</td>
<td>Issuance</td>
<td>Serial numbers for proposed issuance must not already exist in the ITL.</td>
<td>2</td>
</tr>
<tr>
<td>5008</td>
<td>AAU Issuance Quantity</td>
<td>Issuance</td>
<td>The quantity of AAUs issued must not exceed allowed quantity for the Commitment Period.</td>
<td>2</td>
</tr>
<tr>
<td>5009</td>
<td>RMU Issuance Quantity</td>
<td>Issuance</td>
<td>The quantity of RMUs issued must not exceed allowed quantity for each LULUCF Activity Type and Commitment Period.</td>
<td>2</td>
</tr>
<tr>
<td>5010</td>
<td>CDM Issuance Unit Type</td>
<td>Issuance</td>
<td>The type of units to be issued for each CDM Project must be consistent with the Project activity.</td>
<td>2</td>
</tr>
<tr>
<td>5011</td>
<td>Consistency of Unit Type Issued for a LULUCF CDM Project</td>
<td>Issuance</td>
<td>Choice of unit type must be consistent with previous issuance of tCERs and ICERs for the Project.</td>
<td>2</td>
</tr>
<tr>
<td>5012</td>
<td>CDM Issuance Quantity</td>
<td>Issuance</td>
<td>CER, tCER or ICER issuance for each CDM Project must not exceed quantity specified by CDM Executive Board.</td>
<td>2</td>
</tr>
</tbody>
</table>

(cont.)
### Figure 5.10: Transaction-specific Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5013</td>
<td>CDM LULUCF Activity Code</td>
<td>Issuance</td>
<td>The LULUCF Activity Code of CERs, tCERs or ICERs proposed for issuance must be consistent with the project activity.</td>
<td>2</td>
</tr>
<tr>
<td>5014</td>
<td>CDM Project ID</td>
<td>Issuance</td>
<td>A valid CDM Project ID must be present for the issuance of all CERs, tCERs and ICERs.</td>
<td>2</td>
</tr>
<tr>
<td>5015</td>
<td>tCER Expiry Date</td>
<td>Issuance</td>
<td>Expiry Date for tCERs must be consistent with the end date of the Commitment Period subsequent to the Original Commitment Period of the tCER.</td>
<td>2</td>
</tr>
<tr>
<td>5016</td>
<td>lCER Expiry Date</td>
<td>Issuance</td>
<td>Expiry date for ICERs must be consistent with the End Date of the Crediting Period for the Project specified by the CDM Executive Board.</td>
<td>2</td>
</tr>
<tr>
<td>5051</td>
<td>National Registry Conversion</td>
<td>Conversion</td>
<td>The Initiating Registry converting AAUs or RMUs must be a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5052</td>
<td>Holding Account Conversion</td>
<td>Conversion</td>
<td>The Initiating Account for a conversion transaction must be a holding account.</td>
<td>2</td>
</tr>
<tr>
<td>5053</td>
<td>Conversion Eligibility (Track 1)</td>
<td>Conversion</td>
<td>If the unit is a Track 1 ERU, the Party of the Initiating Registry must be determined to meet eligibility criteria 1 through 6.</td>
<td>2</td>
</tr>
<tr>
<td>5054</td>
<td>Conversion Eligibility (Track 2)</td>
<td>Conversion</td>
<td>If the unit is a Track 2 ERU, the Party of the Initiating Registry must be determined to meet eligibility criteria 1, 2 and 4.</td>
<td>2</td>
</tr>
<tr>
<td>5056</td>
<td>Conversion Unit Type</td>
<td>Conversion</td>
<td>Units for conversion must be AAUs or RMUs.</td>
<td>2</td>
</tr>
<tr>
<td>5057</td>
<td>Single Conversion Unit Type</td>
<td>Conversion</td>
<td>A transaction must not convert more than one unit type.</td>
<td>2</td>
</tr>
<tr>
<td>5058</td>
<td>Conversion by Issuing Registry</td>
<td>Conversion</td>
<td>Units for conversion must have been issued by Initiating Registry.</td>
<td>2</td>
</tr>
<tr>
<td>5059</td>
<td>Project ID</td>
<td>Conversion</td>
<td>A valid JI Project ID must be present for the conversion of all ERUs.</td>
<td>2</td>
</tr>
<tr>
<td>5060</td>
<td>JI Conversion Unit Type</td>
<td>Conversion</td>
<td>The type of units to be converted to ERUs for each JI Project must be consistent with Project activity.</td>
<td>2</td>
</tr>
<tr>
<td>5061</td>
<td>Track 2 ERU Conversion Quantity</td>
<td>Conversion</td>
<td>Track 2 ERU Conversion for each Track 2 JI Project must not exceed the quantity specified by the Article 6 Supervisory Committee.</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 5.10: Transaction-specific Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5101</td>
<td>General Transferring Registry Eligibility for External Transfers</td>
<td>External</td>
<td>The Party of an initiating national registry must be determined to meet eligibility criteria 1 through 6, except for the first external transfer of a track 2 ERU which the Registry has converted.</td>
<td>2</td>
</tr>
<tr>
<td>5102</td>
<td>ERU Track 2 Transferring Registry Eligibility for External Transfers</td>
<td>External</td>
<td>If the transaction is the first external transfer of a track 2 ERU which the Registry has converted, the Party of the initiating national registry must be determined to meet eligibility criteria 1, 2 and 4.</td>
<td>2</td>
</tr>
<tr>
<td>5103</td>
<td>Acquiring Registry Eligibility for External Transfers</td>
<td>External</td>
<td>The Party of an acquiring national registry must be determined to meet eligibility criteria 1 through 6, except for transfers initiated by the CDM Registry.</td>
<td>2</td>
</tr>
<tr>
<td>5104</td>
<td>Commitment Period Reserve</td>
<td>External</td>
<td>The total quantity of all units held in a national registry, which may be used for compliance for the applicable Commitment Period of a transaction, must not fall below the CPR level for the Party for that Commitment Period, except where the transaction is a first transfer of Track 2 ERUs converted by the registry. This total quantity is the total of all units in holding and retirement accounts, less first external transfers of track 2 ERUs converted by the registry, expired units remaining in holding and retirement accounts, and required cancellations and replacements which have not been carried out 30 days after the relevant notification was sent by the ITL.</td>
<td>2</td>
</tr>
<tr>
<td>5105</td>
<td>External Transfers to CDM Registry</td>
<td>External</td>
<td>CDM Registry can only receive external transfers to Cancellation accounts for compensating excess issuance of CERs, tCERs and ICERs.</td>
<td>2</td>
</tr>
<tr>
<td>5106</td>
<td>Suspension from Making External Transfers</td>
<td>External</td>
<td>The Party of an initiating national registry must not have been suspended from making external transfers as a result of not meeting its emission target for the previous Commitment Period.</td>
<td>2</td>
</tr>
</tbody>
</table>

(cont.)
<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5151</td>
<td>National Registry Cancellation</td>
<td>Cancellation</td>
<td>Cancellation to Net Source, Non-compliance and Voluntary Cancellation Accounts must take place in a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5152</td>
<td>No Excess Issuance Cancellation</td>
<td>Cancellation</td>
<td>Cancellation to Excess Issuance Cancellation Account must not take place in a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5153</td>
<td>Cancellation Accounts</td>
<td>Cancellation</td>
<td>The Acquiring Account for a cancellation transaction must be a cancellation account.</td>
<td>2</td>
</tr>
<tr>
<td>5154</td>
<td>Cancellation Account Identifier</td>
<td>Cancellation</td>
<td>Account identifiers must be provided for acquiring accounts in cancellation transactions.</td>
<td>2</td>
</tr>
<tr>
<td>5155</td>
<td>Cancellation Account Commitment Period</td>
<td>Cancellation</td>
<td>The unit blocks retired must have the same Applicable Commitment Period as the Cancellation Account.</td>
<td>2</td>
</tr>
<tr>
<td>5156</td>
<td>tCER and lCER Cancellation to Net Source and Non-compliance Cancellation Accounts</td>
<td>Cancellation</td>
<td>tCERs and lCERs cannot be transferred to Net Source Cancellation Accounts or Non-compliance Cancellation Accounts.</td>
<td>2</td>
</tr>
<tr>
<td>5157</td>
<td>Notification ID for tCER and ICER Cancellations to Excess Issuance Cancellation Accounts</td>
<td>Cancellation</td>
<td>tCERs and ICERs may only be transferred to Excess Issuance Cancellation Account in the CDM registry in the case that excess tCER and ICER issuance is being compensated pursuant to a Excess Issuance Notification.</td>
<td>2</td>
</tr>
<tr>
<td>5158</td>
<td>Notification ID for Net Source Cancellations</td>
<td>Cancellation</td>
<td>Units may only be transferred to a net source cancellation account if a notification has been received from the ITL and this ID is reported in the transaction.</td>
<td>2</td>
</tr>
<tr>
<td>5159</td>
<td>Notification ID for Non-compliance Cancellations</td>
<td>Cancellation</td>
<td>Units may only be transferred to a Non-compliance cancellation account if a notification has been received from the ITL and this ID is reported in the transaction.</td>
<td>2</td>
</tr>
<tr>
<td>5201</td>
<td>National Registry Replacement</td>
<td>Replacement</td>
<td>The Initiating Registry replacing units must be a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5202</td>
<td>tCER Replacement Accounts</td>
<td>Replacement</td>
<td>The Acquiring Account for a replacement transaction involving tCERs must be a tCER replacement account.</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 5.10: Transaction-specific Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5203</td>
<td>lCER Replacement Accounts</td>
<td>Replacement</td>
<td>The Acquiring Account for a replacement transaction involving ICERs must be an ICER replacement account.</td>
<td>2</td>
</tr>
<tr>
<td>5204</td>
<td>Replacement Account Identifier</td>
<td>Replacement</td>
<td>Account identifiers must be provided for acquiring accounts in replacement transactions.</td>
<td>2</td>
</tr>
<tr>
<td>5205</td>
<td>Replacement Account Commitment Period</td>
<td>Replacement</td>
<td>The Unit Blocks used for replacement must have the same Applicable Commitment Period as the Replacement Account.</td>
<td>2</td>
</tr>
<tr>
<td>5206</td>
<td>Unit Type to be Replaced</td>
<td>Replacement</td>
<td>Units to be replaced must be tCERs or ICERs.</td>
<td>2</td>
</tr>
<tr>
<td>5207</td>
<td>Multiple Replacement</td>
<td>Replacement</td>
<td>A unit may be replaced only once.</td>
<td>2</td>
</tr>
<tr>
<td>5208</td>
<td>Single Replacement Registry</td>
<td>Replacement</td>
<td>The registry holding the units to be replaced and the replacing units must be the same.</td>
<td>2</td>
</tr>
<tr>
<td>5209</td>
<td>Quantity of Replacement Units</td>
<td>Replacement</td>
<td>The quantity of units replaced must equal the quantity of replacing units.</td>
<td>2</td>
</tr>
<tr>
<td>5210</td>
<td>One-To-Many Replacement Units</td>
<td>Replacement</td>
<td>A transaction cannot contain many-to-many relationships between replaced and replacing blocks.</td>
<td>2</td>
</tr>
<tr>
<td>5211</td>
<td>Location of Replaced tCERs</td>
<td>Replacement</td>
<td>tCERs to be replaced must be held in a Retirement account or a tCER Replacement account.</td>
<td>2</td>
</tr>
<tr>
<td>5212</td>
<td>Location of Replaced ICERs</td>
<td>Replacement</td>
<td>ICERs to be replaced must not be held in Cancellation accounts.</td>
<td>2</td>
</tr>
<tr>
<td>5213</td>
<td>lCER Replacement Units (upon Expiry)</td>
<td>Replacement</td>
<td>ICER Replacement accounts (upon expiry) cannot acquire tCERs or ICERs.</td>
<td>2</td>
</tr>
<tr>
<td>5214</td>
<td>tCER Replacement Units (upon Expiry)</td>
<td>Replacement</td>
<td>tCER replacement accounts (for unit expiry) cannot acquire ICERs.</td>
<td>2</td>
</tr>
</tbody>
</table>
**Figure 5.10: Transaction-specific Checks (cont.)**

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5215</td>
<td>ICER Replacement Units (upon Reversal of Storage or Lack of Certification Report)</td>
<td>Replacement</td>
<td>ICER Replacement accounts (for Reversal in Storage) may not acquire tCERs and may not acquire ICERs with a Project Identifier other than that specified in the replacement notification.</td>
<td>2</td>
</tr>
<tr>
<td>5216</td>
<td>Replacement Notification upon tCER Expiry</td>
<td>Replacement</td>
<td>If provided, the Replacement Notification ID must be valid and must be for replacement upon tCER expiry.</td>
<td>2</td>
</tr>
<tr>
<td>5217</td>
<td>Replacement Notification upon ICER Expiry</td>
<td>Replacement</td>
<td>If provided, the Replacement Notification ID must be valid and must be for replacement upon ICER expiry.</td>
<td>2</td>
</tr>
<tr>
<td>5218</td>
<td>Replacement Notification for Reversal in Storage</td>
<td>Replacement</td>
<td>A valid Replacement Notification ID must be provided for replacement upon reversal in storage.</td>
<td>2</td>
</tr>
<tr>
<td>5219</td>
<td>Replacement Notification for Lack of Certification Report</td>
<td>Replacement</td>
<td>A valid Replacement Notification ID must be provided for replacement upon a lack of Certification Report.</td>
<td>2</td>
</tr>
<tr>
<td>5220</td>
<td>Project ID for ICERs Replacement (upon Reversal of Storage or lack of Certification Report)</td>
<td>Replacement</td>
<td>For ICER replacement transactions upon Reversal of Storage or lack of a Certification Report, the Project ID for the ICERs to be replaced must be consistent with the Project ID contained in the replacement notification.</td>
<td>2</td>
</tr>
<tr>
<td>5251</td>
<td>National Registry Retirement</td>
<td>Retirement</td>
<td>The Initiating Registry retiring units must be a national registry.</td>
<td>2</td>
</tr>
<tr>
<td>5252</td>
<td>Retirement Account</td>
<td>Retirement</td>
<td>The Acquiring Account for a retirement transaction must be a retirement account.</td>
<td>2</td>
</tr>
<tr>
<td>5253</td>
<td>Retirement Account Identifier</td>
<td>Retirement</td>
<td>Account identifiers must be provided for acquiring accounts in retirement transactions.</td>
<td>2</td>
</tr>
</tbody>
</table>

(continues)
**Figure 5.10: Transaction-specific Checks (cont.)**

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5254</td>
<td>Retirement Account Commitment Period</td>
<td>Retirement</td>
<td>The Unit Blocks retired must have the same Applicable Commitment Period as the Retirement Account.</td>
<td>2</td>
</tr>
<tr>
<td>5255</td>
<td>CER, tCER and lCER Retirement Eligibility</td>
<td>Retirement</td>
<td>The Party of the Initiating Registry must be determined to meet eligibility criteria 1 through 6.</td>
<td>2</td>
</tr>
<tr>
<td>5256</td>
<td>tCER and lCER Retirement Limit</td>
<td>Retirement</td>
<td>tCER and ICER retirement must not exceed allowed quantity.</td>
<td>2</td>
</tr>
<tr>
<td>5301</td>
<td>National Registry Carry-over</td>
<td>Carry-over</td>
<td>The Initiating Registry carrying over units must be a national registry.</td>
<td>3</td>
</tr>
<tr>
<td>5302</td>
<td>Holding Account Carry-over</td>
<td>Carry-over</td>
<td>The Initiating Account for a carry-over transaction must be a holding account.</td>
<td>3</td>
</tr>
<tr>
<td>5303</td>
<td>Subsequent Commitment Period</td>
<td>Carry-over</td>
<td>Units may be carried-over only to the next subsequent commitment period.</td>
<td>3</td>
</tr>
<tr>
<td>5304</td>
<td>Units Available for Carry-over</td>
<td>Carry-over</td>
<td>The quantity of units of each unit type carried-over must not exceed the limit of carry-over established by the Compliance Committee for the Party, and reported to the registry in the Unit Carry-over Notification.</td>
<td>3</td>
</tr>
<tr>
<td>5305</td>
<td>RMU Carry-over</td>
<td>Carry-over</td>
<td>RMUs may not be carried over.</td>
<td>3</td>
</tr>
<tr>
<td>5306</td>
<td>ERU (from RMUs) Carry-over</td>
<td>Carry-over</td>
<td>ERUs converted from RMUs may not be carried over.</td>
<td>3</td>
</tr>
<tr>
<td>5307</td>
<td>lCER or tCER Carry-over</td>
<td>Carry-over</td>
<td>tCERS or ICERS may not be carried over.</td>
<td>3</td>
</tr>
<tr>
<td>5310</td>
<td>Notification ID for Carry-over</td>
<td>Carry-over</td>
<td>Units may only be carried over if a notification has been received from the ITL and this ID is reported in the transaction.</td>
<td>3</td>
</tr>
<tr>
<td>5311</td>
<td>Carry-over Unit Type</td>
<td>Carry-over</td>
<td>Unit blocks carried over must be consistent with the unit type specified in the unit carry-over notification.</td>
<td>3</td>
</tr>
</tbody>
</table>

(continues)
Figure 5.10: Transaction-specific Checks (cont.)

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Transaction Type</th>
<th>Check Description</th>
<th>Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>5450</td>
<td>Units for Expiry Date Change</td>
<td>Expiry Date Change</td>
<td>The units for Expiry Date Change must be tCERs or lCERs.</td>
<td>3</td>
</tr>
<tr>
<td>5451</td>
<td>New tCER Expiry Date</td>
<td>Expiry Date Change</td>
<td>The new tCER Expiry Date must be consistent with the End Date of the Commitment Period subsequent to the Original Commitment Period of the tCER.</td>
<td>3</td>
</tr>
<tr>
<td>5452</td>
<td>New lCER Expiry Date</td>
<td>Expiry Date Change</td>
<td>The new lCER Expiry Date is not consistent with the End Date of the renewed Crediting Period for the Project specified by the CDM Executive Board.</td>
<td>3</td>
</tr>
</tbody>
</table>

5.4.10 Registry Messages

These responses may be returned by an Acquiring Registry in response to a proposed external transaction, if the overall status of the transaction from the Acquiring Registry is "Rejected." If the transaction status code is "Accepted," no response code will be sent with the message from the Acquiring Registry.

Figure 5.11: Registry Messages

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Response Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5902</td>
<td>Acquiring account does not exist.</td>
</tr>
<tr>
<td>5903</td>
<td>Acquiring account is not eligible to receive units.</td>
</tr>
<tr>
<td>5904</td>
<td>Transaction inconsistent with Party policy.</td>
</tr>
<tr>
<td>5905</td>
<td>Transaction rejected by account holder.</td>
</tr>
<tr>
<td>5906</td>
<td>Account has been closed.</td>
</tr>
</tbody>
</table>

5.5 Running Totals and Related Limits on Transactions

The ITL maintains a series of running totals in order to quickly execute certain transaction checks without applying additional processing load to the database while relevant checks are being performed. These running totals are calculated by the ITL and must be adjusted automatically when the base data is updated, such that they are kept up-to-date and consistent with the base data. Calculating these running totals must be done efficiently, so that they are adjusted quickly and system resources are saved.

In addition, a series of limits are maintained in the ITL so that they may be used during the checking processes. These limits are based on reference data received from the compilation and accounting database or one of the other secretariat systems (see Annex A). Some of these limits
are cumulative (new reference data are added to relevant reference data received earlier) and are therefore maintained as running totals. However, other limits are not cumulative, such that newly received reference data is used to replace reference data received earlier.

For example, the Adjust Commitment Period Holdings running total (see below) shows the number of units available for compliance in each registry. The Reserve Limit (see below) is supplied by the compilation and accounting database and shows the Commitment Period Reserve of each registry, which is the level below which holdings in a registry must not fall. The ITL compares those two values when an External Transfer is proposed in order to check whether the transaction would violate the Commitment Period Reserve. Other running totals and limits are similarly used to check whether an Issuance transaction of CERs, tCERS or ICERs would exceed the allowable quantity for a CDM Project or whether a Conversion transaction would exceed the allowable quantity of conversion for a JI project. Other running totals and limits are used to check whether required transactions contained in a notification have been fulfilled.

These running total and limits are to be maintained in the Registry Unit Sum table. Running totals and other limits are held for each registry.

See Annex F for running totals and limits which are specified in the process of performing checks. These include the running totals and limits described in the following sections.

5.5.1 Unit Type Issuance Limits

These values are the numbers of AAUs and RMUs allowed to be issued in a registry for each Commitment Period. They are received as reference data from the compilation and accounting database for each registry. The number of AAUs that may be issued is not cumulative and will be received only once per commitment period (this is therefore not a running total). The number of RMUs that may be issued is cumulative and may be received as often as once year for each registry. This number is therefore a running total in that newly-received reference data are added to reference data received earlier. See checks 5008 and 5009 in Annex F.

5.5.2 Unit Type Issuance Levels

These running totals are the number of AAUs and RMUs issued in a registry for each Commitment Period. These values are updated with each relevant Issuance transaction. See checks 5008 and 5009 in Annex F.

5.5.3 Project Issuance Level

This running total is the number of CERs, tCERS or ICERs that have been issued for a specific CDM project. This value is updated with each Issuance of CERs, tCERS or ICERs performed by the CDM Registry. This value is compared with the total allowable issuance for the project, as calculated from Project Action Log. See check 5012 in Annex F.

5.5.4 Converted Level

This running total is the number of AAUs and RMUs in a registry that have been converted to Track 2 ERUs for a specific JI project. This value is updated with each relevant Conversion transaction. This value is compared with the total allowable conversion to Track 2 ERUs for the project, as calculated from Project Action Log. See check 5061 in Annex F.
5.5.5 Reserve Limit (not a running total)

This value is the Commitment Period Reserve and is received as reference data from the compilation and accounting database for each registry. These reference data are not cumulative and will be received a maximum of once per year. A newly-received Reserve Limit must replace reference data received earlier. The Reserve Limit is a minimum limit, such that registries must hold more than this number of units (refer sections 5.5.6 and 5.5.8). See check 5104 in Annex F.

5.5.6 Commitment Period Holdings

This running total is the number of units held in a registry which are potentially available for compliance for a Commitment Period. It is equal to total holdings in holding accounts and the retirement account (for that Commitment Period) of a registry, less units in those accounts which have expired. It is updated with each transaction which affects the total holdings in these accounts and each expiry of a unit. This running total is further modified to form the Adjusted Commitment Period Holdings (refer section 5.5.8). See check 5104 in Annex F.

5.5.7 Track 2 ERU First-time Transfer Level

This running total is the number of External Transfers out of a registry of Track 2 ERUs which were originally converted by the registry. Only the first External Transfers of each such ERU should be counted (if the registry subsequently acquires and then transfers such ERUs again, these transfers should not be counted in the running total). This value is updated with each External Transfer involving such an ERU that has not previously been involved in an External Transfer from the registry. This running total is used in one of the modifications to form the Adjusted Commitment Period Holdings running total (refer section 5.5.8). See check 5104 in Annex F.

5.5.8 Adjusted Commitment Period Holdings

This running total is the Commitment Period Holdings running total after it has been adjusted:

(a) Upwards by the Track 2 ERU First-time Transfer Level running total, and

(b) Downwards by the Number of Units which should have been cancelled or replaced in response to notifications sent by the ITL (refer section 7.3) to the registry but which had not been replaced or cancelled within the 30 day time period (unfulfilled notifications).

This value is updated with:

(a) Each update in the Commitment Period Holdings running total,

(b) Each update in the Track 2 ERU First-time Transfer Level running total, and

(c) Each replacement or cancellation notification which remains unfulfilled after its 30 day time period has passed.

See check 5104 in Annex F.
5.5.9 Retirement Limit (not a running total)

This value is the number of tCERs and ICERs in a registry which are allowed to be retired for a Commitment Period. It is received as reference data from the compilation and accounting database for each registry. This reference data is not cumulative, and will be received only once per Commitment Period for each national registry. See check 5256 in Annex F.

5.5.10 Retirement Level

This running total is the number of tCERs and ICERs that have been retired for a Commitment Period by a registry. This value is updated with each relevant Retirement transaction. See check 5256 in Annex F.

5.5.11 Carry-over Limits (not running totals)

These values are the number of units of each unit type in a registry, which are valid for a particular Commitment Period, which are allowed to be carried over to the next Commitment Period. These values are received as reference data from the compilation and accounting database for each registry. These reference data are not cumulative, and will be received only once per Commitment Period for each national registry. See check 5304 in Annex F.

5.5.12 Carry-over Levels

These running totals are the number of units for a Commitment Period, by each unit type, which have been carried over in a registry. These values are updated with each relevant Carry-over transaction. See check 5304 in annex F.

5.6 Activity Diagrams

The following activity diagrams provide an overview of how the HTTP SOAP request is processed for a transaction. The first diagram describes transactions between a single registry and the ITL. This diagram is applicable to the Issuance, Conversion, Cancellation, Retirement, Carry-over, Replacement, and Expiry Date Change. The second diagram describes an External transaction between two registries and the ITL. These activity diagrams do not show interaction between the ITL and an STL.
5.6.1 Basic Transaction Activity Diagram

Figure 5.12: Simple (Non-STL) Transaction Process

```
[Diagram showing the process flow]
```

Annex 13: ITL Software Specification, Version <1.0>, Draft #8
### 5.6.2 External Transfer Activity Diagram

**Figure 5.13: External (Non-STL) Transaction Process**

- **Initiating Registry**
  - Generate HTTP SOAP Request
  - Wait for response back from Web service
- **Communications Hub**
  - Authenticate Registry (VPN)
  - Check for Well Formness (WSDL)
  - Preliminary Processing (Fig. 5.14)
  - HTTP SOAP Response
  - If failure then send notification with response code errors
- **ITL**
  - Queue Processing Checks (Fig. 5.15)
  - Evaluate Transaction (Fig. 5.16)
  - Generate HTTP SOAP Request
- **Acquiring Registry**
  - Validate Proposal (Fig. 5.17)
  - Determine Route for Proposal (Fig. 5.18)
  - Call AcceptProposal
  - Registry does internal processing of transaction
- **Web Service Terminal**
  - Wait for response back from Web service

- **AcceptNotification**:
  - Send acknowledgement response
  - Send notification with response codes only if failure occurs
  - Send notification with response codes indicating accepted proposal

(cont.)
Figure 5.13  External (Non-STL) Transaction Process (cont.)

- **Initiating Registry**
  - Generate HTTP SOAP Request
  - Wait for response back from web service

- **Communications Hub**
  - Authenticate Registry (VPN)
  - Check for Well Formness (WSDL)
  - Preliminary Processing (Fig. 5.14)

- **ITL**
  - Queue Processing Checks (Fig. 5.15)
  - Evaluate Transaction (Fig. 5.16)
  - Finalise Transaction (Fig. 5.21)

- **Call Accept Notification. Send notification that proposal is completed.**

- **Web service terminates**

- **Send response back that notification was received and logged or return SOAP faults**

- **If no SOAP faults or preliminary check failures then**
5.7 Transaction Flow Diagrams

For all types of transactions, the processing of a transaction proceeds through a standard sequence of steps. The following process flow diagrams supplement the activity diagrams in Section 5.5. Boxes or diamonds that contain a ":" prefix indicate a function which can be found in detail in Annex E. Boxes with the "ref" symbol in the corner indicate that supplemental sequence diagrams are provided.

**Figure 5.14: Preliminary Processing**

This process performs the Preliminary Checks independent of any transactions to the database, ensuring the basic structure of the message is correct and has been parsed to the file log and added to the queue. This process assumes that the message is well-formed and can be deserialized to an object graph.
Figure 5.15: Queue Processing Checks

**sd Queue Processing Checks**

This process performs the supplemental checks for messages being retrieved from the queue. These supplemental checks clear the message for further processing or return an error to the Initiating Registry identifying why the request could not be processed.

1. Get FIFO message from queue for processing.
   - Begin Database Transaction
   - Write_To_Message_Log
   - Message_Viability_Check
2. Check to determine if message is in the correct sequence for processing.
   - Message_Sequence_Check
   - Did any check fail?
     - Yes
       - Send notification to sender, terminate any further processing, and expire message from queue.
     - No
6. Determine where to route transaction based on transaction type and transaction status.
   - From Where did the message originate?
     - Any Registry
       - Commit Database Transaction
       - Send_Notification_To_Registry
     - STL
       - Commit Database Transaction
       - Send_To_STL
3. Check to see if registry has correct status and eligibility to participate in a transaction.
   - Check_Registry_Status
   - Are Initiating and Acquiring Registries valid and eligible?
     - Yes
       - Data_Integrity_Check
     - No
6. Application-level checks for data inputs supplied to the Web service.
   - Did any check fail?
     - Yes
       - Commit Database Transaction
       - Message remains in queue.
     - No
6. Call Evaluate_Response_Object to determine if queue processing checks failed.
   - Retrieve_Message_From_Queue
   - Commit Database Transaction
   - Message_Viability_Check
   - Was message more than 24 hours old?
     - Yes
       - Retrieve_Message_From_Queue
       - Data_Integrity_Check
       - Did any check fail?
         - Yes
           - Commit Database Transaction
           - Message remains in queue.
         - No
6. From Where did the message originate?
   - Any Registry
     - Evaluate Transaction
     - Ref: Evaluate Transaction
   - STL
     - Evaluate STL Notification
     - Ref: Evaluate STL Notification
9. Commit Database Transaction
10. Remove_Message_From_Queue
11. Terminate Queue Checks
Figure 5.16: Evaluate Transaction

This sequence of checks is only performed on messages passed from registries to the ITL (not from the STL).

**Inputs: Transaction Status and Transaction Type**

- What is the Transaction Status?
  - "Proposed"
  - "Accepted," "Rejected," "Completed" or "Terminated"

- What is the Transaction Type?
  - Other Types (1, 2, 4, 5, 6, 7, 8, 10)
  - External (3)

1. **Perform checks**
   - Validate Proposal

2. **Determine Route for Proposal**
3. **Process Non-external Notification**
4. **Process External Notification**

**Remove Message From Queue**
Validate Proposal will record the transaction in the log, apply the checks appropriate to the type of transaction and record the results in the ITL.

For each block:
- :Split_Blocks
  - Split blocks in unit table if necessary.
- :Delineate_Units
  - Set the Unit Status Code to 1 in the Unit Block table to flag units so that they may not be used in any forthcoming proposals.

For each block:
- :Write_Transaction
  - Add the transaction to the Transaction Log table.
- :Write_Transaction_Block
  - For each unit block, add a record to the Transaction Unit Block table.
  - Add a new status record indicating transaction is "Proposed."
- :Write_Transaction_Status
  - Evaluate transaction against the checks. If a transaction level check fails, record response code in Response_Log. If a block level check fails, record the response code in Transaction_Block_History.

Notification ID Provided?
- Yes: Append Record to Transaction_Notification
- No: :Get_Checks

For each check:
- :Execute_Checks
  - Transaction Level Check Fails
  - Block Level Check Fails
    - :Write_to_Response_Log
    - :Write_Block_History
    - Add a new status record indicating Transaction status as either "Checked (Discrepancy)," or "Checked (No Discrepancy)."
  - :Write_Transaction_Status
  - Commit Database Transaction

If transaction is in response to a Notification, record the Notification ID.
Figure 5.18: Determine Route for Proposal

This process determines the route for a proposal.

Inputs: Transaction Status, Transaction Type, Registry Parties

What is the Transaction Status?

- "Checked (Discrepancy)" (Status Code 3)
- "Checked (No Discrepancy)" (Status Code 2)

Is Initiating or Acquiring Registry participating in a supplementary program?

- Yes: Route Proposal to STL for further processing
- No: Call :Trade_Scheme_Member for Initiating Registry and the Acquiring Registry (if applicable) to determine if either party is in a supplementary program.

Is this an external transaction? (type 3)

- Yes: Send_Notification_To_Registry
- No: send_Proposal_To_Registry

Route Proposal to Acquiring Registry

Route Message to Initiating Registry

Send_Notification_To_Registry

Send_Proposal_To_Registry

Send_To_STL
This process handles non-external transactions (types 1, 2, 4, 5, 6, 7, 8, and 10) which have progressed beyond the "Proposed" stage.

**Inputs:** Transaction Status

- "Completed" (status code 4)
- "Terminated" (status code 5)

**What is the Transaction Status?**

- Finalise Transaction
- :Delineate Units

**For each block**

- Free the units involved in the transaction to be used in another transaction

**If the transaction had been validated previously, a notification may have been sent to an STL:**

- Was Prior Transaction Status "Checked (No Discrepancy)" (2)?
  - Yes
  - No

**Call :Trade_Scheme_Member**

**Is Initiating or Acquiring Registry participating in a supplementary program?**

- Yes
- No

**Route message to STL for further processing**

- :Send_To_STL
- Terminate
This process handles external transactions (type 3) which have progressed beyond the "Proposed" stage.

Inputs: Transaction Status

- :Write_Transaction_Status

What is the Transaction Status?

- "Accepted" (Status Code 8)
- "Rejected" (Status Code 6)
- "Terminated" (Status Code 5)
- "Completed" (Status Code 4)

For each block:

- Was Prior Transaction Status "Checked (No Discrepancy)" (2)?
  - Yes
    - Call :Trade_Scheme_Member
    - Finalise Transaction
  - No
    - If the transaction had been validated previously, a notification may have been sent to an STL.
    - Route message to STL for further processing

- Is Initiating Registry participating in a supplementary program?
  - Yes
    - Call :Trade_Scheme_Member
    - Route message to STL for further processing
  - No
    - Route Message to Initiating Registry
    - Send_Notification_To_Registry

- Is Acquiring Registry participating in a supplementary program?
  - Yes
    - Call :Trade_Scheme_Member
    - Route message to STL for further processing
  - No
    - Route Message to Initiating Registry
    - Send_Notification_To_Registry
    - Send_To_STL
    - Delineate_Units
This process updates the unit holdings for a registry and removes the constraint on the units, allowing them to engage in forthcoming transactions.

**What is the Transaction Type?**

- **Issuance (Type 1)**: Delineate Units
  - For each block in the transaction log, add the new blocks into the unit inventory table.
  - For each block, Delineate Units
  - For each block, Update Unit Block
  - Block_Ownership_ or_Account_Type
  - Update Unit Block

- **Conversion, Carryover, or Expiry Date Change (Type 2, 7, 8)**: Delineate Units
  - For each block, remove flag and update the unit identifiers.
  - For each block, Delineate Units
  - Update Unit Block
  - Block_Ownership_ or_Account_Type
  - Process Running Totals
  - Update Running Totals

- **Replacement (Type 6)**: Delineate Units
  - For the blocks to be transferred, set the unit status code to 0 by calling Delineate_Units.
  - Update Block_ Ownership_Or_ Account_Type
  - Replace Unit Block
  - For each block, establish the relationship between the replaced block and the replacing block.

- **Internal/Supplementary Transaction (Type 10)**
  - Finalise Transaction
    - For each block, set the unit status code to 0 and update the unit holding table indicating current ownership of units.
    - Delineate_Units
    - Update Block_ Ownership_Or_ Account_Type
    - Process Running Totals
    - Update Running Totals

- **External, Cancellation, Retirement, (Type 3, 4, 5)**: Complete the transfer of units to the replacement account.
  - Delineate Units
  - Update Block_ Ownership_Or_ Account_Type
  - Replace Unit Block
  - For each block, establish the relationship between the replaced block and the replacing block.

**Transactions**

- **Issuance (Type 1)**
- **Conversion, Carryover, or Expiry Date Change (Type 2, 7, 8)**
- **Replacement (Type 6)**
- **Internal/Supplementary Transaction (Type 10)**
5.8 Transaction State Diagram

The following state diagram describes the possible state change of a transaction on the ITL. The state of a transaction changes as checks or actions process the transaction through the sequence of steps described in Figures 5.12 and 5.13.

Evaluate STL Notification

Evaluate_STL_Notifcation will record the transaction status sent by the STL and relay the status to the registry.

**What is the Transaction Type?**

- **Non-external** (Type 1, 2, 4, 5, 6, 7, 8, 10)
  - Add new status record indicating the transaction has been checked by the STL and its status
  - Write_Transaction_Status
  - Begin Database Transaction
  - Send_Notification_To_Registry
  - Route Message to Initiating Registry
  - Route Message to Acquiring Registry

- **External** (Type 3)
  - Add new status record indicating the transaction has been checked by the STL and its status
  - Write_Transaction_Status
  - Begin Database Transaction
  - Send_Notification_To_Registry
  - Route Message to Initiating Registry
  - Send_Proposal_To_Registry

**What is the Transaction Status?**

- STL Checked (No Discrepancy)
  - Send_Notification_To_Registry
  - Send_Notification_To_Registry
  - Send_Notification_To_Registry

- STL Checked (Discrepancy)
  - EvaluateSTLNotification
  - Route Message to Acquiring Registry

**Commit Database Transaction**
Figure 5.23: Transaction State Diagram

- **Completed**: [Sent notification to the transferring registry and received confirmation from the transferring registry. Units are marked as useable.]
- **Accepted**: [Transaction is an external transfer. Sent proposal to acquiring registry and received back notification of acceptance.]
- **Proposal is received from registry**
- **Sent notification to transferring registry and received confirmation from transferring registry. Units involved in transaction are marked as useable.**
- **Transaction is validated and units involved in transaction are flagged as non-useable due to proposal.**
- **Proposed**
- **Transaction is not validated and units involved in the transaction are marked as non-useable due to discrepancy.**
- **Checked (No Discrepancy)**
- **Sent notification to transferring registry and received confirmation from transferring registry.**
- **[Sent notification to transferring registry and received termination notification back from transferring registry.]**
- **Cancellable**
- **No update to transaction received within 24 hours. Cancellation notification sent to transferring registry and, if applicable, acquiring registry. Unit involved in transaction marked as useable.**
- **Terminated**
- **Cancelled**
- **[Sent notification to transferring registry and received termination notification from transferring registry.]**
- **[Sent notification to transferring registry and received rejection notification from acquiring registry.]**
- **[Transaction is an external transfer. Sent proposal to acquiring registry and received rejected notification from acquiring registry.]**
- **[Transaction is an external transfer. Sent proposal to acquiring registry and received accepted notification from acquiring registry.]**
- **[Transaction is validated and units involved in transaction are flagged as non-useable due to proposal.]**
- **[Transaction is not validated and units involved in the transaction are marked as non-useable due to discrepancy.]**

---

Annex 13: ITL Software Specification, Version <1.0>, Draft #8
6. Reconciliation Process

The ITL reconciles data on unit holdings in registries periodically on the basis of a data snapshot at an agreed upon time. A reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved through a manual intervention process.

For supplementary program registries, the STL may reconcile data on unit holdings in accounts. For such STLs, a reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved through a manual intervention.

The ITL Administrator will initiate the reconciliation process after coordinating with the Registry Administrator and, if appropriate, the STL Administrator. A reconciliation action may be initiated for any stage; however, only a successful unit count by account type (ITL) or by account (STL) will result in the release of inconsistent blocks.

Two jobs will be used to execute a reconciliation action at a registry. The first job will be scheduled to run a minimum of three hours prior to the agreed upon snapshot time. This job will call the Initiate Reconciliation Web service on the affected registry in order to confirm the reconciliation action. The second job will run at the snapshot time and will create the snapshot and, when finished, will request that the registry send its totals, when they are ready.

See also Section 5 of the data exchange standards (Annex 14).

6.1 Reconciliation Snapshot Data

The reconciliation of data in a registry with the ITL and any STL occurs at a particular point in time. To simplify this process, it is recommended that the ITL, registry and the STL (if applicable), record a snapshot of the Unit Block table to perform the analyses and comparisons for each reconciliation action. The time of the action will be negotiated as a regular, scheduled action at a time convenient to the registry.

The snapshot for a registry will contain the following information:

- Reconciliation ID
- Snapshot DateTime
- Account Type
- Account Number
- Account Commitment Period
- Unit Type
- Originating Registry
- Start Block
- End Block
- Applicable Commitment Period

The structure of the snapshot for the ITL will be:

- Reconciliation ID
- Snapshot DateTime
The structure of the snapshot for the STL checking account/unit holdings is recommended to be:

- Reconciliation ID
- Snapshot DateTime
- Holding Registry
- Account ID
- Unit Type
- Originating Registry
- Start Block
- End Block

Within the ITL, the snapshot will be stored in a database table. All data would be stored until a successful reconciliation is completed, or for a defined period of time following the reconciliation snapshot date, whichever occurs later.

A Reconciliation may require a considerable amount of system resources. It should be considered to include provisions in the system design, to improve performance and to avoid the slow down of normal transaction processing.

6.2 Reconciliation Message Checks

When the ITL receives reconciliation messages from registries in response to its reconciliation request, the following types of checks are performed on the messages (see also Section 5.4). These are followed by the reconciliation evaluation itself, which is detailed in the Reconciliation Process flow diagrams and functions described in Annex E.

Figure 6.1: Reconciliation Check Categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Response Code Range</th>
<th>Category Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version and Authentication</td>
<td>1000 - 1299</td>
<td>Checks to validate version of DES.</td>
<td>Message returned with response codes. Message not placed into message queue.</td>
</tr>
</tbody>
</table>

(cont.)
6.2.1 Version and Authentication Checks for Reconciliation

Preliminary checks, including version and authentication checks, are performed upon receipt of the HTTP SOAP request from a registry and do not involve any interaction with the ITL database. If these checks are passed, the message is placed in the message queue for processing. Failures due to authentication and poorly formed XML content are returned as HTTP SOAP fault errors. Failures due to any reconciliation check are returned in the ResponseObject in an HTTP SOAP response. See Figure 5.3.

6.2.2 Message Viability Checks for Reconciliation

As with transactions, all messages from the queue are checked to determine if they are more than 24 hours old. See Figure 5.4.

6.2.3 Registry Validation Checks for Reconciliation

When the message has been retrieved from the message queue and recorded in the message log, checks are performed to determine if the registries involved in the transaction are identifiable and eligible to participate. See Figure 5.5.

Figure 6.1: Reconciliation Check Categories (cont.)

<table>
<thead>
<tr>
<th>Category</th>
<th>Response Code Range</th>
<th>Category Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconciliation Data Integrity</td>
<td>6000 - 6299</td>
<td>Basic checks of data content including numeric ranges and validity of codes.</td>
<td>Message returned with response codes. Message not logged in Reconciliation Log table.</td>
</tr>
<tr>
<td>Reconciliation Message Sequence</td>
<td>6300 - 6399</td>
<td>Checks to validate message order and reconciliation status.</td>
<td>Message returned with response codes. Message not logged in Reconciliation Log table.</td>
</tr>
</tbody>
</table>

Figure 6.2: Additional Registry Checks for Reconciliation

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1510</td>
<td>Registry Available for Reconciliation Action</td>
<td>Registry status must allow reconciliation actions to be conducted.</td>
</tr>
</tbody>
</table>
6.2.4 Data Integrity Checks for Reconciliation

This category of checks is performed by the Reconciliation_Data_Integrity_Check function to identify incoming messages containing data that fail basic data integrity checks. If any data in a message fail these checks, the message is returned to the sender with an appropriate response code. The message is not logged in the Reconciliation Log table and is not processed further. Data integrity checks are critical checks in that if they result in failure, no further checks should be processed.

Note that as part of reconciliation, transactions and unit blocks are passed into the ITL, but those items are minimally checked by the data integrity checks. If there is a problem with the format of a transaction or a unit block, the reconciliation process will identify and log those items as the source of an inconsistency.

Figure 6.3: Summary of Reconciliation Data Integrity Checks

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6201</td>
<td>Reconciliation Identifier</td>
<td>Reconciliation Identifier must be greater than zero.</td>
</tr>
<tr>
<td>6202</td>
<td>Reconciliation Mask</td>
<td>Reconciliation ID must be comprised of a valid registry code followed by numeric values.</td>
</tr>
<tr>
<td>6203</td>
<td>Reconciliation Status Validity</td>
<td>Reconciliation Status must be a value between 1 and 11.</td>
</tr>
<tr>
<td>6204</td>
<td>Reconciliation Snapshot Date/Time</td>
<td>Reconciliation Snapshot must be a date between 01-OCT-2004 and the current date plus 30 days.</td>
</tr>
<tr>
<td>6205</td>
<td>Account Type Validity</td>
<td>Account Type Code must be valid.</td>
</tr>
<tr>
<td>6206</td>
<td>Unit Type Validity</td>
<td>Unit Type Code must be valid.</td>
</tr>
<tr>
<td>6207</td>
<td>Supplementary Unit Type Validity</td>
<td>Supplementary Unit Type Code must be valid.</td>
</tr>
</tbody>
</table>

6.2.5 Message Sequence Checks for Reconciliation Messages Received from Registries

After the data in the message have been checked, the ITL performs checks to ensure that the message received from a registry has been submitted in the proper sequence.
### Figure 6.4: Sequence Checks for Registry Messages

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6301</td>
<td>Reconciliation ID Does Not Exist</td>
<td>Reconciliation ID must exist in the Reconciliation Log table.</td>
</tr>
<tr>
<td>6302</td>
<td>Reconciliation Status Not Valid</td>
<td>Reconciliation status sent by registry must be valid.</td>
</tr>
<tr>
<td>6303</td>
<td>Reconciliation Status Out of Sequence</td>
<td>Incoming reconciliation status should be the same as the reconciliation sequence recorded at the ITL.</td>
</tr>
<tr>
<td>6304</td>
<td>Reconciliation Snapshot DateTime</td>
<td>The registry reconciliation snapshot DateTime must be consistent with the ITL reconciliation snapshot DateTime.</td>
</tr>
</tbody>
</table>

#### 6.2.6 Message Sequence Checks for Reconciliation Messages Received from STL

After the data in the message have been checked, the ITL performs checks to ensure that the message received from an STL has been submitted in the proper sequence.

### Figure 6.5: Sequence Checks for STL Messages

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6311</td>
<td>Reconciliation ID does not exist.</td>
<td>Reconciliation ID sent by the STL must already exist in the ITL.</td>
</tr>
<tr>
<td>6312</td>
<td>Reconciliation Status Not Valid</td>
<td>Reconciliation status sent by the STL must be one of certain enumerated statuses.</td>
</tr>
<tr>
<td>6313</td>
<td>Reconciliation Status of &quot;STL Totals Inconsistent&quot; is Out of Sequence</td>
<td>If the incoming reconciliation status is &quot;STL Totals Inconsistent,&quot; the previously recorded status at the ITL must be &quot;Validated.&quot;</td>
</tr>
<tr>
<td>6314</td>
<td>Reconciliation Status of &quot;STL Unit Blocks Inconsistent&quot; Out of Sequence</td>
<td>If the incoming reconciliation status is &quot;STL Unit Blocks Inconsistent,&quot; the previously recorded status at the ITL must be &quot;STL Totals Inconsistent.&quot;</td>
</tr>
<tr>
<td>6315</td>
<td>Reconciliation Message Out of Sequence</td>
<td>If incoming reconciliation status is &quot;STL Validated,&quot; the previously recorded status at the ITL must be &quot;STL Totals Inconsistent,&quot; or &quot;STL Unit Blocks Inconsistent.&quot;</td>
</tr>
<tr>
<td>6316</td>
<td>Reconciliation Message Out of Sequence</td>
<td>If the incoming reconciliation status is &quot;STL Complete with Manual Intervention,&quot; the previously recorded status at the ITL must be &quot;STL Totals Inconsistent,&quot; or &quot;STL Unit Blocks Inconsistent.&quot;</td>
</tr>
</tbody>
</table>
6.2.7 Other Reconciliation Responses

The following response codes provide information about the results of the reconciliation analyses to the registry or STL.

**Figure 6.6: Other Reconciliation Checks and Messages**

<table>
<thead>
<tr>
<th>Response Code</th>
<th>Check Name</th>
<th>Check Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6410</td>
<td>Account Type/Unit Type Totals</td>
<td>The totals for account types, Commitment Period and unit types must be consistent.</td>
</tr>
<tr>
<td>6420</td>
<td>Account Type/Unit Type Unit Blocks</td>
<td>The registry and ITL unit blocks for each specified account type, Commitment Period and unit type must be consistent.</td>
</tr>
<tr>
<td>6430</td>
<td>Account Type/Unit Type Unit Blocks</td>
<td>If the totals have failed in the previous stage, the Unit Block compare by account type, Commitment Period, and unit type must also fail.</td>
</tr>
<tr>
<td>6430</td>
<td>Unexpected Consistency</td>
<td></td>
</tr>
<tr>
<td>6440</td>
<td>Snapshot DateTime Validity</td>
<td>The DateTime for reconciliation action proposed by the STL must be in the future.</td>
</tr>
<tr>
<td>6450</td>
<td>Ongoing Reconciliation</td>
<td>A reconciliation action cannot be initiated at this registry because there is already an ongoing reconciliation action.</td>
</tr>
<tr>
<td>6600</td>
<td>Successful Reconciliation of Totals</td>
<td>The reconciliation has been completed with a successful reconciliation of unit totals.</td>
</tr>
</tbody>
</table>
6.3 Activity Diagrams

Figure 6.7: Reconciliation Process Flow Stage 1 - Validate Account Totals

[Diagram showing the process flow with stages and activities.]

Stage 1

- **InitiateReconciliation Request**: Send reconciliation request to registry, Send acknowledgement response.
- **Create Snapshot**: Send totals request to registry, Send acknowledgement response.
- **ProvideTotals Request**: Send response back from Web service.
- **Generate Totals**: Send request for account information if supplementary program member.
- **Generate HTTP SOAP Request**: Send response back that Account Type Totals were received and logged or return SOAP faults.
- **Call ReceiveTotals**: Send acknowledgement response.
- **Check Reconciliation Message**: Send response back that Account Type Totals were received and logged or return SOAP faults.
- **InitiateReconciliation Request**: If registry is part of an STL, send notification of Snapshot DateTime.
- **Generate HTTP SOAP Request**: If no SOAP faults or preliminary check failures, then.
- **Create Snapshot**: Generate HTTP SOAP Request.
- **ProvideTotals Request (By Account)**: Validate Totals (Fig. 6.14).
- **Generate Account Totals**: Validate Totals (Fig. 6.13).
- **Generate HTTP SOAP Request**: Validate Totals (Fig. 6.14).
- **Call ReceiveTotals**: Validate Totals (Fig. 6.13).
- **HTTP SOAP Response**: Validate Totals (Fig. 6.14).
- **Web service terminates**: Send acknowledgement response.
- **Receive Totals**: Web service terminates.
- **Annex 13: ITL Software Specification, Version <1.0>, Draft #8**
Figure 6.8: Reconciliation Process Flow Stage 2 - Validate Unit Blocks
Figure 6.9: Reconciliation Process Flow Stage 3 - Review Audit Logs

Stage 3

- Registry
- Communications Hub
- ITL
- STL

ProvideAuditTrail Request
Generate Audit Trail
Generate HTTP SOAP Request
Wait for response back from Web service

Call ReceiveAuditTrail

Authenticate Registry (VPN)
Check for Well Formness (WSDL)
If no SOAP faults or preliminary check failures, then

Version Checks
HTTP SOAP Response

Send response back that audit data was received and logged or return SOAP faults

Write_Audit_Trail_TO_File

Request_Audit_Trail
Generate HTTP SOAP Request
Send Acknowledgement Response

Evaluate Registry Reconciliation Message (Fig. 6.12)
Write_Audit_Trail_TO_File

6.4 Reconciliation Processing Flow Diagrams

Figure 6.10: Confirm Reconciliation

sd Confirm_Reconciliation

This function opens a reconciliation action and confirms the snapshot datetime with the registry.

Input: Registry Code, Data Snapshot DateTime

Set Begin Transaction for database

Begin Database Transaction

:Write_To_Reconciliation_Log

Yes

Is there an ongoing reconciliation action for this registry?

No

:Write_To_Reconciliation_Status

:Send_Reconciliation_Snapshot_DateTime

:Close_Reconciliation_Action

:Trade_Scheme_Member

Yes

If the registry is part of an STL, notify the STL of the date and time the reconciliation will be performed.

No

:Send_Reconciliation_Snapshot_DateTime

Set Commit Transaction for database

Commit Database Transaction

Confirm Reconciliation

Indicate the status of the reconciliation is "Start Denied" (reconciliation status code 7)

Update the End Date for this reconciliation action in the reconciliation log.

Record a new reconciliation action.

Record a new reconciliation action.

Add the End Date for this reconciliation action in the reconciliation log.

Annex 13: ITL Software Specification, Version <1.0>, Draft #8
Figure 6.11: Receive Registry Reconciliation Message

This diagram describes the process of receiving a reconciliation message from a registry.

Web services called by registry to pass information to the ITL and STL.

Preliminary Checks execute the most basic checks before adding the message to the queue.

Validate the message.

Route the message.

sd Receive Registry Reconciliation Message

:ReceiveTotals

:ReceiveUnitBlocks

:ReceiveAuditTrail

:Evaluate_Registry_Reconciliation_Message

:Check_Reconciliation_Message

:Preliminary_Checks

This process shows how the ITL validates a reconciliation message.

1. Begin Database Transaction
2. Retrieve From Message Queue
3. Write To Message Log

- Check if registry has correct status to participate in a transaction (0 or 1)
- Check message age
- Application-level checks for data inputs supplied to the Web service
- Check to determine if message is in the correct sequence for processing

- Call :Evaluate_Response_Object to determine if queue processing checks failed
- Commit Database Transaction
- Set Begin Transaction for database
- Message remains in queue.
- Set Commit Transaction for database
- Terminate any further processing and expire message from queue.
- Set Commit Transaction for database

Did any checks fail?

- Yes:
  - :Reconciliation_Failure_Notification
  - Commit Database Transaction
  - :Remove_From_Message_Queue

- No:
  - :Reconciliation_Data_Integrity_Check

- Did any checks fail?

- Yes:
  - :Message_Viability_Check
  - Was message more than 24 hours old?

- No:
  - :Reconciliation_Message_Sequence_Check

- Did any checks fail?

- Yes:
  - :Evaluate_REGISTRY_Reconciliation_Message
  - :Evaluate_STL_Reconciliation_Message

- No:
  - :Evaluate_REGISTRY_Reconciliation_Message

From where did the message originate?

- Any Registry
- STL
Figure 6.13: Evaluate Registry Reconciliation Message

This diagram describes how to route an incoming message from a registry.

1. **Begin Database Transaction**
   - **What is the Reconciliation Status?**

2. **1st Stage of ITL Reconciliation**
   - **Validate Totals (2)**
   - **Totals Inconsistent (3)**
   - **Initiated (1) with Start Phase = 2**
   - **Initiated (1) with Start Phase = 1**
   - **Validated (2)**

3. **2nd Stage of ITL Reconciliation**
   - **Validate Unit Blocks**

4. **3rd Stage of ITL Reconciliation**
   - **Accept Audit Trail**

5. **Commit Database Transaction**
   - **Remove From Message Queue**

The ITL has completed its reconciliation and forwards the message to the STL.

- **STL Totals Inconsistent (8)**
- **STL Unit Blocks Inconsistent (9)**

Begin Database Transaction

Set Begin Transaction for database

"Initiated" (1) with Start Phase = 1

The ITL has completed its reconciliation and forwards the message to the STL.

- **Forward Totals To STL**
- **Forward Unit Blocks To STL**
- **Forward Audit Trail To STL**

Set Commit Transaction for database and remove from queue.
Figure 6.14: Validate Totals

sd Validate_Totals

This function validates the totals submitted by a registry and initiates the next step in reconciliation.

Compare Totals

Verify that the totals (by account type and unit type) presented by the registry agree with the ITLs records

Are the totals consistent?

Yes

Update the reconciliation status to "Validated" (reconciliation status code 2)

No

Update the reconciliation status to "Totals Inconsistent" (reconciliation status code 3)

:Write_To_Reconciliation_Status

Call Trade_Scheme_Member

Is the registry part of the supplementary program system?

Yes

Request totals (by account) so that reconciliation can continue at the STL

No

Notify registry of successful reconciliation

:Send_Reconciliation_Notification_To_Registry

:Delete_Inconsistent_Block

Continue to next step of reconciliation by requesting unit blocks from registry

:Request_Unit_Blocks_From_Registry

If any blocks remain frozen from a previous reconciliation, delete the Inconsistent_Block and update Unit_Block table so that those blocks may be used in a transaction

:Request_Totals_From_Registry

If any blocks remain frozen from a previous reconciliation, delete the Inconsistent_Block and update Unit_Block table so that those blocks may be used in a transaction

This function validates the totals submitted by a registry and initiates the next step in reconciliation.
This function will compare each unit block sent by the registry against the ITL records. If blocks do not match up, they will be marked as inconsistent.

Check if the unit blocks sent by the registry are consistent with the ITL records.

- **For each inconsistent block:**
  - **:Insert_Inconsistent_Block**
  - **:Write_To_Reconciliation_Status**: "Unit Blocks Inconsistent" (Reconciliation Status Code 4)
  - **:Request_Audit_Trail**

If no inconsistencies are found, a manual intervention is needed to explain why the first step of reconciliation (unit total counts by account type) failed.

Initiate next step of the reconciliation process by requesting a record of all transactions since the last successful reconciliation involving the inconsistent blocks.

Annex 13: ITL Software Specification, Version <1.0>, Draft #8
Figure 6.16: Receive Audit Trail

This diagram describes the process of accepting and recording the transaction history of inconsistent unit blocks.

Write Audit Trail To File

Write to file

The audit trail is stored in XML format in a file.

:Write_To_Message_Log

The message log will record the name of the file created containing the audit trail.

Figure 6.17: Manual Intervention

The System Administrator will compare the audit trail to the ITL transaction history and coordinate a data correction with the registry.

Administrator Review

Manual Intervention

Data is changed to correct the inconsistency at the registry and/or the ITL.

ITL Administrative Application captures changes made for Manual Intervention

Any changes made at the ITL will be recorded by the administrative application.

Change Reconciliation Status to "Manual Intervention Complete"

The System Administrator updates the reconciliation status.

Initiate New Reconciliation Action

A new reconciliation is initiated to verify the registry and the ITL are in sync.
This diagram describes the process of receiving a reconciliation message from the STL.

- Receive STL Reconciliation Notice
- Provide Unit Blocks
- Provide Totals
- Receive Reconciliation Result
- Preliminary Checks execute the most basic checks before adding the message to the queue
- Reconciliation Message Sequence Check
- Evaluate STL Reconciliation Notice

Web services called by the STL to pass information to the ITL and registries.
This diagram describes all the processes that result from receiving a reconciliation message from the STL.

Set Begin Transaction for database

"STL Totals Inconsistent" (8)
Update the reconciliation status to "STL Totals Inconsistent" (Reconciliation Status Code 8).

:Write_To_Reconciliation_Status
Continue to next step of STL Reconciliation by requesting Unit Blocks from registry

:Request_Unit_Blocks_from_Registry
For each block

:Insert_Inconsistent_Block
Record blocks that do not match STL records

:Write_To_Reconciliation_Status
Update the reconciliation status to "STL Unit Blocks Inconsistent" (Reconciliation Status Code 9).

:Write_To_Reconciliation_Status
Continue to next step of STL Reconciliation by requesting Audit Trail from registry

:Request_Audit_Trail

Set Commit Transaction for database

Commit Database Transaction

"STL Validated" (10)
Update the reconciliation status to "STL Validated" (Reconciliation Status Code 10).

:Send_Reconciliation_Notification_To_Registry
Notify registry of successful reconciliation

:Close_Reconciliation_Action
Update the Reconciliation Log to end the reconciliation action

"STL Complete with Manual Intervention" (11)
Update the reconciliation status to "STL Complete with Manual Intervention" (Reconciliation Status Code 11).

:Delete_Inconsistent_Block
Update the Reconciliation Log to end the reconciliation action

:Send_Reconciliation_Notification_To_Registry
Notify registry of completed reconciliation

:Close_Reconciliation_Action
Update the Reconciliation Log to end the reconciliation action
6.5 Reconciliation State Diagrams

Figure 6:20: Registry - ITL Reconciliation State

- [Request for totals is sent to registry]
- [Totals received from registry]
- [Totals match with ITL and positive reconciliation result is sent to registry]
- [Totals do not match with ITL and a request for unit blocks is sent to the registry.]
- [Unit blocks received from registry and inconsistent unit blocks are flagged.]
- [Registry asked to provide audit trail and receive audit trail from registry.]
- [Inconsistency has been manually resolved and confirmation sent to registry. Units involved in inconsistency are marked as useable.]
Figure 6.21: ITL - STL Reconciliation State

[STL requests ITL to start a reconciliation action]

[Registry must be ready to provide their totals:
- by account type
- by commitment period
- by unit type
- by account type and unit type]

[ITL Initiates Reconciliation]

[STL requests Unit Blocks to registry and receives unit blocks. Forward unit blocks to STL]

[STL Totals Inconsistent]

[ITL forwards request for audit trail to the registry and receives the audit trail from the registry. The ITL forwards the audit trail to the STL]

[Audit Trail Available]

[STL and registry manually resolve Inconsistency. ITL receives notification of completed reconciliation.]

[STL Manual Resolved]

[ITL Initiates Reconciliation]

[STL Totals Available]

[Request for totals by account is sent to registry. Totals received and forwarded to STL]

[Validated]

[Request for totals by account type, by commitment period, by unit type, and by account type and unit type is sent to registry. Totals received and forwarded to STL]

[STL Validated]

[STL notifies ITL of matching totals]

[ITL notifies registry of successful reconciliation]

[Complete]

[Process continues with previous diagram]

[Totals do not match with the ITL and a request for unit blocks is sent to the registry]

[Totals match with ITL]

[STL requests ITL to start a reconciliation action]
7. Administrative Processes

See also DES Section 6.

7.1 Transaction Status

The ITL will make available to registries the current status of all transactions. The GetTransactionStatus Web service will provide the current status recorded at the ITL to a requesting registry. The registry will initiate the Web service on the ITL and pass it a transaction ID. The ITL will return the status of the transaction and the date and time the status was last updated. These requests are not recorded or tracked by the ITL. If the requested transaction number is not found in the ITL, Response Code 8001 (Transaction ID Not Found) is returned. See Get_Transaction_Status in Annex E.

7.2 Transaction Clean-up Process

In order to maintain data integrity and to ensure that registries adhere to established timing requirements, on a periodic basis the ITL identifies transactions that are in progress and for which a message has not been received within 24 hours. This check shall be performed once an hour. The ITL cancels these transactions. After the transaction is cancelled, the unit status is modified such that they are available to be involved in another transaction. Response Code 8002 (Transaction Clean-up) is sent to the registries involved in the transaction through the AcceptNotification Web service. The system administrators of the registries should review the notification, investigate the reason for the lack of communication, and reinitiate the transaction as a new transaction, if appropriate. See Figure 7.1 and Transaction_Clean-up in Annex E.

7.3 Notifications

The ITL performs administrative functions that evaluate data and inform registries of specific required actions. Each of the following processes may result in the sending of a notification to one or more registries regarding actions that must be taken. All notifications are sent via the AcceptITLNotice Web service. Their initiation is prompted by the receipt of reference information through the ITL automatic interfaces to secretariat data systems (see Annex A) and should be automated to the extent possible.

7.3.1 Net Source Cancellation

In the case that the review and Compliance Committee procedures under the Kyoto Protocol find that the LULUCF activities of a Party have resulted in a net source of emissions, the ITL will notify the Party of the quantity of units it is required to cancel within 30 days as part of a net source cancellation action. This notification will use the AcceptITLNotice Web service at the registry. The units must be cancelled into a Net Source Cancellation Account (Account Type Code 210). The registry will initiate cancellation transactions, providing reference to the identifier of the notification sent by the ITL so the ITL can track when the registry has completed the required cancellation. See Net_Source_Cancellation in Annex E.

This notification will be initiated by an administrator through the ITL Administrator Application, on the basis of reference information received from the compilation and accounting database.
Figure 7.1: Transaction Clean-up Process

Clean-Up Process

This process identifies transactions that are more than 24 hours old, cancels those transactions, and notifies the involved parties.

Search Transaction Log for all transactions that are in progress

Check Age of each Transaction

Was the last message of the transaction initiated more than 24 hours ago?

Yes

Call Write_Transaction_Status with "Cancelled" (7)

For each block

Free the units involved in the transaction to be used in another transaction

:Delineate_Units

Inform the Initiating Registry the transaction was cancelled.

:Send_Notification_To_Registry

No

Continue with next transaction

Is this an external transaction (type 3)?

Yes

Inform the Acquiring Registry the transaction was cancelled.

:Send_Notification_To_Registry

Call :Trade_Scheme_Member

Is Initiating or Acquiring Registry participating in a supplementary program?

Yes

Inform STL the transaction was cancelled

:Send_To_STL

No

Continue with next transaction

Repeat for next transaction
7.3.2 Non-compliance Cancellation

In the case that the Compliance Committee determines that a Party is in non-compliance with its emissions target under Articles 3.7 and 3.8 of the Kyoto Protocol, the ITL will notify the Party of the quantity of units it is required to cancel within 30 days as part of a non-compliance cancellation action. This notification will use the AcceptITLNotice Web service at the registry. These units must be cancelled into the Non-compliance Cancellation Account (Account Type Code 220). The registry will initiate cancellation transactions, providing reference to the identifier of the notification sent by the ITL so the ITL can track when the registry has completed the required cancellation. See Non-compliance_Cancellation in Annex E.

This notification will be initiated by an administrator through the ITL Administrator Application, on the basis of reference information received from the compilation and accounting database.

7.3.3 Impending tCER or lCER Expiry

Every 24 hours the ITL will execute a job to identify any tCERs or lCERs that will expire within 30 days. After identifying these units, the ITL will notify each registry that holds one of the units that the units must be cancelled or replaced within 30 days. These transactions must include the Notification ID sent to the registry by the ITL.

This notification will be initiated by a batch job that runs periodically and searches for units that are about to expire. The batch job will send a notification to each registry that holds units that are about to expire. See CER_Expired_Check in Annex E.

If tCERs or lCERs expire, a report within the ITL administrative application will highlight any registries that hold expired units. The ITL Administrator will notify the Executive Board that the registry has failed to take appropriate steps to deal with an expired tCER or ICER.

7.3.4 Reversal of Storage for CDM Project

If a reversal of storage of greenhouse gasses occurs at a project, that has resulted in lCER issuance, the CDM Executive Board will provide reference information on the case to the ITL via the CDM information system. A job will determine and then notify registries of the actions they must take. The job will temporarily suspend trading of all units associated with the project and then calculate how many units each registry must replace. Each registry must replace the same percentage of their holdings (excluding cancelled or previously replaced units) as the percentage of the reduction in storage. Each affected registry will be notified through the AcceptITLNotice Web service method. The message will alert each registry to the number of units it must replace.

The registry will then initiate replacement transactions until the appropriate number of lCERs have been replaced. The replacement transactions submitted by the registry must reference the identifier of the notification sent by the ITL so that the ITL can track when the registry has completed replacement.

This notification will be initiated by an administrator, on the basis of reference information received from the CDM information system. The job will be triggered that identifies the registries holding units associated with that project and calculates the number of units each registry must cancel. A notification is sent to each registry indicating the number of units associated with the project that must be cancelled or replaced. See Reversal_of_Storage in Annex E.
Another job will run daily and will check for non-replacement by each registry with the Reversal of Storage Notification. The job will search the notification table for open notifications of this type and then evaluate each registry’s response by searching for transactions that reference the applicable Notification ID. Each registry has 30 days to complete replacements.

The ITL Administrator will periodically report to the Executive Board regarding actions.

7.3.5 Non-submission of Certification Report for CDM Project

If the persons responsible for a project, which has resulted in CER issuance, have not submitted a certification report within the specified timeframe of five years since the last report, the CDM Executive Board will provide reference information on the case to the ITL via the CDM information system. A job will determine and then notify registries that all CERs issued for the project must be replaced. The job will suspend trading of all units associated with the project. The notifications will be sent through the AcceptITLNotice Web service method at the registry.

The registry will initiate cancellation or replacement transactions until all the units have been addressed. These transactions will reference the Notification ID sent to the registry by the ITL.

This notification will be initiated by an administrator, on the basis of reference information received from the CDM information system. The job will be triggered that identifies the registries holding units associated with that project. A notification is sent to each registry identifying each block that must be cancelled or replaced at that registry. See Lack_of_Certification_Report in Annex E.

7.3.6 Excess Issuance for CDM Project

At the request of the CDM Executive Board, in the case that a designated operational entity (DOE) is required to cancel units within 30 days to compensate for excess CERs having been issued for a CDM Project, the ITL will notify registries of the required cancellation to be undertaken by the DOE and provide the Notification ID to the CDM Executive Board for forwarding to the DOE. This notification will use the AcceptITLNotice Web service at registries. These units must be transferred into the Mandatory Cancellation Account (Account Type Code 250) at the CDM Registry. The DOE will then initiate transactions, via registries, providing reference to the Notification ID so that the ITL can track when the required cancellation has been completed.

This notification will be initiated by an administrator, on the basis of reference information received from the CDM information system. A job will be triggered that identifies the registries holding units associated with that project. A job will enter a record in the notification table and send an email to the CDM Executive Board. The notification and email will indicate the number of units that must be cancelled or replaced. See Excess_Issuance_for_CDM_Project in Annex E.

7.3.7 Commitment Period Reserve Change

In the case of an upward revision of the CPR level of a Party above its current holdings of units, the ITL will notify the Party of the quantity of units by which it is required to increase its unit holdings within 30 days. This notification will use the AcceptITLNotice Web service at the registry. The registry will then acquire the necessary number of units from other registries. Since
transactions are submitted by the transferring, not acquiring, registry, these transactions will not reference any Notification ID.

This notification will be initiated by an administrator, on the basis of reference information received from the compilation and accounting database. A job will be triggered that checks the compliance of the registry with the new CPR. If the CPR has been raised and the registry is no longer in compliance with the CPR, a notification will be sent to the registry indicating the number of units it must acquire. See CPR_Change in Annex E.

7.3.8 Unit Carry-over

After the end of a commitment period and the true-up period, the ITL administrator will notify each registry of units which it may carry over to the subsequent commitment period. Any units not carried over in this manner must be cancelled. The ITL will send a notification through the AcceptITLNotice Web service at the registry for each unit type held by a registry. The registry will carry over the number of units of each type or will cancel the units to the Mandatory Cancellation Account (Account Type Code 250). These transactions will reference the Notification ID sent to the registry by the ITL.

A report within the ITL AA will identify those registries that have not addressed all applicable units within 30 days.

This notification will be initiated when an administrator, on the basis of reference information received from the compilation and accounting database. A notification will be sent to the registry that indicates the number of units of that type that may be carried over and identifies all the units of that type that must be either carried over or cancelled. See Outstanding_Unit_Cleanup in Annex E.

7.3.9 Notification Update

After an initial notification has been sent, the ITL may send an additional notification when a registry fulfills its obligation or to update a registry's progress towards meeting the requirement. The Notification ID for this message will be the same as the original, but the notification type will indicate that this is a notification update. The message content will contain remarks that reference the original notification, indicate whether or not the requirement has been met, indicate the number of days left to fulfill the requirement, and update the number of units the registry must address. The notification update is provided for informational purposes only, and the registry does not need to respond to it. See Notification_Requirement_Attainment_Check in Annex E.

7.4 Registry Time Synchronization

In order to maintain consistent system time between the registries and the ITL, the ITL checks the system time of each registry on a periodic basis. If the time is found to be unsynchronized by a specified amount, a message is sent to the system administrator of that registry. In order to accommodate this function each registry must make available a ProvideTime function which is used by the ITL to retrieve the current time of the registry.

Registries must implement the ProvideTime public web service method for the ITL to call. The ITL will compare the time this function returns with the official system time. Detailed specifications for the ProvideTime method are in Annex D to the DES.
The ITL will log the time synchronization result in the System Log and contact the registry manager using a manual process or through a general message if a time problem is identified. See Time_Sync in Annex E.

7.5 General Messages

The AcceptMessage Web service at a registry may be used to deliver general messages to the Registry Administrator. These messages could involve planned ITL maintenance periods, change management, time synchronization problems, or other operational issues and plans.

7.6 Yearly Aggregated Holdings Report

The ITL Administrator Application will include a report that displays aggregated information on unit holdings by unit type and account type in each registry at the end of each calendar year. In order to facilitate this report the ITL will take and store a snapshot of all unit holding data. This snapshot will resemble the snapshots for reconciliation, but will include data from all registries. See Yearly_Aggregated_Holdings_Snapshot in Annex E.