DATA EXCHANGE STANDARDS FOR
REGISTRY SYSTEMS UNDER THE KYOTO PROTOCOL

DRAFT
TECHNICAL SPECIFICATIONS (Version 1.0, Draft #7)

Non-paper

November 3, 2004

Revision History

Date	Version	Description	Author
31/10/2003	1.0	Draft #1	Andrew Howard
14/11/2003	1.0	Draft #2	Andrew Howard
21/11/2003	1.0	Draft #3	Andrew Howard
27/05/2004	1.0	Draft #4	Andrew Howard
08/06/2004	1.0	Draft #5	Andrew Howard
30/07/2004	1.0	Draft #6	Andrew Howard
03/11/2004	1.0	Draft #7	Andrew Howard

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

This document contains technical specifications for data exchange between registries and the Independent Transaction Log (ITL) under the Kyoto Protocol. This exchange of data forms the technical basis for transactions under the mechanisms defined in Articles 6, 12 and 17 of the Kyoto Protocol and the modalities for the accounting of assigned amounts (to demonstrate compliance with emission targets) under Article 7.4 of the Kyoto Protocol.

These technical specifications contain full information on *how* the data exchange standards are to be implemented. They are based on the functional specifications for data exchange, which define in broader terms *what* data are exchanged and *by whom*. The technical specifications are necessary to ensure that the registries and the ITL employ consistent data exchange and messaging functionality.

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 functionality 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 undergoing development is the Community Independent Transaction Log (CITL) for the European Union greenhouse gas emissions trading scheme. This is being developed under Article 20 of EU Directive 2003/87/EC.

1.2 Intended Audience

This document is to guide technical experts in the design, development and implementation of communication functionality in registries and the ITL.

1.3 Scope

The data exchange standards define how data are to be exchanged between national registries, the CDM Registry and the ITL under the Kyoto Protocol, as well as any STLs established. The Technical Specifications include the communication protocols to be used and a messaging architecture that includes an overall design for message management, message content, and data transfer formats. They define in detail the specific data elements to be exchanged between registry systems to support designated functionality throughout the process.

The diagram in Figure 1-1 demonstrates how both national registries and the CDM Registry will both send and receive messages enabling two-way communications exchanges to the ITL through a Communications Hub. The figure further demonstrates how messages will be forwarded to any STLs and will be returned by them to the ITL.

These technical specifications include:

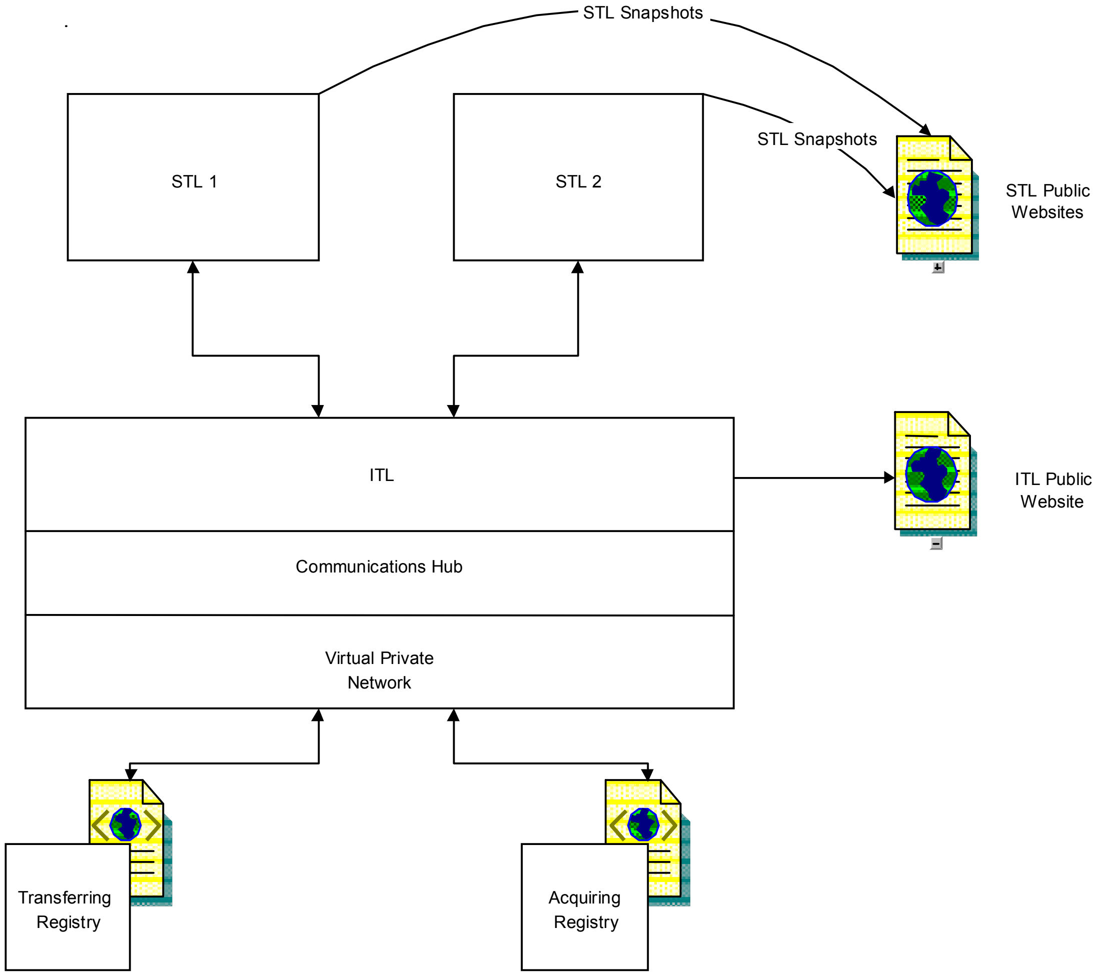
Section 2 Assumptions and Constraints

Facts and constraints identified in the Functional Specifications and held to be true for the Technical Specifications to be valid.

Section 3 Data Exchange Mechanism Specifications

Specifications relating to registration, authentication and communication protocols required.

Figure 1.1: Communication Via the Data Exchange Standards



Section 4 Unit Transaction Processes

Specifications for message exchange relating to unit transactions.

Section 5 Reconciliation Processes

Specifications relating to the process of reconciliation.

Section 6 ITL Administrative Processes

Specifications regarding notifications to registries, time checks and message cancellations.

Section 7 Data Logging Specifications

Specifications for retaining records, utilizing internal logs and communicating transaction data for reconciliation.

Section 8 Change Management Specifications

Specifications to manage and distribute information on changes in the data content, messages, or message sequences to accommodate new requirements.

Section 9 **Registry Initialization Specifications**

Specifications on the start-up processes a registry will be required to complete before initiating communication and data exchange with the ITL.

Annex A **Glossary of Terms**

This annex provides definitions, acronyms and abbreviations relevant to this document.

Annex B **Web Service Operations and Functions for Transaction Processing**

This annex contains the detailed specifications for the Web services and programming functions which receive and/or generate transaction messages.

Annex C **Web Services Operations and Functions for Reconciliation**

This annex contains the detailed specifications for the Web services and programming functions relating to reconciliation.

Annex D **Web Service Operations and Functions for Administrative Processes**

This annex contains the detailed specifications for the Web services and programming functions which receive and/or generate administrative messages.

Annex E **List of Checks and Response Codes for Transaction Processing**

This annex identifies the categories of transaction responses and provides a numeric list of responses.

Annex F **Definition of Identifiers**

This annex provides detailed specifications and rules for creating and using identifiers for entities for which information is exchanged.

Annex G **List of Codes**

This annex identifies the codes which are used to represent a variety of categories, types, and statuses which may be contained in messages.

Annex H **Test Protocols for Data Exchange Specification Implementation**

This annex addresses the test requirements for verifying conformance with the Data Exchange Specifications Version 1.0.

Annex I **Messaging Service Specification**

This annex provides information on the required XML message structure.

Annex J **QA Checklist by Requirement**

This annex lists the requirements in the "Data Exchange Standards for Registry Systems under the Kyoto Protocol: Functional Specifications, Draft Version <7.0>" and cross references the sections of the Technical Specifications which address them.

Annex K **Descriptive Langage (WSDL) Documentation**

This annex provides the WSDLs for the Web services required for message exchange between a registry and the ITL and examples for each transaction type.

Annex L WSDL Examples and Instructions

This annex provides additional information and examples about how data should be provided for each transaction and notification type.

1.4 Definitions, Acronyms, Abbreviations and Terminology

See the glossary in Annex A for definitions, acronyms and abbreviations relating to the Kyoto Protocol and related policy documents defining how the Protocol is to be implemented.

This list is intended to promote a common understanding of terminology which is critical to understanding and interpreting the Technical Specifications, and to ensure that developers and policy analysts use a common vocabulary for describing and discussing the specifications for data exchange.

It is important that readers familiarize themselves with these definitions, as the usage of many terms in this document are specific to these technical specifications and are not generic.

Note in particular that the term "registries" refers to both national registries and the CDM Registry. "Registry systems" refers to both registries and the ITL.

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

- 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

1.6 Multiple Language Support

With the exception of the country codes which utilize the alpha codes in ISO3166, all message content exchanged is represented as numeric values. The numeric codes are listed in Annex G. Therefore, the content of all messages is independent of a specific language.

1.7 Validity of Data

The non-functional requirements for registries and the ITL require accuracy and data integrity. These requirements are addressed throughout these technical specifications, including in particular, the requirements for data elements and message content. The reconciliation process also provides assurance that these non-functional requirements will be met.

2. Assumptions and Constraints

These technical specifications are based upon the derivation documents specified in Section 1.5. In particular, they are based upon the constraints and requirements contained in the Functional Specifications for the Data Exchange Standard. A detailed cross reference of these technical specifications and the specific requirements is included in Annex J.

2.1 Assumptions

The ITL will have access to the Compilation and Accounting Database (C & A Database) of the Secretariat.

The ITL will receive information regarding CDM Projects from the CDM Executive Board and information regarding track 2 joint implemenation projects from the Article 6 Supervisory Committee.

2.2 Constraints

These data exchange standards utilize the following standards:

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

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

- WSDL
  http://www.w3.org/TR/wsdl

3. Data Exchange Mechanism Specifications

3.1 General Requirements

Communications between the registries and the ITL must be secure and processed as real-time transactions. The Functional Specifications for the Data Exchange Standards specify the use of TCP/IP connections using encrypted messages over the Internet. Communications must be protected from modification or interception in transit. Users must be authenticated to ensure their identity and associated permissions. Communications will be initiated by either registries or the ITL and an immediate response will be expected.

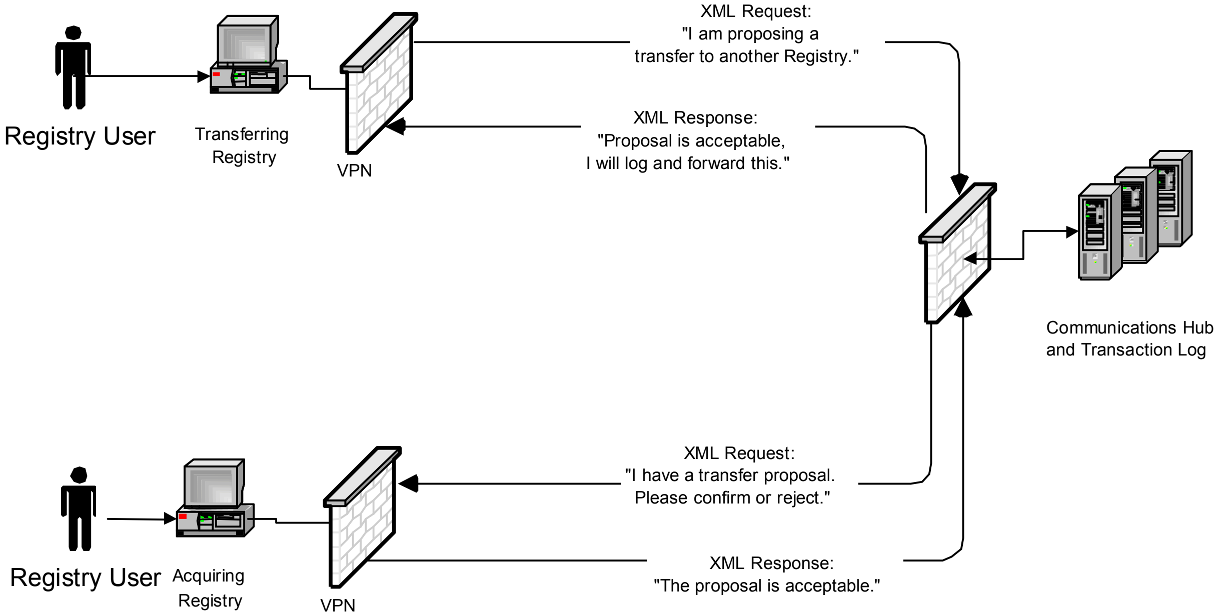
To provide this functionality, the registries and ITL shall utilize a consistent and coordinated set of technical solutions. The technical specifications require:

- Web services using Simple Object Access Protocol (SOAP);
- Hardware-based Virtual Private Network (VPN);
- XML formats adhering to the described standards in Annexes I and K;
- Digital signature authentication; and
- Network time protocols.

Each of these technologies, with the exception of the VPN requirement, is platform and language independent. As discussed below, the hardware specifications for the VPN will take into account cost, interoperability and existing registry hardware, to the extent feasible.

Figure 3.1 provides a diagram of the basic architecture of the data exchange mechanism required for communications between registries and the ITL.

Figure 3.1: Data Exchange Architecture


Registry User
Transferring Registry
VPN
XML Request: "I am proposing a transfer to another Registry."
XML Response: "Proposal is acceptable, I will log and forward this."
Communications Hub and Transaction Log
XML Request: "I have a transfer proposal. Please confirm or reject."
XML Response: "The proposal is acceptable."
Registry User
Acquiring Registry
VPN

3.2 Communications Specifications

All registries and the ITL shall use Web services to support the sending and receiving of messages. Web services enable disparate applications running on different machines to easily exchange data with one another without requiring additional proprietary third-party software or hardware. Web services depend upon a standard XML messaging systems and SOAP and therefore are not tied to any one operating system or programming language. Based on common industry standards and existing technology such as XML and HTTP, web services costs very little to deploy. Any information that is exchanged to and from both registries and the ITL shall be through the use of XML exchanged via SOAP. For technical specifications on the construct of these documents, see Annexes I and K.

SOAP is one of several an XML-based protocols for exchanging information between computers and is widely used in the internet community. Since SOAP runs primarily on top of HTTP and XML, all communications are encrypted using Secure Socket Layer (SSL).

Both the ITL and all registries shall be available for requests via the Internet.
The technical specifications for the functionality of these Web services are defined in the Web services and functions specified in Annexes B, C, and D.

3.3 Data Transfer Security

3.3.1 Virtual Private Network

All communications to and from registries and the ITL shall be protected using hardware-based virtual private network (VPN) technology. VPN technologies provide the ability to "tunnel" through the Internet from one point to another, protecting all communications. Prior to the creation of a VPN tunnel, a digital certificate is issued to a prospective client end-point, allowing the client to provide proof of identity. The client installs the certificate into their VPN end-point. The client initiates the connection and is authenticated by the VPN server. Using digital certificates, the VPN server accesses a central authority to negotiate authentication credentials. During the tunnel creation process, encryption is negotiated, ensuring that all communications through the tunnel are protected.

The ITL shall be located on an Internet-connected network protected by a hardware-based firewall. The firewall shall be configured with rules such that only "registered" clients can make connections to the VPN server. This is achieved through client registries having fixed public IP addresses and the ITL only accepting communications originating from these IP addresses. Client registries shall implement hardware-based VPN end-points for use in connecting to the system. These VPN end-points shall be configured with the appropriate credentials as provided by the ITL Administrators. The client VPN end-points shall be configured to maintain the VPN tunnel permanently, in order to allow reliable, two-way, real-time communication between the ITL and a client registry at all times.

3.3.2 Client VPN Specifications

VPN equipment at the client registries shall be dedicated devices that can reliably terminate the VPN connection to the ITL as well as maintain acceptable performance levels. The recommended VPN equipment adequate for a client registry VPN connectivity is a Cisco PIX firewall/VPN device. Information on the Cisco PIX firewall is available at http://www.cisco.com/warp/public/cc/pd/fw/sqfw5DD/.

3.3.3 IPSec VPN

In addition to the site-to-site VPN infrastructure, the use of IPSec VPN will provide for site-to-site authentication, data integrity, and data encryption. IPSec VPN configurations provide for authentication between two end-points in a VPN connection. The ITL will identify and authenticate the remote client via the IPSec connection using a digital certificate provided by a Certificate Authority.

IPSec also ensures data integrity of all communications passed through the VPN tunnel. Packets of data are hashed and signed using the authentication information established by the

VPN. Data confidentiality is also ensured by IPSec encrypting the data using Triple DES (3DES). This encryption addresses only the network traffic itself, not the application level SOAP communications.

3.3.4 SSL

SSL shall be used for all communications between the registry and the Communications Hub. SSL provides application server-to-application server authentication as well as data encryption. Since IPSec VPN provides only site-to-site authentication, a method is required to authenticate the actual registry communications to the ITL, in particular where multiple registries are hosted on a single site. Additionally, SSL protects any communications that may pass over the networks at the registry site before transport through the VPN on to the ITL.

3.4 The Communications Hub and Message Queue

The security layer and supporting hardware and software between the VPN and ITL database is the Communications Hub. The Communications Hub receives and logs all messages passed through the VPN. The Communications Hub hosts a message queue which processes all incoming messages. The purpose of the queue is to receive and store messages and to provide scalability during peak transaction times.

3.5 Data Transfer Format Specifications

All message packages must utilize XML and conform to the standards in Annex I. WSDL specifications for these XML messages are defined in Annex K.

3.6 Certificate Authority

SSL requires the use of a trusted Certificate Authority in order to realize the full benefit of positive authentication and secure encryption. Trusted Certificate Authority services are provided commercially by several vendors, such as Verisign and Thawte. These vendors verify identity and issue certificates which can be used to positively identify an organization and encrypt data communications between the organization and other certificate holders. These vendors are already widely used and trusted worldwide, with a large percentage of online transactions via SSL using their certificates.

Due to the number of registry end-points and size of the VPN, a third-party managed Certificate Authority will be used, as specified by the ITL.

3.7 User Accounts

The ITL VPN shall register and maintain user IDs and passwords for users who are logging in directly to the ITL's Web application. A user account is valid for an indefinite period of time. The ITL may revoke or replace a user's registration or password if there is a suspected breach of security or rules of behaviour by a user.

3.8 Time Validation Specifications

To ensure that transaction rules are accurately and consistently applied to all proposed transactions, the ITL and registries shall use a consistent convention for recording time, and shall also utilize time synchronization practices and procedures to ensure accurate logging and sequencing of all transactions. Accurate and consistent time clocks are essential to the reconciliation process.

All dates and times shall be recorded as Greenwich Mean Time (GMT).

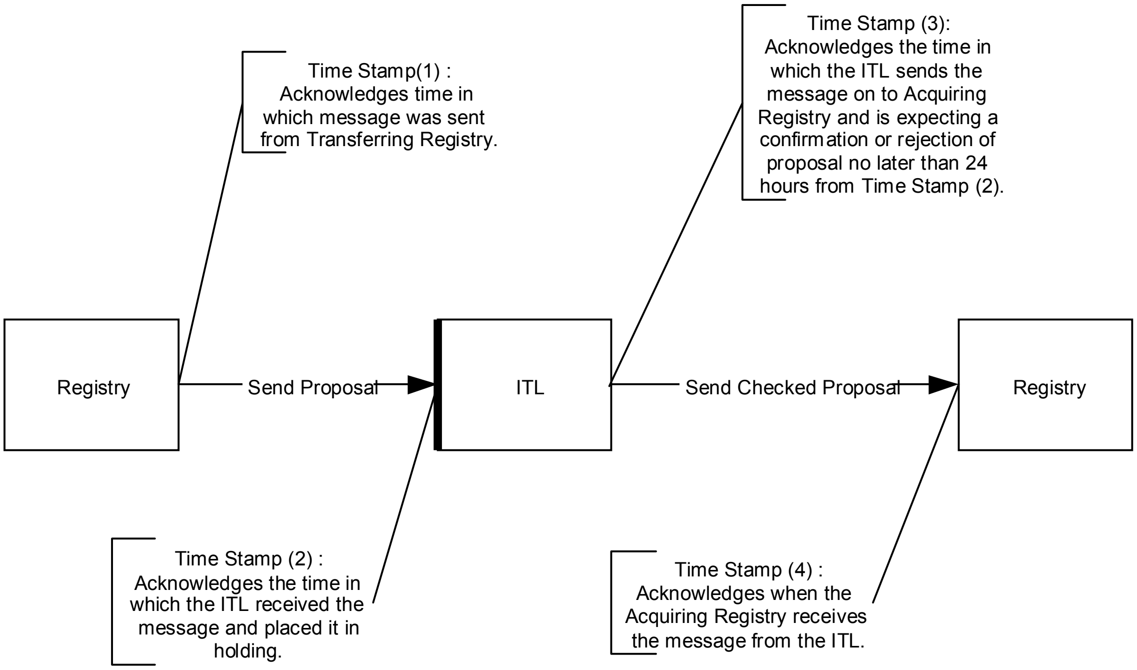
Time information shall be submitted as a date, hour, minute and second in the format:
YYYY–MM–DD HH:MN:SS

All registries and the ITL shall use Network Time Protocol (NTP) version 3 or better software to synchronize their clocks with well known public Stratum 2 time servers. NTP ensures that both the ITL and registries maintain consistent and accurate times. NTP software has been ported to all major computing platforms, and in some cases is bundled with the operating system. Information on NTP is found at http://www.ntp.org/. A list of well known international Stratum 2 NTP time servers is maintained at http://www.eecis.udel.edu/~mills/ntp/clock2a.html. A list of NTP ports to various computing platforms and other resources is found at http://www.ntp.org/links.html. Any NTP version 3 or better client is acceptable. Registries should configure their clients to contact two or more public NTP servers that are close to them in terms of traffic routing on the Internet.

As part of the ITL administrative functions described in Section 6 and Annex D, the ITL shall perform a time check service for registries in which the ITL can request the time. The ITL is responsible for recording the time in which each message is routed through its Communications Hub and enters the message queue.

Figure 3.2 outlines the sequence in which the time stamp of a message is evaluated. Sixty seconds is the recommended allowable time between Request Time Stamp (1) and Response Time Stamp (2). All Web services requests must receive an appropriate response, although this does not mean that any one or all transactions requested in the message is complete. An Acquiring Registry shall complete and send to the ITL a confirmation of the transaction as soon as possible and no later than 24 hours from Time Stamp (2). If a confirmation is not received by the ITL within 24 hours, the transaction is no longer valid and will be cancelled by the ITL.

Figure 3.2: Process Time Stamps


Time Stamp(1) : Acknowledges time in which message was sent from Transferring Registry.
Time Stamp (3): Acknowledges the time in which the ITL sends the message on to Acquiring Registry and is expecting a confirmation or rejection of proposal no later than 24 hours from Time Stamp (2).
Registry
Send Proposal
ITL
Send Checked Proposal
Registry
Time Stamp (2) : Acknowledges the time in which the ITL received the message and placed it in holding.
Time Stamp (4) : Acknowledges when the Acquiring Registry receives the message from the ITL.

3.9 Message Time-to-Live

Messages should be allowed a minimum of sixty seconds in which to respond to the requesting Web service. This time-to-live is the time it takes between the first byte of the request sent by the sender and the last byte of the response received by the recipient. In most cases, the time in which it takes to validate the digital signature, user account and password and verify that the message was a well-formed XML document should not exceed sixty seconds. A registry may elect to exceed this limit and accept messages which exceed this timeframe.

4. Unit Transaction Processes

4.1 Unit Transaction Types

This section of the Technical Specifications addresses the messages and content requirements necessary to support submissions of unit transactions by registries and the validation of those transactions by the ITL. The unit transactions either involve the transfer of ownership of a unit, a change in a attribute of a unit, or the replacement of a tCER or lCER. This section will describe the data exchange flow, the responsibilities of registries, and the responsibilities of the ITL in order to complete a unit transaction.

The following unit transactions are described:

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

4.1.1 Issuance

The issuance of AAUs is undertaken by a Party in its national registry on the basis of its assigned amount under Articles 3.7 and 3.8 of the Kyoto Protocol (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 tCERs, lCERs, and CERs into a pending account is undertaken by the CDM Executive Board, in the CDM Registry, on the basis of verified and certified reductions in greenhouse gas emissions or removals of greenhouse gases from the atmosphere through a CDM Project activity. Issuance of such units is monitored and validated by the ITL.

The issuance process for AAUs, RMUs, tCERs, lCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.2 Conversion

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

The conversion process of AAUs and RMUs to ERUs follows the single registry model of data exchange described in Section 4.3.

4.1.3 External Transfer

The external transfer of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs 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 external transfer of such units is monitored and validated by the ITL.

The external transfer process for AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the two registry model of data exchange described in Section 4.4.

4.1.4 Cancellation

The internal transfer of AAUs, RMUs, ERUs, CERs, tCERs and lCERs to a cancellation account is undertaken by a Party, an entity or the CDM Executive Board, on the basis of the amounts proposed by the transferor. The cancellation of such units is monitored and validated by the ITL.

Although the ITL will notify the registry about units which must be carried over or cancelled at the end of a Commitment Period, it is not necessary to include the Notification ID received from the ITL for subsequent cancellation transactions.

However, when the ITL notifies a registry regarding excess tCER or lCER issuance requiring cancellation of a portion of these units, the Notification ID must be submitted in the Cancellation transaction.

The cancellation of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.5 Replacement

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

For replacements required by a Reversal of Storage action or a Non-submission of Certification action by the CDM Executive Board, the registry must include the Notification ID associated with the replacement transaction. This Notification ID is used to determine if the replacement should be considered when the ITL performs a follow-up evaluation to assess whether the required replacement has been completed.

The replacement of tCERs and lCERs follows the single registry model of data exchange described in Section 4.3.

4.1.6 Retirement

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

The retirement of AAUs, RMUs, ERUs, tCERs, lCERs, and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.7 Carry-over Process

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 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 must be cancelled. The carry-over of units is monitored and validated by the ITL.

At the conclusion of the true-up period for a Commitment Period, the ITL will send notifications to a registry indicating the units which have not been carried over and the limits on carry-overs to the new Commitment Period. There will be a separate notification for each unit type. All carry-over transactions must contain the Notification ID sent by the ITL and must only contain units of the type specified by that Notification ID.

The carry-over of AAUs, ERUs and CERs follows the single registry model of data exchange described in Section 4.3.

4.1.8 Expiry Date Change

The change in the expiry date is undertaken by a Party for tCERs and ICERs. For tCERs, this transaction may be necessary where they are initially issued with an expiry date which is different from that eventually agreed as the end of the subsequent Commitment Period. For ICERs, this transaction will occur when the Executive Board approves the renewal of the crediting period for a Project. The ITL ensures that these expiry date changes are consistent with the appropriate dates and updates the tCER and ICER expiry dates in the ITL database.

The expiry date change transaction follows the single registry model of data exchange described in Section 4.3.

4.1.9 Internal Transfers and Other Transactions Routed to an STL

For transactions routed to an STL, the ITL conducts general transaction checks necessary to mark the unit blocks as unavailable due to a pending transaction and splits the 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.

Supplemental transactions follow either the single registry model or the multiple registry model. If an STL wishes to institute a supplemental transaction to be routed through the ITL, the STL must coordinate the development of this transaction with the ITL Administrator.

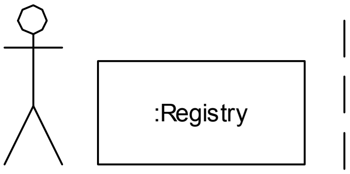
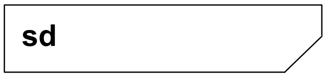
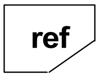
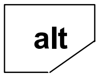
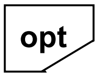
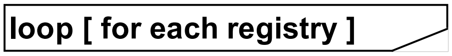
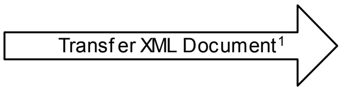
4.2 Description of Data Exchange Flow

The data exchange flow for each unit transaction type follows one of two models: the single registry model or the multiple registry model. Most of the transaction types follow the single registry model. External transactions, and some supplemental transactions not described in this document, follow the multiple registry model. This single registry model is described first, followed by the data exchange model for multiple registries engaged in an external transaction. The sections for both models contain the following subsections:

- UML Behaviour Diagram; and
- Stage Table.

The UML Behaviour Diagram is a high level representation of processes which is designed to capture the participants in each process and the order in which the components are used. Within the diagram, specific technical "components" (representing specific programming logic) are defined. The Behaviour Diagram is based on the standards for the Unified Modeling Language (UML), with text annotations to help non-technical readers interpret it more easily. These diagrams include the following symbols and conventions.

Figure 4.1: Key to UML Diagram

UML Element	Description
Actors & swim lanes  	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 "swim lane" which is directly underneath the participant's icon or box, and represented by a dashed vertical line.
 	This symbol indicates that the diagram is a sequence diagram. The symbol is followed by the name of the process.
 	This symbol indicates that there is a secondary sub-diagram for the component which provides additional detail of the functionality.
 	This symbol indicates that the process supports alternative outcomes in the prior step. Within an alternative, there may be a second alternative scenario, equivalent to programs which contain nested "if…then" statements. In the issuance process, for example, the issuance is either accepted (Result = Success) or a discrepancy is identified (Result = Failed). If successful, the registry can either confirm the issuance or terminate the issuance.
 	This symbol indicates that the process within the box will only be executed if a certain condition is met.
 	This symbol indicates that the process in the box is repeated a number of times. For example in the Time Synchronization process, the processes within the box are executed once for each registry that interacts with the ITL.
 	This symbol represents a message containing an XML document, its transfer and the "acknowledgement" of its receipt. The message is "sent" from one component and "received" by another component, as indicated in footnote (x).
Boxes with dotted outlines	These boxes represent a component or area of functionality necessary to the process, but which does not have specifically defined input or output parameters used for messaging. These components could be defined and implemented by developers in many different ways.
Boxes with solid outlines	These boxes represent a component which performs a specific task necessary to the process. These components either receive or produce the information which is used for messaging.

The Stage Table represents the sequence of events in terms of the "stage" and its relation to the "status" of a transaction or reconciliation action. The stage of a transaction defines where in the process of information exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and received by either a registry or the ITL or when the last step of a process occurs. The order in which each defined stage occurs may vary based on the specific process and based on the results of the ITL validation process. The numbers assigned to stages should not be used as an indicator of acceptable stage sequences.

Figure 4.2: Key to Stages

Processes	Stage Code	Stage	Description
Issuance, Conversion, External Transfers, Internal Transfers, Carry-over, Expiry Date Change	P	Proposed	Proposal issued from Transferring Registry.
	TR	ITL Review	Proposal evaluated at ITL.
	RR	Registry Review	Proposal evaluated at Acquiring Registry.
	RA	Registry Accepted	Acquiring Registry has accepted transaction.
	TA	ITL Accepted	ITL has received the evaluation result (accepted or terminated) from the Acquiring Registry.
	RC	Registry Complete	Registry has completed the transaction.
	TC	ITL Complete	ITL has completed the transaction.

For transaction processes, the stage codes are not submitted in the XML message and are only represented in this table for clarity.

These technical specifications describe in general terms the programming logic that should be implemented at the registries and the ITL to establish reliable communications. A list of functions needed to implement this specification is included for each model. Technical information for transaction functions, including required inputs, outputs, and responses, is included in Annex B.

The results of a transaction evaluation conducted by the ITL or an Acquiring Registry are returned in the XML document in the form of Response codes. Response codes and corresponding checks are grouped by the category of check can be found in Annex E.

4.3 Single Registry Model

The single registry model for transactions applies to the following transaction types:

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

The following steps apply to all the above transactions and describe the sequence of messages necessary to complete the transaction. This description assumes that the transaction is not sent to an STL.

Step 1 – Proposal

The registry sends a proposal for a transaction to the ITL using the AcceptProposal Web service method on the ITL. The proposal contains the transaction type, the units involved in the transaction, the appropriate transferring and acquiring account information, and the appropriate Notification ID information.

Step 2 – ITL Review

The Communications Hub receives the proposal and, once the incoming message is verified to be well formed and authentic, it places the message in a queue for processing. Messages are processed from the queue in the order received. The ITL validates the transaction against the business rules for the appropriate transaction type. If a discrepancy is found, the ITL notifies the registry of the requirement(s) the transaction proposal did not meet. The units involved in the transaction cannot be used in another transaction until the registry sends a termination request.

If the transaction meets all requirements, the ITL records the transaction as pending and marks the units involved in the transaction as unavailable to any other transaction.

The ITL sends the results of the validation, whether a discrepancy was found or not, to the registry via the AcceptNotification Web service method that registries are required to implement. If the ITL identified one or more discrepancies, response codes will be included in the message to indicate the type of discrepancy found.

Step 3 – Registry Complete/Registry Terminate

Once the registry processes the ITL notification it must complete the transaction, either by finalizing it (if no discrepancy was found) or by terminating it. The registry calls the AcceptNotfication Web service method on the Communications Hub to complete the transaction.

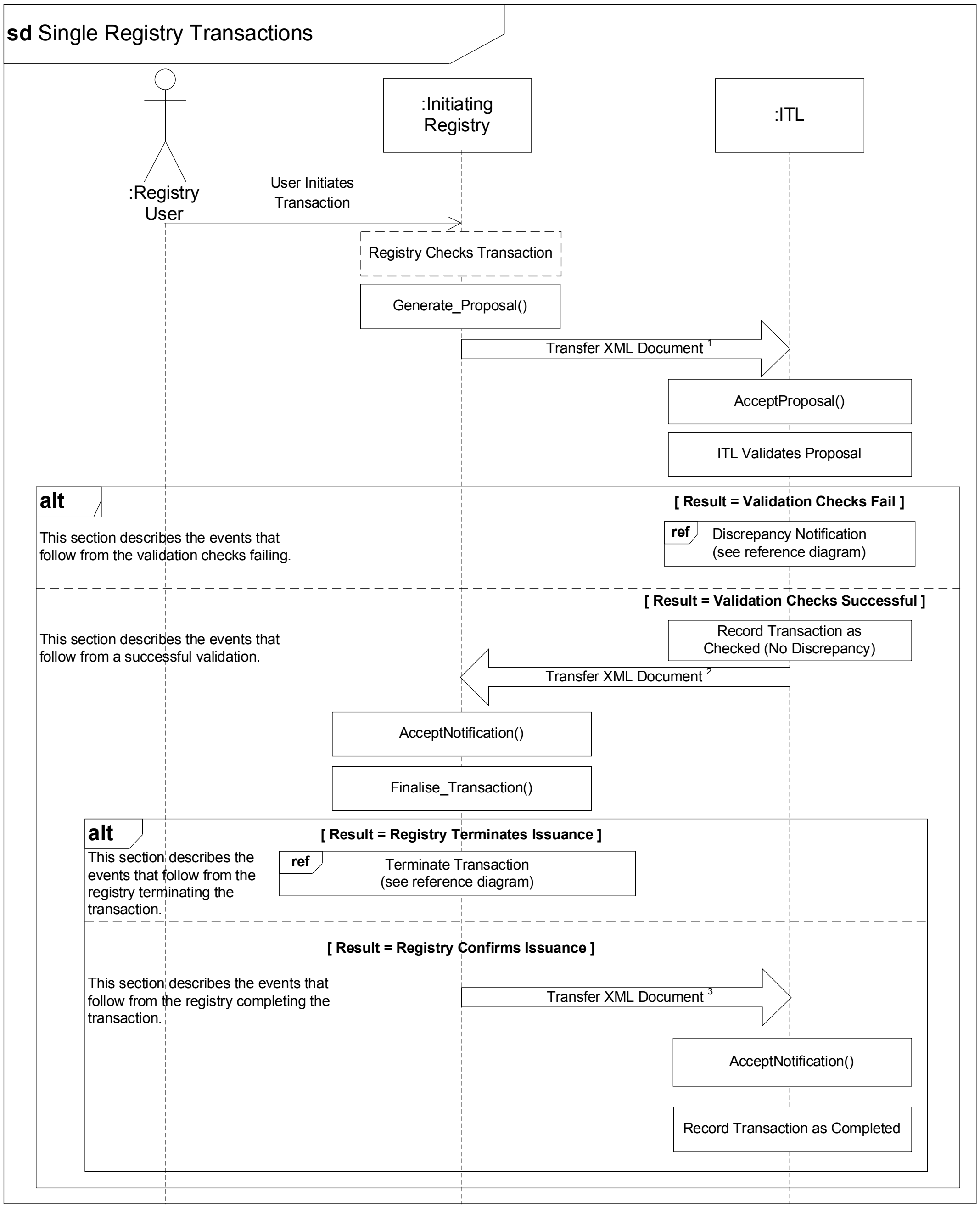
Step 4 – ITL Complete

If the registry requested the ITL to finalize the transaction, the ITL updates its records for the units in the transaction as appropriate for the transaction type. The units are now free to be used in any other transaction. If the registry requested the ITL to terminate the transaction, the ITL will mark the transaction as terminated. The units that had been part of the transaction are now free to be used in another transaction.

The completed transaction has now been logged by the ITL.

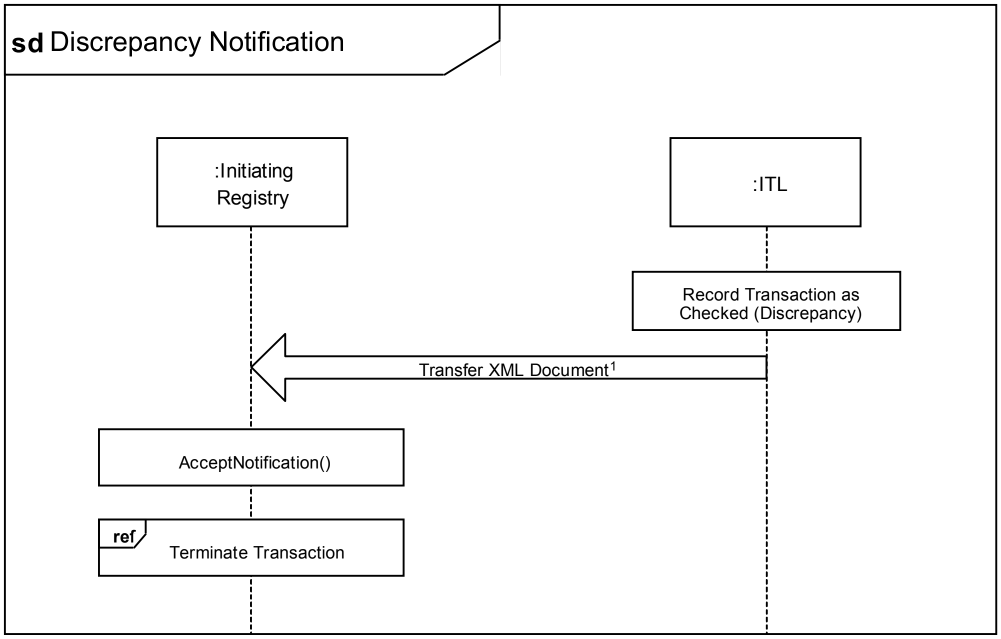
4.3.1 Single Registry Behaviour Diagrams

Figure 4.3: Single Registry Transaction Behaviour Diagram


sd Single Registry Transactions
:Registry User
:Initiating Registry
:ITL
User Initiates Transaction
Registry Checks Transaction
Generate_Proposal()
Transfer XML Document 1
AcceptProposal()
ITL Validates Proposal
alt
[Result = Validation Checks Fail]
This section describes the events that follow from the validation checks failing.
ref
Discrepancy Notification (see reference diagram)
[Result = Validation Checks Successful]
This section describes the events that follow from a successful validation.
Record Transaction as Checked (No Discrepancy)
Transfer XML Document 2
AcceptNotification()
Finalise_Transaction()
alt
[Result = Registry Terminates Issuance]
This section describes the events that follow from the registry terminating the transaction.
ref
Terminate Transaction (see reference diagram)
[Result = Registry Confirms Issuance]
This section describes the events that follow from the registry completing the transaction.
Transfer XML Document 3
AcceptNotification()
Record Transaction as Completed

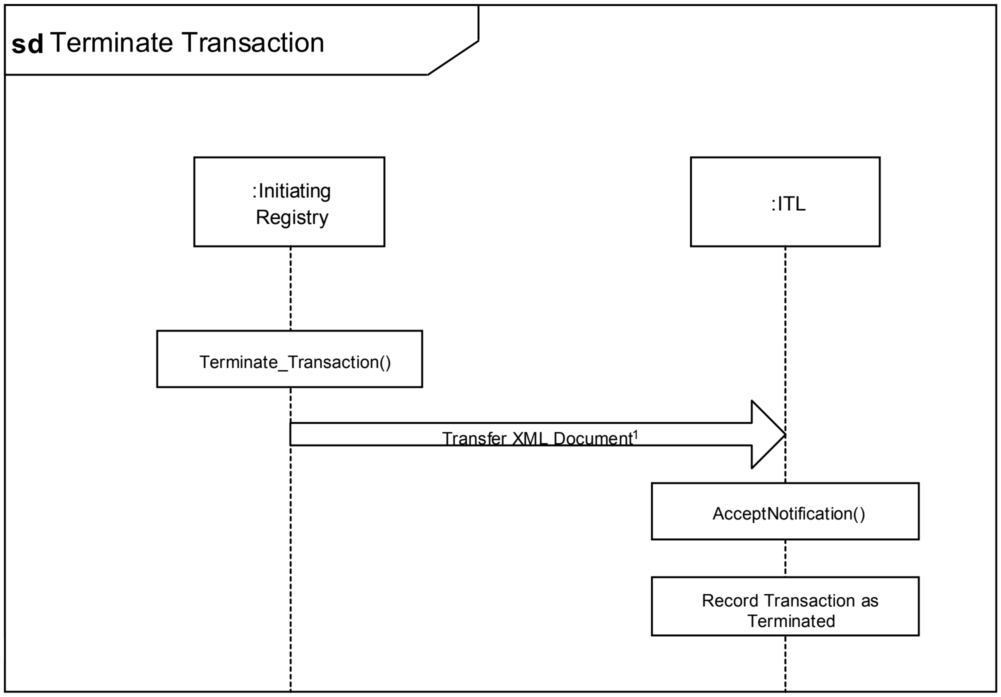
1. Function Generate_Proposal creates an XML document that proposes a transaction and sends the document to the AcceptProposal Web service on the ITL.
2. The ITL creates an XML document to inform the registry that the transaction was successfully validated and sends the document to the AcceptNotification Web service on the registry.
3. Function Finalise_Transaction creates an XML document to inform the ITL that the transaction is complete and sends the document to the AcceptNotification Web service.

Figure 4.4: Discrepancy Notification Sequence Diagram



1. In order to inform the Initiating Registry of a discrepancy the ITL prepares and sends an XML document to the AcceptNotification Web service on the registry.

Figure 4.5: Terminate Transaction Sequence Diagram



1. The Terminate_Transaction function on the registry creates and sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL that the transaction was terminated.

4.3.2 Single Registry Transactions Stage Table

Figure 4.6 describes each stage of the process for single registry transactions. For example, the first row of the table should be read as follows: When the stage is "Proposed," the stage ended with "Message 1" containing the transaction status of "Proposed." The message is sent to the "ITL" and generated by the "registry."

Figure 4.6: Single Registry Stage Table

Stage	Stage Name	Stage Ends With	Transaction Status	Sent To	Generated By
P	Proposed	Message 1	Proposed	ITL	Registry
TR	ITL Review	Message 2	Checked (Discrepancy) or Checked (No Discrepancy)	Registry	ITL
RC	Registry Complete	Message 3	Completed or Terminated	ITL	Registry
TC	ITL Complete	No Message	Completed or Terminated	--	--

4.4 Two Registry Transaction Model

The two registry model for transactions applies to External Transactions in which units are transferred to a different registry. The initial steps are similar to the single registry model, but require the additional step of forwarding the proposal to the Acquiring Registry. The following steps describe the sequence of messages necessary to complete an external transfer:

Step 1 – Proposal

The registry sends a proposal for a transaction to the ITL using the AcceptProposal Web service method on the Communications Hub. The proposal contains the transaction type, the units involved in the transaction, the transferring and acquiring account types, and the transferring and acquiring account identifiers.

Step 2 – ITL Review

The Communications Hub receives the proposal and, once the incoming message is verified to be well formed and authentic, it places the message in a queue for processing. Messages are processed from the queue in the order received. The ITL validates the transaction against the business rules for external transactions. If the transaction meets all requirements, the ITL records the transaction as pending and marks the units involved in the transaction as unavailable to any other transaction. It then forwards the proposal to the Acquiring Registry.

If a discrepancy is found, the ITL will notify the Initiating Registry of the requirement(s) the transaction proposal did not meet. The ITL will also notify the Acquiring Registry that the transaction was not completed. The units involved in the transaction cannot be used in another transaction until the Initiating Registry sends a termination request.

Step 3 – Registry Review

The ITL forwards the transaction proposal to the Acquiring Registry by calling the AcceptProposal Web service method on the Acquiring Registry. The Acquiring Registry evaluates the proposal and either accepts or rejects it. In either case, the Acquiring Registry calls the AcceptNotification Web service method on the ITL to inform the ITL of its evaluation result.

Step 4 – ITL Relay

The ITL updates the transaction status with the result of the Acquiring Registry evaluation and notification and forwards the evaluation result to the Initiating Registry. The ITL calls the AcceptNotification Web service method on the Initiating Registry.

Step 5 – Registry Complete

The registry completes the transaction. The registry calls the AcceptNotification Web service method on the ITL to inform the ITL when it has finished updating its records.

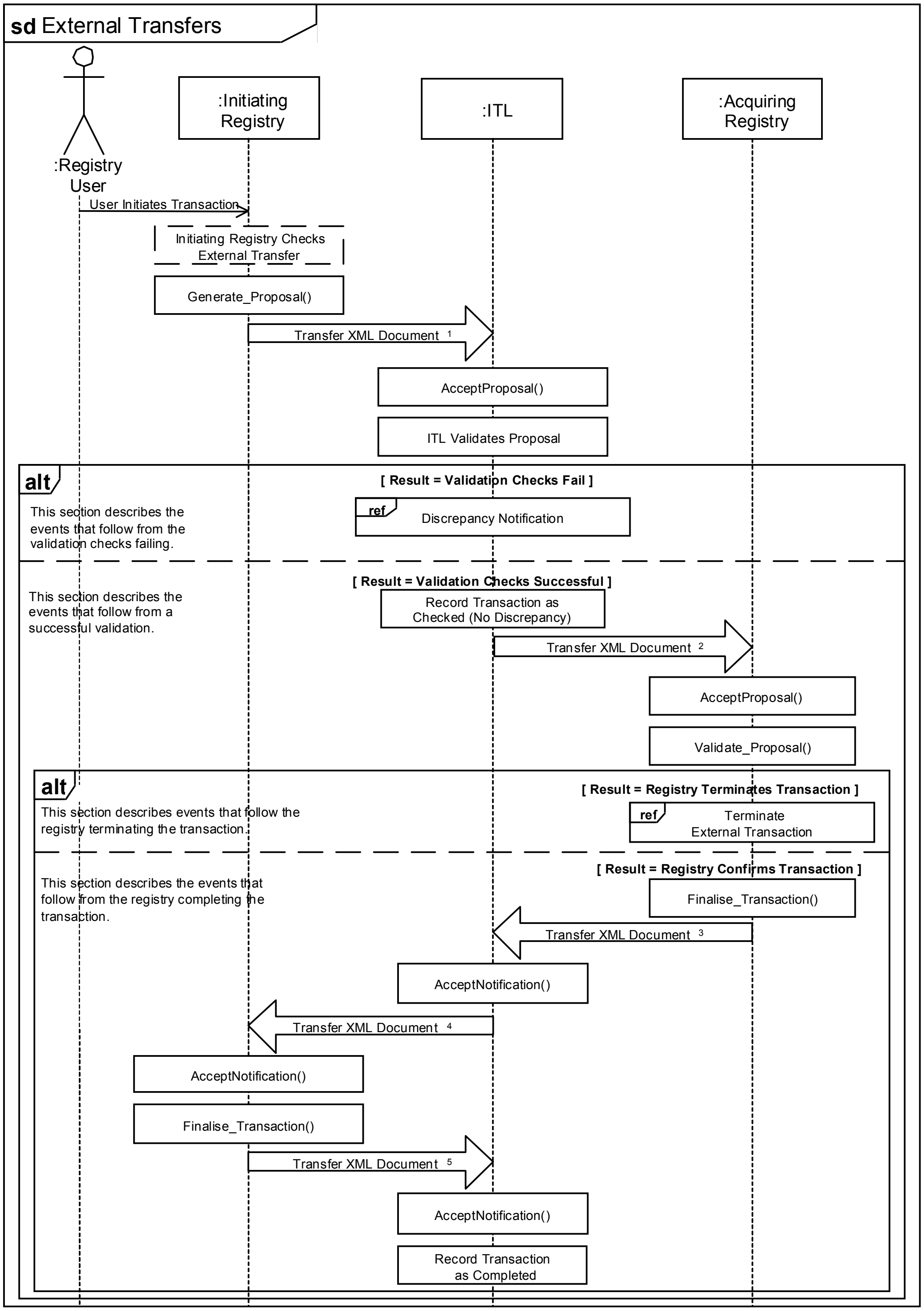
Step 6 – ITL Complete

The ITL completes the transaction. The ITL updates its records for the units in the transaction. The units that had been part of the transaction are now available to be used in another transaction.

The completed transaction has now been logged by the ITL.

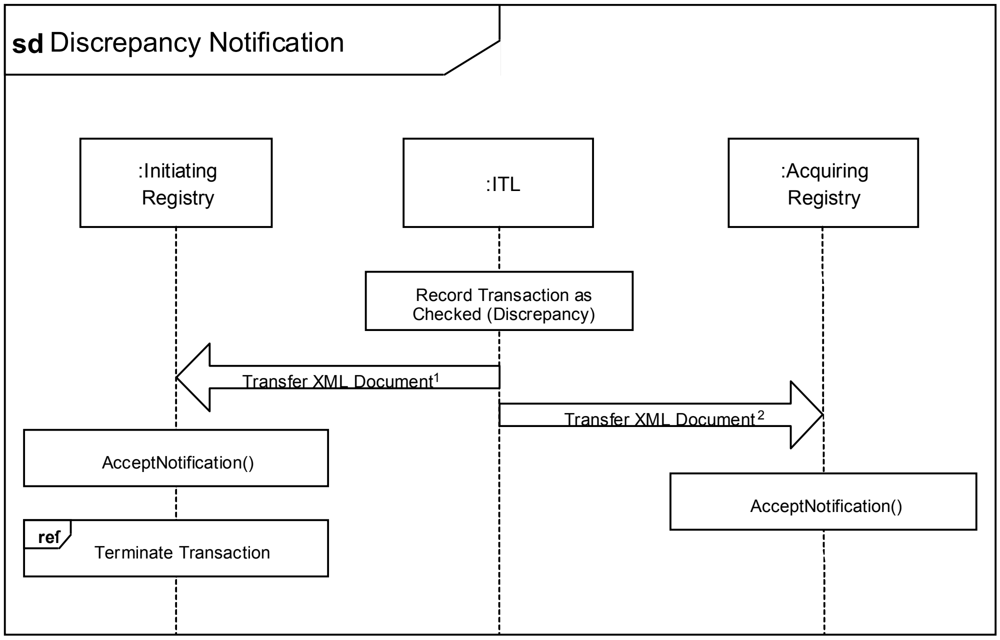
4.4.1 UML Behaviour Diagram for Two Registry Transactions

Figure 4.7: External Transfer Behaviour Diagram


sd External Transfers
:Registry User
:Initiating Registry
:ITL
:Acquiring Registry
User Initiates Transaction
Initiating Registry Checks External Transfer
Generate_Proposal()
Transfer XML Document 1
AcceptProposal()
ITL Validates Proposal
alt
[Result = Validation Checks Fail]
This section describes the events that follow from the validation checks failing.
ref
Discrepancy Notification
[Result = Validation Checks Successful]
This section describes the events that follow from a successful validation.
Record Transaction as Checked (No Discrepancy)
Transfer XML Document 2
AcceptProposal()
Validate_Proposal()
alt
[Result = Registry Terminates Transaction]
This section describes events that follow the registry terminating the transaction.
ref
Terminate External Transaction
[Result = Registry Confirms Transaction]
This section describes the events that follow from the registry completing the transaction.
Finalise_Transaction()
Transfer XML Document 3
AcceptNotification()
Transfer XML Document 4
AcceptNotification()
Finalise_Transaction()
Transfer XML Document 5
AcceptNotification()
Record Transaction as Completed

1. Function Generate_Proposal creates an XML document that proposes a transfer and sends the document to the AcceptProposal Web service on the ITL.
2. The ITL creates an XML document to pass the proposed transfer on to the AcceptProposal Web service on the Acquiring Registry.
3. Function Finalise_Transaction on the Acquiring Registry creates an XML document to inform AcceptNotification on the ITL that transfer was confirmed.
4. AcceptNotification on the ITL creates an XML document to inform the AcceptNotification Web service on the Initiating Registry that the Acquiring Registry confirmed the transfer.
5. Function Finalise_Transaction on the Initiating Registry creates an XML document to inform the ITL through the AcceptNotification Web service that transfer was confirmed.

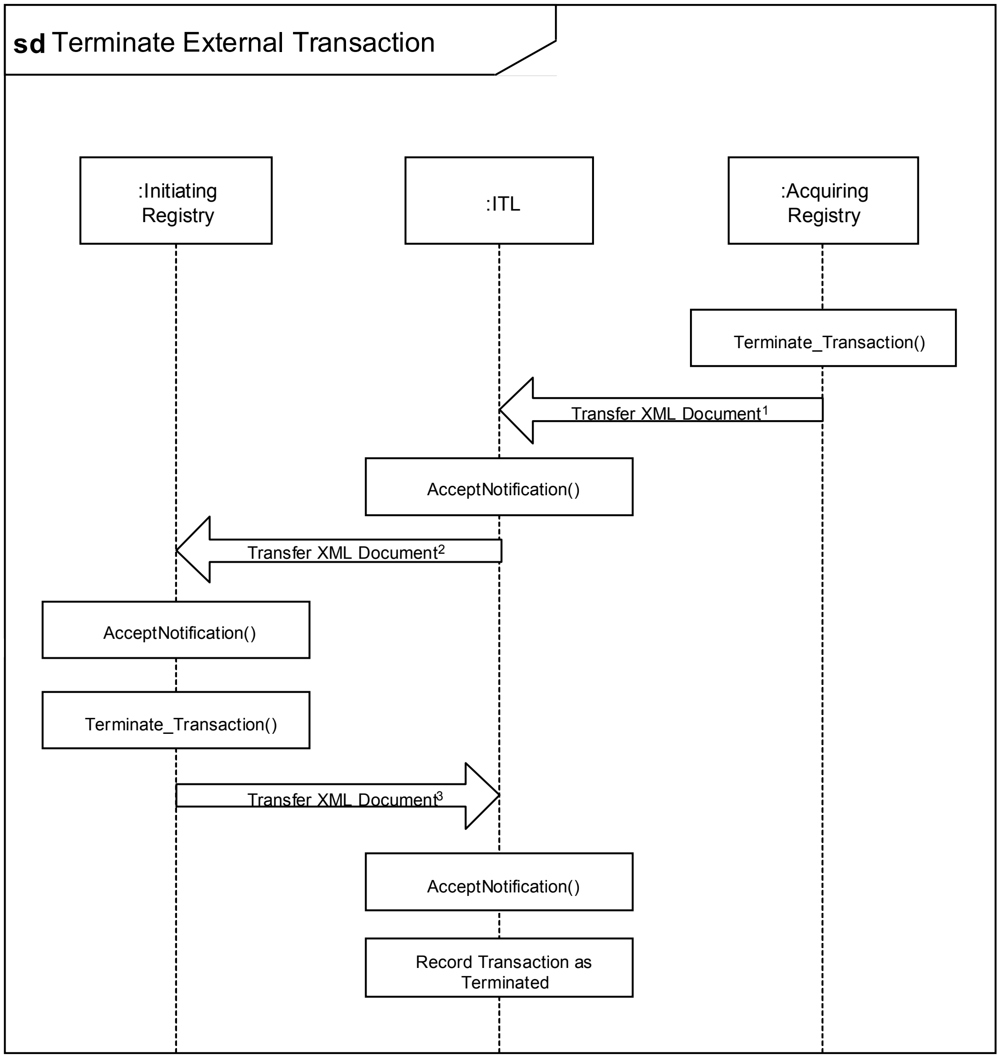
Figure 4.8: Discrepancy Notification Sequence Diagram



1. In order to inform the Initiating Registry of a discrepancy the ITL prepares and sends an XML document to the AcceptNotification Web service on the registry.

2. In order to inform the Acquiring Registry of a discrepancy, the ITL prepares and sends an XML document to the AcceptNotification Web service on the Acquiring Registry.

Figure 4.9: Terminate External Transaction Sequence Diagram



1. The Terminate_Transaction function on the Acquiring Registry sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL of the terminated transaction.

2. The ITL sends an XML document to the AcceptNotification Web service on the Initiating Registry to inform the registry that that Acquiring Registry terminated the transaction.

3. The Terminate_Transaction function on the Initiating Registry creates and sends an XML document to the AcceptNotification Web service on the ITL to inform the ITL that the Initiating Registry terminated the transaction.

4.4.2 Stage Table for Two Registry Transactions

The stage table for Two Registry Transactions (such as External Transfer) contains three different scenarios. The stages for scenario three do not include the Acquiring Registry.

This table describes each stage of the process. For example, the first row of the table should be read as follows: When the stage is "Proposed," the stage ended with "Message 1" containing the transaction status of "Proposed." The message is sent to the "ITL" and generated by the "Transferring Registry."

Figure 4.10: External Transfer Stage Table

Scenario #1	Stage	Stage Name	Stage Ends With	Transaction Status	Sent To	Generated By
Checks (No Discrepancy) Acquiring Registry Accepts	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (No Discrepancy)	Acquiring Registry	ITL
	RR	Registry Review	Message 3	Accepted	ITL	Acquiring Registry
	TA	ITL Accepted	Message 4	Accepted	Transferring Registry	ITL
	RC	Registry Complete	Message 5	Completed	ITL	Transferring Registry
	TC	ITL Complete	No message	Completed	--	--
Scenario #2	**Stage**	**Stage Name**	**Stage Ends With**	**Transaction Status**	**Sent To**	**Generated By**
Checks (No Discrepancy) Acquiring Registry Terminates	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (No Discrepancy)	Acquiring Registry	ITL
	RR	Registry Review	Message 3	Rejected	ITL	Acquiring Registry
	TA	ITL Accepted	Message 4	Rejected	Transferring Registry	ITL
	RC	Registry Complete	Message 5	Terminated	ITL	Transferring Registry
	TC	ITL Complete	No message	Terminated	--	--
Scenario #3	**Stage**	**Stage Name**	**Stage Ends With**	**Transaction Status**	**Sent To**	**Generated By**
Checks (Discrepancy)	**P**	Proposal	Message 1	Proposed	ITL	Transferring Registry
	TR	ITL Review	Message 2	Checked (Discrepancy)	Transferring Registry	ITL
	RC	Registry Complete	Message 3	Terminated	ITL	Transferring Registry
	TC	ITL Complete	No message	Terminated	--	--

4.5 List of Functions for Transaction Data Exchange

4.5.1 Registry Web Services and Functions

In order to participate in data exchange with the ITL, registries must precisely implement Web services that the ITL can use to send it information. The following table shows the Web services methods registries are required to expose for unit transactions. Detailed technical information about the specifications for these Web service methods are in Annex B.

Figure 4.11: Registry Public Web Service Methods

Public Web Service Method	Page
AcceptNotification	B-5
AcceptProposal	B-6

In addition to the above Web service methods that the registry must precisely implement so that they may be used by the ITL, the registry must have capabilities to build transactions, validate transactions, and log transactions. The following functions implement those responsibilities. Note that these functions are not exposed to the public, so they provide more flexibility in how they are implemented.

Figure 4.12: Registry Internal Functions

Private Function	Page
Check_Version	B-7
Data_Integrity_Checks	B-8
Finalise_Transaction	B-9
Generate_Proposal	B-10
Preliminary_Checks	B-11
Update_Units	B-12
Validate_Proposal	B-13
Write_To_File	B-14
Write_To_Message_Log	B-15
Write_Transaction	B-16
Write_Transaction_Block	B-17
Write_Transaction_Status	B-18

4.5.2 ITL Web Services and Functions

Like the registries, the ITL must precisely implement public Web services called AcceptNotification and AcceptProposal to be used by registries in data exchange. Detailed technical information about the specifications for these Web service methods are in Annex E to the ITL Technical Specifications and Annexes I and K to this document.

The ITL also contains extensive functionality for checking and logging data. Detailed technical information about the specifications for these Web service methods are in Annex F of the ITL Technical Specifications and in Annex K to this document.

4.6 Transaction Checks

The ITL executes numerous checks on all transactions to assure the authenticity of a message, the format of the message, the sequence of the message, and the validity of the unit transaction. The following categories of checks are performed on the ITL. The list of specific checks and associated response codes is included in Annex E.

It is recommended that registries implement similar checks to reduce the number of discrepancies identified by the ITL.

4.6.1 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. 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 errors. Failures due to transaction checks are returned in the ResponseObject in an HTTP SOAP response initiated by the ITL to the Originating Registry.

4.6.2 Message Viability Checks

Messages are placed in one of three different queues and are processed on a first-come-first-served basis. The time in which the message is added into the queue becomes the official timestamp in which the ITL acknowledges receipt of the message. However, should the ITL database be unavailable for an extended period of time due to hardware failure, messages remain in the queue until such time in which they can be processed. These checks determine whether the message from the queue is still viable and can be processed.

4.6.3 Registry Validation 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.

4.6.4 Data Integrity Checks for Transactions

This category of checks is performed by the ITL's Data_Integrity_Checks function to identify whether incoming messages contain data that do not meet 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 ITL's Transaction Log table and is not processed further. All data integrity checks are critical checks; if they result in failure, no further checks are processed.

4.6.5 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.

4.6.6 General Transaction Checks

The ITL performs this category of checks for all transaction messages involving unit blocks.

4.6.7 Transaction-specific Checks

The ITL performs this category of checks on all Kyoto transactions for the specified transaction types. The checks include checks performed by the ITL to verify limitations on registry transactions. These include maintaining and monitoring the Commitment Period Reserve and limits on issuance, both by national registries and the CDM Registry.

5. Reconciliation Process

5.1 Reconciliation Process Flow

The data on unit holdings in registries and the ITL are reconciled on a periodic basis on the basis of a data snapshot at a specified time. The snapshot taken must treat proposed transactions (in any status prior to "Completed") as if the transaction had not yet occurred. All unit type and account types for unit blocks held by the registry must be totalled for purposes of reconciliation as if they had not been changed or transferred by any ongoing transactions. This approach is necessary to ensure consistency of totals and unit blocks with the ITL, which will not commit changes in ownership or unit block types (or other attributes) until the message with the transaction status of "Completed" is received from the Initiating Registry.

It is recommended that registries delay committing database transactions for proposed transactions and sending the messages for "Completed" transactions for a short period of time surrounding the reconciliation snapshot date and time. The amount of time recommended will vary based on message processing time and is within the discretion of the Registry Manager. The ITL will not change the processing of messages to avoid possible inconsistency. A prolonged period of registry non-operation or suspension of transactions for reconciliation purposes is not foreseen.

A reconciliation action is completed when no inconsistencies are discovered or when any discovered inconsistencies have been resolved. The reconciliation process is implemented in phases in which different types of data are requested:

- Confirm Reconciliation
- Phase 1 – Validate Totals
- Phase 2 – Validate Unit Blocks
- Phase 3 – Review Audit Logs

Procedures for the use of this reconciliation process are to be agreed among the administrators of the ITL and registries. These procedures will address, for example, the scheduling of reconciliation phases and approaches to manual intervention to correct inconsistencies. They will include the possibility of directly initiating later reconciliation phases without first passing the earlier phases, or choosing to continue with the later phases even where the earlier phases did not identify any inconsistencies.

Confirm Reconciliation

1. Prior to the agreed upon snapshot time, the ITL Administrator opens a reconciliation action for the applicable registry and records the status as "Confirmed" (0).

2. The ITL calls the InitiateReconciliation Web service on the registry. This message contains the designated snapshot time and acts as confirmation of the snapshot date and time previously agreed to by the registry.

3. If the registry is part of a supplementary program, the ITL also calls the InitiateReconciliation Web service on the STL. This message contains the designated snapshot time previously agreed upon.

4. At the designated time the ITL and the registry create a snapshot upon which the following data analysis relies.

Phase 1 – Validate Totals

1. Upon completion of its snapshot, the ITL requests unit holding totals by account type, Commitment Period and unit type from the registry. To do so the ITL calls the ProvideTotals Web service method on the registry.

2. The registry receives the request from the ITL, compiles the totals, and sends the totals to the ITL by calling the ReceiveTotals Web service on the ITL.

3. The ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, reconciliation data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed, the ITL compares the totals sent by the registry with its own records and determines a result.

5. The ITL records the new status of the reconciliation action. If the status is "Validated" (2), the ITL checks if the Party is in a supplementary program. If it is, the ITL initiates STL Reconciliation processing. If the Party is not in a supplementary program, the ITL sends notification to the registry that the reconciliation completed successfully. The ITL also removes the freeze flag from any units that remain flagged from a previous reconciliation action at that registry. If the new status is "Inconsistent Totals" (3), the ITL requests the registry to send unit block details.

Phase 2 – Validate Unit Blocks

1. The ITL calls the ProvideUnitBlocks Web service method on the registry to request that the registry send unit blocks. This request may be limited to unit blocks for a specific unit type, account type, Commitment Period combination that failed the totals check.

2. The registry sends its unit block inventory to the ITL by calling the ReceiveUnitBlocks Web service method.

3. ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, reconciliation data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed, the ITL compares each unit block sent by the registry against the ITL records. If blocks do not match, they are marked as inconsistent.

5. If no inconsistent blocks are found and Phase 2 is not the starting phase, a manual intervention is triggered to explain why the totals check in the first phase of reconciliation failed. If inconsistent blocks are found, the ITL requests the registry to send a transaction history since the last reconciliation for each inconsistent block.

Phase 3 – Review Audit Logs

1. The ITL calls the ProvideAuditTrail Web service method on the registry to request the transaction history for each inconsistent block.

2. The registry sends the transaction history to the ITL by calling the ReceiveAuditTrail Web service method.

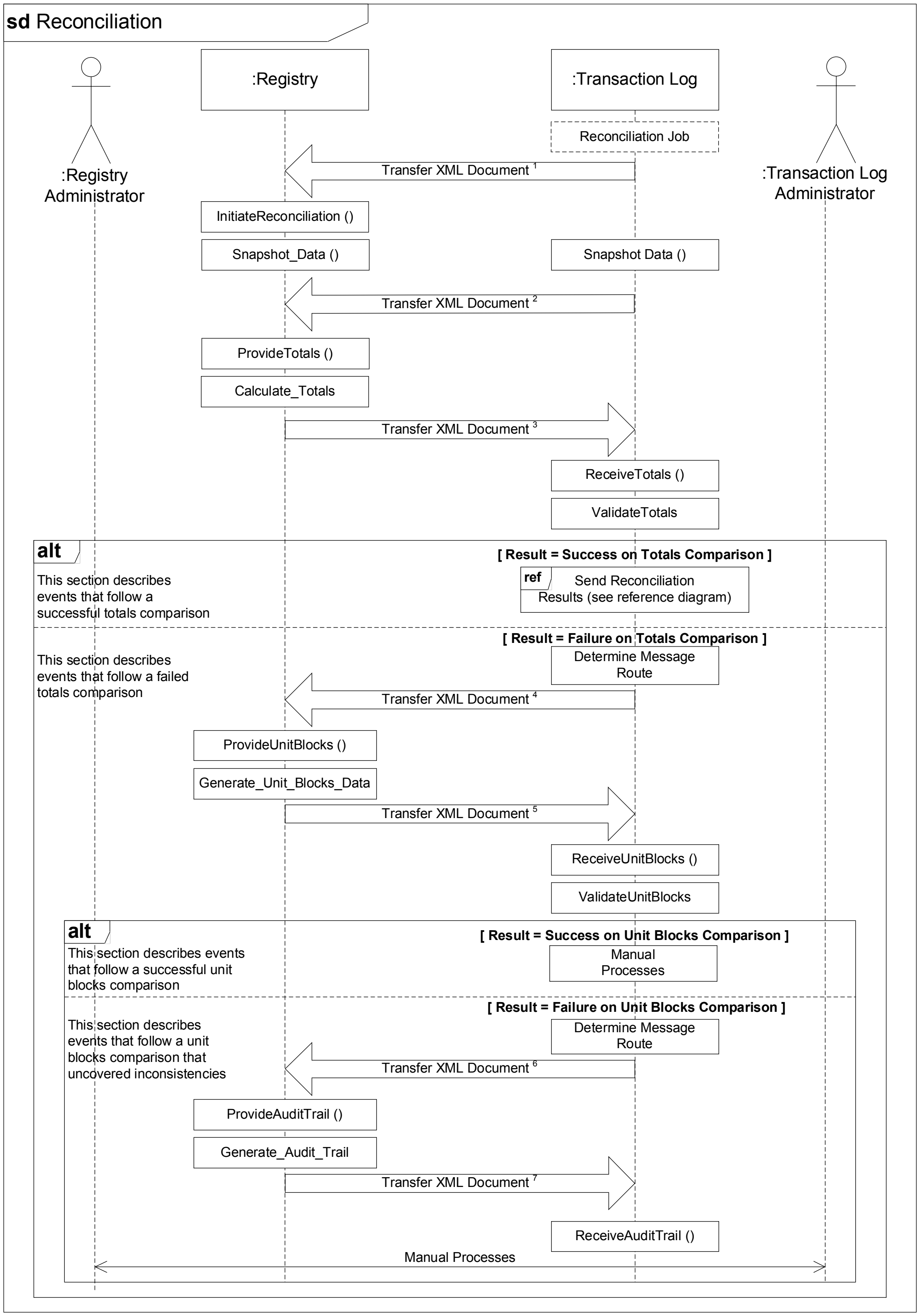
3. The ITL performs preliminary checks on the message and adds the message to the processing queue. When the ITL removes the message from the queue it performs registry validation, reconciliation data integrity and message sequence checks. Failure results in rejection of message without data recording or further processing.

4. If all the checks are passed the ITL writes the transaction history to a flat file and stores the location in the ITL's Message Log table.

5. The ITL and Registry Administrators research and correct the cause of the inconsistency through manual intervention. Once corrected, a new reconciliation is immediately initiated by the ITL Administrator.

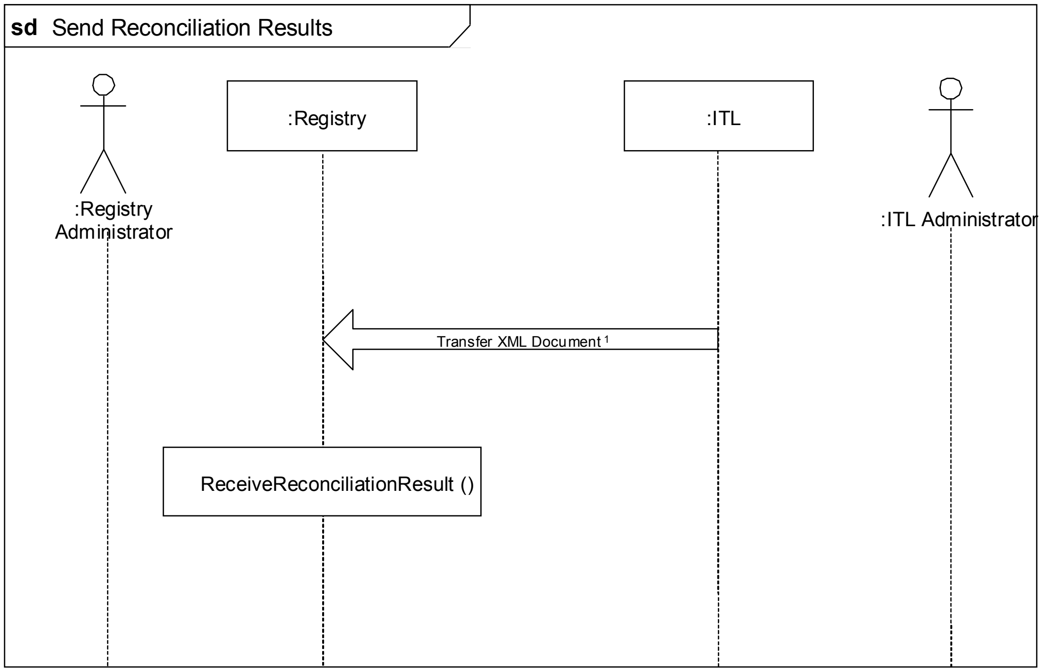
5.2 Reconciliation Behaviour Diagrams

Figure 5.1: Reconciliation Behaviour Diagram


sd Reconciliation
:Registry
:Transaction Log
:Registry Administrator
:Transaction Log Administrator
Reconciliation Job
Transfer XML Document 1
InitiateReconciliation ()
Snapshot_Data ()
Snapshot Data ()
Transfer XML Document 2
ProvideTotals ()
Calculate_Totals
Transfer XML Document 3
ReceiveTotals ()
ValidateTotals
alt
[Result = Success on Totals Comparison]
This section describes events that follow a successful totals comparison
ref
Send Reconciliation Results (see reference diagram)
[Result = Failure on Totals Comparison]
This section describes events that follow a failed totals comparison
Determine Message Route
Transfer XML Document 4
ProvideUnitBlocks ()
Generate_Unit_Blocks_Data
Transfer XML Document 5
ReceiveUnitBlocks ()
ValidateUnitBlocks
alt
[Result = Success on Unit Blocks Comparison]
This section describes events that follow a successful unit blocks comparison
Manual Processes
[Result = Failure on Unit Blocks Comparison]
This section describes events that follow a unit blocks comparison that uncovered inconsistencies
Determine Message Route
Transfer XML Document 6
ProvideAuditTrail ()
Generate_Audit_Trail
Transfer XML Document 7
ReceiveAuditTrail ()
Manual Processes

1. The ITL creates an XML document and sends it to the InitiateReconciliation Web service on the registry.
2. The ITL creates an XML document and sends it to the ProvideTotals Web service on the registry.
3. The registry sends an XML document to the ReceiveTotals Web service on the ITL.
4. The ITL sends an XML document to the ProvideUnitBlocks Web service on the registry.
5. The registry sends an XML document to the ReceiveUnitBlocks Web service on the ITL.
6. The ITL sends an XML document to the ProvideAuditTrail Web service on the registry.
7. The registry sends an XML document to the ReceiveAuditTrail Web service on the ITL.

Figure 5.2: Send Reconciliation Results Behaviour Diagram



1. The UpdateReconciliation function on the ITL sends an XML document to the ReceiveReconciliationResult Web service on the registry to inform the registry that Reconciliation is complete.

5.3 Reconciliation Stage Tables

The Stage Table represents the sequence of events in terms of the "stage" and its relation to the "status" of a reconciliation action. The stage defines where in the process of information exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and received by either a registry or the ITL or when the last step of a process occurs. The order in which each defined stage occurs may vary based on the specific process and based on the results of the ITL validation process. The numbers assigned to stages should not be used as an indicator of acceptable stage sequences.

Figure 5.3: Confirm Reconciliation

Confirm Reconcil -iation	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Confirm Snapshot Time with Registry and STL	Confirm Registry	Message 1	"Confirmed"	Registry	ITL
	Confirm STL	Message 2	"Confirmed"	STL	ITL

Figure 5.4: Reconciliation Phase 1 - Validate Account Totals

Phase 1	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Validated Totals at ITL and STL	Confirm Registry	Message 1	"Confirmed"	Registry	ITL
	Confirm STL	Message 2	"Confirmed"	STL	ITL
	Request	Message 3	"Initiated"	Registry	ITL
	Totals Sent	Message 4	"Initiated"	ITL	Registry
	Totals Evaluated	Message 5	"Validated"	Registry	ITL
	Totals By Account Sent	Message 6	"Validated"	ITL	Registry
	Totals By Account Sent	Message 7	"Validated"	STL	ITL
	Totals Evaluated by STL	Message 8	"STL Validated"	ITL	STL
	Totals Evaluated by STL	Message 9	"STL Validated"	Registry	ITL
	Reconciliation Complete	No message	"STL Validated"	--	--

Figure 5.5: Reconciliation Phase 2 - Validate Unit Blocks

Phase 2	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Inconsistent Totals at ITL	Confirm Registry	Message 1	"Confirmed"	Registry	ITL
	Confirm STL	Message 2	"Confirmed"	STL	ITL
	Request	Message 3	"Initiated"	Registry	ITL
	Totals Sent	Message 4	"Initiated"	ITL	Registry
	Totals Evaluated	Message 5	"Totals Inconsistent"	Registry	ITL
	Unit Blocks Sent	Message 6	"Totals Inconsistent"	ITL	Registry
	Unit Blocks Evaluated	Message 7	"Unit Blocks Inconsistent"	Registry	ITL
	Audit Trail Sent	Message 8	"Unit Blocks Inconsistent"	ITL	Registry
	Manual Intervention	Message 9	"Complete with Manual Intervention"	Registry	ITL
	Reconciliation Complete	No Message	"Complete with Manual Intervention"	--	--
	New reconciliation action will be initiated by ITL				

Figure 5.6: Reconciliation Phase 3 - Review Audit Logs

Phase 3	Stage Name	Stage Ends With	Reconciliation Status	Sent To	Generated From
Supplementary Program, Validated Totals at ITL, Inconsistent Totals at STL	Confirm Registry	Message 1	"Confirmed"	Registry	ITL
	Confirm STL	Message 2	"Confirmed"	STL	ITL
	Request	Message 3	"Initiated"	Registry	ITL
	Totals Sent	Message 4	"Initiated"	ITL	Registry
	Totals Evaluated	Message 5	"Validated"	Registry	ITL
	Totals By Account Sent	Message 6	"Validated"	ITL	Registry
	Totals By Account Sent Relay	Message 7	"Validated"	STL	ITL
	Totals Evaluated By STL	Message 8	"STL Totals Inconsistent"	ITL	STL
	Totals Evaluated By STL Relay	Message 9	"STL Totals Inconsistent"	Registry	ITL
	Unit Blocks Sent to STL	Message 10	"STL Totals Inconsistent"	ITL	Registry
	Unit Blocks Sent to STL relay	Message 11	"STL Totals Inconsistent"	STL	ITL
	STL Unit Blocks Inconsistent	Message 12	"STL Unit Blocks Inconsistent"	ITL	STL
	STL Unit Blocks Inconsistent Relay	Message 13	"STL Unit Blocks Inconsistent"	Registry	ITL
	Audit Trail Sent to STL	Message 14	"STL Unit Blocks Inconsistent"	ITL	Registry
	Audit Trail Sent to STL Relay	Message 15	"STL Unit Blocks Inconsistent"	STL	ITL
	Manual Intervention	Message 16	"STL Complete with Manual Intervention"	ITL	STL
	Manual Intervention	Message 17	"STL Complete with Manual Intervention"	Registry	ITL
	Reconciliation Complete	No Message	"STL Complete with Manual Intervention"	--	--
	New reconciliation action will be requested by STL				

5.4 List of Functions for Reconciliation Process

5.4.1 Registry Web Services and Functions

In order to participate in reconciliation with the ITL, registries must precisely implement Web services that the ITL can use to send it information. The following table shows the Web services methods registries are required to expose for reconciliation. Detailed technical information about the specifications for these Web service methods are in Annex C.

Figure 5.7: Registry Public Web Service Methods

Public Web Service Method	Figure
InitiateReconciliation	C5
ProvideAuditTrail	C6
ProvideTotals	C7
ProvideUnitBlocks	C8
ReceiveReconciliationResult	C9

In addition to the above Web service methods that the registry must precisely implement so that they may be used by the ITL, the registry must have additional capabilities to build transactions, validate transactions, and log transactions. The following functions implement those responsibilities. Note that these functions are not exposed to the public, so they provide more flexibility in how they are implemented.

Figure 5.8: Registry Internal Functions

Private Function	Figure
Close_Reconciliation_Action	C2
Snapshot_Data	C10
Write_To_Reconciliation_Log	C11
Write_To_Reconciliation_Status	C12

5.4.2 ITL Web Services and Functions

Like the registries, the ITL must precisely implement public Web services called ReceiveTotals, ReceiveUnitBlocks, and ReceiveAuditTrail to be used by registries in data exchange. The following tables list the public Web service methods provided by the ITL for reconciliation. Detailed technical information about the specifications for these Web services can be found in the ITL Technical Specifications.

The ITL also contains functionality for recording data. Detailed technical information about the specifications for these functions can be found in the ITL Technical Specifications.

5.5 Reconciliation Checks and Responses

The ITL executes numerous checks on all reconciliation messages to assure authenticity, format, and sequence of message. The following categories of checks are performed on the ITL. The list of specific checks and associated response codes is included in Annex E.

5.5.1 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.

5.5.2 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 reconciliation action are identifiable and eligible to participate.

5.5.3 Data Integrity Checks for Reconciliation

This category of checks is performed by the ITL to identify whether incoming messages contain data not meeting 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 and is not processed further. Data integrity checks are critical checks and if they result in failure, no further checks are 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.

5.5.4 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 has been submitted in the proper sequence, including whether process status is consistent and appropriate.

5.5.5 Other Reconciliation Checks and Messages

The ITL performs this category of checks to compare a registry's unit holding records with the ITL unit holding records.

6. ITL Administrative Functions

The ITL has five types of administrative Web service functions that involve transmitting messages to registries. These are for ensuring the integrity of the transactions and the wider accounting of units under the Kyoto Protocol. Some messages are periodic while others are initiated by administrators of the ITL or registries:

- Transaction Clean-up. This function cancels transactions that are not completed within the allowable timeframe and notifies registries of this action using AcceptNotification Web service.

- Notifications. This function on the ITL sends notifications to the AcceptITLNotice Web service on registries to notify them of actions which they need to undertake. The messages require registries to submit transactions.

- General Messages. The ITL may send general messages to a registry through the AcceptMessage Web service.

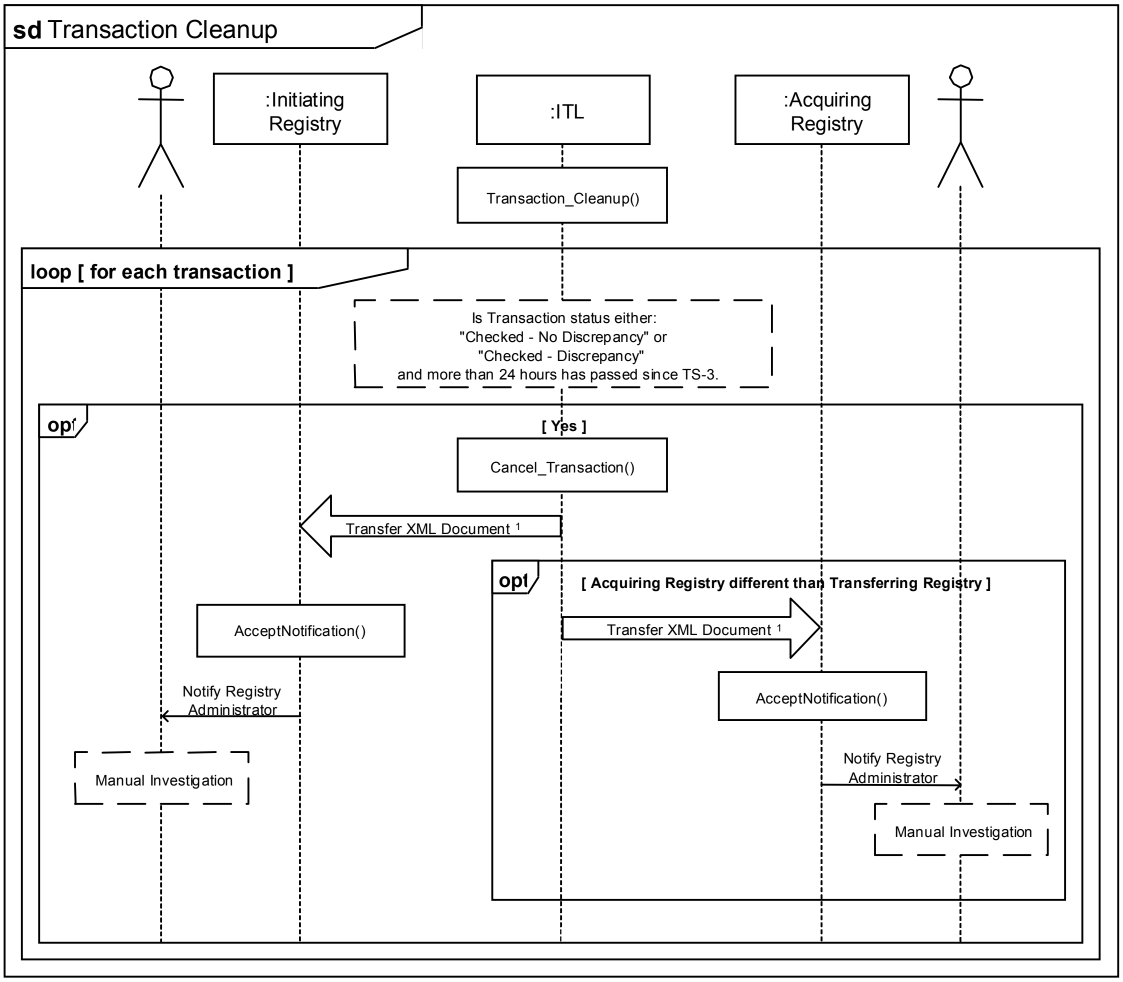
- Transaction Status Information. The registry may at any time submit a request to the GetTransactionStatus Web service on the ITL Communications Hub to provide the status of a specific transaction. This is a synchronous communication and the ITL provides the current status of the transaction as an immediate response.

- Time Synchronization. The ITL may at any time submit a request to the ProvideTime Web service on a registry to provide the time. This is a synchronous communication and the registry must provide the time as an immediate response.

6.1 24-Hour Transaction Clean-up

In order to maintain data integrity, the ITL identifies transactions in progress 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 and a notification is sent to the registries involved in the transaction. 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. The clean-up process uses the AcceptNotification Web service to send the message about a cancelled transaction to the registry.

Figure 6.1: Transaction Clean-up Diagram



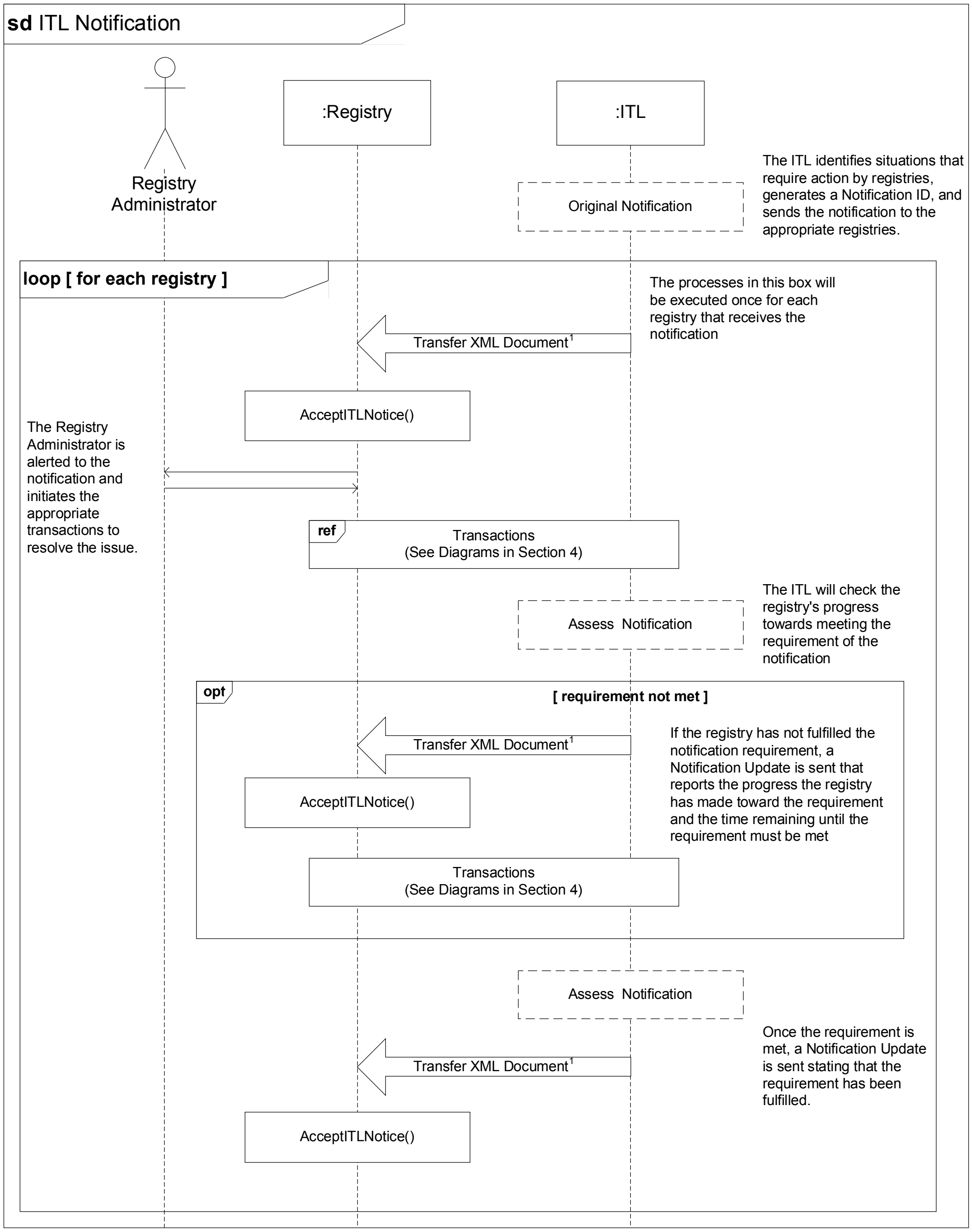
sd Transaction Cleanup
:Initiating Registry
:ITL
:Acquiring Registry
Transaction_Cleanup()
loop [for each transaction]
Is Transaction status either: "Checked - No Discrepancy" or "Checked - Discrepancy" and more than 24 hours has passed since TS-3.
opt
[Yes]
Cancel_Transaction()
Transfer XML Document 1
opt
[Acquiring Registry different than Transferring Registry]
Transfer XML Document 1
AcceptNotification()
AcceptNotification()
Notify Registry Administrator
Notify Registry Administrator
Manual Investigation
Manual Investigation

1. The CancelTransaction function on the Transaction Log creates and sends an XML document to the AcceptNotification Web service on the registry to inform the registry that the transaction has been cancelled.

6.2 Notifications

The ITL performs these administrative functions upon initiation by the ITL Administrator to evaluate data and inform the registries of specific actions required. Each of these functions may result in notification to one or more registries regarding actions that must be taken by a registry. Each of these notifications are associated with a Notification Type Code defined in Annex G and may be repeated by the ITL as reminders of the required action. All notifications use the AcceptITLNotice Web service.

Figure 6.2: ITL Notification


sd ITL Notification
Registry Administrator
:Registry
:ITL
Original Notification
The ITL identifies situations that require action by registries, generates a Notification ID, and sends the notification to the appropriate registries.
loop [for each registry]
The processes in this box will be executed once for each registry that receives the notification
Transfer XML Document1
AcceptITLNotice()
The Registry Administrator is alerted to the notification and initiates the appropriate transactions to resolve the issue.
ref
Transactions (See Diagrams in Section 4)
Assess Notification
The ITL will check the registry's progress towards meeting the requirement of the notification
opt
[requirement not met]
Transfer XML Document1
If the registry has not fulfilled the notification requirement, a Notification Update is sent that reports the progress the registry has made toward the requirement and the time remaining until the requirement must be met
AcceptITLNotice()
Transactions (See Diagrams in Section 4)
Assess Notification
Once the requirement is met, a Notification Update is sent stating that the requirement has been fulfilled.
Transfer XML Document1
AcceptITLNotice()

1. The function prepares and sends an XML document to the AcceptITLNotice Web service on the registry.

6.2.1 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. These 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.

6.2.2 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 valid for the subsequent Commitment Period that it is required to cancel within 30 days as part of a non-compliance cancellation action. 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.

6.2.3 Impending tCER or lCER Expiry

The ITL will notify each registry of the unit blocks of any tCERs or lCERs that are to expire within 30 days. The notification indicates that the specified tCERs or lCERs are to be replaced or cancelled before their expiry dates. The registry will initiate replacement or cancellation transactions that reference the Notification ID sent by the ITL. Units cancelled will be transferred to the Voluntary Cancellation Account (Account Type Code 230).

6.2.4 Reversal of Storage for CDM Project

At the request of the CDM Executive Board in the case that a reversal of storage of greenhouse gases has occurred at a CDM Project, the ITL will temporarily suspend transfers of all lCERs generated by the Project (except to cancellation or replacement accounts). The ITL will then calculate how many units each registry must replace, on the basis of their holdings (excluding cancelled or previously replaced units) of the affected lCERs, and notify each affected registry of the requirement to replace this quantity of lCERs within 30 days.

The registry will then initiate replacement 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 replacement. Once the required replacement has been completed, the ITL will restore the eligibility of the lCERs to be transferred.

6.2.5 Non-submission of Certification Report for CDM Project

At the request of the CDM Executive Board in the case that the participants in a CDM Project have not submitted a certification report for the Project, the ITL will make lCERs generated by the Project ineligible for transfer (except to replacement and cancellation accounts). The ITL will notify each affected registry that these lCERs must be replaced or cancelled within 30 days. The registry will then initiate replacement or 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 replacement.

6.2.6 Notification Regarding Excess Issuance for CDM Project

In the case that the CDM Executive Board requires a designated operational entity (DOE) to transfer units to a cancellation account, within 30 days, as a result of excess CERs, tCERs or lCERs having been issued for a CDM Project, it shall inform the DOE of this requirement and provide it with a Notification ID. The ITL will notify registries of the required cancellation to be undertaken by the DOE, using the same Notification ID provided to the DOE by the CDM Executive Board. The units must be cancelled into the Excess Issuance Cancellation Account (Account Type Code 240) at the CDM Registry. The entity will then initiate transactions, via registries, providing reference to the Notification ID so the ITL can track when the required cancellation has been completed.

6.2.7 Commitment Period Reserve

Where an upward revision of a Party's CPR level raises it above the registry's current holdings of units, or where the cancellation or replacement of units within the registry reduces unit holdings below the CPR level, the ITL will notify the Party of the quantity of units by which it is required to increase its unit holdings within 30 days. The registry will acquire sufficient units from other registries to meet this requirement. Since transactions are submitted by the transferring, not acquiring, registry, these transactions will not reference any Notification ID.

6.2.8 Unit Carry-over

After the end of the true-up period and after the Compliance Committee has completed its consideration of all information reviewed under Article 8 of the Kyoto Protocol, the ITL notifies each registry of:

- All units of each type within that registry for that Commitment Period that have not been retired, cancelled, or used in replacement.

- The number of units of each type which the registry may carry-over within 30 days.

A separate notification will be sent for each unit type. The registry will then initiate carry-over transactions, up to the limits specified in the notifications, within 30 days. For all carry-over transactions, the transaction must contain the Notification ID. The registry will also cancel, within 30 days, any units specified in the notifications which are not carried over. For any cancellation transactions, the proposal should not contain the Notification ID.

6.2.9 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.

6.3 General Messages

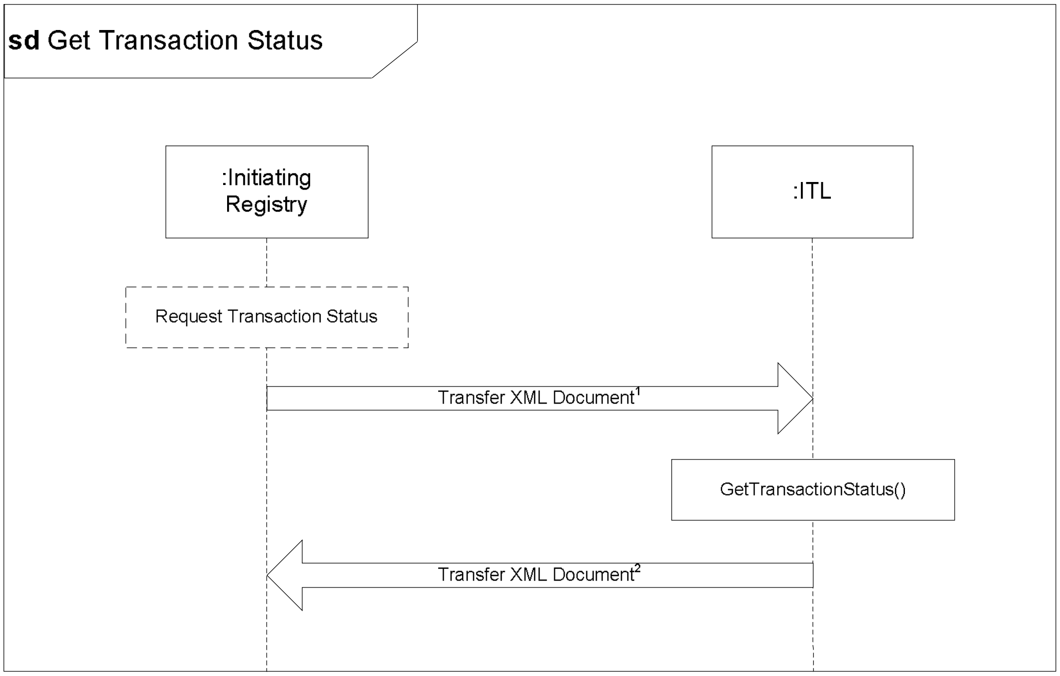
The AcceptMessage Web service at a registry may be used to deliver general messages to the Registry Administrator. These messages could involve planned ITL maintenance periods, change management, time synchronization problems, or other operational issues and plans. This communication channel offers a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

6.4 Transaction Status Service

The ITL provides a public Web service to return the current status of a transaction at the ITL. This service may be used by registries to query the status of a transaction for which verification has not yet been received.

Registries may call the GetTransactionStatus Web service method on the ITL with a specified transaction number, and the most recent transaction status will be returned immediately to the registry.

Figure 6.3: Get Transaction Status Diagram



1. The Initiating Registry prepares and sends a request to the GetTransactionStatus function on the Transaction Log.
2. The GetTransactionStatus prepares and sends a response to the calling function on the Initiating Registry with the status of the specified transaction.

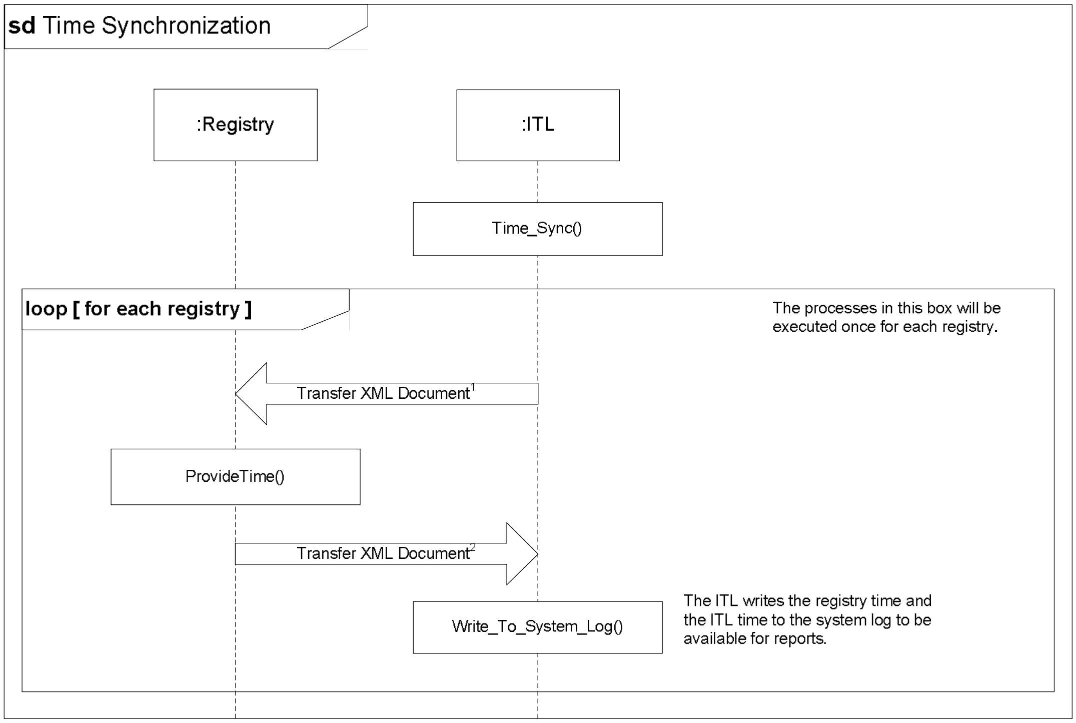
6.5 Time Synchronization

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 more than a specified amount, a message is sent to the Registry 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.

The ITL will log the time synchronization result and contact the Registry Administrator using a manual process or through a general message if a time problem is identified.

Figure 6.4: Time Synchronization Diagram


sd Time Synchronization
:Registry
:ITL
Time_Sync()
loop [for each registry]
The processes in this box will be executed once for each registry.
Transfer XML Document1
ProvideTime()
Transfer XML Document2
Write_To_System_Log()
The ITL writes the registry time and the ITL time to the system log to be available for reports.

1. The function Time_Sync prepares and sends a request to the ProvideTime Web service on the registry.
2. The ProvideTime function returns a response to the ITL with the current system time on the registry.

7. Data Logging Specifications

To support the need for the Transaction Log and registries to maintain accurate and consistent information, and to provide tools for use in the reconciliation process to resolve inconsistencies, five types of data logs shall be maintained by the registries and the ITL:

- A transaction log (including both transaction summary and detailed unit holdings);
- A reconciliation history log;
- A notification log;
- An internal audit log; and
- A message archive.

These logs are required to support auditing functionality, both internal and external. The reconciliation process constitutes one type of external audit of a registry.

All data in these data logs shall be maintained until, at minimum, the end of the third Commitment Period after the applicable Commitment Period of the associated units. In the case of ICERs, all related data shall be maintained until, at minimum, the end of the second Commitment Period after the latest crediting period of the associated units. Data older than one year may be archived to a secure location outside of the registry or Transaction Log, as long as it is can be retrieved or accessed within a 48 hour period should an inconsistency or question arise.

General messages, such as those received through the Accept Message Web service, should be maintained by the registry, but no specific logs are required or recommended.

7.1 Transaction Log

The Transaction Log contains a record of each proposed transaction sent to the ITL. Each record contains a summary of the transaction content and the subsequent outcome of the transaction. Registries will be required to provide Transaction Log data to the ITL involving specific units if an inconsistency is found for specific units during a reconciliation process.

In general, it is recommended that the logging of a transaction message sent to the ITL occur after the receipt of a SOAP response indicating that the message was successfully transmitted and received.

The information in Figure 7.1 shall be maintained in the Transaction Log. A specific data model for these data is not required.

Figure 7.1: Transaction Log Attributes

Attribute	Notes
Transaction Identifier	
Transaction Type	
Supplementary Transaction Type	Required for registries subject to a supplementary program. This would be null for non-EU registries.
Transferring Account Type	
Transferring Account	
Acquiring Registry Identifier	
Acquiring Account	

(cont.)

Figure 7.1: Transaction Log Attributes (cont.)

Attribute	Notes
Acquiring Account Type	
Notification ID	
Transaction Status	Contained in child table.
Transaction Status Date-Time	Contained in child table.
Unit Block(s)	Contained in child table.
Response Code(s)	Contained in child table along with unit block data.

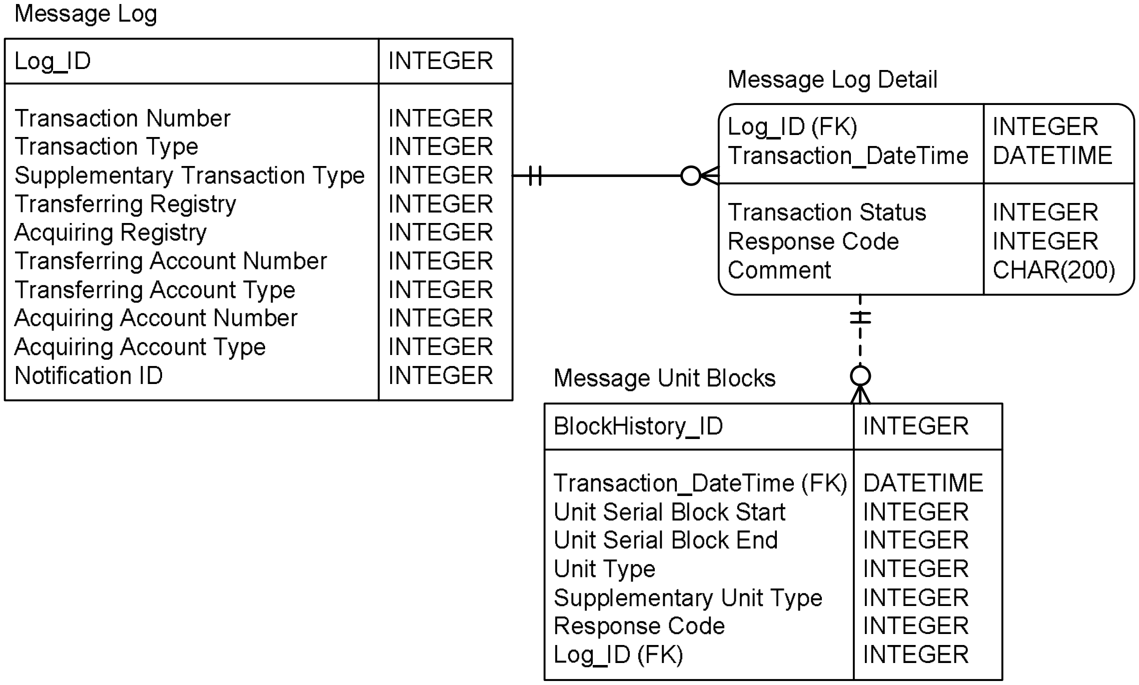
To support this information, three tables in parent-child relationships are necessary:

- A parent table containing the transaction identifier, related attributes and status information;

- A child table identifying the various statuses a transaction may be processed through; and

- A second child table identifying serial blocks and response code results. The diagram in Figure 7.2 below contains an example entity-relationship model of these tables.

Figure 7.2: Transaction Log Entity Relationship Diagram



Message Log

Log_ID	INTEGER
Transaction Number Transaction Type Supplementary Transaction Type Transferring Registry Acquiring Registry Transferring Account Number Transferring Account Type Acquiring Account Number Acquiring Account Type Notification ID	INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER

Message Log Detail

Log_ID (FK) Transaction_DateTime	INTEGER DATETIME
Transaction Status Response Code Comment	INTEGER INTEGER CHAR(200)

Message Unit Blocks

BlockHistory_ID	INTEGER
Transaction_DateTime (FK) Unit Serial Block Start Unit Serial Block End Unit Type Supplementary Unit Type Response Code Log_ID (FK)	DATETIME INTEGER INTEGER INTEGER INTEGER INTEGER INTEGER

7.2 Reconciliation History Log

The Reconciliation Log contains a record of each reconciliation action conducted by the ITL for a registry. As described in Section 5, each Reconciliation action consists of multiple steps or sub-processes. This Reconciliation Log contains one or more records for each step in a Reconciliation action.

The Reconciliation process is initiated and driven by messages from the ITL to a registry. The registry shall log each request and its response in its Reconciliation Log. The ITL shall maintain a parallel Reconciliation Log containing all requests, responses received, and results sent to a registry. Although information in the Reconciliation Log are not shared directly as part of the Reconciliation itself, access to this information by the Registry Administrator may be necessary to identify the manual intervention needed in order to resolve inconsistencies.

In general, it is recommended that the logging of a transaction message sent to the ITL occur after the receipt of a SOAP response indicating that the message was successfully transmitted and received.

The information in Figure 7.3 shall be maintained in the Reconciliation Log. A specific data model for these data is not required.

Figure 7.3: Reconciliation History Log Attributes

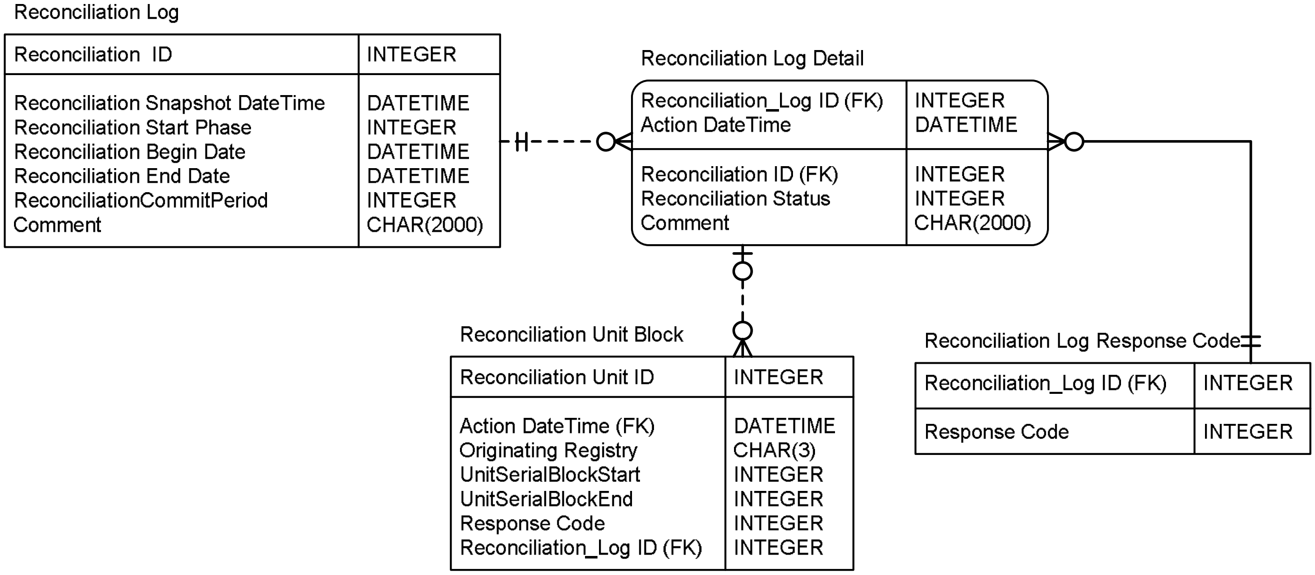
Attribute	Notes
Reconciliation ID	Unique identifier for a reconciliation action as requested by the ITL
Reconciliation Begin Date	
Reconciliation End Date	
Reconciliation Snapshot DateTime	
Reconciliation Phase	
Response Code	Contained in child table.
Unit Blocks	Contained in child table.
Reconciliation Comment	Information recorded by the Registry Manager regarding corrective actions for manual intervention.
Reconciliation Status	Contained in child table.
Reconciliation Status Log DateTime	Contained in child table.

To support this information, three tables in parent-child relationships are necessary:

- A parent table containing the reconciliation number, date and time reconciliation was initiated, ended, phase requested, and the date and time the snapshot of current holdings was requested;
- A child table identifying changes in the status of the reconciliation action;
- A child table identifying the response codes returned by the ITL and associated with the change of a status; and
- A child table to the specific activity containing the unit blocks that were identified as inconsistent and by the ITL corresponding response code information.

The diagram in Figure 7.4 below contains an example entity-relationship model of this information.

Figure 7.4: Reconciliation History Log Entity Relationship Diagram



Reconciliation Log

Reconciliation ID	INTEGER
Reconciliation Snapshot DateTime Reconciliation Start Phase Reconciliation Begin Date Reconciliation End Date ReconciliationCommitPeriod Comment	DATETIME INTEGER DATETIME DATETIME INTEGER CHAR(2000)

Reconciliation Log Detail

Reconciliation_Log ID (FK) Action DateTime	INTEGER DATETIME
Reconciliation ID (FK) Reconciliation Status Comment	INTEGER INTEGER CHAR(2000)

Reconciliation Unit Block

Reconciliation Unit ID	INTEGER
Action DateTime (FK) Originating Registry UnitSerialBlockStart UnitSerialBlockEnd Response Code Reconciliation_Log ID (FK)	DATETIME CHAR(3) INTEGER INTEGER INTEGER INTEGER

Reconciliation Log Response Code

Reconciliation_Log ID (FK)	INTEGER
Response Code	INTEGER

7.3 Notification Log

Each registry and the ITL shall also maintain a log of notifications generated by the ITL and sent to a registry via the AcceptITLNotice Web Service. These notifications inform the registry regarding specific actions that should be taken relating to units. See Section 6.

The Notification Log at the registry shall contain the attributes in Figure 7.5. A specific data model for these data is not required.

Figure 7.5: Notification Log Attributes

Attribute	Notes
Notification ID	As generated by the ITL and sent to the registry.
Notification Status	
Notification Type	
Notification Date	
Total Units	If appropriate. For example, notifications relating to Reversal of Storage.
Project ID	If appropriate.
Notification Text or Message Location	To store a complete copy of the notification content.

7.4 Internal Audit Log

Each registry and the ITL shall also maintain an internal log of changes to data which are critical to the transaction or reconciliation process. The scope and design of this functionality is the responsibility of the Registry Administrator. The internal audit log shall capture information on internal and external transactions, including in particular the user ID and date/time of all recorded transactions. Information contained in this log is not shared directly with the ITL. It is required to provide additional information for use by the Registry Administrator for manual intervention when an inconsistency is discovered in the reconciliation process.

The internal audit log shall contain the attributes in Figure 7.6. A specific data model for these data is not required.

Figure 7.6: Internal Log Attributes

Attribute	Notes
Activity Type	For example, insert, delete, update, login attempt.
Activity Date-time	
Entity Affected	For example, table name.
Field Modified	Attribute in table that was updated.
Old Value	
New Value	
User ID	Person who executed change if not performed through Web service.
Source of Activity	Identifies the server or workstation activity was submitted on.

7.5 Message Archive

Each registry and the ITL are required to store a copy of messages sent and received, in their entirety, as stand alone files. These files provide additional information for use by the registry or the ITL Administrator when an inconsistency is discovered which relates to a messaging problem which cannot be resolved through the use of the transaction history or internal audit logs.

The location and the medium for this storage are at the discretion of the registry or the ITL Administrator. The naming convention of the files must enable an authorized user to retrieve the file for a specific transaction or reconciliation. It is recommended that the files be stored in compressed formats using the following naming convention:

aa_bb-##############-cc.zip

where:

aa = Registry country code per ISO3166, and "ITL" for Transaction Log messages, and "CDM" for CDM Registry messages

bb-##############-## = Transaction identifier
and cc = sequential number generator

For example, a file sent from the ITL about a proposed German transaction would be named:

ITL_DE_152_1.zip

8. Change Management Specifications

8.1 Objectives

The Technical Specifications for Data Exchange provide a stable, agreed-upon platform for the development and deployment of the communications modules built for registries and the ITL. It is expected, however, that changes to the messages and to the criteria to ensure data quality and accuracy may be necessary over time. It is a requirement of the Technical Specifications for Data Exchange that changes provide backward compatibility to the extent possible. It is the goal of these requirements to allow existing functionality to remain valid when new requirements or changes are necessary.

Anticipating these needs, this specification establishes a technical architecture that allows adjustments within the specification without imposing significant additional development costs to registries or the ITL. A change management process, supported by Registry and ITL Administrators, to determine when and how this will be managed, shall also be established.

8.2 Procedural Controls

To coordinate the change management process, the following will be defined:

- A process to receive requests for changes in technical specifications, including message content and criteria, etc., and for the assessment of these requests;

- A communication mechanism for informing all participants of upcoming changes, including schedule, specific impacts, instructions, etc.;

- A change management process for developers or technical managers of the registries; and

- The consequences of failing to adopt required changes.

8.3 Technical Specifications

To minimize the impact of changes and to manage the process of ensuring that all participants implement required changes within a necessary timeframe, registries and the ITL must conform to the following technical specification:

8.3.1 Version Definition

The Technical Specifications shall be assigned a version number, managed through the ITL. A version number consists of two elements, a major version number and a minor version number. Major version numbers will change infrequently and reflect a fundamental change in the architecture of a core component that would invalidate any previous versions. It is likely that a major version change would require coding changes to be undertaken. A minor version number indicates small changes to message content or validation rules that would not require coding changes and could be implemented completely within the existing messages structure.

Within each XML message sent or received from the ITL, the major and minor version numbers are checked. A registry that has an incompatible major version number will have all of its requests rejected and receive a response indicating that the major version of the Data Exchange Standards is out-of-date. A registry that has an incompatible minor version number will be directed to an upgrade site and may or may not have the request processed based on the nature of the change.

8.4 ITL Web Portal

The ITL will provide an extranet website that will post information on version status as well as allow registered users to review upcoming functional changes, time for implementation, and the technical specifications for these changes. Users who have valid user accounts and passwords through the VPN will have access to this site. The site will maintain a history of all version changes and patches released from the site. It will be managed by the ITL Administrator.

8.4.1 Web Service Modifications

Changes to Web services or subsequent functions that require new parameters constitute a major version change. All registries will have a specified period of time to comply with the new requirements. Detailed specifications will be provided with sample testing procedures for the registries to test the new components against the ITL.

8.4.2 Support Table Content Modification

Changes to data content in the form of new response codes or support tables are considered minor version changes. These data will be available in XML format for download from the ITL website. Included in these tables are codes identifying which response codes are new, have been modified, or have been retired. Registries must refresh their tables with current support table data as needed.

9. Registry Initialization Specifications

Initialization is the process of bringing a registry system on-line, allowing it to fully participate with the ITL in a trading scheme. Prior to a registry participating in message exchange with the ITL, the registry must comply with a series of initialization requirements and procedures. These tasks ensure that the registry meets both the functional and non-functional requirements of the Data Exchange Standards and will be able to converse consistently with the ITL. The registry will not be able to participate in any transactions until all initialization tasks are complete.

9.1 Staff Identification and Planning

The first step to initialize a registry is to identify those individuals responsible for the registry and its operation, including the Registry Manager. The Registry Manager, or a person assigned by the Party, must submit a schedule and plan for initialization to the ITL Manager. The schedule and plan, should detail, in writing, the timing projected for each major initialization task, including projected start and end dates. This is necessary so that the ITL Managers can ensure that the appropriate level of support and assistance is available during this period. Since it is anticipated that multiple registries will be in the process of testing and initialization simultaneously, the ITL Manager will assign a primary staff person to work with each registry during this period. It will be important for the ITL and registry staff to develop a working relationship and excellent communication.

Although no specific format is required, it is recommended that the schedule and plan address the following areas, which comprise an initialization checklist:

Registry Checklist

- Assign Registry Manager and support staff
- Define initialization schedule with projected milestone completion dates. Initialization should be completed within a 2-month time period.
- Submit Registry Documentation to ITL Manager
- Enable VPN access to and from ITL
- Obtain and test registry digital signature
- Perform tests and assess results received back from the ITL Manager
- Set up production environment
- Verify time synchronization with ITL
- Provide government account information for Cancellation, Replacement, and Retirement Accounts for the first Commitment Period

ITL Checklist

- Assign ITL Manager and primary staff to work with Registry Manager
- Review registry schedule and set up ITL schedule
- Review registry documentation
- Enable VPN access to and from registry
- Set up digital signature and participate in security and authentication tests
- Assist with registry tests
- Participate in tests and prepare analysis of test results
- Receive account and contact data from registry
- Set up production data for registry
- Process and quality assure government account information

9.2 Documentation

The first task of a registry is to provide documentation of their registry system. This documentation is needed to show that non-functional requirements of the DES are met and that the registry will be operated in a manner consistent with excellent operating practices. These requirements ensure the national registry has an adequate plan for addressing operational and security requirements of the application.

9.2.1 Database and Application Backup

A database and application backup plan should be submitted outlining a detailed backup plan for the production database and software. It is recommended that database backups be performed at a minimum frequency of daily.

Specific elements of the plan include:

- Identification of personnel responsible for backup (include a primary individual and an alternate, or a staffing plan);

- Identification of specific back up schedule and procedures (i.e., backup at 7pm each evening from terminal X by User ID Y);

- Identification of backup media and its location;

- Identification of the number of backup generations planned;

- Definition of strategy to monitor performance of backup tasks, including notification of backup failures, log review, spot checks, audit, management reporting, etc.;

- Identification of scope or content of backup procedures (i.e., database, application software, server logs, etc.); and

- Identification of backup hardware and software.

9.2.2 Disaster Recovery Plan

A disaster recovery plan designed to ensure business continuity in the event of catastrophic failure or disruption of the host environment should be submitted in conjunction with the backup procedures. The primary objective of a disaster recovery plan is to enable an organization to survive a disaster and to reestablish normal business operations as quickly as possible. In order to survive a catastrophic event, the organization must assure that critical operations can resume normal processing within a reasonable time frame. A contingency plan should be laid out in the event that the primary facility cannot perform required daily operations. In order for this plan to be effective, periodic testing and evaluation should be performed to ensure validity and viability.

Specific elements of the plan include:

- Identification of an off-site facility with adequate disk space/storage and availability to serve as an emergency hosting environment;

- Definition of specific minimum hardware and software requirements to host the registry on a temporary basis;
- Definition of roles and responsibilities for primary and alternate personnel at the off-site location;

- Definition of roll-back procedures to step back to the latest backup. This may include obtaining daily transactions from the ITL that were not included in the last backup;

- Notify all appropriate parties that a contingency plan is in effect (i.e., ITL, other registries or users);

- Identification of off-site location of documentation and procedure manuals, as well as any paper-based forms, necessary to deploy under a Disaster Recovery scenario;

- Definition of periodic testing strategy to demonstrate readiness to implement disaster recovery plan; and

- Definition of expectation for time frame in which registry could begin operation following a disaster. The time frame would depend on the volume of transactions, cost and other factors and is not expected to be the same for each registry.

9.2.3 Security Plan

A security plan is defined in order to protect the application and data from unrestricted and unsolicited use. Secure access to the data should be provided at multiple intervals to insure redundancy of protection. For Web security, the following three primary areas should be addressed:

1. Server security. The Web and/or database server should be secured not only by user id and password but also physically to prevent unauthorized access to the data and application. As with most dynamic connectors to databases, a connection with full access must be granted to the Web server because various queries will need to access different tables or views to construct the HTML from the query. To prevent unauthorized use of these open data connections, the servers should be physically secured. In addition, security can be assigned at the table level on a database.

2. User-authentication security. This level of security insures no unauthorized access to information in the registry. This is accomplished by requiring unique user id's and passwords that are regularly maintained by a Systems Administrator.

3. Session security. This level of security insures that data is not intercepted as it is broadcast over the Internet. This is accomplished by encrypting data passed to and from the registry.

Specific elements of the plan include:

- Definition of rules and responsibilities for security, recognizing that actions by persons are the most significant contributing factor to the success or failure of security planning;

- Determine physical access to the Web and/or Database server;

- Assign a network and database administrator and alternates, user id, passwords, and specific responsibilities;

- Activate audit trails recording activities at the server, database and data levels;

- Employ encryption of data transferred to and from the registry;

- Require frequent changes to password, restricting replication over a period of time;

- Require passwords of specific length with a specific number of alpha and numeric characters. For example, 6 digit passwords with a minimum of 2 numbers; and

- Delete all unused User ids and passwords immediately and remove inactive user ids from the database on a regular basis.

9.2.4 Application Logging Documentation

To demonstrate conformance with Section 7 of the Data Exchange Standards, the Registry Manager is asked to provide a summary of the registry capability to maintain database logs and activity logs.

- Database Logging. Database administrators are required to implement transaction logging where logs for files can be periodically shipped to a remote server or alternate site. For Oracle databases, this is the equivalent of archive logging; for MS SQL Server this might be implemented with log shipping.

- Activity logging**.** Activity logging should be utilized to track unauthorized attempts to log on to the server as well as general usage.

The documentation should include:

- Definition of regular backup and archival of transaction logs;

- Definition of hardware utilized to store logs; and

- Assignment of personnel to review activity logs on a regular basis.

9.2.5 Time Validation Plan

For the successful data exchange, a registry must define and follow specific procedures to validate server time on a periodic basis.

The plan should include:

- Schedule for periodic time validation;
- Identification of time server to provide validation;
- Assignment of personnel to perform or monitor time validation;
- Maintenance of documentation of time validations and any time adjustments resulting;
- Definition of tolerance for time validation discrepancies; and
- Definition of process for adjusting time.

9.2.6 Version Change Management

A clear migration path should exist to upgrade from version to version of registry software and database schemas. When a new version is released it must go through the testing sequence to insure that it is operable. This invokes preparing a testing environment and a test plan and a migration path to move the code and database schema to production assuming it has passed the testing sequence.

The Change Management Plan should include:

- Deployment Strategy
- Test Plan
- Notification strategy
- Data management/loading plan

9.2.7 Test Plan and Test Report

The test plan ensures that a registry has performed basic testing and is capable of participating in the tests outlined in Annex H which are required of a registry prior to being authorized to submit production transactions to the ITL. The test plan describes the various levels and types of testing that will be done throughout development.

A test plan should be outlined that steps through the basic system tasks to ensure no changes made in the test environment will affect day-to-day processing. This should cover System Administrator functionality, as well as all user-level testing. All test cases should be documented and archived for proof of concept and documentation purposes. A migration plan should be clearly outlined to move the test code, schema, or data to the production environment with minimal impact to the overall system (choose times of least traffic).

The test plan should include:

- Description of overall test strategy, testing procedures and documentation;

- Identification of Test criteria;

- Identification of Testing tools;

- Assignment of personnel to perform testing of the software, both on the initial release and for an upgrade in hardware or software;

- Description of test environment and management of that environment to ensure that results replicate the results expected in a production environment;

- Evidence that the plan provides for systematic testing in logical order of all module; subsystem, and system requirements against a well-defined set of test cases;

- A method for documenting the performance of all tests in a test log; a method for identifying and reporting any anomalies or errors; and a procedure for tracking problems from detection to resolution;

- Plans for creating the test environment, including all needed software and hardware purchases, are consistent with the application development schedule; and

Evidence that regression testing is a fundamental element of the plan.

9.2.8 Operational Plan

The operational plan ensures that the registry has appropriately planned and staffed the operational requirements of the registry, so that the ITL Administrator can foresee that the registry will continue to maintain effective operation once the registry has been approved to operate in a production mode with the ITL. The operational plan will address many of the requirements for the initial approval, but will provide a demonstration that the initial standards and requirements will be addressed on an ongoing basis.

The operational plan should include:

- Definition of operational logs and record keeping;
- Staffing and management; including training
- Security management plan
- Ongoing performance evaluations and assessments
- Data management strategy, including archiving and data quality assessment
- Modernization and technology assessment strategy
- Technical support plan

9.3 Initialization Tests

Once the ITL Administrator is satisfied these requirements have been met, the national registry can begin to establish electronic communication with the ITL. These procedures are performed in stages and are described below. Detailed requirements and processes for these tests are defined in Annex H.

Figure 9.1: Table of Initialization Tests

Test	Test Type	Who Initiates the Test	Description of Test
Communication Initialization	Required	Registry and ITL	Installation, initialization and test of the VPN.
ITL Extranet Login	Required	Registry	Creation and verification of ITL extranet account and password.
Registry Transaction Web Services	Required	Registry	Registry tests ITL Web services.
ITL Transaction Web Services	Required	ITL	ITL tests the registry Web services.
Query Services	Recommended	Registry	Test of querying capabilities.
Registry Reconciliation Web Services	Required	ITL	Web service test from ITL requesting reconciliation data.
ITL Reconciliation Web Services	Required	Registry	Web service test in which registry submits reconciliation data.
Data Request	Required	ITL	Web service initiated by the ITL requesting data from a registry.
Data Identifier Initialization	Recommended	Registry	Download and import of lookup table data from ITL extranet website.

9.4 Communication Initialization

A registry must be able to demonstrate that a secure communication channel can be established to and from the ITL Communications Hub. The Registry Manager may elect for the ITL Manager to remotely administer the VPN. This requires the ITL Manager to assist or direct the installation and configuration of the VPN at the registry network. The test to validate connectivity can be performed at the time of installation. If the Registry Manager elects to install and configure the VPN, then an appointment shall be negotiated with the ITL Manager to test VPN connectivity. The test must demonstrate that the ITL and registry are able to connect to and send transmissions to and from each other. The IP address for both the registry VPN and ITL VPN shall be recorded and documented as valid and trusted connections to each other. This test shall be conducted and completed within a single business week. The ITL Manager shall notify the Registry Manager whether the test result was accepted or rejected as incomplete. The results of the Communication Initialization Test are recorded in the ITL database.

The tests conducted are as follows:

- Registry must test for Internet access;

- Registry records IP address of ITL for site-to-site configuration of the VPN. ITL will supply the configuration specifications for the VPN;

- Registry pings ITL IP address to validate VPN hardware can see ITL VPN;

- ITL records IP address of registry;

- Registry acquires digital certificate from Third Party Certificate Authority and install appropriate files;

- ITL is sent public key of certificate either from Certificate Authority or from the registry; and

- Authentication test of Digital Certificate is initiated by the sending and receiving of public keys.

9.5 Access to ITL Website

There are two websites that support distribution of data from the ITL database, one which is public and one which requires security to access.

9.5.1 Public ITL Website

Access to the ITL public website for the purposes of querying unit transparent data does not require an account or password. The Registry Manager is responsible for checking the site to review content and alert the ITL Manager of any discrepancies in data. This test is for the benefit of the registry and does not require confirmation from the ITL. This test can be performed at any time.

9.5.2 Access to ITL Extranet

The ITL extranet hosts information regarding change management files or patches as well as XML datasets of all response codes and key identifier tables. Access to this site requires a login and password. The Registry Manager must request an account and password for access to this site. This test is for the benefit of the registry and does not require confirmation from the ITL. This test can be performed at any time.

9.6 Web Services Testing

The ITL Manager will host both a test and production environment for registries to test Web services independent of production data. Testing of Web services includes registries' testing Web services against the ITL and the ITL Manager testing the registries' Web services.
All registries must first test all Web services through the ITL test environment. These tests shall confirm that all Web services have met functional specifications. The Registry Manager shall negotiate a timeframe for conducting these tests with the ITL Manager. The ITL Manager shall notify the Registry Manager as to whether the test results were accepted or rejected as incomplete. The results of the Web Services Test are recorded in the ITL database. These tests shall include:

- Registry Transaction Web Service Tests
- ITL Transaction Web Service Tests
- Query Web Service Tests
- Registry Reconciliation Web Services
- ITL Reconciliation Web Services

9.7 Request for Other Data

The ITL requires that a registry provide information to the ITL for possible distribution to the various ITL websites. These data include information regarding account and representative information as well as general queries for time synchronization. The ITL Manager shall negotiate a time frame for performing these requests with a Registry Manager. These tests shall be conducted and completed within a single business day. The ITL Manager shall notify the Registry Manager regarding whether the test results were accepted or rejected as incomplete. The results of the Data Request Test are recorded in the ITL database. The following Web services will be tested for the submission of data. For additional information on the content and methodology for Web service testing, see Annex H.

9.8 Data Identifier Initialization

Registries are expected to load all response code data as well as data for all key identifier lookup tables into their systems. The following table identifies the data that the registry needs to download from the ITL extranet website for integration into their systems. This initialization is for the benefit of the registry and does not require confirmation from the ITL. This data initialization can be performed at any time. The datasets are listed below.

Figure 9.2: Look-up Table Initialization

Data Set	Description
Account Type Code	Account Descriptions
Registry Code	ISO-1066 Country Codes and Identifiers for Registries
Unit Type Code	Identifies type of unit
Supplementary Unit Type Code	Identifies additional unit type codes for an STL
Transaction Type Code	Identifies the type of transaction
Transaction Status Code	Identifies the status of a transaction
Response Catalog	Lists of all possible response code and descriptors
Supplementary Transaction Code	Identifies additional transaction type codes for an STL
Notification Type Code	Identifies types of notifications

9.9 Full System Test

Once each individual component of the registry-ITL communication system has been tested, the registry should initiate a series of transactions and allow them to continue to completion without manual intervention. These transactions will be documented in the test plan and will be representative of the different types of communications between the registry and the ITL. Once each transaction has ended with a predictable result, the registry is certified to use the production environment.

9.10 Reconciliation Services and Schedule

Initially, requests for reconciliation data shall be requested daily and after a timeframe in which no errors have been reported over a sample period of time involving numerous transactions the ITL Manager and Registry Manager may negotiate a less frequent time frame for reconciliation requests. The Registry Manager may also negotiate the time of day in which it does not present an undo burden to provide reconciliation data requests to the ITL. Failure to provide reconciliation data when requested can cause the suspension of transaction privileges.

9.11 Government Account Information

Prior to receiving authorization to operate in production mode with the ITL, the registry will provide to the ITL information about the account identifiers and account types for all government accounts used by the registry for holding units for cancellation, retirement and replacement purposes. The registry will provide this information to the ITL Administrator in a data file in the comma delimited format as defined in Figure 9.3.

Figure 9.3: Government Account File Specifications

Column #	Data Element	Data Attributes	Description
1	Registry Code	Alphanumeric (3)	Per Annex F
2	Account Identifier	Numeric	Per Annex F
3	Commitment Period	Numeric	Per Annex G
4	Account Type Code	Numeric	Per Annex G
5	Account Status	Numeric	1 = Open 2 = Closed
6	Action	Numeric	1 = Add 2 = Update 3 = Delete (or archive)

This format will be used on an ongoing basis to update these records, which are used to verify that the Commitment Period and account type are accurate when transactions are submitted to cancel, retire or replace units.

For registries which will initialize for the ITL after first initializing with the STL, the STL will provide the required government account data to the ITL for this purpose.