DATA EXCHANGE STANDARDS FOR REGISTRY SYSTEMS UNDER THE KYOTO PROTOCOL

DRAFT
TECHNICAL SPECIFICATIONS

ANNEXES

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

[This page intentionally left blank.]

Annex A
Glossary of Terms

Figure A1: Glossary of Terms

Term	Definition
AAU	Assigned amount units. These are tradable units derived from an Annex I Party's emissions target under the Kyoto Protocol. They may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
Account	An account is used to partition a registry and can hold units. There are three accounts types: holding account, cancellation account and retirement account.
Accuracy	Condition in which information is not modified randomly by the software system.
Acknowledgement	An acknowledgement is the communication that is returned by a Web service (located at either the Transaction Log or at a registry) that a message has been successfully received. The acknowledgement occurs before the message is evaluated in any way other than format checks and minimum version requirements.
Administrator	A role to configure and maintain a software system. Configuration can range from system set-up to amending data and parameters within the system.
Annex I Party	A Party to the UNFCCC listed in Annex I to the UNFCCC. These are industrialized countries, including those with economies in transition.
Article	An Article of the Kyoto Protocol.
Attribute	Identifier for a piece of information.
Audit	Checking of recorded data.
Authentication	The process to confirm the identity of a user.
Authorization	The process to verify a permission to do something.
Cancellation	Cancellation is the action taken by the ITL for a proposed transaction when no response has been received from a registry within 24 hours.
CER	Certified Emission Reduction unit. These are tradable units generated by Projects that reduce emissions in Non-Annex I Parties under the CDM. They may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
CDM	Clean Development Mechanism under Article 12 of the Kyoto Protocol. Projects in Non-Annex I Parties under the CDM result in reduced emissions, or enhanced removals. These Projects generate credits in the form of CERs, tCERs or lCERs.
CDM Executive Board	The board supervising the CDM. It is serviced by the Secretariat.
CDM Project	A Project under the Clean Development Mechanism under Article 12 of the Kyoto Protocol.

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
CDM Registry	The registry established by the CDM Executive Board on behalf of Non-Annex I Parties hosting CDM Projects. It is to ensure the accurate accounting of transactions of CERs, tCERs, and lCERs by those Parties. Its administrator is the Secretariat.
Certificate Authority	Provides a Digital Certificate for site-to-site authentication to positively identify an organization and encrypt data communications between the organization and other certificate holders.
Commitment Period	A specified period in which an Annex I Party is to show compliance with its emissions target. The first Commitment Period for the Kyoto Protocol is from 2008 to 2012.
Communications Hub	The central communications component integrated in the ITL, through which all registries, the ITL, and any STLs communicate.
Component	A component is a group of programming functions that perform related tasks.
Conference of the Parties (COP)	The supreme decision-making body under the UNFCCC. Attended by delegations from all state Parties to the UNFCCC. The COP generally meets once a year.
Crediting Period	The period for which emission reductions or enhancement in removal of greenhouse gases from the atmosphere from a CDM or Joint Implementation Project are monitored and verified.
Customisation	Configuration of systems toward specific user needs within certain boundaries.
Denial of Service Attack	A very high number of requests in very short period aimed at a software system with the goal of achieving an overload and crash of that software system.
Digital Certificate	Provided by the Certificate Authority to ensure authentication of documents.
Discrepancy	A discrepancy is a finding by the ITL that a proposed transaction does not conform to agreed transaction rules.
Downtime	The time in which a software system is not available for use.
Emissions Trading	The trading of units which may count towards compliance by Annex I Parties with their emissions targets. Emissions trading is provided for under Article 17 of the Kyoto Protocol. Domestic (e.g., UK) and regional (e.g., EU) emissions trading schemes are also being established under the umbrella of Article 17 emissions trading under the Kyoto Protocol.
Encryption	A way of protecting data from unauthorized access.
Entities	Legal entities authorized by a government to participate in emissions trading or joint implementation Projects. Private and/or public entities involved in the CDM. Such entities may be from public, private or non-governmental sectors.

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
ERU	Emission reduction units. These are tradable units generated by Joint Implementation Projects in Annex I Parties. They may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
Exchange Mechanism	System for exchanging data.
Finalization	Finalization is the action taken by a registry to complete a transaction which has been validated by the Transaction Log.
Function	A specific section of programming code within a component which performs a specific task.
Functional Requirement	Requirement for which the quality test result is binary (e.g., yes/no or right/wrong).
GUI	Graphical User Interface.
Inconsistency	An inconsistency is a finding by the ITL that the unit information provided to the registry as part of a data reconciliation process differs from the information retained by the ITL.
Integrity	Assurance that data cannot be modified by any Party not authorized to do so.
International Transaction Log (ITL)	An electronic database established by the Secretariat to monitor the validity of transactions between registries under the Kyoto Protocol.
Invalidation	An invalidation is a finding by the Transaction Log that a message does not conform to the messaging requirements (including data formats, identifiers, etc.) in these Technical Specifications.
Joint Implementation Project	A Project under Joint Implementation under Article 6 of the Kyoto Protocol.
Kyoto Protocol	Allied agreement to the UNFCCC containing emission reduction targets for Annex I Parties.
lCER	Long-term Certified Emission Reduction unit. These are tradable units generated by Projects that enhance removals of greenhouse gases from the atmosphere in Non-Annex I Parties under the CDM. lCERs expire at the end of the crediting period of the Project (though these crediting periods may be renewed such that the Project may continue for up to 60 years). lCERs may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
Logging	Functionality of a software system that stores information on the system for auditing and tracking.

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
Major Version Number	A major version number is the number assigned to the Technical Specifications for the Data Exchange Standards for purposes of identifying a specific set of technical requirements. The major version number changes only when a change in the Technical Specifications requires programming changes in a registry.
Minor Version Number	A minor version number is the number assigned to the Technical Specifications for the Data Exchange Standards for the purposes of identifying a specific set of technical requirements. The minor version number changes do not require programming changes within registries. These changes may involve response code table updates, for example.
Message	A message is a communication between the ITL and a registry or STL. It includes all data exchanged, including transaction and reconciliation data, requests for logs and responses. Messages are transported through HTTP SOAP requests.
National Registry	A registry established by an Annex I Party.
Net Source	A Net Source is an activity which emits more greenhouse gases than it absorbs over a given period. A Net Source Cancellation is a transaction specific to the case where a LULUCF activity under Article 3.3 or 3.4 of the Kyoto Protocol, which would generally result in RMU issuance through its net absorption of greenhouse gases, is found to be a Net Source.
Non-Annex I Party	A Party to the UNFCCC which is not listed in Annex I to the UNFCCC. These are developing countries.
Non-functional Requirement	Requirement for which the quality test result is measure or a score (e.g., from 1-10 or high/medium/low)
Notification	A notification is a communication to a registry from the ITL about a required or recommended action involving unit transactions.
Party	A state that has ratified the Kyoto Protocol.
Process	The business area or category of interaction between registries and the ITL. The primary processes are unit issuance, unit conversion, external transfers, internal transfers (including cancelation replacement, and retirement), unit carry-over, expiry date change, and reconciliation. In addition, there is an ITLadministration process which addresses the need to manage message exchange failures, reference data, and manual intervention relating to reconciliation processes.
Protocol	Formal rules describing how to transmit data.
Reconciliation	The process by which data from different registry systems are compared and inconsistencies are resolved.
Recovery	The complete re-installation and reconfiguration of data or a software system.

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
Registry	A software system for the accounting of transactions in AAUs, RMUs, ERUs,CERs, tCERs and lCERs. Includes national registries and the CDM registry.
Registry System	Generic term for national registries, the CDM Registry and Transaction Logs.
Removal	Removals of greenhouse gases from the atmosphere through LULUCF activities. Such removals may lead to the generation of RMUs, tCERs or lCERs. They are the "opposite" of emissions of greenhouse gasses.
Response	A response is the data sent following the processing of a proposed transaction. Typically the response includes the transaction ID, an indicator that the proposed transaction was successful or unsuccessful, and, if unsuccessful, the response code(s) providing the reason for the failure.
Reversal of Storgage	A Reversal of Storage refers to a case in which an afforestation or reforestation activity under a CDM project is found to be a Net Source. Where this CDM Project has previously generated the issuance of lCERs through its net absorption of greenhouse gases, a Reversal of Storage would require replacement of lCERs equal to the quantity of the Reversal of Storage.
RMU	Removal units. These are tradable units generated on the basis of removals of greenhouse gases from the atmosphere through LULUCF activities under Articles 3.3 and 3.4 of the Kyoto Protocol. They may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
Robust	A characteristic of a software system that describes the extent to which it is protected from loss of service or data integrity.
Role	A role is a set of permissions for functions that a person is allowed to perform. A role may be assigned to a user (person) or a group.
Scalability	The ability of a software system to handle higher workload than initially planned without modifying the program code.
Secretariat	Secretariat to the UNFCCC.
SOAP	Simple Object Access Protocol.
Stage	The stage of a transaction or reconciliation defines where in the process of data exchange a particular message or evaluation occurs. A stage ends and a new stage begins when a message has been successfully transmitted and occurred by either a registry or the ITL or when the last step of a process occurs.
Status	The transaction and reconciliation status describe the current state of the review process. As each process moves through the defined stages, the status will be updated to reflect the result of the ITL (or an external registry) evaluation. A transaction status might move from proposed to checked (no discrepancy), and to final. A reconciliation status might change from initated to ITL validated.

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
Supplementary Program	An emissions tracking program for GHG emissions which operates as a complementary program to the Kyoto Protocol and shares communications with Kyoto Protocol participants through ITL Communications Hub.
Supplementary Transaction Log (STL)	An electronic database established for a Supplementary Program to monitor the validity of transactions between registries under that program.
tCER	Temporary Certified Emission Reduction unit. These are tradable units generated by Projects that enhance removals of greenhouse gases from the atmosphere in Non-Annex I Parties under the CDM. tCERs expire at the end of the Commitment Period subsequent to the Commitment Period within which they were issued. tCERs may be counted by Annex I Parties towards compliance with their emissions target and are equal to one tonne of carbon dioxide equivalent gases.
Termination	Termination is the action taken by a registry to end a proposed transaction which has been determined to be invalid, for which a discrepancy has been identified, for which the allowable response time has lapsed, or which it no longer wishes to process.
Transaction	The term transaction is used to describe a unique operation on a unit or block of units. A transaction is comprised of a series of actions related to a specific process. Each "transaction" is processed in stages and results in the return of a message to the registry identifying subsequent data on the transaction. A resubmission of the same information, following a transaction failure, is a new transaction.
True-up Period	The period from the end of the Commitment Period (2012) until 100 days after the completion of the Kyoto Protocol reviews of emissions information relating to the Commitment Period. Transfers of units may continue to take place during this period. The true-up period may therefore last until some time in 2015.
UML	Unified Modeling Language. Notation standard for describing software systems.
Unavailable Status	Units which are involved in transactions that have been proposed and received by the ITL, and are waiting for a response from either another registry (for an external transfer) or the proposing registry, are "unavailable" for other transfer. These units are flagged as unavailable. Similarly, a unit involved in an inconsistency is marked as unavailable until the inconsistency is resolved. An ICER generated by a CDM Project for which a reversal of storage has occurred, or for which a certification report has not been submitted, is also marked as unavailable.
UNFCCC	United Nations Framework Convention on Climate Change. This is the framework treaty to which the Kyoto Protocol is allied.
Unit	Generic term for AAUs, RMUs, ERUs, CERs, tCERs and ICERS.
Universal Time	Equivalent to Greenwich Mean Time (24-hour clock).

(cont.)

Figure A1: Glossary of Terms (cont.)

Term	Definition
User	A person (human being) who interacts with a system.
User Acceptance Test	A test performed by a user of the system against a set of predefined test cases.
User Interface	The interface used by a person to interact with an application.
Virus	A software program that harms software systems or other software programs.
VPN	Virtual Private Network.
Web Service	A Web service is a group of operations that perform communication tasks to and from a registry and the ITL. Separate Web services are defined for different processes.
WSDL	Web Service Description Language.
XML	Extensible Markup Language. Standard used for structured data storage.

[This page intentionally left blank.]

Annex B
Web Service Functions for Transaction Processing

1. Introduction

This annex contains a specification for the Web services and functions the registries must implement in order to establish communication with the ITL for transaction processing. Each Web service operation and suggested function is detailed in order to specify how the registry passes information to and from the ITL. Accordingly, the data elements that are passed into and out of a Function accessible through Web services are defined.

Some of these operations and functions are exposed to the public through Web services and others are functions that operate internally to the registry. In general, the input and output parameters have not been specified for functions that operate internally to a system. The design of the Web services public functions, including how they store and process data, may vary from registry to registry.

Only Web services and operations that are required for a registry are listed in this annex. For a complete list of the corresponding and interacting Web service operations and functions performed by the ITL, reference the ITL Technical Specifications document, Annex E.

2. Objects and Structures

The following sections explain the conventions and structured data storage utilised in the specifications for each function.

2.1 Result Identifier

Each function returns a Result Identifier. This value will indicate whether the function succeeded or failed. A failure could be the result of a business decision (a failed check) or the result of an unanticipated exception error (a run time error). The convention used here is that zero (0) indicates failure and one (1) indicates success.

2.2 Transaction Object

The Transaction Object is used to store all the elements involved in a transaction required for processing. The Transaction Object is a parent class to the TransactionUnitBlock object. The structure of the Transaction Object is as follows:

```
Structure TransactionObject
        TransactionIdentifier As String
        TransactionType As Integer
        SuppTransactionType As Integer (optional)
        TransferringRegistryIdentifier As String
        TransferringRegistryAccountType As Integer
        TransferringRegistryAccountIdentifier As Long (optional)
        AcquiringRegistryIdentifier As String
        AcquiringRegistryAccountType As Integer
        AcquiringRegistryAccountIdentifier As Long (optional)
        NotificationIdentifier As Integer (optional)
        TransactionBlocks (array) As TransactionUnitBlockObject
```

This object contains a SuppTransactionType element which is used by the Supplementary Transaction Log for the European Commission GHG Trading Program.

2.3 TransactionUnitBlock Object

The functions below reference a TransactionUnitBlock Object to obtain the unit blocks associated with a transaction. Often, an array of TransactionUnitBlock Objects is called for because more than one block can be associated with a transaction. The structure of the TransactionUnitBlock Object is as follows:

```
Structure TransactionUnitBlockObject
        UnitSerialBlockStart As Long
        UnitSerialBlockEnd As Long
        OriginatingRegistryIdentifier As String
        UnitType As Integer
        SuppUnitType As Integer (optional)
        OriginalCommitPeriod As Integer
        ApplicableCommitPeriod As Integer
        LULUCFActivity As Integer (optional)
        ProjectIdentifier As Integer (optional)
        Track As Integer (optional)
        BlockRole As String (optional)
        TransferringRegistryAccountType As Integer (optional)
        TransferringRegistryAccountIdentifier As Long (optional)
        AcquiringRegistryAccountType As Integer (optional)
        AcquiringRegistryAccountIdentifier As Long (optional)
        YearInCommitmentPeriod As Integer (optional)
        InstallationIdentifier As Long (optional)
        ExpiryDate As DateTime (optional)
```

This object contains a SuppUnitType, Year in Commitment Period, and Installation Identifiers which are used by the Supplementary Transaction Log for the European Commission's GHG trading program.

2.4 UnitBlock_Identity Object

The UnitBlock_Identity Object contains the attributes that uniquely identify a unit block and is generally returned as part of a notification to a Registry in conjunction with a CheckResponse Object.

```
Structure UnitBlock_IdentityObject
        UnitSerialBlockStart As Long
        UnitSerialBlockEnd As Long
        OriginatingRegistryIdentifier As String
```

2.5 Check Response Object

The CheckResponse Object is used to collect all checks that are applicable to a process. As each check is evaluated, the result identifier is modified to indicate a success or failure for the check. When the CheckResponse Object is returned, it only includes those response codes that have failed.

```
Structure CheckResponseObject
        ResultIdentifier As Integer
        ResponseCode As Integer
```

2.6 Evaluation Result Object

This object stores the specific CheckResponse Object along with the affected Unit Blocks.

```
Structure EvaluationResultObject
        CheckResponseObject
        UnitBlocks (array) As UnitBlock_Identity Object
```

3. Specified Functions

The following functions and Web service operations will be implemented to carry out the duties of a registry. They are listed in alphabetical order.

Figure B1: AcceptNotification (Web Service)

Purpose
This is the HTTP SOAP request that registries and the ITL use to send notifications regarding transactions in process. This Web service is hosted on both the ITL and the registry. See Annex K for specifications on the WSDL requirements.
Inputs
TransactionObject, EvaluationResultObject
Process
This Web service performs preliminary checks on the content of the XML message before processing.
Outputs
Result_Identifier
Recommended Functions
Data_Integrity_Check

Figure B2: AcceptProposal (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to receive proposals from registries. Registries also must implement this service to receive proposals for external transfers from the ITL.
Inputs
TransactionObject
Process
This Web service should perform preliminary checks on the content of the XML message before logging the message and writing contents of the message to a file.
Outputs
Result_Identifier
Recommended Functions
Data_Integrity_Check Write_To_File Write_To_Message_Log

Figure B3: Check_Version (Function)

Purpose
This function compares the major and minor version number in the HTTP SOAP request. These values are checked against the current version of the DES. Major version numbers must match or any forthcoming transactions to the ITL will be rejected. Messages with an outdated minor version are still processed, although a warning is issued. This is an optional function for registries.
Inputs
MajorVersion, MinorVersion
Process
Compare MajorVersion. If not a match, take corrective actions to update Registry system. This requires compliance with change management processes as defined on the ITL intranet website. Compare MinorVersion. If not a match, then be prepared to update system with minor patches or enhancements as recommended on the ITL intranet website.
Outputs
CheckResponseObject

Figure B4: Data_Integrity_Check (Function)

Purpose
This function will check the data in the TransactionObject to ensure that it meets the minimum requirements to begin processing the proposed transaction. This is an optional function for registries.
Inputs
TransactionObject
Process
For the following processes, ensure that the incoming message meets the following criteria. See Annex E for a list of appropriate response codes to return for data integrity errors.
Outputs
CheckResponseObject TransactionObject

Figure B5: Finalise_Transaction (Function)

Purpose
When a transaction is complete, this function is called to update the unit holding records of the registry.
Inputs
TransactionObject
Process
This function evaluates the transaction type to determine the data which must be updated. If the transaction type = 1 (Issuance) insert records representing unit blocks and accounts, as appropriate. If the transaction type = 2, 7 or 8 (Conversion, Carry-over, or Expiry Date Change), the units are free to be used in a trade. Registries will commit the change of unit type and project ID (for Conversion), the applicable Commitment Period (for Carry-over), or the expiry date (for Expiry Date Change). If the transaction type = 3, 4, 5 (External, Cancellation, Retirement) or (10 and the supplementary transaction types = 52, 53, 55, 01, 02, 41), the units are free to be used in a trade. Registries will commit the transfer ownership of the units to the acquiring Registry and accounts. If the transaction type = 6 (Replacement), the units are free to be used in a trade. Registries will commit the transfer to the replacement account and record the relationship between the two types of blocks.

Figure B6: Generate_Proposal (Function)

Purpose
This function creates the TransactionObject and corresponding TransactionUnitBlock for submission of a proposed transaction to the ITL. This function invokes AcceptProposal() on the ITL.
Inputs
Outputs
TransactionObject, TransactionUnitBlock
Call(s)
Write_Transaction Write_Transaction_Block Write_Transaction_Status

Figure B7: Preliminary_Checks (Function)

Purpose
The registry calls this function when a message is received through one of its Web services. This function will authenticate the sender of the message and check the version number. If the message contains a transaction, it will call another function to write the message to a file. If there are no problems, the message is ready to be processed.
Inputs
TransactionObject, ReconciliationObject
Process
If this message contains a transaction, call Write_To_File to record the contents of the message.
Outputs
Result_Identifier
Call(s)
Check_Version Write_To_File

Figure B8: Update_Units (Function)

Purpose
This function will record the transaction as final and update other records relating to the transaction, as appropriate.
Inputs
TransactionObject, TransactionUnitBlockObject
Process
If the transaction type is External (3), the transaction is recorded as final and the units are removed from the appropriate accounts. If transaction type is Carry-over (7), then the Applicable Commitment Period for the unit is updated to the current Commitment Period. If the transaction type is Conversion (2), and the unit was not previously (2) RMU, then the Unit Type is updated to (3) for ERU and the Project ID is updated. If the transaction type is Conversion (2), and the previous unit type was (2) RMU, the Unit Type is updated to (4) for ERU converted from RMU and the Project ID is updated. If the transaction type is Cancellation (4) or Retirement (5), then transaction is recorded as final and the units moved to the appropriate account(s). If the transaction type is Expiry Date Change (8), update the expiry date. If the transaction type is Internal/Supplementary (10-51), (10-52) or (10-55), or (4-3) then the Supplementary Unit Type is updated to the appropriate code for the new Supplementary Transaction Type Code.
Outputs
Result_Identifier

Figure B9: Validate_Proposal (Function)

Purpose
This function is used only by acquiring registries to validate external transfers. This function could call other functions on the registry to perform additional checks.
Inputs
TransactionObject, TransactionUnitBlockObject
Process
Incoming proposals are evaluated and processed. Data integrity checks should be applied to proposed transaction. Log and record the transaction, applicable blocks and the transaction status. If the proposal is accepted or rejected, log the new transaction status and call the ITL AcceptNotification Web service, submitting the TransactionObject and TransactionUnitBlockObject with the transaction status and any applicable response codes.
Calls
Data_Integrity_Check Write_To_Message_Log Write_Transaction Write_Transaction_Block Write_Transaction_Status AcceptNotification

Figure B10: Write_To_File (Function)

Purpose
This function will create a text file, write the contents of the HTTP SOAP Request to the file, then add the file to a master Zip file.
Inputs
Web_Service_URL, Transaction_Type, XML Message Content, File_name
Table(s)
This function does not interact with the database.
Process
Retrieve the to and from elements in the contents of the SOAP request. Retrieve the Registry Code and Transaction Identifier values. Generate the file name by concatenating these two values along with a random number. Write contents of XML body to text file and store in the Zip file. If the file fails to write to file, return HTTP SOAP response error.
Outputs
Result_Identifier
Call(s)

Figure B11: Write_To_Message_Log (Function)

Purpose
This function will append a record to the Message Log. The Message Log table records the history of all message exchange. The contents of any HTTP SOAP Request received involving a transaction are parsed and constructed as text files that are then stored in a larger Zip file. The Message Log tracks the time when the file was created, the name of the file and the name of the Zip file in which it has been compressed.
Inputs
Filename, Master_File_Name, Registry_Code, Reconciliation ID, Transaction ID, Web service
Table(s)
Message Log
Process
Append to Message_Log, Registry_Code, File_Name, Master_File_Name, SystemTime( ) Reconciliation ID or Transaction ID, Web service to Registry_Code, File_Name, File_Path, Submission_Date, Recon_ID, Transaction_ID, Web_Service.
Outputs
Result_Identifier
Call(s)

Figure B12: Write_Transaction (Function)

Purpose
This function will insert a record into the Transaction Log table.
Inputs
TransactionObject
Table(s)
Transaction Log
Process
Append to Transaction Log the elements in the TransactionObject.
Outputs
Result_Identifier
Call(s)
Write_Transaction_Block

Figure B13: Write_Transaction_Block (Function)

Purpose
This function will insert a record into the Transaction Block table.
Inputs
TransactionObject, TransactionUnitBlockObject
Table(s)
Transaction_Block
Process
For each instance of the TransactionUnitBlockObject array associated with the TransactionObject, insert a record into the Transaction Block table.
Outputs
Result_Identifier
Call(s)

Figure B14: Write_Transaction_Status (Function)

Purpose
This function will insert a record into the Transaction Log History table.
Inputs
TransactionObject
Table(s)
Transaction Log History
Process
Append a record to the Transaction Log History table with Transaction ID, system time stamp and current transaction status from TransactionObject.
Outputs
Transaction Status ID, unique identifier for new Transaction Status History record, Result_Identifier
Call(s)

Annex C
Web Services and Functions for Reconciliation

1. Introduction

This annex contains a specification for the Web services and functions the registries must implement in order to establish communication with the ITL for reconciliation. Each Web service operation and suggested function is detailed in order to specify how the registry passes information to and from the ITL. Accordingly, the data elements that are passed into and out of a function accessible through Web services are defined.

Some of these operations and functions are exposed to the public through Web services and others are functions that operate internally to the registry. In general, the input and output parameters have not been specified for functions that operate internally to a system. The design of the Web services public functions, including how they store and process data, may vary from registry to registry.

Only Web services and operations that are required for a registry are listed in this annex. For a complete list of the corresponding and interacting Web service operations and functions performed by the ITL, reference the ITL Technical Specifications document, Annex E.

2. Objects and Structures

The following sections explain the conventions and structured data storage utilised in the specifications for each function. Also consult the TransactionObject, TransactionUnitBlockObject, UnitBlockIdentityObject, and CheckResponseObject defined in Annex B.

2.1 Reconciliation Object

The Reconciliation Object is used to describe a reconciliation action and its current status.

```
Structure ReconciliationObject
        ReconciliationIdentifier As String
        ReconciliationBeginDate As DateTime
        ReconciliationEndDate As DateTime
        ReconciliationSnapshotDateTime As DateTime
        ReconciliationStatusCode As Integer
        ReconciliationStatusDateTime As DateTime
        ReconciliationStartPhaseCode As Integer
        ReconciliationCommitPeriod As Integer
        ReconciliationComment As String
        ReconciliationResponseCodes (Array) As Integer
```

2.2 Totals Object

The Totals Object is used by the reconciliation process to store the number of units held by a registry, account type, or account.

```
Structure TotalsObject
        ReconciliationIdentifer As String
        ReconciliationSnapshotDate As DateTime
        HoldingRegistry As String
        AccountType As Integer
        AccountCommitPeriod As Integer
        AccountIdentifier As Integer
        UnitType As Integer
        SupplementaryUnitType As Integer
        UnitCount As Integer
```

3. Functions and Web Services

Figure C1: Calculate_Totals (Function)

Purpose

This function will calculate the total number of units held at the registry in response to a request from the ITL. The totals will be calculated by querying the snapshot data that was stored at the agreed upon date and time for the reconciliation.

The ITL may request totals only for units held by a specific account type or unit type. If no specific requests are made, the totals should be calculated for each distinct combination of account type and unit type.

Inputs

ReconciliationObject, Holding Account Type, Unit Type

Process

If no limiting criteria were passed into the ProvideTotals service,

Calculate the number of units held grouped by account type, Account Commitment Period, and unit type. Below is an example of an SQL query to calculate the totals.

```
Select Account Type, Account Commitment Period, Unit Type,
Sum(end_block - start_block + 1) as Unit Totals
From Reconciliation Snapshot Data
Where Reconciliation ID = current reconciliation action ID
Group By Account Type, Account Commitment Period, and Unit Type
```

If limiting criteria were passed into the ProvideTotals service,

Calculate the number of units held by the specified account type, Account Commitment Period, or unit type. Below is an example of an SQL query to calculate these totals.

```
Select Account Type, Account Commitment Period, Unit Type,
Sum(end_block - start_block + 1)  as Unit Totals
From Reconciliation Snapshot Data
Where Reconciliation ID = current reconciliation ID
and Account Type = account type specified by ITL
and Unit Type = unit type specified by ITL
Group By Account Type, Account Commitment Period, and Unit Type
```

Store the totals returned in an array of the TotalObject, and call the ReceiveTotals Web service method on the ITL.

(cont.)

Figure C1: Calculate_Totals (Function) (cont.)

Outputs
TotalsObject array
Call(s)

Figure C2: Close_Reconciliation_Action (Function)

Purpose
This function will update the registry's Reconciliation Log with the end date of a reconciliation action.
Inputs
ReconciliationObject
Process
Update Reconciliation Log tables so that Reconciliation End Date = date and time when ITL confirms reconciliation compliance. This information is received from the ReceiveReconciliationResult Web service operation.
Outputs
Result_Identifier
Call(s)

Figure C3: Generate_Audit_Trail (Function)

Purpose

This function identifies past transactions involving unit blocks that are in question. In order to identify the source of an inconsistency, the ITL will request data for all the transactions that occurred within a specified time period for the unit blocks in question. This function will accumulate a list of transactions by querying the transaction log for the specified time period and send the audit trail to the ITL.

Inputs

ReconciliationObject, UnitBlock_IdentityObject, Begin Transaction Time, End Transaction Time

Process

Query the Transaction Log and the Transaction Block tables and return all available information for the specified units and time period. Below is an outline of an SQL query to generate the audit trail.

```
Select Transaction Number, Transaction Date, Transaction Type, Transferring Registry,
Acquiring Registry, Transferring Account, Acquiring Account
Originating Registry, Start Block, End Block, Unit Type, Applicable Commit Period, Original
Commit Period, Project ID, LULUCF Activity Code, Track ID, Expiry Date

From Transaction Log, Transaction Block
Where Transaction Date >= requested begin transaction time
and Transaction Date <= requested end transaction time
and Originating Registry = requested originating registry
and (       (Start Block >= requested start block and Start Block <= requested end block)
            or
            (End Block >= requested start block and End Block <= requested end block)
```

Store the audit trail to a TransactionObject array, and call the ReceiveAuditTrail Web service method on the ITL.

Outputs

UnitBlockObject array

Call(s)

Figure C4: Generate_Unit_Blocks (Function)

Purpose

This function identifies unit blocks requested by the ITL for comparison during reconciliation. This function will accumulate a list of unit blocks by querying the snapshot unit block data that was stored at the requested date and time for the reconciliation.

The ITL may request only the unit blocks of a specific Unit Type and/or held by a specific Account Type be sent. If no criteria are specified, all unit blocks should be sent to the ITL.

Inputs

ReconciliationObject, Holding Account Type, Unit Type

Process

```
If no limiting criteria were passed to the ProvideUnitBlocks service,
    Generate a complete list of unit blocks.  Below is an outline of an SQL query to generate
    the list of unit blocks.

        Select Account Type, Unit Type, Originating Registry Code, Applicable
        Commitment Period, Start Block, End Block
        From Reconciliation Snapshot Data
        Where Reconciliation ID = current reconciliation action ID

If limiting criteria were passed to the ProvideUnitBlocks service,
    Generate a list of unit blocks that meet the specified criteria.  Below is an outline of an SQL
    query to generate the list of unit blocks.

        Select Account Type, Unit Type, Originating Registry Code, Applicable
        Commitment Period, Start Block, End Block
        From Reconciliation Snapshot Data
        Where Reconciliation ID = current reconciliation action ID
        and Account Type = account type specified by ITL
        and Unit Type = unit type specified by ITL
        and Account Commit Period = Account Commit Period specified by ITL
```

Store the Unit Blocks identified in an array of the UnitBlockObject, and call the ReceiveUnitBlocks Web service method on the ITL.

Outputs

TransactionObject array

Call(s)

Figure C5: InitiateReconciliation (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to confirm an upcoming reconciliation action.
Inputs
ReconciliationObject
Process
This service receives the reconciliation identifier and snapshot date and time from the ITL. This time will have already been agreed upon by the ITL and registry administrators. Upon receipt of this message the registry should record the reconciliation identifier with a status of 0 ("Confirmed") and schedule the snapshot (if it has not already done so). There is no response to the ITL at this point. See Section 5 for detailed information on the reconciliation process.
Outputs
CheckResponseObject
Call(s)
Write_To_Reconciliation_Log Write_To_Reconciliation_Status

Figure C6: ProvideAuditTrail (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to request audit trail data from a registry.
Inputs
ReconciliationObject
Process
This function receives a request from the ITL to provide a transaction history for a specified set of units within a specified timeframe. The requested units may be explicitly listed, or they may be requested for a specified unit type or account type of the owner. If a cancellation or retirement account type is specified, a specific Commitment Period for the account may also be specified. Call Write_To_Reconciliation_Unit_Block to record as inconsistent any unit blocks that have been listed. If this request follows a request to provide unit blocks, call Write_To_Reconciliation_Status to record the new status as "Unit Blocks Inconsistent." If this is the first request for information by the ITL during this reconciliation action (i.e. the ITL skipped the request for totals and units blocks), update the Reconciliation Log to indicate the start phase for this action is Phase 3. Call Write_To_Reconciliation_Status to record the new status as "Initiated." See Section 5 for detailed information on the reconciliation process.
Outputs
CheckResponseObject
Call(s)
Write_To_Reconciliation_Status Write_To_Reconciliation_Unit_Block

Figure C7: ProvideTotals (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to initiate a request for total number of units held by each registry.
Inputs
ReconciliationObject
Process
This service receives a request for the total number of units held by each registry. It will be called by the ITL upon completion of the data snapshot for the reconciliation action. The request may be for a specified account type, unit type, or a combination of the two. If a cancellation or retirement account type is specified, a specific Commitment Period for the account may also be specified. If no limiting criteria are passed, the request is for all totals, grouped by account type and unit type. Call Write_To_Reconciliation_Log to record the start phase of this reconciliation action as Phase 1. Call Write_To_Reconciliation_Status to record the latest status as "Initiated." See Section 5 for detailed information on the reconciliation process.
Outputs
CheckResponseObject
Call(s)
Write_To_Reconciliation_Log Write_To_Reconciliation_Status

Figure C8: ProvideUnitBlocks (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to initiate a request for the unit blocks held at a registry.
Inputs
ReconciliationObject
Process
This service receives a request for the unit blocks held at a registry. The request may be for a specified account type, unit type, or a combination of the two. If a cancellation or retirement account type is specified, a specific Commitment Period for the account may also be specified. If no limiting criteria are passed, the request is for all unit blocks. If this request follows a prior request to provide total counts, call Write_To_Reconciliation_Status to record the new status as "Totals Inconsistent." If this is the first request for information during this reconciliation action (i.e. the ITL skipped the request for totals), update the Reconciliation Log to indicate the start phase for this action is Phase 2. Call Write_To_Reconciliation_Status to record the new status as "Initiated." See Section 5 for detailed information on the reconciliation process.
Outputs
CheckResponseObject
Call(s)
Write_To_Reconciliation_Status Write_To_Reconciliation_Log

Figure C9: ReceiveReconciliationResult (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to inform the registry that it has finished processing a reconciliation action.
Inputs
ReconciliationObject
Process
This Web service receives the result of the ITL's reconciliation analysis. Registries will record the results of the analysis in their Reconciliation Logs. The ITL will call this service when there is an abnormal end to the ITL reconciliation analysis. The reason for the abnormal end will be described by the array of response codes. The ITL will also call this service when the reconciliation action is successfully completed.
Outputs
CheckResponseObject
Call(s)
Close_Reconciliation_Action Write_To_Reconciliation_Log Write_To_Reconciliation_Status

Figure C10: Snapshot_Data (Function)

Purpose
This function will take a "snapshot" of the data at the agreed upon time. The reconciliation action will be based on this data snapshot.
Inputs
ReconciliationObject
Process
At the specified time, create a "snapshot" of the unit block data. The fields needed in the snapshot for each unit block include: • Holding Account Type Code • Holding Account Commitment Period • Unit Type Code • Originating Registry Code • Applicable Commitment Period • Start Block Number • End Block Number The Reconciliation ID should also be stored. These data should be preserved at least until the reconciliation action has closed.
Outputs
Result_Identifier
Call(s)

Figure C11: Write_To_Reconciliation_Log (Function)

Purpose
This function inserts a new record into the Reconciliation Log table.
Inputs
ReconciliationObject
Table(s)
Reconciliation Log
Process
Append to Reconciliation Log table as follows: • Recon_ID = input reconciliation ID • Recon_Action BeginDatetime = input reconciliation begin DateTime • Recon_Log_Comment = input comment • Recon_Action EndDateTime = input reconciliation end DateTime • Recon_Snapshot_DateTime = input reconciliation snapshot DateTime • Recon_Start Phase_Code = input reconciliation starting phase • Recon_CommitPeriod = input reconciliation Commitment Period
Outputs
Result_Identifier
Call(s)

Figure C12: Write_To_Reconciliation_Status (Function)

Purpose

This function inserts a new record into the Reconciliation_Status_History table.

Inputs

ReconciliationObject

Table(s)

Reconciliation Status History

Process

Append to Reconciliation Status History table as follows:

- Recon_ID = input reconciliation ID
- Recon_Status_Code = input reconciliation status code
- Recon_Comment = input comment
- Recon_Log_DateTime = system time

For each response code (if any) associated with this reconciliation status, append to the Reconciliation_Log_Response_Codes as follows:

- response code = input response code

Outputs

Result_Identifier
ReconciliationObject

Call(s)

Figure C13: Write_To_Reconciliation_Unit_Block (Function)

Purpose
This function will add a record to the Reconciliation_Unit_Block table to indicate that there is an inconsistency between the registry and the ITL concerning the ownership of that unit block.
Inputs
ReconciliationObject, UnitBlockIdentityObject
Table(s)
Reconciliation Status History
Process

For each Unit Block append to the Reconciliation_Unit_Block table as follows:
ReconciliationID = input reconciliation ID
ActionDateTime = system DateTime
OriginatingRegistry = input originating registry
UnitSerialBlockStart = input block start
UnitSerialBlockEnd = input block end
ResponseCode = input response code

Outputs
Result_Identifier
Call(s)

Annex D
Web Services for Administrative Processes

1. Introduction

This annex contains a specification for the Web services and functions the registries must implement in order to establish communication with the ITL for administrative processes. Each Web service operation is detailed in order to specify how the registry passes information to and from the ITL. The design of the Web services public functions, including how they store and process data, may vary from registry to registry.

Only Web services and operations that are required for a registry are listed in this annex. For a complete list of the corresponding and interacting Web service operations and functions performed by the ITL, reference the ITL Technical Specifications document, Annex E.

2. Objects and Structures

The Web services defined for administrative processes make extensive use of the TransactionObject and the TransactionUnitBlockObject. See Annex B for a definition of these objects.

Figure D1: AcceptITLNotice (Web Service)

Purpose
This HTTP SOAP service receives incoming messages forwarded from the ITL. This Web service may be used to receive notifications from the ITL. The ITL will call this Web service to inform the registry about issues identified by administrative and clean-up processes on the ITL. This Web service will be used by the following processes: 1. Inform the registry when cancellation of units is required as part of a net source cancellation action. 2. Inform the registry when cancellation of units is needed as a result of the Party being found by the Compliance Committee to be in non-compliance with its emissions target for the previous Commitment Period. 3. Inform the registry of units that are about to expire. 4. Inform the registry of the need to replace or cancel units due to a Reversal of Storage at a Project. 5. Inform the registry of lCERs that may not be traded due to a lack of certification report for the project associated with the lCERs. 6. Inform the registry that the CDM Registry issued too many CERs for a Project and that it will be necessary for the operational entity responsible for that Project to cancel or replace some units. 7. Inform the registry of changes in the Commitment Period Reserve (CPR). 8. Inform the registry of Outstanding Units at the end of the Commitment Period. 9. Inform the registry of the status of a prior notification.
Inputs
From, To, MajorVersion, MinorVersion, MessageContent, MessageDateTime, NotificationStatus, NotificationType, NotificationID, ProjectID, UnitType, TargetValue, ActionDueDate, TransactionUnitBlockObjectArray
Process
Upon receipt of an incoming message, the message will be logged and the system administrator notified.
Outputs
Result_Identifier
Call(s)

Figure D2: AcceptMessage (Web Service)

Purpose
This HTTP SOAP service receives incoming messages forwarded from the ITL. This Web service may be used to receive messages from the ITL or another registry. The Content input element is flexible and can contain a large amount of text. This allows the function to act as a generic service for registries to send messages to each other.
Inputs
From, To, MajorVersion, MinorVersion, MessageContent, MessageDateTime
Process
Upon receipt of an incoming text message from the AcceptMessage Web service, the system administrator shall be alerted of its arrival.
Outputs
Result_Identifier
Call(s)

Figure D3: GetTransactionStatus (Web Service)

Purpose
This is the HTTP SOAP request registries use to retrieve the current status of a transaction at the ITL.
Inputs
Transaction ID
Process
This function will query the ITL database and return the latest status of a specified transaction. It will also return the date and time the status was last updated.
Outputs
TransactionStatus TransactionStatusUpdateDateTime
Call(s)

Figure D4: ProvideTime (Web Service)

Purpose
This is the HTTP SOAP request that the ITL uses to request the current time from a registry. The request provides the ITL time as an informational service to assist in recognizing time synchronization problems.
Inputs
From, To, MajorVersion, MinorVersion, ITL SystemTime
Process
The ITL will periodically request a registry to return the current time in order to ensure that the system time is in synchronization with the ITL.
Outputs
SystemTime Result_Identifier ResponseCodes
Call(s)

[This page intentionally left blank.]

Annex E
List of Checks and Response Codes for Message Processing

1. Summary of Response Specifications

The ITL sends responses to registries to report the result of message processing. Responses can be “simple” with only two pieces of data to report, or “complex” containing large record sets of data. Simple responses, which check for a minimal version control test, merely send back an indication that the message was received successfully and that the message is queued for processing. Complex responses are messages that are sent after a transaction or reconciliation action has been evaluated by the ITL. If the transaction or the reconciliation action was unsuccessful, the ITL sends a message containing response codes providing the reason for the failure.

Each registry shall have a response lookup table and the capability to receive and import updates to this table to support change management specifications described in Section 8. Registries shall use programming techniques which facilitate updates to codes in these tables without programming changes whenever possible.

A registry may not change the number associated with a response or the meaning of the response description. Each registry may elect to translate the response description into a different language for display or reporting purposes only. A registry may request that a response be added to the system by submitting a written request to the ITL administrator (or through an alternative process).

Response codes are assigned to the following range of numbers by category.

Figure E1: Check Categories

Category	Response Code Range	Category Description	Action Upon Failure
Version and Authentication	1000 - 1299	Checks to authenticate sender and to validate version of DES during preliminary processing.	Message returned with response codes or HTTP Soap Error. Message not placed into message queue (unless only a minor version inconsistency is identified).
Message Viability	1300 - 1399	Checks to determine whether the message is viable when processed from the queue.	Message returned with response codes. Message not logged in the Transaction Log table.
Registry Validation	1500 - 1599	Checks to validate status of registry during queue processing.	Message returned with response codes. Message not logged in the Transaction Log table.
Transaction Data Integrity	2000 - 2999	Basic checks of data content including numeric ranges and validity of codes during queue processing.	Message returned with response codes. Message not logged in the Transaction Log table.
Message Sequence for Transaction Messages	3000 – 3999	Checks to validate message order and transaction status.	Message returned with response codes. Message not logged in the Transaction Log table.

(cont.)

Figure E1: Check Categories (cont.)

Category	Response Code Range	Category Description	Action Upon Failure
General Transaction Checks	4000 - 4999	Checks applicable to all transactions involving unit blocks. These checks are applicable to all transactions, except for Issuance.	Message returned with response codes and transaction status. Message logged in the Transaction Log table.
Transaction-specific Checks	5000 - 5899	Kyoto Protocol transaction checks specific to designated transaction types.	Message returned with response codes and transaction status. Message logged in the Transaction Log table.
Registry Messages	5900 - 5999	Response codes generated by registries.	Response codes sent with transactions to other parties.
Reconciliation Data Integrity	6200 - 6299	Basic checks for data content in reconciliation messages.	Message returned with response codes. Message not logged in Reconciliation Log table.
Reconciliation Message Sequence	6300 - 6399	Checks to validate message order and reconciliation status.	Message returned with response codes. Message not logged in ITL Reconciliation Log table.
Other Reconciliation Checks and Messages	6400 - 6500	Basic reconciliation checks.	Message returned with response codes and transaction status. Message logged in ITL Reconciliation Log table.

2. Message Validity Response Codes

The following responses are used in checks for well-formedness, DES version compatibility, and registry validation. These response codes apply to all data exchange processes.

Figure E2: Responses

Response Code	Check Name	Check Category	Check Description	Transaction Type
SOAP error	Certificate Check	Version and Authentication	Certificate must be recognized.	All
SOAP error	SOAP Identifier	Version and Authentication	Initiating Registry must be consistent with sender of SOAP message.	All
SOAP error	WSDL Check	Version and Authentication	Message must conform to WSDL.	All
1031	Major Version	Version and Authentication	Major Version number in transaction message must match current Major Version number for DES.	All

(cont.)

Figure E2: Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
1032	Minor Version	Version and Authentication	Minor Version number in transaction message should match current Minor Version number for DES.	All
1301	Message Age	Message Viability	Message must be processed within 24 hours of submission.	All
1501	Initiating Registry	Registry	Initiating Registry must be listed in Registry table.	All
1503	Initiating Registry Transactions Status	Registry	Initiating Registry status must allow transactions to be proposed.	All
1504	Acquiring Registry Transactions Status	Registry	Acquiring Registry status must allow transactions to be accepted.	All
1510	Registry Reconciliation Status	Registry	Registry status must allow reconciliation actions to be conducted.	All

3. ITL Transaction Response Codes

The following responses are used by the ITL to support transaction processing.

Figure E3: ITL Transaction Responses

Response Code	Check Name	Check Category	Check Description	Transaction Type
2001	Transaction Mask	Data Integrity	Transaction ID must be comprised of a valid registry code followed by numeric values.	All
2002	Transaction Type Code	Data Integrity	Transaction type Code must be valid.	All
2003	Supplementary Transaction Type Code	Data Integrity	Supplementary Transaction Type Code must be valid.	All
2004	Transaction Status Code	Data Integrity	Transaction status code must be valid.	All
2005	Transaction Status DateTime	Data Integrity	Transaction Status DateTime must be before the current DateTime and no older than two weeks.	All
2006	Account Type Code	Data Integrity	Account Type Code must be valid.	All

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
2007	Initiating Account Identifier	Data Integrity	Initiating Account Identifier must be greater than zero.	All
2008	Acquiring Account Identifier	Data Integrity	Acquiring Account Identifier must be greater than zero.	All
2009	Notification Type Code	Data Integrity	Notification Type Code must be valid.	All
2010	Originating Registry	Data Integrity	The Originating Registry of all unit blocks must be valid.	All
2011	Unit Type Code	Data Integrity	Unit Type Code must be valid.	All
2012	Supplementary Unit Type Code	Data Integrity	Supplementary Unit Type Code must be valid.	All
2013	Unit Serial Block	Data Integrity	Unit Serial block start and Unit Serial block end must be present.	All
2014	Unit Serial Range	Data Integrity	Unit Serial block end must be greater than or equal to the Unit Serial block start.	All
2015	LULUCF Activity Code	Data Integrity	RMUs, ERUs converted from RMUs, tCERs and lCERs must have a valid LULUCF activity code.	All
2016	No LULUCF Activity Code	Data Integrity	AAUs, ERUs converted from AAUs and CERs must not have a LULUCF activity code.	All
2017	Project ID	Data Integrity	ERUs, CERs, tCERs, and lCERs must have a valid Project ID.	All
2018	No Project ID	Data Integrity	AAUs or RMUs must not have a Project ID.	All
2019	ERU Track Code	Data Integrity	ERUs must have a valid track code.	All
2020	No Track Code	Data Integrity	AAUs, RMUs, CERs, tCERs and lCERs must not have a track code.	All
2021	Expiry Date	Data Integrity	tCERs and lCERs must have an Expiry Date.	All
2022	No Expiry Date	Data Integrity	AAUs, RMUs, ERUs and CERs must not have an Expiry Date.	All
3001	Transaction ID Not Unique	Sequence	Transaction ID for proposed transactions must not already exist in the ITL.	All

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
3002	Prior Record of Transaction ID from Registry	Sequence	Transaction ID for ongoing transactions must already exist in the ITL."	All
3003	Transaction Status Out of Sequence for Prior Completed Status	Sequence	Previous completed transactions cannot be completed again."	All
3004	Transaction Status Out of Sequence for Prior Rejected Status	Sequence	Previously rejected transactions cannot be completed.	All
3005	Transaction Status Out of Sequence for Prior ITL Discrepancy Status	Sequence	Transactions for which an ITL discrepancy has been previously identified cannot be completed.	All
3006	Transaction Status Out of Sequence for Prior STL Discrepancy Status	Sequence	Transactions for which an STL discrepancy has been previously identified cannot be completed.	All
3007	Transaction Status Out of Sequence for Prior Terminated Status	Sequence	Previously terminated transactions cannot be completed."	All
3008	Transaction Status Out of Sequence for Prior Cancelled Status	Sequence	Previously cancelled transactions cannot be completed.	All
3009	Transaction Status Out of Sequence for Prior Accepted Status	Sequence	Previously accepted external transactions cannot be terminated.	External
3010	Transaction Status Out of Sequence for Accepted or Rejected Status	Sequence	Transaction status of Accepted or Rejected is not valid for non-external transactions.	All
3011	Transaction Status Not Compatible with Initiating Registry	Sequence	Transaction status from Initiating Registry must indicate status of Proposed, Completed, or Terminated.	All
3012	Transaction Status Not Compatible with Acquiring Registry	Sequence	Transaction status from Acquiring Registry must indicate status of Rejected or Accepted.	External

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
3501	Transaction Status Not Compatible with an STL	Sequence	Transaction status from STL must indicate status of Discrepancy or No Discrepancy.	All
3502	Prior record of Transaction ID from STL	Sequence	Transaction ID for ongoing transactions must exist in ITL.	All
4001	Applicable Commitment Period	General Transaction	Applicable Commitment Period must correspond to the current or next Commitment Period (including their true-up periods).	All
4002	Prior Record of Units	General Transaction	Units identified in the transaction must already exist in the ITL.	All
4003	Registry Holds Units	General Transaction	Units identified in the transaction must be held by Initiating Registry.	All
4004	Unit Block Attributes	General Transaction	All attributes of all unit blocks must be consistent with ITL unit block attributes except where attributes are changed by the current transaction.	All
4005	Single Applicable Commitment Period	General Transaction	All unit blocks in transaction must be for a single Applicable Commitment Period.	All
4006	Acquiring and Transferring Registry Consistency	General Transaction	For all transactions except for external transfers, the Initiating and Acquiring Registries must be the same.	All
4007	Acquiring and Transferring Registries for External Transactions	General Transaction	For external transfers, the Initiating and Acquiring Registries must be different.	All
4008	Units Have ITL Inconsistencies	General Transaction	Units identified in the transaction must not have inconsistencies identified through reconciliation with the ITL.	All
4009	Units Have STL Inconsistencies	General Transaction	Units identified in the transaction must not have inconsistencies identified through reconciliation with an STL.	All
4010	Units are Unavailable	General Transaction	Units identified in the transaction must not be involved in another transaction.	All
4011	Units are Cancelled	General Transaction	Cancelled units must not be subject to further transactions.	All

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
4012	Units are Retired	General Transaction	Retired units must not be subject to further transactions.	All
4013	Units are Expired	General Transaction	Expired tCERs and lCERs must not be subject to further transactions, except internal transfers to a Type 5 cancellation account.	All
4014	Units Previously Used in Replacement	General Transaction	Units previously used to replace tCERs or lCERs must not be subject to further transactions.	All
4015	lCER Transaction Ineligibility	General Transaction	lCERs must not be transferred to a holding or retirement account where the CDM Executive Board has notified a replacement requirement for the associated Project.	All
5001	National Registry Issuance	Transaction-specific	AAUs and RMUs must be issued by a national registry.	Issuance
5002	No ERU Issuance	Transaction-specific	ERUs cannot be issued.	Issuance
5003	CDM Registry Issuance	Transaction-specific	CERs, tCERs and lCERs must be issued by the CDM Registry.	Issuance
5004	Single Issuance Unit Type	Transaction-specific	A transaction must not issue more than one Unit Type.	Issuance
5005	Single Issuance Commitment Period	Transaction-specific	The Original Commitment Period must be the same for all units issued by the transaction.	Issuance
5006	Consistent Applicable Commitment Period	Transaction-specific	The Applicable Commitment Period must be the same as the Original Commitment Period for all units issued by the transaction.	Issuance
5007	Issued Serial Numbers	Transaction-specific	Serial numbers for proposed issuance must not already exist in the ITL.	Issuance
5008	AAU Issuance Quantity	Transaction-specific	The quantity of AAUs issued must not exceed allowed quantity for the Commitment Period.	Issuance
5009	RMU Issuance Quantity	Transaction-specific	The quantity of RMUs issued must not exceed allowed quantity for each LULUCF Activity Type and Commitment Period.	Issuance

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5010	CDM Issuance Unit Type	Transaction-specific	The type of units to be issued for each CDM Project must be consistent with the Project activity.	Issuance
5011	Consistency of Unit Type Issued for a LULUCF CDM Project	Transaction-specific	Choice of unit type must be consistent with previous issuance of tCERs and lCERs for the Project.	Issuance
5012	CDM Issuance Quantity	Transaction-specific	CER, tCER or lCER issuance for each CDM Project must not exceed quantity specified by CDM Executive Board.	Issuance
5013	CDM LULUCF Activity Code	Transaction-specific	The LULUCF Activity Code of CERs, tCERs or lCERs proposed for issuance must be consistent with the project activity.	Issuance
5014	CDM Project ID	Transaction-specific	A valid CDM Project ID must be present for the issuance of all CERs, tCERs and lCERs.	Issuance
5015	tCER Expiry Date	Transaction-specific	Expiry Date for tCERs must be consistent with the end date of the Commitment Period subsequent to the Original Commitment Period of the tCER.	Issuance
5016	lCER Expiry Date	Transaction-specific	Expiry date for lCERs must be consistent with the End Date of the Crediting Period for the Project specified by the CDM Executive Board.	Issuance
5051	National Registry Conversion	Transaction-specific	The Initiating Registry converting AAUs or RMUs must be a national registry.	Conversion
5052	Holding Account Conversion	Transaction-specific	The Initiating Account for a conversion transaction must be a holding account.	Conversion
5053	Conversion Eligibility (Track 1)	Transaction-specific	If the unit is a Track 1 ERU, the Party of the Initiating Registry must be determined to meet eligibility criteria 1 through 6.	Conversion
5054	Conversion Eligibility (Track 2)	Transaction-specific	If the unit is a Track 2 ERU, the Party of the Initiating Registry must be determined to meet eligibility criteria 1, 2 and 4.	Conversion
5056	Conversion Unit Type	Transaction-specific	Units for conversion must be AAUs or RMUs.	Conversion

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5057	Single Conversion Unit Type	Transaction-specific	A transaction must not convert more than one unit type.	Conversion
5058	Conversion by Issuing Registry	Transaction-specific	Units for conversion must have been issued by Initiating Registry.	Conversion
5059	Project ID	Transaction-specific	A valid JI Project ID must be present for the conversion of all ERUs.	Conversion
5060	JI Conversion Unit Type	Transaction-specific	The type of units to be converted to ERUs for each JI Project must be consistent with Project activity.	Conversion
5061	Track 2 ERU Conversion Quantity	Transaction-specific	Track 2 ERU Conversion for each Track 2 JI Project must not exceed the quantity specified by the Article 6 Supervisory Committee.	Conversion
5101	General Transferring Registry Eligibility for external transfers	Transaction-specific	The Party of an initiating national registry must be determined to meet eligibility criteria 1 through 6, except for the first external transfer of a track 2 ERU which the Registry has converted.	External
5102	ERU Track 2 Transferring Registry Eligibility for External Transfers	Transaction-specific	If the transaction is the first external transfer of a track 2 ERU which the Registry has converted, the Party of the initiating national registry must be determined to meet eligibility criteria 1, 2 and 4.	External
5103	Acquiring Registry Eligibility for External Transfers	Transaction-specific	The Party of an acquiring national registry must be determined to meet eligibility criteria 1 through 6, except for transfers initiated by the CDM Registry.	External

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5104	Commitment Period Reserve	Transaction-specific	The total quantity of all units held in a national registry, which may be used for compliance for the applicable Commitment Period of a transaction, must not fall below the CPR level for the Party for that Commitment Period, except where the transaction is a first transfer of Track 2 ERUs converted by the registry. This total quantity is the total of all units in holding and retirement accounts, less first external transfers of track 2 ERUs converted by the registry, expired units remaining in holding and retirement accounts, and required cancellations and replacements which have not been carried out 30 days after the relevant notification was sent by the ITL.	External
5105	External Transfers to CDM Registry	Transaction-specific	CDM Registry can only receive external transfers to Cancellation accounts for compensating excess issuance of CERs, tCERs and lCERs.	External
5106	Suspension from making external transfers	Transaction-specific	The Party of an initiating national registry must not have been suspended from making external transfers as a result of not meeting its emission target for the previous Commitment Period.	External
5151	National Registry Cancellation	Transaction-specific	Cancellation to Net Source, Non-Compliance and Voluntary Cancellation Accounts must take place in a national registry.	Cancellation
5152	No Excess Issuance Cancellation	Transaction-specific	Cancellation to Excess Issuance Cancellation Account must not take place in a national registry.	Cancellation
5153	Cancellation Accounts	Transaction-specific	The Acquiring Account for a cancellation transaction must be a cancellation account.	Cancellation
5154	Cancellation Account Identifier	Transaction-specific	Account identifiers must be provided for acquiring accounts in cancellation transactions.	Cancellation
5155	Cancellation Account Commitment Period	Transaction-specific	The unit blocks retired must have the same Applicable Commitment Period as the Cancellation Account.	Cancellation

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5156	tCER and lCER Cancellation to Net Source and Non-compliance Cancellation Accounts	Transaction-specific	tCERs and lCERs cannot be transferred to Net Source Cancellation Accounts or Non-compliance Cancellation Accounts.	Cancellation
5157	Notification ID for tCER and lCER Cancellations to Excess Issuance Cancellation Accounts	Transaction-specific	tCERs and lCERs may only be transferred to Excess Issuance Cancellation Account in the CDM registry in the case that excess tCER and lCER issuance is being compensated pursuant to a Excess Issuance Notification.	Cancellation
5158	Notification ID for Net Source Cancellations	Transaction-specific	Units may only be transferred to a net source cancellation account if a notification has been received from the ITL and this ID is reported in the transaction.	Cancellation
5159	Notification ID for Non-compliance Cancellations	Transaction-specific	Units may only be transferred to a non-compliance cancellation account if a notification has been received from the ITL and this ID is reported in the transaction.	Cancellation
5201	National Registry Replacement	Transaction-specific	The Initiating Registry replacing units must be a national registry.	Replacement
5202	tCER Replacement Accounts	Transaction-specific	The Acquiring Account for a replacement transaction involving tCERs must be a tCER replacement account.	Replacement
5203	lCER Replacement Accounts	Transaction-specific	The Acquiring Account for a replacement transaction involving lCERs must be an lCER replacement account.	Replacement
5204	Replacement Account Identifier	Transaction-specific	Account identifiers must be provided for acquiring accounts in replacement transactions.	Replacement
5205	Replacement Account Commitment Period	Transaction-specific	The Unit Blocks used for replacement must have the same Applicable Commitment as the Replacement Account.	Replacement
5206	Unit Type to be Replaced	Transaction-specific	Units to be replaced must be tCERs or lCERs.	Replacement
5207	Multiple Replacement	Transaction-specific	A unit may be replaced only once.	Replacement

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5208	Single Replacement Registry	Transaction-specific	The registry holding the units to be replaced and the replacing units must be the same.	Replacement
5209	Quantity of Replacement Units	Transaction-specific	The quantity of units replaced must equal the quantity of replacing units.	Replacement
5210	One-To-Many Replacement Units	Transaction-specific	A transaction cannot contain many-to-many relationships between replaced and replacing blocks.	Replacement
5211	Location of Replaced tCERs	Transaction-specific	tCERs to be replaced must be held in a Retirement account or a tCER Replacement account.	Replacement
5212	Location of Replaced lCERs	Transaction-specific	lCERs to be replaced must not be held in Cancellation accounts.	Replacement
5213	lCER Replacement Units (upon Expiry)	Transaction-specific	lCER Replacement accounts (upon expiry) cannot acquire tCERs or lCERs.	Replacement
5214	tCER Replacement Units (upon Expiry)	Transaction-specific	tCER replacement accounts (for unit expiry) cannot acquire lCERs.	Replacement
5215	lCER Replacement Units (upon Reversal of Storage or Lack of Certification Report)	Transaction-specific	lCER Replacement accounts (for Reversal in Storage) may not acquire tCERs and may not acquire lCERs with a Project Identifier other than that specified in the replacement notification.	Replacement
5216	Replacement Notification upon tCER Expiry	Transaction-specific	If provided, the Replacement Notification ID must be valid and must be for replacement upon tCER expiry.	Replacement
5217	Replacement Notification upon lCER Expiry	Transaction-specific	If provided, the Replacement Notification ID must be valid and must be for replacement upon lCER expiry.	Replacement
5218	Replacement Notification for Reversal in Storage	Transaction-specific	A valid Replacement Notification ID must be provided for replacement upon reversal in storage.	Replacement
5219	Replacement Notification for Lack of Certification Report	Transaction-specific	A valid Replacement Notification ID must be provided for replacement upon a lack of Certification Report.	Replacement

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5220	Project ID for ICERs Replacement (upon Reversal of Storage or lack of Certification Report)	Transaction-specific	For ICER replacement transactions upon Reversal of Storage or lack of a Certification Report, the Project ID for the ICERs to be replaced must be consistent with the Project ID contained in the replacement notification.	Replacement
5251	National Registry Retirement	Transaction-specific	The Initiating Registry retiring units must be a national registry.	Retirement
5252	Retirement Account	Transaction-specific	The Acquiring Account for a retirement transaction must be a retirement account.	Retirement
5253	Retirement Account Identifier	Transaction-specific	Account identifiers must be provided for acquiring accounts in retirement transactions.	Retirement
5254	Retirement Account Commitment Period	Transaction-specific	The Unit Blocks retired must have the same Applicable Commitment as the Retirement Account.	Retirement
5255	CER, tCER and lCER Retirement Eligibility	Transaction-specific	The Party of the Initiating Registry must be determined to meet eligibility criteria 1 through 6.	Retirement
5256	tCER and lCER Retirement Limit	Transaction-specific	tCER and lCER retirement must not exceed allowed quantity.	Retirement
5301	National Registy Carry-over	Transaction-specific	The Initiating Registry carrying over units must be a national registry.	Carry-over
5302	Holding Account Carry-Over	Transaction-specific	The Initiating Account for a carry-over transaction must be a holding account.	Carry-over
5303	Subsequent Commitment Period	Transaction-specific	Units may be carried-over only to the next subsequent commitment period.	Carry-over
5304	Units Available for Carry-over	Transaction-specific	The quantity of units of each unit type carried-over must not exceed the limit of carry-over established by the Compliance Committee for the Party and reported to the registry in the unit carry-over notification.	Carry-over
5305	RMU Carry-over	Transaction-specific	RMUs may not be carried over.	Carry-over
5306	ERU (from RMUs) Carry-over	Transaction-specific	ERUs converted from RMUs may not be carried over	Carry-over

(cont.)

Figure E3: ITL Transaction Responses (cont.)

Response Code	Check Name	Check Category	Check Description	Transaction Type
5307	lCER or tCER Carry-over	Transaction-specific	tCERs or lCERs may not be carried over.	Carry-over
5310	Notification ID for Carry-over	Transaction-specific	Units may only be carried over if a notification has been received from the ITL and this ID is reported in the transaction.	Carry-over
5311	Carry-over Unit Type	Transaction-specific	Unit blocks carried over must be consistent with the unit type specified in the unit carry-over notification.	Carry-over
5450	Units for Expiry Date Change	Transaction-specific	The units for Expiry Date Change must be tCERs or lCERs.	Expiry Date Change
5451	New tCER Expiry Date	Transaction-specific	The new tCER Expiry Date must be consistent with the End Date of the Commitment Period subsequent to the Original Commitment Period of the tCER.	Expiry Date Change
5452	New lCER Expiry Date	Transaction-specific	The new lCER Expiry Date is not consistent with the End Date of the renewed Crediting Period for the Project specified by the CDM Executive Board.	Expiry Date Change

4. Registry Transaction Response Codes

The following codes are generated by the recipient of an External Transfer.

Figure E4: Registry Messages

Response Code	Response Description
5902	Acquiring account does not exist.
5903	Acquiring account is not eligible to receive units.
5904	Transaction inconsistent with Party policy.
5905	Transaction rejected by account holder.
5906	Account has been closed.

5. ITL Reconciliation Response Codes

The following codes may be generated by the ITL when processing reconciliation messages.

Figure E5: ITL Reconciliation Responses

Response Code	Check Name	Check Description
6201	Reconciliation Identifier	Reconciliation Identifier must be greater than zero.
6202	Reconciliation Mask	Reconciliation ID must be comprised of a valid registry code followed by numeric values.
6203	Reconciliation Status Validity	Reconciliation status must be a value between 1 and 11.
6204	Reconciliation Snapshot DateTime	Reconciliation snapshot must be a date between 01-OCT-2004 and the current date plus 30 days.
6205	Account Type Validity	Account Type must be valid.
6206	Unit Type Validity	Unit Type Code must be valid.
6207	Supplementary Unit Type Validity	Supplementary Unit Type Code must be valid.
6301	Reconciliation ID Does Not Exist	Reconciliation ID must exist in the Reconciliation Log table.
6302	Reconciliation Status Not Valid	Out of Sequence reconciliation status sent by registry is invalid.
6303	Reconciliation Status Out of Sequence	Incoming reconciliation status must be the same as the reconciliation status recorded by the ITL.
6304	Consistent Reconciliation Snapshot DateTime	The registry reconciliation snapshot DateTime must be consistent with the ITL Reconciliation Snaphot DateTime.
6311	Reconciliation ID Sent by STL Does Not Exist	Reconciliation ID sent by the STL must already exist in the ITL unless the STL is requesting the ITL to initiate a new reconciliation action.
6312	Reconciliation Status Not Valid	Reconciliation status sent by the STL must be one of certain enumerated statuses.
6313	Reconciliation Status of "STL Totals Inconsistent" is Out of Sequence	If the incoming reconciliation status is "STL Totals Inconsistent," the previously recorded status at the ITL must be "Validated."
6314	Reconciliation Status of "STL Unit Blocks Inconsistent" Out of Sequence	If the incoming reconciliation status is "STL Unit Blocks Inconsistent", the previously recorded status at the ITL must be "STL Totals Inconsistent."
6315	Reconciliation status of "STL Validated" is out of sequence.	If the incoming reconciliation status is "STL Validated," the previously recorded status at the ITL must be "Validated," "STL Totals Inconsistent," or "STL Unit Blocks Inconsistent."
6316	Reconciliation Status of "STL Complete with Manual Intervention" is Out of Sequence	If the incoming reconciliation status is "STL Complete with Manual Intervention," the previously recorded status at the ITL must be "STL Totals Inconsistent," or "STL Unit Blocks Inconsistent."
6410	Account444 Type/Unit Type Totals	The totals for account types, commitment period, and unit types must be consistent.

(cont.)

Figure E5: ITL Reconciliation Responses (cont.)

Response Code	Check Name	Check Description
6420	Account Type/Unit Type Unit Blocks	The registry and ITL unit blocks for a specified account type commitment period and unit type must be consistent.
6430	Account Type/Unit Type Unit Blocks Unexpected Consistency	If the totals have failed in the previous stage, the Unit Block compare by account type, commitment period, and unit type must also fail.
6440	Snapshot DateTime Validity	The DateTime for reconciliation action proposed by the STL must be in the future.
6450	Ongoing Reconciliation	A reconciliation action cannot be initiated at the registry because there is already an ongoing action.
6600	Successful Reconciliation of Totals	The reconciliation has been completed with a successful reconciliation of unit totals.

Annex F
Definition of Identifiers

1. Introduction

This annex provides information on the required structure of identification numbers for the core entities associated with data exchange.

2. General Rules for XML Formats

Identifiers shall be transmitted in XML format as an element tag comprised of attributes (components of the identifier). For numeric elements, leading zeros should not be included as place holders for a number shorter than the maximum length. For components which do not apply (such as project ID for an AAU), an attribute is not required.

3. Recommended Display and Report Formats

For display and reporting, it is recommended that each element of the identifier be separated by a single dash "-" and no spaces. For example, a transaction number would be represented as follows: NZ-132-1. Leading zeros would not be displayed. Please note that the separating dash is not included in the XML structure for identifiers and should not be stored data. Consistent with the requirements below, all serial numbers shall be stored as elements.

4. Serial Numbers

The serial number of a unit shall be unique throughout all registries and the ITL.

Serial numbers are defined only by registries.

Whenever possible, a set of units shall be transmitted as a unit block defined by the starting block number and the ending block number. Within a unit block, every element of the serial number must be identical except for the unique number element. Number elements within a block must be complete and consecutive.

When necessary to perform a transaction, track, record, or otherwise characterize a unit or unit block, registries or the ITL shall create multiple unit blocks from a single unit block.

Although each unit is identified uniquely by its originating registry code and its assigned unique number and this identification should be used by registries, communication about unit(s) will conform to the Unit Block definition requirements below. For a single unit the start and end block elements contain the same value.

Figure F1: Serial Number Identifiers

Identifier	Display Order	Identifier Required for the Following Unit Types	Data Type	Length	Range or Codes
Originating Registry	1	All	Alphanumeric	3	Two-letter country codes in ISO3166, as of 01 January 2005
Unit Type	2	All	Numeric	2	1 = AAU 2 = RMU 3 = ERU converted from AAU 4 = ERU converted from RMU 5 = CER 6 = tCER 7 = lCER
Supplement-ary Unit Type	3		Numeric	2	Blank for Kyoto-only Units, or as defined by STL
Unit Serial Block Start	4	All	Numeric	15	Unique numeric values assigned by registry from 1 – 999,999,999,999,999
Unit Serial Block End	5	All	Numeric	15	Unique numeric values assigned by registry from 1 – 999,999,999,999,999
Original Commitment Period	6	All	Numeric	2	1 – 99
Applicable Commitment Period	7	All	Numeric	2	1 – 99
LULUCF Activity	8	RMU, ERU (converted from an RMU), tCER, lCER	Numeric	3	1 = Afforestation and reforestation 2 = Deforestation 3 = Forest management 4 = Cropland management 5 = Grazing land management 6 = Revegetation

(cont.)

Figure F1: Serial Number Identifiers (cont.)

Identifier	Display Order	Identifier Required for the Following Unit Types	Data Type	Length	Range or Codes
Project Identifier	9	ERU, CER, ICER, tCER	Numeric	7	Unique numeric value assigned by registry for Project
Track	10	ERU	Numeric	2	1 or 2
Expiry Date	11	ICER, tCER	Date		Expiry Date for tCERs or ICERs

5. Account Numbers

The account number shall be unique throughout all registries.

An account number for an account that is deactivated or deleted cannot be reused.

Account numbers are created and defined only by registries.

Holding accounts do not have an applicable Commitment Period.

Figure F2: Account Number Identifiers

Element	Display	Data Type	Length	Range or Codes
Registry Identifier	1	Alphanumeric	3	Two-letter country codes in ISO3166, as of 01 January, 2005, or CDM for pending and cancellation accounts in the CDM Registry
Account Type	2	Numeric	3	100 = Holding Account 110 = Pending Account 120 = Operator Holding Account 121 = Person Holding Account 210 = Net Source Cancellation Account 220 = Non-compliance Cancellation Account 230 = Voluntary Cancellation Account 240 = Excess Issuance Cancellation Account 250 = Mandatory Cancellation 300 = Retirement Account 411 = tCER Replacement Account for Expiry 421 = lCER Replacement Account for Expiry 422 = lCER Replacement Account for Reversal in Storage 423 = lCER Replacement Account for Non-submission of Certification Report
Account Identifier	3	Numeric	15	Unique numeric values assigned by registry from 1 – 999,999,999,999,999
Applicable Commitment Period	4	Numeric	2	0 for all holding accounts 1 – 99 for retirement and cancellation accounts

6. Transaction Numbers

The transaction number shall be unique within the registries and within the ITL.

A transaction number for a transaction that is terminated or cancelled cannot be reused. Resubmission of a transfer for which a transaction has been terminated or cancelled shall be assigned a new, unique transaction number.

Transaction numbers are defined only by registries.

Figure F3: Transaction Numbers

Identifier	Display	Data Type	Length	Range or Codes
Registry Identifier	1	Alphanumeric	3	Two-letter country codes in ISO3166, as of 01 January, 2005, or CDM for transfers from the pending account in the CDM Registry.
Transaction Identifier	2	Numeric	15	Unique numeric values assigned by registry from 1 – 999,999,999,999,999

7. Reconciliation Numbers

The reconciliation number shall be unique throughout all registries and the ITL.

Reconciliation numbers are defined only by the ITL at the that a reconciliation is confirmed by the ITL for a registry. The reconciliation ends when the ITL determines there are no discrepancies or when a manual intervention to correct inconsistencies is complete. If a resubmission of information is needed because of a data format problem, the same reconciliation number shall be used.

Figure F4: Reconciliation Numbers

Identifier	Display Order	Data Type	Length	Range or Codes
Registry Identifier	1	Alphanumeric	3	Two-letter country codes in ISO3166, as of 01 January, 2005 and CDM for CDM Registry reconciliation
Reconciliation Identifier	2	Numeric	15	Unique numeric values assigned by Transaction Log from 1 – 999,999,999,999,999

8. Project Numbers

The Project number shall be unique.

Project numbers are defined only by registries or the CDM Executive Board (in cooperation with the CDM Registry Administrator).

Figure F5: Project Numbers

Identifier	Display Order	Data Type	Length	Range or Codes
Party Identifier	1	Alphanumeric	3	Two-letter country codes in ISO3166, as of 01 January, 2005
Project Identifier	2	Numeric	7	Unique numeric values assigned by registry for Project

9. Notification Numbers

The Notification number shall be unique.

Notification numbers are defined only by the ITL.

Notification numbers are not registry or party specific and are numeric values, maximum length 12.

Annex G
List of Codes

Annex G defines the codes for all support tables used in these technical specifications. These codes define the acceptable values for elements defined in Annex F or otherwise required for data exchange.

Figure G1: List of Tables

Table	Description
Account Type Code	Identifies the type of unit accounts in a registry.
Commitment Period Code	Identifies the Commitment Period.
LULUCF Activity Code	Identifies Land Use, Land Use Change, and Forestry categories.
Notification Status Code	Identifies status of notification request.
Notification Type Code	Identifies notification type codes.
Party Type Code	Defines the role of the Party.
Reconciliation Status Code	Identifies the status of a reconciliation process.
Transaction Status Code	Identifies the status of a transaction.
Transaction Type Code	Identifies the type of transaction.
Unit Type Code	Identifies the type of unit.

Figure G2: Account Type Code

Code	Description
100	Holding Account
110	Pending Account
120	Operator Holding Account
121	Person Holding Account
210	Net Source Cancellation Account (Type 1) (national registries only)
220	Non-compliance Cancellation Account (Type 2) (national registries only)
230	Voluntary Cancellation Account (Type 3) (national registries only)
240	Excess Issuance Cancellation Account (Type 4) (CDM Registry only)
250	Mandatory Cancellation Account (Type 5)
300	Retirement Account
411	tCER Replacement Account for Expiry (Type 1)
421	lCER Replacement Account for Expiry (Type 1)
422	lCER Replacement Account for Reversal in Storage (Type 2)
423	lCER Replacement Account for Non-submission of Certification Report (Type 3)

Figure G3: Commitment Period Code

Code	Description
0	Supplementary Program Commitment Period (2005-2007)
1	First Commitment Period
2	Second Commitment Period
3	Third Commitment Period
4	Fourth Commitment Period

Figure G4: LULUCF Activity Codes

Code	Description
1	Afforestation and reforestation
2	Deforestation
3	Forest management
4	Cropland management
5	Grazing land management
6	Revegetation

Figure G5: Notification Status Code

Code	Description
1	Initial
2	Incomplete
3	Complete

Figure G6: Notification Type Codes

Code	Description
1	Net Source Cancellation
2	Non-compliance Cancellation
3	Impending Expiry of tCER or ICER
4	Reversal of Storage for CDM Project
5	Non-submission of Certification Report for CDM Project
6	Excess Issuance for CDM Project
7	Commitment Period Reserve
8	Unit Carry-over
9	Notification Update

Figure G7: Party Type Codes

Code	Description
1	Initiating Registry
2	Acquiring Registry

Figure G8: Reconciliation Status Code

Code	Description
0	Confirmed
1	Initiated
2	Validated
3	ITL Totals Inconsistent
4	ITL Unit Blocks Inconsistent
5	ITL Completed
6	ITL Completed with Manual Intervention
7	ITL Start Request Denied
8	STL Totals Inconsistent
9	STL Unit Blocks Inconsistent
10	STL Validated
11	STL Completed with Manual Intervention

Figure G9: Transaction Status Code

Code	Description
1	Proposed
2	Checked (No Discrepancy)
3	Checked (Discrepancy)
4	Completed
5	Terminated
6	Rejected
7	Cancelled
8	Accepted
9	STL Checked (No Discrepancy)
10	STL Checked (Discrepancy)

Figure G10: Transaction Type Code

Code	Description
1	Issuance - Initial creation of a unit
2	Conversion - Transformation of a unit to create an ERU
3	External - External Transfer of unit between registries
4	Cancellation - Internal transfer of unit
5	Retirement - Internal transfer of unit
6	Replacement - Replacement of tCER or lCER
7	Carry-over - Change of validity to subsequent Commitment Period
8	Expiry Date Change
10	Internal transfer of a unit/supplementary program transaction

Figure G11: Unit Type Code

Code	Description
0	Non-Kyoto Unit
1	AAU - Assigned Amount Unit
2	RMU - Removal Unit
3	ERU - Emission Reduction Unit (converted from an AAU)
4	ERU – Emission Reduction Unit (converted from an RMU
5	CER - Certified Emission Reduction Unit
6	tCER - Temporary CER
7	lCER - Long-term CER