

Electronic Tagging — Functional Specifications

Version 1.651

June 19, 2000

**Transaction Information System
Working Group**



North American Electric Reliability Council

Table of Contents

TABLE OF CONTENTS.....	I
------------------------	---

CHAPTER 1 — FUNCTIONAL DESCRIPTION 1

1.1 INTRODUCTION.....	1
<i>Constrained Path Method Resolution</i>	1
<i>Expected Benefits</i>	2
1.2 DEFINITIONS.....	3
1.3 SYSTEM CONCEPTS.....	4
1.3.1 <i>System Architecture</i>	5
1.3.2 <i>Transaction Identification</i>	6
1.3.3 <i>Test Tags</i>	7
1.3.4 <i>Transaction States</i>	8
1.3.5 <i>Timing Requirements</i>	13
1.3.6 <i>Tag Auditing</i>	14
1.4 FUNCTIONAL CONCEPTS.....	15
1.4.1 <i>Tag State Management</i>	15
1.4.2 <i>Tag Submission</i>	15
1.4.3 <i>Tag Distribution</i>	15
1.4.4 <i>Tag Evaluation</i>	16
1.4.5 <i>Tag Notification</i>	16
1.4.6 <i>Tag Forwarding</i>	17
1.4.7 <i>Tag Querying</i>	17
1.4.8 <i>Tag Cancellation</i>	17
1.4.9 <i>Tag Adjustment</i>	18
1.4.10 <i>Tag Replacement</i>	19
1.4.11 <i>Adjustment Notification</i>	20
1.5 FUNCTIONAL REQUIREMENTS	21
1.5.1 <i>Tag Agent Service</i>	21
1.5.2 <i>Tag Authority Service</i>	26
1.5.3 <i>Tag Approval Service</i>	49
1.5.4 <i>Interchange Distribution Calculator</i>	57

CHAPTER 2 — PROTOCOL DESCRIPTION 60

2.1 INTRODUCTION.....	60
2.2 USE OF THE TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL.....	60
2.2.1 <i>Establishing Connections</i>	61
2.2.1.1 <i>Partial Connection Failures</i>	61
2.3 USE OF THE HYPERTEXT TRANSPORT PROTOCOL.....	61
2.3.1 <i>HTTP Requests</i>	62
2.3.2 <i>HTTP Responses</i>	62

2.4	USE OF THE TAGGING MESSAGING PROTOCOL	63
2.4.1	<i>Generic Request Structure</i>	63
2.4.2	<i>Generic Response Structure</i>	63
2.4.3	<i>Requests and Responses</i>	64
2.4.4	<i>Protocol Error Description</i>	80
CHAPTER 3 — DATA MODEL FOR TAG INFORMATION.....		82
3.1	INTRODUCTION	82
3.2	DATA ENCODING RULES	82
3.3	TAG DATA DESCRIPTION	84
3.3.1	<i>Header Format</i>	84
3.3.2	<i>Table Formats</i>	85
3.3.3	<i>Data Model Extensions</i>	111
3.3.4	<i>Intra-Table Validation Rules</i>	112
3.3.5	<i>The END OF DATA Marker</i>	112
3.3.6	<i>Attribute Dictionary</i>	113
APPENDIX A — TMP PROTOCOL SPECIFICATION		117
A.1	AUGMENTED BNF	117
A.2	BASIC RULES	118
A.3	BASIC DATA TYPES	119
A.4	BASIC DATA ENCODING RULES	119
A.5	TMP MESSAGES	119
A.6	TMP REQUESTS	120
TMP REQUESTS (CONTINUED)		121
A.7	TMP RESPONSES	122
A.8	SPECIAL TAG DATA	122
APPENDIX B — ACTIVITY DIAGRAMS		123
B.1	TAG STATE MANAGEMENT	124
B.2	TAG SUBMISSION	125
B.3	TAG DISTRIBUTION	126
B.4	TAG EVALUATION	127
B.5	TAG NOTIFICATION	1
B.6	TAG FORWARDING	129

B.7 TAG QUERYING..... 130

B.8 TAG CANCELLATION 131

B.9 TAG ADJUSTMENT 132

B.10 TAG REPLACEMENT..... 133

 B.10.1 REPLACEMENT INITIALIZATION 133

 B.10.2 REPLACEMENT COMPLETION 134

APPENDIX C — SAMPLE TMP DIALOGS 135

 C.1 State Management..... 135

 C.2 Tag Submission 135

 C.3 Tag Distribution..... 137

 C.3.1. Via ASSESS Message 137

 C.3.2. Via INFO Message..... 139

 C.4 Tag Evaluation..... 141

 C.5 Tag Notification 142

 C.5.1. Via NOTIFY Message 142

 C.6 Tag Forwarding 143

 C.7 Tag Querying 145

 C.7.1. Via STATUS Message 145

 C.7.2. Via DSTATUS Message 146

 C.8 Tag Cancellation..... 148

 C.9 Tag Adjustment 149

 C.10 Tag Replacement..... 150

APPENDIX D — BACKUP FAX FORM 152

APPENDIX E — ELECTRONIC TAGGING – REGISTRY DEFINITION VERSION 1.6..... 1

1. INTRODUCTION..... 4

2. REGISTRY FILE UPDATES AND LOCATION 4

 2.1. E-TAG REGISTRY 4

 2.2. NERCTAG REGISTRY..... 5

3. BASIC RULES 6

4. CSV FILE FORMATS..... 7

5. DATABASE DEFINITIONS..... 8

 5.1. ENTITY_REGISTRY 9

 5.2. CA_REGISTRY 14

 5.3. PSE_REGISTRY 19

 5.4. SC_REGISTRY 24

 5.5. TP_REGISTRY 29

 5.6. CHANGE_LOG 33

 5.7. PRODUCT_REGISTRY..... 35

 5.8. REGISTRY_VERSION..... 36

 APPENDIX F — E-Tag Requirements Necessary for Implementation of NHM Service..... 1

Chapter 1 — Functional Description

1.1 Introduction

This document describes the functional requirements and detailed technical specifications for the implementation of an electronic Transaction Information System (TIS) also referred to as Electronic Tagging. These requirements and specifications are an extension to the policies and procedures established to identify and communicate interchange transaction information between parties (tags) as implemented in the interim TIS (iTIS) in accordance with NERC Policy 3.

Constrained Path Method Resolution

In November 1998, the NERC Operating Committee adopted a resolution to use the Constrained Path Method (CPM) as the basis for determining Interchange Transaction curtailment priorities as part of the NERC Transmission Loading Relief (TLR) Procedure. That resolution called for:

1. By December 1, 1998, all Regional Councils are responsible for monitoring and reporting to the NERC Operating Committee Secretary each month the compliance with the Requirements of Policies 3, “Interchange,” and 9, “Security Coordinator Procedures.” This includes compliance with those Requirements for tag development and transmittal by the Purchasing-Selling Entities, receipt and implementation by the Control Areas, and forwarding to the interim Interchange Distribution Calculator by the Control Areas or Security Coordinators. The Regional reports shall include lists of the entities, including Control Areas and Purchasing-Selling Entities, who are and are not in compliance. These reports will be posted on the NERC web site. Furthermore, Purchasing-Selling Entities shall report to the Operating Committee Secretary whenever they find a Control Area is not in compliance with Policy 3 for investigation.
2. By December 16, 1998, the Transaction Information System Working Group (TISWG) shall post a written functional specification for a tagging system that will enable the electronic dissemination of tags for next-hour Interchange Transactions in accordance with NERC Operating Policy 3 and Appendix 3A. This specification shall be posted on the NERC web site for public access.
3. By February 1, 1999, NERC shall offer testing and testing resources for verification of electronic tagging of all Interchange Transactions, and verification that all are correctly input to the interim Interchange Distribution Calculator.
4. By February 1, 1999, NERC shall implement an information plan to assist the Control Areas, Purchasing-Selling Entities, and Security Coordinators with the tagging requirements for the implementation of CPM.
5. By March 1, 1999, all electronic tagging systems, including those provided by NERC and other parties, submitted by February 1, 1999, shall have been tested for compliance with this specification. The names of those compliant systems tested by the TISWG will be posted on the NERC web site.
6. By March 1, 1999, all Control Areas and Purchasing-Selling Entities shall use an electronic tagging system that is compliant with the standards and specifications approved by the TISWG as detailed in Resolution 2.

7. By May 1, 1999, NERC shall begin full implementation of CPM, with electronic tags required for all Interchange Transactions.

The NERC TISWG has prepared this document in fulfillment of that resolution.

Expected Benefits

The specifications in this document are expected to:

- Ensure that all tags can be electronically transferred between all market segments with assurance of receipt,
- Improve the speed and efficiency of transmitting tags to help support the hourly market,
- Improve the quality of and uniformity of tags to increase their usefulness, and
- Assure that all tags entering the Interchange Distribution Calculator (IDC) are unique and accurately reflect transactions actually occurring on the system.

Successful implementation of electronic TIS depends on a common agreement of:

- The roles and responsibilities of all parties to an Interchange Transaction,
- the specific data required to represent an Interchange Transaction, and
- the specific mechanisms to be used to exchange that data electronically.

The Functional Requirements, Data Model Description, and Protocol Description sections of this document address each of these issues.

1.2 Definitions

Term	Definition
Approval Entity	Control Areas or Transmission Providers as identified on the transaction path of a tag.
Approval Rights	The rights that an entity has to approve, deny, curtail, or otherwise modify a tag.
Approval Status	Status of each approval entity along the path.
Author Rights	The rights a tag author has to withdraw, cancel, or terminate a tag.
Composite Information	That collection of data associated with a tag that is subject to modification over time and effects the overall processing or interpretation of the tag. Includes composite status, composite stop date/time, and adjustment information.
Composite Status	Overall status of the transaction/tag.
Control Area/CA	An electrical system bounded by interconnection (tie line) metering and telemetry.
Generation (Source) Control Area/GCA	Control area in which generation is located.
IDC	Interchange Distribution Calculator.
Load (Sink) Control Area/LCA	Control area in which the load is located.
Master Registry	Data set provided by NERC describing entity information, such as name, acronym, phone numbers, service URLs, etc... of registered participants.
Purchasing-Selling Entity/PSE	Any entity eligible to apply for an order requiring a Transmitting utility to provide Transmission Services under Section 211 of the Federal Power Act.
Receiving Control Area/RCA	Any Control Area receiving upstream power.
Security Coordinator/SC	An entity that provides the security assessment and emergency operations coordination for a group of control areas.
Sending Control Area/SCA	Any Control Area sending power downstream.
Sink	Final Point of Delivery for the transaction; the actual load.
Source	Initial Point of Receipt for the transaction; the actual generation facility.
Tag Agent Service	Software component used to generate and submit new Tags to an Authority.

Term	Definition
Tag Approval Service	Software component used to indicate individual path approvals when requested by Authority Service.
Tag Author	Entity that writes and submits a tag.
Tag Authority Service	Software component that receives Agent submissions and forwards them to the appropriate Approval Services. Also maintains master copy of Tag, its status, and responds to queries regarding the Tags in its possession.
Tag ID	Identifier of the Tag currently represented by combining GCA, PSE, a unique number, and LCA.
Tag Key	Security data associated with a particular Tag and entity used to authenticate and determine permissions of the entity.
TIS	Transaction Information System - Tagging.
TMS	Transaction Management System - Scheduling.
Transmission Provider/TP	An entity that owns, operates, or manages transmission facilities.

1.3 System Concepts

The functional requirements for an all-electronic TIS (electronic tagging) must address the basic information and data exchange needs of:

- initial creation of an electronic “tag” representing the transaction with basic syntactical error handling,
- dissemination of the tag to all parties directly involved in the transaction,
- collection of approvals from all parties, and
- uploading of approved tags to the Interchange Distribution Calculator (Eastern Interconnection only).

In addressing each of these needs, consideration must be given to 1) which parties to a transaction have responsibility for the data at each step in the exchange, 2) insuring data integrity without duplicate data entry or potential for replication errors, 3) minimizing the number of data transfers between parties, and 4) providing mechanisms to recognize and overcome system failures.

This document approaches the functional requirements for electronic tagging by defining three “services.”

The functionality that must be supported by each of these services and the entity responsible for providing for these services are defined. There are no restrictions with regard to who may provide these services (i.e., the responsible entity or any one of a number of third-party service providers) nor any restrictions on

which services (or all) that a third-party service provider could offer. **Under no circumstances shall a provider of any of these services require any other service provider to implement additional features or functionality beyond these specifications as a condition to properly performing the obligations associated with that service.**

1.3.1 System Architecture

The system is defined through three basic services, as shown in figure 1.3.1. Explanations of each of the services and their purpose follows.

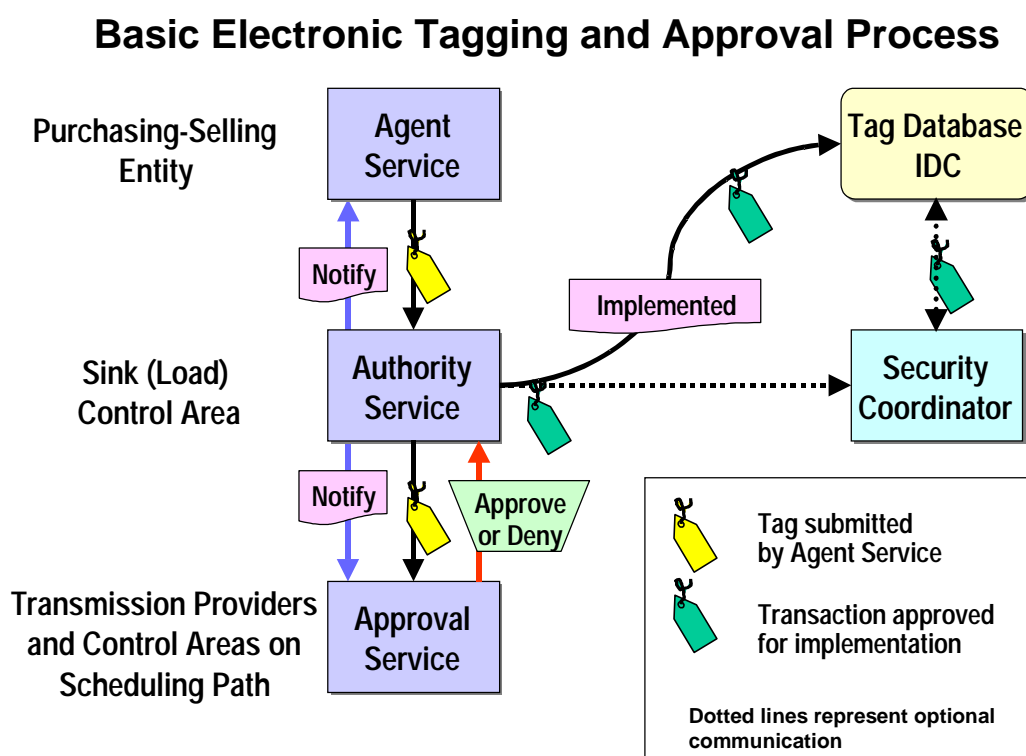


Figure 1.3.1

1.3.1.1 Tag Agent Service

The Tag Agent Service provides for the initial creation of an electronic tag representing an interchange transaction and the transfer of that information to the appropriate Tag Authority Service. Purchasing-Selling Entities (PSEs) and other Tag Authors are responsible for providing this service directly or by arranging with a third party to provide this service as their agent. Tags created by the Tag Agent Service are forwarded to the Tag Authority Service associated with the Sink (Load) Control Area (LCA). The Tag Agent Service provides a mechanism for the Tag Author to view the approval status of their transactions either by simple polling or via an optional unsolicited notification mechanism. The Tag Agent Service also provides facilities for the Tag Author to force the withdrawal, cancellation, or early termination of

any of their tags. Tag Agent services may optionally (or as decided by region) accept notification of transactions to which their operators are a party as well.

1.3.1.2 Tag Authority Service

The Tag Authority Service provides the focal point for all interactions with a tag and maintains the single authoritative “copy of record” for each tag received from any Tag Agent Service. Every Control Area is responsible for providing this service directly or by arranging with a third party to provide this service as its agent. The Tag Authority Service forwards all received tags to the Tag Approval Service associated with each entity identified in the transaction as having “approval rights” over that transaction, and collects approvals/denials issued by these Tag Approval Services. Based on time and/or the messages received from the Tag Approval Services, the Tag Authority arbitrates and sends the final disposition of the tag to the originating Tag Agent and all Tag Approval Services associated with the transaction, and, at the direction of the Sink (Load) Control Area, to that CAs designated forwarding location (e.g., IDC or CA’s Security Coordinator). The Tag Authority Service also provides the capability for both Tag Agent and Tag Approval Services to interrogate the current approval status of any transaction tag on demand.

1.3.1.3 Tag Approval Service

The Tag Approval Service receives all tags/transactions submitted by Tag Agent Services via the appropriate Tag Authority Service, and provides a means for the approval entity to send an approve or deny response to the Tag Authority Service (through Tag Evaluation), as well as curtail an existing tag (through Tag Adjustment). Control Areas and Transmission Providers are responsible for providing this service directly or for arranging with a third party to provide this service as their agent.

1.3.2 Transaction Identification

All Interchange Transaction tags shall be uniquely identified by a Tag ID. Electronic communications between Tag Agent, Authority, and Approval Services shall require the association of an additional Tag Key to control all interactions related to a given transaction. The following subsections describe the requirements for the creation of the Tag ID and Tag Key.

1.3.2.1 Tag ID

Every transaction shall be identified by a unique Tag ID based on key attributes of the transaction as specified in the Data Model:

- Source (Generation) Control Area Code
- PSE Code (PSE originating the transaction tag)
- Unique transaction identifier
- Sink (Load) Control Area Code

Since this Tag ID and the contents of the transaction tag contain potentially commercially sensitive information, all components of the Tagging Information System shall treat such information as confidential.

All components of the Tagging Information System shall reject any attempt to submit as new a Tag ID that is identical to an existing transaction’s Tag ID for a period of one (1) year from the stop date and time associated with the existing tag. Tag Agent Services shall be required to ensure that each Tag ID is unique for a period of not less than one (1) year from the stop date and time associated with the last transaction that was assigned that Tag ID.

1.3.2.2 Tag Key

The electronic exchange of tag information shall require the assignment of unique “Tag Keys” to be associated with the transaction Tag ID. Tag Keys together with the Tag ID controls communication between the Tag Agent, Authority, and Approval Services.

The Tag Key shall consist of the following information:

- 1– 6 character code associated with the entity creating the key
- Unique 12 character alphanumeric (0–9, A–Z, a–z; case sensitive) security token

The Tag Agent generates a unique Tag Key to associate with the tag at the time of submission. This Tag Key consists of the PSE code associated with the Tag Author and a unique 12 character token. All subsequent messages exchanged between the Tag Agent Service and the Tag Authority Service in regard to the transaction tag shall refer to both the Tag ID and Tag Key assigned by the Tag Author’s Tag Agent Service.

The Tag Authority shall also generate unique Tag Keys to be associated with the transaction tag on the initial transmission of the tag to each of the appropriate Tag Approval Services. Each of these Tag Keys shall consist of the CA code associated with the Sink (Load) Control Area and a unique 12 character token. All subsequent messages exchanged between the Tag Authority Service and a given Tag Approval Service in regard to the transaction tag shall refer to both the Tag ID and Tag Key assigned by the LCA’s Tag Authority.

The unique security token portion of the Tag Key should either be random or have the appearance of randomness. Although schemes may be used to generate a key, these schemes should not be obvious to the interested observer (for example, APR05991240X is obviously a date and time, but a ciphered version of this, KYZ71434450H, might not be). The Tag Key should be considered a security mechanism, and as such, should not be easily deducible by parties lacking first-hand knowledge of the specific Tag Key generation mechanism employed by the system.

It should be noted that each Tag Authority service is assigned by NERC a unique key for interaction with the IDC. This key is only to be used for communication with the IDC, and must be kept confidential. Communication from the IDC to each Tag Authority service is secured by this key as well. NERC will notify each registered Tag Authority service with that Authority's unique IDC key to be used in all messages between the IDC and Tag Authority.

1.3.3 Test Tags

Test Tags are tags used for the purpose of troubleshooting a system or component of the system. All services (Tag Agent, Authority, and Approval) shall accept Test Tags and process them in an identical fashion to all other tags, with the following exceptions:

- Viewing applications MUST indicate to the user that the tag is a Test Tag.
- Test Tags do not require an approving party to evaluate the tag within the Assessment Time as defined in Policy 3.
- Test Tags are not to be treated as actual tags (the information contained within a Test Tag should not be used to make any business decisions).

- The Tag Authority Service shall not initiate the forwarding of these test Tags to the Interchange Distribution Calculator at any time.

In addition, the following rules must be observed with regard to test tags:

- Test Tags may ONLY be used for troubleshooting purposes. System Development, Training, and Demonstration, as well as any other non-troubleshooting related need may NOT utilize the Test Tag feature.
- A particular PSE (as listed in the Master Registry) may only issue a total of ten (10) Test Tags per clock hour. Any Test Tag submissions exceeding this number may be rejected at the option of the service being sent the Test Tag.
- Test Tags may be rejected at the option of the service provider if they are sent during the last twenty minutes of a clock hour (i.e., xx:40 – yy:00).

1.3.4 *Transaction States*

Fundamental to the electronic exchange and management of a transaction/tag is the representation of the various “states” a single tag/transaction may adopt from the time of its inception, through the approval cycle, and finally to the end of its life. These states are reflected by two separate attributes in the Data Model that represent:

- An approval status for the tag/transaction with respect to each separate entity that has “approval rights” over the transaction, and
- A single composite status of the tag/transaction based on those individual approvals.

For each tag submitted by a Tag Agent Service, the corresponding Tag Authority Service is responsible for maintaining the Approval Status value associated with each entity that has “approval rights” over the transaction. The Tag Authority Service is also responsible for resolving the Composite Status of the tag based on these separate Approval Status values, and returning both the set of individual Approval Status values and the Composite Status value of the tag when queried by either the Tag Agent or Tag Approval Services.

The particular status values that any tag/transaction may adopt over time are defined below. Where appropriate, status values are assigned names already in common usage for managing the transmission reservation process on OASIS.

1.3.4.1 **Approval Status**

The following table defines the unique Approval Status values the Tag Authority Service shall maintain in association with each entity that has “approval rights” over a tag/transaction. These states are maintained in the STATUS table, as defined in the data model. The Tag Authority Service shall maintain a time stamp corresponding to the date and time the Approval Status associated with each entity is updated, as well as a time stamp corresponding to the date and time the tag was initially transferred to each entity’s Tag Approval Service. All time stamps are to be based on the time the Tag Authority Service received the appropriate information.

The valid state transitions for Approval Status are illustrated in Figure 1.3.3.1. Tag Approval Services direct the Tag Authority to change Approval Status through appropriate message exchanges. The Tag Authority also may effect changes to Approval Status as part of a manual override facility.

Failure to transfer the transaction tag information to a particular Tag Approval Service shall be indicated by the Approval Status of COMM_FAIL. Failure of a Tag Approval service to accept transfer of a tag due to syntactic validation errors, invalid Tag ID, etc., shall be indicated by the Approval Status of INVALID. The REASON field of the STATUS Table shall be the first 80 characters of rules violations returned by the Approval party. COMM_FAIL and INVALID may be updated by an Approval Entity, or may be overridden to DENIED, STUDY, or APPROVED by an appropriately authorized individual/entity as part of the remedial action taken in the event of a communication outage, etc. However, the Tag Authority shall prohibit any attempt to override Approval Status at any time following a change in composite state to any of the following: CONDITIONAL, IMPLEMENT, DENIED, WITHDRAWN, CANCELLED, TERMINATED, or ADJUSTED.

The Tag Authority shall accept any Approval Service directed change in Approval Status that is a valid state transition at any point in time prior to the achievement of the following composite states: CONDITIONAL, IMPLEMENT, DENIED, WITHDRAWN, CANCELLED, TERMINATED, or ADJUSTED. On any change made to an individual entity's Approval Status, the Tag Authority shall reevaluate the tag's Composite Status value based on the then current set of Approval Status values. Not all changes in Approval Status will be reflected in a corresponding change to Composite Status, as limited by the valid state transitions for Composite Status.

Approval Status Value	Description/Interpretation
PENDING	Initial status for all tags received by Tag Authority; Tag Authority has initiated transfer of tag to the appropriate Tag Approval Service.
COMM_FAIL	Set by Tag Authority after all attempts to transfer a transaction tag to a particular Tag Approval Service has failed (i.e., failed to establish a connection or data transfer was not acknowledged by Tag Approval Service). Also used to indicate that an Approval Entity does not have a registered URL for their Tag Approval Service.
QUEUED	Status value set by the Tag Authority after successful transfer of a transaction tag to the Tag Approval Service (i.e., valid connect, send, positive-acknowledge application software handshake completed). QUEUED status indicates that the transaction tag has been successfully received, passed all syntactic validity checks, and has been entered into the Tag Approval Service for review by the approving entity.
STUDY	Set by Tag Authority in response to request from Tag Approval Service indicating that evaluation of the transaction is under way. This is an informational state only that may or may not be implemented by a particular Tag Approval Service.
INVALID	Status value set by the Tag Authority in response to the Tag Approval Service's refusal to accept the transaction tag. INVALID status indicates that the transaction tag failed one or more syntactic validity checks performed by the Tag Approval Service. This error is fairly severe, in that it indicates that the Authority and Approval services do not agree on the syntactical validity of the tag.
DENIED	Status set by the Tag Authority in response to the Tag Approval Service indicating that the approval entity has denied the tag/transaction.

Approval Status Value	Description/Interpretation
APPROVED	Status set by the Tag Authority in response to the Tag Approval Service indicating that the approval entity has approved the tag/transaction.
<null>	Indicates that a PSE has no registered Tag Agent Listener Service (per the Master Registry).

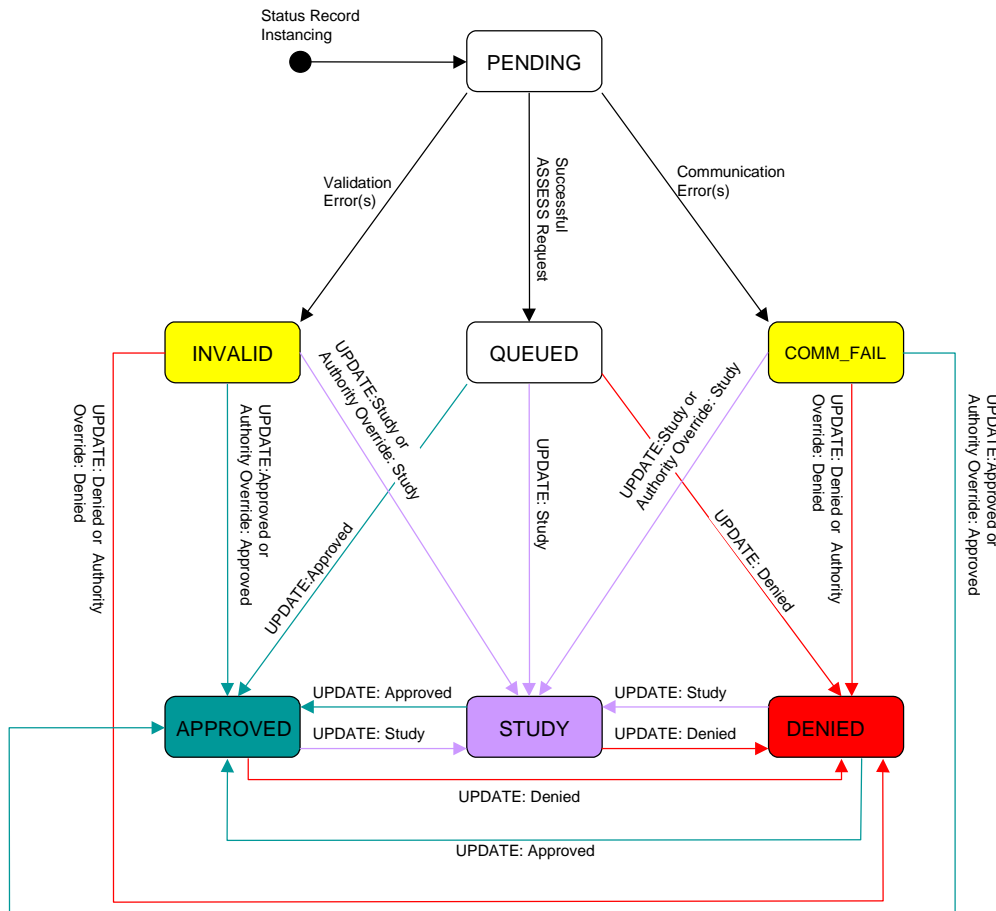


Figure 1.3.3.1

1.3.4.2 Composite Status

The following table defines the unique Composite Status values the Tag Authority Service shall maintain based on the individual Approval Status values collected in association with each entity that has “approval rights” over a tag/transaction. The Tag Authority Service shall maintain a time stamp corresponding to the date and time the tag’s Composite Status is updated as well as information regarding the entity that caused the change in status.

The valid state transitions for Composite Status for tags submitted “on time” are illustrated in Figure 1.3.3.2a. The valid state transitions for Composite Status for tags NOT submitted “on time” are illustrated in Figure 1.3.3.2b. The ATTN_REQD Composite Status may be overridden indirectly to PENDING or LATE as a result of Approval action or the overriding of an individual entity’s Approval Status. Not all changes in Approval Status will be reflected in a corresponding change to Composite Status, as limited by the valid state transitions for Composite Status.

Composite Status Value	Description/Interpretation
PENDING	Initial status for all tags received “on time” by the Tag Authority (as defined in Policy 3); Tag Authority has initiated transfer of tag to the appropriate Tag Approval Services.
LATE	Initial status for all tags received “late” by the Tag Authority (as defined in Policy 3); Tag Authority has initiated transfer of tag to the appropriate Tag Approval Services.
DENIED	Set by the Tag Authority when the nominal approval time period established by the Industry has lapsed, the COMPOSITE STATE is currently PENDING, and one or more approval entities have a state of DENIED. Also set when the nominal approval time period established by the Industry has lapsed, the COMPOSITE STATE is currently LATE, and one or more approval entities does not have a state of APPROVED.
IMPLEMENT	Set immediately by the Tag Authority when all approval entities have a state of APPROVED.
CONDITIONAL	Set by the Tag Authority when the nominal approval time period established by the Industry has lapsed, the composite state is PENDING, and not all approval entities have APPROVED, but all have one of the following states: QUEUED, STUDY, APPROVED. This state is to be interpreted identically to the IMPLEMENT state, with the understanding that those entities with approval rights that have not explicitly APPROVED the transaction should have agreed with their adjacent path participants to implement the transaction.
WITHDRAWN	Set by Tag Authority on receipt of a Tag Cancellation request from the Tag Agent Service associated with the Tag Author prior to the transaction’s Composite Status being set to either IMPLEMENT, CONDITIONAL, or DENIED, and where the effective date/time is equal to the transaction’s start date/time (i.e., the tag is cancelled in its entirety).
CANCELLED	Set by Tag Authority on receipt of a Tag Cancellation request from the Tag Agent Service associated with the Tag Author after the Composite Status has been set to either IMPLEMENT or CONDITIONAL and where the effective date/time is equal to the transaction’s start date/time (i.e., the tag is cancelled in its entirety).

Composite Status Value	Description/Interpretation
TERMINATED	Set by Tag Authority on receipt of a Tag Cancellation request from the Tag Agent Service associated with the Tag Author after the Composite Status has been set to either IMPLEMENT, CONDITIONAL, or TERMINATED and where the effective date/time is after the transaction’s start date/time (i.e., the transaction is being halted prematurely).
ADJUSTED	Set by Tag Authority on receipt of a Tag Adjustment request from a Tag Approval service.

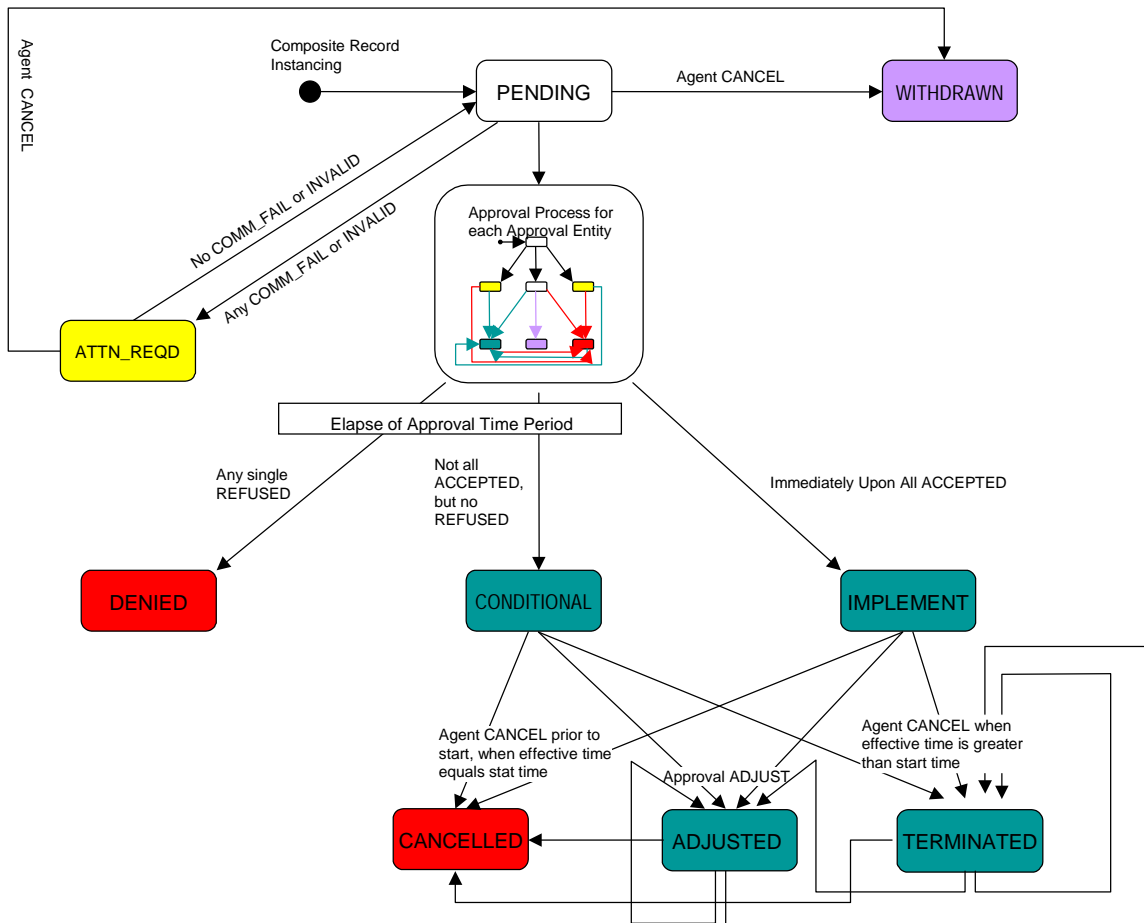


Figure 1.3.3.2a — “On Time” State Transitions

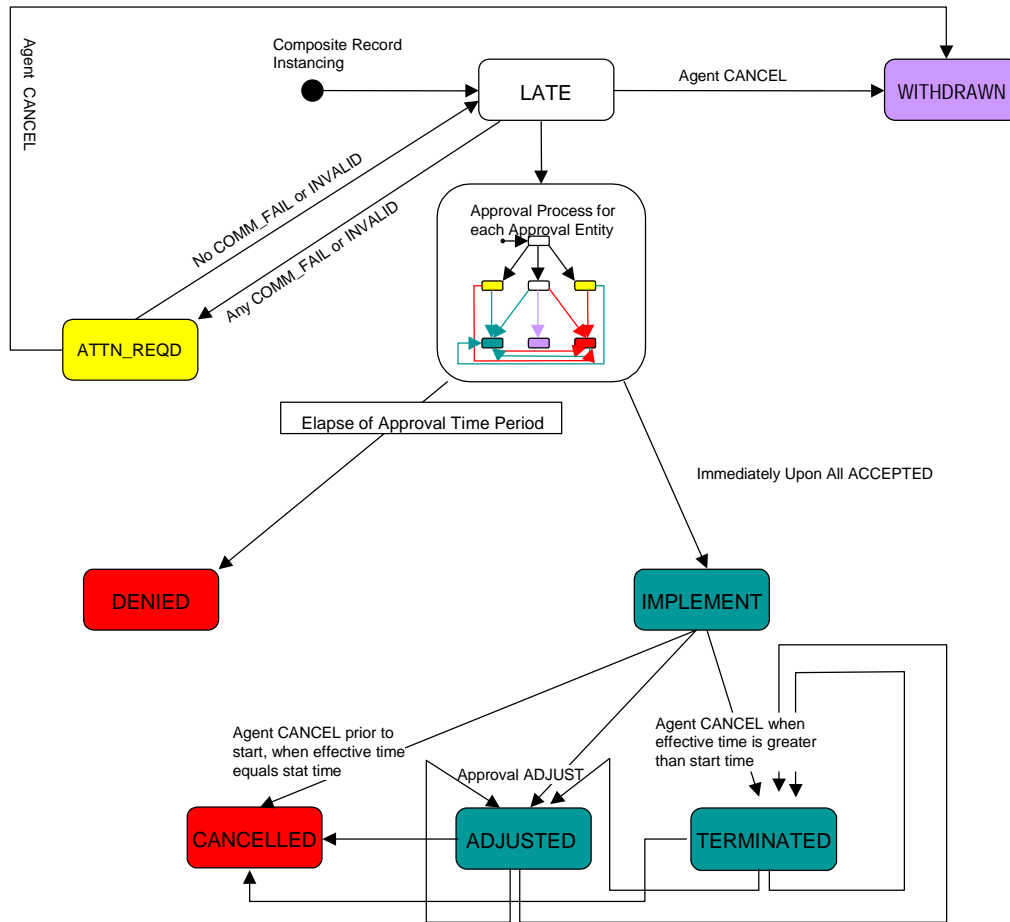


Figure 1.3.3.2b — “Late” State Transitions

1.3.5 Timing Requirements

ETAG shall observe timing guidelines as set forth in NERC Policy 3.

To enforce Tag submission and Assessment timing requirements, the Tag Authority Service shall maintain system time to an accuracy of one (1) second traceable to the National Institute of Standards and Technology (NIST). Tag Approval and Agent services are encouraged to keep their time synchronized in this manner as well.

All times communicated through ETAG shall be noted in Central Standard Time. This does not require a particular interface display only CST; however, it does require that any system using time zones other than CST properly convert those times into CST prior to communicating with other systems.

NOTE: Timing requirements will be defined in Policy 3, Appendix 3A1. Until such time that Policy 3, Appendix 3A1, is implemented, the following timing guidelines shall be observed.

Transaction Duration	Submit Deadline	Submission Time	Assessment Time	Time to Start of Transaction
Less than 24 Hours	20 Minutes prior to start	20 Min to 4 Hours prior to start	≤ 10 Min from tag receipt	≥ 10 Min
		≥ 4 Hours prior to start	≤ 2 Hours from tag receipt	≥ 2 Hours
24 Hours or longer	4 Hours prior to start	≥ 4 Hours prior to start	≤ 2 Hours from tag receipt	≥ 2 Hours

1.3.6 Tag Auditing

Each service shall be responsible for keeping audit information describing its interactions with other services. These requirements are described below.

1.3.6.1 Message Rejection Log

Any service that rejects a message (other than STATUS or DSTATUS requests) should log the type of rejection, the date/time of the rejection, the sending entity (if identifiable), and the Tag ID (if identifiable). STATUS and DSTATUS rejections may be logged at the option of the provider. This information must be kept available by written request for a minimum of ninety (90) days after the rejection.

1.3.6.2 SUCCESS Tag Archive

Every service shall keep available for retrieval every tag received via SUBMIT or ASSESS messages until that tag's stop date/time is more than ninety (90) days in the past. Tag Authority Services should have this information available to Approval and Agent systems through standard ETAG querying mechanisms throughout the ninety-day period, as well as through written request by other parties who may require data but not be participants listed on the tag (i.e., NERC). Tag Agent and Approval Services must have these tags available by written request.

1.3.6.3 FAIL Tag Log

The Tag Authority service shall maintain a record of events where a response of FAIL was received when sending an ASSESS. The log shall reflect the Tag Id, the date/time of the rejection, the entity rejecting the message as invalid, and the reasons given for the rejection. This information must be kept available by written request for a minimum of ninety (90) days after the rejection. This information must be available to parties involved in the ASSESS message, as well as other parties who may require data but not be participants listed on the tag (i.e., NERC).

1.3.6.4 Statistics

Every service shall maintain statistical information as defined below. This information should be logged as it occurs, NOT after the fact. In this manner, services may accurately reflect data before it is modified through overrides or updates. This information should be grouped per week basis (from 00:00 CS Sunday to 23:59:59 CS Saturday) and must be available by written request for a minimum of ninety (90) days. This information must be available to parties who may require data but not be participants to any specific tag (i.e., NERC).

- Number of LATE tags, by PSE
- Number of tags that have been assigned a state of ATTN_REQD
- Number of Tags marked INVALID by Provider
- Number of Tags marked COMM_FAIL by Provider

1.3.6.5 Authority TMP Log

All Tag Authority Services shall archive all TMP dialogues (all received and issued TMP messages and their associated responses). These message logs need not be available for online query, however, upon written request from NERC, Authority operators must be able to supply written reports within a reasonable amount of time (minimum one working week) listing TMP traffic for a particular entity or transaction. This information shall be kept from the implementation of the 1.5 Specification forward until such time this requirement is removed.

1.4 Functional Concepts

1.4.1 Tag State Management

Tag State Management defines the process through which the Tag Authority Service takes action based upon certain events. This process is handled completely within the Tag Authority Service.

Upon receipt of certain messages or occurrence of certain events, the Tag Authority Service must evaluate the state of the tag in question and modify it as appropriate or take certain actions. The modification of state of itself may prompt additional actions. This may include, but is not limited to, Tag Notification and Tag Forwarding, as described below.

1.4.2 Tag Submission

Tag Submission describes the process in which the Tag Agent Service submits a tag to the Tag Authority Service for processing. This is accomplished through the use of the SUBMIT message.

The Tag Agent Service first determines whether or not the tag in question has met all the validity requirements defined in this document. If so, the Tag Agent connects to the Tag Authority Service for the tag's Load Control Area and effects an exchange of information with that Tag Authority Service. The Tag Authority Service determines whether or not the tag in question has met all the validity requirements defined in this document. If so, the Tag Authority Service processes the tag and responds with a SUCCESS to the Tag Agent Service; if not, the Tag Authority Service responds with a FAIL to the Tag Agent Service.

1.4.3 Tag Distribution

Tag Distribution describes the process in which the Tag Authority Service submits a tag to various entities involved with the transaction represented therein. This is accomplished via two methods: the ASSESS message, and the INFO message.

1.4.3.1 Entities with Approval Rights

Tag Distribution begins immediately after the acceptance of a Tag Submission. Using information obtained during the processing of the Tag Submission, the Tag Authority Service connects to the various

entities with Approval rights over the transaction and effects an exchange of information with each entity's Tag Approval Service utilizing the ASSESS message. Each Tag Approval Service determines whether or not the tag in question has met all the validity requirements defined in this document. If so, the Tag Approval Service processes the tag and responds with a SUCCESS to the Tag Authority Service; if not, the Tag Approval Service responds with a FAIL to the Tag Authority Service. A failure will initiate Tag Notification to the Tag Author, advising them of the error.

1.4.3.2 Entities without Approval Rights

The Tag Authority Service must also issue informational messages to entities that are party to the transaction described on the tag, but do not have explicit approval rights. Currently, these informational messages are only issued when a tag represents business partially or completely within WSCC. These messages are handled via the INFO message. Using information obtained during the processing of the Tag Submission, the Tag Authority Service connects to the Tag Agent Notification URL associated with each entity and effects an exchange of information with that Tag Agent Service utilizing the INFO message. Each Tag Agent Service determines whether or not the tag in question has met all the validity requirements defined in this document. If so, the Tag Agent Service processes the tag and responds with a SUCCESS to the Tag Authority Service; if not, the Tag Agent Service responds with a FAIL to the Tag Authority Service. If the Tag Agent does not have a valid notification URL, the Tag Authority proceeds without error. It should be noted that the Tag Author does NOT receive INFO messages, and is not considered part of the Tag Distribution process.

1.4.4 Tag Evaluation

Tag Evaluation describes the process of entities agreeing or disagreeing with the data represented on the tag and communicating this resolution back to the Tag Authority Service. This communication is accomplished through the use of the UPDATE message.

Upon receipt of an ASSESS message, the Tag Approval Service should present for evaluation the tag included in that message to the appropriate representative(s) of the Entity for which the Tag Approval Service provides. The representative(s) verify that the tag does indeed represent a valid transaction, and attempt to analyze the security impacts that may be incurred by that transaction. Based on their findings, the representative(s) utilize their Tag Approval Service to connect to the Tag Authority Service and communicate their decision as to the state of the tag in question via an UPDATE message. The Tag Authority Service determines whether or not their UPDATE message has met all the validity requirements defined in this document. If so, the Tag Authority Service processes the UPDATE message and responds with a SUCCESS to the Tag Approval Service; if not, the Tag Authority Service responds with a FAIL to the Tag Approval Service. Based on the change effected by the UPDATE message, the Tag Notification process may be initiated (to either all entities, or only the Tag Author).

1.4.5 Tag Notification

Tag Notification defines the process through which the Tag Authority Service informs one or more transaction path participants of changes to the state of a tag. This is accomplished in the NOTIFY message.

1.4.5.1 Entities with Rights to View the Complete Tag

Tag Notification begins either 1) immediately after the change of a tag's COMPOSITE State DateTime (all parties are notified), or 2) at the change of an Approval State to DENIED (Tag Author notification only). The Tag Authority Service connects to one or more of the various entities with rights to view the complete tag and effects an exchange of information with their Tag Approval or Agent Service utilizing

the NOTIFY message. Each service determines whether or not the message meets all the validity requirements defined in this document. If so, the service processes the message and responds with a SUCCESS to the Tag Authority Service; if not, the service responds with a FAIL to the Tag Authority Service.

1.4.6 Tag Forwarding

Tag Forwarding defines the process through which the Tag Authority Service sends tag information regarding flowing transactions to regional Security Coordinators and/or the Interchange Distribution Calculator. This is accomplished via the use of the IMPLEMENT message.

Tag Forwarding is initiated upon either the matriculation of a tag to a state that indicates transaction implementation or the modification of a tag that has already achieved such state (i.e., adjustment, termination, etc...). The Tag Authority Service connects to the registered forwarding service for the Load Control Area and effects an exchange of information with said service. The service determines whether or not the message meets all the validity requirements defined by its interface. If so, the service processes the message and responds with a SUCCESS to the Tag Authority Service; if not, the service responds with a FAIL to the Tag Authority Service.

1.4.7 Tag Querying

Tag Querying defines the process through which an entity may request the Tag Authority Service the most current state of a tag or a complete copy of a tag. These entities are defined only by the fact that they present a valid Tag ID and an associated valid Tag Key. This is accomplished via two methods: the STATUS message, and the DSTATUS message.

In both cases, the requesting service connects to the registered Tag Authority Service for the Load Control Area of the transaction represented by the tag in question and effects an exchange of information with the Tag Authority Service. The Tag Authority Service determines whether or not the message meets all the validity requirements defined in this document. If so, the service processes the message and responds with a SUCCESS to the requesting service; if not, the service responds with a FAIL to the requesting service.

It should be noted that excessive tag querying is not allowed. No one entity may request the status of a specific tag more than once every thirty (30) seconds. Authorities may optionally refuse to accept requests that exceed this figure. However, the Authority must be capable of responding to a request for a specific tag by a specific party at least once every thirty seconds.

1.4.8 Tag Cancellation

Tag Cancellation defines the process through which a Tag Author may withdraw, cancel, or prematurely terminate a tag. This is accomplished through the use of the CANCEL message.

Tag Agent Services issue cancellation requests to the Tag Authority Service. Upon receipt of a cancellation request, the Tag Authority Service verifies that the cancellation can indeed be effected and that the request is valid. If so, the Tag Authority Service processes the message and responds with a SUCCESS to the issuing Tag Agent Service; if not, the Tag Authority Service responds with a FAIL to the issuing Service. The time and nature of the request may determine the method that the Tag Authority Service uses to process the request. Tag Cancellation is concluded by initiating the Tag Notification process.

1.4.9 Tag Adjustment

Tag Adjustment defines a process through which a tag representing an implemented transaction may have its energy profile curtailed by the Load Control Area for a defined period of time. This is accomplished through the use of the ADJUST message.

Upon identification of the need for an adjustment to a tag, the issuing service connects to the Tag Authority Service associated with the tag and effects an exchange of information. The Tag Authority Service determines whether or not the message meets all the validity requirements defined in this document. If so, the Tag Authority Service processes the message and responds with a SUCCESS to the issuing service; if not, the Tag Authority Service responds with a FAIL to the issuing service.

Upon acceptance of an adjustment, the tag Authority Service notifies all parties to the transactions of the adjustment through the Tag Notification process.

For any given point in time, the lesser of the following two values will describe expected flow:

- The most recent adjustment affecting that point in time, or
- The original value specified for that point in time in the ENERGY table.

The following example and Figure 1.4.11 describes the adjustment process:

1. An on-peak tag is submitted that runs at 100 MW.
2. A curtailment occurs that requires the transaction to be cut by 40 MW.
3. After a few hours, the TLR is released, and another adjustment is made to reload the transaction to its original level of flow (100 MW)

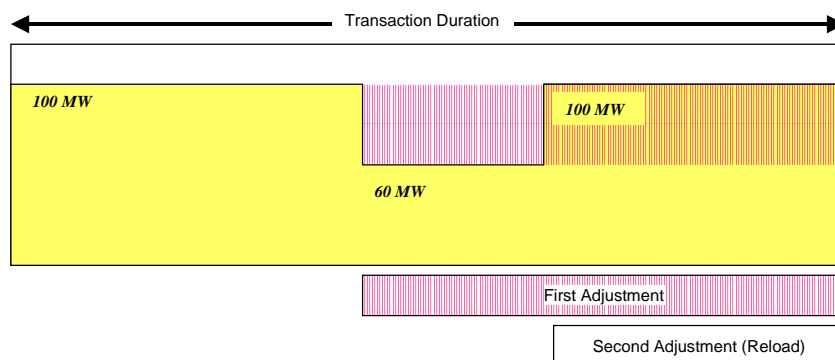


Figure 1.4.11

The intent of this function is to allow for both full and partial curtailments throughout the life of the tag.

1.4.10 Tag Replacement

Tag Replacement defines a process through which a tag author may tie the cancellation of one tag to the acceptance of another. This is accomplished by the inclusion of one or more tag Id's at the time of a tag submission.

Upon receipt of one or more Tag IDs in the Tag Agent Service's submission message, the Tag Authority Service verifies that the tags listed for replacement are in its possession and that they are valid replacement candidates. If so, the Tag Authority Service processes the message and responds with a SUCCESS to the Tag Agent Service; if not, the Tag Authority Service responds with a FAIL to the Tag Agent Service.

Upon acceptance of the tag containing the replacement request, the tags identified for replacement are effectively cancelled.

The following example and Figure 1.4.3 describes the replacement process:

1. The Tag Author submits a tag representing a 50MW On-Peak Schedule.
2. At 12:10, the Tag Author finds a different source that can come on line at 13:00.
3. At 12:30, the Tag Author submits a tag to REPLACE the previous tag, starting at 13:00.
4. At 12:40, the new tag is APPROVED by all parties
5. The new Tag is changed to a state of IMPLEMENT.
6. Immediately following the change to IMPLEMENT, the first tag is TERMINATED effective 13:00. All parties are notified of the termination, and the termination is forwarded to the IDC
7. All parties are notified of the new tags IMPLEMENT status, and the new tag is forwarded to the IDC.

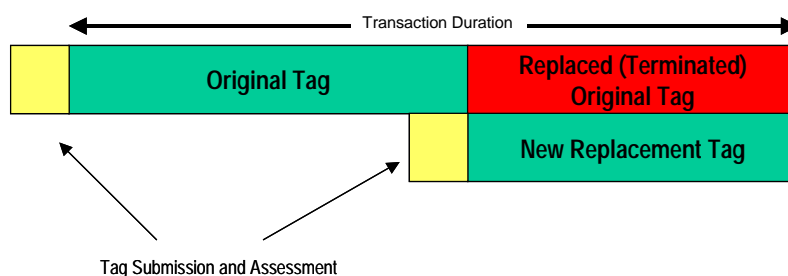


Figure 1.4.3

If the second tag has not been accepted, the first would remain in effect. One tag may replace multiple tags, provided the Sink CA remains the same. It should be noted that only the Tag Author may replace a tag. Tags may not be replaced if referenced by another outstanding replacement request.

1.4.11 Adjustment Notification

The IDC generates suggested adjustments to tagged transactions to mitigate transmission constraints during a TLR. Adjustment notification is the electronic process used to communicate those suggested tag adjustments to the LCA, GCA and Tag Author. When tagged interchange transactions are identified by the IDC to have their energy profile MWs adjusted, the IDC initiates the adjustment notification process by issuing an ADJUST_LIST message to the Tag Approval service of each GCA, LCA, and Authoring PSE associated with the tags to be adjusted. The ADJUST_LIST message indicates the Tag-ID, time frame, maximum suggested MW level at which the transaction should be allowed to continue, and a reason for the schedule change. Each ADJUST_LIST message will contain all the tags from the associated cut list for a particular entity (GCA, LCA, or Authoring PSE) as accepted by the LCA Security Coordinators. ADJUST_LIST messages can also be used to communicate single tag adjustments, such as the need to “hold” a tag that has been received after the reallocation deadline.

As a safe guard to insure the authenticity of an ADJUST_LIST message, Tag Approval and Tag Agent notification services may issue the ADJUST_CHECK message to the IDC upon receipt of an ADJUST_LIST message. On receipt of an ADJUST_CHECK message, the IDC will verify that it in fact has an IDC ADJUST_LIST message assigned to that unique key identifier and respond with a success message.

Note that the tag adjustment notification process merely communicates recommended actions to be taken on the specified tags. NERC Policy defines the process by which the parties to the transaction determine the actual level of adjustment to be effected on each tag and then communicate that adjustment through the Tag Adjustment process.

1.5 Functional Requirements

1.5.1 Tag Agent Service

1.5.1.1 Introduction

All Purchasing-Selling Entities (PSEs) and any other parties responsible for arranging Interchange Transactions shall communicate the necessary information as described in the Tag Data Model via the Tag Agent Service. The Tag Agent Service shall comply with all functional requirements set forth in this document. Users may elect to comply with these Tag Agent Service requirements using internally developed hardware/software; third party developed hardware/software, or third party subscription type services.

In the case of the use of any third party supplied hardware, software, or service, the user shall recognize that the user is ultimately and directly responsible for their compliance with all procedures, policies, and standards applicable to the arrangement of Interchange Transactions.

The Tag Agent Service shall provide facilities to:

- Accept and validate input tag/transaction data from the user.
- Generate all required tables and data elements and those optional tables and data elements necessary to completely specify the transaction as defined in the tag Data Model based on user input data.

- Assign and maintain the correspondence between each transaction's Tag ID and Tag Key.
- Identify the Tag Authority Service associated with the Sink (Load) Control Area in the transaction and electronically communicate the Tag ID, Tag Key, and tag data to that Tag Authority Service.
- For WSCC Participants, receive unsolicited information messages regarding tags that they are a party to but for which they have no direct approval rights.
- Query Tag Authority Services for the current status of each transaction submitted by the user (or transaction to which the user has both Tag ID and Tag Key).
- Provide the means for the user to terminate any pending or active transaction submitted by the user (or transaction to which the user has both Tag ID and Tag Key) at a specified point in time.
- Optionally receive unsolicited notification from the Tag Authority Service of changes in the status of a tag with which the user is a participant.

Information systems designed to provide more than one electronic tagging service (e.g., Tag Agent and Tag Authority Services) are free to use any internal or proprietary mechanisms to convey tag information between those functional services, but must still comply with all technical standards and protocols related to the exchange of transaction information with tagging related services provided by (or for) others.

1.5.1.2 Registry Usage

The Tag Agent Service shall be responsible for maintaining an updated list of all registered PSEs, Transmission Providers (TPs), Control Areas (CAs), and any other such entities whose identities must be uniquely specified in connection with the arrangement of an Interchange Transaction. The Master Registry of all such entities shall be maintained and available for downloading from NERC. The Tag Agent Service shall supply a procedure to allow updates from the Master Registry on demand or on a prescheduled interval.

At the option of the user, the Tag Agent Service may support the receipt of unsolicited transaction tag status notification messages sent by Tag Authority Services. To enable the delivery of these notification messages, the user must register the appropriate service identification information in the NERC Master Registry and be capable of receiving Tag Notification messages.

1.5.1.3 Tag Data Entry and Viewing

The Tag Agent Service shall provide a mechanism for the user to input, edit, and view Interchange Transaction tags, as well as perform all other functional requirements described herein. The exact nature of this user interface is beyond the scope of this document, with the exception that the user shall have the facilities to supply all transaction related information (as described in the Data Model) necessary to create complete, valid Interchange Transaction tags, as well as the interfaces to view those tags.

1.5.1.3.1 Tag IDs

Each Interchange Transaction tag submitted for approval to any Tag Authority Service by the Tag Agent Service shall be identified by a Tag ID, as described in the Data Model. This Tag ID must not be identical to any used previously to represent transactions with effective stop dates less than one year in the past.

1.5.1.3.2 Tag Keys

A unique Tag Key shall be associated with the initial transmission of a transaction tag from the Tag Agent Service to the appropriate Tag Authority Service. The Tag Agent Service shall be responsible for

generating this Tag Key consisting of the PSE code associated with the Tag Author and a unique 12 character token. All subsequent messages exchanged between the Tag Agent Service and Tag Authority Service in regard to this transaction tag shall refer to both the Tag ID and Tag Key assigned by the user's Tag Agent Service.

1.5.1.3.3 *Date and Time Handling*

The Tag Agent Service shall be responsible for the conversion of all date and time related input fields to Central Standard Time (CS) prior to the transaction tag being transferred to the Tag Authority Service. The use of 24:00 as a time representation is invalid. 00:00 is to be used to represent the beginning of a day. The Tag Agent user interface is free to accept and manage the conversion of any appropriate date/time formats at the discretion of the service provider. The internal representation of date and time within the Tag Agent is also entirely at the discretion of the service provider. However, all electronic transmittal of data shall be in CS time.

1.5.1.4 **Data Validation**

The Tag Agent Service shall ensure, prior to submission, that all data elements in a tag are legitimate and that no validation rules (as defined in the data model) have been broken.

1.5.1.5 **Function Implementation**

1.5.1.5.1 *Tag State Management*

The Tag Agent Service has no specific responsibilities with regard to managing tag state. However, the Tag Agent Service should be able to parse and store all tag state information received from the Tag Authority Service when it is either 1) communicated through Tag Notification or 2) returned in response to a Tag Agent Service issued message.

1.5.1.5.2 *Tag Submission*

Upon a user request to submit a transaction tag for implementation, the Tag Agent Service must first verify that the message it is preparing to send does not violate any rules defined in the data model.

Next, the Tag Agent Service must determine which Tag Authority Service shall be used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Agent Service to determine where to send the Submit request.

Once the appropriate Tag Authority Service is determined, the Tag Agent Service shall assign a unique Tag Key to the tag.

Next, the Tag Agent Service will connect to the Tag Authority Service and initiate the transfer of the transaction information to that Authority Service using the "SUBMIT" message, as defined in the Protocol Description. After the transfer is complete, the Tag Agent Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Agent Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to each of those approvers.

If the Tag Authority Service responds with a FAIL message, the Tag Agent Service is provided with a list of protocol errors and/or validation rule violations.

1.5.1.5.3 Tag Distribution

The Tag Agent Service may optionally implement an interface that accepts tag notifications. Currently, this functionality (with regard to Tag Distribution) is only available for tags that are either partially or fully implemented in WSCC. If implemented, the Tag Agent Service listener shall accept inbound connections and process them as follows:

First, the Tag Agent Service verifies the message meets all appropriate protocol and validation rules, as defined in the Protocol Description and Data Model. Tag Agents are distributed Tag Information through the use of the “INFO” message.

Next, the Tag Agent Service processes the message. If the Tag Agent Service finds no additional protocol errors or validation rule violations, the Tag Agent Service shall respond to the issuer of the notification that the notification attempt was a SUCCESS and use the contents of the message to update information currently stored by the Tag Agent Service.

If the Tag Agent Service did find errors or violations, the Tag Agent Service shall respond to the issuer with a FAIL message.

1.5.1.5.4 Tag Evaluation

The Tag Agent Service has no responsibilities with regard to the evaluation of tag information.

1.5.1.5.5 Tag Notification

The Tag Agent Service may optionally implement an interface that accepts tag notifications. If implemented, the Tag Agent Service listener shall accept inbound connections and process them as follows:

First, the Tag Agent Service verifies the message meets all appropriate protocol and validation rules, as defined in the Protocol Description and Data Model. Tag Agents are distributed Tag Information through the use of the “NOTIFY” message.

Next, the Tag Agent Service processes the message. If the Tag Agent Service finds no additional protocol errors or validation rule violations, the Tag Agent Service shall respond to the issuer of the notification that the notification attempt was a SUCCESS and use the contents of the message to update information currently stored by the Tag Agent Service.

If the Tag Agent Service did find errors or violations, the Tag Agent Service shall respond to the issuer with a FAIL message.

1.5.1.5.6 Tag Forwarding

The Tag Agent Service has no responsibilities with regard to forwarding of tag information.

1.5.1.5.7 Tag Querying

The Tag Agent Service issues queries for a tags status by first determining which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Agent Service to determine where to send the query request.

Once the appropriate Tag Authority Service is determined, the Tag Agent Service connects to the Tag Authority Service and initiates a request for the status of the tag to that Authority Service using either the

“STATUS” or “DSTATUS” messages, as defined in the Protocol Description. After the transfer is complete, the Tag Agent Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Agent Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the tag, and the current disposition of the tag with regard to of each of those approvers. If the original request was a “DSTATUS” request, the Tag Authority Service will additionally return all tag information known by that Tag Authority Service to relate to that tag.

If the Tag Authority Service responds with a FAIL message, the Tag Agent Service is provided with a list of protocol errors and/or validation rule violations.

1.5.1.5.8 Tag Cancellation

Upon a user’s request to cancel a previously submitted tag, the Tag Agent Service must first determine which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Agent Service to determine where to send the Cancel request.

Once the appropriate Tag Authority Service is determined, the Tag Agent Service connects to the Tag Authority Service and initiates the transfer of the cancellation of the tag to that Authority Service using the “CANCEL” message, as defined in the Protocol Description. After the transfer is complete, the Tag Agent Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Agent Service is provided with information that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to of each of those approvers.

If the Tag Authority Service responds with a FAIL message, the Tag Agent Service is provided with a list of protocol errors and/or validation rule violations.

1.5.1.5.9 Tag Adjustment

The Tag Agent Service has no responsibilities with regard to adjustment of tag information.

1.5.1.5.10 Tag Replacement

Users may request, at the time of submission, that a transaction tag submitted for implementation be used to replace one or more existing tags in states of IMPLEMENT, CONDITIONAL, ADJUSTED, or TERMINATED. This request is achieved through the use of optional fields defined in the SUBMIT message, as described in the Protocol Description. Other than this additional data, the Tag Replacement process is handled identically to the Tag Submission process (with regard to the functional implementation of the Tag Agent Service). Once a tag has been marked for replacement, it cannot be referred to by another replacement request while the first replacement request is in a state of PENDING, LATE, or ATTN_REQD.

1.5.1.5.11 Adjustment Notification

The Tag Agent Service may optionally implement an interface that accepts tag notifications. In addition to the standard Tag Notification functions described previously, registration of a Tag Agent notification URL will enable the forwarding of ADJUST_LIST messages from the Interchange Distribution Calculator. On receipt of an ADJUST_LIST message from the Interchange Distribution Calculator, the Tag Agent Service listener shall accept the message and take appropriate actions to alert the Tag Author

of impending adjustments to ongoing transactions. At the discretion of the Tag Agent listener service, the ADJUST_CHECK message may be directed to the IDC to verify that the ADJUST_LIST message was indeed authored by the IDC.

1.5.1.6 Failure Recovery

See Policy 3, Appendix 3A3 “Failure Procedures.”

1.5.1.7 Availability and Performance Requirements

There are no defined availability and performance requirements for a Tag Agent Service, other than the complete implementation of the function described within this document. Failure to meet the standards of this document shall be considered a violation of NERC Policy 3 and subject to investigation.

1.5.2 Tag Authority Service

1.5.2.1 Introduction

All entities responsible for Control Area (CA) operations shall provide the necessary hardware and software systems to implement the Tag Authority Service. The Tag Authority Service shall comply with all functional requirements set forth in this section. CAs may elect to comply with these Tag Authority Service requirements using internally developed hardware/software; third party developed hardware/software, or third party subscription type services.

In the case of the use of any third party supplied hardware, software, or service, the CA shall recognize that they are ultimately and directly responsible for their compliance with all procedures, policies, and standards applicable to the arrangement of Interchange Transactions.

The Tag Authority Service shall provide facilities to:

- Accept input tag/transaction data transferred in compliance with the Protocol Description from any Tag Agent Service.
- Provide immediate syntactical validation of the incoming data stream and respond accordingly.
- Identify all entities having approval rights over the transaction, and transfer the tag to the associated Tag Approval Services for evaluation (see Tag Distribution).
- Identify all entities entitled to an informational copy of the transaction, and transfer the tag to the associated Tag Agent Services (see Tag Distribution).
- Manage each transaction’s individual Approval and overall Composite Status based on communication with the Tag Agent and Tag Approval Services (see Tag State Management).
- Verify the identity of each approval entity attempting to approve or deny a tag based on the Tag ID and Tag Key assigned by the Tag Authority to that entity, and update the transaction’s Approval and Composite Status as appropriate.
- Verify the identity of any party attempting to cancel a tag based on Tag ID and Tag Key and process that request as appropriate.

- Verify the identity of each approval entity attempting to adjust the MW output of a tag based on the Tag ID and Tag Key assigned by the Tag Authority to that entity, and process that request as appropriate.
- Generate notification messages to Tag Approval and Tag Agent Services as appropriate.
- Respond to inquiries for transaction information made by Tag Agent or Approval Services.
- Forward all tags to a designated location as identified by information defined in the Master Registry associated with the Sink (Load) Control Area as appropriate.

Information systems designed to provide more than one electronic tagging service (e.g., Tag Authority and Tag Approval Services) are free to use any internal or proprietary mechanisms to convey tag information between those functional services, but must still comply with all technical standards and protocols related to the exchange of transaction information with tagging related services provided by (or for) others.

1.5.2.2 Registry Usage

The Tag Authority Service shall be responsible for maintaining an updated list of all registered PSEs, Transmission Providers (TPs), Control Areas (CAs), and any other such entities whose identities must be uniquely specified in connection with the arrangement of an Interchange Transaction. The Master Registry of all such entities shall be maintained and available for downloading from NERC. The Tag Authority Service shall supply a procedure to allow updates from the Master Registry on demand or on a prescheduled interval.

Each Control Area (CA) shall provide the necessary information to identify in the Master Registry who serves as their Tag Authority Service when that particular CA is referenced as the Sink (Load) Control Area (LCA) in a transaction tag.

1.5.2.3 Tag Data Entry and Viewing

The Tag Authority Service is primarily an automated manager of data that should require little intervention. However, certain events may require interaction. To this end, The Tag Authority Service shall provide a mechanism for a user to view Interchange Transaction tags and directly modify entity states, as well as perform all other functional requirements described herein. The exact nature of this user interface is beyond the scope of this document, with the exception that the user shall have the facilities to view all transaction related information (as described in the Data Model) necessary to represent a complete, valid Interchange Transaction tag.

1.5.2.3.1 IDC Tag Keys

Each Tag Authority Service shall be assigned a unique IDC Tag Key to be used when communicating with the IDC. All IMPLEMENT messages issued during the Tag Forwarding process must use this IDC Tag Key in order to be considered valid. The IDC will reject any IMPLEMENT messages without a valid IDC Tag Key. The IDC Tag Key should be considered confidential.

1.5.2.4 Data Validation

The Tag Authority Service shall ensure, upon receipt and prior to transmission of tag data, that all data elements in a tag are legitimate and that no validation rules (as defined in the data model) have been broken.

1.5.2.5 Function Implementation

1.5.2.5.1 Tag State Management

Tag State Management defines the Tag Authority Service's responsibility for changing and maintaining a record of a tag's various states. The Tag Authority Service must evaluate all Approval states for the tag upon any change in one single state, then potentially take action based on the results of that evaluation, and potentially change Composite state. Throughout the next sections, the specific rules for actions to take based on each evaluation are listed.

In addition to the state management rules defined in the sections below, there are other Composite state changes that may occur based on time, rather than on specific user action.

If a tag's Assessment Time (as defined in Policy 3) expires, the tag has a Composite State of PENDING and:

- all CA and TP Entities with DISTRIBUTE_METHODs of ASSESS have one of the following states: NULL, QUEUED, STUDY, APPROVED, or
- all TPs having BUYATMARKET requests for Product 0-NextHour Service have states of APPROVED,

the tag shall its Composite State changed to CONDITIONAL, and all rules following standard Tag Forwarding are to be observed. The Composite Table shall be modified as follows:

Composite State	CONDITIONAL
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	CA
Entity Code	<the registered NERC acronym of the Load Control Area>
Operator ID	NULL
Reason	NULL

If a tag's Assessment Time (as defined in Policy 3) expires, the tag has a Composite State of PENDING, and:

- one or more CA and TP Entities with DISTRIBUTE_METHODs of ASSESS have a state of DENIED, or
- any TPs having BUYATMARKET requests for Product 0-NextHour Service have one of the following states: NULL, QUEUED, STUDY, APPROVED,

the tag shall its Composite State changed to DENIED. The Composite Table shall be modified as follows:

Composite State	DENIED
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	<CA>
Entity Code	<the registered NERC acronym of the Load Control Area>
Operator ID	NULL
Reason	One or more entities DENIED this tag.

Tags that have a Composite State of LATE at the close of that tag's Assessment Time (as defined in Policy 3) shall have their Composite State changed to DENIED. The Composite Table shall be modified as follows:

Composite State	DENIED
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	CA
Entity Code	<the registered NERC acronym of the Load Control Area>
Operator ID	NULL
Reason	Late

Tag Authority Services must also provide the capability for an operator to manually override selected Approval States. The Tag Authority Service is required to allow both the COMM_FAIL and INVALID states to be overridden to the states of either APPROVED or DENIED. Override of an Approval state should modify the appropriate Approval Entity's STATUS table record(s) as follows:

Entity Type	<the entity type of the entity being overridden>
Entity Code	<the registered NERC acronym of the entity being overridden>

Entity State	<state requested by the entity being overridden>
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	LCA Override
Reason	<previous state, either COMM_FAIL or INVALID>
Distribute_Method	<no change>
Notify_Method	<no change>

Upon manual override of an Approval State, all rules following standard Tag Evaluation are to be observed, including the evaluation of the COMPOSITE table and any further state changes necessitated by the evaluation.

Any change in Composite State should create notifications to all entities in the STATUS table with a Notify Method of 1 or 2.

1.5.2.5.2 *Tag Submission*

Tag Submission defines the process through which a Tag Agent Service sends tag information to the Tag Authority Service. The Tag Authority Service listener shall accept inbound connections and process them as follows:

First, the Tag Authority Service verifies the message meets all appropriate protocol and validation rules, as defined in the Protocol Description and Data Model. Agent/Authority credentials should be verified, and the Tag Key should be noted for storage. At this point, the Tag Authority Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be a SUBMIT message. If the SUBMIT message contains REPLACE information, the actions described in Tag Replacement must be executed as well.

Next, the Tag Authority Service processes the message. The first step in this processing is to build that STATUS table. The STATUS table is built using the structure defined in the Data Model. Records are instanced and initialized using the following strategy:

The first record shall be that of the Tag Author, as defined in the REQUESTOR table. Default values for this record are:

Entity Type	PSE
Entity Code	<the registered NERC acronym of the tag author>
Entity State	NULL
State Date Time	NULL
Submit Date Time	<the current date and time>
Operator ID	<the identifier associated with the user that authored the tag>
Reason	NULL
Distribute_Method	NULL
Notify_Method	NOTIFY if the PSE has a registered listener URL per the NERC registry, otherwise NULL

The following records are built by including the entities from the PROVIDER table. The process identifies in order all the Purchasing Selling Entities listed in the Provider table. No Purchasing Selling Entity may appear twice in the STATUS table. Default values for these records are:

Entity Type	PSE
Entity Code	<the registered NERC acronym of the entity>
Entity State	NULL
State Date Time	NULL
Submit Date Time	NULL
Operator ID	NULL
Reason	NULL
Distribute_Method	INFO if any CAs are in WSCC; otherwise NULL
Notify_Method	NOTIFY if any CAs are in WSCC; otherwise NULL

The next records are also built by reading entities from the PROVIDER table. The process identifies, in order, all Transmission Providers listed in the Provider table. No Transmission Provider entity may appear twice in the STATUS table. Default values for this record are:

Entity Type	TP
Entity Code	<the registered NERC acronym of the Transmission Provider>
Entity State	PENDING
State Date Time	<the current date and time>
Submit Date Time	NULL
Operator ID	NULL
Reason	NULL
Distribute_Method	ASSESS
Notify_Method	NOTIFY

Next, the process identifies, in order, all the Control Areas listed in the Provider table. No Control Area entity may appear twice in the STATUS table. Default values for this record are:

Entity Type	CA
Entity Code	<the registered NERC acronym of the Control Area>
Entity State	PENDING
State Date Time	<the current date and time>
Submit Date Time	NULL
Operator ID	NULL
Reason	NULL
Distribute_Method	ASSESS
Notify_Method	NOTIFY

A final record must be added to identify the forwarding service associated with the Load Control Area in the Master Registry. Default values for this record are:

Entity Type	SC
-------------	----

Entity Code	<the registered NERC acronym of the Load Control Area>
Entity State	NULL
State Date Time	NULL
Submit Date Time	NULL
Operator ID	NULL
Reason	NULL
Distribute_Method	NULL
Notify_Method	NULL

Upon completion of the building of the STATUS table, the Tag Authority Service must build the COMPOSITE table. The COMPOSITE table is built using the structure defined in the Data Model. The COMPOSITE table has one record. If the tag is received earlier or equal to the required submission deadline as defined in Policy 3, the record is initialized using the following default values:

Composite State	PENDING
Composite DateTime	<the current date and time>
Start DateTime	<derived using instructions contained in the Data Model>
Stop DateTime	<derived using instructions contained in the Data Model>
Entity Type	PSE
Entity Code	<the registered NERC acronym of the Tag Author>
Operator ID	<the identifier associated with the user that authored the tag>
Reason	NULL

If the tag is received after the deadline, but less than one hour past the derived start time of the tag, the record is initialized using the following default values:

Composite State	LATE
Composite DateTime	<the current date and time>
Start DateTime	<derived using instructions contained in the Data Model>
Stop DateTime	<derived using instructions contained in the Data Model>
Entity Type	PSE
Entity Code	<the registered NERC acronym of the Tag Author>
Operator ID	<the identifier associated with the user that authored the tag>
Reason	NULL

The date and time entered into the Composite Date Time field should initially be the same as that entered in the date time field for each of the approval entities and is the basis on which evaluation time period is determined.

If the Tag Authority Service finds no additional protocol errors or validation rule violations during the above processes, the Tag Authority Service shall respond to the issuer of the request that the Submission attempt was a SUCCESS. At this time, the Tag ID, Tag Author's Tag Key, and Tag Data should be stored and the Tag Distribution process started.

If the Tag Authority Service did find errors or violations, the Tag Authority Service shall respond to the issuer with a FAIL message.

1.5.2.5.3 **Tag Distribution**

Tag Distribution defines the process in which the Tag Authority Service sends requests for Tag Evaluation to all parties with evaluation responsibilities. The process begins by examining the STATUS table.

For each unique entity with a Distribute_Method of ASSESS, the process below should occur. Uniqueness is defined as the distinct value created when combining Entity Code and registered Approval URL from the Master Registry for that Entity. In this manner, no entity with both the same Entity Code and Approval URL shall receive duplicate requests to ASSESS.

Determine the registered Tag Approval Service for the entity based on the information contained in the Master Registry. Once the appropriate Tag Approval Service is determined, the Tag Authority Service shall assign a unique Tag Key to the entity for use in interacting with the tag.

Next, the Tag Authority Service will connect to the Tag Approval Service and initiate the transfer of the transaction information to that Tag Approval Service using the "ASSESS" message, as defined in the Protocol Description. After the transfer is complete, the Tag Authority Service will await the disposition of its request.

If the Tag Approval Service responds with a SUCCESS message, the Tag Authority Service should modify the state of that entity in the Status table as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	QUEUED
State Date Time	<the current date and time>
Submit Date Time	<the current date and time>
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>
Notify_Method	<no change>

If the Tag Approval Service responds with a FAIL message, the Tag Authority Service is provided with a list of protocol errors and/or validation rule violations. The Tag Authority Service should modify the state of that entity in the Status table as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	INVALID
State Date Time	<the current date and time>
Submit Date Time	NULL
Operator ID	NULL
Reason	<first 80 characters of the error message after the word FAIL>
Distribute_Method	<no change>
Notify_Method	<no change>

If the Tag Authority Service is unable to contact the entity, as defined in the Protocol Specification, then the Tag Authority Service should modify the state of the entity as described below:

Entity Type	<no change>
-------------	-------------

Entity Code	<no change>
Entity State	COMM_FAIL
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>
Notify_Method	<no change>

In the event of a state change to COMM_FAIL or INVALID, the Composite State should also be modified as described below:

Composite State	ATTN_REQD
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	<set to the entity type of the entity with the state of INVALID or COMM_FAIL. If entity is both a CA and a TP, list CA.>
Entity Code	<the registered NERC acronym of the entity with the state of INVALID or COMM_FAIL >
Operator ID	NULL
Reason	NULL

Finally, any change in Entity State to COMM_FAIL or INVALID should initiate the Tag Notification Process to the Tag Author (it is not necessary to send Notification to any other party).

For each unique entity listed with a Distribute_Method of INFO, the following process should occur. Uniqueness is defined as the distinct value created when combining Entity Code and registered Tag Agent Notification URL from the Master Registry for that Entity. In this manner, no entity with both the same Entity Code and Agent Notification URL shall receive duplicate INFO messages.

Determine the registered Tag Agent Service for the entity based on the information contained in the Master Registry. Once the appropriate Tag Agent Service is determined, the Tag Authority Service shall assign a unique Tag Key to the entity for use in interacting with the tag.

Next, the Tag Authority Service will connect to the Tag Agent Service and initiate the transfer of the transaction information to that Tag Agent Service using the “INFO” message, as defined in the Protocol Description. After the transfer is complete, the Tag Authority Service will await the disposition of its request.

If the Tag Agent Service responds with a SUCCESS message, the Tag Authority Service should modify the state of that entity in the Status table as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	QUEUED
State Date Time	<the current date and time>
Submit Date Time	<the current date and time>
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>
Notify_Method	<no change>

If the Tag Agent Service responds with a FAIL message, the Tag Authority Service is provided with a list of protocol errors and/or validation rule violations. The Tag Authority Service should modify the state of that entity in the Status table as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	INVALID
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	<first 80 characters of the error message after the word FAIL>
Distribute_Method	<no change>
Notify_Method	<no change>

If the Tag Authority Service is unable to contact the entity as defined in the Protocol Specification, or the entity does not have a registered Tag Agent Service, then the Tag Authority Service should modify the state of the entity as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	COMM_FAIL
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	No response or no registered listener.
Distribute_Method	<no change>
Notify_Method	<no change>

1.5.2.5.4 *Tag Evaluation*

The Tag Evaluation process defined the manner in which Tag Approval Services communicate their evaluation decision back to the Tag Authority Service and effect a change in state. The Tag Authority Service listener shall accept inbound connections and process them as follows:

First, the Tag Authority Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Authority Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be an UPDATE message

Next, the Tag Authority Service processes the message. The Tag Authority verifies that all fields required with the UPDATE message are present and valid. If any of the following conditions are true, then the tag may not be updated and a FAIL message should be returned:

- the Tag Authority cannot associate both the Tag ID and Tag Key with an existing tag, or
- the request lacks any of the data defined in the Protocol Specification, or
- the request represents a STUDY or DENIED request and no reason has been specified, or
- the Composite State of the tag to be updated is not PENDING, LATE, or ATTN_REQD.

If the Tag Authority Service finds errors or violations, the Tag Authority Service shall respond to the issuer with a FAIL message and message processing should end.

If no errors are found, the Tag Authority Service shall modify the state(s) of the appropriate Entities in the Status table as described below. The Appropriate Entities shall be those to which the presented Tag Key was assigned. In this manner, an Entity with approval rights over both Control Area functions and Transmission Provider functions may approve both items with only one message.

Entity Type	<no change>
Entity Code	<no change>
Entity State	<as indicated by the UPDATE message>
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	<the identifier associated with the user that issued the UPDATE request>
Reason	<the REASON for the UPDATE, if provided (NOTE: As described above, this is a REQUIRED message attribute for update messages that change state to DENIED or STUDY)>
Distribute_Method	<no change>
Notify_Method	<no change>

The Tag Authority Service shall respond to the issuer of the request that the Update attempt was a SUCCESS.

Should a party set their state to DENIED, the Tag Authority Service should implement a Tag Notification message to the Tag Agent Service of the Tag Author. However, this denial in itself does not necessitate a change to the COMPOSITE table.

Upon any change in Approval State, the Tag Authority Service should evaluate the tag for a potential state change. Tags that, during evaluation, have all of their CA and TP Approval entities with a DISTRIBUTION_METHOD of ASSESS States changed to APPROVED shall immediately have their Composite State changed to IMPLEMENT. The Composite Table shall be modified as follows:

Composite State	IMPLEMENT
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	CA
Entity Code	<the registered NERC acronym of the Load Control Area>
Operator ID	NULL
Reason	NULL

Following a Composite State Change to IMPLEMENT, the Tag Authority should begin Tag Notification to all parties, Tag Replacement Completion (if appropriate), and Tag Forwarding to the IDC.

1.5.2.5.5 **Tag Notification**

Tag Notification describes the process in which the Tag Authority service alerts one or more participants of changes in a tags state. Generally, Tag Notification to all parties is begun on any change to the COMPOSITE State Date Time. Tag Notification is sent only to the Tag Author on any STATUS Entity State change to DENIED, COMM_FAIL, or INVALID.

Entities listed in the Status Table with a Notify Method of NOTIFY are notified through the use of the NOTIFY message. The Tag Authority determines the registered Approval or Agent Notification Service URLs for the entity or entities that require notification though use of the Master Registry.

Once the appropriate service is determined, Tag Authority Service will connect to that service and initiate the transfer of the transaction information to service using the “NOTIFY” message, as defined in the Protocol Description. After the transfer is complete, the Tag Authority Service will await the disposition of its request.

If the service responds with a SUCCESS message, the Tag Authority Service shall continue processing as normal.

If the service responds with a FAIL message, the Tag Authority Service shall make note of the failure in its internal logs and continue processing as normal. A FAIL message does not necessitate a change in Entity or Composite state.

1.5.2.5.6 **Tag Forwarding**

Tag Forwarding defines the process in which a Tag Authority Service forwards an implemented tag (a tag with a state of IMPLEMENT, CONDITIONAL, CANCELLED, TERMINATED, or ADJUSTED) to the Load Control Area’s forward service as listed in the Master Registry (if one is so defined). Test tags should not be forwarded

First, the Tag Authority Service examines the Master Registry and verifies that the listed Load Control Area for the tag has a forwarding URL defined. If not, the Tag Forwarding process should be aborted without error.

If the Tag Forwarding URL is present, the Tag Authority Service should modify the data of the “SC” record in the STATUS table as follows:

Entity Type	<no change>
Entity Code	<no change>
Entity State	PENDING
State Date Time	<the current date and time>
Submit Date Time	NULL
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>

Notify_Method	<no change>
---------------	-------------

Once modified, the Tag Authority Service will connect to that URL and initiate the transfer of the transaction information to service using the “IMPLEMENT” message, as defined in the Protocol Description. It should be noted that the Tag Authority Service must use its unique “IDC Tag Key” when presenting its request to the IDC. After the transfer is complete, the Tag Authority Service will await the disposition of its request.

If the service responds with a SUCCESS message, the Tag Authority Service shall modify the data of the “SC” record as follows:

Entity Type	<no change>
Entity Code	<no change>
Entity State	APPROVED
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>
Notify_Method	<no change>

If the service responds with a FAIL message, the Tag Authority Service shall modify the data of the “SC” record as follows:

Entity Type	<no change>
Entity Code	<no change>
Entity State	INVALID
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	<first 80 characters of the error message after the word FAIL>
Distribute_Method	<no change>
Notify_Method	<no change>

If the Tag Authority Service is unable to contact the entity, as defined in the Protocol Specification, then the Tag Authority Service should modify the state of the entity as described below:

Entity Type	<no change>
Entity Code	<no change>
Entity State	COMM_FAIL
State Date Time	<the current date and time>
Submit Date Time	<no change>
Operator ID	NULL
Reason	NULL
Distribute_Method	<no change>
Notify_Method	<no change>

If an error occurs in the transfer of data to the forwarding entity, the Tag Authority Service shall alert an operator of the condition. However, no change in Composite State should be made as a result of a failure to transfer data to the forwarding entity.

Tag Authority Services should provide mechanisms allowing for on-demand forwarding of a tag to the IDC. Should an Authority operator have reason to believe the Tag Forwarding process has failed, the operator shall use this feature to initiate the tag forwarding sequence again to ensure all tags are received by the IDC.

1.5.2.5.7 Tag Querying

Tag Querying defines how the Authority behaves due to requests for information from various entities. The Tag Authority Service listener shall accept inbound connections and process them as follows:

First, the Tag Authority Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Authority Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be either a STATUS or DSTATUS message.

Next, the Tag Authority Service processes the message. The first step of this process is to determine whether or not the tag associated with the request may be processed and that all required fields for the request are present and valid. If the Tag Authority Service cannot associate both the Tag ID and Tag Key with an existing tag, then the tag may not be retrieved and a FAIL message should be returned.

If the message request is valid, then the Tag Authority Service should respond with a SUCCESS message, as defined in the Protocol Description.

At the provider's option, excessive queries for tag status may be refused. Further information is provided in Section 1.4.7.

1.5.2.5.8 **Tag Cancellation**

Tag Cancellation defines how the Tag Authority Service behaves due to cancellation requests from the Tag Agent Service at various points in time. The Tag Authority Service listener shall accept inbound connections and process them as follows:

First, the Tag Authority Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Authority Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be a CANCEL message

Next, the Tag Authority Service processes the message. The first step of this process is to determine whether or not the tag associated with the request may be cancelled and that all required fields for the request are present and valid. If any of the following conditions are true, then the tag may not be cancelled and a FAIL message should be returned:

- the Tag Authority cannot associate both the Tag ID and Tag Key with an existing tag, or
- the request lacks any of the data defined in the Protocol Specification, or
- the Composite State of the tag to be cancelled is DENIED, WITHDRAWN, or CANCELLED, or
- the effective date/time of the cancellation is less than the tag's start date/time (as maintained in Composite table Start_DateTime field), or
- the effective date/time of the cancellation is greater than the tag's stop date/time (as maintained in Composite table Stop_DateTime field), or
- the effective date/time of the cancellation is in the past (prior to current time in CS), or
- the Composite State of the tag to be cancelled is PENDING, LATE, or ATTN_REQ, and the effective date/time of the cancellation is not equal to the tag's start date/time (as maintained in Composite table Start_DateTime field), or
- the Tag Key presented is not that of the Tag Author.

If the Tag Authority Service finds errors or violations, the Tag Authority Service shall respond to the issuer with a FAIL message and message processing should end.

Once determined to be a valid cancel request, the new Composite State of the tag is based on the current Composite State and the effective date/time of the cancellation relative to the start date/time of the tag.

- If the Composite state is PENDING, LATE, or ATTN_REQD, and the effective date/time of the cancellation is equal to the tag's start date/time, the Tag Authority shall set the Composite State to WITHDRAWN.
- If the Composite State is IMPLEMENT, TERMINATED, ADJUSTED, or CONDITIONAL, and the effective date/time of the cancellation is equal to the tag's start date/time, the Tag Authority shall set the composite state to CANCELLED.
- If the Composite State is IMPLEMENT, TERMINATED, ADJUSTED, or CONDITIONAL, and the effective date/time of the cancellation is greater than the tag's start date/time, the Tag Authority shall set the composite state to TERMINATED.

Based on the appropriate state chosen above, the Composite table should be modified in the following manner:

Composite State	<as described above>
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<set to the Cancel Date Time specified in the Cancel request>
Entity Type	PSE
Entity Code	<the registered NERC acronym of the Tag Author>
Operator ID	<the identifier associated with the user that issued the CANCEL request>
Reason	<the REASON for the cancellation, as specified in the CANCEL request>

The Tag Authority Service shall respond to the issuer of the request that the Cancellation attempt was a SUCCESS. At this time the Tag Notification process should be started.

Any successful CANCEL request should start the Tag Notification process for all entities in the STATUS table with a Notify Method of NOTIFY. Tag Forwarding should occur if the state of the tag is TERMINATED or CANCELLED.

1.5.2.5.9 *Tag Adjustment*

Tag Adjustment defines the process in which the Tag Authority accepts and applies adjustments to an existing tag. The Tag Authority Service listener shall accept inbound connections and process them as follows:

First, the Tag Authority Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Authority Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be an ADJUST message

Next, the Tag Authority Service processes the message. The first step of this process is to determine whether or not the tag associated with the request may be adjusted and that all required fields for the request are present and valid. If any of the following conditions are true, then the tag may not be adjusted and a FAIL message should be returned:

- the Tag Authority cannot associate both the Tag ID and Tag Key with an existing tag, or
- the Tag Key presented is not associated with the Load Control Area, or
- the request lacks any of the data defined in the Protocol Specification, or
- the Composite State of the tag to be adjusted is not IMPLEMENT, CONDITONAL, TERMINATED, or ADJUSTED, or
- The start time of the time period being adjusted is more than one hour in the past.

If the Tag Authority Service finds errors or violations, the Tag Authority Service shall respond to the issuer with a FAIL message and message processing should end.

If the adjustment is valid, then an adjust record is created to represent the adjustment. If there is no ADJUST table present, it is created; otherwise, the record is appended to the existing ADJUST table. The adjust record is populated as follows:

Adjust Date Time	<the current date and time>
Start Date Time	<the start date and time specified in the ADJUST message>
Stop Date Time	<the stop date and time specified in the ADJUST message>
Megawatt Cap	<the megawatt Cap specified in the ADJUST message>
Entity Type	<set to the entity type of the entity issuing the ADJUST request>
Entity Code	<the registered NERC acronym of the entity issuing the ADJUST request>
Operator ID	<the identifier associated with the user that issued the ADJUST request>
Reason	<the REASON for the adjustment, as specified in the ADJUST request>

Any successful ADJUST request also modifies the COMPOSITE table in the following fashion:

Composite State	ADJUSTED
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<no change>
Entity Type	<set to the entity type of the entity issuing the ADJUST request. If entity is both a CA and a TP, list CA.>
Entity Code	<the registered NERC acronym of the entity issuing the ADJUST request>
Operator ID	<the identifier associated with the user that issued the ADJUST request>
Reason	<the REASON for the adjustment, as specified in the ADJUST request>

Any successful adjustment should initiate the Tag Notification process for all entities in the STATUS table with a Notify Method of NOTIFY. Tag Forwarding to the IDC should occur as well.

1.5.2.5.10 Tag Replacement

Should a SUBMIT request be accompanied by REPLACE information, it is assumed that the Tag Agent Service is requesting a Tag Replacement. There are two parts to Tag Replacement: Replacement Initialization and Replacement Completion.

Replacement Initialization defines the actions taken by a Tag Authority Service upon receipt of a Tag Replacement request. When the request is received, the Tag Authority Service must first verify the possession of the tags in question and that the keys supplied for those tags are indeed valid Tag Author keys. The Tag Authority Service should respond with a FAIL message if:

- any of these tags are not found, or
- their keys are invalid, or
- their Composite stop date/time is in the past, or
- their composite state is not IMPLEMENT, CONDITIONAL, ADJUSTED, or TERMINATED, or
- they are currently referenced by another replacement tag, which is in state of PENDING, LATE, or ATTN_REQD.

Otherwise, the Tag Authority Service continues and validates the message in the same manner as a standard Tag Submission.

Following successful validation, the Tag Authority Service examines each tag to be replaced. If a tag to be replaced has a COMPOSITE Stop Date Time that is earlier than the COMPOSITE Start Date Time of the new tag, it is ignored. However, if the two times are equal or the new Start Date Time is earlier than the old Stop Date Time, the Tag Authority builds the REPLACE table, as defined in the Data Model. The REPLACE table is populated with the Tag ID of each tag to be replaced that has a COMPOSITE Stop Date Time later than or equal to the COMPOSITE Start Date Time of the replacing tag. The REPLACE table becomes an additional table associated with the replacing tag.

The Tag Authority Service then returns to the normal processing of a Tag Submission. After completion of the Tag Submission process, the tag goes through the normal Tag Distribution and Evaluation process.

Replacement Completion occurs at the point in time that the Replacing Tag reaches a state of either IMPLEMENT or CONDITIONAL. Upon achievement of either of these states, the Tag Authority Service shall retrieve from the REPLACE table the list of all tags being replaced and modify their Composite Tables as follows:

Composite State	CANCELLED, or TERMINATED (as appropriate, based on the current state of the tag being replaced, the current state, and the effective time of the replacement)
Composite DateTime	<the current date and time>
Start DateTime	<no change>
Stop DateTime	<the later of either the start date/time of the Replacing Tag or the Start Date/Time of the tag being replaced>
Entity Type	PSE
Entity Code	<the registered NERC acronym of the Purchasing Selling Entity>
Operator ID	<the identifier associated with the user that authored the replacing tag>

	tag>
Reason	REPLACED BY <the Tag ID of the Replacement Tag>

Under no circumstances may a tag being replaced have its Stop Date Time increased to a point further in the future than it's current Stop Date Time. Should this instance occur, it should be interpreted to mean the tag being replaced was cancelled directly by the Tag Author. If the new Stop Date Time is equal to the old Stop Date Time, the old date time and composite information should remain. In both cases, no error should occur, and the attempt to modify the composite table should be aborted.

Following modification of the replaced tags Composite table, the Tag Notification and Tag Forwarding processes is initiated for each of the replaced tags. Upon completion, the Tag Notification and Tag Forwarding processes is initiated for the new tag replacing the ones that were just modified.

1.5.2.6 Failure Recovery

Tag Authorities shall be responsible for alerting their operators to any failure of the message distribution system (any outbound messages from the Tag Authority). The manner in which these and other messaging failures are to be handled is described within NERC Policy. See Policy 3, Appendix 3A3 "Failure Procedures."

1.5.2.7 Availability and Performance Requirements

Until otherwise established by the Industry, the target availability requirement for a Sink (Load) Control Area's Tag Authority Service shall be 99.5%. This value will be reviewed and updated as appropriate.

The Tag Authority Service shall be required to complete the transfer of all new tags (via the ASSESS message to the Tag Approval Services), and the notification of all PSEs, approval entities, and forwarding services on changes to the Composite Status (via the NOTIFY message) within certain maximum time limits as established by the Industry. Until such limits are established, the Tag Authority shall be required to complete the delivery or exhaust the necessary attempts to deliver these messages within two (2) minutes.

1.5.3 Tag Approval Service

1.5.3.1 Introduction

All entities that may have "approval rights" over any Interchange Transaction shall provide the necessary hardware and software systems to implement the Tag Approval Service. The Tag Approval Service shall comply with all functional requirements set forth in this section. Approval entities may elect to comply with these Tag Approval Service requirements using internally developed hardware/software; third party developed hardware/software, or third party subscription type services.

In the case of the use of any third party supplied hardware, software or service, the approval entity (CA or TP) shall recognize that they are ultimately and directly responsible for their compliance with all procedures, policies and standards applicable to the arrangement of Interchange Transactions.

Tag Approval Service shall be responsible for providing the following functions:

- Accept input tag/transaction data transferred in compliance with the Protocol Description from any Tag Authority Service.
- Provide immediate syntactical validation of the incoming data stream and respond accordingly (i.e., provide for positive acknowledgement of receipt of the tag).
- Communicate approval, denial, study, and adjustment information to the Tag Authority managing the transaction electronically in compliance with the Protocol Description.
- Receive notification messages from the Tag Authority.
- Query the appropriate Tag Authority for the current status of each transaction submitted for approval.

Information systems designed to provide more than one electronic tagging service (e.g., Tag Authority and Tag Approval Services) are free to use any internal or proprietary mechanisms to convey tag information between those functional services, but must still comply with all technical standards and protocols related to the exchange of transaction information with tagging related services provided by (or for) others.

1.5.3.2 Registry Usage

The Tag Approval Service shall be responsible for maintaining an updated list of all registered PSEs, Transmission Providers (TPs), Control Areas (CAs), and any other such entities whose identities must be uniquely specified in connection with the arrangement of an Interchange Transaction. The Master Registry of all such entities shall be maintained and available for downloading from NERC. The Tag Approval Service shall supply a procedure to allow updates from the Master Registry on demand or on a prescheduled interval.

Each Control Area (CA) and Transmission Provider (TP) shall provide the necessary information to identify in the Master Registry who serves as their Tag Approval Service when that particular CA or TP is deemed to have approval rights for a particular transaction tag.

1.5.3.3 Tag Data Entry and Viewing

The Tag Approval Service is the main interface through which entities with approval rights to a tag alert the tag author and each other of their decisions to approve or deny a tag as being a valid representation of a scheduled transaction. To this end, The Tag Approval Service shall provide a mechanism for a user to view Interchange Transaction tags and directly modify the entity state(s) over which they have control, as well as perform all other functional requirements described herein. The exact nature of this user interface is beyond the scope of this document, with the exception that the user shall have the facilities to view all transaction related information (as described in the Data Model) necessary to represent a complete, valid Interchange Transaction tag.

1.5.3.4 Data Validation

The Tag Approval Service shall ensure, upon receipt and prior to transmission of tag data, that all data elements in a tag are legitimate and that no validation rules (as defined in the data model) have been broken.

1.5.3.5 Function Implementation

1.5.3.5.1 *Tag State Management*

The Tag Approval Service has no specific responsibilities with regard to managing tag state. However, the Tag Approval Service should store any changes in tag state made by the Tag Authority Service when they are either 1.) Communicated through Tag Notification or 2.) Returned in response to a Tag Approval Service issued message.

1.5.3.5.2 *Tag Submission*

The Tag Approval Service has no specific responsibilities with regard to tag submission.

1.5.3.5.3 *Tag Distribution*

Tag Distribution defines the manner in which the Tag Approval Service accepts and processes incoming tag evaluation requests. The Tag Approval Service listener shall accept inbound connections and process them as follows:

First, the Tag Approval Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Approval Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be an ASSESS message.

Next, the Tag Approval Service processes the message. If the Tag Approval Service finds no additional protocol errors or validation rule violations during the processing of the message, the Tag Approval Service shall respond to the issuer of the request that the Submission attempt was a SUCCESS. At this time, the Tag ID, Tag Key, and Tag Data should be stored and the Tag Evaluation process started.

If the Tag Authority Service did find errors or violations, the Tag Approval Service shall respond to the issuer with a FAIL message.

1.5.3.5.4 *Tag Evaluation*

Tag Evaluation defines the manner in which a Tag Approval Service communicates the decision of its operator (with regard to the accuracy of a previously supplied tag) back to the Tag Authority Service for that particular tag.

Upon a user's request to update the status a tag, the Tag Approval Service must first determine which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the Update request.

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the transfer of the update of status of the tag to that Authority Service using the "UPDATE" message, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with information that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to of each of those approvers.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.5.5 Tag Notification

Tag Notifications pass information about the most current state of a particular tag to the Tag Approval Service. The Tag Approval Service listener shall accept inbound connections and process them as follows:

First, the Tag Approval Service verifies the message meets all appropriate protocol and validation rules, as defined in the Protocol Description and Data Model.

Next, the Tag Approval Service processes the message. If the Tag Approval Service finds no additional protocol errors or validation rule violations, the Tag Approval Service shall respond to the issuer of the notification that the notification attempt was a SUCCESS and use the contents of the message to update information currently stored by the Tag Approval Service.

If the Tag Approval Service did find errors or violations, the Tag Approval Service shall respond to the issuer with a FAIL message.

1.5.3.5.6 Tag Forwarding

The Tag Approval Service has no specific responsibilities with regard to tag forwarding.

1.5.3.5.7 Tag Querying

The Tag Approval Service issues queries for a tags status by first determining which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the query request

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the query for information about the tag to that Authority Service using either the “STATUS” or “DSTATUS” messages, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to of each of those approvers. If the original request was a “DSTATUS” request, the Tag Authority Service will additionally return all tag information known by that Tag Authority Service to relate to that tag.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.5.8 Tag Cancellation

The Tag Approval Service has no specific responsibilities with regard to tag cancellation.

1.5.3.5.9 Tag Adjustment

Tag Adjustment defines the manner in which a Load Control Area's Tag Approval Service communicates the desire of its operator to modify for a period of time the represented megawatt flow in a particular tag back to the Tag Authority Service for that particular tag.

Upon a user's request to adjust a tag, the Tag Approval Service must first determine which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the Adjust request.

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the transfer of the Adjustment to the tag to that Authority Service using the "ADJUST" message, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to each of those approvers, as well as the confirmation of the accepted adjustment.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.5.10 Tag Replacement

The Tag Approval Service has no specific responsibilities with regard to tag replacement. However, it is the responsibility of the Tag Approval Service to appropriately highlight the existence of the tag's REPLACE table so that the operator can evaluate the tag in light of the pending replacement.

1.5.3.5.11 Adjustment Notification

The Tag Approval Service receives ADJUST_LIST messages from the Interchange Distribution Calculator. The Tag Approval Service shall accept the ADJUST_LIST message and take appropriate actions to alert the Tag Approval entity of impending adjustments to ongoing transactions. At the discretion of the Tag Approval Service, the ADJUST_CHECK message may be directed to the Interchange Distribution Calculator to verify that the message was indeed authored by the IDC.

1.5.3.6 Failure Recovery

See Policy 3, Appendix 3A3 "Failure Procedures."

1.5.3.7 Availability and Performance Requirements

Until otherwise established by the Industry, the target availability requirement for a provider's Tag Approval Service shall be 99.5%. This value will be reviewed and updated as appropriate.

The Tag Approval Service shall be required to complete the transfer of approval/denial information (via the UPDATE message to the Tag Authority Service) within certain maximum time limits as established by the Industry. Until such limits are established, the Tag Approval shall be required to complete the delivery or exhaust the necessary attempts to deliver these messages within two minutes.

1.5.3.8 Registry Usage

The Tag Approval Service shall be responsible for maintaining an updated list of all registered PSEs, Transmission Providers (TPs), Control Areas (CAs), and any other such entities whose identities must be uniquely specified in connection with the arrangement of an Interchange Transaction. The Master Registry of all such entities shall be maintained and available for downloading from NERC. The Tag Approval Service shall supply a procedure to allow updates from the Master Registry on demand or on a prescheduled interval.

Each Control Area (CA) and Transmission Provider (TP) shall provide the necessary information to identify in the Master Registry who serves as their Tag Approval Service when that particular CA or TP is deemed to have approval rights for a particular transaction tag.

1.5.3.9 Tag Data Entry and Viewing

The Tag Approval Service is the main interface through which entities with approval rights to a tag alert the tag author and each other of their decisions to approve or deny a tag as being a valid representation of a scheduled transaction. To this end, The Tag Approval Service shall provide a mechanism for a user to view Interchange Transaction tags and directly modify the entity state(s) over which they have control, as well as perform all other functional requirements described herein. The exact nature of this user interface is beyond the scope of this document, with the exception that the user shall have the facilities to view all transaction related information (as described in the Data Model) necessary to represent a complete, valid Interchange Transaction tag.

1.5.3.10 Data Validation

1.5.3.11 Function Implementation

1.5.3.11.1 *Tag State Management*

The Tag Approval Service has no specific responsibilities with regard to managing tag state. However, the Tag Approval Service should store any changes in tag state made by the Tag Authority Service when they are either 1.) Communicated through Tag Notification or 2.) Returned in response to a Tag Approval Service issued message.

1.5.3.11.2 *Tag Submission*

The Tag Approval Service has no specific responsibilities with regard to tag submission.

1.5.3.11.3 *Tag Distribution*

Tag Distribution defines the manner in which the Tag Approval Service accepts and processes incoming tag evaluation requests. The Tag Approval Service listener shall accept inbound connections and process them as follows:

First, the Tag Approval Service verifies the message meets all appropriate protocol rules, as defined in the Protocol Description. At this point, the Tag Approval Service should identify the type of request being sent. For the purpose of this discussion, the request being sent shall be an ASSESS message.

Next, the Tag Approval Service processes the message. If the Tag Approval Service finds no additional protocol errors or validation rule violations during the processing of the message, the Tag Approval Service shall respond to the issuer of the request that the Submission attempt was a SUCCESS. At this time, the Tag ID, Tag Key, and Tag Data should be stored and the Tag Evaluation process started.

If the Tag Authority Service did find errors or violations, the Tag Approval Service shall respond to the issuer with a FAIL message.

1.5.3.11.4 Tag Evaluation

Tag Evaluation defines the manner in which a Tag Approval Service communicates the decision of its operator (with regard to the accuracy of a previously supplied tag) back to the Tag Authority Service for that particular tag.

Upon a user's request to update the status a tag, the Tag Approval Service must first determine which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the Update request.

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the transfer of the update of status of the tag to that Authority Service using the "UPDATE" message, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with information that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to of each of those approvers.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.11.5 Tag Notification

Tag Notifications pass information about the most current state of a particular tag to the Tag Approval Service. The Tag Approval Service listener shall accept inbound connections and process them as follows:

First, the Tag Approval Service verifies the message meets all appropriate protocol and validation rules, as defined in the Protocol Description and Data Model.

Next, the Tag Approval Service processes the message. If the Tag Approval Service finds no additional protocol errors or validation rule violations, the Tag Approval Service shall respond to the issuer of the notification that the notification attempt was a SUCCESS and use the contents of the message to update information currently stored by the Tag Approval Service.

If the Tag Approval Service did find errors or violations, the Tag Approval Service shall respond to the issuer with a FAIL message.

1.5.3.11.6 Tag Forwarding

The Tag Approval Service has no specific responsibilities with regard to tag forwarding.

1.5.3.11.7 Tag Querying

The Tag Approval Service issues queries for a tags status by first determining which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the query request

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the query for information about the tag to that Authority Service using either the “STATUS” or “DSTATUS” messages, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to of each of those approvers. If the original request was a “DSTATUS” request, the Tag Authority Service will additionally return all tag information known by that Tag Authority Service to relate to that tag.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.11.8 Tag Cancellation

The Tag Approval Service has no specific responsibilities with regard to tag cancellation.

1.5.3.11.9 Tag Adjustment

Tag Adjustment defines the manner in which a Load Control Area’s Tag Approval Service communicates the desire of its operator to modify for a period of time the represented megawatt flow in a particular tag back to the Tag Authority Service for that particular tag.

Upon a user’s request to adjust a tag, the Tag Approval Service must first determine which Tag Authority Service is being used to manage the transaction. The Sink (Load) Control Area designated in the Tag ID along with the associated Tag Authority Service identification information registered by that Control Area in the Master Registry shall be used by the Tag Approval Service to determine where to send the Adjust request.

Once the appropriate Tag Authority Service is determined, the Tag Approval Service connects to the Tag Authority Service and initiates the transfer of the Adjustment to the tag to that Authority Service using the “ADJUST” message, as defined in the Protocol Description. After the transfer is complete, the Tag Approval Service will await the disposition of its request.

If the Tag Authority Service responds with a SUCCESS message, the Tag Approval Service is provided with additional information created by the Tag Authority Service that describes the current state of the tag, the entities with approval rights over the transaction represented by the tag, and the current disposition of the tag with regard to each of those approvers, as well as the confirmation of the accepted adjustment.

If the Tag Authority Service responds with a FAIL message, the Tag Approval Service is provided with a list of protocol errors and/or validation rule violations.

1.5.3.11.10 Tag Replacement

The Tag Approval Service has no specific responsibilities with regard to tag replacement. However, it is the responsibility of the Tag Approval Service to appropriately highlight the existence of the tag's REPLACE table so that the operator can evaluate the tag in light of the pending replacement.

1.5.3.11.11 Adjustment Notification

The Tag Approval Service receives ADJUST_LIST messages from the Tag Authority service. The Tag Approval Service shall accept the ADJUST_LIST message and take appropriate actions to alert the Tag Approval entity of impending adjustments to ongoing transactions. At the discretion of the Tag Approval Service, the ADJUST_CHECK message may be directed to the IDC to verify that the ADJUST_LIST message was indeed authored by the IDC.

It should be noted that the ADJUST_LIST message is designed in a manner such that a Load Control Area's Tag Approval Service may use its information as a template to prepare ADJUST messages.

1.5.3.12 Failure Recovery

See Policy 3, Appendix 3A3 "Failure Procedures."

1.5.3.13 Availability and Performance Requirements

Until otherwise established by the Industry, the target availability requirement for a provider's Tag Approval Service shall be 99.5%. This value will be reviewed and updated as appropriate.

The Tag Approval Service shall be required to complete the transfer of approval/denial information (via the UPDATE message to the Tag Authority Service) within certain maximum time limits as established by the Industry. Until such limits are established, the Tag Approval shall be required to complete the delivery or exhaust the necessary attempts to deliver these messages within two minutes.

1.5.4 Interchange Distribution Calculator

1.5.4.1 Introduction

The Interchange Distribution Calculator (IDC) is used by Security Coordinators (SCs) to identify transactions for curtailment, reallocation, and reloading. Functions of the IDC with regard to security analysis and operations are determined by the NERC IDC Working Group. The information below describes the role of the IDC with regard to the E-Tag system.

1.5.4.2 Registry Usage

The Interchange Distribution Calculator shall be responsible for maintaining an updated list of all registered PSEs, Transmission Providers (TPs), Control Areas (CAs), and any other such entities whose identities must be uniquely specified in connection with the arrangement of an Interchange Transaction. The Master Registry of all such entities shall be maintained and available for downloading from NERC. The Interchange Distribution Calculator shall supply a procedure to allow updates from the Master Registry on demand or on a prescheduled interval.

1.5.4.3 Tag Data Viewing

User Interface rules for the IDC are defined by the NERC IDC Working Group.

1.5.4.4 Data Validation

The Interchange Distribution Calculator shall ensure, upon receipt and prior to transmission of tag data, that all data elements in a tag are legitimate and that no validation rules (as defined in the data model) have been broken.

1.5.4.5 Function Implementation

1.5.4.5.1 Tag State Management

The Interchange Distribution Calculator has no specific responsibilities with regard to managing tag state. However, the Interchange Distribution Calculator should store any changes in tag state made by the Tag Authority Service when they are communicated through Tag Forwarding. Additionally, the Interchange Distribution Calculator is used by Security Coordinators to submit (via the ADJUST_LIST message) requested schedule changes to the GCA, LCA, and Authoring PSE of a transaction.

1.5.4.5.2 Tag Submission

The Interchange Distribution Calculator has no specific responsibilities with regard to tag submission.

1.5.4.5.3 Tag Distribution

The Interchange Distribution Calculator has no specific responsibilities with regard to tag distribution.

1.5.4.5.4 Tag Evaluation

The Interchange Distribution Calculator has no specific responsibilities with regard to tag evaluation. .

1.5.4.5.5 Tag Notification

The Interchange Distribution Calculator has no specific responsibilities with regard to tag notification.

1.5.4.5.6 Tag Forwarding

Other than processing inbound IMPLEMENT messages for storage and internal use, the Interchange Distribution Calculator has no specific responsibilities with regard to tag forwarding.

1.5.4.5.7 Tag Querying

The Interchange Distribution Calculator has no specific responsibilities with regard to tag querying.

1.5.4.5.8 Tag Cancellation

The Interchange Distribution Calculator has no specific responsibilities with regard to tag cancellation.

1.5.4.5.9 Tag Adjustment

The Interchange Distribution Calculator has no specific responsibilities with regard to tag adjustment.

1.5.4.5.10 Tag Replacement

The Interchange Distribution Calculator has no specific responsibilities with regard to tag replacement.

1.5.4.5.11 Adjustment Notification

The Interchange Distribution Calculator shall be responsible for issuing ADJUST_LIST messages to communicate Security Coordinator requests for changes to schedules.

Security Coordinators utilizing IDC recommended schedule changes to manage a constraint shall utilize the functions of the IDC to generate a list of schedule changes for distribution to regional Security

Coordinators affected by the changes. Upon either 1.) The acknowledgment of all regional security coordinators of the requested schedule changes (including both acceptance of and modification to requested schedule changes), or 2.) the elapsing of five (5) minutes since the distribution of said schedule changes to the regional Security Coordinators affected by the changes, the IDC will prepare and disseminate ADJUST_LIST messages to each of the Generation Control Areas, Load Control Areas, and Authoring Purchasing/Selling entities affected by the requested schedule changes.

Each Generation Control Area, Load Control Area, and Authoring Purchasing/Selling Entity associated with a request for relief shall be sent one ADJUST_LIST message that contains a list of all tags whose schedules have been requested to be changed. No single entity shall receive more than one list for any specific SC initiated request for relief. No entity shall be provided with information regarding any tag to which they do not serve a role as GCA, LCA, or Authoring PSE. For example, given the following tags:

AAAA_MMMM01_BBBB

AAAA_MMMM02_CCCC

BBBB_MMMM01_CCCC

If these three tags were scheduled to be modified, a total of five (5) adjust list messages would be required (one to AAAA listing tags 1 and 2, one to BBBB listing tags 1 and 3, one to CCCC listing tags 2 and 3, one to MMMM01 listing tags 1 and 3, and one to MMMM02 listing tag 2).

For each recipient of the ADJUST_LIST message, the IDC shall generate a unique “Tag Key” similar to those used by the Tag Authority. This key shall utilize the format described in Appendix A. The prefix identifying the key issuer shall be the token “IDC” (i.e., “IDCxxxxxxxx”). This key shall be stored and used to later authenticate the ADJUST_LIST message.

When generating ADJUST_LIST messages, the IDC shall populate each data row of the ADJUST_LIST message as follows:

Tag ID	<Tag ID of the transaction associated with the schedule being requested to change>
Start DateTime	<Time at which the schedule change is being requested to be implemented. Typically, this will be the top of the hour for which relief is being requested. However, the receiving entity should recognize that this time is only a guideline and an approximation for the IDC, and does not represent explicit schedule times accounting for ramping or other operational issues>
Stop DateTime	<Time at which the schedule was originally planned to be completed; the Composite Stop Date Time of the tag>
MW Cap	<The level prescribed by the IDC, or the level assigned by the Regional SC (if different from the IDC prescription). If the Regional SC did not acknowledge the relief request, must be the level originally prescribed by the IDC>
Operator ID	NULL

Reason	<The TLR Incident Reference number associated with the event causing the request. If not associated with a SC issued schedule change request (such as a tag put on hold due to missing the reallocation deadline), should be a descriptive reason (i.e., “Missed Reallocation Deadline”)>
--------	---

After issuing an ADJUST_LIST message, the IDC will be expected to verify the validity of this message by responding appropriately to an ADJUST_CHECK message. Tag Approval and Agent services wishing to verify an ADJUST_LIST message will send an ADJUST_CHECK message to the IDC. The IDC should verify that the key presented in the ADJUST_CHECK message matches one associated with an issued ADJUST_LIST. If a match is found, the IDC should respond with a SUCCESS message. If not, the IDC should respond with a FAIL message.

1.5.4.6 Failure Recovery

Failure Recovery rules for the IDC are defined by the NERC IDC Working Group.

1.5.4.7 Availability and Performance Requirements

Availability and Performance Requirements for the IDC are defined by the NERC IDC Working Group.

Chapter 2 — Protocol Description

2.1 Introduction

The following sections describe the communications protocols for sharing ETAG data. ETAG utilizes various industry standard protocols; these are not defined in detail within this document. Should detailed information be needed, users are referred to appropriate documents.

2.2 Use of the Transmission Control Protocol/Internet Protocol

The TIS services defined in this document utilize the public Internet as their physical communication layer. Therefore, the underlying root protocol for this specification shall be TCP/IP. Additionally, the TIS services defined in this document shall send data via Port 80, the common known port for HTTP, using TCP connections.

When participating entities register for service, they will be required to supply information on the manner in which their implementation will address certain needs. Explicitly, they will need to define:

URL for Tag Authority Service (Control Areas only)

URL for SC/IDC Forwarding (Control Areas only)

URL for Tag Approval Service (Control Areas and Transmission Providers only)

URL for Tag Agent Service (optional for Purchasing Selling Entities)

For the purposes of this document, a URL (Uniform Resource Locator) can be considered a two-part description of a resource. The first part describes the scheme used to communicate and the host the communication is to take place with:

<http://www.nerc.com>

The second part is the URI, or Uniform Resource Identifier. It describes a particular resource on a host:

`/~gads/meetings.html`

This distinction is important in that when implementing this Interface, the first portion of a URL will define the host to connect to, while the URI will define what resource to apply HTTP request to. Therefore, the following URL:

<http://www.nerc.com/~gads/meetings.html>

would be interpreted in the following manner:

<TCP/IP command> connect to “www.nerc.com”

<Application specific command> write the HTTP request to the connection

In the above example, the request would be “GET /~gads/meetings.html HTTP/1.0”

2.2.1 *Establishing Connections*

Establishing connections should be handled in the manner defined by the TCP/IP protocol.

For connections between Tagging Services (Tag Agent, Tag Authority, and Tag Approval):

Should a connection attempt fail or any response other than SUCCESS or FAIL be received, the service initiating the connection request must follow the procedures below prior to assuming the recipient’s service is unavailable and indicating a message failure:

At least three (3) attempts must be made to make the connection, with no less than five (5) seconds between each attempt, with the maximum time between the first and last attempts not to exceed two (2) minutes.

For connections between Tag Authority Services and the IDC:

Should a connection attempt fail or any response other than SUCCESS or FAIL be received, the Tag Authority initiating the connection request must follow the procedures below prior to assuming the IDC is unavailable and indicating a message failure:

At least three (3) attempts must be made to make the connection, with no less than thirty (30) seconds between each attempt, with the maximum time between the first and last attempts not to exceed one (1) hour.

2.2.1.1 *Partial Connection Failures*

Should a connection attempt appear to fail between the Agent, Authority, and/or Approval Services, yet messaging succeed, an invalid set of errors may be encountered by re-sending the same message (i.e., Tag ID Not Unique errors), leading the sender to report incorrect error information. The following strategies should be utilized for overcoming this problem:

SUBMIT: If an Agent submits a Tag which is IDENTICAL in all ways (save the information created and/or modified by the Authority) to one sent previously, the Authority should respond with a success message that returns the current HEADER, STATUS, COMPOSITE, REPLACE (if present), and ADJUST (if present) tables.

ASSESS: If an Approval receives an ASSESS which is IDENTICAL in all ways (save the STATUS and COMPOSITE tables) to one sent previously, the Approval should respond with a success message that returns the HEADER, and apply any updated information contained within the STATUS and COMPOSITE tables.

CANCEL: If an Authority receives a CANCEL message which attempts to set the COMPOSITE table_STOP_DATETIME to a date/time that has already been set (i.e., ABC wants to set the DATETIME to 16:00, but it is already set to 16:00), the Authority should respond with a success message that returns the HEADER, STATUS, COMPOSITE, and ADJUST (if present) tables.

2.3 Use of the Hypertext Transport Protocol

The TIS services defined in this document utilize HTTP 1.0. HTTP 1.1 is not supported. For more information, please see RFC 1945 (T. Berners-Lee, et al).

2.3.1 HTTP Requests

The TIS services defined in this document utilize a single HTTP method: the POST method. This method is used for sending data to a server for processing. The standard format of an HTTP Request Header is as follows:

<HTTP method> <resource URI> <HTTP Version>

In this implementation, all Request Headers will exist as the following:

POST <resource URI> HTTP/1.0

This specifies the POST method is to be used, the path and name of the processing resource, and that using HTTP 1.0 is the protocol and version being used. Additional header fields required are described below:

Content-type: application/x-tmpdata

Declares that the type of data attached to the POST request will be a TMPDATA Message

Content-length: <integer>

Describes in bytes the length of the following attachment. The recipient utilizes this byte length to retrieve the Payload

A Carriage Return/Line Feed terminates each header line. The request is completed by sending a Carriage Return/Line Feed on an empty line marking the end of the HTTP headers, followed by the Entity Data or Payload. The payload for the purposes of this document shall be a Tagging Messaging Protocol message.

2.3.2 HTTP Responses

HTTP Responses are returned to a client with the following syntax:

<HTTP Version> <Status Code> <Explanation>

The status codes below are utilized and understood by the TIS services defined in this document:

200	OK	States that the POST request was accepted and appears to be valid
400	Bad Request	States that the POST request was accepted but appears to point to an invalid URI or does not contain a valid Content-Type

Successful responses will be followed with an entity descriptor, describing the data to follow:

Content-type: application/x-tmpdata

Declares that the type of data attached to the response will be a message utilizing the Tagging messaging Protocol

Content-length: <integer>

Describes in bytes the length of the following attachment. The recipient uses this byte length to retrieve the Payload.

A Carriage Return/Line Feed terminates each response line. The response is completed by sending a Carriage Return/Line Feed on an empty line marking the end of the HTTP response, followed by the Entity Data or Payload. The payload for the purposes of this document shall be a Tagging Messaging Protocol message.

The server terminates the connection when the last of the payload has been transmitted.

For more information on HTTP, please see RFC 1945 (T. Berners-Lee, et al).

2.4 Use of the Tagging Messaging Protocol

2.4.1 *Generic Request Structure*

The Tagging Messaging Protocol (TMP) describes messages defined in the Functional Requirements document. All messages are structured in a similar manner, with optional Request Data and Service Identification Keys used in certain messages.

<Request type> <Target Entity> <Tag ID> <Tag Key> {<Request Data>}

<Tag Data | Special Data>

<Request ending>

Descriptions of the various requests and their associated responses can be found below.

Target Entity specifies to whom the request message is being sent.

Tag ID defines the Tag to be associated with the request, while Tag Key is a code describing either the sender's or the requestor's credentials with regard to the particular tag.

The combination of Tag ID and Tag Key identifies from or to whom (or on whose behalf) the request is being issued.

Request Data is information associated with a particular type of request.

Tag Data is information formatted according to the rules defined in the Data Model.

Special Data refers to information associated with the Tag that is associated with a particular type of request.

Request Ending is a token used to mark the closing of the request.

The end of a line is signified by a Line Terminator, as defined in Appendix A. The final line of the payload must ALWAYS be ended with a Carriage Return/Linefeed Pair.

2.4.2 Generic Response Structure

Responses returned from the server take the format:

<Response type>

<Tag Data | Explanation>

<Response ending>

There are only two possible response types — SUCCESS and FAIL.

A SUCCESS message will return Tag Data back to the requestor as defined with each message.

FAIL messages will respond with one or more explanations describing the failure. Failures may be due to protocol errors and/or validation rule violations. Each explanation must be listed on a separate line. The end of a line is signified by a Line Terminator, as defined in Appendix A.

Explanations are formatted as follows:

<protocol error code | validation rule code> <description>

Examples of valid FAIL messages are:

FAIL

010000Cannot Locate GCA_PSE1234567_LCA

FAIL_END

FAIL

050506Line 4 PRHSW is not a registered acronym

FAIL_END

2.4.3 Requests and Responses

2.4.3.1 The SUBMIT Message

2.4.3.1.1 Usage

Used to submit a Tag from Tag Agent Service to Tag Authority Service. The “Replace Tag ID” and “Replace Tag Key” values are optional parameters designating Tags to be replaced on approval of the submitted tag. Multiple (separated by semicolons) Replace Tag ID and Tag Key pairs (separated by commas) may be included within the left and right brace characters (“{ }”). Tag Data is defined as the following information as defined in the Data Model: HEADER, TAG, REQUESTOR, SOURCE, SINK, PROVIDER, ENERGY, LOSSES, any other tables supplied by the Tag Agent and the END OF DATA MARKER. Empty tables must be omitted. At a minimum, the following tables must be provided: HEADER, TAG, REQUESTOR, PROVIDER, ENERGY, and the END OF DATA MARKER.

SUBMIT <Target Entity> <Tag ID> <Tag Key> {<Replace Tag ID>,<Replace Tag Key>;..}

<Tag Data>

SUBMIT_END

Example:

SUBMIT LCA GCA_PSE123A456_LCA PSE1234567890AB

“GCA_PSE123A456_LCA”, “V1.6”, “NNN”

TAG, { ...

...END

SUBMIT_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted, that the Tag Submission process is complete and the Tag Authority Service is beginning the Tag Distribution Process. Tag Data is defined as the following information as defined in the Data Model: HEADER, STATUS, COMPOSITE, and REPLACE (if appropriate).

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.1.2 **Protocol Errors**

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

030000	Invalid Authority
--------	-------------------

060103	Stale Tag Submission
060104	Table not allowed on Submit

2.4.3.2 The ASSESS Message

2.4.3.2.1 Usage

Used to submit Tag from Tag Authority Service to Tag Approval Service. Tag Data is defined as the following information as defined in the Data Model: HEADER, TAG, REQUESTOR, SOURCE, SINK, PROVIDER, ENERGY, LOSSES, REPLACE, STATUS, COMPOSITE, any other tables supplied by the Tag Agent in the original request, any tables added by the Authority, and the END OF DATA MARKER. Empty tables must be omitted.

ASSESS <Target Entity> <Tag ID> <Tag Key>

<Tag Data>

ASSESS_END

Example:

```
ASSESS GCA GCA_PSE123A456_LCA LCA1234567890AB
```

```
“GCA_PSE123A456_LCA”,”V1.6”,”NNN”
```

```
TAG,{...
```

```
...END
```

```
ASSESS_END
```

Possible return values are either SUCCESS or FAIL:

```
SUCCESS
```

```
<Tag Data>
```

```
SUCCESS_END
```

This response shows the request has been accepted, the Tag Distribution process (with regard to the target entity) is complete, and that the Tag Evaluation process has been started. Tag Data is defined as the following information as defined in the Data Model: HEADER.

```
FAIL
```

```
<Reason(s) for failure>
```

```
FAIL_END
```

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.2.2 **Protocol Errors**

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

040000	Tag ID Not Unique
05xxxx	Data Model Rules Violation
060001	Unknown or Inappropriate Target Entity

2.4.3.3 The INFO Message

2.4.3.3.1 Usage

Used to submit Tag from Tag Authority Service to Tag Agent Service for transactions wholly or partially within WSCC. Tag Data is defined as the following information as defined in the Data Model: HEADER, TAG, REQUESTOR, SOURCE, SINK, PROVIDER, ENERGY, LOSSES, REPLACE, STATUS, COMPOSITE, any other tables supplied by the Tag Agent in the original request, any tables added by the Authority, and the END OF DATA MARKER. Empty tables must be omitted.

INFO <Target Entity> <Tag ID> <Tag Key>

<Tag Data>

INFO_END

Example:

```
INFO PQRPM GCA_PSE123A456_LCA LCA1234567890AB
```

```
“GCA_PSE123A456_LCA”,”V1.6”,”NNN”
```

```
TAG,{...
```

```
...END
```

```
INFO_END
```

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted, the Tag Distribution process (with regard to the target entity) is complete, and that the Tag Evaluation process has been started. Tag Data is defined as the following information as defined in the Data Model: HEADER.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.3.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

040000	Tag ID Not Unique
05xxxx	Data Model Rules Violation
060001	Unknown or Inappropriate Target Entity

2.4.3.4 The UPDATE Message

2.4.3.4.1 Usage

Used to submit approvals from Tag Approval Service to Tag Authority.

UPDATE <Target Entity> <Tag ID> <Tag Key>

“<APPROVED | DENIED | STUDY>“,”<Operator ID>“,”<REASON>“

UPDATE_END

Example:

UPDATE LCA GCA_PSE123A456_LCA LCA1234567890AB

“APPROVED”,”John Doe”,

UPDATE_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted and the current state defined by the Tag Evaluation of the updating entity has been set appropriately. Tag Data is defined as the following information as defined in the Data Model: HEADER, STATUS, and COMPOSITE. Empty tables must be omitted.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.4.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
060001	Unknown or Inappropriate Target Entity
060002	Operator ID Missing
060003	Reason Missing
060401	Invalid State Specified

2.4.3.5 The NOTIFY Message

2.4.3.5.1 Usage

Used to notify all parties Tag Approval Services (and optionally the Tag Agent Service) of tag changes in state or profile. Tag Data is defined as the following information as defined in the Data Model: HEADER, STATUS, and COMPOSITE, and ADJUST (if appropriate). Empty tables must be omitted.

NOTIFY <Target Entity> <Tag ID> <Tag Key>

<Tag Data>

NOTIFY_END

Example:

```
NOTIFY GCA GCA_PSE123A456_LCA LCA1234567890AB
```

```
“GCA_PSE123A456_LCA”,”V1.6”,”NNN”
```

```
STATUS,{...
```

```
NOTIFY_END
```

Possible return values are either SUCCESS or FAIL:

```
SUCCESS
```

```
<Tag Data>
```

```
SUCCESS_END
```

This response shows the request has been accepted, the Tag Notification process has been completed with regard to the Target Entity, and the receiver has in its possession the most recent copy of the tag. Tag Data is defined as the following information as defined in the Data Model: HEADER.

```
FAIL
```

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.5.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
05xxxx	Data Model Rules Violation
060001	Unknown or Inappropriate Target Entity

2.4.3.6 The IMPLEMENT Message

2.4.3.6.1 Usage

Used to submit Tag from Tag Authority to a Sink (Load) CA's designated forwarding location. Tag Data is defined as the following information, as described in the Data Model: HEADER, TAG, REQUESTOR, SOURCE, SINK, PROVIDER, ENERGY, LOSSES, REPLACE, ADJUST, STATUS, COMPOSITE, any tables supplied by the Tag Agent with the original request, and tables added by the Tag Authority, and the END OF DATA MARKER. Empty tables must be omitted.

IMPLEMENT <Target Entity> <Tag ID> <IDC Tag Key>

<Tag Data>

IMPLEMENT_END

Example:

```
IMPLEMENT SC GCA_PSE123A456_LCA LCA1234567890AB LCABA0987654321
```

```
“GCA_PSE123A456_LCA”,”V1.6”,”NNN”
```

```
TAG,{...
```

```
...END
```

```
IMPLEMENT_END
```

Possible return values are either SUCCESS or FAIL:

```
SUCCESS
```

<Tag Data>

SUCCESS_END

This response shows the request has been accepted and the Tag Forwarding process is complete. Tag Data is defined as the following information as described in the Data Model: HEADER.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.6.2 **Protocol Errors**

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

05xxxx	Data Model Rules Violation
060001	Unknown or Inappropriate Target Entity

2.4.3.7 The STATUS Message

2.4.3.7.1 **Usage**

This message is used by an entity to request the status information associated with a Tag from the Tag Authority Service.

STATUS <Target Entity> <Tag ID> <Tag Key>

STATUS_END

Example:

STATUS LCA GCA_PSE123A456_LCA PSE1234567890AB

STATUS_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted and that the Tag Querying process had been completed. Tag Data is defined as the following information as defined in the Data Model: HEADER, ADJUST, COMPOSITE, and STATUS. Empty tables must be omitted.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.7.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
060001	Unknown or Inappropriate Target Entity
060004	Excessive Tag Query

2.4.3.8 The DSTATUS Message

2.4.3.8.1 Usage

This message is used by an entity to request a detailed copy of the entire Tag from the Tag Authority.

DSTATUS <Target Entity> <Tag ID> <Tag Key>

DSTATUS_END

Example:

DSTATUS LCA GCA_PSE123A456_LCA PSE1234567890AB

DSTATUS_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted and that the Tag Querying process had been completed. All tables associated with the requested Tag are returned. Tag Data is defined as all Tag information an Authority has regarding a certain Tag, as described in the Data Model. The Authority must supply: HEADER, TAG, REQUESTOR, SOURCE, SINK, PROVIDER, ENERGY, LOSSES, REPLACE, ADJUST, STATUS, COMPOSITE, the END OF DATA MARKER, and all other tables stored with the tag in question. With the exception of REPLACE, ADJUST, STATUS, COMPOSITE, and any other tables created after the tag was submitted, the Tag Data shall be returned verbatim to the data originally submitted in the Tag Agent's SUBMIT message. Empty tables must be omitted.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.8.2 **Protocol Errors**

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
060001	Unknown or Inappropriate Target Entity
060004	Excessive Tag Query

2.4.3.9 The CANCEL Message

2.4.3.9.1 **Usage**

Used by the Tag Agent to Cancel, Terminate, or Withdraw a transaction through contact with the Tag Authority. A reason MUST be specified for the cancellation. Cancel Date/Time is the point in time at which the Cancellation should take effect.

CANCEL <Target Entity> <Tag ID> <Tag Key>

<Cancel Date/Time>,”<Operator ID>“,”<Reason>“

CANCEL_END

Example:

CANCEL LCA GCA_PSE123A456_LCA PSE1234567890AB

01/01/1999 22:00,”John Doe”,”Withdrawn”

CANCEL_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response indicates that the CANCEL request has been accepted, that the Tag Cancellation Process is complete, and the Tag Notification process (and Tag Forwarding process, if appropriate) is being started.

Tag Data is defined as the following information in the Data Model: HEADER, COMPOSITE, STATUS, and ADJUST (if appropriate). Empty tables must be omitted.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors and/or validation rules violations. The protocol errors and/or validation rules violations are listed. Validation rules are described in the Data Model.

2.4.3.9.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
060001	Unknown or Inappropriate Target Entity
060002	Operator ID Missing
060003	Reason Missing
060901	Cancel Date/Time Is Invalid

2.4.3.10 The ADJUST Message

2.4.3.10.1 Usage

Used to submit Energy Profile adjustments to Tag Authority in the event of a curtailment or should some other change be needed. MW Cap MAY be a Null value.

ADJUST <Target Entity> <Tag ID> <Tag Key>
<Adjust Start Date Time>,<Adjust Stop Date Time>,<MW Cap>,"<Operator ID>","<Reason>"

ADJUST_END

Example:

ADJUST LCA GCA_PSE123A456_LCA LCA1234567890AB

04/01/1999 14:00,04/01/1999 16:00,50,"IDC","TLR TP_990401_1400"

ADJUST_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

<Tag Data>

SUCCESS_END

This response shows the request has been accepted, the Tag Adjustment process has been completed, and the Tag Notification process has begun. Tag Data is defined as the following information as described in the Data Model: HEADER, STATUS, COMPOSITE, ADJUST.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.10.2 **Protocol Errors**

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
020000	Unknown Tag Key
060001	Unknown or Inappropriate Target Entity
060002	Operator ID Missing
060003	Reason Missing
061001	Invalid Adjust Start Date Time Specified
061002	Invalid Adjust Stop Date Time Specified
061003	Invalid MW Cap Specified
061004	Stale Tag Adjustment
061005	Entity does not have rights to ADJUST

2.4.3.11 **The ADJUST_LIST Message**

2.4.3.11.1 **Usage**

Used by the IDC to submit a list of Tag IDs to Tag Approval Services and Tag Agent Services that are to be adjusted as specified in the event of a curtailment or should some other change be needed. MW Cap MAY be a Null value.

When issued by the IDC, the "Authentication Key" field must specify a randomly generated key to be assigned to the recipient of the message. The "Authentication Key" field must comply with the general format for a Tag Key (entity code plus 12 character unique security token). The Entity Code to be used in creating the key shall be the phrase "IDC."

Receipt of the ADJUST_LIST message by the Tag Approval or Tag Agent notification services creates no additional action on the part of the service except to present that information to the LCA, GCA or PSE operator.

The ADJUST_LIST message list itself represents a request from the security coordinator for the LCA to take ADJUST action on one or more specified tags. At the approval of the LCA operator, the LCA should then issue the appropriate ADJUST messages (which would then be automatically followed by the appropriate NOTIFY and IMPLEMENT messages) indicating that the requested action has indeed taken place. The mechanism for completing that process should be vendor specific.

ADJUST_LIST <Target Entity> <Authentication Key>
 "<Tag ID 1>" <Adjust Start Date Time>,<Adjust Stop Date Time>,<MW Cap>,"<Operator ID>","<Reason>"

"<Tag ID 2>," <Adjust Start Date Time>,<Adjust Stop Date Time>,<MW Cap>,"<Operator ID>","<Reason>"

.
 .
 .

"<Tag ID n>," <Adjust Start Date Time>,<Adjust Stop Date Time>,<MW Cap>,"<Operator ID>","<Reason>"

ADJUST_LIST_END

Example:

ADJUST_LIST LCA IDC1234567890AB

"GCA1_PSE123A456_LCA",04/01/1999 14:00,04/01/1999 16:00,50,"","TLR
 TP_990401_1400"

"GCA2_PSE456A987_LCA",04/01/1999 14:00,04/01/1999 16:00,25,"","TLR
 TP_990401_1400"

"GCA3_PSE789A123_LCA",04/01/1999 14:00,04/01/1999 16:00,0,"","TLR
 TP_990401_1400"

ADJUST_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

SUCCESS_END

This response shows the request has been accepted and the data has been presented to the operator for implementation. No additional information is conveyed by the SUCCESS message other than that the message was successfully received. Note that this is a request for adjustment by the IDC and specifically not an adjustment itself.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.11.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

010000	Tag Does Not Exist
060001	Unknown or Inappropriate Target Entity
060003	Reason Missing
061001	Invalid Adjust Start Date Time Specified
061002	Invalid Adjust Stop Date Time Specified
061003	Invalid MW Cap Specified
061004	Stale Tag Adjustment

2.4.3.12 The ADJUST_CHECK Message

2.4.3.12.1 Usage

Used by either Tag Approval or Tag Agent notification services to verify that an ADJUST_LIST message received was in fact generated by the Interchange Distribution Calculator. The "Authentication Key" in the ADJUST_LIST message is dictated to be of the general form for any Tag Key, i.e., 1-6 character entity code issuing the key followed by 12-character unique security token. On receipt of the ADJUST_LIST message, the receiving party may verify the authenticity of the message by issuing an ADJUST_CHECK message to the IDC. If the IDC verifies that the Authentication Key presented in the

ADJUST_CHECK message was issue by the IDC, a SUCCESS response is issued; otherwise a FAIL response is issued. Target Entity shall always be the phrase “IDC.”

ADJUST_CHECK IDC <Authentication Key>

ADJUST_END

Example:

ADJUST_CHECK IDC IDC1234567890AB

ADJUST_CHECK_END

Possible return values are either SUCCESS or FAIL:

SUCCESS

SUCCESS_END

This response shows that the Tag Authority identified by "Target Entity" verifies that the "Authentication Key" presented corresponds to a valid unique identification key assigned by that Authority to an ADJUST_LIST message received from the IDC.

FAIL

<Reason(s) for failure>

FAIL_END

This response indicates the request has NOT been accepted because it has protocol errors. The protocol errors and/or validation rules violations are listed.

2.4.3.12.2 Protocol Errors

Acceptable protocol errors are listed below. For a complete list of protocol errors, please see Protocol Error Descriptions.

020000	Unknown Authentication Key
060001	Unknown or Inappropriate Target Entity

2.4.4 Protocol Error Description

The following table lists the various protocol errors and describes their meaning.

Error Code	Error Name	Error Description
010000	Tag Does Not Exist	The Tag ID referenced cannot be found in the Tag database of the target service.

020000	Unknown Tag Key	The Tag Key in the request was not found in the list of valid keys for the Tag ID referenced.
030000	Invalid Authority	The Authority contacted is not the proper Authority for the included transaction.
040000	Tag ID Not Unique	The issuer already has a tag in its possession that uses the Tag ID presented. Therefore, the ID is invalid.
05xxxx	Data Model Rules Violation	The tag data included in the original request did not pass all validation rules as defined in the Data Model. “xxxx” defines the rule that was violated.
06xyyy	Message Error	A message specific error has occurred. “xx” defines the number of the message, while “yy” defines the rule violated. An “xx” of zero indicates the rule is shared by two or more messages.
060001	Unknown or Inappropriate Target Entity	The issuer does not recognize or is not a provider of service for the entity described by the Target Entity field.
060002	Operator ID Missing	The Operator ID field is required.
060003	Reason Missing	The Reason field is required.
060004	Excessive Tag Query	An entity may, at their option, reject queries for status that occur at a frequency of greater than one per thirty (30) seconds, per specific tag, per specific requesting entity.
060101	Unknown REPLACE Tag ID	The Tag Authority Service does not have a tag in its possession hat uses the Tag ID presented.
060102	Unknown REPLACE Tag Key	The Tag Authority Service has possession of the tag defined in the Replace Tag ID field, but the Tag Key presented is not one of the Tag Keys associated with that Tag ID.
060103	Stale Tag Submission	A tag may not be submitted with a start date/time more than one hour in the past.
060104	Table not allowed on SUBMIT	Certain tables may not be sent to an Authority Service via a SUBMIT message, including STATUS, COMPOSITE, ADJUST, or REPLACE.
060401	Invalid State Specified	A state other than those defined in the message description is being presented.
060901	Cancel Date/Time Is Invalid	The date/time specified is not a valid date/time, as described in the Functional Specification.
060902	Stale Tag Cancel	A CANCEL request may not specify an effective date/time in the past.

061001	Invalid Adjust Start Date Time Specified	The date time specified is not valid, as described in the Functional Specification.
061002	Invalid Adjust Stop Date Time Specified	The date time specified is not valid, as described in the Functional Specification.
061003	Invalid MW Cap Specified	The MW Cap specified is not valid, as described in the Functional Specification.
061004	Stale Tag Adjustment	An adjustment may not adjust hours more than one hour in the past.
061005	Entity does not have rights to ADJUST	An entity wishing to ADJUST must possess and present a Tag Key associated with either a Control Area or a Transmission Provider.

Chapter 3 — Data Model for Tag Information

3.1 Introduction

This section describes the electronic tagging data model and its characteristics. The data model is described and transported in Comma Separated Value (CSV) format amongst the three services described in the functional section.

3.2 Data Encoding Rules

With regard to data encoding, there are four data type conventions used:

Strings	<p>Strings shall be represented by enclosing the intended characters within double quotation marks (ASCII character 34). All printing characters of the ASCII set maybe used to assemble a string (save the double quotation mark itself). However, Data Model limitations will often prove more restrictive. As defined in this document, a string must contain at least one character; otherwise, the value should be represented as NULL.</p> <p>Examples:</p> <p>“Test”</p> <p>“This is a test!”</p> <p>“This is test #20,”</p>
Numeric Values	<p>Numeric values shall be represented by simply listing the intended value.</p> <p>Examples:</p> <p>24</p> <p>0</p> <p>32767</p>

Date and/or Time Values	<p>Date and Time values shall be represented by simply listing the intended value. Note that when combining dates and times, they must be separated by a single space.</p> <p>Examples:</p> <p>01/01/1999</p> <p>14:00</p> <p>18:22:34</p> <p>02/07/2000 06:00</p> <p>09/12/2001 17:02:28</p>
NULLs	<p>NULLs shall be represented by an absence of data. Double Quotes with no data (i.e., "") is not a valid representation of NULL.</p>

Values are combined to form records through the use of commas. The end of a record is signified by a Line Terminator, as defined in Appendix A. For example, the following record represents a string, a number, and a date/time:

“This is a test!”,32767,09/12/2001 17:02:28<Line Terminator>

The first record of a tag has a fixed and rigid set of attributes (the header) and the last record has one fixed attribute (the end of data marker). Single values are logically clustered together to form additional records, which are in turn clustered together to form tables.

Tables are identified through the use of a table name, followed by a comma, an open brace (ASCII character 123), and a Line Terminator:

TABLE_NAME,{<Line Terminator>

Records that are associated with the particular table name are then listed, separated by Line Terminators:

“This is a test!”,32767,09/12/2001 17:02:28<Line Terminator>

“This is also a test!”,65535,08/11/2000 14:22:54<Line Terminator>

The table is closed by a close brace (ASCII character 125), a comma, and a record count:

},2<Line Terminator>

Each row in a table is a “record” with attributes in a fixed sequence and number. No other attributes may be present and all must be present. Each field in a record is separated by a comma (ASCII Character 44).

Records comprised solely of NULLs are invalid.

3.3 Tag Data Description

3.3.1 Header Format

The first header record is fixed in content and sequence with three attributes.

Attribute	Description
TAG_ID	Derived through a concatenation of attributes found in the tag utilizing the underscore character as the component separator
VERSION	File stream version control. Changed when parsing algorithm must be altered due to data model changes.
CONTROL	Format parsing option control string. Used to identify certain global properties of a tag.

Only one header is allowed in any tag.

3.3.1.1 Usage

The Header is used to identify the tag itself, as well as supply basic information about the tag.

“AAAA_PPPPPP1234567_BBBB”, “V1.6”, “NNN”

The first field is the Tag ID. As a string, it must be enclosed in double quotes.

TAG ID FORMULA is:

Generating CA_CODE + “_” + PSE_CODE + TAG_CODE + “_” + Load CA_CODE

The Generating CA code is obtained from the first record in the Provider table.

The PSE Code and TAG_CODE are obtained from the Requestor table.

The Load CA Code is obtained from the last record in the Provider table.

The second field is the Version Control String. As a string, it must be enclosed in double quotes. For this data model, the value shall always be “V1.6”

The third field is the Format Parsing Option Control String. As a string, it must be enclosed in double quotes. The Format Parsing Option Control String is used to identify certain properties of the tag as a whole. Current definitions are below.

Position	Code	Description
1	N/Y	“Test TAG” Flag (“N” is a real TAG, “Y” is only a test). All components of the TIS system will react to a tag in test as if it is real. The purpose of the flag is to have backend systems ignore the contents.
2	N/Y	“MRD” Flag (“N” is a active tag, “Y” is an inactive MRD tag). Only “N” is now valid.
3	N/Y	Only “N” is now valid.

3.3.1.2 Validation Rules

Header Rules (050100-050199)	
Rule	Rule Description
0501xx	Generic header rule identifier, where xx represents any 2-digit number
050101	There must be at least three (3) attributes in the header line
050102	There must be no more than three (3) attributes in the header line
050103	The Tag ID attribute must match the Tag Id in the message header
050104	The Version attribute must be literally “V1.6”
050105	The Format Parsing Option Control String must be properly formatted
050106	The test flag must be either “Y” or “N”
050107	The MRD flag must be “N”
050108	The unused third flag must be “N”
050199	Undocumented HEADER error

3.3.2 Table Formats

This section describes the description of the current TABLES with the required number of attributes in their required sequence for an interchange transaction tag.

3.3.2.1 TAG Table

Table	Attribute ID	Description
TAG	STATE	Unused–Required to be NULL
	START_DATE	First Date when the Energy Profile begins
	STOP_DATE	Last Date when the Energy Profile ends
	TIME_ZONE	Required to be “CS”
	REMARKS	Contains remarks or comments of the Tag Author. If no remarks supplied, required to be NULL.
	REASON	Unused–Required to be NULL
	DAY_REPEAT	Day Profile Repeat String (Sunday through Saturday Mapping)

The TAG table must contain only ONE row.

3.3.2.1.1 Usage

The TAG Table is also used to communicate information that describes the tag as a whole.

The first field must be NULL. This field was defined in a previous model and is not currently used.

The second field represents the beginning of the period for which the Day Repeat and Energy Profile values should be used to determine energy flow. This field must always be encoded as a date value. Specific formatting is defined in the Attribute Dictionary.

The third field represents the end of the period for which the Day Repeat and Energy Profile values should be used to determine energy flow. This field must always be encoded as a date value. Specific formatting is defined in the Attribute Dictionary.

The fourth field represents the effective time zone in which date/time values in the tag are represented. As a string, it must be enclosed in double quotes. For this data model, the value shall always be “CS.”

The fifth field represents comments made by the Tag Author at the time of tag creation. As a string, it must be enclosed in double quotes.

The sixth field must be NULL. This field was defined in a previous model and is not currently used.

The seventh field represents, in Sunday to Saturday order, the days of the week and their relation to the energy profile. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary. For day that is represented, there shall be one of two possible values:

Y	The energy profile is expected to be executed starting on the day represented.
N	The energy profile is NOT expected to be executed starting on the day represented.

3.3.2.1.2 **Validation Rules**

TAG Rules (050200-050299)	
Rule	Rule Description
0502xx	Generic TAG rule identifier, where xx represents any 2-digit number.
050201	Table is required, but is not included.
050202	There must be at least one (1) record in the TAG table.
050203	There must be no more than one (1) record in the TAG table.
050204	There must be at least seven (7) attributes in each TAG table record.
050205	There must be no more than seven (7) attributes in each TAG table record.
050206	The START_DATE attribute must be specified in “MM/DD/YYYY” format and must represent a valid date (e.g. 02/29/1997 is not a valid START_DATE value).
050207	The Stop_Date attribute must be specified in “MM/DD/YYYY” format, and must represent a valid date (e.g. 02/29/1997 is not a valid Stop_Date value).
050208	The Stop_Date must be no earlier than the Start_Date; that is, either the two dates must be the same, or the Stop_Date must be later than the Start_Date.
050209	The Time_Zone attribute is invalid, must be literally “CS”.
050210	The Day_Repeat attribute has an invalid value.
050211	A Tag may be considered repeating only if the difference between the derived Start Date Time and derived Stop Date Time is greater than 24 hours.
050212	The REMARKS attribute value must follow the formatting rules defined in the Attribute Dictionary.
050213	The REASON attribute value must be NULL.
050299	Undocumented TAG table error.

3.3.2.2 COMPOSITE Table

Table	Attribute ID	Description
COMPOSITE	COMPOSITE_STATE	Contains the current composite state as described in the Functional Description.
	COMPOSITE_DATETIME	Date and Time (in CS) when the above state attribute was set.
	START_DATETIME	Effective Date and Time (in CS) when this transaction represented begins. Derived from the Start_Date and Day_Repeat properties of the Tag table and the first Start_Time in the Energy table.
	STOP_DATETIME	Effective Date and Time (in CS) when this transaction represented stops. Derived from the Stop_Date and Day_Repeat properties of the Tag table and the last Stop_Time in the Energy table.
	ENTITY_TYPE	Must set to one of “CA”, “TP”, “PSE”, or “SC” corresponding to entity that resulted in change of COMPOSITE_STATE to current value.
	ENTITY_CODE	Must be set to registered code corresponding to entity that resulted in change of COMPOSITE_STATE to current value.
	OPERATOR_ID	When known, must be set to user-supplied identification of responsible party corresponding to entity that resulted in change of COMPOSITE_STATE to current value. If not known, required to be NULL.
	REASON	When known, must be set to user supplied “reason” corresponding to entity that resulted in change of COMPOSITE_STATE to current value. If not known, required to be NULL.

The Tag Authority Service creates the COMPOSITE table as part of tag submission/tag distribution process. The Composite table must contain only ONE row.

3.3.2.2.1 Usage

The COMPOSITE table is used to describe information regarding the overall state of the tag.

The first field contains the active state of the tag. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The second field contains the point in time at which the above state was set. This field must always be encoded as a date/time value. Specific formatting is defined in the Attribute Dictionary.

The third and fourth fields represent the points in time at which energy should first begin to flow and at which energy should last stop flowing (excluding ramp times). These fields must always be encoded as date/time values. Specific formatting is defined in the Attribute Dictionary. The values of these fields are derived in the following manner:

Tags in which the TAG Day_Repeat string is “NNNNNNN”

These tags are considered to represent transactions with a “continuous” profile, which means that the Energy table should be assumed to represent one continuous time period. Continuous profiles are usually limited to transaction spanning 24 hours or less, but may represent longer time periods.

To determine the Start_DateTime, combine the TAG Start_Date and the first Start_Time in the Energy table. To determine the Stop_DateTime, the profile must be expanded completely using the following rules:

- Process the Energy Table from top to bottom, one row at a time.
- If the Stop_Time for a row is less than or equal to the Start_Time for that row, the date should be incremented by one day. Otherwise, the date should be left the same.
- If the Start_Time for a row is less than the Stop_Time for the previous row, the date should be incremented by one day. Otherwise, the date should be left the same.

After the last Stop_Time is processed, the resultant date shall be compared to the TAG Stop_Date. If less than or equal to the TAG Stop_Date, then the TAG Stop_Date shall be combined with the last Stop_Time in the Energy table to create the Stop_DateTime. If greater than the TAG Stop_Date, then the tag has been entered incorrectly and an error should be returned.

Tags in which the TAG Day_Repeat string is NOT “NNNNNNN”

These tags are considered to represent transactions with a “repeating” profile, which means the Energy Table should be assumed to execute in its entirety two or more times over a time period. Repeating profiles always represent a total schedule duration of greater than 24 hours (however, the ENERGY table will represent at most a single 24-hour period that repeats).

Begin with the set of all days greater than or equal to the TAG Start_Date and less than or equal to the TAG Stop_Date. From this set, select the first date that falls on a day of the week specified as a “Y” in the TAG Day_Repeat string. This date shall be combined with the first Start_Time in the Energy table to create the Start_DateTime.

From the same set, select the last date that falls in the week specified as a Y in the TAG Day_Repeat string. If it is possible to execute the profile described in the Energy table and have it complete prior to 00:00 the day after the date listed in the TAG StopDate field, then this date should be considered the “effective stop date.” If not, the first day earlier that both satisfies this criterion and represents a day of week that has been marked as a “Y” should be used as the “effective stop date.” In either case, this “effective stop date” shall be combined with the last Stop_Time in the Energy table to create the Stop_DateTime.

The fifth field represents the type of entity that set the most recent entity state. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The sixth field represents the registered NERC acronym of the entity that set the most recent entity state. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The seventh field represents the representative of the above entity that took action to set the most recent entity state. As a string, it must be enclosed in double quotes.

The eighth field represents the reason the above representative took action to set the most recent entity state. As a string, it must be enclosed in double quotes.

3.3.2.2.2 Validation Rules

COMPOSITE Rules (050300-050399)	
Rule	Rule Description
0503xx	Generic COMPOSITE rule identifier, where xx represents any 2-digit number
050301	COMPOSITE table is required, but is not included.
050302	There must be at least one (1) record in the COMPOSITE table.
050303	There must be no more than one (1) record in the COMPOSITE table.
050304	There must be at least eight (8) attributes in each COMPOSITE table record.
050305	There must be no more than eight (8) attributes in each COMPOSITE table record.
050306	The Composite_State attribute must have one of the values listed in the functional specification.
050307	The Composite_DateTime attribute must be a valid time, formatted as specified in the Attribute Dictionary.
050308	The Start_DateTime attribute must be a valid time, formatted as specified in the Attribute Dictionary.
050309	The Stop_DateTime attribute must be a valid time, formatted as specified in the Attribute Dictionary.
050310	The Stop_DateTime value must be no earlier than the Start_DateTime value.
050311	If a reason is required for the Composite_State value being shown in this record, then the Reason attribute must not be missing, and must not be NULL.
050312	The Reason attribute value must follow the formatting rules defined in the Attribute Dictionary.
050313	The SUBMIT message may not send a COMPOSITE table to the Authority.
050399	Undocumented COMPOSITE table error.

3.3.2.3 STATUS Table

Table	Attribute ID	Description
STATUS	ENTITY_TYPE	Provider Type (“CA”, “TP”, “PSE”, “SC”)
	ENTITY_CODE	Registered Provider Code
	ENTITY_STATE	This will contain the “current” approval state for the entity listed as described in the Functional Description. May be NULL.
	STATE_DATETIME	Date & Time (in CS) of when the entity listed set the above STATE attribute.
	SUBMIT_DATETIME	Date & Time (in CS) of successful transfer of ASSESS message from the Tag Authority to Approval for listed entity, INFO message from the Tag Authority to Agent for listed entity, or of receipt of SUBMIT message from Agent to Authority for listed entity. If message has not yet been transferred, required to be NULL.
	OPERATOR_ID	If known, must be a reference to representative of the providing entity that set current Entity State.
	REASON	When specifying a STATE that requires a REASON, must be non-NULL. If state has not yet been set by providing entity, must be NULL. If state does not require a reason, but does not prohibit supply of one, may be NULL or non-NULL.
	DISTRIBUTE_METHOD	Defines method of Tag Distribution to be used with this entity. If not supplied, must be NULL.
	NOTIFY_METHOD	Defines method of Tag Notification to be used with this entity. If not supplied, must be NULL.

The Tag Authority Service creates the STATUS table as part of tag submission processing based on the contents of the PROVIDER table. The STATUS table may contain multiple rows.

3.3.2.3.1 Usage

The STATUS table is used to record information describing the entities with rights to a transaction and the current state associated with each of those entities. Single records define each entity.

The first field represents the type of entity that has rights over the transaction. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The second field represents the registered NERC acronym of the entity that has rights over the transaction. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The third field contains the active state of the tag with regard to the specific entity described above. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The fourth field contains the point in time at which the above state was set. This field must always be encoded as a date/time value. Specific formatting is defined in the Attribute Dictionary.

The fifth field contains the point in time at which the tag was originally communicated to or from the entity. This field must always be encoded as a date/time value. Specific formatting is defined in the Attribute Dictionary.

The sixth field represents the representative of the above entity that took action to set the most recent entity state. As a string, it must be enclosed in double quotes.

The seventh field represents the reason the above representative took action to set the most recent entity state. As a string, it must be enclosed in double quotes.

The eighth field represents the method to be used to distribute the tag to the entity defined in the second field. Valid methods are ASSESS and INFO. If the this field is NULL, no distribution should be used.

The ninth field represents the method to be used to Notify the entity defined in the second field of any changes to the tag's state or composition. Currently, the only valid method is NOTIFY. If this field is NULL, no notification should be used.

3.3.2.3.2 Validation Rules

STATUS Rules (050400-050499)	
Rule	Rule Description
0504xx	Generic STATUS rule identifier, where xx represents any 2-digit number.
050401	STATUS table is required, but is not included.
050402	The requirements specification define which entities must have records in the STATUS table. If the table ends before all entities have been specified, but every record that is in the table matches the entity that would be expected in that position, then return this error code.
050403	The requirements specification specifies which entities must have records in the STATUS table. If any record in the table has an entity code, which is not the code that is required by the requirements specification at that position, then return this error code.
050404	There must be at least nine (9) attributes in each STATUS table record.
050405	There must be no more than nine (9) attributes in each STATUS table record.
050406	The ENTITY_TYPE attribute must have one of the listed values.
050407	The ENTITY_CODE attribute must have a registered value. Error message returned should include version number of NERC Registry consulted in making error decision.
050408	The ENTITY_STATE attribute must have one of the values listed in the functional specification.

STATUS Rules (050400-050499)	
Rule	Rule Description
050409	The ENTITY_DATETIME attribute must be a valid time, formatted as specified in the Attribute Dictionary.
050410	The SUBMIT_DATETIME attribute must be a valid time, formatted as specified in the Attribute Dictionary.
050411	The OPERATOR_ID attribute value must follow the formatting rules defined in the Attribute Dictionary.
050412	If a reason is required for the ENTITY_STATE value being shown in this record, then the Reason attribute must not be missing, and must not be NULL.
050413	The Reason attribute value must follow the formatting rules defined in the Attribute Dictionary.
050414	The Distribute_Method attribute value must follow the formatting rules defined in the Attribute Dictionary.
050415	The Notify_Method attribute value must follow the formatting rules defined in the Attribute Dictionary.
050499	Undocumented STATUS table error.

3.3.2.4 REQUESTOR Table

Table	Attribute ID	Description
REQUESTOR	PSE_CODE	TSIN Registered Code of requesting PSE.
	TAG_CODE	Must match to the seven (7) character tag code field used to create first attribute of the Tag's HEADER (TAG ID).
	DEAL_REF	For use as a contract reference or common terminology used across multiple transactions when TAG_CODE must be changed. If not supplied, must be NULL.
	OPERATOR_ID	Must be a reference to representative of the requesting entity that authored the tag.
	PHONE	Supplied to override registered default. If not supplied, must be NULL.
	PHONE24	Supplied to override registered default. If not supplied, must be NULL.
	FAX	Supplied to override registered default. If not supplied, must be NULL.

The REQUESTOR table may only contain ONE row.

3.3.2.4.1 Usage

The Requestor table is used to describe information about the entity on whose behalf the tag is being authored.

The first field represents the registered NERC acronym of the entity on whose behalf the tag is being authored. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The second field represents a series of characters that are used to identify the transaction as distinct from other transaction using the same Generating CA, PSE, and Load CA. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The third field represents a series of characters that are used to identify the transaction within any system operated by the requestor. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The fourth field represents the representative of the above entity that authored the tag. As a string, it must be enclosed in double quotes.

The fifth field represents the telephone number of the above representative. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The sixth field represents the telephone number of the above entity at which a representative of that entity may be reached 24 hours a day, 7 days a week, 365 days a year. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The seventh field represents the fax number of the above entity. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

3.3.2.4.2 Validation Rules

REQUESTOR Rules (050500-050599)	
Rule	Rule Description
0505xx	Generic REQUESTOR rule identifier, where xx represents any 2-digit number.
050501	REQUESTOR table is required, but is not included.
050502	If the REQUESTOR table is required, then there must be at least one (1) record in the table.
050503	There must be no more than one (1) record in the REQUESTOR table.
050504	There must be at least seven (7) attributes in the REQUESTOR record.
050505	There must be no more than seven (7) attributes in the REQUESTOR record.
050506	The PSE_CODE value must be a registered PSE acronym. Error message returned should include version number of NERC Registry consulted in making error decision.
050507	The TAG_CODE value must follow the formatting rules defined in the Attribute Dictionary.
050508	The OPERATOR_ID value must follow the formatting rules defined in the Attribute Dictionary.
050509	The Phone value must follow the formatting rules defined in the Attribute Dictionary.
050510	The PHONE24 value must follow the formatting rules defined in the Attribute Dictionary.
050511	The Fax value must follow the formatting rules defined in the Attribute Dictionary.
050599	Undocumented REQUESTOR table error.

3.3.2.5 SOURCE Table

Table	Attribute ID	Description
SOURCE	SOURCE_CODE	Must represent service point of the ultimate source (generator). This would typically be an IPP, NUG, etc. Could also be a company identification or generation zone when used with a large Control Area.
	PHONE	Supplied to override registered default. If not supplied, must be NULL.
	FAX	Supplied to override registered default. If not supplied, must be NULL.

The SOURCE table may contain only ONE row.

3.3.2.5.1 Usage

The SOURCE table is used to describe the entity that will be generating the energy described in the tag.

The first field represents the ultimate source of the energy or generator. As a string, it must be enclosed in double quotes.

The second field represents the telephone number of the above entity. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The third field represents the fax number of the above entity. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

3.3.2.5.2 Validation Rules

SOURCE Rules (050600-050699)	
Rule	Rule Description
0506xx	Generic SOURCE rule identifier, where xx represents any 2-digit number
050601	There must be no more than one (1) record in the SOURCE table.
050602	There must be no more than three (3) attributes in the SOURCE record.
050603	The SOURCE_CODE value must be a registered SOURCE acronym. Error message returned should include version number of NERC Registry consulted in making error decision. <i>Note: This rule cannot be evaluated until SOURCE values are registered.</i>
050604	The Phone value must follow the formatting rules defined in the Attribute Dictionary.
050605	The Fax value must follow the formatting rules defined in the Attribute Dictionary.
050606	There must be no less than three (3) attributes in the SOURCE record.

SOURCE Rules (050600-050699)	
Rule	Rule Description
050607	There must be no less than one (1) record in the SOURCE table.
050699	Undocumented SOURCE table error.

3.3.2.6 SINK Table

Table	Attribute ID	Description
SINK	SINK_CODE	Must represent service point of the ultimate sink load. This would typically be TDU, etc., or become large customers as retail access unfolds. Could also be a company identification or load zone when used with a large Control Area.
	PHONE	Supplied to override registered default. If not supplied, must be NULL.
	FAX	Supplied to override registered default. If not supplied, must be NULL.

The SINK table may contain only ONE row.

3.3.2.6.1 Usage

The SINK table is used to describe the entity that will be utilizing the energy described in the tag.

The first field represents the ultimate sink of the energy or load. As a string, it must be enclosed in double quotes.

The second field represents the telephone number of the above entity. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The third field represents the fax number of the above entity. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

3.3.2.6.2 Validation Rules

SINK Rules (050700-050799)	
Rule	Rule Description
0507xx	Generic SINK rule identifier, where xx represents any 2-digit number.
050701	There must be no more than one (1) record in the SINK table.
050702	There must be no more than three (3) attributes in the SINK record.

SINK Rules (050700-050799)	
Rule	Rule Description
050703	The SINK_CODE value must be a registered SINK acronym. Error message returned should include version number of NERC Registry consulted in making error decision. <i>Note: This rule cannot be evaluated until SINK values are registered.</i>
050704	The Phone value must follow the formatting rules defined in the Attribute Dictionary.
050705	The Fax value must follow the formatting rules defined in the Attribute Dictionary.
050706	There must be no less than three (3) attributes in the SINK record.
050707	There must be no less than one (1) record in the SINK table.
050299	Undocumented SINK table error.

3.3.2.7 PROVIDER Table

Table	Attribute ID	Description
PROVIDER	CA_CODE	If present, must be TSIN Registered Control Area Code of provider. If representing first record (Generating Control Area) or last record (Load Control Area), must be present. Otherwise, optional (depending on structure of the path). If not supplied, must be NULL.
	TP_CODE	If present, must be TSIN registered TP Code of provider. If not supplied, must be NULL.
	PSE_CODE	If present, must be TSIN registered PSE Code of provider. If not supplied, must be NULL.
	TP_PRODUCT_CODE	Designates, by 1 st character, the NERC Curtailment priority (currently the values 0–7) consistent with the transmission product specified by TP_ASSIGNMENT_REF. Must be present if TP_ASSIGNMENT_REF is present; otherwise, must be NULL.
	TP_PATH	Designates the transmission path to be used on TPs system. If associated TP_CODE appears twice in the PROVIDER table, this field is required to be non-NULL. If TP_CODE is NULL, must be NULL.
	TP_ASSIGNMENT_REF	References transmission rights on the system associated with TP_CODE of the entity indicated by PSE_CODE. If TP_CODE is non-NULL, this field is required to be non-NULL. May be specified on records with NULL TP_CODE only when associating multiple OASIS reservations with a single path segment. Otherwise, must be NULL.
	TP_PRODUCT_LEVEL	When specified, indicates the maximum MW amount of capacity reserved under the associated TP_ASSIGNMENT_REF that may be used to satisfy the requirements for a given transmission path segment. May only be specified on records with non-NULL TP_ASSIGNMENT_REF. Otherwise, must be NULL.
	MISC_INFO	Used to supply to a provider information that does not relate to a NERC curtailment priority indexed product. If not supplied, must be NULL.
	MISC_REF	Used to supply to a provider a reference that does not relate to a NERC curtailment priority indexed product. If not supplied, must be NULL.

The PROVIDER table may contain multiple rows.

3.3.2.7.1 Usage

The PROVIDER table is used to illustrate the use of various provider services in order supply energy to a customer. Multiple records are used to define the relationships between each of the providers of service.

The first field is used to represent the registered NERC acronym of a Control Area scheduling interchange for the transaction represented by the tag. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary. The first Control Area in this table is ALWAYS considered the Generating Control Area, and the last is ALWAYS considered the Load Control Area.

The second field is used to represent the registered NERC acronym of a Transmission Provider providing transmission rights to allow the interchange to occur. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The third field is used to represent the registered NERC acronym of a Purchasing/Selling Entity 1.) Transferring ownership or title of a energy asset, 2.) Allowing the use of transmission rights by the Tag Author, or 3.) A combination of both these functions. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The fourth field is used to represent a NERC Curtailment Priority indexed transmission product that had been reserved for use through an associated Transmission Provider defined above. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The fifth field is used to represent the reserved path associated with the above Transmission Provider and product. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The sixth field is used to represent the transmission used (e.g. OASIS assignment reference) by the tag author or associated PSE and associated with the above Transmission Provider, product, and path. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

The seventh field is used to represent the maximum amount (in whole megawatts) of the available reserved transmission capacity for a particular reservation to be used to fulfill transmission requirements for the represented transaction. This field must always be encoded as a numeric value. Specific formatting is defined in the Attribute Dictionary. The eighth field is used to provide information not related to transmission products to a provider. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary. Some examples of proper use would be to indicate Firm Generation, Non-Firm Generation, Load, or other information.

The ninth field is used to provide a reference number not related to transmission products to a provider. As a string, it must be enclosed in double quotes. Specific formatting is defined in the Attribute Dictionary.

If multiple transmission reservations are desired to support a SINGLE transmission segment, convention dictates that the transmission “stack” should be represented in the following manner:

First row of the Stack:

CA Code	May be any appropriate control area scheduling interchange. If not provided, must be NULL.
TP Code	Must be the Transmission Provider whose services are being combined to form the stack.
PSE Code	May be any appropriate Purchasing/Selling Entity. If not provided, must be NULL.
TP Product Code	Must be a NERC Curtailment Priority Indexed product associated with the TP Assignment Ref listed on this line.
TP Path	May be a valid path associated with the first of the transmission reservations to be consumed. If not provided, must be NULL.
TP Assignment Ref	Must be a valid reference to a previously arranged transmission reservation (or the phrase BUYATMARKET if the PSE wishes to request 0-NextHour service through the tag).
TP Product Level	May indicate the maximum amount (in whole megawatts) to be consumed from the first reservation. If not provided, must be NULL. If NULL, it is assumed that the entire available capacity from the reservation may be consumed, if needed.
Misc Info	May indicate additional information. If not provided, must be NULL.
Misc Ref	May indicate additional information. If not provided, must be NULL.

Subsequent rows of the Stack:

CA Code	Must be NULL.
TP Code	Must be NULL.
PSE Code	May be any appropriate Purchasing/Selling Entity. If not provided, must be NULL.
TP Product Code	Must be a NERC Curtailment Priority Indexed product associated with the TP Assignment Ref listed on this line.
TP Path	Must be NULL.
TP Assignment Ref	Must be a valid reference to a previously arranged transmission reservation (or the phrase BUYATMARKET if the PSE wishes to request 0-NextHour service through the tag).
TP Product Level	May indicate the maximum amount (in whole megawatts) to be consumed from the first reservation. If not provided, must be NULL. If NULL, it is

	assumed that the entire available capacity from the reservation may be consumed, if needed.
Misc Info	May indicate additional information. If not provided, must be NULL.
Misc Ref	May indicate additional information. If not provided, must be NULL.

Multiple reservations are consumed from top to bottom in the order they are listed.

When evaluating a path segment listing multiple OASIS reservations, the Transmission Provider shall ascribe MW capacity from the energy profile to the designated OASIS reservation up to the specified TP_PRODUCT_LEVEL or the energy profile amount, whichever is less. If TP_PRODUCT_LEVEL is not specified (NULL), the Transmission Provider shall ascribe MW capacity from the energy profile to the designated OASIS reservation up to the reserved capacity available for scheduling or the energy profile, whichever is less. Cross validation of TP_ASSIGNMENT_REF with TP_PRODUCT_CODE, PSE_CODE (if specified), and TP_PRODUCT_LEVEL (if specified) must be performed as part of the tag evaluation process, and shall not be performed as part of the syntactic validation checks executed by any of the Tag Services.

Example:

CA	TP	PSE	TP Product Code	TP Path	TP Assignment Ref	TP Product Level	Misc Info	Misc Ref
AAA	AAA	PQRPM	4-NW	AAA-BBB	1234	50		
		PQRPM	2-NH		5678			
		STVPM	2-NH		9012			

In the example above, assume an Energy Profile that varies from 50 MW from 08:00 to 09:00, 75 MW from 09:00 to 10:00, and 100 MW from 10:00 to 11:00. Further, assume that reservation 1234 has 100 MW available, reservation 5678 has 30 MW available, and reservation 9012 has 40 MW available.

During the first hour, 50 MW is needed to satisfy the needs of the transaction. The first reservation listed, 1234, has 100 MW available; 50 of which the Tag Author has listed are available for this transaction. In this first hour, only 50 MW of reservation 1234 are used. The other reservations are not consumed.

During the second hour, 75 MW is needed to satisfy the needs of the transaction. The first reservation listed, 1234, has 100 MW available; 50 of which the Tag Author has listed are available for this transaction. 25 MW is still required to meet the needs of the transaction. The second reservation, 5678, has 30 MW available. 25 MW of this will be used to meet the needs of the transaction, for a total of 75 MW being supplied by 1234 and 5678. 9012 has still not been consumed.

During the third hour, 100 MW is needed to satisfy the needs of the transaction. The first reservation listed, 1234, has 100 MW available; 50 of which the Tag Author has listed are available for this transaction. 50 MW is still required to meet the needs of the transaction. The second reservation, 5678,

has 30 MW available. As there has been no maximum specified in the TP Product Level, all 30 MW of this will be used to meet the needs of the transaction, for a total of 80 MW being supplied by 1234 and 5678. 20 MW is still required to meet the needs of the transaction. Reservation 9012 has 40 MW available, so 20 MW of this will be used. In this case, the first reservation has been consumed to its limit of 50 MW (leaving 50 MW available for other uses), the second has been consumed in its entirety, and the third has been consumed as needed to supply the remaining 20 MW.

3.3.2.7.2 Validation Rules

PROVIDER Rules (050800-050899)	
Rule	Rule Description
0508xx	Generic PROVIDER rule identifier, where xx represents any 2-digit number
050801	The PROVIDER table is required, but is not included.
050802	The PROVIDER table must always specify at least a generation record and a load record. If the table contains fewer than two (2) records, then return this error code.
050803	The CA_CODE must not be NULL in the first record of the PROVIDER table.
050805	The CA_CODE must not be NULL in the last record of the PROVIDER table.
050807	There must be at least nine (9) attributes in each PROVIDER record.
050808	There must be no more than nine (9) attributes in each PROVIDER record.
050809	The CA_CODE must be a registered CA code value, or NULL. Error message returned should include version number of NERC Registry consulted in making error decision.
050810	The TP_PRODUCT_CODE, if non-null, must contain a NERC Curtailment Priority indexed transmission product.
050811	The TP_CODE must be a registered TP code value, or NULL. Error message returned should include version number of NERC Registry consulted in making error decision.
050813	The PSE_CODE must be a registered PSE code value, or NULL. Error message returned should include version number of NERC Registry consulted in making error decision.
050818	The TP_PATH value must follow the formatting rules defined in the Attribute Dictionary.
050819	The TP_ASSIGNMENT_REF value must follow the formatting rules defined in the Attribute Dictionary.
050820	The TP_PRODUCT_LEVEL value must follow the formatting rules defined in the Attribute Dictionary.
050821	The MISC_INFO value must follow the formatting rules defined in the Attribute Dictionary.
050822	The MISC_REF value must follow the formatting rules defined in the Attribute Dictionary.
050823	The TP_PRODUCT_CODE must be non-NULL if the TP_ASSIGNMENT_REF is non-NULL.

PROVIDER Rules (050800-050899)	
Rule	Rule Description
050824	BUYATMARKET 0-NextHour Service was requested from a Transmission Provider that does not, according to the Master Registry, offer BUYATMARKET 0-NextHour Service.
050825	A PSE other than the Requestor was listed next to a Transmission Provider being asked to provide 0-NextHour Service via BUYATMARKET.
050826	At least one Transmission Provider (and associated Product and Assignment Reference) must be included in the Provider table.
050899	Undocumented PROVIDER table error.

3.3.2.8 ENERGY Table

Table	Attribute ID	Description
ENERGY	START_TIME	Start time (in CS) for described flow of energy.
	STOP_TIME	Stop time (in CS) for described flow of energy.
	MW_LEVEL	Describes level of demand in megawatts for the time period specified. Must be described in whole megawatts.
	MWH	Describes level of total demand in megawatt-hours for the time period specified. Must be described in whole megawatts. If not supplied, must be NULL.
	RAMP_START	Time (in CS) at which generator will begin ramping to desired level. If not supplied, must be NULL.
	RAMP_TIME	Duration (in minutes) over which ramping will occur to meet desired level. If not supplied, must be NULL.

The ENERGY table may have multiple rows.

3.3.2.8.1 Usage

The ENERGY table is used to describe the flow of energy over time. Multiple records are used to indicate the various levels over a defined time period.

The first field is used to define the time at which a flow of energy should begin. This field must always be encoded as a /time value. Specific formatting is defined in the Attribute Dictionary.

The second field is used to define the time at which a flow of energy should stop. This field must always be encoded as a /time value. Specific formatting is defined in the Attribute Dictionary.

The third field is used to represent the amount of demand in whole megawatts for the period defined by the times described above. This field must always be encoded as a numeric value. Specific formatting is defined in the Attribute Dictionary.

The fourth field is used to describe the total amount of demand in whole megawatt-hours for the period defined by the times described above. This field must always be encoded as a numeric value. Specific formatting is defined in the Attribute Dictionary.

The fifth field is used to describe the time at which generation ramping should occur. This field must always be encoded as a /time value. Specific formatting is defined in the Attribute Dictionary.

The sixth field is used to describe in whole minutes the duration over which ramping should occur. This field must always be encoded as a numeric value. Specific formatting is defined in the Attribute Dictionary.

3.3.2.8.2 Validation Rules

ENERGY Rules (050900-050999)	
Rule	Rule Description
0509xx	Generic ENERGY rule identifier, where xx represents any 2-digit number.
050901	The ENERGY table is required.
050902	If the ENERGY table is required, it must have at least one (1) record.
050903	A repeating profile must not exceed 24 hours. A continuous profile must not exceed the transaction STOP_DATE.
050904	There must be at least six (6) attributes in each ENERGY record.
050905	There must be no more than six (6) attributes in each ENERGY record.
050906	The START_TIME value must follow the formatting rules defined in the Attribute Dictionary.
050907	The STOP_TIME value must follow the formatting rules defined in the Attribute Dictionary.
050908	The MW_LEVEL value must follow the formatting rules defined in the Attribute Dictionary.
050909	The MWH value must follow the formatting rules defined in the Attribute Dictionary.
050910	The RAMP_START value must follow the formatting rules defined in the Attribute Dictionary.
050911	The RAMP_TIME value must follow the formatting rules defined in the Attribute Dictionary.
050999	Undocumented ENERGY table error.

3.3.2.9 LOSSES Table

Table	Attribute ID	Description
LOSSES	TP_CODE	Must be TSIN registered TP Code of provider for which losses are being described.
	START_TIME	Start time (in CS) of period for which losses are being described.
	STOP_TIME	Stop time (in CS) of period for which losses are being described.
	PATH	Represents path information for listed TP.
	MW_POR	Level of energy accepted in whole megawatts at Point of Receipt.
	MW_POD	Level of energy delivered in whole megawatts at Point of Delivery.
	LOSS_SUPPLY	Used to indicate type of loss provision being exercised. If not supplied, must be NULL.
	SUPPLY_REF	Used to provide additional information describing certain loss supplies. If not supplied, must be NULL.

The LOSSES table must have at least one row for each TP in the PROVIDER table.

3.3.2.9.1 Usage

The LOSSES table serves two primary functions. It identifies the type of loss accounting used by each transmission provider in the transaction path and the associated energy to be scheduled at the providers point of receipt and point of delivery interfaces. The LOSSES table is optional, meaning it need only be provided when both (1) losses have been incurred and (2) one or more of the transmission providers on the transaction path require accounting for the losses to be described within the ETAG. If no losses are incurred, or if all providers on the transaction path agree to the manner in which losses are to be handled, then the LOSSES table is not required. However, if any one provider requires losses to be shown, then losses must be shown for all providers.

If a TP is entered, non-zero MW_POR and MW_POD levels must be indicated (no percents or \$) and some sort of loss accounting must be represented (in-kind, financial, internal, or external).

The START_TIME and STOP_TIME totals cannot describe more than a single 24-hour period if a “weekday” is checked, but can have a “sliding” start.

The START_TIME and STOP_TIME values must either match the ENERGY table row for row or provide further granularity (i.e., the ENERGY table may specify a six hour start/stop period, but the LOSSES table may further divide this into six one hour start/stop periods). In either case, the LOSSES for the duration of the entire profile must be accounted for, with the first START_TIME and last

STOP_TIME of the LOSSES table equal to the first START_TIME and last STOP_TIME of the ENERGY table.

The PATH is required when the TP is supplying transmission on more than one “transaction path segment” and the value must be equal to the path name used in the TP_PATH field of the PROVIDER table.

The MW_POR for the first TP should be equal to the ENERGY profile, for all others, should be equal to MW_POD of immediate upstream TP.

The POD_MW may not become zero at any time.

There are four possible values for the LOSS_SUPPLY field. They are as follows:

- **<NULL> (In-Kind).** A NULL LOSS_SUPPLY indicates that the TP has chosen to schedule losses in kind. Losses are determined by comparing MW_POR and MW_POD.
- **FIN (Financial).** A LOSS_SUPPLY of “FIN” represents that the losses have been purchased from the TP via provisions in his tariff. A non-null SUPPLY_REF can accompany a LOSS_SUPPLY of “FIN,” but is not required.
- **INT (Internal).** A LOSS_SUPPLY of “INT” represents that the losses have been purchased from an entity other than the TP who can, by virtue of being directly connected to the TP’s network, schedule losses directly onto the TP’s grid. A non-null SUPPLY_REF indicating the entity supplying the losses can accompany a LOSS_SUPPLY of “INT”, but is not required.
- **EXT (External).** A LOSS_SUPPLY of “EXT” represents that the losses have been purchased from an entity other than the TP that is external to the TP’s network. The SUPPLY_REF in this case is required and should be the NERC TAG_ID for the transaction supplying the losses (although no particular value is required or validated against). The loss TAG must be separately submitted and approved by all parties including the TP. The loss transaction does not necessarily have to exactly match the transaction timetable unless so required by the TP.

This specification does not require all Transmission Providers to have these services, but if these services are in their tariffs, this method of identification must be used.

3.3.2.9.2 Validation Rules

LOSSES Rules (051000-051099)	
Rule	Rule Description
0510xx	Generic LOSSES rule identifier, where xx represents any 2-digit number.
051001	A repeating profile must not exceed 24 hours. A continuous profile must not exceed the transaction STOP_DATE.
051002	The LOSSES profile must be entirely contained within the ENERGY profile.
051003	There must be at least eight (8) attributes in each LOSSES record.
051004	There must be no more than eight (8) attributes in each LOSSES record.

LOSSES Rules (051000-051099)	
Rule	Rule Description
051005	The TP_CODE value must be a registered TP acronym. Error message returned should include version number of NERC Registry consulted in making error decision.
051006	The TP_CODE value must appear in the PROVIDER table.
051007	The TP_CODE values must appear in the same order that they appear in the PROVIDER table.
051008	The START_TIME value must follow the formatting rules defined in the Attribute Dictionary.
051009	The STOP_TIME value must follow the formatting rules defined in the Attribute Dictionary.
051010	The Path value must follow the formatting rules defined in the Attribute Dictionary.
051011	The Path value must appear in the PROVIDER table, with the same TP_CODE.
051012	The TP_CODE and Path values must appear in the same order that they appear in the PROVIDER table.
051013	The MW_POR value must follow the formatting rules defined in the Attribute Dictionary.
051014	The MW_POD value must follow the formatting rules defined in the Attribute Dictionary.
051015	The LOSS_SUPPLY value must follow the formatting rules defined in the Attribute Dictionary.
051016	The SUPPLY_REF value must follow the formatting rules defined in the Attribute Dictionary.
051099	Undocumented LOSSES table error.

3.3.2.10 ADJUST Table

Table	Attribute ID	Description
ADJUST	ADJUST_DATETIME	Date and time (in CS) when adjustment was issued.
	START_DATETIME	Start date and time (in CS) of period for which adjustment is being described.
	STOP_DATETIME	Stop date and time (in CS) of period for which adjustment is being described.
	MW_CAP	Indicates in whole megawatts the highest level at which a transaction will run. If not present, must be NULL.
	ENTITY_TYPE	Must indicate the entity type of the adjusting entity.
	ENTITY_CODE	Provides further information describing the identity of the adjusting party. If not present, must be NULL.
	OPERATOR_ID	Must be a reference to representative of the entity that issued the adjustment.
	REASON	Must provide information explaining necessity of adjustment, such as TLR Incident Ref numbers, constraint, etc...

The ADJUST table may contain multiple rows.

3.3.2.10.1 Usage

If the ADJUST Table is included, it describes a MW_CAP to limit the Energy Profile for the points in time described by the START_DATETIME and STOP_DATETIME. Both of these DATETIMES must fall within the window defined by the derived times described in the COMPOSITE START and STOP DATETIMES.

ADJUST records are evaluated for a given point in time by looking at the MW_CAP from the most recent ADJUST record that affects the point in time in question, as determined by the ADJUST_DATETIME. This value is then compared to the MW_LEVEL value specified in the Energy Profile for the same point in time in question. The lower of the two values becomes the Derived MW_LEVEL. If a Null MW_CAP value is the most recent value specified, the MW_LEVEL from the Energy Profile is used.

Reloading may be accomplished by resetting the MW_CAP for a future point in time for the transaction. For example:

Incident GHI_050197_1400 causes a Level 3 TLR to be initiated at 14:00 on May 1, 1997. Transaction ABC_PQRPM1234567_GHI describes a transaction that runs 100 MW from 06:00 to 22:00. It must be curtailed by 50MW. It is unknown how long the TLR will last. The ADJUST table would appear as follows:

```
ADJUST,{
```

```
05/01/1997 14:01:22,05/01/1999 14:00,05/01/1999 22:00,50,"CA","GHI","John Doe", "TLR
GHI_050197_1400"
```

```
},1
```

Since the duration of the TLR is unknown, the STOP_DATETIME from the COMPOSITE table of the transaction has been chosen to be the STOP_DATETIME for the ADJUST table. In this manner, the ADJUST is in effect until another ADJUST takes place to supercede it. Another option would be to add ADJUST records that modify the schedule on an hour-by-hour basis, letting the later hours in the schedule continue unless an associated ADJUST record is found.

At 15:00, the TLR is released. In order to restore the schedule to its original levels, an ADJUST record is inserted with a Null MW_CAP value. The ADJUST table should now appear as follows:

```
ADJUST,{
```

```
05/01/1997 14:01:22,05/01/1999 14:00,05/01/1999 22:00,50,"CA","GHI","John Doe", "TLR
GHI_050197_1400"
```

```
05/01/1997 14:58:43,05/01/1999 15:00,05/01/1999 22:00,,"CA","GHI","John Doe", "TLR Released"
```

```
},2
```

The transaction is restored to 100 MWs for 15:00 to 22:00 by virtue of a more recent ADJUST record that specifies a null MW_CAP, thereby overriding the prior 50 MW MW_CAP.

3.3.2.10.2 Validation Rules

ADJUST Rules (051100-051199)	
Rule	Rule Description
0511xx	Generic ADJUST rule identifier, where xx represents any 2-digit number.
051101	The ADJUST Start Datetime must be greater than or equal to the Composite Start Datetime and may not be NULL.
051102	The ADJUST Stop Datetime must be less than or equal to the Composite Stop Datetime and may not be NULL.
051103	There must be at least eight (8) attributes in each ADJUST record.
051104	There must be no more than eight (8) attributes in each ADJUST record.
051105	The ENTITY_TYPE value must be valid as defined in the Attribute Dictionary.
051106	The ENTITY_CODE value, when paired with the above ENTITY_TYPE must represent a valid NERC registered acronym.
051107	MW_CAP must follow the formatting rules defined in the Attribute Dictionary.
051108	OPERATOR_ID must follow the formatting rules defined in the Attribute Dictionary.
051109	REASON must follow the formatting rules defined in the Attribute Dictionary.

ADJUST Rules (051100-051199)	
Rule	Rule Description
051110	ADJUST START Datetime must be less than the ADJUST Stop Datetime
051199	Undocumented ADJUST table error.

3.3.2.11 REPLACE Table

Table	Attribute ID	Description
REPLACE	TAG_ID	Specifies the tag by Tag ID that is to be acted upon based on the COMPOSITE state of the tag in which this table appears.

The REPLACE table is created by the Tag Authority service based on information supplied by the Tag Agent on initial submission of the tag (i.e., this table is only created if one or more valid “replace pairs” were included in the submission). A single new tag may replace multiple tags with each tag to be replaced specified in separate records in the REPLACE table.

3.3.2.11.1 Usage

The REPLACE table lists tags that are to be Canceled and/or Terminated upon the successful implementation (through movement of the replacing tag’s state to IMPLEMENT or CONDITIONAL). Tag IDs must follow the formatting rules defined in the data dictionary.

3.3.2.11.2 Validation Rules

REPLACE Rules (051200-051299)	
Rule	Rule Description
0512xx	Generic REPLACE rule identifier, where xx represents any 2-digit number.
051201	The TAG_ID field must be a validly formatted Tag ID.
051299	Undocumented REPLACE table error.

3.3.2.11.3

3.3.3 Data Model Extensions

Recognizing the need for an open document definition in order to promote a flexible development environment, the following rules have been established in order to maintain the integrity of the data format. Software developers must follow these rules in order for their software to meet acceptance criteria.

- The Data Encoding structure used to communicate tag data must be maintained.
- All data must be contained inside the Data Encoding structure.
- Any additional data must be defined in the standardized Table format, as defined in the Data Encoding structure.

- Any newly defined Tables must be registered with NERC before being used to avoid duplicate table names.
- New Tables cannot use names defined by or registered with NERC.
- No extension to the Data Model shall be required as a condition to performing the functional requirements as defined in this document of any Tag Agent, approval, or Authority Service.
- Tables may not replace existing tables or reformat existing tables; they must provide information not contained in the existing table set. Duplication of information already stored within the tag is not allowed.

To accommodate these potential extensions to the Data Model, the Tag Authority Service must maintain the tag and all associated tables as it was originally received. With the exception of the Tag Authority Service's addition of tables as described in this document, the Tag Authority Service must forward the complete tag with all tables intact as it was originally received. The Tag Agent and Approval Services must ignore the presence of any tables that they do not recognize.

3.3.4 Intra-Table Validation Rules

Intra-Table Validation rules are special rules that involve more than one table due to some dependency.

Intra-Table Validation Rules (509900-509999)	
Rule	Rule Description
5099xx	Generic Intra-Table Validation rule identifier, where xx represents any 2-digit number.
509901	Tags specified as Repeating in the TAG table may not have an ENERGY table specifying more than 24 hours.
509902	Repeat Days in the TAG table, and specified Start/Stop dates in the TAG table, and records from the ENERGY table are in conflict. Extrapolation generates a zero megawatt tag.
509999	Undocumented Intra-Table Validation Error.

3.3.5 The END OF DATA Marker

3.3.5.1.1 Usage

The END OF DATA marker is used to communicate to a service that data transferred represents all tag data known to exist for a particular Tag ID. The END OF DATA marker is defined as the simple non-quoted string:

END

When present, the END OF DATA marker must be followed by a Line Terminator as defined in Appendix A. There may be only one END OF DATA marker in a tag.

3.3.5.1.2 Validation Rules

END OF DATA MARKER Rules (1300-1399)

Rule	Rule Description
0513xx	Generic TAG rule identifier, where xx represents any 2-digit number.
051301	End of data marker must be present when message requires a complete tag data set.
051399	Undocumented END OF DATA MARKER table error.

3.3.6 Attribute Dictionary

There are several data types in use in the data model. These are defined as:

TIME	CS Time, defined in HH:MM format. Valid times are 00:00 to 23:59. <i>24:00 Is Invalid.</i>
DATE	CS Date, defined in MM/DD/YYYY.
DATETIME	CS Date/Time, defined in MM/DD/YYYY HH:MM. Rules for both TIME and DATE apply.
P_DATETIME	Precise CS Date/Time, defined in MM/DD/YYYY HH:MM:SS. Valid times are 00:00:00 to 23:59:59
POS_INT	Whole number used to represent a quantity of indivisible (for the purposes of this document) units. The set of all positive integers. Limited to nine digits in length.
WHOLE	Whole number used to represent a quantity of indivisible (for the purposes of this document) units. The set of all positive integers and zero. Limited to nine digits in length.
STRING	Array of characters used to represent non-numeric data.

Value restrictions describe rules that identify what may or may not be contained in a certain string value. They are described using augmented Backus-Naur Form, which is explained in detail in Appendix A.

Attribute ID	Data Type	Value Restriction
ADJUST_DATETIME	P_DATETIME	N/A
CA_CODE	STRING	1*4UPALPHANUM
CONTROL	STRING	3 (“Y” “N”)
DAY_REPEAT	STRING	7 (“Y” “N”)
DEAL_REF	STRING	1*20TEXT
DISTRIBUTE_METHOD	STRING	“ASSESS” “INFO”
ENTITY_CODE	STRING	1*6UPALPHANUM
ENTITY_STATE	STRING	APPROVAL_STATE
ENTITY_TYPE	STRING	(“CA” “TP” “PSE” “SC”)
FAX	STRING	PHONESTRING
LOSS_SUPPLY	STRING	“FIN” “INT” “EXT”
MISC_INFO	STRING	1*20TEXT
MISC_REF	STRING	1*20TEXT
MW_CAP	WHOLE	N/A
MW_LEVEL	WHOLE	N/A
MW_POD	POS_INT	N/A
MW_POR	POS_INT	N/A
MWH	WHOLE	N/A
NOTIFY_METHOD	STRING	“NOTIFY”
OPERATOR_ID	STRING	1*25TEXT
PATH	STRING	1*50TEXT
PHONE	STRING	PHONESTRING
PHONE24	STRING	PHONESTRING
PSE_CODE	STRING	1*6UPALPHANUM

RAMP_START	TIME	N/A
RAMP_TIME	POS_INT	N/A
REASON	STRING	1*80TEXT
REMARKS	STRING	1*80TEXT
SINK_CODE	STRING	1*12UPALPHANUM
SOURCE_CODE	STRING	1*12UPAPLHANUM
START_DATE	DATE	N/A
START_DATETIME	DATETIME	N/A
START_TIME	TIME	N/A
STATE	STRING	COMPOSITE_STATE
STATE_DATETIME	P_DATETIME	N/A
STOP_DATE	DATE	N/A
STOP_DATETIME	DATETIME	N/A
STOP_TIME	TIME	N/A
SUBMIT_DATETIME	P_DATETIME	N/A
SUPPLY_REF	STRING	1*25TEXT
TAG_CODE	STRING	7UPALPHANUM
TAG_ID	STRING	TAG_ID
TIME_ZONE	STRING	“CS”
TP_ASSIGNMENT_REF	STRING	1*20TEXT
TP_CODE	STRING	1*4UPAPHANUM
TP_PATH	STRING	1*50TEXT
TP_PRODUCT_CODE	STRING	DIGIT 0*11TEXT
TP_PRODUCT_LEVEL	POS_INT	N/A
VERSION	STRING	“1.6”

Appendix A — TMP Protocol Specification

A.1 Augmented BNF

All of the mechanisms specified in the previous document are described below in augmented Backus-Naur Form (BNF) similar to that used by RFC 1945. Implementers will need to be familiar with the notation in order to understand this specification. The augmented BNF includes the following constructs:

name = definition

The name of a rule is simply the name itself (without any enclosing “<“ and “>“) and is separated from its definition by the equal “=” character. White space is only significant in that indentation of continuation lines is used to indicate a rule definition that spans more than one line. Certain basic rules are in uppercase, such as SP, CRLF, DIGIT, ALPHA, etc. Angle brackets are used within definitions whenever their presence will facilitate discerning the use of rule names.

“literal”

Quotation marks surround literal text. Unless stated otherwise, the text **MUST** be in upper case.

rule1 | rule2

Elements separated by a bar (“|”) are alternatives, e.g., “yes | no” will accept yes or no.

(rule1 rule2)

Elements enclosed in parentheses are treated as a single element. Thus, “(elem (foo | bar) elem)” allows the token sequences “elem foo elem” and “elem bar elem”.

*rule

The character “*” preceding an element indicates repetition. The full form is “<n>*<m>element” indicating at least <n> and at most <m> occurrences of element. Default values are 0 and infinity so that “*(element)” allows any number, including zero; “1*element” requires at least one; and “1*2element” allows one or two.

N rule

Specific repetition: “<n>(element)” is equivalent to “<n>*<n>(element)”; that is, exactly <n> occurrences of (element). Thus 2DIGIT is a 2-digit number, and 3ALPHA is a string of three alphabetic characters.

A.2 Basic Rules

The following rules are used throughout this specification to describe basic parsing constructs. The US-ASCII coded character set is defined by ANSI X3.4-1986.

OCTET	=	Any 8-bit sequence of data
CHAR	=	Any US-ASCII character (octets 0 - 127)
UPALPHA	=	Any US-ASCII uppercase letter “A”..”Z”
LOALPHA	=	Any US-ASCII lowercase letter “a”..”z”
ALPHA	=	UPALPHA LOALPHA
TEXT	=	Excepting ASCII Character 34, any single US ASCII Character with a value of greater than or equal to 32, but less than or equal to 126
DIGIT	=	Any US-ASCII digit “0”..”9”
INT_DIGIT	=	Any single US-ASCII digit, excluding zero (“1..9”)
CR	=	US ASCII Character 13 (Carriage Return)
LF	=	US ASCII Character 10 (Line Feed)
CRLF	=	(CR LF)
LT	=	(CRLF LF CR CR LF)
SP	=	US ASCII Character 32 (Space)
<“>	=	US ASCII Character 34 (Double Quote)
<(>	=	US ASCII Character 40 (Open Parenthesis)
<)>	=	US ASCII Character 41 (Close Parenthesis)
<,>	=	US ASCII Character 44 (Comma)
<->	=	US ASCII Character 45 (Dash)
<.>	=	US ASCII Character 46 (Period)
</>	=	US ASCII Character 47 (Forward Slash)
<:>	=	US ASCII Character 58 (Colon)
<:>	=	US ASCII Character 59 (Semi-Colon)
<X>	=	US ASCII Character 88 (Uppercase X)
<_>	=	US ASCII Character 95 (Underscore)
<{>	=	US ASCII Character 123 (Open Brace)
<}>	=	US ASCII Character 125 (Close Brace)

A.3 Basic Data Types

This describes, in BNF, the information from section 3.3.7, with regard to definitions of attributes

```

TIME           = 2DIGIT <:> 2DIGIT
DATE           = 2DIGIT </> 2DIGIT </> 4DIGIT
DATETIME      = DATE SP TIME
P_DATETIME    = DATE SP TIME <:> 2DIGIT
POS_INT       = INT_DIGIT *8DIGIT
WHOLE         = 1*9DIGIT
STRING        = 1*TEXT

```

A.4 Basic Data Encoding Rules

This describes, in BNF, the information from section 3.2.

```

QUOTEDSTRING  = <"> STRING <">
NUMBER        = (POS_INT | WHOLE)
DATEORTIME    = (TIME | DATE | DATETIME | P_DATETIME)

FIELD         = (QUOTEDSTRING | NUMBER | DATEORTIME)
RECORD        = *(FIELD <,>) FIELD LT
TABLENAME     = STRING <,> <{}> LT
TABLECLOSE    = <{}> <,> POS_INT LT
TABLE         = TABLENAME 1*RECORD TABLECLOSE

HEADER        = QUOTEDSTRING <,> QUOTEDSTRING <,> QUOTEDSTRING LT
END_OF_DATA_MARKER = "END" LT

TAG_DATA      = (HEADER 1*TABLE END_OF_DATA_MARKER)
               | (HEADER 1*TABLE)
               | (HEADER)

```

A.5 TMP Messages

A TMP message may either be a request or a response.

```

TMP_MESSAGE   = (TMP_REQUEST | TMP_RESPONSE)

```

A.6 TMP Requests

There are several different types of request, as defined in section 2.4.3.

TAG_ID	=	1*4UPALPHANUM <_> 1*6UPALPHANUM 7UPALPHANUM <_> 1*4UPALPHANUM
TAG_KEY	=	1*6UPALPHANUM 12ALPHANUM
TARGET_ENTITY	=	1*6UPAPLHANUM
OPERATOR_ID	=	QUOTEDSTRING
REASON	=	QUOTEDSTRING
CANCEL_DATETIME	=	DATETIME
START_DATETIME	=	DATETIME
STOP_DATETIME	=	DATETIME
MW_CAP	=	WHOLE
REPLACE_PAIR	=	TAG_ID <,> TAG_KEY
REPLACE_DATA	=	<{> REPLACE_PAIR *(<,> REPLACE_PAIR) <}>
SUBMIT_HEADER	=	“SUBMIT” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY *1REPLACE_DATA LT
SUBMIT_FOOTER	=	“SUBMIT_END” CRLF
SUBMIT	=	SUBMIT_HEADER TAG_DATA SUBMIT_FOOTER
ASSESS_HEADER	=	“ASSESS” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
ASSESS_FOOTER	=	“ASSESS_END” CRLF
ASSESS	=	ASSESS_HEADER TAG_DATA ASSESS_FOOTER
INFO_HEADER	=	“INFO” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
INFO_FOOTER	=	“INFO_END” CRLF
INFO	=	INFO_HEADER TAG_DATA INFO_FOOTER
UPDATE_HEADER	=	“UPDATE” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
UPDATE_MESSAGE	=	<“> (“APPROVED” “DENIED” “STUDY”) <“> <,> OPERATOR_ID <,> *1REASON LT
UPDATE_FOOTER	=	“UPDATE_END” CRLF
UPDATE	=	UPDATE_HEADER UPDATE_MESSAGE UPDATE_FOOTER
NOTIFY_HEADER	=	“NOTIFY” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
NOTIFY_FOOTER	=	“NOTIFY_END” CRLF
NOTIFY	=	NOTIFY_HEADER TAG_DATA NOTIFY_FOOTER

TMP Requests (continued)

IMPLEMENT_HEADER	=	“IMPLEMENT” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
IMPLEMENT_FOOTER	=	“IMPLEMENT_END” CRLF
IMPLEMENT	=	IMPLEMENT_HEADER TAG_DATA IMPLEMENT_FOOTER
STATUS_HEADER	=	“STATUS” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
STATUS_FOOTER	=	“STATUS_END” CRLF
STATUS	=	STATUS_HEADER STATUS_FOOTER
DSTATUS_HEADER	=	“DSTATUS” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
DSTATUS_FOOTER	=	“DSTATUS_END” CRLF
DSTATUS	=	DSTATUS_HEADER DSTATUS_FOOTER
CANCEL_HEADER	=	“CANCEL” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
CANCEL_MESSAGE	=	CANCEL_DATETIME <,> OPERATOR_ID <,> REASON LT
CANCEL_FOOTER	=	“CANCEL_END” CRLF
CANCEL	=	CANCEL_HEADER CANCEL_MESSAGE CANCEL_FOOTER
ADJUST_HEADER	=	“ADJUST” SP TARGET_ENTITY SP TAG_ID SP TAG_KEY LT
ADJUST_MESSAGE	=	START_DATETIME <,> STOPDATETIME <,> MWCAP <,> OPERATOR_ID <,> REASON LT
ADJUST_FOOTER	=	“ADJUST_END” CRLF
ADJUST	=	ADJUST_HEADER ADJUST_MESSAGE ADJUST_FOOTER
ADJUST_LIST_HEADER	=	“ADJUST_LIST” SP TARGET_ENTITY SP TAG_KEY LT
ADJUST_LIST_MESSAGE	=	<”> TAG_ID <”> <,> START_DATETIME <,> STOP_DATETIME <,> MWCAP <,> OPERATOR_ID <,> REASON LT
ADJUST_LIST_FOOTER	=	“ADJUST_LIST_END” CRLF
ADJUST_LIST	=	ADJUST_LIST_HEADER 1*ADJUST_LIST_MESSAGE ADJUST_LIST_FOOTER
ADJUST_CHECK_HEADER	=	“ADJUST_CHECK” SP ”IDC” SP TAG_KEY LT
ADJUST_CHECK_FOOTER	=	“ADJUST_CHECK_END” CRLF
ADJUST_CHECK	=	ADJUST_CHECK_HEADER ADJUST_CHECK_FOOTER
TMP_REQUEST	=	(SUBMIT ASSESS INFO UPDATE NOTIFY IMPLEMENT STATUS DSTATUS CANCEL ADJUST ADJUST_LIST ADJUST_CHECK)

A.7 TMP Responses

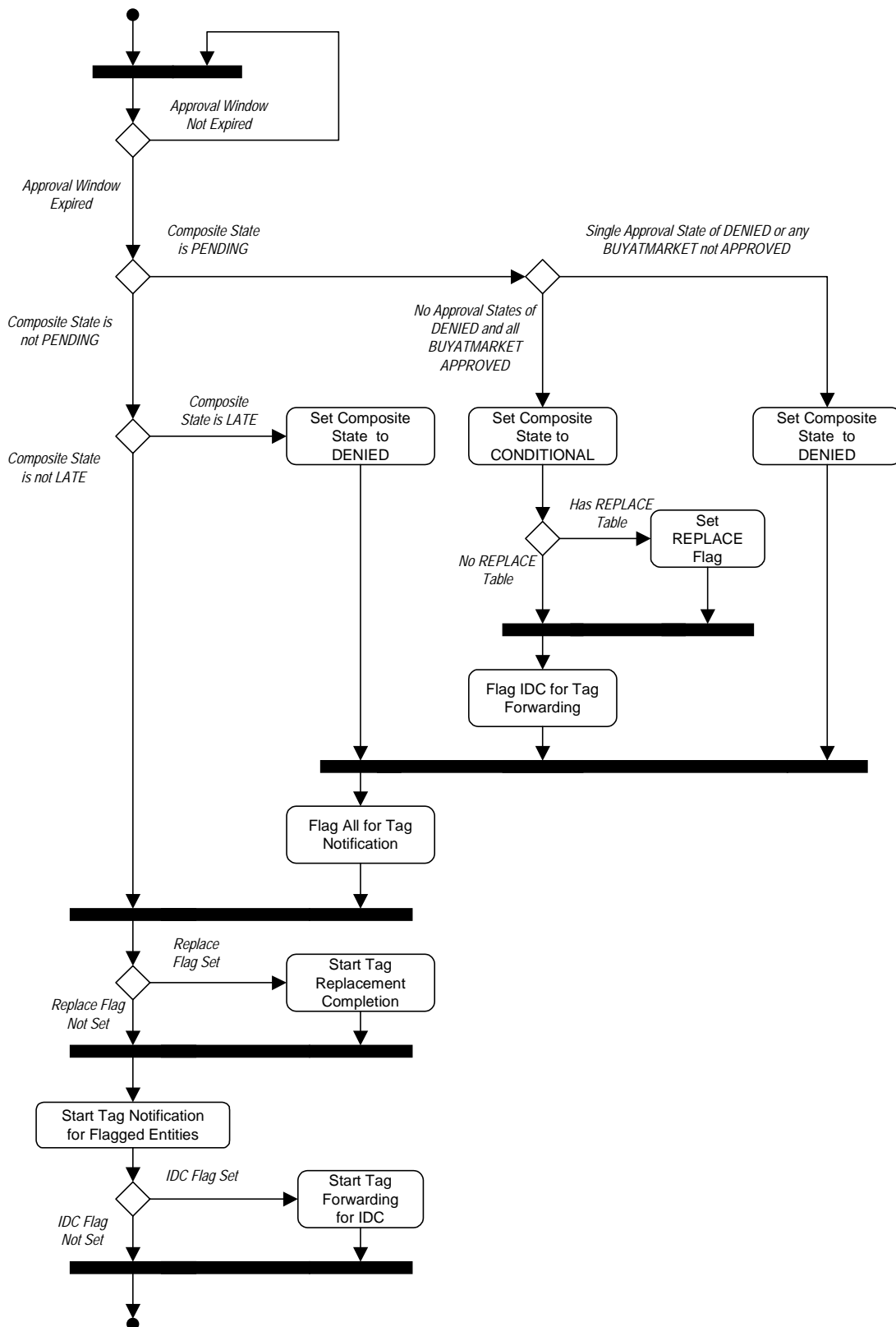
For each request, there will be an appropriate response. These responses are described in section 2.4.3.

SUCCESS_HEADER	=	“SUCCESS” LT
SUCCESS_MESSAGE	=	*1TAG_DATA
SUCCESS_FOOTER	=	“SUCCESS_END” CRLF
SUCCESS	=	SUCCESS_HEADER SUCCESS_MESSAGE SUCCESS_FOOTER
FAIL_HEADER	=	“FAIL” LT

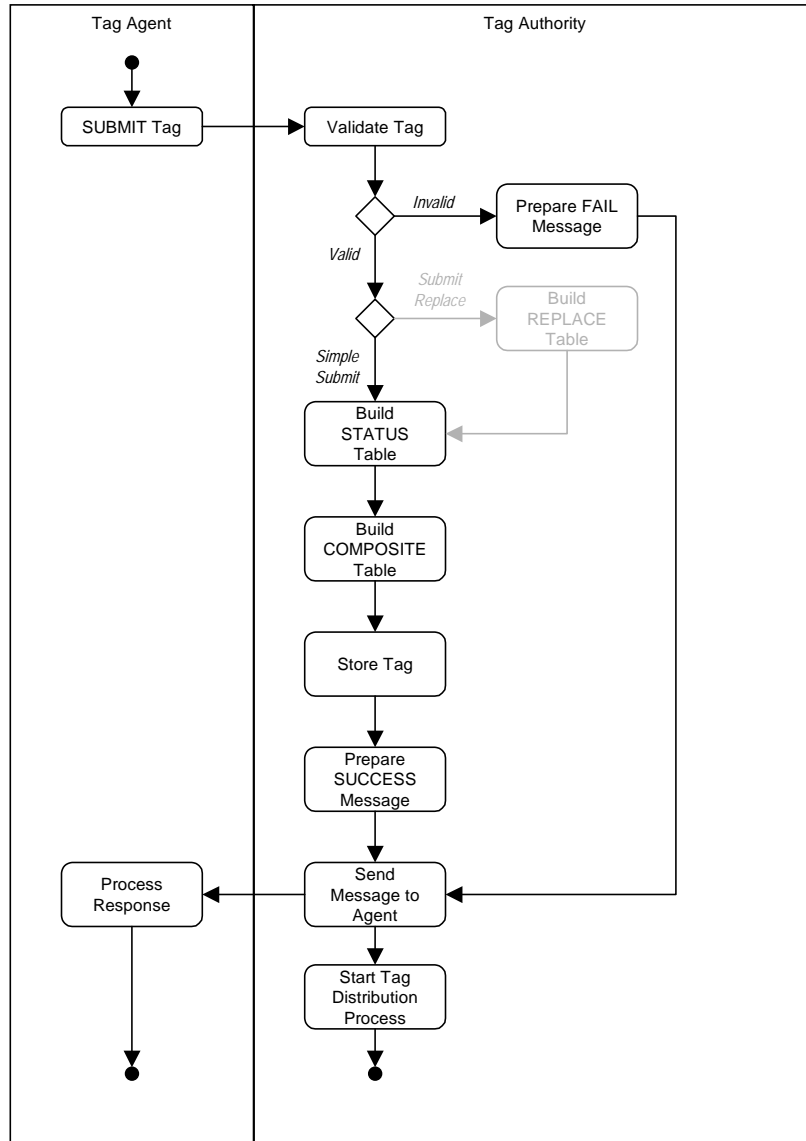
Appendix B — Activity Diagrams

The diagrams on the following pages illustrate the various E-Tag Functional Concepts from a messaging and processing point of view.

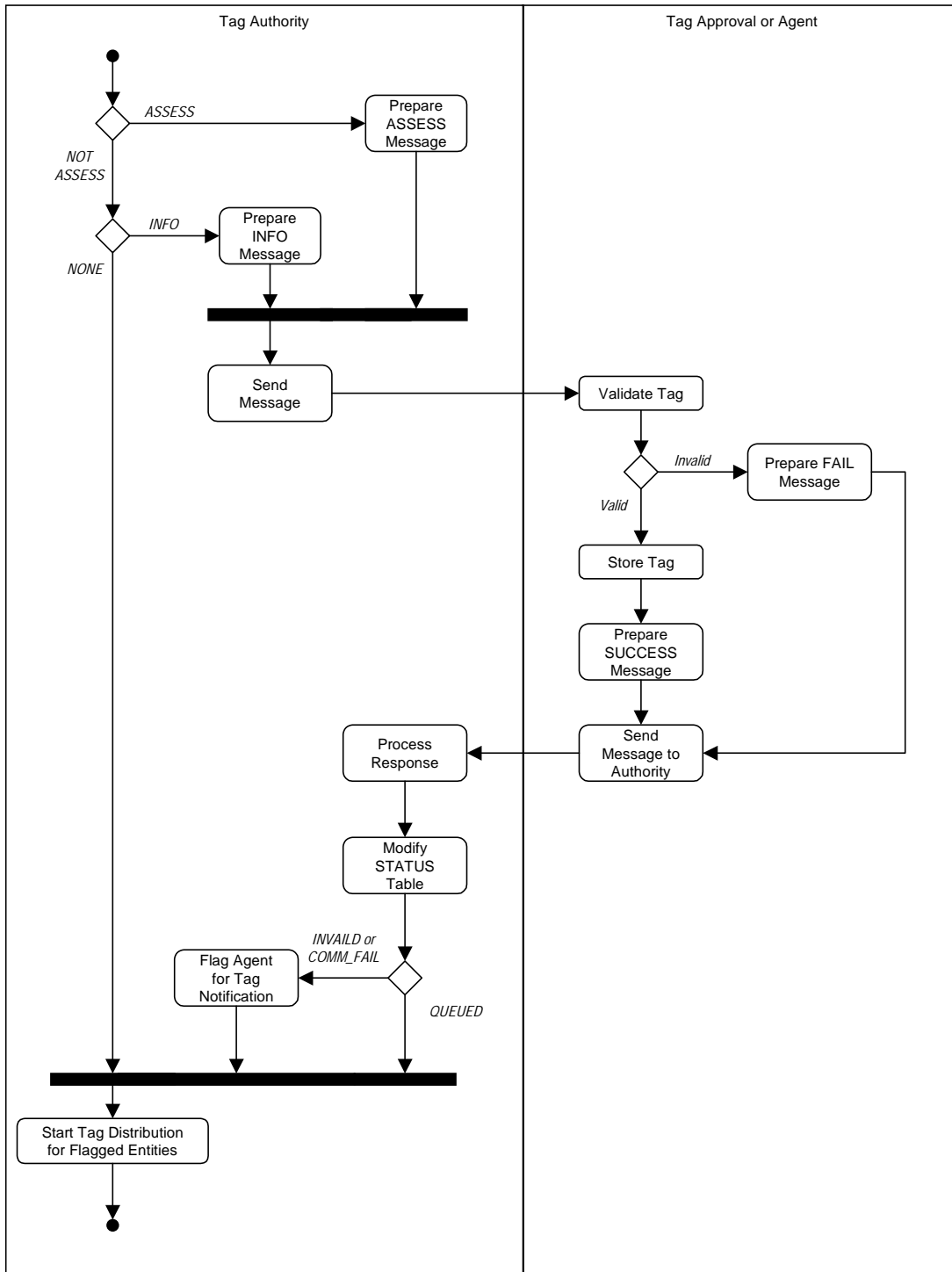
B.1 Tag State Management



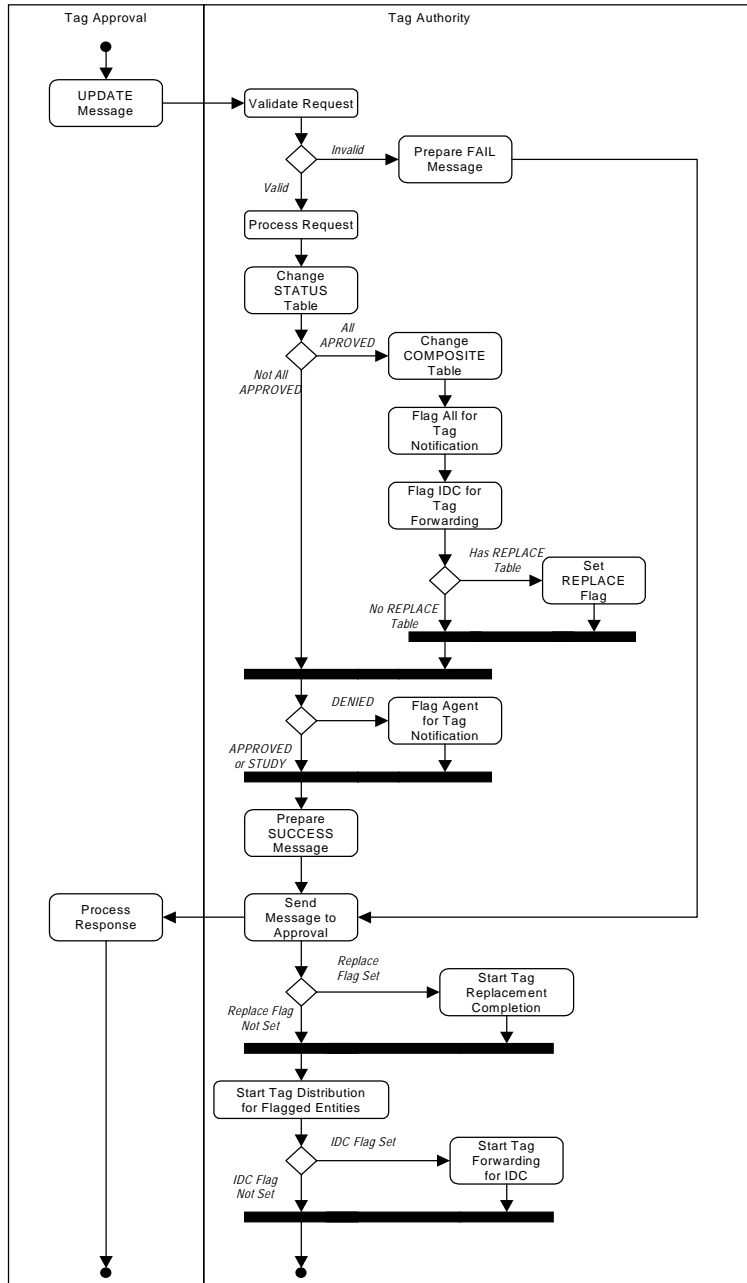
B.2 Tag Submission



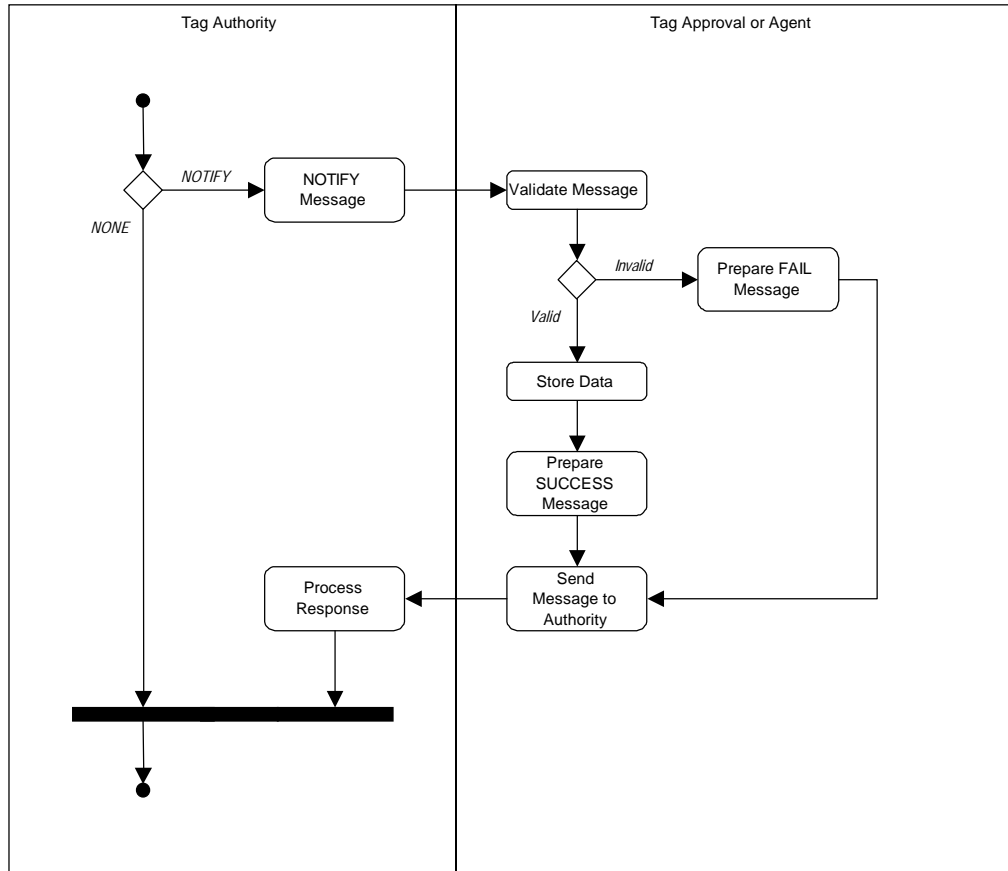
B.3 Tag Distribution



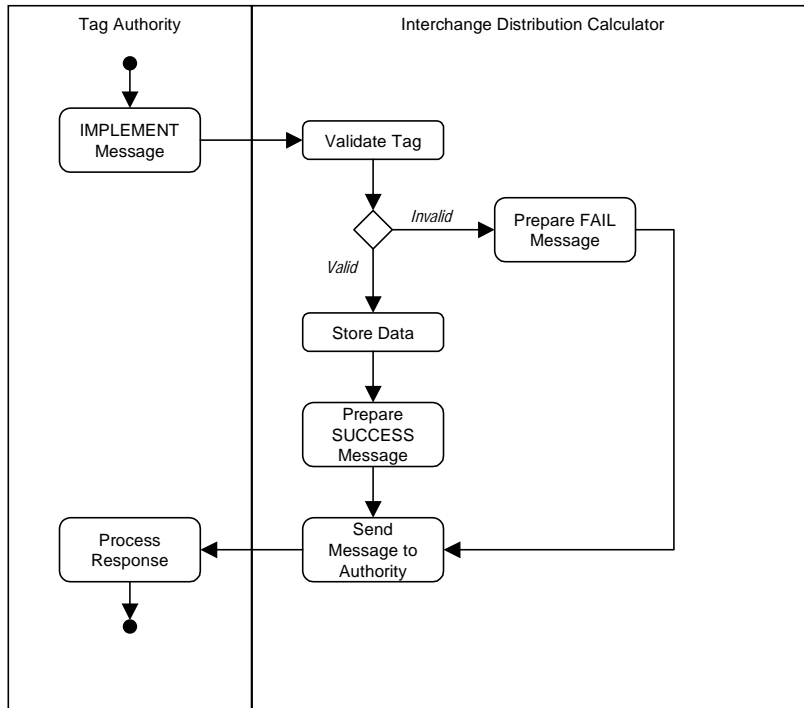
B.4 Tag Evaluation



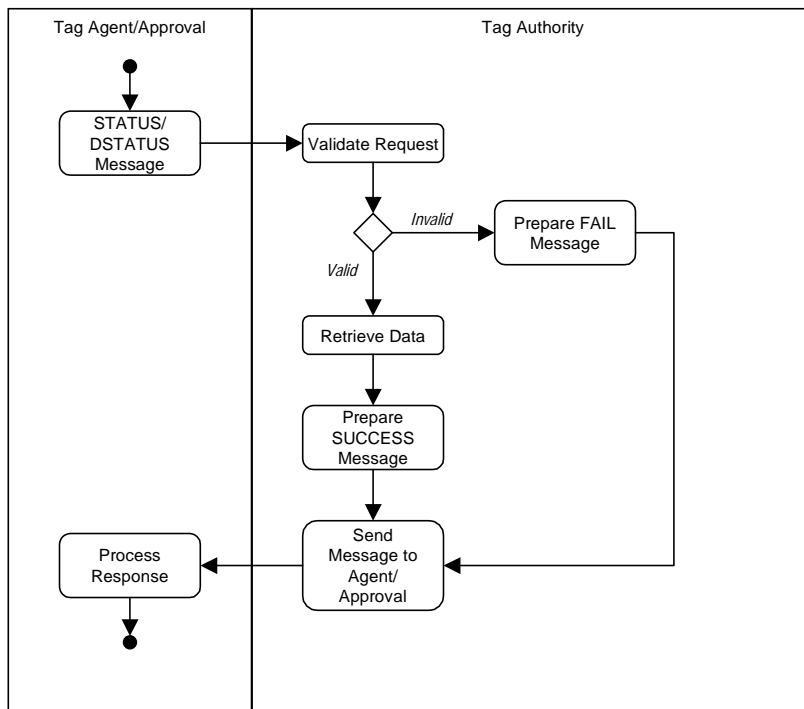
B.5 Tag Notification



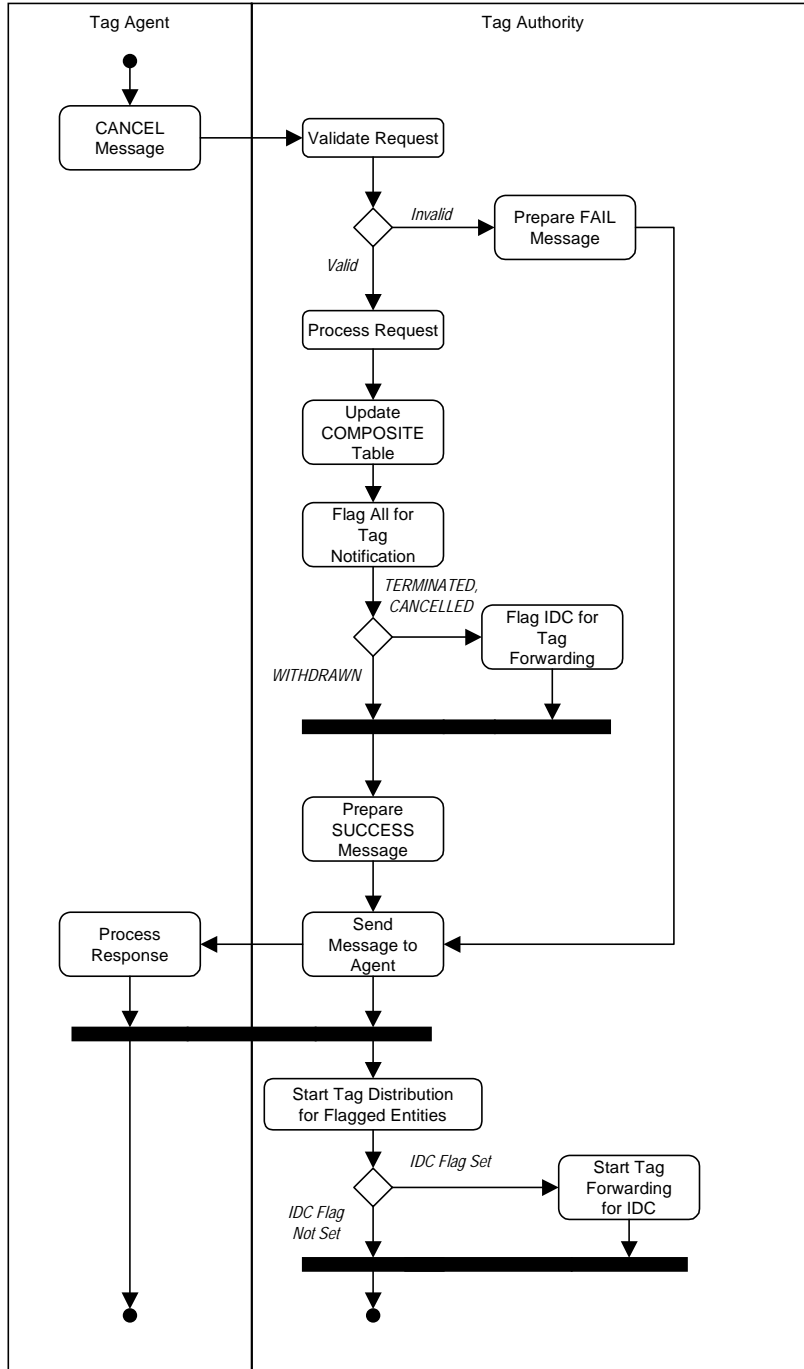
B.6 Tag Forwarding



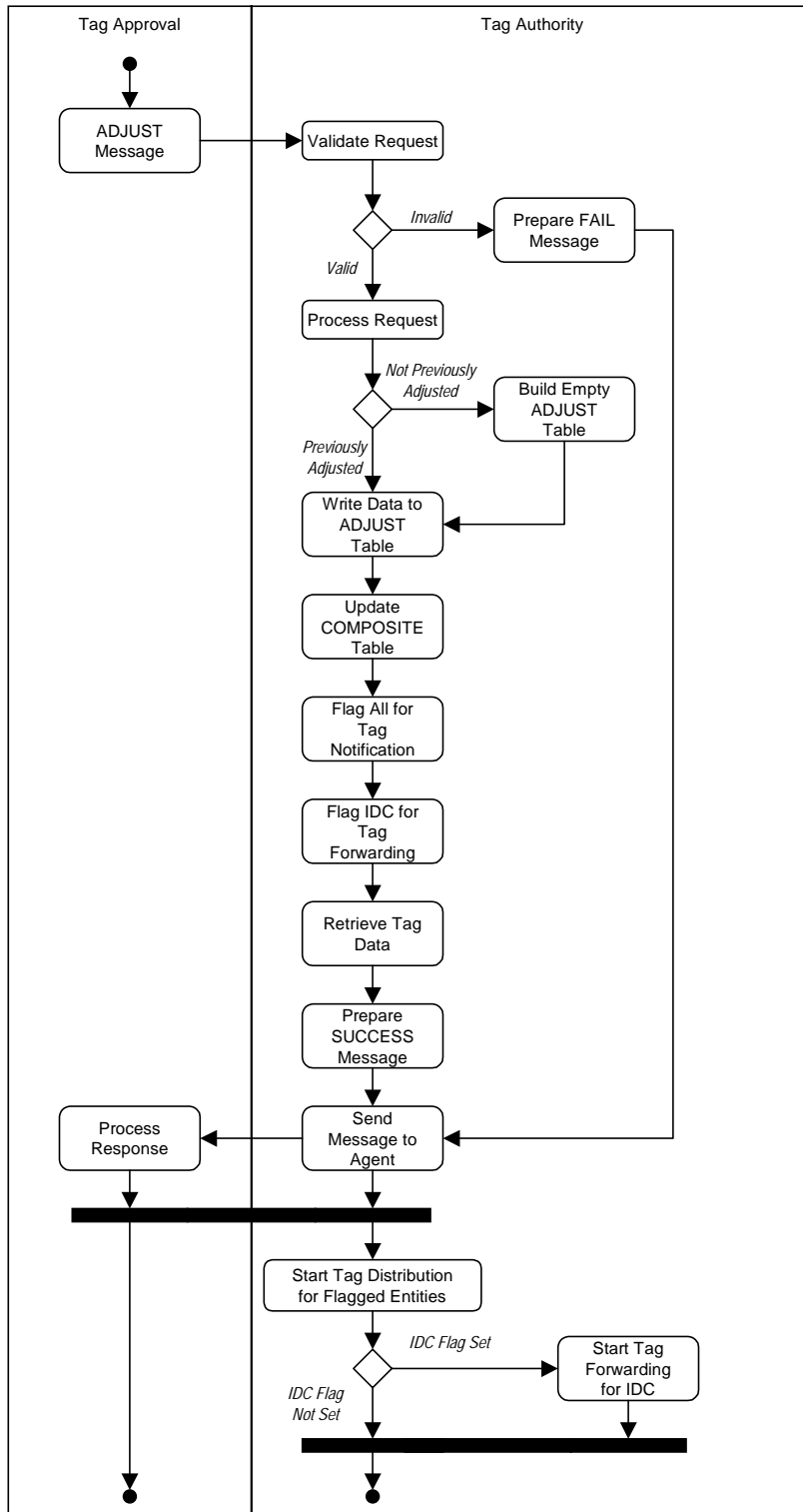
B.7 Tag Querying



B.8 Tag Cancellation

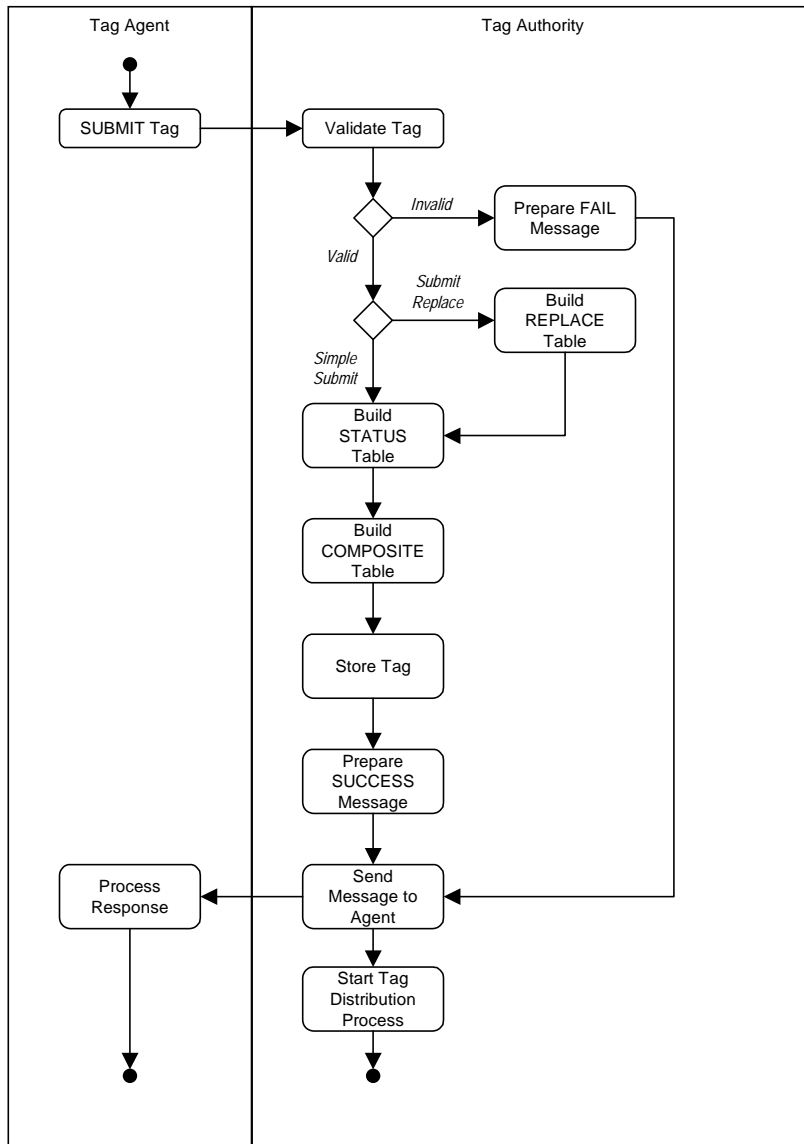


B.9 Tag Adjustment

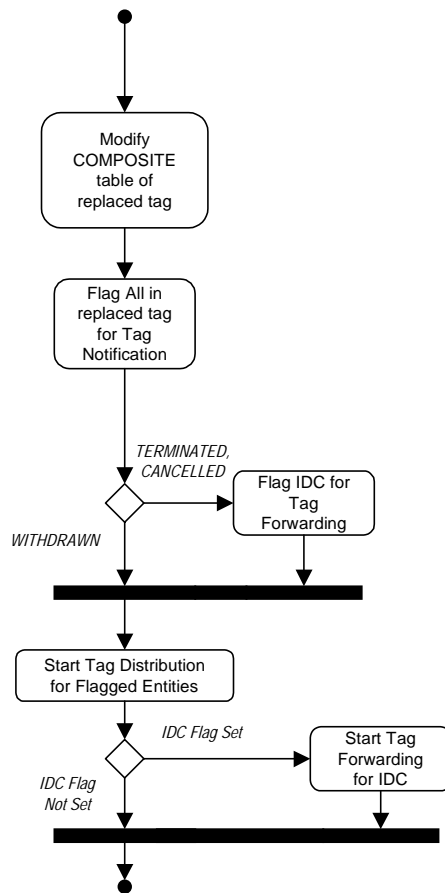


B.10 Tag Replacement

B.10.1 Replacement Initialization



B.10.2 Replacement Completion



Appendix C — Sample TMP Dialogs

The dialogs below illustrate TMP messaging required to implement functions described in Section 1.4.

C.1 State Management

No TMP Dialogs Necessary

C.2 Tag Submission

Tag Agent Service:

```
SUBMIT DDDD AAAA_PPPPPP1234567_DDDD PPPPPP1A2b3C4D5E6f
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"
TAG,{
,01/01/2000,01/01/2000,"CS","EXAMPLE","NNNNNNN"
},1
REQUESTOR,{
"PPPPPP","1234567","JOHN DOE",,,
},1
PROVIDER,{
"AAAA","AAAAPM",,,,"NONFIRM",
,"AAAA","AAAAPM","2-NH","AAAA-BBBB","111111",,,
"BBBB","BBBB","BBBPPM","2-NH","AAAA-CCCC","222222",,,
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,,
"DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,,
"DDDD","PPPPPP",,,,"DDDD-LD",
},6
ENERGY,{
06:00,22:00,100,,
},1
END
SUBMIT_END
```

Tag Authority:

SUCCESS

"AAAA_PPPPP1234567_DDDD","V1.6","NNN"

COMPOSITE,{

"PENDING",12/31/1999 17:00:00,01/01/2000 06:00,01/01/2000 22:00,"PSE","PPPPPP","JOHN DOE",

},1

STATUS,{

"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,, "NOTIFY"

"PSE","AAAAPM",,,,,,

"PSE","BBBBPM",,,,,,

"TP","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"TP","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"TP","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"TP","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"CA","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"CA","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"CA","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"CA","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"

"SC","DDDD",,,,,,

},12

SUCCESS_END

C.3 Tag Distribution

C.3.1. Via ASSESS Message

Tag Authority:

```

ASSESS AAAA AAAA_PPPPPP1234567_DDDD DDDD4d5E6f7G8h9I

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

TAG,{
,01/01/2000,01/01/2000,"CS","EXAMPLE",,"NNNNNNN"
},1

COMPOSITE,{
"PENDING",12/31/1999 17:00:00,01/01/2000 06:00,01/01/2000 22:00,"PSE","PPPPPP","JOHN DOE",
},1

STATUS,{
"PSE","PPPPPP",,"12/31/1999 17:00:00","JOHN DOE",,"NOTIFY"

"PSE","AAAAPM",,,,,,

"PSE","BBBBPM",,,,,,

"TP","AAAA","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"TP","BBBB","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"TP","CCCC","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"TP","DDDD","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"CA","AAAA","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"CA","BBBB","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"CA","CCCC","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"CA","DDDD","PENDING",12/31/1999 17:00:00,,"ASSESS","NOTIFY"

"SC","DDDD",,,,,,

},12

REQUESTOR,{

"PPPPPP","1234567",,"JOHN DOE",,

},1

PROVIDER,{

"AAAA",,"AAAAPM",,,,,,"NONFIRM",

```

```
,"AAAA","AAAAPM","2-NH","AAAA-BBBB","111111",,,  
"BBBB","BBBB","BBBBPM","2-NH","AAAA-CCCC","222222",,,  
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,,  
"DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,,  
"DDDD",,"PPPPPP",,,,,,"DDDD-LD",  
},6  
ENERGY,{  
06:00,22:00,100,,  
},1  
END  
ASSESS_END
```

Tag Approval:

```
SUCCESS  
"AAAA_PPPPP1234567_DDDD","V1.6","NNN"  
SUCCESS_END
```

C.3.2. Via INFO Message

Tag Authority:

```

INFO AAAAPM AAAA_PPPPPP1234567_DDDD DDDD9Q0r1S2i3U4V
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"
TAG,{
,01/01/2000,01/01/2000,"CS","EXAMPLE","NNNNNNN"
},1
COMPOSITE,{
"PENDING",12/31/1999 17:00:00,01/01/2000 06:00,01/01/2000 22:00,"PSE","PPPPPP","JOHN DOE",
},1
STATUS,{
"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,, "NOTIFY"
"PSE","AAAAPM",,,,,,
"PSE","BBBBPM",,,,,,
"TP","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"TP","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"TP","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"TP","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"CA","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"CA","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"CA","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"CA","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS","NOTIFY"
"SC","DDDD",,,,,,
},12
REQUESTOR,{
"PPPPPP","1234567",,"JOHN DOE",,,,
},1
PROVIDER,{
"AAAA",,"AAAAPM",,,,, "NONFIRM",
,"AAAA","AAAAPM","2-NH","AAAA-BBBB","111111",,,,

```

```
"BBBB","BBBB","BBBBPM","2-NH","AAAA-CCCC","222222",,,  
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,,  
,"DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,,  
"DDDD","PPPPPP",,,,,,"DDDD-LD",  
},6  
ENERGY,{  
06:00,22:00,100,,  
},1  
END  
ASSESS_END
```

Tag Agent:

```
SUCCESS  
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"  
SUCCESS_END
```

C.4 Tag Evaluation

Tag Approval:

UPDATE DDDD AAAA_PPPPPP1234567_DDDD DDDD4d5E6f7G8h9I

"APPROVED","JOE SMITH",

UPDATE_END

Tag Authority:

SUCCESS

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

COMPOSITE,{

"PENDING",12/31/1999 17:00:00,01/01/2000 06:00,01/01/2000 22:00,"PSE","PPPPPP","JOHN DOE",

},1

STATUS,{

"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,,,"NOTIFY"

"PSE","AAAAPM",,,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"PSE","BBBBPM",,,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,,"ASSESS","NOTIFY"

"TP","BBBB","QUEUED",12/31/1999 17:00:05,12/31/1999 17:00:05,,,,"ASSESS","NOTIFY"

"TP","CCCC","QUEUED",12/31/1999 17:00:06,12/31/1999 17:00:06,,,,"ASSESS","NOTIFY"

"TP","DDDD","QUEUED",12/31/1999 17:00:06,12/31/1999 17:00:06,,,,"ASSESS","NOTIFY"

"CA","AAAA","QUEUED",12/31/1999 17:00:07,12/31/1999 17:00:07,,,,"ASSESS","NOTIFY"

"CA","BBBB","QUEUED",12/31/1999 17:00:07,12/31/1999 17:00:07,,,,"ASSESS","NOTIFY"

"CA","CCCC","QUEUED",12/31/1999 17:00:07,12/31/1999 17:00:07,,,,"ASSESS","NOTIFY"

"CA","DDDD","QUEUED",12/31/1999 17:00:07,12/31/1999 17:00:07,,,,"ASSESS","NOTIFY"

"SC","DDDD",,,,,,

},12

SUCCESS_END

C.5 Tag Notification

C.5.1. Via NOTIFY Message

Tag Authority:

```

NOTIFY AAAA AAAA_PPPPP1234567_DDDD DDDD4d5E6f7G8h9I

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

COMPOSITE,{

"IMPLEMENT",12/31/1999 17:08:12,01/01/2000 06:00,01/01/2000 22:00,"CA","DDDD",,
},1

STATUS,{

"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,, "NOTIFY"

"PSE","AAAAPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"PSE","BBBBPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,,"ASSESS","NOTIFY"

"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,,"ASSESS","NOTIFY"

"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,,"ASSESS","NOTIFY"

"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,,"ASSESS","NOTIFY"

"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,,"ASSESS","NOTIFY"

"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,,"ASSESS","NOTIFY"

"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,,"ASSESS","NOTIFY"

"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,,"ASSESS","NOTIFY"

"SC","DDDD",,,,,,

},12

NOTIFY_END

```

Tag Approval (NOTE: Also may be used with Tag Agent):

```

SUCCESS

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

SUCCESS_END

```

C.6 Tag Forwarding

Tag Authority:

```

IMPLEMENT DDDD AAAA_PPPPP1234567_DDDD DDDD9Jop85r4ww2L

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

TAG,{

,01/01/2000,01/01/2000,"CS","EXAMPLE",,"NNNNNNN"

},1

COMPOSITE,{

"IMPLEMENT",12/31/1999 17:08:12,01/01/2000 06:00,01/01/2000 22:00,"CA","DDDD",,

},1

STATUS,{

"PSE","PPPPPP",,,12/31/1999 17:00:00,"JOHN DOE",,"NOTIFY"

"PSE","AAAAPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,

"PSE","BBBBPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,"ASSESS","NOTIFY"

"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,"ASSESS","NOTIFY"

"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,"ASSESS","NOTIFY"

"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,"ASSESS","NOTIFY"

"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,"ASSESS","NOTIFY"

"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,"ASSESS","NOTIFY"

"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,"ASSESS","NOTIFY"

"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,"ASSESS","NOTIFY"

"SC","DDDD","PENDING",12/31/1999 17:08:13,,,,

},12

REQUESTOR,{

"PPPPPP","1234567",,"JOHN DOE",,

},1

PROVIDER,{

"AAAA",,"AAAAPM",,,,,,"NONFIRM",

,"AAAA",,"AAAAPM",,"2-NH",,"AAAA-BBBB",,"111111",,,

"BBBB",,"BBBB",,"BBBBPM",,"2-NH",,"AAAA-CCCC",,"222222",,,

```

```
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,,  
, "DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,,  
"DDDD",,"PPPPPP",,,,,,"DDDD-LD",  
},6  
ENERGY,{  
06:00,22:00,100,,  
},1  
END  
IMPLEMENT_END
```

Interchange Distribution Calculator:

```
SUCCESS  
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"  
SUCCESS_END
```

C.7 Tag Querying

C.7.1. Via STATUS Message

Tag Agent Service:

STATUS DDDD AAAA_PPPPPP1234567_DDDD PPPPPP1A2b3C4D5E6f

STATUS_END

Tag Authority:

SUCCESS

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

COMPOSITE,{

"IMPLEMENT",12/31/1999 17:08:12,01/01/2000 06:00,01/01/2000 22:00,"CA","DDDD",,

},1

STATUS,{

"PSE","PPPPPP",,,12/31/1999 17:00:00,"JOHN DOE",,,,"NOTIFY"

"PSE","AAAAPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,

"PSE","BBBBPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,"ASSESS","NOTIFY"

"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,"ASSESS","NOTIFY"

"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,"ASSESS","NOTIFY"

"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,"ASSESS","NOTIFY"

"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,"ASSESS","NOTIFY"

"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,"ASSESS","NOTIFY"

"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,"ASSESS","NOTIFY"

"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,"ASSESS","NOTIFY"

"SC","DDDD","APPROVED",12/31/1999 17:08:14,,,,,

},12

SUCCESS_END

C.7.2. Via DSTATUS Message**Tag Agent Service:**

DSTATUS DDDD AAAA_PPPPPP1234567_DDDD PPPPPP1A2b3C4D5E6f

DSTATUS_END

Tag Authority:

SUCCESS

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

TAG,{

,01/01/2000,01/01/2000,"CS","EXAMPLE","NNNNNNN"

},1

COMPOSITE,{

"IMPLEMENT",12/31/1999 17:08:12,01/01/2000 06:00,01/01/2000 22:00,"CA","DDDD",,

},1

STATUS,{

"PSE","PPPPPP",,,12/31/1999 17:00:00,"JOHN DOE",,,,"NOTIFY"

"PSE","AAAAPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"PSE","BBBBPM",,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,"ASSESS","NOTIFY"

"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,"ASSESS","NOTIFY"

"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,"ASSESS","NOTIFY"

"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,"ASSESS","NOTIFY"

"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,"ASSESS","NOTIFY"

"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,"ASSESS","NOTIFY"

"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,"ASSESS","NOTIFY"

"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,"ASSESS","NOTIFY"

"SC","DDDD","APPROVED",12/31/1999 17:08:14,,,,,

},12

REQUESTOR,{

```
"PPPPPP","1234567","JOHN DOE",,  
  
},1  
  
PROVIDER,{  
  
"AAAA","AAAAAPM",,,,,,"NONFIRM",  
  
,"AAAA","AAAAAPM","2-NH","AAAA-BBBB","111111",,  
  
"BBBB","BBBB","BBBBPM","2-NH","AAAA-CCCC","222222",,  
  
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,  
  
,"DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,  
  
"DDDD","PPPPPP",,,,,,"DDDD-LD",  
  
},6  
  
ENERGY,{  
  
06:00,22:00,100,,  
  
},1  
  
END  
  
SUCCESS_END
```

C.8 Tag Cancellation

Tag Agent Service:

CANCEL DDDD AAAA_PPPPPP1234567_DDDD PPPPPP1A2b3C4D5E6f

01/01/2000 06:00,"JOHN DOE","Found new source"

CANCEL_END

Tag Authority:

SUCCESS

"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"

COMPOSITE,{

"CANCELLED",01/01/2000 04:22:56,01/01/2000 06:00,01/01/2000 06:00,"PSE","PPPPPP","JOHN DOE","Found new source"

},1

STATUS,{

"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,,,"NOTIFY"

"PSE","AAAAPM",,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"PSE","BBBBPM",,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,

"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,,"ASSESS","NOTIFY"

"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,,"ASSESS","NOTIFY"

"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,,"ASSESS","NOTIFY"

"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,,"ASSESS","NOTIFY"

"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,,"ASSESS","NOTIFY"

"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,,"ASSESS","NOTIFY"

"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,,"ASSESS","NOTIFY"

"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,,"ASSESS","NOTIFY"

"SC","DDDD","APPROVED",12/31/1999 17:08:14,,,,,

},12

SUCCESS_END

C.9 Tag Adjustment

Tag Approval:

```
ADJUST DDDD AAAA_PPPPPP1234567_DDDD DDDD4d5E6f7G8h9I
01/01/2000 06:00,01/01/2000 08:00,50,"JOE SMITH","TLR AGATE010100"
ADJUST_END
```

Tag Authority:

```
SUCCESS
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"
COMPOSITE,{
"ADJUSTED",12/31/1999 17:08:12,01/01/2000 06:00,01/01/2000 22:00,"CA","DDDD",,
},1
STATUS,{
"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,, "NOTIFY"
"PSE","AAAAPM",,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,
"PSE","BBBBPM",,,12/31/1999 17:00:05,12/31/1999 17:00:05,,,,
"TP","AAAA","APPROVED",12/31/1999 17:04:22,12/31/1999 17:00:05,"JOE SMITH",,,"ASSESS","NOTIFY"
"TP","BBBB","APPROVED",12/31/1999 17:05:15,12/31/1999 17:00:05,"Mark Roberts",,,"ASSESS","NOTIFY"
"TP","CCCC","APPROVED",12/31/1999 17:07:37,12/31/1999 17:00:06,"JIM HUGGINS",,,"ASSESS","NOTIFY"
"TP","DDDD","APPROVED",12/31/1999 17:07:16,12/31/1999 17:00:06,"BOB DAVIS",,,"ASSESS","NOTIFY"
"CA","AAAA","APPROVED",12/31/1999 17:08:11,12/31/1999 17:00:07,"Tim Heath",,,"ASSESS","NOTIFY"
"CA","BBBB","APPROVED",12/31/1999 17:07:27,12/31/1999 17:00:07,"LISA HALL",,,"ASSESS","NOTIFY"
"CA","CCCC","APPROVED",12/31/1999 17:08:12,12/31/1999 17:00:07,"JUAN TAYLOR",,,"ASSESS","NOTIFY"
"CA","DDDD","APPROVED",12/31/1999 17:07:42,12/31/1999 17:00:07,"Ann Williams",,,"ASSESS","NOTIFY"
"SC","DDDD","APPROVED",12/31/1999 17:08:14,,,,,
},12
ADJUST,{
01/01/2000 05:52:13,01/01/2000 06:00,01/01/2000 08:00,50,"CA","AAAA","JOE SMITH","TLR AGATE010100"
},1
SUCCESS_END
```

C.10 Tag Replacement

Tag Agent Service:

```
SUBMIT DDDD AAAA_PPPPPP1234567_DDDD PPPPPP1A2b3C4D5E6f{AAAA_PPPPPP9876543_DDDD,PPPPPP77hwUKw6y52p}
```

```
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"
```

```
TAG,{
```

```
,01/01/2000,01/01/2000,"CS","EXAMPLE","NNNNNNN"
```

```
},1
```

```
REQUESTOR,{
```

```
"PPPPPP","1234567","JOHN DOE",,
```

```
},1
```

```
PROVIDER,{
```

```
"AAAA","AAAAPM",,"NONFIRM",
```

```
,,"AAAA","AAAAPM","2-NH","AAAA-BBBB","111111",,
```

```
"BBBB","BBBB","BBBPM","2-NH","AAAA-CCCC","222222",,
```

```
"CCCC","CCCC","PPPPPP","2-NH","BBBB-DDDD","333333",,
```

```
,,"DDDD","PPPPPP","2-NH","CCCC-DDDD","444444",,
```

```
"DDDD","PPPPPP",,"DDDD-LD",
```

```
},6
```

```
ENERGY,{
```

```
06:00,22:00,100,,
```

```
},1
```

```
END
```

```
SUBMIT_END
```

Tag Authority:

```
SUCCESS
```

```
"AAAA_PPPPPP1234567_DDDD","V1.6","NNN"
```

```
COMPOSITE,{
```

```
"PENDING",12/31/1999 17:00:00,01/01/2000 06:00,01/01/2000 22:00,"PSE","PPPPPP","JOHN DOE",
```

```
},1
STATUS,{
"PSE","PPPPPP",,,,12/31/1999 17:00:00,"JOHN DOE",,,, "NOTIFY"
"PSE","AAAAPM",,,,,,
"PSE","BBBBPM",,,,,,
"TP","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"TP","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"TP","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"TP","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"CA","AAAA","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"CA","BBBB","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"CA","CCCC","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"CA","DDDD","PENDING",12/31/1999 17:00:00,,,, "ASSESS", "NOTIFY"
"SC","DDDD",,,,,,
},12
REPLACE,{
"AAAA_PPPPP9876543_DDDD"
},1
SUCCESS_END
```

Appendix D — Backup Fax Form

In the event of a system failure, affected parties may be called upon to fax transaction information to other affected parties. In this event, the attached form may be used to fax appropriate tag information to other parties. Other forms that convey the same information are also allowed. All fax communications should be noted in Central Standard Time.

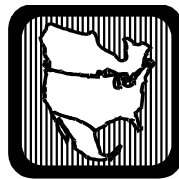
Appendix E — Electronic Tagging – Registry Definition Version 1.6

The E-Tag Registry Definition Version 1.6 will be incorporated in Version 1.6 of the E-Tag Functional definition for concurrent implementation.

Electronic Tagging – Registry Definition

Version 1.65

June 1, 2000



North American Electric Reliability Council

Table of Contents

1.	INTRODUCTION	E-ERROR! BOOKMARK NOT DEFINED.
2.	REGISTRY FILE UPDATES AND LOCATION	E-ERROR! BOOKMARK NOT DEFINED.
2.1.	E-TAG REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
2.2.	NERCTAG REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
3.	BASIC RULES.....	E-ERROR! BOOKMARK NOT DEFINED.
4.	CSV FILE FORMATS	E-ERROR! BOOKMARK NOT DEFINED.
5.	DATABASE DEFINITIONS	E-ERROR! BOOKMARK NOT DEFINED.
5.1.	ENTITY_REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
5.2.	CA_REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
5.3.	PSE_REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
5.4.	SC_REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
5.5.	TP_REGISTRY	E-ERROR! BOOKMARK NOT DEFINED.
5.6.	CHANGE_LOG	E-ERROR! BOOKMARK NOT DEFINED.
5.7.	PRODUCT_REGISTRY.....	E-ERROR! BOOKMARK NOT DEFINED.
5.8.	REGISTRY_VERSION.....	E-ERROR! BOOKMARK NOT DEFINED.

1. Introduction

This document codifies the format and content of the data files of the North American Electric Reliability Council (NERC) Electronic Tagging Registry. That Registry shares most data elements with used for the Open Access Same-time Information System (OASIS).

WARNING: Formats of the Registry data described in this document supercede those contained in the attribute specifications in the database (MDB file). Those attributes will not exactly match this document. No conclusions can be drawn from the internal MDB design specification settings.

2. Registry File Updates and Location

2.1. E-TAG Registry

Daily updates will be made to the Registry.

Registry information will be available for download in the following files:

File	Contents
Registry_Version.CSV	Registry version and effective date information in CSV format
CA_Registry.CSV	CA Registry information in CSV format
TP_Registry.CSV	TP Registry information in CSV format
PSE_Registry.CSV	PSE Registry information in CSV format
SC_Registry.CSV	SC Registry information in CSV format
Change_Log.CSV	Log of Registry changes
Product_Registry.CSV	Product Codes in CSV format
Entity_Registry.CSV	Registry of all Entities in CSV format
Full_Registry.MDB	Data from all of the above files in MDB format

The Registry updates are processed at 2345 CS on Sunday through Thursday nights to allow capture of the current Registry data and production of the new pending files for download. New Pending files are released at about 1200 CS on Monday through Friday to go Active 12 hours later. The 12-hour lag is to allow OASIS and tagging systems to be updated with the new Registry information.

The Registry is reopened at 2300 CS after the download files are updated. Completed processing and are made available.

The Active files, which were the pending files from the previous update, become available for download 15 minutes later at 0000 CS. These files are available at:

<ftp://ettagftp.nerc.com/active>

These files are made available to enable OASIS and tagging systems to determine if they are in synchronism with the current registry, or to reload current files when necessary.

The revised pending are not immediately available for download. They are held for review until about noon CS to allow a quality assurance check of the files. The new pending files become effective 24 hours later at 0000 CS. These files are available at:

<ftp://etagftp.nerc.com/pending>

Registry files are archived daily with the retired active files date-stamped for storage. The archive of Registry files is available at:

<ftp://etagftp.nerc.com/archive>

E-Tag Registry files will be date stamped using the Registry_Version table in the Full_Registry.MDB file, and the Registry_Version.CSV file.

2.2. NERCTag Registry

The TISLIST registry updates will cease with the implementation of Version 1.5 of E-Tag and the implementation of E-Tag by WSCC.

NERCTag Registry information are in the following files:

File	Contents
CA Sheet.CSV	Contains CA Registry information in CSV format
Product Sheet.CSV	Contains Product Codes in CSV format
PSE Sheet.CSV	Contains PSE Registry information in CSV format
TP Sheet.CSV	Contains TP Registry information in CSV format
TISLISTS_Version.CSV	Contains TISLISTS Registry version and effective date information in CSV format. This is a new table within the TISLISTS.MDB file as well.
TISLISTS.MDB	Contains data from all of the above files in MDB format

3. Basic Rules

The following rules are used throughout this specification to describe basic parsing constructs. The US-ASCII coded character set is defined by ANSI X3.4-1986.

OCTET	= <any 8-bit sequence of data>
CHAR	= <any US-ASCII character (octets 0 - 127)>
UPALPHA	= <any US-ASCII uppercase letter "A".."Z">
LOALPHA	= <any US-ASCII lowercase letter "a".."z">
ALPHA	= UPALPHA LOALPHA
DIGIT	= <any US-ASCII digit "0".."9">
ALPHANUM	= ALPHA DIGIT
UPALPHANUM	= UPALPHA DIGIT
CTL	= <any US-ASCII control character (octets 0 - 31) and
DEL (127)>	
TEXT	= <any CHAR except CTLs and ASCII double-quote
mark (34)>	
CR	= <US-ASCII CR, carriage return (13)>
LF	= <US-ASCII LF, linefeed (10)>
SP	= <US-ASCII SP, space (32)>
HT	= <US-ASCII HT, horizontal-tab (9)>
CRLF	= (CR LF)
LT	= CR LF CRLF (LF CR)
NULL	= <NULL>
<">	= <US-ASCII double-quote mark (34)>
<_>	= <US-ASCII underscore (95)>
<,>	= <US-ASCII comma (44)>
<{>	= <US-ASCII open brace (123)>
<}>	= <US-ASCII close brace (125)>
</>	= <US-ASCII forward slash (47)>
<:>	= <US-ASCII colon (58)>
<.>	= <US-ASCII period (46)>
<(>	= <US-ASCII open parentheses (40)>
<)>	= <US-ASCII close parentheses (41)>
<X>	= <US-ASCII uppercase X (88)>
<->	= <US-ASCII dash (45)>
<&>	= <US-ASCII ampersand (38)>

4. CSV File Formats

The data content, sequence of data and formats for the comma separated value (CSV) files are equal to those of the associated table in the database files.

Handling of Nulls

These files are generated directly from the Microsoft Access 97 database. Therefore, their content format for handling of nulls is dictated by Microsoft's data export rules from Access 97:

For numeric fields – Nulls appear in the CSV file as two adjacent commas (,,) when within the data string, and no entry appears at all if at the end of the data string.

For Text fields – Nulls appear in the CSV file as two adjacent double-quotation marks (“”) regardless of position in the data stream.

5. Database Definitions

The following tables contain the data definitions and formats for the E-Tag Registry files. The associated CSV files follow the exact same data content, sequence and format.

WARNING: Formats of the Registry data described in this document supercede those contained in the attribute specifications in the database (MDB file). Those attributes may not exactly match this document. No conclusions can be drawn from the internal MDB design specification settings.

5.1. Entity_Registry

Note: This table is very similar to the data provided in the REG.TXT CSV file or the REG table in the Access97.MDB file.

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
NERC_ID	<u>AUTO-GENERATED</u> - A unique number within the database, auto generated for the Entity Registration record	1{NUMERIC}255		Auto Number
Duns	<u>REQUIRED</u> - 9 digit unique number acquired from DUN & BRADSTREETS.	1{UPALPHANUM}9	<.>&<_> <,><->	No white space allowed
Entity_Code	<u>REQUIRED</u> - Business Entity's code	1{UPALPHANUM}4		No white space allowed
Entity_Type	<u>REQUIRED</u> - Entity Type. Stores only O/SE, TC/PSE, Observer, and Node.	1{ALPHA}8	"O/SE" "TC/PSE" "Observer" "Node"	Trailing white space allowed
Entity_Name	<u>REQUIRED</u> - Business Entity's name	1{ALPHANUM}255	<.>&<_> <,><->	White space allowed

Entity_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Address_Line_One	<u>REQUIRED</u> - Street Address	1{ALPHANUM}255	<.><&><_> <,><-></> <\>	White space allowed
Address_Line_Two	<u>OPTIONAL</u> - Street Address (cont.) or PO BOX	0{ALPHANUM}255	<.><&><_> <,><->	White space allowed
City	<u>REQUIRED</u> - City information.	1{ALPHA}255	<.><&><_> <,><->	No white space allowed
State	<u>REQUIRED</u> - State information.	1{ALPHA}255	<.><&><_> <,><->	No white space allowed
Zip_Code	<u>REQUIRED</u> - Alpha-numeric field used to store Postal or Zip Code	1{ALPHANUM}50	<->	Trailing White space allowed
Country	<u>REQUIRED</u> - Country information.	1{ALPHANUM}255	<.><&><_> <,><-><@>	No white space allowed between characters
Prim_Contact	<u>REQUIRED</u> - Primary contact name for Business Entity.	1{ALPHA}255	<.><&><_> <,><->	White space allowed

Entity_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Prim_Phone	<u>REQUIRED</u> - Primary contact phone number	*1CountryCode(>AreaCode<)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(><)><->	CountryCode and Extension are optional. A space must separate the "X" from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Prim_Fax	<u>REQUIRED</u> - Primary contact fax number	*1CountryCode(>AreaCode<)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT	<(><)><->	CountryCode and Extension are optional. A space must separate the "X" from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Prim_Email	<u>REQUIRED</u> - Primary Contact e-mail address	1{ALPHANUM}255	<.><&><_> <,><-><@>	No white space allowed between characters

Entity_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Admin_Contact	<u>REQUIRED</u> - Administrative contact name for Business Entity	1{ALPHANUM}255	<.><&><_> <,><->	White space allowed
Admin_Phone	<u>REQUIRED</u> - Administrative contact phone number	*1CountryCode<(>AreaCode< >)>Exchange<< >Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(>< >><- >	CountryCode and Extension are optional. A space must separate the "X" from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

Entity_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Admin_Email	<u>REQUIRED</u> - Administrative Contact e-mail address	1{ALPHANUM}255	<.>&><_> <,><-><@>	No white space allowed
Entity_URL	<u>REQUIRED</u> – The Date that this record becomes active.	1{ALPHANUM}255	<.>&><_> <,><-><:> <:></>	No white space allowed between characters
Begin_Date	<u>REQUIRED</u> – Effective begin date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
End_Date	<u>OPTIONAL</u> – Effective end date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
Initial_Date	<u>AUTO GENERATED</u> - A Date/time stamp the time this record was entered.	1{ALPHANUM}8	<:>	MM/DD/YYYY No white space allowed

5.2. CA_Registry

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Record_ID	<u>REQUIRED</u> – Unique record identifier	1{DIGIT}10		Value less than 2,147,483,647
Tag_Code	<u>REQUIRED</u> – Tagging Code	1{ UPALPHA}4		No white space allowed
Entity_Name	<u>REQUIRED</u> - Formal full name of the registered entity name	1{ALPHANUM}255	<.>&<_><,>><->	White space allowed
Contact24	<u>REQUIRED</u> – 24 hour contact person	1{ALPHANUM}255	<.>&<_><,>><->	White space allowed
Phone24	<u>REQUIRED</u> – 24 hour phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{E xtension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(>)><->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

CA_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
FAX	<u>REQUIRED</u> – 24 hour FAX number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(>)><- >	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Agent_URL	<u>OPTIONAL</u> – URL for Agent Service	0{TEXT}255		No white space allowed
Authority_URL	<u>REQUIRED</u> – URL for Authority Service	1{TEXT}255		No white space allowed
Approval_URL	<u>REQUIRED</u> – URL for Approval Service	1{TEXT}255		No white space allowed
Forward_URL	<u>OPTIONAL</u> – URL for forward on IMPLEMENT State Change	0{TEXT}255		No white space allowed
Entity_Code	<u>REQUIRED</u> – Parent Entity Code	1{UPALPHANUM}4		No white space allowed

CA_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Tag_Code_Type	<u>AUTO GENERATED</u> – Type of this Tag Code: PSE, CA, TP, or SC	2{UPALPHA}3	“CA”	No white space allowed
NERC_ID	<u>REQUIRED</u> - NERC Master Entity ID Number	1{DIGIT}7		Auto Generated
Begin_Date	<u>REQUIRED</u> – Effective begin date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
End_Date	<u>OPTIONAL</u> – Effective end date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
SC_Code	<u>REQUIRED</u> – Associated Security Coordinator Code	1{UPALPHANUM}4		No white space allowed
Region	<u>REQUIRED</u> – Associated NERC Region	1{TEXT}20	“ECAR”, “ERCOT”, “FRCC”, “MAAC”, “MAIN”, “MAPP”, “NPCC”, “SERC”, “SPP”, “WSCC-AZNMSNV”, “WSCC-CAMX”, “WSCC-NWPP”, “WSCC-RMPA”	No white space allowed
0-NHMBAM_Flag	<u>INVALID</u> – NextHour Market participation flag	Null	Null	Not valid for CAs

ALT1Desc	<u>OPTIONAL</u> – First Alternate Contact Description	1{TEXT}50	<.>&<_><,><->	White space allowed
ALT1Phone	<u>OPTIONAL</u> – First Alternate Phone number	<p>*1CountryCode(<AreaCode <)>Exchange<>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>A. Example: 123(123)456-7890 X 1234567890</p>	<(><)><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>
ALT2Desc	<u>OPTIONAL</u> – Second Alternate Contact Description	1{TEXT}50	<.>&<_><,><->	White space allowed
ALT2Phone	<u>OPTIONAL</u> – Second Alternate phone number	<p>*1CountryCode(<AreaCode <)>Exchange<>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<(><)><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>

ALT3Desc	<u>OPTIONAL</u> – Third Alternate Contact Description	1{TEXT}50	<.>&><_><,><->	White space allowed
ALT3Phone	<u>OPTIONAL</u> – Third Alternate phone number	<p>*1CountryCode<(>AreaCode<(>Exchange<(>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<(><)><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>

5.3. PSE_Registry

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Record_ID	<u>REQUIRED</u> – Unique record identifier	1{DIGIT}10		Value less than 2,147,483,647
Tag_Code	<u>REQUIRED</u> – Tagging Code	1{ UPALPHANUM }6		No white space allowed
Entity_Name	<u>REQUIRED</u> - Formal full name of the registered entity name	1{ALPHANUM}255	<.>&<_><,><->	White space allowed
Contact24	<u>REQUIRED</u> – 24 hour contact person	1{ALPHANUM}255	<.>&<_><,><->	No white space allowed
Phone24	<u>REQUIRED</u> – 24 hour phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{E xtension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(>)><->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

PSE_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
FAX	<u>REQUIRED</u> – 24 hour FAX number	*1CountryCode<(>AreaCode<(>Exchange<(>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT	<(>)<(<->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Agent_URL	<u>OPTIONAL</u> – URL for Agent Service	0{TEXT}255		No white space allowed
Authority_URL	<u>INVALID</u> – URL for Authority Service	Null	Null	Not valid for PSE
Approval_URL	<u>INVALID</u> – URL for Approval Service	Null	Null	Not valid for PSE
Forward_URL	<u>INVALID</u> – URL for forward on IMPLEMENT State Change	Null	Null	Not valid for PSE
Entity_Code	<u>REQUIRED</u> – Parent Entity Code	1{UPALPHANUM}4		No white space allowed
Tag_Code_Type	<u>AUTO GENERATED</u> – Type of this Tag Code: PSE, CA, TP, or SC	2{UPALPHA}3	“CA” or “PSE”	No white space allowed

PSE_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
NERC_ID	<u>REQUIRED</u> - NERC Master Entity ID Number	1{DIGIT}7		Auto Generated
Begin_Date	<u>REQUIRED</u> – Effective begin date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
End_Date	<u>OPTIONAL</u> – Effective end date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
SC_Code	<u>INVALID</u> – Associated Security Coordinator Code	Null	Null	Not valid for PSEs
Region	<u>INVALID</u> – Associated_NERC Region	Null	Null	Not valid for PSEs
0-NHMBAM_Flag	<u>INVALID</u> – NextHour Market participation flag	Null	Null	Not valid for PSEs
ALT1Desc	<u>OPTIONAL</u> – First Alternate Contact Description	1{TEXT}50	<.><&><_><,><->	White space allowed

ALT1Phone	<u>OPTIONAL</u> – First Alternate Phone number	<p>*1CountryCode<(>AreaCode<(>Exchange<(>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>B. Example: 123(123)456-7890 X 1234567890</p>	<(><)><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>
ALT2Desc	<u>OPTIONAL</u> – Second Alternate Contact Description	1{TEXT}50	<.><&><_><,>><->	White space allowed
ALT2Phone	<u>OPTIONAL</u> – Second Alternate phone number	<p>*1CountryCode<(>AreaCode<(>Exchange<(>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<(><)><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>
ALT3Desc	<u>OPTIONAL</u> – Third Alternate Contact Description	1{TEXT}50	<.><&><_><,>><->	White space allowed

<p>ALT3Phone</p>	<p><u>OPTIONAL</u> – Third Alternate phone number</p>	<p>*1CountryCode(<AreaCode <)>Exchange<>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<p><(>) > < - ></p>	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>
------------------	---	---	--------------------------------------	---

5.4. SC_Registry

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Record_ID	<u>REQUIRED</u> – Unique record identifier	1{DIGIT}10		Value less than 2,147,483,647
Tag_Code	<u>REQUIRED</u> – Tagging Code	1{ UPALPHA }4		No white space allowed
Entity_Name	<u>REQUIRED</u> - Formal full name of the registered entity name	1{ALPHANUM}255	<.><&><_><,>><->	White space allowed
Contact24	<u>REQUIRED</u> – 24 hour contact person	1{ALPHANUM}255	<.><&><_><,>><->	No white space allowed
Phone24	<u>REQUIRED</u> – 24 hour phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(> <) > < - >	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

SC_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
FAX	<u>REQUIRED</u> – 24 hour FAX number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT	<(>)><->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Agent_URL	<u>INVALID</u> – URL for Agent Service	Null	Null	Not valid for SC
Authority_URL	<u>OPTIONAL</u> – URL for Authority Service	0{TEXT}255		Optional for SC to receive tags as intermediary and forwarding URL to IDC No white space allowed
Approval_URL	<u>INVALID</u> – URL for Approval Service	Null	Null	Not valid for SC
Forward_URL	<u>OPTIONAL</u> – URL for forward on IMPLEMENT State Change	0{TEXT}255		Optional forwarding URL to IDC No white space allowed
Entity_Code	<u>REQUIRED</u> – Parent Entity Code	1{UPALPHANUM}4		No white space allowed

SC_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Tag_Code_Type	<u>AUTO GENERATED</u> – Type of this Tag Code: PSE, CA, TP, or SC	2{UPALPHA}3	“SC”	No white space allowed
NERC_ID	<u>REQUIRED</u> – NERC Master Entity ID Number	1{DIGIT}7		Auto Generated
Begin_Date	<u>REQUIRED</u> – Effective begin date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
End_Date	<u>OPTIONAL</u> – Effective end date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
SC_Code	<u>INVALID</u> – Associated Security Coordinator Code	Null	Null	Not valid for SC
Region	<u>REQUIRED</u> – Associated NERC Region	1{TEXT}20	“ECAR”, “ERCOT”, “FRCC”, “MAAC”, “MAIN”, “MAPP”, “NPCC”, “SERC”, “SPP”, “WSCC- AZNMSNV”, “WSCC-CAMX”, “WSCC-NWPP”, “WSCC-RMPA”	No white space allowed
0-NHMBAM_Flag	<u>INVALID</u> – NextHour Market participation flag	Null	Null	Not valid for SCs

ALT1Desc	<u>OPTIONAL</u> – First Alternate Contact Description	1{TEXT}50	<.>&<_><,><->	White space allowed
ALT1Phone	<u>OPTIONAL</u> – First Alternate Phone number	<p>*1CountryCode(<AreaCode<)>Exchange<>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>C. Example: 123(123)456-7890 X 1234567890</p>	<(< >) < < - >	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>
ALT2Desc	<u>OPTIONAL</u> – Second Alternate Contact Description	1{TEXT}50	<.>&<_><,><->	White space allowed
ALT2Phone	<u>OPTIONAL</u> – Second Alternate phone number	<p>*1CountryCode(<AreaCode<)>Exchange<>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<(< >) < < - >	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>

ALT3Desc	<u>OPTIONAL</u> – Third Alternate Contact Description	1{TEXT}50	<.><&><_><,><->	White space allowed
ALT3Phone	<u>OPTIONAL</u> – Third Alternate phone number	<p>*1CountryCode<()AreaCode<()>Exchange<()>Number*1{Extension}10</p> <p>Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT</p> <p>Example: 123(123)456-7890 X 1234567890</p>	<()><->	<p>CountryCode and Extension are optional.</p> <p>A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.</p>

5.5. TP_Registry

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Record_ID	<u>REQUIRED</u> – Unique record identifier	1{DIGIT}10		Value less than 2,147,483,647
Tag_Code	<u>REQUIRED</u> – Tagging Code	1{ UPALPHA }4		No white space allowed
Entity_Name	<u>REQUIRED</u> – Formal full name of the registered entity name	1{ALPHANUM}255	<.>&<_><,><->	White space allowed
Contact24	<u>REQUIRED</u> – 24 hour contact person	1{ALPHANUM}255	<.>&<_><,><->	No white space allowed
Phone24	<u>REQUIRED</u> – 24 hour phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{E xtension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(>)><->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

TP_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
FAX	<u>REQUIRED</u> – 24 hour FAX number	*1CountryCode<()AreaCode<()>Exchange<()>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT	<()><()>	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
Agent_URL	<u>INVALID</u> – URL for Agent Service	Null	Null	Not valid for TP
Authority_URL	<u>INVALID</u> – URL for Agent Service	Null	Null	Not valid for TP
Approval_URL	<u>REQUIRED</u> – URL for Approval Service	1{TEXT}255		No white space allowed
Forward_URL	<u>INVALID</u> – URL for forward on IMPLEMENT State Change	Null	Null	Not valid for TP
Entity_Code	<u>REQUIRED</u> – Parent Entity Code	1{UPALPHANUM}4		No white space allowed
Tag_Code_Type	<u>AUTO GENERATED</u> – Type of this Tag Code: PSE, CA, TP, or SC	2{UPALPHA}3	“TP”	No white space allowed

TP_Registry (continued)

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
NERC_ID	<u>REQUIRED</u> - NERC Master Entity ID Number	1{DIGIT}7		Auto Generated
Begin_Date	<u>REQUIRED</u> – Effective begin date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
End_Date	<u>OPTIONAL</u> – Effective end date for this entry	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
SC_Code	<u>REQUIRED</u> – Associated Security Coordinator Code	1{ UPALPHA }4		No white space allowed
Region	<u>REQUIRED</u> – Associated NERC Region	1{TEXT}20	“ECAR”, “ERCOT”, “FRCC”, “MAAC”, “MAIN”, “MAPP”, “NPCC”, “SERC”, “SPP”, “WSCC- AZNMSNV”, “WSCC-CAMX”, “WSCC-NWPP”, “WSCC-RMPA”	No white space allowed
0-NHMBAM_Flag	<u>REQUIRED</u> – NextHour Market BUYATMARKET participation flag	1UPALPHA	“Y” “N”	No white space allowed

ALT1Desc	<u>OPTIONAL</u> – First Alternate Contact Description	1{TEXT}50	<.><&><_><,><->	White space allowed
ALT1Phone	<u>OPTIONAL</u> – First Alternate Phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT D. Example: 123(123)456-7890 X 1234567890	<(> <) > < - >	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.
ALT2Desc	<u>OPTIONAL</u> – Second Alternate Contact Description	1{TEXT}50	<.><&><_><,><->	White space allowed
ALT2Phone	<u>OPTIONAL</u> – Second Alternate phone number	*1CountryCode<(>AreaCode <)>Exchange<>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(> <) > < - >	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

ALT3Desc	<u>OPTIONAL</u> – Third Alternate Contact Description	1{TEXT}50	<.>&<_><,><->	White space allowed
ALT3Phone	<u>OPTIONAL</u> – Third Alternate phone number	*1CountryCode<(>AreaCode<(>Exchange<(>Number*1{Extension}10 Where: CountryCode = 3DIGIT AreaCode = 3DIGIT Exchange = 3DIGIT Number = 4DIGIT Extension = <SP><X><SP> 1*10DIGIT Example: 123(123)456-7890 X 1234567890	<(><)><->	CountryCode and Extension are optional. A space must separate the “X” from the phone number and the extension. No other spaces are allowed. See Appendix A of the Functional Specification for additional detail.

5.6. CHANGE_LOG

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
Record_ID	<u>AUTO-GENERATED</u> - Record identifier	1{NUMERIC}255		Auto Number
NERC_ID	<u>REQUIRED</u> – NERC Master Entity ID Number	1{DIGIT}7		Auto Number
Table_Name	<u>REQUIRED</u> – Name of Table in which a change has occurred	1{ALPHANUM}255	<_>	No white space allowed

Field_Changed	<u>REQUIRED</u> – Name of Field in which a change has occurred	1{ALPHANUM}255	<_>	No white space allowed
Old_Information	<u>REQUIRED</u> – Old data value	1{TEXT}255		White space allowed
New_Information	<u>REQUIRED</u> – New data value	1{TEXT}255		White space allowed
Effective_Date	<u>AUTO GENERATED</u> – Date change became effective	2DIGIT</>2DIGIT</>4DIGIT	</>	MM/DD/YYYY No white space allowed
Change_Request_DateTime	<u>AUTO GENERATED</u> – Date and time change was requested	2DIGIT</>2DIGIT </> 4DIGIT<SP>2DIGIT<:>2DIGIT<:>2DIGIT	</><:>	MM/DD/YYYY HH:MM:SS Single space required between Date and Time

5.7. Product_Registry

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
ProductSeq	<u>AUTO-GENERATED</u> - Record identifier	1{DIGIT}7		Auto Number
Code	<u>REQUIRED</u> – Product Code	1{UPERALPHANUM}4	<->	No white space allowed
Product	<u>REQUIRED</u> – Product description	1{ALPHANUM}255	<.>&><_><,> <-><@></>	White space allowed

5.8. Registry_Version

Attribute ID	Description	Attribute Format	Value Restriction (Allowed)	Comments
intVersionId	<u>AUTO-GENERATED</u> - Unique version identifier	1{NUMERIC}255		Auto Number
strVersion	<u>AUTO GENERATED</u> – Version of Master Registry	1{DIGIT}2 <.> 1{DIGIT}2 <.> 4DIGIT		Date driven
dtVersionDate	<u>AUTO GENERATED</u> – Date this version was issued	2DIGIT</>2DIGIT </> 4DIGIT	</>	MM/DD/YYYY No white space allowed
dtEffectiveDate	<u>AUTO GENERATED</u> – Date this version becomes effective	2DIGIT</>2DIGIT </> 4DIGIT	</>	MM/DD/YYYY No white space allowed

APPENDIX F — E-Tag Requirements Necessary for Implementation of NHM Service

NERC and those supporting the development of NHM Service are in the process of implementing the revisions to electronic tagging needed for NHM Service, as follows:

- I. NERC shall define the following items:

0-NextHour (0-NX): Transmission product representing hourly, next-hour, single hour transmission service. Shall have a curtailment level index of zero (0).

BUYATMARKET: A key phrase associated with a request to purchase a transmission product at the posted offer rate on a provider's OASIS.
- II. Transmission providers shall register with NERC their offering of the 0-NextHour Service to the marketplace. NERC shall maintain this information in the NERC Master Registry.
- III. E-Tag Authority systems shall reject as invalid any Tag from a Tag Author that requests 0-NextHour Service product from a provider that has not registered to offer 0-NextHour Service.
- IV. Valid requests shall be distributed to all appropriate Control Areas and Transmission Providers as currently described in the E-Tag Functional Specification.
- V. Transmission Providers, when presented with a request for a 0-NextHour product with a listed OASIS Reference of BUYATMARKET shall process the request in the following manner:
 - a. Evaluate the submitted tag, and indicate acceptance or refusal of the proposed transaction through the normal E-tag process.
 - b. Within one hour of the requested start of the transaction assign an OASIS reservation on the Transmission Provider's OASIS node on the Tag Author's behalf. The CAPACITY shall be determined by the Transmission Provider, but should be determined based on one or more of the following:
 - i. The amount of the Transmission Provider Product, if specified.
 - ii. In accordance with the Transmission Provider's tariff, the MW amount at the POR or POD for that Provider in the Loss Table, if Transmission Provider Product is not specified.
 - iii. The MW amount in the Energy Profile, if neither Transmission Provider Product amount nor Provider Loss Table amounts are specified.
 - c. Assign the OASIS reservation a final status in accordance with the final disposition of the ETAG as follows:
 - i. CONFIRMED if the Tag's Composite Status reaches IMPLEMENT or CONDITIONAL.

- ii. REFUSED if the Provider's Approval Status is REFUSED.
 - iii. ANNULLED if the Provider's Approval Status is ACCEPTED but the Tag's Composite Status is DENIED or WITHDRAWN.
- VI. Tags in which the Approval Time expires prior to the Transmission Provider setting their approval to ACCEPTED for a 0-NextHour product requested via BUYATMARKET shall automatically have the corresponding Approval State(s) associated with the incomplete 0-NX request(s) set to DENIED. The reason(s) listed shall be “OASIS Reservation Not Accepted.” These tags shall then have their status evaluated per the E-Tag Functional Specification, which will result in a Composite State of DENIED.

For more information, please see “Business Practice Standards for OASIS Transactions.”