RECOMMENDATION TO NAESB EXECUTIVE COMMITTEE
For Quadrant: Retail Electric, Retail Gas, Wholesale Electric
As Revised by the Retail ECs on 5/5/04

Requesters: REQ & RGQ TEIS & WGQ EDM
Request No.: WGQ Annual Plan Item 7, RGQ Annual Plan Item 12, REQ Annual Plan Item 14
Request Title: Internet Electronic Transport Version 2.0

1. RECOMMENDED ACTION:

_X_Accept as requested
___Accept as modified below
___Decline

EFFECT OF EC VOTE TO ACCEPT RECOMMENDED ACTION:

_X_Change to Existing Practice
___Status Quo

2. TYPE OF DEVELOPMENT/MAINTENANCE

Per Request:

_X_Initiation
___Modification
___Interpretation
___Withdrawal

___Principle
___Definition
_X_Business Practice Standard
___Document
___Data Element
___Code Value
___X12 Implementation Guide
___Business Process Documentation

Per Recommendation:

_X_Initiation
___Modification
___Interpretation
___Withdrawal

___Principle
___Definition
_X_Business Practice Standard
___Document
___Data Element
___Code Value
___X12 Implementation Guide
___Business Process Documentation

3. RECOMMENDATION

SUMMARY: Accept as recommended the following high-level guide to implementing various technologies relating to Internet Electronic Transport (IET) for the Retail Gas, Retail Electric and Wholesale Gas industries.

RECOMMENDED STANDARDS: Please see attached redline**.**

4. SUPPORTING DOCUMENTATION

a. Description of Request:

2004 WGQ Annual Plan Item 7 – Prepare a common NAESB Electronic Transport (ET) and WGQ Quadrant Electronic Delivery Mechanism (WGQ QEDM) manuals using version 1.6 NAESB WGQ EDM tab 6 and applicable 4.x.x standards as a base.

2004 RGQ Annual Plan Item 12 – Work jointly with the WGQ EDM subcommittee and the REQ TEIS subcommittee to establish standards for the NAESB Internet Electronic Transport.

2004 REQ Annual Plan Item 14 – Work jointly with the WGQ EDM subcommittee and the RGQ TEIS subcommittee to establish standards for the NAESB Internet Electronic Transport.

b. Description of Recommendation:

The proposed standards are the result of a series of meetings and conference calls held by the Retail Electric Quadrant Technical Electronic Implementation Subcommittee (TEIS), Retail Gas Quadrant TEIS, and Wholesale Gas Quadrant Electronic Delivery Mechanism (EDM) Subcommittee begun in March of 2002 and culminating with a vote to recommend the proposed standards to the Executive Committee during a conference call on March 16, 2004.

See the TEIS and EDM meeting minutes, attachments, and transcripts for the supporting documentation, discussion, and voting records for the following dates:

March 5, 2003
April 16, 2003
June 9, 2003
September 8, 2003
October 14-15, 2003
November 17-18, 2003
December 15, 2003
January 21-22, 2004
March 2-3, 2004
March 16, 2004

c. Business Purpose:

The business purposes for the recommended standards are as follows:

Energy companies need to exchange information and data with other energy companies. Internet ET enables this with the following advantages:

Security. Internet ET incorporates the PAIN security principles of Privacy, Authenticity, Integrity and Non-repudiation.

Standardized Process. Internet ET standardizes how packages are exchange, regardless of the business process, the trading partner, or the energy quadrant.

Audit Trail. Internet ET gives both Sender and Receiver a detailed audit trail, enabling better controls and less errors.

Error Notification. Internet ET prescribes how errors are to be handled, and provides a foundation for efficient and quick resolution to errors.

Minimum technology requirements. Internet ET is built on low-cost technology and readily-available Web browser and open source technology.

Interactive and Batch Capabilities. Internet ET provides mechanisms for both fully-automated and manual-assisted business processes.

Any Payloads. Internet ET can deliver any kind of payload, whether it is EDI, flat-files, XML, documents, etc.

Software Standards. The Internet ET standards increase the likelihood that software vendors will provide Commercial Off-The-Shelf (COTS) software packages.

d. Commentary/Rationale of Subcommittee(s)/Task Force(s):

The proposed standards were developed in a consensus-oriented process with active participation from all REQ, RGQ and WGQ Segments. That a degree of consensus was reached is evidenced by the unanimous passage of a motion during the March 16, 2004 conference call to recommend the proposed standards under consideration to the Executive Committee.

1 – EXECUTIVE SUMMARY

The North American Energy Standards Board (NAESB) Wholesale Gas Quadrant (WGQ), Retail Electric Quadrant (REQ), and Retail Gas Quadrant (RGQ) have developed standards for electronic commerce over the Internet. The Internet Electronic Transport (Internet ET) standards enable the rapid, reliable, and safe transportation of electronic information between NAESB trading partners.

This document is a high-level guide to implementing various technologies necessary to communicate transactions and other electronic data using standard protocols. As such, this guide is not intended to be a comprehensive, in-depth manual. Where possible, this guide points to more in-depth material. The Reference section provides locations on the Internet to obtain more information as well as recommended books and periodicals.

Parties should refer to market Governing Documents for specific implementations of Internet ET.

BUSINESS REASONS FOR USING INTERNET ET

Energy companies need to exchange information and data with other energy companies. Internet ET enables this with the following advantages:

Security. Internet ET incorporates the PAIN security principles of Privacy, Authenticity, Integrity and Non-repudiation.

Standardized Process. Internet ET standardizes how packages are exchange, regardless of the business process, the trading partner, or the energy quadrant.

Audit Trail. Internet ET gives both Sender and Receiver a detailed audit trail, enabling better controls and less errors.

Error Notification. Internet ET prescribes how errors are to be handled, and provides a foundation for efficient and quick resolution to errors.

Minimum technology requirements. Internet ET is built on low-cost technology and readily-available Web browser and open source technology.

Interactive and Batch Capabilities. Internet ET provides mechanisms for both fully-automated and manual-assisted business processes.

Any Payloads. Internet ET can deliver any kind of payload, whether it is EDI, flat-files, XML, documents, etc.

Software Standards. The Internet ET standards increase the likelihood that software vendors will provide Commercial Off-The-Shelf (COTS) software packages.

OVERVIEW OF ELECTRONIC TRANSPORT LIFE CYCLE

In the Internet ET life-cycle, the party sending data, the ‘Sender’, creates an electronic package by encrypting the data payload and applying appropriate header ‘envelope’ information such as 'to' and 'from'. This electronic package is submitted to the trading partner's SSL Web server as an HTTP Request using the POST method.

The receiving party, the 'Receiver', receives and decrypts the package, then forwards the payload data to back-office processes. A Receipt is sent from the Receiver to the Sender with timestamps and any error notices. The Receiver back-office systems process the data according to NAESB quadrant-specific Electronic Delivery Mechanisms (QEDM), quadrant-specific standards (e.g. 'Nominations'), Trading Partner Agreements, and related documents. If the Receiver decrypts in a separate process, the Receiver may send an Error Notification package to the Sender to identify errors found during decryption.

Trading partners can be either the Sender or Receiver depending on what information and data needs to be exchanged.

The Internet ET standards focus on the transport of the electronic package and not the contents of the package. Each business process may define different contents, and the Internet ET is designed to work with any type of contents (e.g. EDI, flat files, etc).

The following are Internet ET life-cycle scenarios:

1. **Success**. The Successful scenario is when the electronic package was delivered with no errors, and the Sender has received a Receipt from the Receiver.
2. **Invalid Package Response**. The Invalid Package Response scenario is when the Receiver was unable to disassemble the electronic package, and has sent an HTTP Response to the Sender notifying them of package errors.
3. **Invalid Package Error Notification**. The Invalid Package Error Notification scenario is when a Receiver detects an error in the package AFTER the Response is sent. This scenario exists when a Receiver has implemented processes where the decryption occurs after the Response is sent. Decryption errors are communicated to the Sender via an HTTP Request using the Internet ET Error Notification format.
4. **Exchange Failure**. The Exchange Failure scenario is when a Sender is unable to establish and/or maintain a connection with the Server to send an electronic package to the Receiver.

Errors detected after successful decryption (e.g. format errors, EDI errors, etc) are beyond the scope of the Internet ET, and can be found in the QEDM standards.

Parties implementing Internet ET should become familiar with the following components of the Internet ET:

- Internet ET Network and Communications Requirements
- Sending Internet ET Electronic Packages
- Receiving Internet ET Electronic Packages
- Security

KEY ASSUMPTIONS

This document makes the following assumptions:

- **Platform Independence.** An Internet ET implementation can communicate with all trading partners in the energy industry, regardless what hardware, operating system and programming languages trading partners use.
- **Open Standards.** NAESB has adopted open standard technologies to provide flexibility and scalability.
- **Payload Content Independence.** Internet ET standards focus on the transport of the electronic package, and not the contents of the package. Each business process may define different contents. Internet ET is designed to work with any type of content (e.g. EDI, flat files, etc). The Internet ET's main function is to get the package from point X to point Y reliably with privacy, authentication, integrity, and non-repudiation.
- **Importance of the Technical Exchange Worksheet (TEW)**. Internet ET relies on the exchange of technical information between trading partners to establish and maintain reliable Internet ET production. This worksheet is intended to establish communications between two parties. Additional requirements and information may be required. Refer to your quadrant-specific EDM (QEDM). A sample TEW is included in Appendix C. The TEW may be a part of a Trading Partner Agreement (TPA).
- **Testing With Internet ET Trading Partners.** Since the Internet ET is not platform-specific, testing with other trading partners on a variety of platforms is very important in ensuring that each Internet ET application is compatible with a range of platforms used by various trading partners. Testing should ensure receipt of the package, proper decryption, and appropriate Receipts were sent.
- **Business Process Considerations.** Implementers of business processes that use Internet ET should be aware of the following issues that may impact business process design:
    - o The Internet Lacks Quality of Service (QoS). The Internet is unable to assign priority to file transfers. High-priority NAESB Internet ET package transfers such as Nominations have no priority over low-priority Internet transfers such as music MP3 files or other lower-priority NAESB Internet ET transfers. Business processes that have firm or tight Internet ET transfer timing requirements should be constructed to properly mitigate the risk associated with this lack of guaranteed QoS on the Internet. QoS may be improved by using a private network in lieu of the Internet.
    - o Clock Synchronization. The Internet ET allows +/- 5 seconds variance from an NIST atomic clock. Business processes with more stringent requirements may need to implement more restrictive synchronization requirements and processes.
    - o Exchange Failures. When trading partners systems are failing, parties are required to attempt to send Internet ET packages 3 times over a minimum period of 30-minutes before notifying trading partners of exchange failures. Business processes with more stringent requirements may need to implement more restrictive exchange failure requirements and processes.
- **Examples Provided in this Document.** The examples provided in this document are for illustration only. Implementers should rely on the standards and not on these examples when implementing the Internet ET.

2 – VERSION HISTORY

3 – INTRODUCTION

The North American Energy Standards Board (NAESB) is a voluntary non-profit organization comprised of members from all aspects of the greater gas and electric industries. ~~The NAESB mission is to take the lead in developing and implementing standards across the industry to simplify and expand electronic communication, and to streamline business practices. The vision of NAESB is a seamless North American marketplace for energy, as recognized by its customers, the business community, industry participants and regulatory bodies.~~

NAESB Internet Electronic Transport (Internet ET) Standards are used by the Wholesale Gas Quadrant (WGQ), Retail Electric Quadrant (REQ), and the Retail Gas Quadrant (RGQ) for the electronic transport of transactions and other information payloads between trading partners.

NAESB recognizes that as the energy industry evolves and uses NAESB standards, additional and amended NAESB standards will be necessary. Any industry participant seeking additional or amended standards (including principles, definitions, standards, data elements, process descriptions, technical implementation instructions) should submit a request detailing the change to the NAESB office so that the appropriate process may take place to amend the standards.

TAB 1 Executive Summary

Provides a brief outline of the industry business situation which is the basis for development of this guide.

TAB 2 Version Notes

Contains notes about this version, and, if appropriate, a brief summary of changes from the immediately preceding version.

TAB 3 Introduction

Provides a background statement about NAESB's Mission and the underlying concepts behind the design and use of this guide.

TAB 4 Business Process & Practices

Provides a brief overview of the business process and the NAESB-approved principles, definitions and standards related to the business process covered by this guide.

TAB 5 Related Standards

Provides a reference to any related standards.

TAB 6 Technical Implementation – Internet Electronic Transport (Internet ET)

Provides an overview of the business process for Internet ET.

Data Dictionary

Provides definition of the standard data elements and the usage requirements for each element.

Batch Flow Diagram

Sending Electronic Packages

Provides instructions to develop mechanisms for sending of NAESB standard format data files.

Receiving Electronic Packages

Provides instructions to develop mechanisms for receiving of NAESB standard format data files.

Security

Provides guidelines for data privacy, data integrity, authentication and non-repudiation of inbound and outbound packages.

Other Considerations

Provides information regarding error notification and testing. Includes a reference guide and examples for repudiation and validation.

TAB 7 Testing Guidelines

Provides guidelines for testing the Internet ET standards.

TAB 8 Appendices

Table 1 – Internet ET Error Codes

Appendix A – Reference Guide

Appendix B – Internet ET FAQ

Appendix C – Sample Technical Exchange Worksheet (TEW)

Appendix D – Cross Reference Between Internet ET and WGQ EDM Version 1.7 Standards

4 – BUSINESS PROCESS AND PRACTICES

A. OVERVIEW

Role of Internet Electronic Transport (ET) in NAESB WGQ, REQ, and RGQ Quadrants

Business processes defined by NAESB Quadrants require the exchange of transactions and transaction data. The Internet ET, in concert with Quadrant-specific Electronic Delivery Mechanisms (QEDMs), enables NAESB parties to securely and reliably exchange transactions over the Internet. Internet ET electronic 'packages' are created using the standards defined in this document.

Version 2.0 of the Internet ET standard incorporates all electronic transport technical specifications of the NAESB WGQ EDM Version 1.7.

Roles in Internet ET

In the Internet ET life-cycle, one party sends a package, and the other party receives the package. The party sending the package is referred to as the Sender or Client, and the party receiving the package is also referred to as the Receiver or Server.

NAESB business processes often require that parties act in both the Sender and Receiver roles. For example, once the Receiver of a payload file of Nominations has successfully processed the payload, they switch to the Sender role to send Nomination acknowledgements back to the original Sender. Internet ET implementations may need to implement both Sender and Receiver capabilities.

The standards adopted for Internet ET should be adhered to by the trading parties as minimum standards. A trading party may offer additional functions or features as options but should not require their use. Such additional features or functions are termed 'mutually agreed to'. If both trading partners agree on the inclusion, the additional feature requirements will be met. If either trading party does not agree to the inclusion of additional features, then the partners must allow for transmission and receipt of data using the minimum standards.

To establish an Internet ET trading partnership with another company, a company needs to exchange technical information about their Internet ET implementation. This may include:

- Contact information
- Public Keys, including key exchange and update policies
- Test URLs
- Production URLs, including alternative paths if available
- Common Code Identifiers (e.g. DUNS number)
- Use of 'time-c-qualifier' if in REQ or RGQ

This may be exchanged using a Technical Exchange Worksheet (TEW). A sample TEW is in Appendix C. In some cases, this information may be exchanged with a Trading Partner Agreement.

Implementation Approaches

The NAESB Internet ET can be constructed using any IT deployment model, including the use of in-house development, consulting/development help from a third-party, Commercial Off-The-Shelf (COTS) software, or an outsourced solution with a third-party. The best solution for each organization must be determined based on the assessment of specific needs and the resources available to that organization.

All parties should fully investigate the ramifications of implementing electronic commerce using the Internet. This includes ensuring that all customer data, internal data, and applications are secured from intruders or other unauthorized parties.

Participation in electronic commerce over the Internet involves hardware, software, and technical expertise. Hardware requirements may include a server to receive incoming Internet ET packages and a firewall to block intruder access. Software includes operating software for the servers, including the firewall, programming languages which support Internet technologies, and encryption/decryption software to provide security during the transfer. Technical expertise may be involved in the development and maintenance of server applications to process incoming files as well as applications to initiate communication with the server of your trading partner.

~~Third-party providers offer a variety of services from a full ‘turn key’ solution to assistance where you require it, including programming, system configuration, system administration and private communication links. Criteria for selecting an outsourced Internet ET service provider should consider their ability and experience with Internet ET standards for HTTP Request and Response validation and processing.~~

Internet ET Network and Communication Requirements

Trading partners should maintain redundant connections to the public Internet for Internet ET sites. These redundant connections should be topographically diverse paths to minimize the probability of a single point of failure. Three possible approaches to redundant connections are:

1. Maintain multiple ISPs and multiple points of connectivity, each of which was identified by the same URL making the process of redundancy transparent to the Sender.
2. Maintain different Internet connectivity URLs (presumably on topographically different ISPs). For this to result in communication redundancy, the Sender should know of the existence of the secondary URL and have programming in place that will automatically switch batch-browser transmissions to the secondary URL when the primary URL is unavailable.
3. Maintain multiple connections to the same ISP. This involves only one URL but the presumption would be that the ISP would provide alternate diverse paths for the URL.

Servers may maintain multiple URLs and, if such have been disclosed, the Sender should attempt to use these during primary URL outages. The redundant public Internet connections can be through a single ISP or multiple ISPs. If multiple URLs are provided for Internet ET access, the following conditions should be met:

- The information provided by each URL should be exactly the same, although the ‘trans-id’ sequences may differ.
- The trading partners should be informed of both URLs and their availability.
- The URLs should be identified as primary and secondary if either:

There is a TSP connection speed difference between the URLs (The faster connection listed as primary)	OR	One URL is only available when the other is down (primary URL being the most available)

- The URLs should be listed as primary and alternate if:

The URLs have the same TSP connection speed	AND	The URLs are customarily available simultaneously

In the context of communication redundancy, a URL is considered available if all the TCP/IP facilities are properly functioning up to and including the HTTP service. This includes firewalls, DNS servers, routers, hubs, LANs, etc. between ~~your~~ HTTP server's and ~~your~~ Internet Service Provider's point of presence.

In this context redundancy refers to normal operations redundancy, not to disaster recovery contingencies. Disaster recovery contingencies are not addressed in NAESB Internet ET standards.

Private network connections to access NAESB Internet ET sites may be at any point on a party's firewall boundary at the party's discretion on a non-discriminatory basis. The specific type and speed of their connection should be mutually agreed. It is at the discretion of the party how multiple private network connections should be managed.

TCP Communications

NAESB Internet ET Principle 4.1.x37 and NAESB Internet ET Standard 4.3.x70 restrict the TCP ports used as a standard for Internet ET communications. The use of NAESB standard TCP ports may require modifications in the Sender's and Receiver's firewalls to allow for communications with various trading partners' Internet ET implementations. Parties should indicate to their trading partners which specific TCP ports are required to be opened to conduct electronic communication.

Internet ET allows the following TCP Ports (not UDP ports)

- HTTP HTTPS 80, 443, 5713, 6112, 6304, 6874, 7403
- TCP Optional 8001-8020**

**The reservation of 20 optional ports provides for additional security and for implementations such as load balancing. Parties should minimize the number of ports used for Internet ET.

Other Communication Protocols

HTTP POST - HTTP POST is the standard method for transporting Internet ET packages to trading partners. The POST method allows the upload of complete datasets without special encoding.

MIME 'multi-part' - Internet ET packages are created using the 'multi-part' content type.

Sending Internet ET Packages

Internet ET supports both interactive and batch browsers. Interactive web browsers provide for low-cost access to Internet ET capabilities. A batch browser allows organizations to maximize their level of automation. The batch browser can be an event-driven mechanism used to push Internet ET packages to ~~your~~ trading partners in real-time or near real-time, while providing better audit trails.

Receiving Internet ET Packages

Receiving Internet ET packages and transaction payloads requires a Receiving Program. The Receiving Program:

- Parses the Internet ET package parameters and files to determine if the appropriate parameters were transmitted
- Saves a log including a timestamp for the package
- Stores the payload file
- Sends the Receipt as an HTTP Response to the Sender/Client with the timestamp and other required Receipt elements

In some cases the Receiving Program decrypts the file prior to sending the Receipt. In this scenario decryption errors would be communicated in the Receipt. Some trading partners decrypt after sending the Receipt. Decryption errors detected after the Receipt is sent are communicated to trading partners using Internet ET Error Notification standards. Parties should notify trading partners of how decryption errors will be communicated.

If trading partners mutually agree to use signed Receipts, then the application would additionally attach a digital signature to the Receipt.

After the Receiving Program performs its functions without errors, the payload file is forwarded to other processes including security, translation, and back-office systems.

Security

NAESB Internet ET establishes several security measures as standards to ensure a minimum level of confidence in conducting business over the Internet, and to provide uniformity in the implementation of security. Four security concepts, often referred to by the acronym PAIN, are vital to protecting Internet ET packages:

- Data **P**rivacy
- **A**uthentication
- Data **I**ntegrity
- **N**on-repudiation

Data Privacy and Encryption

Privacy is the assurance to an entity that no one can read a particular piece of data except the receiver(s) explicitly intended. Data privacy is accomplished by encrypting payload files. Internet ET allows encryption using:

OpenPGP, defined by (IETF RFC 2440) with modifications described in this specification	**OR**	PGP 2.6 or higher, with RSA keys can be used on a mutually agreed basis

Internet ET uses base64-encoding and 128-bit SSL to protect username and password.

Authentication

Authentication is the assurance to one entity that another entity is who he/she/it claims to be. Basic authentication is the required standard to prevent intruders from connecting to Internet ET Web sites. Internet ET uses 128-bit SSL-protected usernames and passwords to establish authentication. Optional techniques such as firewall security enable further authentication.

Integrity

Integrity is the assurance to an entity that data has not been altered, intentionally or unintentionally, between there and here, or between then and now. Data Integrity is established via OpenPGP/PGP encryption, and via the 'content-length' HTTP header field.

Non-Repudiation

Non-repudiation is the assurance to an entity that a party cannot deny having engaged in the transaction, or having sent the electronic message. It is like a Notary seal. The Sender of a file may optionally include in the Internet ET package a digital signature that is created using their Private Key. The Receiver knows the Sender is legitimate by decoding the digital signature using the Sender's Public Key.

B. GENERAL STANDARDS

Principles:

0.1.1 An entity is a person or organization with sufficient legal standing to enter into a contract or arrangement with another such person or organization (as such legal standing may be determined by those parties) for the purpose of conducting and/or coordinating energy transactions.

0.1.2 There should be a unique entity common code for each entity name and there should be a unique entity name for each entity common code.

Standards:

0.3.1 Entity common codes should be 'legal entities', that is, Ultimate Location, Headquarters Location, and/or Single Location (in Dun & Bradstreet Corporation ('D&B') terms). However, in the following situations, a Branch Location (in D&B terms) can also be an entity common code:

1. when contracting party provides a D-U-N-S® Number at the Branch Location level;

OR

2. to accommodate accounting for an entity that is identified at the Branch Location level.

C. INTERNET ELECTRONIC TRANSPORT RELATED STANDARDS

Principles:

[10].1.1 The Internet Electronic Transport (ET) does not pick winners, rather it should create an environment where the marketplace can dictate a winner or winners (4.1.2).

[10].1.2 Internet ET solutions should be cost effective, simple and economical (4.1.3).

[10].1.3 Internet ET solutions should provide for a seamless marketplace for energy (4.1.4).

[10].1.4 Parties should interface with third-party vendors according to NAESB Internet ET standards (4.1.6).

[10].1.5 Electronic communications between parties to the transaction should be done on a non-discriminatory basis, whether through an agent or directly with any party to the transaction (4.1.7).

[10].1.6 Protocols and tools that parties elect to support should be 'Internet-compatible' (4.1.12).

[10].1.7 The industry should use standard policies and guidelines for testing (4.1.14).

[10].1.8 The NAESB Internet ET should not set standards for site-level security. Individual organization security standards should be relied upon (4.1.15).

[10].1.9 Trading partners should maintain redundant connections to the public Internet for NAESB Internet ET Web sites. These redundant connections should be topographically diverse (duality of) paths to minimize the probability of a single point of failure (4.1.36).

[10].1.10 Trading Partners should mutually select and use a version of the NAESB Internet ET standards under which to operate, unless specified otherwise by government agencies. Trading Partners should also mutually agree to adopt later versions of the NAESB Internet ET standards, as needed, unless specified otherwise by government agencies (4.1.39).

Definitions:

[10].2.1 'Internet ET Testing'. Testing electronic packages between trading partners includes testing of: A) Connectivity; B) Encryption/Decryption; and C) Digital signatures where appropriate (4.2.20).

[10].2.2 'Fail-over' defines a prescribed process executed when a NAESB Internet ET Client fails to establish a connection to the target NAESB Internet ET Server (4.2.21x).

[10].2.3 'Trading Partner' is a party that enters into an agreement with another party to transact business electronically using the Internet ET standard (4.2.22x).

[10].2.4 'Originating party' is any party originating/creating the package. This could also include a third-party (4.2.23x).

[10].2.5 'Third-Party' is any organization that a trading party uses to provide services to comply with the required elements of the Internet ET (4.2.24x).

[10].2.6 'Receiving Party' is any party that hosts (either in-house or outsourced) an Internet ET compliant server capable of receiving Internet ET packages (4.2.25x).

[10].2.7 ‘Receiving Program’ is a program or set of programs that process HTTP Requests from a Sender. The Receiving Program is responsible for generating the ‘gisb-acknowledge-receipt’, which includes any party that hosts (either in-house or outsourced) an Internet ET compliant server capable of receiving Internet ET packages (4.2.25x).

[10].2.8 ‘Trading Partner Agreement’, or ‘TPA’ is a legal agreement between trading parties. The TPA often dictates service level agreements and problem remediation processes. The TPA may include technical exchange information such as URLs, et cetera (4.2.26x).

[10].2.9 ‘Batch Browser’. A Browser that can be run with little or no manual operation or intervention. See ‘Browser’.

[10].2.10 ‘Browser’. A software program capable of generating HTTP Requests, including HTTP POST requests.

[10].2.11 ‘Client’. The computer hardware and software used by the Sender to transmit an Electronic Package to the Receiver’s Server. A Client can be fully-automated or manual.

[10].2.12 ‘COTS’. Commercial Off-The-Shelf; software that can be purchased and that requires little or no customization.

[10].2.13 ‘Electronic Package’. A data stream sent via HTTP POST that contains envelope header information and Payload File(s). The Payload Files are encrypted using defined Internet ET encryption techniques.

[10].2.14 ‘Error Notification’. Error Notification is a package sent from the Receiver of the original data to the Sender when errors are trapped after the Internet ET Receipt is sent. This is normally used for decryption errors detected after the Internet ET Receipt has been sent.

[10].2.15 ‘HTTP Request’. The stream of data sent from the Client to the Server that includes header information and payload data.

[10].2.16 ‘HTTP Response’. The stream of data sent from the Server to the Client in response to an HTTP Request, including the Receipt.

[10].2.17 ‘HTTP Server’. The computer hardware and software used by the Receiver to receive HTTP Requests from the Sender’s Client, and to send HTTP Responses to the Sender’s Client. The Server is an HTTP/Web Server.

[10].2.18 ‘IETF’. Internet Engineering Task Force; a body of technical experts that set standards for the Internet known as Request for Comments (RFC’s).

[10].2.19 ‘Interactive Browser’. A Browser that requires manual operation or intervention. See ‘Browser’.

[10].2.20 ‘Internet EDM’. The GISB and NAESB WGQ standards up to and including Version 1.7. The ‘Internet ET’ and ‘QEDM’ standards were derived from these WGQ EDM standards.

[10].2.21 ‘Internet ET’ or ‘Internet Electronic Transport’. The NAESB standards for the secure transport of electronic information between trading partners, building upon WGQ EDM Version 1.7.

[10].2.22 ‘Payload Files’. The data contents inside of an electronic package. NAESB Internet

ET is content-independent.

[10].2.23 ‘Protocol Failure’. A protocol failure occurs any time a sending party’s NAESB Internet ET server cannot connect to the receiving party’s NAESB Internet ET server. For example, if a server tries to connect to a server and fails, or tries to post a file and fails, this is a protocol failure.

[10].2.24 ‘Exchange Failure’. An exchange failure is when a sending party’s NAESB Internet ET server has had three or more protocol failures over a period of time no less than thirty minutes and no more than two hours.

[10].2.25 ‘QEDM’. Quadrant-specific Electronic Delivery Mechanism; the set of standards for each NAESB quadrant that define the EDM standards for EDI, flat-files, electronic bulletin boards, and other technologies. The QEDM excludes electronic transport practices and standards. The QEDMs were derived from the GISB and NAESB WGQ Internet EDM standards.

[10].2.26 ‘Receipt’. The HTTP Response sent from the Receiver to the Sender that includes the ‘gisb-acknowledge-receipt’ section with a timestamp and OK/error status.

[10].2.27 ‘Receiver’. The party that receives an Internet ET electronic package.

[10].2.28 ‘Sender’. The party that sends an Electronic Package.

[10].2.29 ‘QoS’. Quality of Service; term used to define what level of network bandwidth is guaranteed or assured. The Internet does not offer guaranteed quality of service.

[10].2.30 ‘Technical Exchange Worksheet’ or ‘TEW’. A document or worksheet used to communicate important information related to the technical implementation of Internet ET; includes information such as URLs, contacts and Public Key policies.

[10].2.31 ‘TCP’. Transmission Control Protocol; IETF RFCs 793, 1122, 1323
See http://www.itprc.com/tcpipfaq/default.htm.

[10].2.32 ‘RSA’. A mathematical algorithm for encryption developed by Rivest/Shamir/Adleman. See http://world.std.com/~franl/crypto/rsa-guts.html.

[10].2.33 ‘SSL’. Secure Sockets Layer; a privacy technique that uses encryption to hide information from electronic observers on the Internet. See http://developer.netscape.com/docs/manuals/security/sslin/contents.htm.

[10].2.34 ‘PGP’. Pretty Good Privacy; software used to create Public and Private Keys for privacy and digital signature applications. See http://www.uk.pgp.net/pgpnet/pgp-faq/

[10].2.35 ‘Private Key’. The sequence of digits known as a ‘key’ that is kept private by the owner of a digital certificate, and is used by the certificate owner in encryption and decryption algorithms.

[10].2.36 ‘Public Key’. The sequence of digits known as a ‘key’ that an owner of a digital certificate shares with trading partners. The trading partners use the public key in encryption and decryption algorithms in electronic transactions with the certificate owner.

[10].2.37 ‘HTTP’. Hypertext transport protocol; Assumes version HTTP/1.1; IETF RFCs 2616, 2069. See http://www.w3.org/Protocols/Specs.html.

[10].2.38 ‘MIME’. Multipurpose Internet Mail Extensions; IETF RFCs 2045, 2046, 2047, 2048, 2049; See http://www.faqs.org/rfcs/rfc2045.html.

Standards:

[10].3.1 All parties sending and receiving data should accept a TCP/IP connection (4.3.1x).

[10].3.2 Trading partners should retain audit trail data for at least 24 months. This data retention requirement does not otherwise modify statutory, regulatory, or contractual record retention requirements (4.3.4).

[10].3.3 The designated Internet ET Server/Receiver site should be accessible via the public Internet. This does not preclude location of the designated site on a private intranet, as long as the designated site is also accessible via the public Internet (4.3.7).

[10].3.4 The minimum acceptable protocol should be HTTP. All sending and receiving parties should be capable of sending and receiving the HTTP versions supported by NAESB Internet ET (4.3.8).

[10].3.5 A timestamp designates the time a file is received at the Receiver's designated site. The timestamp consists of the 'time-c' data element, and in some cases the 'time-c-qualifier' data element. Refer to QEDM standards for use of the 'time-c-qualifier' (4.3.9).

[10].3.6 The Receiver generates a timestamp upon the successful receipt of a complete file. The timestamp should be generated by the Receiving Program immediately, prior to further processing by the Receiving Program.

[10].3.7 After timestamp generation, the Receiver and sends an immediate HTTP Response to the Sender. The 'gisb-acknowledgement-receipt', which includes the timestamp data element(s), is the primary part of the HTTP Response. (4.3.9)

[10].3.8 The Server clock generating the timestamp should be synchronized with the National Institute of Standards and Technology (NIST) time in order to mitigate discrepancies between the clocks of the Sender and Receiver. Computer clocks should be synchronized as necessary to ensure at minimum +/- 5 second synchronization with an atomic clock. Specific business processes may have tighter synchronization requirements (4.3.10x).

[10].3.9 The HTTP Response should be sent to the Internet Protocol (IP) address of the HTTP Request (4.3.11x).

[10].3.10 At a minimum, one designated site for receipt should be identified for each trading partner. That site should be identified by a specific Uniform Resource Locator (URL). This does not preclude multiple designated sites being mutually agreed to between trading partners (4.3.12).

[10].3.11 The Sender should make three attempts to complete a unit of work. A unit of work consists of one complete HTTP POST transaction as defined in the technical specification of the HTTP protocol (IETF RFC 1945) (4.3.13).

[10].3.12 A failure to complete a unit of work is a protocol failure.

[10].3.13 Three protocol failures within a 30-minute timeframe is an exchange failure.

[10].3.14 The Internet ET roles for Sender and Receiver are defined in the following table. The entire table defines a unit of work:

Client (Sender)	Server (Receiver)	Receiving Program (Receiver)
	Listen for Connect	
Connect	Accept Connection	
Write HTTP Request	Read HTTP Request	Start of Receipt

Write HTTP Request	Read HTTP Request	
EOF (send)	Read HTTP Request	End of Receipt
Read HTTP Response	Write HTTP Response	
Received		
EOF HTTP Response		

(Cross Reference 4.3.14)

[10].3.15 Trading partners should implement all security features (privacy, secure authentication, integrity, and non-repudiation) using a file-based approach via a commercially-available implementation of:

- An OpenPGP product as defined by IETF RFC 2440, or
- On a mutually agreed basis, PGP version 2.6 or greater using the RSA algorithm to generate keys

(Cross Reference 4.3.15)

[10].3.16 Trading partners should implement basic authentication.

[10].3.17 Encryption keys should be self-certified. The exchange of Public keys should be ~~done~~completed electronically such as via email. ~~in a secure manner such as via postal mail. Key policies, including key exchange policies should be communicated to trading partners.~~The exchange of Private keys, if applicable, should be done in a secure manner such as via postal or courier mail. Key policies, including key exchange policies should be communicated to trading partners.

[10].3.18 Encryption keys should have a limited lifetime whose duration is determined by the key's owner. A key's end of life is expressed in the expiration date field contained in each Public Key. A lifetime of one year or less is recommended.

[10].3.19 Internet protocols should be used for accessing all industry business functions (4.3.36).

[10].3.20 Batch and Interactive Browsers should use Internet-compatible common browser software (4.3.37).

[10].3.21 Trading partners should use common codes for legal entities for the Internet ET 'to' and 'from' data elements (4.3.56x).

[10].3.22 Private network connections to NAESB Internet ET servers, which include all NAESB Internet ET standardized Internet communication, may be at any point on a party's firewall boundary at the party's discretion on a non-discriminatory access basis. The specific type and speed of these connections should be mutually agreed. It is at the discretion of each party on how multiple private network connections should be managed, so long as such management is done on a non-discriminatory access basis (4.3.64).

[10].3.23 Parties should be limited to the NAESB Internet ET approved list of available TCP ports for Internet ET implementations (4.3.70x).

[10].3.24 Internet ET implementations should not require any inbound ports to be opened on the Sender's firewall. (4.3.71, 4.1.37)

[10].3.25 Internet ET Servers should use 128-bit Secure Socket Layer (SSL) encryption (4.3.88).

D. Interpretations

NAESB has adopted the following interpretations of WGQ standards that relate to Internet ET implementation.

7.3.50 The question is whether individual implementations are free to use HTTP HEAD command, prior to using the POST command to deliver the NAESB payload. When implementing a NAESB Internet ET solution, the standard clearly relies on the HTTP protocol spec for details of how to implement the protocol. It is also clear that the HTTP POST command should be used, and not the GET command.

Interpretation:

The use of the HTTP HEAD command in NAESB Internet ET is an option, and as such its implementation between trading partners is solely on a 'mutually agreed to' basis, i.e. the Requester is free to propose the use of the HEAD command to its trading partners, but the Requester cannot insist upon its use. Moreover, the Requester must still provide for transmission and receipt, via the standards, to those trading partners that do not consent to the use of the HEAD command. If the Requester seeks the use of the HEAD command as an explicit requirement of NAESB Internet ET they are directed to submit a Request for Standard to NAESB.

5 – RELATED STANDARDS

COMMON CODES

Internet ET uses the D-U-N-S® Number as the common company identifier for the HTTP Request and Response data dictionary 'to' and 'from' HTTP header elements. The D-U-N-S® Number is a 9-digit number assigned to companies by the Dun & Bradstreet Corporation (D&B). The D-U-N-S+4® Number is a 10- to 13-digit number, where characters 10 through 13 are arbitrarily assigned by the owner of the D-U-N-S® Number.

For Internet ET Common Code purposes, an entity will use one and only one D-U-N-S® Number. Entity common codes should be 'legal entities,' that is, Ultimate Location, Headquarters Location, and/or Single Location (in D&B terms). However, in the following situations, a Branch Location (in D&B terms) can also be an entity common code:

1. When the contracting party provides a D-U-N-S® Number at the Branch Location level.
2. To accommodate accounting for an entity that is identified at the Branch Location level.

Since D&B offers customers the option of carrying more than one D-U-N-S® Number per entity, please refer to NAESB's Web Page at www.naesb.org for directions on determining the one and only one D-U-N-S® Number constituting the NAESB Internet ET Entity Common Code.

QUADRANT-SPECIFIC ELECTRONIC DELIVERY MECHANISMS (QEDM)

In NAESB business processes, the Internet ET standards are used in conjunction with Quadrant-Specific Electronic Delivery Mechanism standards, found in the QEDM book for each Quadrant. These standards include, but are not limited to, X12/EDI standards, flat-file standards, web standards, etc.

PARTY ROLES

Various types of parties are involved in NAESB business processes and the use of Internet ET, including distribution companies, end-users, regulatory entities, service providers, and suppliers.

6 – TECHNICAL IMPLEMENTATION - INTERNET ET

INTERNET ET TECHNOLOGIES

The NAESB Internet ET uses the following technologies and components to securely and reliably transport electronic packages to trading partners:

- OpenPGP and PGP encryption and digital signatures
- TCP/IP and HTTP POST. Internet ET uses a specifically-structured HTTP POST to transport payload data from one trading partner to the other
- MIME multi-part encoding. Internet ET package structure requires that each section of the package be encoded
- A ‘Client’, running at the Sender’s site as ‘batch’ or ‘interactive’ browser software. This software is referred to in this document as ‘Client’
- A ‘Server’ running at the Receiver’s site, usually on a dedicated computer. This is a Web or HTTP server, and is referred to in this document as ‘Server’

ELECTRONIC TRANSPORT LIFE CYCLE

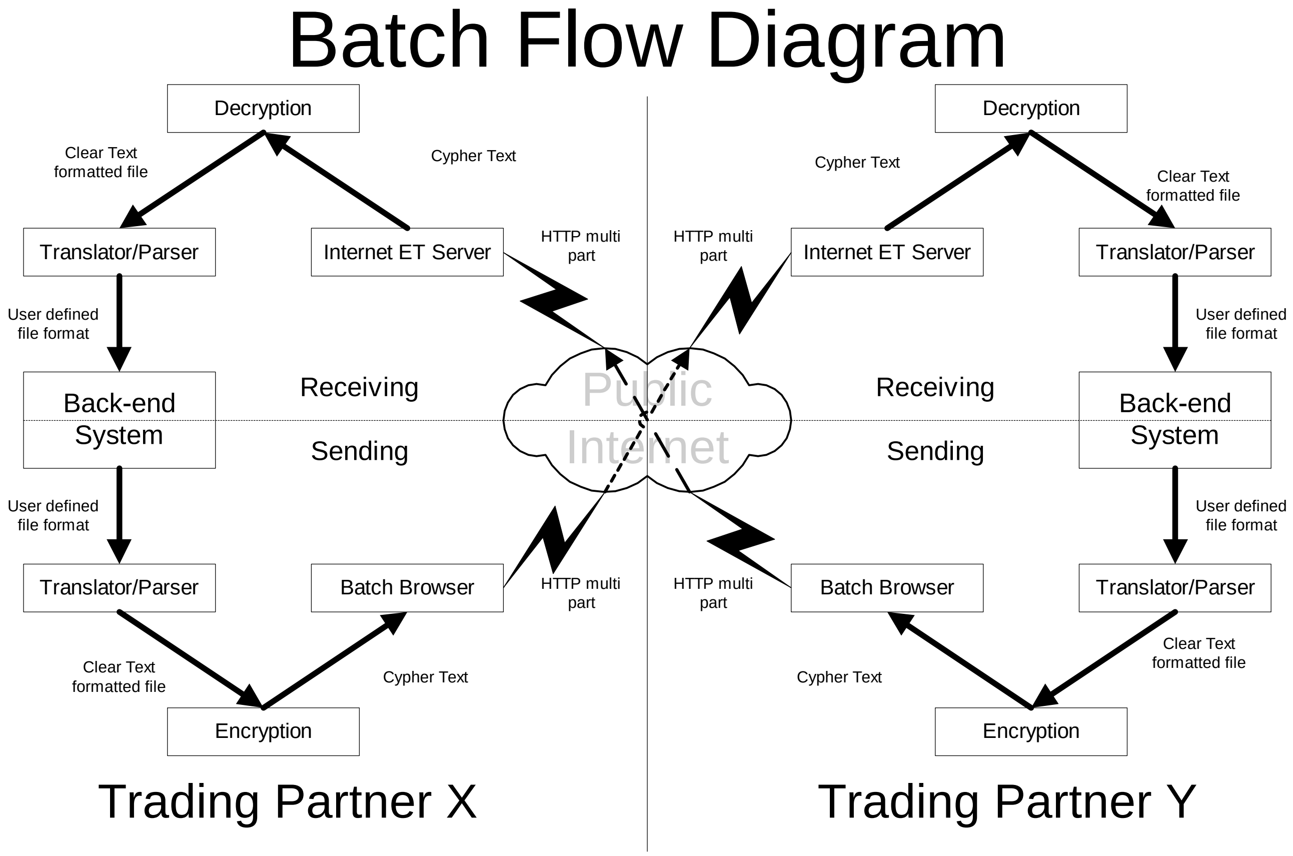
The life cycle of an Electronic Package using Internet ET is described below:

Sender		Receiver
• Collects payload data to be sent • Encrypts payload • Prepares digital signature if necessary		
• Uses browser to create Electronic Package multi-part HTTP Request with header data elements and payload. • Uses HTTP POST to send the electronic package to the Receiver	↘	
		• Receives the HTTP Request on their Web/HTTP Server • Validates Sender information from HTTP Request Header data elements and payload • **Decrypts payload file • Prepares Receipt • Checks digital signatures if necessary
	↙	• Sends Receipt with either ‘OK’ or error message
• Updates logs • If errors, correct errors then repeat process		• **Decrypts payload file
	↙	• If errors in decryption, sends Error Notification to Sender
• Receives Error Notification • Updates logs • correct errors then repeat process		• Updates logs • If no errors, Receiver processes contents of payload

**Parties may choose to decrypt file before or after Receipt is sent to Sender.

Batch Flow Diagram

The flow of data to and from trading partners in an automated environment is diagrammed below.



ANATOMY OF AN INTERNET ET PACKAGE

An Internet ET package consists of the following sections:

- **Envelope header.** This section contains the envelope information needed to communicate who the Sender and Receiver are, as well as other envelope information.
- **Payload.** This section contains the payload file. Internet ET allows for only one payload file per package.
- **Digital Signatures.** If used, the package should contain a section that is the digital signature.

ENVELOPE DATA DICTIONARY

The data dictionary on the next page details standard data elements, each with element name and description.

Data Dictionary for Internet ET

Business Name	Definition	Format	Usage*	Condition
from**	the party sending the transaction	Common Code Identifier format	in Request; M	used in file transmittal; displayed in HTTP Response; and, used in posting back decryption-related errors
input-data	the filename for the transaction data set transmitted	including drive letter and directory name with filename if needed	in Request; M	used in file transmittal of any transaction data sets; and, used for posting back all transaction value pairs for a transmittal that had decryption-related errors.
input-format	descriptor of the data format used for the file transmitted	as defined by QEDM	in Request; M	NAESB standard format indicator used in file transmittal
receipt-disposition-to	the party to receive receipts, the value should be the same as the 'from'	Common Code Identifier format	in Request; M	used in file transmittal and in posting error notifications
receipt-report-type	type of receipt type being requested by Sender	gisb-acknowledgement-receipt	in Request; M	used in file transmittal and in posting error notifications
receipt-security-selection	used to request signed receipts	signed-receipt-protocol=required,pgp-signature;signed-receipt-micalg=required,md5	in Request; MA	used in file transmittal and in posting error notifications
refnum	used by the party to assign a unique message identifier for tracing purposes	maximum 40 character integer value	in Request; MA	May be used by Sender to send tracking information to a recipient. Use of this data element is by mutually agreed. This data element is conceptually similar to a Message-ID filed within RFC 822.

request-status	status describing success or failure of transmission at recipient Server	ok;
		EEDM###:error description;
		WEDM###:warning description.
		see Table A, 'Internet EDM Standard Error Codes and Messages'
			in Response;
			M
				'ok' is returned if all is fine with processing; error messages/warnings and their related descriptions are returned if problems were encountered in processing.

server-id	uniquely identifies the Server processing the transaction	domainname or hostname.domainname; no embedded spaces allowed	in Response; M	displayed in the HTTP Response and posted back for any decryption-related errors

Business Name	Definition	Format	Usage*	Condition
time-c	the time file transfer is complete at the Server	yyyymmddhhmmss	in Response; M	displayed in the HTTP Response and posted back for any decryption-related errors; refer to QEDM for quadrant-specific use
time-c-qualifier	delta from UTC (ref ISO 8601)	-ZZ; +ZZ	in Response; MA	displayed in the HTTP Response and posted back for any decryption-related errors; refer to QEDM for quadrant-specific use
to **	the party the transaction was sent to	Common Code Identifier format	in Request; M	used in file transmittal and displayed in HTTP Response and posted back for any decryption-related errors
transaction-set	name of the document type being sent	8 character code; refer to NAESB REQ Implementation Guide, Related Standards Tab, Hypertext Transfer Protocol (HTTP) section, HTTP transaction-set Code Values table.	in Request; MA	used in file transmittal
trans-id	sequential number assigned to the transaction by the Server upon processing before being passed to the decryption process	integer up to 15 characters in length	in Response; M	displayed in the HTTP Response and posted back for any decryption-related errors
version	the NAESB Internet ET version being used by the Sender	numeric, decimal notation (e.g. 1.6)	in Request; M	used in file transmittal and in posting error notifications

*The **Usage** column defines whether the element appears in the HTTP Request (Client-generated) or the HTTP Response (Server-generated), the order in which the element appears in the data stream, and whether the field is Mandatory (M) or Mutually-Agreed-To (MA).

** Common Code Identifier

SENDING INTERNET ET PACKAGES

General Flow

The following is an example of the steps necessary to send an Internet ET package:

1. Open HTTP connection
2. Check connection status. If in error, re-queue package according to Internet ET standards. This check should be performed here and throughout the following processes.
3. Post, including a) Authentication, b) Send multipart form, c) Receive HTTP Response data
4. Check connection status. If in error re-queue package according to Internet ET standards
5. Check HTTP status code (200 is good, less than 300 may be acceptable). If status is not successful re-queue package according to Internet ET standards
6. Close connection - wait for other end to close in a reasonable time
7. Parse HTTP Response data elements
8. If request-status ok, then log success
9. If request-status error, then log error
10. If no valid request-status re-queue package according to Internet ET standards
11. Remove package from sending queue when successful or when failed completely

If trading partners agree to implement signed receipts, then the sending party must include the 'receipt-security-selection' data element in the posted data. The receiving party must digitally sign the 'gisb-acknowledgement-receipt' and encapsulate the 'gisb-acknowledgement-receipt' and digital signature body parts within a MIME envelope with a 'content-type' of 'application/pgp-signature'.

Using an Interactive Browser for Internet ET

Electronic packages can be uploaded to a trading partner using an interactive browser secured using SSL 128-bit encryption. Sending electronic packages via an interactive browser is ideal for a small volume of package transfers, or as a back-up method to any batch or automated process.

To use an interactive browser to upload data, an HTML document must be created with an HTML <FORM> element that allows the Sender to type in any necessary data elements, such as 'to', 'from', 'input-format', and the name of the file to be uploaded. When the user submits the form, an HTTP POST is sent to the Server with the package, which includes the uploaded file and the required data elements.

The following example is an HTML document with a form that specifies the POST method and contains the required data elements. This type of HTML form could be used with any browser that supports multipart POST with a file upload.

EXAMPLE: HTML DOCUMENT WITH A FORM FOR MULTIPART POST USING AN INTERACTIVE BROWSER:

```
<HTML><HEAD><TITLE>NAESB Internet ET Package Upload</TITLE><H1><CENTER>NAESB Internet ET
Package Upload</CENTER></H1></HEAD>
<BODY><HR>
<FORM ENCTYPE="multipart/form-data" ACTION="http://www.target.server/cgi-bin/upload.exe"
METHOD=”POST”>
Enter Common Code Identifier for ‘From’ and ‘To’:
From:
<INPUT TYPE="text" NAME="from" SIZE=20 VALUE=""><br>
To:
<INPUT TYPE="text" NAME="to" SIZE=20 VALUE=""><br>
NAESB Internet ET Version:
<INPUT TYPE=”text” NAME=”version” SIZE=5 VALUE=”1.6”><br>
Deliver Receipt To:
<INPUT TYPE=”text” NAME=”report-disposition-to” SIZE=20 VALUE=””><br>
Receipt Type:
<INPUT TYPE=”text” NAME=”receipt-report-type” SIZE=30
VALUE=”gisb-acknowledgement-receipt”><br>

IF requesting signed receipts also include:  Receipt Type:
<INPUT TYPE=”text” NAME=”receipt-security-selection” SIZE=30
    VALUE=”signed-receipt-protocol=required, pgp-signature; signed-receipt-
micalg=required, md5”><br>
Format of this file:
<INPUT TYPE="text" NAME="input-format" SIZE=6 VALUE="X12"><br>
Send this file:
<INPUT NAME="input-data" TYPE="FILE"><br>
<INPUT TYPE="submit" VALUE="Send File"><br>
</FORM>
</BODY></HTML>
```

The important characteristics of the form within the HTML document are:

- ENCTYPE= specifies the encoding type. The ‘multipart/form-data’ encoding type is identified as the standard encoding methodology.
- ACTION= specifies the URL that will receive the uploaded data. The TEW or TPA identifies the URLs for both parties.
- METHOD= specifies the HTTP protocol method. ‘POST’ has been defined as the Internet ET standard method.
- <INPUT ...>. HTML INPUT elements include the required data elements such as ‘from’, ‘to’, and ‘input-format’. Refer to the data dictionary for the complete list of required data elements.

When a user selects the ‘Send File’ button, the interactive browser will take the values entered in the input fields and reformat them into a data stream, formatted according to the encoding type. The file identified for upload is opened and its contents are included in the data stream. The data stream is then sent to the URL specified by ‘**ACTION=**‘, which indicates a Server Receiving script or program written to receive the package.

Using a Batch Browser for Internet ET

A batch browser is used by companies that want to automate their transport processes and/or prefer to minimize human involvement. A batch browser is initiated by a program or a script.

A batch browser can be created via custom programming. A batch browser is coded to perform the same formatting as an interactive browser, formatting a data stream that conforms to the

HTTP and Internet ET protocols. A batch browser must be coded as a 'TCP sockets' program. See the section 'Writing a Batch Browser'.

Authentication

Userids and passwords must be base64-encoded. HTTP basic authentication includes a 'userid' and 'password'. Interactive browsers include a basic authentication feature that automatically prompts for 'userid' and 'password'. In a batch browser, the authentication must be specifically coded. The 'userid' and 'password' are to be base64-encoded within the document header. Base64-encoding utilities are readily available on the Internet as either public domain software or commercial libraries.

Server Response

The Server will send a 'gisb-acknowledgement-receipt' in the HTTP Response to the Client before dropping the Client's connection. If the transacting parties agree to use signed receipts, the Server applies a digital signature to the 'gisb-acknowledgement-receipt' and encapsulates the entire package in a MIME envelope of 'content-type: application/pgp-signature'.

The 'gisb-acknowledgement-receipt' returned from the Server contains the 'time-c' and the 'time-c-qualifier' (where applicable) Receipt timestamps that are recorded when the final byte from the package upload is received and stored. This Receipt timestamp is the official timestamp regarding transaction turnaround deadlines as defined in Internet ET and QEDM standards. This timestamp and all other pertinent package transmittal information should be logged by the Receiver when the posted package is stored on the Server, and logged by the Client. Errors or warnings should be logged at both the Client and Server.

Sender HTTP Request Data Elements

The HTTP Request will provide all required data elements in the ORDER DEFINED BELOW. Any 'mutually-agreed-upon' data elements will follow the required data elements in the data stream. Refer to the section 'Data Dictionary for Internet ET' for descriptions of these data elements.

<u>Required Data Elements, Listed in the Required Order:</u>

1. from
2. to
3. version
4. receipt-disposition-to
5. receipt-report-type
6. input-format
7. input-data

<u>Mutually Agreed Upon Data Elements</u>

8. transaction-set
9. receipt-security-selection

Writing a Batch Browser

A batch browser Client needs to simulate the actions of an interactive browser Client. As stated earlier, the interactive browser Client will take the HTML form, reformat the information according to the HTTP protocol, then send the data stream to the Server. The reformatting adds a header and places field delimiters around the data items.

A batch browser needs to produce the same kind of data stream and, therefore, writing a batch browser requires some specific knowledge of the HTTP protocol.

EXAMPLE: A TYPICAL HEADER SENT TO THE SERVER

```
POST /cgi-bin/AS2dispatcher HTTP/1.1
Referer: http://www.get.a.life/upl.htm
Connection: Keep-Alive
User-Agent: brow v0.1 XYZ Corp.
Host: localhost
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
content-type: multipart/form-data; boundary=---------------------------87453838942833
Content-Length: 5379
```

POST Line - In the example above, the first line indicates the POST method was used and identifies which Receiving Program to call:

```
POST /cgi-bin/AS2dispatcher HTTP/1.1
```

Content Type - The ‘content-type’ line indicates that the encoding method is multipart, and identifies the character string used as the boundary.

```
content-type: multipart/form-data; boundary=---------------------------87453838942833
```

Boundary String - The ‘boundary=‘ identifies the string that will appear between each field as a delimiter. In this example, the boundary is comprised of 27 hyphen characters followed by a number.

```
content-type: multipart/form-data; boundary=---------------------------87453838942833
```

The boundary can be ANY character string that you choose. The string used CANNOT OCCUR ANYWHERE ELSE IN THE PACKAGE BEING SENT. This is usually accomplished by using either the system clock or a random number so that even if by some remote chance the string appears in the document it would not appear in any re-transmission of the file. It is strongly recommended that a relatively long string be used as a boundary.

The boundary string, when used as a separator, REQUIRES TWO HYPHEN CHARACTERS APPENDED TO THE FRONT of the string. The LAST boundary required in the form is TWO HYPHEN CHARACTERS APPENDED TO THE BACK of the separator boundary, used to indicate to the Server program that this is the end of the data.

```
---------------------------87453838942833--
```

Content Length - The ‘content-length’ value should match the number of bytes contained in the entity body including the characters in the boundary lines, variable content, blank lines, etc. ‘content-length’ indicates to the Server how much data are going to come after this point. In the example above, the content length is:

```
Content-Length: 5379
```

Envelope / Required Data Elements. The envelope information for the package (‘to’, ‘from’, etc) is included in a series of boundaries that include the ‘content-disposition’ and ‘name=‘

qualifiers, followed by the data element value. The example below includes the ‘from’ field as ‘123456789’ and the ‘to’ field as ‘234567890’.

```
-----------------------------87453838942833
content-disposition: form-data; name="from"

123456789
-----------------------------87453838942833
content-disposition: form-data; name="to"

234567890
```

The ‘content-disposition’ identifier defines that ‘form-data’ is contained in the element. The ‘name=‘ identifier defines the name of the data element. These data element names must match the name specified by Internet ET Data Dictionary. The ‘name=‘ identifier is not completely relevant since the fields should be present in the correct order, but this field should be checked to verify the validity of the form content.

The actual data value of the field is always preceded by a blank line. This is typically used as a marker for the Server program to indicate that a data value will follow. For example, note the blank line preceding ‘X12’ in the example. In most programming libraries and commercial products the starting delimiter is ‘\r\n\r\n’ (C notation).

```
-----------------------------87453838942833
content-disposition: form-data; name="version"

1.64
-----------------------------87453838942833
content-disposition: form-data; name="receipt-disposition-to"

123456789)
-----------------------------87453838942833
content-disposition: form-data; name="receipt-report-type"

gisb-acknowledgement-receipt
-----------------------------87453838942833
content-disposition: form-data; name="input-format"

x12
```

Payload. The content or ‘payload’ (EDI, etc) is encrypted and included in its own boundary section.

The data field containing the Internet ET payload file has two extra identifiers. The ‘filename=‘ element indicates the name of the file sent from by the Sender. In the example the name of the file is **‘c:\temp\smallnom.bin’**.

```
content-disposition: form-data; name="input-data"; filename=”c:\temp\smallnom.bin”
```

The ‘content-type’ element indicates the type of the data being transmitted according to accepted Internet standards.

```
content-type: multipart/encrypted; boundary=--boundary2--200309090001;
protocol="application/pgp-encrypted"
```

Note that encrypted files can be multipart also, which means they will have their own boundary string.

```
-----------------------------87453838942833
content-disposition: form-data; name="input-data"; filename=”c:\temp\smallnom.bin”
content-type: multipart/encrypted; boundary=--boundary2--200309090001;
protocol="application/pgp-encrypted"

----boundary2--200309090001
content-type: application/pgp-encrypted

Version: 1

----boundary2--200309090001
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: PGP 6.5

hQCMAzRG1pEOIOvdAQP+JMr0m/9+8yOL60Z9Vr6fFV81FCExB/o0xmwiMkiwYsHsz0e8sb7ErC340MrNA/dw3taGM
jmI+CXYRF/PLEdg1NZE1ZCtNeL4YdIHAMLWwODGlQxhSucz8rMSgQ5mZzcOJwBdWLW70efgsu/9UljuJjYc1uZ6C0
3eFQv/43fkB+alATtgydxX4g8QK664ad+Jo/XUICSmWBL66fqJR1KLeLf4wTaqGy174Aq48Wpwvg1Eh785zC03UAw
0qg0ugMt86dPeyd91e2JigqwDYEf/DYEKD0J9BGiGpS/uAupNKj8Ocp2IWClxKOGUbxpVNOnNTqWHS/GntegvDE/7
/ewCxDxsnmQS95pOl141QZ1RqbeNaqx2Dq/ra9g65HNchOCzjul5Vi8HHf6Yhg2WnROe+npByyCue6rihqgNVOJwj
0Cvzpb4JE+gMDf3q4ISUb1Fv7/+SSFHDdnhdC5YTpqf1Bc3B07hiLmtTXqNit31EbX9UVElObzSa9ZhxbC6/eSl7N
uf5ZTDsh9nrk+QQJ6FeC9W4cqXLj7IZySaRO8Vtff+4ktqeuhYusT4kSpnk027aw4O/5jomUkfb22CAe4=
=Oiuo
-----END PGP MESSAGE-----
----boundary2--200309090001--
```

Boundary String Terminators - Each multipart stream must be terminated with the boundary string terminator. After the contents of the last data field, the boundary string and the required two-hyphen terminator indicate the end of the multipart encrypted payload. A second boundary terminator string indicates the end of the package:

```
----boundary2--200309090001--
-----------------------------87453838942833--
```

<u>EXAMPLE: AN X12 EDI FILE ENCRYPTED WITH PGP</u>

```
content-type: multipart/encrypted; boundary=--boundary2--200309090001;
protocol="application/pgp-encrypted"
----boundary2--200309090001
content-type: application/pgp-encrypted
Version: 1
----boundary2--200309090001
content-type: application/octet-stream
-----BEGIN PGP MESSAGE-----
Version: PGP 6.5
hQCMAzRG1pEOIOvdAQP+JMr0m/9+8yOL60Z9Vr6fFV81FCExB/o0xmwiMkiwYsHsz0e8sb7Er340MrNA/dw3taGMj
mI+CXYRF/PLEdg1NZE1ZCtNeL4YdIHAMLWwODGlQxhSucz8rMSgQ5mZzcOJwBdWLW70efgsu/9UljuJjYc1uZ6C03
eFQv/43fkB+alATtgydxX4g8QK664ad+Jo/XUICSmWBL66fqJR1KLeLf4wTaqGy174Aq48Wpwvg1Eh785zC03UAw0
qg0ugMt86dPeyd91e2JigqwDYEf/DYEKD0J9BGiGpS/uAupNKj8Ocp2IWClxKOGUbxpVNOnNTqWHS/GntegvDE/7/
ewCxDxsnmQS95pOl141QZ1RQbeNaqx2Dq/ra9g65HNchOCzjul5Vi8HHf6Yhg2WnROe+npByyCue6rihqgNVOJwj0
cVzpb4JE+gMDf3q4ISUb1Fv7/+SSFHDdnhdC5YTpqf1Bc3B07hiLmtTXqNit31EbX9.UVElObzSa9ZhxbC6/eSl7N
uf5ZTDsh9nrk+QQJ6FeC9W4cqXLj7IZySaRO8Vtff+4ktqeuhYusT4kSpnk027aw4O/5jomUkfb22CAe4=
=Oiuo
-----END PGP MESSAGE-----
----boundary2--200309090001--
```

EXAMPLE: AN X12 EDI DATA STREAM BEFORE ENCRYPTION:

```
content-type: application/EDI-X12

ISA~00~ ~01~AAA6300300~14~1234567890000 ~14~2345678900000
...  more data from the X12 file…
IEA~1~000003616
```

EXAMPLE: A COMPLETE ELECTRONIC PACKAGE DATA STREAM

```
POST /cgi-bin/AS2dispatcher HTTP/1.1
Referer: http://www.get.a.life/upl.htm
Connection: Keep-Alive
User-Agent: brow v0.1 XYZ Corp.
Host: localhost
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
content-type: multipart/form-data; boundary=---------------------------87453838942833
Content-Length: 5379

-----------------------------87453838942833
content-disposition: form-data; name="from"

123456789
-----------------------------87453838942833
content-disposition: form-data; name="to"

234567890
-----------------------------87453838942833
content-disposition: form-data; name="version"

1.46
-----------------------------87453838942833
content-disposition: form-data; name="receipt-disposition-to"

123456789
-----------------------------87453838942833
content-disposition: form-data; name="receipt-report-type"

gisb-acknowledgement-receipt
-----------------------------87453838942833
content-disposition: form-data; name="input-format"

X12
-----------------------------87453838942833
content-disposition: form-data; name="input-data"; filename=”c:\temp\smallnom.bin”
content-type: multipart/encrypted; boundary=--boundary2--200309090001;
protocol="application/pgp-encrypted"

----boundary2--200309090001
content-type: application/pgp-encrypted

Version: 1

----boundary2--200309090001
content-type: application/octet-stream

-----BEGIN PGP MESSAGE-----
Version: PGP 6.5

hQCMAzRG1pEOIOvdAQP+JMr0m/9+8yOL60Z9Vr6fFV81FCExB/o0xmwiMkiwYsHsz0e8sb7ErC340MrNA/dw3taGM
jmI+CXYRF/PLEdg1NZE1ZCtNeL4YdIHAMLWwODGlQxhSucz8rMSgQ5mZzcOJwBdWLW70efgsu/9UljuJjYc1uZ6C0
3eFQv/43fkB+alATtgydxX4g8QK664ad+Jo/XUICSmWBL66fqJR1KLeLf4wTaqGy174Aq48Wpwvg1Eh785zC03UAw
0qg0ugMt86dPeyd91e2JigqwDYEf/DYEKD0J9BGiGpS/uAupNKj8Ocp2IWClxKOGUbxpVNOnNTqWHS/GntegvDE/7
/ewCxDxsnmQS95pOl141QZ1RqbeNaqx2Dq/ra9g65HNchOCzjul5Vi8HHf6Yhg2WnROe+npByyCue6rihqgNVOJwj
0cVzpb4JE+gMDf3q4ISUb1Fv7/+SSFHDdnhdC5YTpqf1Bc3B07hiLmtTXqNit31EbX9UVElObzSa9ZhxbC6/eSl7N
uf5ZTDsh9nrk+QQJ6FeC9W4cqXLj7IZySaRO8Vtff+4ktqeuhYusT4kSpnk027aw4O/5jomUkfb22CAe4=
=Oiuo
-----END PGP MESSAGE-----
----boundary2--200309090001--
-----------------------------87453838942833--
```

RECEIVING INTERNET ET PACKAGES

General Flow

The following is an example of the steps necessary to receive an Internet ET package:

1. Parse multi-part form
2. Validate HTTP Request data elements
3. If HTTP Request data elements in error, return appropriate Internet ET standard error code in the HTTP Response data elements
4. Save data
5. Create ‘gisb-acknowledgement-receipt’
6. If using signed receipts, produce a digital signature over the ‘gisb-acknowledgement-receipt’ created in step 5.
7. Encapsulate the ‘gisb-acknowledgement-receipt’ and digital signature body parts in a ‘Content-Type’ of ‘multipart/signed envelope’
8. Return HTTP Response with the ‘gisb-acknowledgement-receipt’ object back to Client
9. Close connection
10. Log final results
11. Route data file to the next process based upon ‘input-format’

Overview of Web Server Receiving Programs

The HTTP Server receives the POST and calls the appropriate Receiving script or program to:

- parse the incoming HTTP Request
- create the Receipt timestamp using the current date and time
- create an HTML Response to the Client

An Internet ET Receiving Program may be implemented using a variety of technologies and techniques, including Active Server Pages (ASP), Common Gateway Interface (CGI), Java Server Pages (JSP), Java Servlets, and Personal Home Pages (PHP). The Internet ET is supported by most commercially available Web/HTTP servers.

The Receiving Program and Process

The Receiving Program must be able to parse the multi-part form. It accomplishes this by finding the boundary string in the ‘content-type’ header and scanning for its occurrences further within the uploaded stream. Upon finding these boundary strings, the program must next determine the ‘content-disposition’ for each data element. This allows detection of the required text elements as well as the Internet ET payload file.

The Receiving Program only stores the payload file and is not concerned with the content of the payload file, which is encrypted. It will use the ‘content-length’ to determine how much data to expect in the body of the package.

A Receiving process requires an executable program or module that is called by the Server when it is identified by a POST operation.

When the Server receives a POST it will first read the header and populate environment variables before calling the Receiving Program. Most HTTP servers read header variables and populate environment variables. Check your HTTP server documentation for more information.

EXAMPLE: A SAMPLE HTTP POST HEADER

```
POST /cgi-bin/AS2dispatcher HTTP/1.1
Referer: http://www.get.a.life/upl.htm
Connection: Keep-Alive
User-Agent: brow v0.1 XYZ Corp.
Host: localhost
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
content-type: multipart/form-data; boundary=---------------------------87453838942833
Content-Length: 5379
```

After reading the HTTP header information, the Server will buffer the remaining data transmitted and call the Receiving Program specified in the POST statement. Do not assume that the Receiving Program is called as soon as the header is read, which can impact your receipt timestamp. The more common implementations buffer the entire transmission before calling the program. Check your server implementation if this characteristic is important to you.

The Receiving Program will have the following data stream available, and will have most of the header data available in environment variables.

EXAMPLE: DATA STREAM AVAILABLE TO RECEIVING PROGRAM

```
-----------------------------87453838942833
content-disposition: form-data; name="from"
123456789
-----------------------------87453838942833
content-disposition: form-data; name="to"
234567890
-----------------------------87453838942833
content-disposition: form-data; name="version"
1.64
-----------------------------87453838942833
content-disposition: form-data; name="receipt-disposition-to"
123456789
-----------------------------87453838942833
content-disposition: form-data; name="receipt-report-type"
gisb-acknowledgement-receipt
-----------------------------87453838942833
content-disposition: form-data; name="input-format"
X12
-----------------------------87453838942833
content-disposition: form-data; name="input-data"; filename=”c:\temp\smallnom.bin”
content-type: multipart/encrypted; boundary=--boundary2--200309090001;
protocol="application/pgp-encrypted"
----boundary2--200309090001
content-type: application/pgp-encrypted
Version: 1
----boundary2--200309090001
content-type: application/octet-stream
-----BEGIN PGP MESSAGE-----
Version: PGP 6.5
hQCMAzRG1pEOIOvdAQP+JMr0m/9+8yOL60Z9Vr6fFV81FCExB/o0xmwiMkiwYsHsz0e8sb7ErC340MrNA/dw3taGM
jmI+CXYRF/PLEdg1NZE1ZCtNeL4YdIHAMLWwODGlQxhSucz8rMSgQ5mZzcOJwBdWLW70efgsu/9UljuJjYc1uZ6C0
3eFQv/43fkB+alATtgydxX4g8QK664ad+Jo/XUICSmWBL66fqJR1KLeLf4wTaqGy174Aq48Wpwvg1Eh785zC03UAw
0qg0ugMt86dPeyd91e2JigqwDYEf/DYEKD0J9BGiGpS/uAupNKj8Ocp2IWClxKOGUbxpVNOnNTqWHS/GntegvDE/7
/ewCxDxsnmQS95pOl141QZ1RQbeN.aqx2Dq/ra9g65HNchOCzjul5Vi8HHf6Yhg2WnROe+npByyCue6rihqgNVOJw
j0cVzpb4JE+gMDf3q4ISUb1Fv7/+SSFHDdnhdC5YTpqf1Bc3B07hiLmtTXqNit31EbX9UVElObzSa9ZhxbC6/eSl7
Nuf5ZTDsh9nrk+QQJ6FeC9W4cqXLj7IZySaRO8Vtff+4ktqeuhYusT4kSpnk027aw4O/5jomUkfb22CAe4=
=Oiuo
-----END PGP MESSAGE-----
----boundary2--200309090001--
-----------------------------87453838942833--
```

This Receiving Program should check for basic validity in the environment variables and the data stream, and then parse the variables/data from the format. Data validations should include:

- The 'REQUEST_METHOD' environment variable is 'POST'.
- The 'CONTENT_TYPE' environment variable should be 'multipart/form-data' and the boundary, which cannot appear anywhere in the transaction being sent.
- The input stream should support binary mode to accommodate encrypted files.
- Each data element should be preceded by the boundary with the required two hyphen characters appearing before it.
- Each data element should contain the correct name on the 'content-disposition' line.
- Each data element should have a blank line ('\r\n\r\n' in C+ notation) before the start of the data.
- All tag values in the HTTP header should be evaluated in a case insensitive manner.
- Improperly formatted input. Finding the end of the stream using both 'content-length' and the boundary string terminator end mark is a good method to detect improperly formatted input.

Acknowledgement Receipt: 'gisb-acknowledgement-receipt'

The Acknowledgement Receipt ('Receipt') is critical to non-repudiation and business process timing. Immediately after the Receiving Program receives the last byte of data from the Sender, the Receiving Program should record the time and construct a 'gisb-acknowledgement-receipt'. This Receipt is sent from the Receiving Program to the Client prior to closing the HTTP connection.

The Receipt is a MIME-formatted text stream that includes the HTTP Response data elements (time-c, time-c-qualifier for REQ/RGQ, request-status, server-id, trans-id) in a 'multipart/report' MIME envelope.

If signed Receipts are used, the 'gisb-acknowledgement-receipt' including the 'multipart/report' envelope, is digitally signed, producing an 'application/pgp-encrypted' body part. Both the 'multipart/report' 'gisb-acknowledgement-receipt' and the 'application/pgp-signature' body parts are placed in a 'multipart/signed' envelope and the entire package is returned to the Sender.

The Receipt name 'gisb-acknowledgement-receipt' retains the 'gisb-' prefix to assure compatibility with legacy GISB EDM implementations. The name is only used in the 'report-type' data element for the MIME part.

Additional Receiving Program Functions

- All data element names of the HTTP Request and Response fields will be in lower case. Note that the Internet ET standard format file contained in the Request and Response may follow a different standard.
- Carriage returns and line feeds will be ignored in all files.
- A field delimiter of '*' will be used in the HTTP Response. Please refrain from displaying a '*' anywhere else in the response so as not to confuse programs that need to parse on this basis.
- No spaces should surround the equal sign or the field delimiter.
- The required data elements must appear first in the HTTP Response and in the order specified. Additional information can be included after the required elements at the server's discretion.

- The first occurrence of the field name within the response will contain the value.
- If an HTML response is given, all data must be presented in a user-readable fashion. For example, if the required machine-readable fields are embedded in comments, another representation of these fields must be presented to the user.

Receiving Process URL Implementation Guidelines

Each company must offer at least one URL to accept files using Internet ET. Companies can offer multiple URLs. Though companies are free to construct a Web site with multiple ‘single-purpose’ URLs (e.g. nominations.xyzcorp.com; enrollments.xyzcorp.com) NAESB recommends the use of one ‘general-purpose’ URL.

The Receiving Program may initiate error notifications after the ‘gisb-acknowledgement-receipt’ is sent (e.g. file decryption errors). Error notifications posted to the Sender would be directed to the Sender’s general-purpose URL.

All URLs that will be required for use in the Internet ET process must be agreed to and defined in a Technical Exchange Worksheet (TEW) or a Trading Partner Agreement (TPA).

HTTP Response ‘gisb-acknowledgement-receipt’ Data Elements

Required HTTP Response Data Elements (listed in the required order)	
WGQ	REQ/RGQ
time-c request-status server-id trans-id	time-c time-c-qualifier request-status server-id trans-id

Examples of HTTP Response Required Data Elements:

EXAMPLE: RESPONSE, SUCCESSFUL, MULTIPART FORMAT:

```
content-type: multipart/report; report-type="gisb-acknowledgement-receipt";
boundary="NAESB7867"

--NAESB7867
content-type: text/html

<HTML><HEAD><TITLE>Acknowledgement Receipt Success</TITLE></HEAD> <BODY><P>
time-c=19960619082855*
request-status=ok*
server-id=coolhost*
trans-id=234423897*
</P> </BODY></HTML>
--NAESB7867
content-type: text/plain

time-c=19960619082855*
request-status=ok*
server-id=coolhost*
trans-id=234423897*
--NAESB7867--
```

Example: Response, Successful, Multipart format, time-c-qualifer for time zone:

```
content-type: multipart/report; report-type="gisb-acknowledgement-receipt";
boundary="NAESB7867"

--NAESB7867
content-type: text/html

<HTML><HEAD><TITLE>Acknowledgement Receipt Success</TITLE></HEAD> <BODY><P>
time-c=19960619082855*
time-c-qualifier=-05*
request-status=ok*
server-id=coolhost*
trans-id=234423897*
time-c-qualifer=-0400
</P> </BODY></HTML>
--NAESB7867
content-type: text/plain

time-c=19960619082855*
time-c-qualifier=-05*
request-status=ok*
server-id=coolhost*
trans-id=234423897*
time-c-qualifer=-0400
--NAESB7867--
```

Example: Response, error, multipart format:

```
content-type: multipart/report; report-type="gisb-acknowledgement-receipt";
boundary="NAESB7866"

--NAESB7866
content-type: text/html

<HTML><HEAD><TITLE>Acknowledgement Receipt Error</TITLE></HEAD> <BODY><P>
time-c=19960619082855*
request-status=EEDM106: Invalid To Common Code Identifier*
server-id=coolhost*
trans-id=234423897*
</P> </BODY></HTML>
--NAESB7866
content-type: text/plain

time-c=19960619082855*
request-status=EEDM106: Invalid To Common Code Identifier*
server-id=coolhost*
trans-id=234423897*
--NAESB7866--
```

EXAMPLE: RESPONSE, WARNING, MULTIPART FORMAT:

```
content-type: multipart/report; report-type="gisb-acknowledgement-receipt";
boundary="NAESB7866"

--NAESB7866
content-type: text/html

<HTML><HEAD><TITLE>Acknowledgement Receipt Warning</TITLE></HEAD> <BODY><P>
time-c=19960619082855*
request-status=WEDM100: Transaction Set Sent, Not Mutually Agreed*
server-id=coolhost*
trans-id=234423897*
</P> </BODY></HTML>
--NAESB7866
content-type: text/plain

time-c=19960619082855*
request-status= WEDM100: Transaction Set Sent, Not Mutually Agreed *
server-id=coolhost*
trans-id=234423897*
--NAESB7866--
```

EXAMPLE: RESPONSE, SUCCESSFUL , SIGNED RECEIPT:

```
content-type:multipart/signed; micalg=pgp-md5; protocol="application/pgp-signature";
boundary=--boundary2--200309090001

----boundary2--200309090001

content-type: multipart/report; report-type="gisb-acknowledgement-receipt";
boundary="NAESB7867"

--NAESB7867
content-type: text/html

<HTML><HEAD><TITLE>Acknowledgement Receipt Success</TITLE></HEAD> <BODY><P>

time-c=19960619082855*
request-status=ok*
server-id=coolhost*
trans-id=234423897*

</P> </BODY></HTML>

--NAESB7867
content-type: text/plain.
time-c=19960619082855*
request-status=ok*
server-id=coolhost*
trans-id=234423897*
--NAESB7867--
----boundary2--200309090001
content-type: application/pgp-signature

-----BEGIN PGP MESSAGE-----

Version: 2.6.26.5

iQCVAwUBMJrRF2N9oWBghPDJAQE9UQQAtl7LuRVndBjrk4EqYBIb3h5QXIX/LC//JV5bNvkZIGPIcEmI5iFd9boEg
vpirHtIREEqLQRkYNoBActFBZmh9GC3C041WGquMbrbxc+nIs1TIKlA08rVi9ig/2Yh7LFrK5Ein57U/W72vgSxLh
e/zhdfolT9BrnHOxEa44b+EI=
=ndaj

-----END PGP MESSAGE-----

----boundary2--200309090001—
```

EXAMPLE: HTML FORMAT RESPONSE, SUCCESSFUL:

```
<html><head><title>Upload OK</title></head>
<body>
<!-- time-c=19960123203618*-->
<!-- request-status=ok* -->
<!-- server-id=coolhost*-->
<!-- trans-id=232323897*-->
<h1>Upload OK</h1>
<b>File Saved at (time-c):</b> 19960123203618<br>
<b>Status (request-status):</b>ok<br>
<b>Server (server-id):</b>coolhost<br>
<b>Transaction ID (trans-id):</b>232323897<br>
</body></html>
```

SENDING INTERNET ET ERROR NOTIFICATIONS

When a Client sends an Internet ET package to a Server, the Server responds with a Receipt. Further back-office processing (e.g. decryption) may be required, and additional errors may be found.

Error Notification transactions are used to communicate transport errors found by the Receiver after the initial receipt is sent to the Sender.

Errors from translation and other back-office processing are outside the scope of the Internet ET.

When a file passes the decryption step, no error notification is sent back to the Client. If the decryption step fails, an error notification must be sent to the Client.

The Error Notification format applies to the posting of an error message after the Sender's session has been disconnected. This error notification is used only if the original HTTP Response is returned with an 'ok'.

Additionally, trading partners are permitted to use digitally-signed error notifications, if both parties mutually agree to do so.

Required Error Notification Data Elements

The data elements for the error notification are the same as those described in the Section 'Sending Transactions', with the exception of the 'input-format' and 'input-data' elements. The file containing the data elements for error notification should not be encrypted.

Required Data Elements for Error Notification (listed in the required order):

1. 'from'
2. 'to'
3. 'input-format'

Error Notification 'input-data' Element Specifications:

- The file containing the data elements for error notification should not be encrypted.
- All data element names will be in lower case in the Error Notification.
- Carriage returns and line feeds will be ignored in all files.

- A field delimiter of '*' will be used in the Error Notification. Please refrain from displaying a '*' anywhere else in the error notification so as not to confuse programs that need to parse on this basis.
- No spaces should surround the equal sign or the field delimiter.
- The required data elements must appear first in the response.
- Additional information can be included after the required elements at the server's discretion.
- The first occurrence of the field name within the response will contain the value.
- An error notification contains two body parts nested within a multipart/report outer envelope with the content-type of 'gisb-error-notification'.
- The first body part contains human readable content in HTML. The second body part contains machine readable content in plain text. Additionally, consenting trading partners can mutually agree to digitally sign error notifications.
- If digital signatures are used, the multipart/report containing the Error Notification is used to create a digital signature body part, identified by a 'content-type' of application/pgp-signature. Both the multipart/report Error Notification and application/pgp-encrypted digital signature body parts are combined in a multipart/signed envelope.

Example: Error Notification Internet ET Package:

```
POST /cgi-bin/AS2dispatcher HTTP/1.1
Referer: http://www.acmeenergy/upl.htm
Connection: Keep-Alive
User-Agent: brow v0.1 XYZ Corp.
Host: localhost
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, */*
content-type: multipart/form-data; boundary=---------------------------87453838942833
Content-Length: 1958
-----------------------------87453838942833
content-disposition: form-data; name="from"

234567890
-----------------------------87453838942833
content-disposition: form-data; name="to"

123456789
-----------------------------87453838942833
content-disposition: form-data; name="version"

1.6
-----------------------------87453838942833
content-disposition: form-data; name="receipt-disposition-to"

123456789
-----------------------------87453838942833
content-disposition: form-data; name="receipt-report-type"

gisb-acknowledgement-receipt
-----------------------------87453838942833
content-disposition: form-data; name="input-format"

error
-----------------------------87453838942833
content-disposition: form-data; name="input-data"; filename=”c:\temp\error.not”
content-type: multipart/report; report-type="gisb-error-notification";
boundary="NAESB7868"

--NAESB7868
content-type: text/html

<HTML><HEAD><TITLE>Error Notification</TITLE></HEAD> <BODY><P>
orig-from=123456789*
orig-to=234567890*
orig-input-format=X12*
resp-time-c=19960619102855*
resp-server-id=coolhost*
resp-trans-id=234423897*
request-status=EEDM601: Public Key Invalid*
comments=Please contact 1-800-555-1212 for correct public key*
</P> </BODY></HTML>

--NAESB7868
content-type: text/plain

orig-from=123456789*
orig-to=234567890*
orig-input-format=X12*
resp-time-c=19960619102855*
resp-server-id=coolhost*
resp-trans-id=234423897*
request-status=EEDM601: Public Key Invalid*
comments=Please contact 1-800-555-1212 for correct public key*
--NAESB7868--
-----------------------------87453838942833—

Signed Error Notification
```

```
content-type:multipart/signed; micalg=pgp-md5; protocol="application/pgp-signature";
boundary=--boundary2--200309090001

----boundary2--200309090001

content-type: multipart/report; report-type="gisb-error-notification";
boundary="NAESB7868"

--NAESB7868
content-type: text/html

<HTML><HEAD><TITLE>Error Notification</TITLE></HEAD> <BODY><P>
orig-from=123456789*
orig-to=234567890*
orig-input-format=X12*
resp-time-c=19960619102855*
resp-server-id=coolhost*
resp-trans-id=234423897*
request-status=EEDM601: Public Key Invalid*
comments=Please contact 1-800-555-1212 for correct public key*

</P> </BODY></HTML>

--NAESB7868
content-type: text/plain

orig-from=123456789*
orig-to=234567890*
orig-input-format=X12*
resp-time-c=19960619102855*
resp-server-id=coolhost*
resp-trans-id=234423897*
request-status=EEDM601: Public Key Invalid*
comments=Please contact 1-800-555-1212 for correct public key*

--NAESB7868--
----boundary2--200309090001

content-type: application/pgp-signature
-----BEGIN PGP MESSAGE-----

Version: 6.5

iQCVAwUBMJrRF2N9oWBghPDJAQE9UQQAtl7LuRVndBjrk4EqYBIb3h5QXIX/LC//JV5bNvkZIGPIcEmI5iFd9boEg
vpirHtIREEqLQRkYNoBActFBZmh9GC3C041WGquMbrbxc+nIs1TIKlA08rVi9ig/2Yh7LFrK5Ein57U/W72vgSxLh
e/zhdfolT9BrnHOxEa44b+EI=
=ndaj

-----END PGP MESSAGE-----

----boundary2--200309090001--
```

Pre-validation before Decryption

Proper trapping of the range of decryption process errors listed in Table A (Internet EDM Standard Error Messages and Codes) may require program code which is external to the decryption algorithm. Some versions of the PGP software do not explicitly discriminate between EEDM601, EEDM602, EEDM603, and EEDM699 type errors.

Under such a circumstance, files inbound to the decryption process should be preprocessed to trap the errors not identified by the PGP version being used. For example, searching the file for the text strings ‘BEGIN PGP MESSAGE’ and ‘END PGP MESSAGE’ can quickly identify ‘EEDM602 File not encrypted’ and ‘EEDM603 Encrypted file truncated’ type errors when the implemented PGP version only identifies decryption success, invalid Public Key (EEDM601), and decryption failure (EEDM699).

SECURITY

Internet ET security requirements include four primary security aspects: data Privacy, data Integrity, Authentication, and Non-repudiation (PAIN).

- Data privacy: unauthorized parties cannot decipher the content of the data.
- Authentication: the Receiver is certain of the identity of the Sender.
- Data integrity: unauthorized parties cannot modify or corrupt the data.
- Non-repudiation: the Sender cannot deny ownership of the transaction if it was sent with their digital signature.

In general, these needs are met by using the Basic Authentication capability of the Web server and the encryption and digital signature capability of the Open PGP and PGP security application for securing transactions.

Understanding OpenPGP and PGP

Pretty Good Privacy (PGP) is the name of the chosen security application. OpenPGP is the Internet Engineering Task Force (IETF) standard version of PGP that excludes all patented algorithms, allowing free commercial use of the standard. Both OpenPGP and PGP use a Public Key/Private Key pair to secure and sign files for transfer. The Private Key must be known only to the company that generated it. The Public Key counterpart is shared with trading partners.

Each company must generate its Public Key and Private Key pair. The RSA key generation algorithm should be chosen for versions of PGP ~~which~~that offer alternatives. Implementers of OpenPGP should choose DSA and El Gamal when creating their key pair. The Public Key~~s~~ should be distributed electronically to the company's trading partners. Private keys are not typically exchanged with trading partners. In the event that a Private Key needs to be exchanged, the exchange should occur in a secure manner such as postal or courier mail.~~using a secure method (e.g., courier mail) to the company's trading partners.~~

You must use the utmost care in protecting your Private Key. If an untrusted party has your Private Key, your security is compromised. It is recommended that a key size of 1024 be chosen when generating the key pair. This provides a significantly secure transaction.

When a company wishes to send transactions to its trading partner, it will use the partner's Public Key to encrypt the file. Encryption provides data privacy. Only the Private Key counterpart can decrypt this file.

When the sending party encrypts the file, it also uses its own Private Key to 'sign' the transaction. The receiving party can use the Sender's Public Key to verify the signature. The digital signature provides non-repudiation.

Encryption / Digital Signature

Encryption and digital signatures are applied to payload files before they are sent by the batch browser. The use of internal file or payload encryption such as X12.58 encryption is outside the scope of NAESB encryption standards but does not conflict with OpenPGP/PGP.

Encryption and digital signatures are created using OpenPGP, or on a mutually agreed basis, PGP version 2.6 or greater. Regardless of encrypting in a manual or automated fashion, it is essential that the correct Public Key of the trading partner be used to encrypt and just as essential that the correct Sender's own Private Key be used to digitally sign the file.

Digital signatures may also be applied, on a mutually-agreed-upon basis, to the HTTP Response by the Receiver of the package.

Decryption / Digital Signature Verification

After a package is received and processed by the Receiving Program, it is ready to be decrypted and have its digital signature verified. Given the correct userID for a trading partner, OpenPGP/PGP uses the appropriate key pair to encrypt, sign and decrypt. Upon request for signature verification, the OpenPGP/PGP will return a human-readable descriptive text such as DUNS number or company name.

When digital signatures are applied, on a mutually-agreed-upon basis, the HTTP Response received by the Sender of the transaction may be verified to ensure non-repudiation of receipt of the transaction.

Throughput Considerations

Encryption, digital signing, decryption and signature verification are all very CPU intensive. Companies anticipating large volumes of Internet ET traffic should research state-of-the-art techniques for scalability, including but not limited to:

- separating decryption and signature verification processing from web server receiving and processing
- passing secured or to-be-secured packages to a separate computer for security processing
- optimizing CPU and memory on security processing computers
- real-time or near real-time monitoring of website performance

Implementers of Internet ET sites should review and evaluate Domain Name Server (DNS) cache refresh intervals so as to ensure trading partner address changes are recognized on a timely basis.

Decryption and digital signature verification may not necessarily be processed by the Receiving Program prior to the 'gisb-acknowledgement-receipt' being sent to the Sender. As a result, the Sender may get an HTTP Response indicating a successful transfer but still not know if the file was successfully decrypted by the Receiver. Guidelines for communicating decryption errors found after the initial HTTP Response is sent are in Section 'Sending Error Notification Transactions' and Table A, 'Internet EDM Standard Error Codes and Messages'.

Security Requirements

Basic Authentication. Basic authentication, also known as realm one security, has been defined as one of the security standards for transmission on the Internet. The userid and password will be assigned by the server party according to site standards. The TPA must identify the userid and password for this security as well as procedures for changing the password, if applicable.

OpenPGP or PGP File Encryption. Payload files are encrypted using OpenPGP (IETF RFC 2440), or on a mutually agreed basis, PGP 2.6 or greater (using keys generated with the RSA algorithm). Free software implementations of the OpenPGP standard are available at http://www.gnupg.org/.

Firewall. A firewall should be deployed to protect HTTP servers.

CLIENT AND SERVER SPECIFICATIONS

Synchronization. Each Client and Server should be synchronized to a clock in the network of atomic clocks that is accessible via the Internet. The Client and Server should be synchronized as necessary to ensure synchronization with an atomic clock +/- 5 seconds. Please refer to Appendix A, ‘Time Synchronization’ for references on public sites for synchronization.

7 – TESTING GUIDELINES

NAESB INTERNET ET TEST GUIDELINES

Implementation of Internet ET requires testing to assure all parties are prepared to operate according to the Internet ET. This document focuses on testing standards for establishing Internet ET connectivity with a trading partner. Testing for transaction and other Quadrant-specific testing standards can be found in each Quadrant's QEDM.

Internet ET Connectivity testing standards may include:

- Connectivity test scripts. These scripts define the steps needed to adequately test connectivity.
- Technical Exchange Worksheet (TEW). This worksheet defines important operations parameters for a trading partner. The parameters include Internet ET URL's, contacts and other information. See Appendix C for an example TEW.

Common Internet ET errors include:

- Misspelled keywords (e.g. 'content-type'), or spacing in a keyword
- Header 'content-type' missing
- MIME boundary not correct
- Malformed MIME segments
- Content-length does not match actual length
- PGP MIME malformed (found with some versions of PGP)

GENERAL TESTING ASSUMPTIONS

The following assumptions apply to Internet ET testing:

- This document covers Internet ET testing. Transactions and business process test plans can be found in the QEDM.
- Testing may uncover problems. Problems found during testing should be expected.
- Testing is a basic demonstration of competency, and may not uncover all problems that may eventually require correction.
- In normal circumstances, trading partner to trading partner Internet ET connectivity testing takes approximately two weeks.

TESTING GOALS

The primary testing goals of this Internet ET ~~testing~~ are: ~~??LS align testing goals subbullets with common errors~~

- Establish Internet ET connectivity between trading partners including Internet connections and encryption compatibility.
- Validate Internet ET header formatting and delimiters
- Validate that normal production transaction files can be delivered.
- Validate that a large file (1MB or larger) can be delivered.
- Validate that Internet ET Receipts ('gisb-acknowledgement-receipt') are being delivered.
- Validate that protocol failures are handled properly.
- Validate that exchange failures are handled properly.

- Validate that encryption/decryption and digital signature failures are handled properly.

TEST EXECUTION

Test Scripts

Test scripts provide a step-by-step process for testing trading partner Internet ET connectivity. Test script scenarios test for both positive (accept) and negative (reject) results. Typical test scripts involve an exchange (Request and Response) of data between trading partners. with each TP confirming receipt of test file exchange via normal Internet ET standards. A copy of the payload file can be sent via e-mail for verification.

Test scripts can validate:

- That received files were not corrupted.
- Fail-over mechanisms by simulating a protocol failure and an exchange failure, triggering the appropriate notices to the TP contacts.
- Encryption failure processes by simulating an encryption/decryption failure, triggering the appropriate notices to the TP contacts.
- System time clock synchronization

Recommended Internal Tests

In addition to tests executed with trading partners, the following tests are recommended as internal tests of Internet ET systems.

- Acquire or develop an HTML page for interactive file upload (sample code is earlier in this document). Test the interactive file upload to your own server using an interactive browser.
- Stress Test. Ability to send and receive large production files (e.g. 10MB minimum uncompressed) and simultaneous usage. Simultaneous loading can be tested by requesting several other trading partners and/or several parties within your own company conduct Internet ET transfers concurrently.
- Fail-over Test. Test any processes triggered by a protocol or exchange failure by your trading partner.
- Invalid Userid/Passwords. Thoroughly test using the incorrect userid and password against the secure directory.
- Simulated Errors. Test various simulated errors in both file transfers and in OpenPGP or PGP decryption.

8 –APPENDICES

TABLE A – INTERNET ET STANDARD ERROR CODES AND MESSAGES

These errors and warnings are strictly related to problems found in the Receiving Program or decryption levels of processing before translation. Errors and warnings generated by the Client batch browser are assumed to be documented at the Client site to distinguish them from problems occurring in the Receiving Program or decryption. Numbering schemes and descriptions should aid in this distinction.

EEDM### standard error format with ### representing a numeric value; further processing will not take place

WEDM### standard warning format with ### representing a numeric value; further processing will take place

The string for the error or warning should appear in the following format:

[*Validation Code*]:[*Description*];[*supplemental message to be defined by the issuing site up to 80 characters*]

Internet ET Standard Error Codes and Messages

Validation Code	Description	Data Element	Data Element Required or. Mutually Agreed
EEDM100	Missing ‘from’ Common Code Identifier code	from	required
EEDM101	Missing ‘to’ Common Code Identifier	to	required
EEDM102	Missing input format	input-format	required
EEDM103	Missing data file	input-data	required
EEDM104	Missing transaction set	transaction-set	mutually agreed
EEDM105	Invalid ‘from’ Common Code Identifier	from	required
EEDM106	Invalid ‘to’ Common Code Identifier	to	required
EEDM107	Invalid input format	input-format	required
EEDM108	Invalid transaction set	transaction-set	mutually agreed
EEDM109	No parameters supplied	parameter string	required
EEDM110	Invalid ‘version’	version	required
EEDM111	Missing ‘version’	version	required
EEDM112	‘receipt-security-selection’ not mutually agreed	receipt-security-selection	mutually agreed
EEDM113	Invalid ‘receipt-security-selection’	receipt-security-selection	mutually agreed
EEDM114	Missing ‘receipt-disposition-to’	receipt-disposition-to	required
EEDM115	Invalid ‘receipt-disposition-to’	receipt-disposition-to	required
EEDM116	Missing ‘receipt-report-type’	receipt-report-type	required
EEDM117	Invalid ‘receipt-report-type’	receipt-report-type	required
EEDM118	Missing ‘receipt-security-selection’	receipt-security-selection	mutually agreed
EEDM119	Mutually agreed element, refnum, not present	refnum	mutually agreed
EEDM601	Public key invalid	file itself	required - security
EEDM602	File not encrypted	file itself	required - security
EEDM603	Encrypted file truncated	file itself	required - security
EEDM604	Encrypted file not signed or signature not matched	file itself	required - security
EEDM699	Decryption Error		required for general decryption errors not specifically identified by OpenPGP or PGP messages or exit codes
EEDM701	Sending party not associated with Receiving party		required
EEDM702	Package file format not recognized by Receiving party		required when the file format is not recognized by the receiver (e.g. not expecting 855 or not expecting Flat-File or XML)
EEDM703	Data set exchange not established for Trading Partner		required if the translator does not handle this exception
EEDM999	System error		required for general system errors to indicate severe errors in processing at the receiving site
WEDM100	Transaction set sent not mutually agreed	transaction-set	mutually agreed
WEDM102	‘receipt-security-selection’ not mutually agreed	receipt-security-selection	mutually agreed
WEDM103	Missing ‘receipt-security-selection’	receipt-security-selection	mutually agreed
WEDM104	Element refnum received, not mutually agreed; ignored	Refnum	mutually agreed

APPENDIX A - Reference Guide

Receiving Program

Receiving Programs can be written using Active Server Pages (ASP), Common Gateway Interface (CGI), Java Server Pages (JSP), Java Servlet technology, PHP and other technologies.

Information on ASP may be found on Microsoft's web site (www.microsoft.com).

A source on CGI is a book entitled 'Special Edition Using CGI' by Jeffrey Dwight and Michael Erwin.

Information on JSP and Servlet technology may be found at SUN's web site (http://java.sun.com).

Firewall Security

A source which covers this topic in detail is a book entitled 'Firewalls and Internet Security: Repelling the Wily Hacker' by William Cheswick and Steven Bellovin.

NAESB

NAESB Web Site: (www.naesb.org) Primary reference for energy industry standards.

HTTP

The NAESB Internet ET architecture is based on HTTP 1.1, and all implementations should be compatible with this version. All aspects of HTTP, HTML, and other Web-related topics are documented at: http://www.w3.org/pub/WWW/

General information regarding HTTP with basic terminology included are documented at:
http://www.w3.org/pub/WWW/Protocols/HTTP/1.1/spec.html

Syntax information for multipart can be found in IETF RFC1341 section 7.2. (www.ietf.org).

HTML

Information on HTML 4.0 may be found at http://www.w3.org/TR/REC-html40/.

http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

OpenPGP Software

The IETF OpenPGP standard is available at http://www.ietf.org/rfc/rfc2440.txt

Software implementations of the OpenPGP standard are freely available for commercial use from the Free Software Foundation at http://www.gnupg.org.

PGP Software

PGP is available for a variety of operating systems and platforms. For more information contact Network Associates (http://www.nai.com) or PGP Corporation (http://www.pgp.com)

Time Synchronization

Time synchronization is required to assure that all trading partners' transaction times are accurate. Testing has shown that the clocks on all computer systems drift. Time accuracy is dependent on how much a system's clock drifts, how frequently it is resynchronized and the accuracy of the source used for synchronization.

Each NAESB business process may have unique time-synchronization requirements. Refer to the QEDM for time-synchronization standards for target markets. Servers need to be time-synchronized according to the standards needed for the most-restrictive target market (i.e. smallest drift allowance).

Authoritative time synchronization is now being provided by governmental agencies around the world based on a synchronized network of atomic clocks. In the United States this includes the U. S. Naval Observatory and the National Institute of Standards and Technology.

An easy way to obtain the current time is from the U. S. Naval Observatory's Web site at tycho.usno.navy.mil/cgi-bin/timer.pl. The output from this page can easily be edited and reformatted to set a local system's time. Commercial, shareware and public domain packages are also available to synchronize system times, including IETF NTP, Internet daytime, nisttime / usnotime.

Further information on time synchronization may be found at the following Web sites:

- http://tycho.usno.navy.mil/ntp.html
- http://www.ccd.bnl.gov/xntp

APPENDIX B – FREQUENTLY ASKED QUESTIONS

Q1: How many times do I attempt to send an Internet ET package unsuccessfully before I notify my partner?......51
Q2: Do I send my 'gisb-acknowledgement-receipt' before or after I decrypt the Internet ET package? 51
Q3: What cryptographic algorithms should we use or not use?......51
Q4: Use of 'time-c-qualifier' across quadrants. We understand that the retail quadrants require the 'time-c-qualifier' for 'gisb-acknowledgement-receipt', while the WGQ does not require this data element. If we participate in multiple quadrants, which standard do we use?......52
Q5: NAESB EDM / AS2 Compatibility. What is the status of NAESB compatibility with AS2?..52
Q6: Atomic Clock Synchronization. How often do we need to synchronize our system clocks with an atomic clock?......52
Q7: Internet Continuous Connection. As an end user, do I need a continuously-connected internet Web server to participate in the Internet EDM in the energy industry, or can I just use a dial-up connection to my ISP and my favorite shrink-wrapped browser software?....52
Q8: Use of ANSI X12.58. If we use ANSI X12.58 encryption do we still need to use OpenPGP or PGP encryption? 52
Q9: What does NAESB recommend for the OpenPGP/PGP descriptive text?......52

Q1: How many times do I attempt to send an Internet ET package unsuccessfully before I notify my partner?

A: The Internet ET 'exchange failure' standard requires that you attempt to send a package at least three times over a 30- to 120-minute period. At minimum, this means 30 minutes has elapsed between your first failed attempt and your third failed attempt. At maximum, 120 minutes has elapsed between your first failed attempt and your third failed attempt. You should not wait longer than 120 minutes between your first failed attempt and your last failed attempt to notify your trading partner.

For example, if you make your first attempt at time 00:00:00, and your third attempt at time 00:30:00, your second attempt can occur any time between the first and third. If the third attempt fails, you have an 'exchange failure' and should notify your trading partner.

Q2: Do I send my 'gisb-acknowledgement-receipt' before or after I decrypt the Internet ET package?

A: Either. If you decrypt packages after you have sent the 'gisb-acknowledgement-receipt', errors found must be communicated to your trading partners using the Error Notification transaction. You should indicate in your TEW when you will decrypt packages.

Regardless of when you decrypt, the 'time-c' timestamp does not change. It is always the time the last byte was received by the Server from the Sender.

Q3: What cryptographic algorithms should we use or not use?

A: OpenPGP implementations should use DSA and El Gamal, and PGP implementations should use RSA.

Q4: Use of 'time-c-qualifier' across quadrants. We understand that the retail quadrants require the 'time-c-qualifier' for 'gisb-acknowledgement-receipt', while the WGQ does not require this data element. If we participate in multiple quadrants, which standard do we use?

A: You are required to follow the quadrant standards that govern the transaction or business process. For example, if you are executing a WGQ nomination, then you should adhere to WGQ standards, which do not require the 'time-c-qualifier'. If you are executing an REQ enrollment, you need to adhere to the REQ standards, which require 'time-c-qualifier'. Of course, all parties can mutually-agree to use the 'time-c-qualifier'.

Q5: NAESB EDM / AS2 Compatibility. What is the status of NAESB compatibility with AS2?

A: AS2 and NAESB EDM are no longer compatible. The GISB/NAESB EDM and AS2 standards were separated as of version 12 of AS2. The AS2 standard now supports the UCC profile, and not the GISB profile. At this time NAESB is not pursuing an IETF standard for the Internet ET.

Q6: Atomic Clock Synchronization. How often do we need to synchronize our system clocks with an atomic clock?

A: Systems should be synchronized as often as necessary to maintain the required +/- 5 second variance with the NIST atomic clock. Some business processes may require more stringent synchronization. Refer to quadrant standards for time-synchronization standards of business processes.

Q7: Internet Continuous Connection. As an end user, do I need a continuously-connected internet Web server to participate in the Internet EDM in the energy industry, or can I just use a dial-up connection to my ISP and my favorite shrink-wrapped browser software?

A: An interactive browser connection is not enough to actively participate in the system. Internet ET requires a Server with a permanent Internet connection capable of receiving files without operator intervention. This Server may exist at a service provider.

Q8: Use of ANSI X12.58. If we use ANSI X12.58 encryption do we still need to use OpenPGP or PGP encryption?

A: Yes. The use of encryption such as X12.58 on payload files is outside the scope of the NAESB encryption standards.

Q9: What does NAESB recommend for the OpenPGP/PGP descriptive text?

A: There are no Internet ET standards for the information provided in the OpenPGP/PGP descriptive text data element. Implementers are encouraged to use their company name in this data element.

APPENDIX C – SAMPLE TECHNICAL EXCHANGE WORKSHEET (TEW)

Company and Contact Information

Company info:	
Service Provider info (optional):	

Contacts	Business Contact	Technical Contact
Primary Name:		
Telephone:		
Fax:		
E-mail:		
Secondary Name:		
Telephone:		
Fax:		
E-mail:		

Transport Specifications	Test	Production
DUNS/DUNS+4 Number		
HTTP ‘to’ Value		
HTTP ‘from’ Value		
Using ‘time-c-qualifier’ in Receipt? (Y/N)		
Decryption After Receipt/Using Error Notification Transaction (Yes/No)		
Primary Internet ET URL		
Server Name:		
CGI Path:		
Port:		
Userid:		
Password:		
PGP Public Key	Distribution	Distribution
Finger Print	Distributed with Key	Distributed with Key
Userid (Alpha, spaces, numbers only; no special characters)		

CROSS-REFERENCE BETWEEN INTERNET ET TRANSPORT AND WGQ EDM VERSION 1.7

'**' denotes that actual language of the WGQ EDM standard differs from the language of the Internet ET standard. This cross-reference was prepared in March of 2004. It is intended to be a resource to help implementers find sections from the old WGQ EDM in the new Internet ET standard.

Internet ET Standard	WGQ EDM Standard	Internet ET Standard Narrative
0.1.1	0.1.1	An entity is a person or organization with sufficient legal standing to enter into a contract or arrangement with another such person or organization (as such legal standing may be determined by those parties) for the purpose of conducting and/or coordinating energy transactions.
0.1.2	0.1.2	There should be a unique entity common code for each entity name and there should be a unique entity name for each entity common code.
0.3.1	0.3.1	Entity common codes should be 'legal entities', that is, Ultimate Location, Headquarters Location, and/or Single Location (in Dun & Bradstreet Corporation ('D&B') terms). However, in the following situations, a Branch Location (in D&B terms) can also be an entity common code: 1) when contracting party provides a DUNS Number at the Branch Location level; OR 2) to accommodate accounting for an entity that is identified at the Branch Location level.
[10].1.1	4.1.2.	The Internet Electronic Transport (ET) does not pick winners, rather it should create an environment where the marketplace can dictate a winner or winners
[10].1.2	4.1.3.	Internet ET solutions should be cost effective, simple and economical
[10].1.3	4.1.4.	Internet ET solutions should provide for a seamless marketplace for energy
[10].1.4	4.1.6.	Parties should interface with third-party vendors according to NAESB Internet ET standards
[10].1.5	4.1.7.	Electronic communications between parties to the transaction should be done on a non-discriminatory basis, whether through an agent or directly with any party to the transaction
[10].1.6	4.1.12.	Protocols and tools that parties elect to support should be 'Internet-compatible'
[10].1.7	4.1.14.	The industry should use standard policies and guidelines for testing
[10].1.8	4.1.15.	The NAESB Internet ET should not set standards for site-level security. Individual organization security standards should be relied upon
[10].1.9	4.1.36.	Trading partners should maintain redundant connections to the public Internet for NAESB Internet ET Web sites. These redundant connections should be topographically diverse (duality of) paths to minimize the probability of a single point of failure
[10].1.10	4.1.39.	Trading Partners should mutually select and use a version of the NAESB Internet ET standards under which to operate, unless specified otherwise by government agencies. Trading Partners should also mutually agree to adopt later versions of the NAESB Internet ET standards, as needed, unless specified otherwise by government agencies
[10].2.1	4.2.20.	'Internet ET Testing'. Testing electronic packages between trading partners includes testing of: A) Connectivity; B) Encryption/Decryption; and C) Digital signatures where appropriate
[10].2.2	4.2.21**	'Fail-over' defines a prescribed process executed when a NAESB Internet ET Client fails to establish a connection to the target NAESB Internet ET Server
[10].2.3	4.2.22**	'Trading Partner' is a party that enters into an agreement with another party to transact business electronically using the Internet ET standard
[10].2.4	4.2.23**	'Originating party' is any party originating/creating the package. This could also include a third-party
[10].2.5	4.2.24**	'Third-Party' is any organization that a trading party uses to provide services to comply with the required elements of the Internet ET

Internet ET Standard	WGQ EDM Standard	Internet ET Standard Narrative
[10].2.6	4.2.25**	'Receiving Party' is any party that hosts (either in-house or outsourced) an Internet ET compliant server capable of receiving Internet ET packages
[10].2.7	4.2.25**	'Receiving Program' is a program or set of programs that process HTTP Requests from a Sender. The Receiving Program is responsible for generating the 'gisb-acknowledge-receipt', which includes any party that hosts (either in-house or outsourced) an Internet ET compliant server capable of receiving Internet ET packages
[10].2.8	4.2.26**	'Trading Partner Agreement', or 'TPA' is a legal agreement between trading parties. The TPA often dictates service level agreements and problem remediation processes. The TPA may include technical exchange information such as URLs, et cetera
[10].3.1	4.3.1**	All parties sending and receiving data should accept a TCP/IP connection
[10].3.2	4.3.4.	Trading partners should retain audit trail data for at least 24 months. This data retention requirement does not otherwise modify statutory, regulatory, or contractual record retention requirements
[10].3.3	4.3.7.	The designated Internet ET Server/Receiver site should be accessible via the public Internet. This does not preclude location of the designated site on a private intranet, as long as the designated site is also accessible via the public Internet
[10].3.4	4.3.8.	The minimum acceptable protocol should be HTTP. All sending and receiving parties should be capable of sending and receiving the HTTP versions supported by NAESB Internet ET
[10].3.5	4.3.9.	A timestamp designates the time a file is received at the Receiver's designated site. The timestamp consists of the 'time-c' data element, and in some cases the 'time-c-qualifier' data element. Refer to QEDM standards for use of the 'time-c-qualifier'
[10].3.6	4.3.9	The Receiver generates a timestamp upon the successful receipt of a complete file. The timestamp should be generated by the Receiving Program immediately, prior to further processing by the Receiving Program.
[10].3.7	4.3.9	After timestamp generation, the Receiver and sends an immediate HTTP Response to the Sender. The 'gisb-acknowledgement-receipt', which includes the timestamp data element(s), is the primary part of the HTTP Response.
[10].3.8	4.3.10**	The Server clock generating the timestamp should be synchronized with the National Institute of Standards and Technology (NIST) time in order to mitigate discrepancies between the clocks of the Sender and Receiver. Computer clocks should be synchronized as necessary to ensure at minimum +/- 5 second synchronization with an atomic clock. Specific business processes may have tighter synchronization requirements
[10].3.9	4.3.11**	The HTTP Response should be sent to the Internet Protocol (IP) address of the HTTP Request
[10].3.10	4.3.12.	At a minimum, one designated site for receipt should be identified for each trading partner. That site should be identified by a specific Uniform Resource Locator (URL). This does not preclude multiple designated sites being mutually agreed to between trading partners
[10].3.11	4.3.13.	The Sender should make three attempts to complete a unit of work. A unit of work consists of one complete HTTP POST transaction as defined in the technical specification of the HTTP protocol (IETF RFC 1945)
[10].3.14	4.3.14	The Internet ET roles for Sender and Receiver are defined in the following table. The entire table defines a unit of work:
[10].3.15	4.3.15	Trading partners should implement all security features (privacy, secure authentication, integrity, and non-repudiation) using a file-based approach via a commercially-available implementation of: A) An OpenPGP product as defined by IETF RFC 2440, or B) On a mutually agreed basis, PGP version 2.6 or greater using the RSA algorithm to generate keys
[10].3.16	4.3.15	Trading partners should implement basic authentication.
[10].3.17	4.3.15	Encryption keys should be self-certified. The exchange of keys should be done in a secure manner such as via postal mail. Key policies, including key exchange policies should be communicated to trading partners.

Internet ET Standard	WGQ EDM Standard	Internet ET Standard Narrative
[10].3.18	4.3.15	Encryption keys should have a limited lifetime whose duration is determined by the key's owner. A key's end of life is expressed in the expiration date field contained in each Public Key. A lifetime of one year or less is recommended.
[10].3.19	4.3.36.	Internet protocols should be used for accessing all industry business functions
[10].3.20	4.3.37.	Batch and Interactive Browsers should use Internet-compatible common browser software
[10].3.21	4.3.56**	Trading partners should use common codes for legal entities for the Internet ET 'to' and 'from' data elements
[10].3.22	4.3.64.	Private network connections to NAESB Internet ET servers, which include all NAESB Internet ET standardized Internet communication, may be at any point on a party's firewall boundary at the party's discretion on a non-discriminatory access basis. The specific type and speed of these connections should be mutually agreed. It is at the discretion of each party on how multiple private network connections should be managed, so long as such management is done on a non-discriminatory access basis
[10].3.23	4.3.70**	Parties should be limited to the NAESB Internet ET approved list of available TCP ports for Internet ET implementations
[10].3.24	4.3.71, 4.1.37	Internet ET implementations should not require any inbound ports to be opened on the Sender's firewall.
[10].3.25	4.3.88.	Internet ET Servers should use 128-bit Secure Socket Layer (SSL) encryption
7.3.50	7.3.50	The question is whether individual implementations are free to use HTTP HEAD command, prior to using the POST command to deliver the NAESB payload. When implementing a NAESB Internet ET solution, the standard clearly relies on the HTTP protocol spec for details of how to implement the protocol. It is also clear that the HTTP POST command should be used, and not the GET command.