



**XML REFERENCE MANUAL AND USER GUIDE FOR
DTD VERSION 5.0 /REV B**

Copyright © 2008 Network Solutions. All rights reserved.

NETWORK SOLUTIONS PROPRIETARY AND CONFIDENTIAL

This document includes trade secrets and/or proprietary commercial and/or financial information (collectively "Network Solutions Proprietary Information") belonging to Network Solutions (Network Solutions, LLC). Unauthorized disclosure and/or use of Network Solutions Proprietary Information without the express written consent of Network Solutions is strictly prohibited. This User's Guide may not be reproduced without the express written consent of Network Solutions.

DISCLAIMER AND LIMITATION OF LIABILITY

Network Solutions has made efforts to ensure the accuracy and completeness of the information in this document. However, Network Solutions makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. Network Solutions assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, Network Solutions assumes no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights of third parties. Nothing herein grants the reader any license to make, use, or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

Network Solutions reserves the right to make changes to any information herein without further notice.

TRADEMARKS

Network Solutions, and other trademarks, service marks and logos are registered or unregistered trademarks of Network Solutions and its subsidiaries in the United States and in foreign countries.

REVISION/CHANGE NOTICE

The following table tracks/lists revisions and changes to information presented in this document that have been implemented due to upgrades, corrections and modifications to the Network Solutions Partner Protocol.

Revision/Change	Date	Revision /Change Description
Rev A	01/08/2008	First version of the 5.0 documentation. Please see What's New document for changes between version 4.5 and version 5.0 of the Partner Protocol.
Rev B	08/25/2008	Second version of the 5.0 documentaion for new TLDs: AR.COM, GD, CX

RELATED/APPLICABLE DOCUMENTATION

The documents described in the following table provide information helpful to Partners/customers using XML to perform registration tasks associated with the Partner Protocol.

Title	Description
Wholesale XML FAQ	Provides various information regarding performance, product cycles and Registry specific information.
Network Solutions Partner Protocol DTD, Version 5.0, What's New	Outlines the new features and enhancements in this release since version 4.6 Rev D.

TABLE OF CONTENTS

RELATED/APPLICABLE DOCUMENTATION.....	ii
TABLE OF CONTENTS.....	iii
DEFINITIONS	x
LIST OF ACRONYMS AND ABBREVIATIONS	xii
1.0 INTRODUCTION	1
1.1 Extensible Markup Language (XML).....	1
1.2 Hypertext Transfer Protocol (HTTP)	1
1.2.1 Sending XML to the HTTPS Proxy Server	4
1.2.2 Special Characters in the XML Requests	4
1.2.3 XML Cardinalities	5
1.3 Network Solutions Partner Protocol	5
1.3.1 User Requests/Responses.....	7
1.3.2 Order Requests/Responses.....	7
1.3.3 Lookup Requests/Responses.....	9
1.3.4 UserLookup Requests/Responses	9
1.3.5 Partner Manager Requests/Responses	9
1.3.6 Available Requests/Responses.....	10
1.3.7 Order Status Requests/Order Responses.....	10
1.3.8 Email Forwarding Requests.....	10
1.3.9 Email Forwarding Lookup Requests/Responses	11
1.3.10 Security Requests/Responses.....	11
1.3.11 Authorized Agent.....	11
1.4 TLD Dependent Behavior	12
1.4.1 TLD Response Time	15
1.4.2 DNS and Name Server Configuration Requirements	15
1.4.3 Primary Name Server.....	16
1.4.4 Zone Serial Number	16
1.4.5 Zone Refresh Frequency	16
1.4.6 Zone Retry Frequency.....	16
1.4.7 Minimum TTL (Time To Live)	16

1.4.8	Expiration Frequency	17
1.4.9	Zone Name Servers and Registry Name Servers	17
1.4.10	MX Record Required	17
1.4.11	Name Servers on separate Class C Networks	17
1.5	Partner Preference and Profile Options	17
1.6	Aging Rules.....	18
2.0	XML COMMAND SEQUENCE.....	20
2.1	User Request (UserRequest.dtd).....	20
2.1.1	Create Request	20
2.1.1.1	CreateIndividual.....	20
2.1.1.2	CreateBusiness.....	23
2.1.2	Modify Request.....	26
2.1.2.1	ModifyIndividual	26
2.1.2.2	ModifyBusiness	27
2.2	Order Request (OrderRequest.dtd)	29
2.2.1	Create Request	31
2.2.1.1	CreateRegistration.....	31
2.2.1.2	CreatePrivateRegistration	42
2.2.1.3	CreateParking.....	44
2.2.1.4	Create WebForwarding.....	53
2.2.1.5	CreateHost.....	55
2.2.1.6	CreateSSL	57
2.2.2	Modify Request.....	68
2.2.2.1	ModifyRegistration.....	68
2.2.2.2	ModifyPrivateRegistration.....	71
2.2.2.3	ModifyParking	72
2.2.2.4	ModifyHost	76
2.2.3	Delete Request	78
2.2.3.1	DeleteRegistration.....	78
2.2.3.2	DeletePrivateRegistration	79
2.2.3.3	DeleteParking.....	81
2.2.3.4	DeleteHost.....	82
2.2.3.5	CascadeDeleteDomain.....	84
2.2.4	Renew Request.....	86
2.2.4.1	RenewRegistration.....	86

2.2.4.2	RenewPrivateRegistration.....	87
2.2.4.3	RenewParking.....	89
2.2.4.4	CascadeRenewDomain	91
2.2.4.5	RenewSSL.....	93
2.2.5	Convert Request.....	100
2.2.5.1	ConvertToRegistration.....	100
2.2.5.2	ConvertToParking.....	102
2.2.6	RNCA Request.....	104
2.2.6.1	TransferInternalToRegistration.....	104
2.2.6.2	TransferInternalToParking.....	107
2.2.6.3	ModifyCustomerName	109
2.2.7	CRA Request	110
2.2.7.1	InboundTransferToRegistration.....	111
2.2.7.2	InboundTransferToParking.....	116
2.2.8	RetrieveAuthCode Request.....	120
2.2.9	Secured Socket Layer Product Management (SSL).....	123
2.2.9.1	ReissueSSL	123
2.2.9.2	RevokeSSL	127
2.2.9.3	DownloadCertificate	129
2.2.9.4	GetSealCode	132
2.3	Lookup Request (LookupRequest.dtd).....	135
2.3.1	LookupHost.....	136
2.3.2	LookupDomain	137
2.4	Partner Manager Request (PartnerManagerRequest.dtd).....	142
2.4.1	FindAllCustomersForPartner	143
2.4.2	FindAllDomainsForCustomer.....	144
2.4.3	FindAllDomainsForPartner.....	146
2.4.4	FindDomainsWithPrivateRegistration	148
2.4.5	FindDomainsWithWebForwarding.....	149
2.4.6	ConsolidateAllCustomerProducts.....	150
2.4.7	ConsolidateSelectedCustomerProducts	152
2.4.8	GetPartnerData.....	153
2.4.9	FindAllSSLProductsForPartner	154
2.4.10	FindSSLProductsForPartner	156
2.4.11	FindDetailedSSLProduct	158
2.5	Available Request (AvailableRequest.dtd).....	160
2.5.1	VerifyDomainName.....	161

2.5.2	VerifyDomains.....	162
2.5.3	VerifyDomainsAdvanced	163
2.5.4	VerifyDomainsAll.....	164
2.5.5	GenerateDomains.....	169
2.5.6	GenerateDomainsAdvanced	171
2.5.7	GenerateDomainsAll.....	174
2.5.8	IsDomainCandidateForInboundTransfer	178
2.5.9	IsDomainCandidateForInternalTransfer	179
2.6	Order Status Request (OrderStatusRequest.dtd).....	180
2.6.1	Order	180
2.7	User Lookup Request (UserLookupRequest.dtd).....	184
2.7.1	LookupUser.....	184
2.7.2	CheckLoginNameAvailability	186
2.8	Email Forwarding Request (EmailForwardingRequest.dtd)	187
2.8.1	CreateEmailForwardingAndDomain	187
2.8.1.1	CreateRegistration.....	188
2.8.1.2	CreateParking.....	190
2.8.2	ModifyEmailForwarding	193
2.8.3	DeleteEmailForwardingAndDomain	194
2.8.4	RenewEmailForwardingAndDomain	196
2.9	Email Forwarding Lookup Request (EmailForwardingLookupRequest.dtd).....	198
2.9.1	LookupEmailForwarding.....	198
2.10	Security Request (SecurityRequest.dtd)	200
2.10.1	AuthenticatePartner.....	200
2.10.2	AuthenticatePartnerAndGetPartnerInfo.....	201
2.10.3	AuthenticateCustomer.....	202
2.10.4	AuthenticateCustomerAndGetCustomerInfo.....	203
2.10.5	AuthorizePartner	204
2.10.6	AuthorizeCustomer	205
3.0	RESPONSES	207
3.1	User Response (UserResponse.dtd)	207
3.2	Order Response (OrderResponse.dtd).....	207
3.3	Lookup Response (LookupResponse.dtd)	209

3.3.1	Lookup Response for Domain	209
3.3.2	Lookup Response for Host.....	210
3.4	Partner Manager Response (PartnerManagerResponse.dtd)	210
3.4.1	PartnerManager Responses for FindAllDomainsForCustomer	211
3.4.2	PartnerManager Response for FindAllCustomersForPartner	212
3.4.3	PartnerManager Responses for GetPartnerData	212
3.4.4	PartnerManager Responses for FindAllSSLProductsForPartner and FindSSLProductsForPartner	213
3.4.5	PartnerManager Response for FindDetailedSSLProduct.....	214
3.5	Available Response (AvailableResponse.dtd).....	215
3.6	Order Status Response	216
3.7	User Lookup Response (UserLookupResponse.dtd)	216
3.8	Email Forwarding Response.....	217
3.9	Email Forwarding Lookup Response (EmailForwardingLookupResponse.dtd) ...	217
3.10	Security Response (SecurityResponse.dtd).....	218
4.0	ERROR HANDLING.....	218
4.1	Malformed XML.....	218
4.2	Validation Errors.....	220
4.3	Data Errors.....	226
4.4	XML Status Codes	227
5.0	USER TROUBLESHOOTING	227
5.1	Connectivity.....	228
5.1.1	Ping	228
5.1.2	Telnet	228
5.1.3	Submit Request.....	229
6.0	HELP DESK INFORMATION.....	230
Appendix A	Document Type Definitions (DTD)s	231
A.1	AvailableRequest DTD	231

A.2	AvailableResponse DTD	231
A.3	Common Declarations DTD	233
A.4	DomainData DTD	235
A.5	DomainProduct DTD	236
A.6	EmailForwardingLookupRequest DTD	240
A.7	EmailForwardingLookupResponse DTD	241
A.8	EmailForwardingRequest DTD	242
A.9	ErrorResponse DTD	243
A.10	HostProduct DTD	243
A.11	LookupRequest DTD	245
A.12	LookupResponse DTD	246
A.13	OrderRequest DTD	246
A.14	OrderResponse DTD	247
A.15	OrderStatusRequest DTD	248
A.16	PartnerManagerRequest DTD	249
A.17	PartnerManagerResponse DTD	250
A.18	PrivateRegistrationProduct DTD	251
A.19	RequestHeader DTD	251
A.20	ResponseHeader DTD	252
A.21	SecurityRequest DTD	252
A.22	SecurityResponse DTD	253
A.23	UserLookupRequest DTD	253
A.24	UserLookupResponse DTD	254
A.25	UserRequest DTD	255
A.26	UserResponse DTD	257

A.27 WebForwardingProduct DTD..... 258

A.28 SSLProduct DTD 258

Appendix B XML Status Codes..... 260

B.1 UserRequest DTD..... 260

B.2 LookupUserRequest.DTD 264

B.3 SecurityRequest.DTD 265

B.4 AvailableRequest.DTD 269

B.5 OrderRequest.DTD..... 275

B.7 EmailForwardingRequest.DTD 316

B.8 EmailForwardingLookupRequest.DTD..... 323

B.9 PartnerManagerRequest.DTD..... 324

B.10 XML Status Codes 329

 B.10.1 Status Codes 329

 B.10.2 Error Codes 330

 B.10.3 Validation Codes..... 332

Appendix C Country Codes 334

Appendix D XML Default Field Validation Rules 338

Appendix E Common Data Elements 343

Appendix F Server Selection Software for Secured Socket Layer Products .. 349

DEFINITIONS

Advance DNS: With a Parking a domain registration the Partner has extensive control over the DNS record sets (i.e. A records, MX records, and CNAME records). The Partner can set the admin contact and Network Solutions always serves as the technical contact.

A Record: Address record that provides name-to-address mapping.

AuthCode: A domain name password.

Authorized Agent: A user who is allowed access to the Network Solutions Partner Protocol. Currently, the Authorized Agent is known as a System User and may be either a Partner or a Customer.

BulkWhoisOptOut: When creating a product, the customer can choose to have the domain name information removed from any bulk listings of domain names and contacts. By default, the domain name information will be in the listings.

Business: A company or organization. A business account is created using the company's name. Legal contact information is also required for a business account. Network Solutions will contact the Business's Legal Contact in any communications for this domain.

CNAME Record: Canonical name (for aliases).

Customer: Either a business or an individual who registers domains or purchases other Network Solutions services via a Partner's web page.

Domain Protect: An indicator on a domain that prevents the domain from being transferred from Network Solutions. To transfer a domain, the Domain Protect indicator must be removed.

Host: Also called a name server. A computer that has both the software and the data (zone files) needed to resolve domain names to Internet Protocol (IP) numbers.

Host Coordinator: A user who has the authority to change the IP address information for a host.

Host ID: A Network Solutions NIC Handle that identifies a host (name server).

Individual: A single person not associated with a company or business. An individual account is created using the first and last name of the individual.

IP Address: A unique, numeric identifier used to specify hosts and networks. Internet Protocol (IP) numbers are part of a global, standardized scheme for identifying machines that are connected to the Internet. Technically speaking, IP numbers are 32 bit addresses that consist of four octets, and they are expressed as four numbers between 0 and 255, separated by periods

Inbound Transfer: The process of moving a domain name to Network Solutions from another Registrar. An Inbound Transfer is requested by a Customer for the movement of a domain (or domains) that currently reside with a non-Network Solutions Registrar to Network Solutions.

Internal Transfer: An Internal Transfer is the transfer of a customer account within Network Solutions. The Types of Internal Transfers are: Retail to Retail, Retail to Partner, Partner to Retail, Partner to Partner and within Partner.

Legal Lock: The state in which a domain is set that prohibits any modification action from completing. The Legal Lock state must be removed before the Transfer and Modify actions will work. Renewals will still be processed for domains on Legal Lock. Only Network Solutions can place a domain on Legal Lock, Partners and customers cannot request that a domain be placed on Legal Lock.

MX Record: The mail exchanger for a domain. This should be a host that will either process or forward mail for the domain.

NIC Handle: A NIC Handle is a unique identifier, which can be up to 10 alpha-numeric characters, assigned to each domain name record, contact record, and network record in the Network Solutions domain name database.

Parking: "Parking" a domain allows the Partner to essentially reserve the domain for future use or sale. Network Solutions manages these domains through internally operated DNS name servers. The Partner is not able to directly manipulate the DNS entries for this type of domain. The Partner can set the admin contact and Network Solutions always serves as the technical contact.

Partner: The company contractually associated with Network Solutions to provide domain name registration services using the Network Solutions Partner Protocol.

Private Registration: An additional service associated with a domain name registration that allows the domain holder's contact information to be masked.

Product ID: A unique identifier for any product available through the Network Solutions Partner Protocol.

Registration: The process through which individuals and organizations obtain a domain name. Registration of a domain name enables the individual or organization to use that particular domain name for a specified period of time, provided certain conditions are met and payment for services is made. A Registration product requires the user to specify the hosts in order for the domain to properly resolve. If no hosts are specified on a registration, the domain will not resolve.

System User: Either a Partner or Customer that acts as an Authorized Agent.

User: A “user” in the context of this document can serve dual roles. One role is as the account owner and the other role is as a contact for a domain. For instance, a user can be the account owner (i.e. a user who has acquired one or more products). However, a user can also serve as the technical contact for a domain. The key to this concept is really the notion of who purchased the product. The user that owns the product is the account owner.

User ID: The unique number that is associated with a user.

Web Forwarding: Web Forwarding has the advantage of making multiple sites look like a single site. You can register misspellings, alternate extensions (e.g., .biz, .net, etc.) and/or abbreviations and then forward them to your primary Web site.

LIST OF ACRONYMS AND ABBREVIATIONS

CGI	Common Gateway Interface
CRA	Change of Registrar Agreement
DTD	Document Type Definition
GNU	GNU's Not Unix
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol through Secure Socket Layer
LWP	Light Weight Protocol
SGML	Standard Generalized Markup Language
SSL	Secure Socket Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TLD	Top Level Domain
URI	Universal Resource Identifier
URL	Universal Resource Locator
URN	Uniform Resource Name
NSPP	Network Solutions Partner Protocol
XML	Extensible Markup Language
XSL	Extensible Style Language
XQL	XML Query Language

1.0 INTRODUCTION

This document is intended to serve as a reference manual and user's guide for Network Solutions Partners engaged in using Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) to perform legitimate tasks within authorized boundaries established for Network Solutions Partner Protocol. This document is intended to speak to the developer who will create an interface that utilizes the Partner Protocol to create new customers, register domains and purchase other Network Solutions products, as well as manage existing customers and domains. The document is not intended to teach the XML language or the HTTP Protocol as it is expected that the user is already familiar with its basic uses and applications. Brief descriptions of XML, HTTP, and Partner Protocol, along with a brief description of international (multilingual) domains, are contained in the following subparagraphs.

1.1 Extensible Markup Language (XML)

XML is a document description language, much like the Hypertext Markup Language (HTML) that is used to construct web pages. XML, however, is much more versatile than HTML and as such has profound implications on how we view what the web is and what it can do.

HTML has a set of predefined tags that describe components of a document and how a browser should present that information. XML shares many of the same characteristics of HTML and looks very similar; however, it is far more flexible. Using either Document Type Definitions (DTDs) or an XML-Schema, it is possible to define new tags that are useful for a wide variety of applications. These applications can range from an information presentation format like HTML, to an electronic business application. XML is a rapidly growing technology with many applications being created every day.

Network Solutions has developed a set of DTDs that define tags and a data format that can be used to purchase and manage a number of products and services. The Network Solutions Partner Protocol allows Partners to communicate with Network Solutions to purchase and manage products for themselves or their customers. Using XML, Partners are able to create a website for their customers and have it interact with the Network Solutions fulfillment system.

For more information on XML the reader is referred to the website <http://xml.com>. Additional useful information concerning specifications and the standardization process is available at the site <http://www.w3c.org/xml>, which is the official site for XML development.

1.2 Hypertext Transfer Protocol (HTTP)

Network Solutions designed the XML infrastructure as a client-server architecture. The interaction between the client and the server is very simple. The client, a Network Solutions Partner application, initiates a connection with the server, the Network Solutions XML processing engine, for the purpose of executing a transaction. The Partner Protocol is the XML that passes between client and the server. Each Partner Protocol transaction consists of a request/response pair. A client sends the Partner Protocol request to the XML engine via the HTTP and then waits for a Partner Protocol response on the same connection. The connection terminates when the client receives the Partner Protocol response.

HTTP is a protocol built on top of the Transmission Control Protocol (TCP) with the lightness and speed necessary for a distributed collaborative hypermedia information system. HTTP is basically a stateless protocol.

Each HTTP communication session consists of four basic transaction components:

- **Connection** A connection by the client to the server. Port 80 is often used for HTTP communication, but other non-reserved ports may be specified in the URL. The Partner Protocol utilizes port 8010 and 8020; the Network Solutions designated ports for secure HTTP traffic (i.e. HTTPS).
- **Request** A client's request message to the server.
- **Response** A server response message to the client.
- **Close** Closing the connection by either/or both parties.

Figure 1 illustrates the four transaction components of a typical client server transaction.

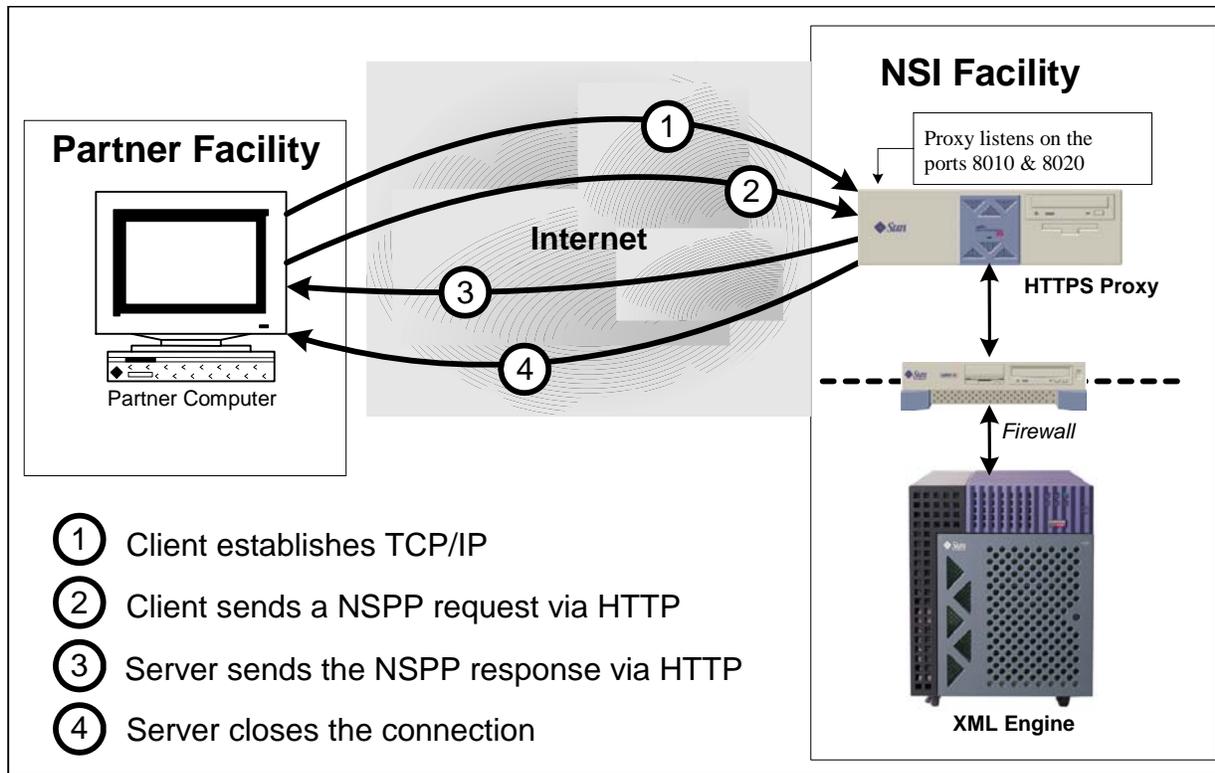


Figure 1. Typical Partner Protocol Client Server Transaction.

After establishing a normal TCP connection, the HTTP protocol provides some simple header information that tells the receiving system what to expect and how to respond. The header information includes:

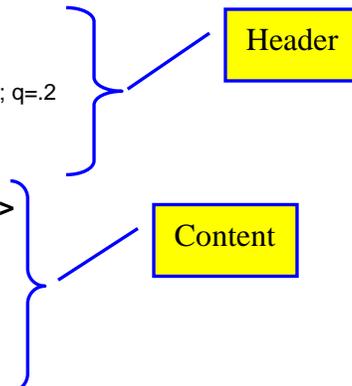
- The type of HTTP Request (POST or GET),
- The HTTP protocol version,
- The file/command it was trying to access,
- What types of responses it accepts,
- What type of content will follow (must be text/xml),
- The total length of all content, and
- The content of the request itself.

The following sample HTTP request illustrates the actual content that a client sends to the Network Solutions server to accomplish a task (in this case a name availability request). This example includes the header information section followed by the respective content information section.

Example - HTTP Request

```
POST/invoke/vpp/AvailabilityService HTTP/1.0
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Content-type: text/xml
Content-length: 265
```

```
<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
```



```
<PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <VerifyDomainName>example.com</VerifyDomainName>
</Body>
</AvailableRequest>
```

The response from the HTTP server also contains some header information:

- The protocol version,
- The HTTP status (Code and Description) of the request/response,
- Possible cookie information for session tracking,
- The content type of the response,
- The length of the response, and
- The response itself.

Each request is followed by a corresponding response back to the client that issued the request. The HTTP Response shown below illustrates a typical response that a client could expect after issuing the previous request. This example includes the header information section followed by the respective content information section.

Example – HTTP Response

```
HTTP/1.1 200 OK
Set-Cookie: ssid=136455371555550; path=/;
Content-Type: text/xml; charset=UTF8
Connection: Close
Content-Length: 524
```

Header

```
<?xml version="1.0"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2006-08-01T17:24:13EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <NotAvailable>
      <DomainName>example.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </NotAvailable>
  </Body>
</AvailableResponse>
```

Content

HTTP can be passed over a Secure Socket Layer (SSL) connection to ensure security and privacy. HTTP sent via SSL is commonly referred to as HTTPS, although technically this is not an official standard.

By requiring that Partner Protocol transmissions utilize SSL, Network Solutions ensures that all Partner Protocol communication across the open Internet is protected and secure. Previous paragraphs indicated that all communication between client and server is done via HTTP. However, a more accurate statement is that all communication relies on HTTPS to send and receive the XML data. In fact non-SSL connections are refused by the proxy server shown in Figure 1.

In order to harness the capabilities of HTTPS, the Partner must have an HTTPS client. Various HTTPS clients can be found on the Internet. For Java, SUN has provided the reference implementation of their HTTPS connection at:

<http://java.sun.com/products/jsse/index.html>. For Perl, the Light Weight Protocol (LWP) module allows programs to use HTTP and HTTPS. Information about LWP is available at: <http://www.cpan.org>.

For further information about HTTP & HTTPS the reader is encouraged to use the World Wide Web as a resource and/or to read the book *TCP/IP Illustrated* by W. Richard Stevens.

1.2.1 Sending XML to the HTTPS Proxy Server

The HTTPS proxy server shown in Figure 1 listens on two ports: 8010 and 8020. The design goal behind the two-port strategy is to provide the customer with optimal performance during times of heavy name availability activity. The Partner must send all name availability requests to port 8010. All other requests must be sent to port 8020. Sending a command to the wrong port results in a failure response. Using the two-port strategy, Network Solutions is able to dedicate processing resources to non-availability requests so that the Partner response is not disproportionately affected by a flood of availability requests.

When sending XML to the Partner Protocol server, the Partner should set the path component of the destination URLs to the following values:

Port 8010: /invoke/PartnerProtocol/AvailabilityService

Port 8020: /invoke/PartnerProtocol/TransactionService

Sending a command to the wrong path or to the right path on the wrong port will result in a failure response. Table 3 defines which port each type of request should be sent to. The Partner needs to use the path that corresponds with the designated port in order for the request to be submitted successfully.

Note: All requests sent to either port 8010 or port 8020 must be specified as *Content-Type: text/xml*.

1.2.2 Special Characters in the XML Requests

Certain characters such as the ampersand, "&", receive special treatment in XML. In order to send an ampersand to the XML engine the Partner must actually send the character sequence "&";. Table 1 defines the complete list of special characters in XML:

Table 1. XML Special Characters.

Desired Character	Corresponding XML Character sequence
&	&
<	<
>	>
'	'
"	"

1.2.3 XML Cardinalities

Cardinalities in XML define the number of occurrences of the preceding term. There are three common modifiers used in XML in order to express cardinality, they are: ? * +. Table 2 defines these cardinalities:

Table 2. XML Cardinalities.

Modifier	Meaning
?	Zero or one of the preceding term (i.e. optional).
+	One or more of the preceding term.
*	Zero or more of the preceding term.

1.3 Network Solutions Partner Protocol

From a high level perspective, the Partner Protocol is the "electronic language" used to communicate with the Partners. The Partner Protocol is built on top of the XML, and is defined in a set of DTDs that can be found in Appendix A.

As illustrated in Figures 2 through 8 there are less than a dozen types of requests the Partner can transmit to Network Solutions. For each type of request there is a corresponding format for a response from Network Solutions. Table 3 describes these DTDs and the types of requests that can be accomplished with each. The subparagraphs following the table provide additional details about the XML content that corresponds to the DTDs. Section 2.0 defines in greater detail each user request, all the required and optional fields in the request, and examples of each request and expected response. Appendix D contains the validation rules and sample values for each of the fields defined in Section 2.0.

Table 3. DTD User Actions and Descriptions.

DTD	User Actions	Description	Port
UserRequest UserResponse	CreateIndividual, CreateBusiness, ModifyIndividual, ModifyBusiness	These DTDs define the XML structure for the request/response a Partner uses to modify accounts, admin contacts, and tech contacts.	8020

DTD	User Actions	Description	Port
OrderRequest OrderResponse	CreateRegistration, ModifyRegistration, ConvertToRegistration, DeleteRegistration, RenewRegistration, CreatePrivateRegistration, RenewPrivateRegistration, DeletePrivateRegistration, ModifyPrivateRegistration, TransferInternalToRegistration, InboundTransferToRegistration, CreateParking, DeleteParking, RenewParking, ModifyParking, TransferInternalToParking, InboundTransferToParking, ConvertToParking, CreateWebForwarding, ModifyWebForwarding, DeleteWebForwarding, RenewWebForwarding, CascadeDeleteDomain, CascadeRenewDomain, RenewDomainsOnly, CreateHost, ModifyHost, DeleteHost, ModifyCustomerName, RetrieveAuthCode.	These DTDs define the XML structure for purchasing and managing products.	802 0
OrderStatusRequest	Order	These DTDs define the XML used to determine the progress of a queued order. The response is an OrderResponse.	802 0
LookupRequest LookupResponse	LookupDomain and LookupHost.	These DTDs define the XML structure for retrieving information about domains or hosts in the Network Solutions system.	802 0
UserLookupRequest UserLookupResponse	LookupUser and CheckLoginNameAvailability.	These DTDs define the XML structure for retrieving information about a user in the Network Solutions system.	802 0
PartnerManagerRequest PartnerManagerResponse	FindAllCustomersForPartner, FindAllDomainsForCustomer, FindAllDomainsForPartner, FindAllDomainsWithPrivateRegistration, FindAllDomainsWithWebForwarding, ConsolidateAllCustomerProducts, ConsolidateSelectedCustomerProducts, and GetPartnerData.	These DTDs define the XML structure that allows the Partner to query for lists of domains or users that they, or their customers, control.	802 0
AvailableRequest AvailableResponse	VerifyDomainName, VerifyDomains, GenerateDomains, VerifyDomainsAdvanced, GenerateDomainsAdvanced, VerifyDomainsAll, GenerateDomainsAll, IsDomainCandidateForInboundTransfer, and IsDomainCandidateForInternalTransfer.	These DTDs define the XML structure for checking if a domain or domains are available for purchase as well as generate alternative domains.	801 0

DTD	User Actions	Description	Port
EmailForwardingRequest	CreateEmailForwardingAndDomain, ModifyEmailForwarding, DeleteEmailForwardingAndDomain, and RenewEmailForwardingAndDomain.	These DTDs define the XML used to create and modify the email forwarding for name domains.	8020
EmailForwardingLookupRequest EmailForwardingLookupResponse	LookupEmailForwarding	These DTDs define the XML structure for retrieving information about email forwarding for .name domains.	8020
SecurityRequest SecurityResponse	AuthenticatePartner, AuthenticatePartnerAndGetPartnerInfo, AuthenticateCustomer, AuthenticateCustomerAndGetCustomerInfo, AuthorizePartner, and AuthorizeCustomer.	These DTDs define the XML structure for authenticating and authorizing Partners and customers.	8020

1.3.1 User Requests/Responses

At this point it is worth reiterating the relationship between DTDs and XML content. The DTDs describe the rules for valid XML content. The XML content is the actual data sent between the client and the server across the HTTPS connection. Stated another way, the DTDs define the Partner Protocol and the XML content is an actual instance of the Partner Protocol.

As illustrated in Figure 2, a Partner wanting to create and/or modify users in the Network Solutions database must send an XML message corresponding to the UserRequest DTD, and the response will be a message adhering to the UserResponse DTD. The UserRequest DTD utilizes element definitions also found in the RequestHeader and CommonDeclarations DTDs. A user can be an individual or a business. The only real difference between an individual user and a business user is the data fields used to describe them. For instance, an individual user has a first name, middle name, and last name, whereas a business has a company name followed by a legal contact name. Sections 2.1.1 and 2.1.1.2 illustrate this concept in greater detail.

Each newly created user is assigned to the Partner that created the user. A user can own domains, or be the technical or admin contact for one or more domains. All these requests are completed synchronously (i.e. the response contains the requested data or an explanation of the failure).

Note: A "user" in the context of this document can serve multiple roles. One role is as the account owner and the other role is as a contact for a domain/host. For instance, a user can be the account owner (i.e. a user who has acquired one or more products). However, a user can also serve as the technical contact for a domain. The key to this concept is really the notion of who purchased the product. The user that owns the product is the account owner.

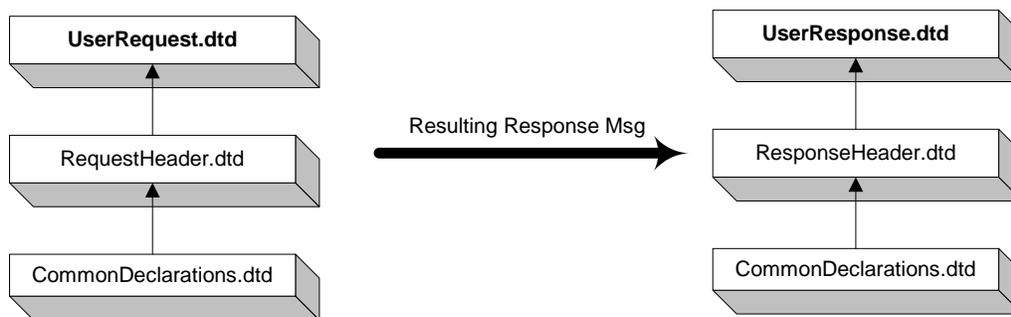


Figure 2. UserRequest and UserResponse DTD Dependencies.

1.3.2 Order Requests/Responses

A Partner wanting to create and/or manipulate Network Solutions products must send an XML message corresponding to the OrderRequest DTD and will receive a response defined by the OrderResponse DTD. The relationship between these DTDs and their dependencies are

illustrated in Figure 3. The OrderRequest DTD depends on element declarations defined in DomainProduct, HostProduct, CommonDeclarations, and the RequestHeader DTDs.

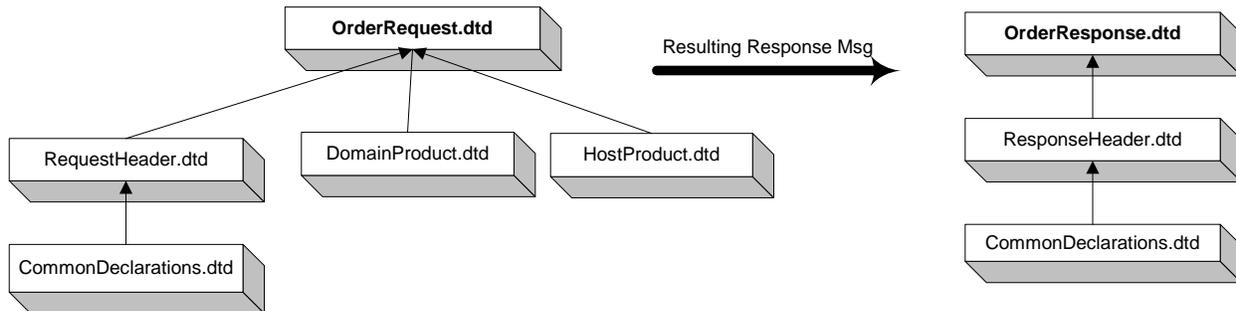


Figure 3. OrderRequest and OrderResponse DTD Dependencies.

As defined in the DomainProduct DTD there are two types of domain products that Network Solutions offers to the Partners: Registered domains and Parked domains. Table 4 explains the characteristics and differences of each domain type.

Table 4. Domain Product Descriptions.

Domain Type	Description
Registered Domain	This type of domain requires the Partner to supply his or her own name servers. All Domain Name System (DNS) manipulation is done on the specified servers. Network Solutions may have direct access to the specified name servers, but most often they are controlled by the Partner themselves or the Partner's end customer. The Partner can set the admin and technical contact.
Parked Domain	This type of domain allows the Partner to essentially reserve the domain for future use or sale. Network Solutions manages these domains through internally operated DNS name servers. The Partner is not able to directly manipulate the DNS entries for this type of domain. The Partner can set the admin contact. Network Solutions always serves as the technical contact for this type of domain.

Note: The default Billing Contact for all domain products, regardless of the type, is the Partner who acquired the domain.

Network Solutions handles OrderRequests differently than UserRequests. OrderRequests are handled asynchronously (in other words the request is queued for a short period of time while back-office processing is completed). The queue strategy provides a consistent user experience across different domain Registries and under heavy load. When a Partner sends XML corresponding to an OrderRequest, Network Solutions performs the following actions:

1. For domain purchases, Network Solutions immediately reserves the domain name in the appropriate Registry for TLDs whose registry type is a thin registry. For TLDs that have a thick registry type, Network Solutions queues the order and registers the domain at the earliest practical moment, given registry availability and system load. The DomainReservationStatus field in the order response indicates if the domain has been reserved or queued. See Section 3.2 for details.

Note: Please see section 1.4 for more information on explanation of thin and thick registry types, and how each Registry processes orders.

2. Network Solutions places the remaining "Back-office" processing (billing, DB modifications, etc) in a process queue (which is processed in a FIFO basis).
3. Network Solutions sends a response back to the client detailing the status of the order.

The order is usually submitted in 4-8 seconds and completed within 1 to 2 minutes. This approach speeds the response time to the Partner and allows Network Solutions to take orders when portions of the infrastructure (both internal and external) are down for maintenance or other reasons.

It also is important to note that in some cases, there is TLD specific behavior. For example, some TLDs require minimum registration terms. In such cases, if the minimum registration terms are not met, the order will not succeed in the Partner Protocol. Details regarding this type of TLD specific behavior can be found in section 1.4.

Note: Please review section 1.4 for more information regarding TLD specific behavior such as minimum and maximum registration terms.

1.3.3 Lookup Requests/Responses

XML corresponding to the LookupRequest and LookupResponse DTDs return information concerning products or hosts. Partners can retrieve information about any of their users' domains, hosts, or products that they created or purchased. As illustrated in Figure 4, the LookupRequest DTD depends on element declarations defined in the RequestHeader, CommonDeclarations, DomainProduct, and HostProduct DTDs.

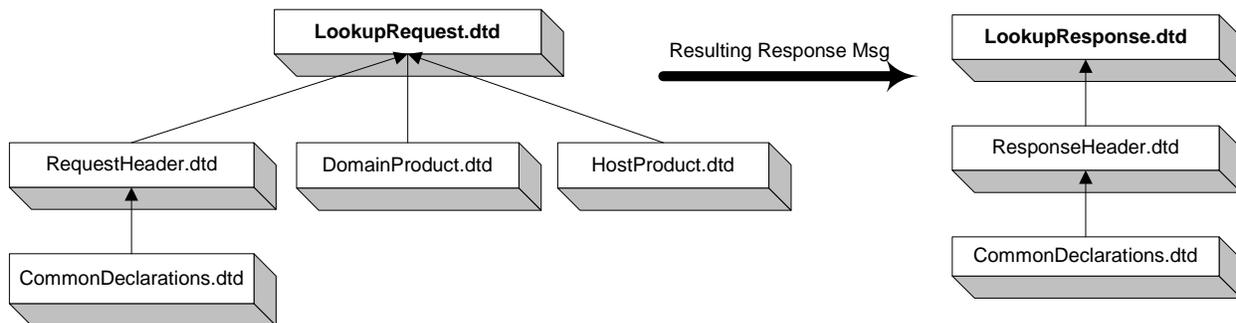


Figure 4. LookupRequest and LookupResponse DTD Dependencies.

1.3.4 UserLookup Requests/Responses

XML corresponding to the UserLookupRequest and UserLookupResponse DTDs return information concerning users. Partners can retrieve information about any of the users that they created. As illustrated in Figure 5, the UserLookupRequest DTD depends on elements found in the RequestHeader and CommonDeclarations DTDs.

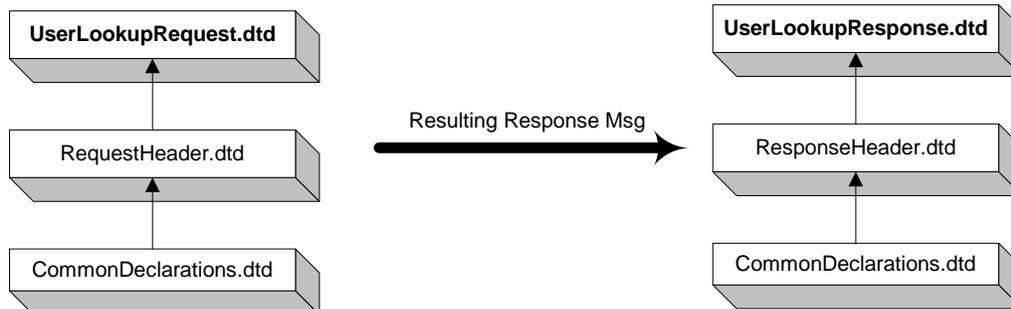


Figure 5. UserLookupRequest and UserLookupResponse DTD Dependencies.

1.3.5 Partner Manager Requests/Responses

XML corresponding to the PartnerManagerRequest and PartnerManagerResponse DTDs allow the Partner to do limited, real-time domain and customer queries. These queries could take some time to complete depending on the number of customers and domains purchased for the Partner. As illustrated in Figure 6, the PartnerManagerRequest DTD depends on elements found in the RequestHeader and CommonDeclarations DTDs.

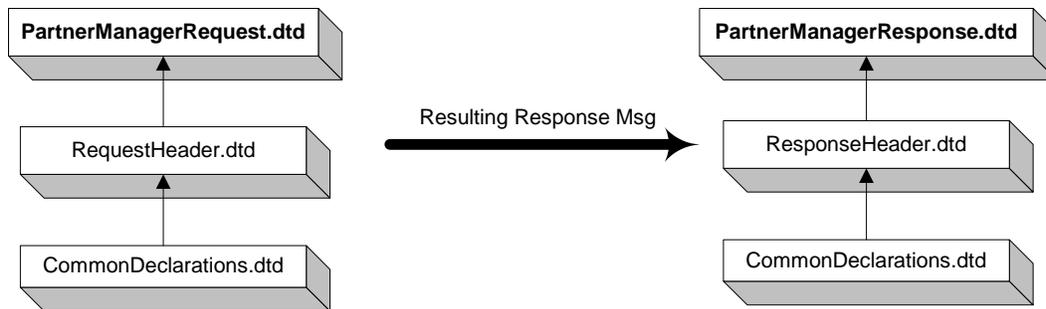


Figure 6. PartnerManagerRequest and Response DTD Dependencies.

1.3.6 Available Requests/Responses

XML corresponding to the AvailableRequest and AvailableResponse DTDs provides the Partner with the ability to check the availability of one or more domains, or generate a list of possibilities for a domain prefix. The Generate Domains list shows the domains that are available, those that are already taken, or those that, for one reason or another Network Solutions were unable to determine the status (such as when the Registry is unavailable). As illustrated in Figure 7, the AvailableRequest DTD depends on elements found in the RequestHeader and CommonDeclarations DTDs.



Figure 7. AvailableRequest and AvailableResponse DTD Dependencies.

1.3.7 Order Status Requests/Order Responses

XML corresponding to the OrderStatusRequest DTD returns the status of an Order recently sent to the Network Solutions system. It returns an OrderResponse that details the current status of each line item (domain or value added product) and where the order is in the fulfillment cycle (pending/fulfilled). Since an order can contain many product requests, these individual requests within the order are broken down into line items. Each line item has a unique ID, status, and product information. Essentially, the details of the order are captured by the order's line items. As illustrated in Figure 8, the OrderStatusRequest DTD depends on elements found in the RequestHeader and CommonDeclarations DTDs.

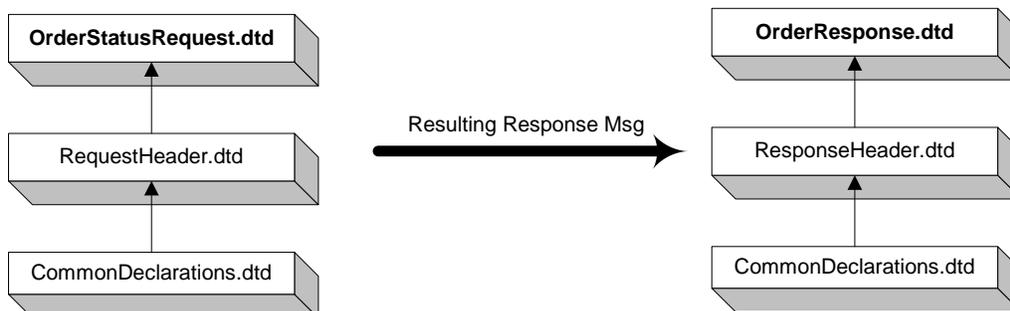


Figure 8. OrderStatusRequest and OrderStatusResponse DTD Dependencies.

1.3.8 Email Forwarding Requests

XML corresponding to the EmailForwardingRequest DTD is directly related to the .name bundle, which includes a .name domain and a corresponding email address that must be

forwarded to an active email box. The EmailForwarding Request allows the user to purchase a .name bundle and modify the email forwarding address.

1.3.9 Email Forwarding Lookup Requests/Responses

XML corresponding to the EmailForwardingLookupRequest and the EmailForwardingLookup Response DTDs return information concerning the forwarding email address associated with a .name domain.

1.3.10 Security Requests/Responses

XML corresponding to the SecurityRequest and SecurityResponse DTDs allows the user to authenticate and authorize both Partner and customer information. The Partner may also request the organization or first and last name of the customer or Partner being validated. It also allows the Partner to verify if a product or customer is associated with the Partner, or if a product is associated with a particular customer. As illustrated in Figure 9, the SecurityRequest DTD depends on elements found in the RequestHeader and CommonDeclarations DTDs.

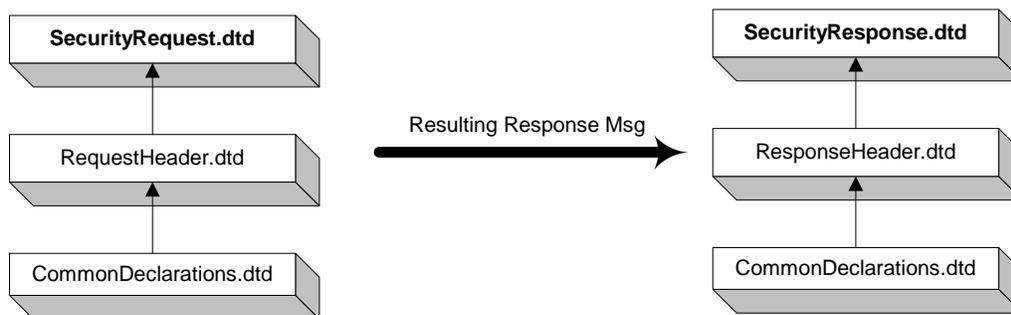


Figure 9. SecurityRequest and SecurityResponse DTD Dependencies.

1.3.11 Authorized Agent

The Partner Protocol provides the Partner with considerable flexibility and power. One area where Partner Protocol flexibility is most apparent is in the authorization strategy. The authorization strategy is simply the approach that is employed by the Partner to validate that the XML request is indeed authorized. The Partner Protocol accommodates two types of authorization approaches: Super-User (Partner) authorization and Customer-Level authorization. The differences between the two authorization approaches are described below.

Super-User authorization allows the Partner to authorize a request without having to specify the password of the customer who “owns” the product on which the request is operating. Stated simply, Super-User authorization allows the Partner to operate on any product within their customer set without having to specify the password of the customer. The Super-User authorization approach should only be used on an as-needed basis and when the Partner is using the PartnerManager Requests to retrieve data for all domains or all customers that belong to the Partner. It is not recommended for typical web presence applications.

If the Partner is providing a web-presence directly for its end customers, then it is essential that the Partner not utilize the Super-User authorization. The Authorized Agent should in this case be the customer, not the Partner. The web pages (in this example), would need to collect the end customer’s UserID and Password, and use these values in the AuthorizedAgent element. This would ensure that the Partner Protocol authorizes the Partner’s “end customer” for each request, and would only allow the request to proceed if the end customer’s ID / password combination is authenticated.

The following sample XML contains a partial Partner Protocol request that uses Super-User authorization.

Sample – Super-User Authorization

```
<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
```

Partner ID / password is always required here

```

<ModifyBusiness>
<AuthorizedAgent SystemUserType="partner">
  <SystemUser>
    <UserID>123</UserID>
    <Password>partner</Password>
  </SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
</ModifyBusiness>
</Body>
</UserRequest>
    
```

Partner ID / password authorizes this request

This is the customer ID on which the request operates

Note: The Partner should act as the Authorized Agent only in specific situations when the Partner wants to explicitly bypass the customer authorization mode of the Partner Protocol. Such a situation might be if the Partner wants to modify the password for a customer account because the end customer forgot their password.

Customer-Level authorization is the most common method of authorization and is the recommended approach for web based application development. If the customer is the Authorized Agent, then the UserID will be the customer's account number and will also be used as the CustomerID. This customer will only be allowed to make changes on the products that he/she owns. The following sample XML contains a partial Partner Protocol request that uses Customer-Level authorization.

Sample – Customer-Level Authorization

```

<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments>Example Authorized Agent is Customer</Comments>
  </RequestHeader>
  <Body>
    <ModifyBusiness>
      <AuthorizedAgent SystemUserType="customer">
        <SystemUser>
          <UserID>456</UserID>
          <Password>customer</Password>
        </SystemUser>
      </AuthorizedAgent>
      <CustomerID>456</CustomerID>
    </ModifyBusiness>
  </Body>
</UserRequest>
    
```

Partner ID / password is always required here

Customer ID / password authorizes this request

This is the customer ID on which the request operates

1.4 TLD Dependent Behavior

Some of the behavior of the Network Solutions Wholesale System depends on which Registry is authoritative for the TLD in question. Registries are independent, and can have different requirements on the domain names under their control. Table 5 highlights some of the more common TLD dependent domain restrictions.

Table 5. TLD Registry Reference Table.

TLD	Registry Type	Premium Domain Names	Minimum Domain Name Length (not including "." & TLD)	Minimum Registration
com/net	thin	No	2	1 year

org	thick	No	3	1 year
info	thick	No	3	1 year
biz	thick	No	2	1 year
tv	thin	No	4	1 year
cc	thin	No	3	1 year
ws	thin	No	4	1 years
tc	thin	No	2	1 year***
gs	thin	No	2	1 year***
ms	thin	No	2	1 year***
bz	thin	No	2	1 year
vg	thin	No	2	1 year***
name	thick	No	3 (last name)	1 year*
us	thick	No	2	1 years**
at	thick	No	3	1 year
be	thick	No	3	1 year
co.uk	thick	No	3	2 years
org.uk	thick	No	3	2 years
me.uk	thick	No	3	2 years
com.mx	thick	No	3	2 years
co.nz	thick	No	3	1 year
net.nz	thick	No	3	1 year
org.nz	thick	No	3	1 year
de	thick	No	3	1 year
cn	thick	No	3	1 year****
com.cn	thick	No	3	1 year****
net.cn	thick	No	3	1 year****
org.cn	thick	No	3	1 year****
ah.cn	thick	No	3	1 year****
bj.cn	thick	No	3	1 year****
cq.cn	thick	No	3	1 year****
fj.cn	thick	No	3	1 year****
gd.cn	thick	No	3	1 year****
gs.cn	thick	No	3	1 year****
gx.cn	thick	No	3	1 year****
gz.cn	thick	No	3	1 year****
ha.cn	thick	No	3	1 year****
hb.cn	thick	No	3	1 year****
he.cn	thick	No	3	1 year****
hi.cn	thick	No	3	1 year****
hk.cn	thick	No	3	1 year****
hl.cn	thick	No	3	1 year****
hn.cn	thick	No	3	1 year****
jl.cn	thick	No	3	1 year****
js.cn	thick	No	3	1 year****
jx.cn	thick	No	3	1 year****
ln.cn	thick	No	3	1 year****
mo.cn	thick	No	3	1 year****
nm.cn	thick	No	3	1 year****
nx.cn	thick	No	3	1 year****
qh.cn	thick	No	3	1 year****
sc.cn	thick	No	3	1 year****
sd.cn	thick	No	3	1 year****
sh.cn	thick	No	3	1 year****

sn.cn	thick	No	3	1 year****
sx.cn	thick	No	3	1 year****
tj.cn	thick	No	3	1 year****
tw.cn	thick	No	3	1 year****
xj.cn	thick	No	3	1 year****
yn.cn	thick	No	3	1 year****
zj.cn	thick	No	3	1 year****
tw	thick	No	3	1 year****
com.tw	thick	No	3	1 year****
org.tw	thick	No	3	1 year****
eu	thick	No	4	1 year
mobi	thick	No	3	1 year*****
am	thick	No	2	1 year
fm	thick	No	3	1 year
us.com	thick	No	3	1 year*****
uk.com	thick	No	3	1 year*****
eu.com	thick	No	3	1 year*****
de.com	thick	No	3	1 year*****
la	thick	No	3	1 year*****
br.com	thick	No	3	1 year*****
cn.com	thick	No	3	1 year*****
gb.com	thick	No	3	1 year*****
gb.net	thick	No	3	1 year*****
hu.com	thick	No	3	1 year*****
jpn.com	thick	No	3	1 year*****
kr.com	thick	No	3	1 year*****
no.com	thick	No	3	1 year*****
qc.com	thick	No	3	1 year*****
ru.com	thick	No	3	1 year*****
sa.com	thick	No	3	1 year*****
se.com	thick	No	3	1 year*****
se.net	thick	No	3	1 year*****
uk.net	thick	No	3	1 year*****
uy.com	thick	No	3	1 year*****
za.com	thick	No	3	1 year*****
es	thick	No	3	1 year
com.es	thick	No	3	1 year
org.es	thick	No	3	1 year
nom.es	thick	No	3	1 year
in	thick	No	3	1year ****
co.in	thick	No	3	1year ****
net.in	thick	No	3	1year ****
org.in	thick	No	3	1year ****
firm.in	thick	No	3	1year ****
gen.in	thick	No	3	1year ****
ind.in	thick	No	3	1year ****
ar.com	thick	No	3	1year*****
gd	thin	No	3	1 year***
cx	thick	No	3	1 year***

**The accepted PurchasePeriod values for the .name TLD are 1, 2 and 10 years.*

****The accepted PurchasePeriod values for the .us TLD are 1, 2, 5 and 10 years.**

*****The ExpirationDate for these TLDs cannot exceed 5 years.**

******The accepted PurchasePeriod values is 1 to 5 years and .tw are 1, to 10 years.**

*******The accepted PurchasePeriod values are 1,2,3,4,5,6,7,8,9,10 years**

Note: Various validation rules also vary by TLD. See Appendix D for a detailed description of the validation rules specific to each TLD.

Registry Type is either thin or thick. A thin Registry only records the expiration date and nameserver information for a domain name. A thick Registry also requires all of the contact information for the domain.

For a thin Registry, the Wholesale system reserves the name in the Registry before the system returns any response. The completion of the order request within the Network Solutions back-end systems can take a short while to complete, but the domain has already been acquired on behalf of the customer. DomainReservationStatusType in DomainReservationStatus of the OrderResponse will be Reserved.

For a thick Registry, the Wholesale system validates the information provided in the order request, and returns a general Order Submitted type of response. The domain name is not immediately reserved in the Registry. This is because to reserve a name is the same as completing the entire order, which is more time consuming. Partners can lookup the status of the order to determine if the order is successfully fulfilled.

DomainReservationStatusType in DomainReservationStatus of the OrderResponse will be Queued.

Premium Domain Names means the registry reserves certain domain names to be sold at a non-standard price. The domain names above the standard price cannot be purchased via the Network Solutions Wholesale System.

1.4.1 TLD Response Time

The following ccTLDs will usually fulfill within 48 hours: .co.uk, .me.uk, .org.uk, .be, .at, .nz, .de, .am, .fm, .es, .com.es, .org.es, .nom.es, .in, .co.in, .net.in, .org.in, .firm.in, .gen.in, .ind.in. The following ccTLDs usually fulfill within 14 days: .com.mx.

1.4.2 DNS and Name Server Configuration Requirements

Some registries require that a domain's zone be setup **prior** to registration and that the setup adhere to certain requirements. The goal of these Registry requirements is to help ensure that DNS lookups happen efficiently and to lessen the likelihood that "bad" or un-necessary domain and nameserver registrations reside in the Registry. The TLDs for which these requirements are imposed and verified by the registries are listed below in Table 6. Detailed descriptions of these requirements are described in the subsequent sections. In cases where a required value varies from what is in the detailed requirement description the required value is indicated in Table 6.

Table 6. DNS and Zone File Requirements Per TLD

Requirement	.de	.com.mx	.be	.co.nz	.net.nz	.org.nz	.at
Primary Name Server		X	X	X	X	X	X
Zone Serial Number	X	X	X	X	X	X	X
Zone Refresh Frequency	10000 - 86400	X	X	X	X	X	X
Zone Retry Frequency	1800 - 28800	X	X	X	X	X	X
Minimum TTL	180 -345600	X	3600 - 86400	X	X	X	X
Expiration Frequency	604800 - 3600000	X	X	X	X	X	X
Zone Name Servers same as Registry Name Servers	X	X	X	X	X	X	X
MX Record Required	X						
Name Servers on Separate Class C Networks	X						

1.4.3 Primary Name Server

The zone file for the domain being registered must be setup on the primary and all secondary name servers prior to the registration attempt. The primary name server specified in the SOA (Start of Authority) record of that zone file must be the first name server (host) identified for the domain being registered. An example SOA record may look something like this:

```
example.de IN SOA ns1.somedomain.de. admin.mail.example.de. (  
                2002061301 ; Serial number  
                10080      ; Refresh frequency  
                3600       ; Retry frequency  
                2592000    ; Expiration frequency  
                86400 )    ; Minimum TTL
```

So to meet the Primary Name Server requirement, using the above example, the Host1 XML element in a CreateRegistration or ModifyRegistration request sent to the Partner Protocol server should specify the name server (host) ns1.somedomain.de.

1.4.4 Zone Serial Number

The zone file for the domain being registered must be setup prior to the registration attempt, and the serial number specified in the SOA record of that zone file must be identical for all name servers (hosts) for the domain being registered.

So to meet the Serial Number requirement, using the above example, the name servers specified by all of the Host[1,2,3,4] XML elements in a CreateRegistration or ModifyRegistration request sent to the Partner Protocol server should specify identical Serial number values (2002061301 in the example above).

1.4.5 Zone Refresh Frequency

The zone file for the domain being registered must be setup prior to the registration attempt, and the refresh frequency specified in the SOA record of that zone must be between 1200 seconds (20 minutes) and 43200 (12 hours).

1.4.6 Zone Retry Frequency

The zone file for the domain being registered must be setup prior to the registration attempt, and the retry frequency specified in the SOA record of that zone must be less than the refresh frequency referred to in the Zone Refresh Frequency Requirement.

1.4.7 Minimum TTL (Time To Live)

The zone file for the domain being registered must be setup prior to the registration attempt, and the minimum TTL value specified in the SOA record of that zone file must be between 3600 seconds (1 hour) and 86400 seconds (24 hours).

1.4.8 Expiration Frequency

The zone file for the domain being registered must be setup prior to the registration attempt, and the expiration frequency specified in the SOA record of that zone must be greater than the retry frequency and the minimum TTL values.

1.4.9 Zone Name Servers and Registry Name Servers

The zone file for the domain being registered must be setup prior to the registration attempt, and the name servers (hosts) listed as NS records in the zone must be the same name servers (hosts) submitted to the registry (via the Partner Protocol server) when registering the domain.

1.4.10 MX Record Required

The zone file for the domain being registered must be setup prior to the registration attempt, and the zone must specify an MX record under that domain that resolves directly to a host using an A record. The MX host resolution must not depend on a CNAME record.

1.4.11 Name Servers on separate Class C Networks

At least one of the name servers specified by the Host[1,2,3,4] XML elements in a Partner Protocol CreateRegistration or ModifyRegistration request must be on a different class C network than the other name servers serving that domain.

1.5 Partner Preference and Profile Options

Partners will have the option of establishing Partner preferences and options in Partner Protocol. Creating these profiles will aid communication between the Partner and Network Solutions for system outages, successful/failed transactions, domain renewal, and other pertinent information.

Table 7. Partner Preference and Profile Options lists the Partner Preference and Profile Options.

Table 7. Partner Preference and Profile Options

Preference/Option Name	Description
System Outage Notification Email Address	The email address where notifications will be automatically sent to notifying a Partner of scheduled and unscheduled downtime in the Wholesale OTE and production systems.
Success Notification Email Address	The email address where notifications will be automatically sent to when a transaction has successfully completed.
Failure Notification Email Address	The email address where notifications will be automatically sent to when a transaction has failed.
Renewal Notification Email Address	The email address where renewal notification will be automatically sent to on (or about) the 1st and the 15th of each month. On each date (1st, 15th) the Partner will be emailed a consolidated list of domains (products) that are coming up for renewal.
Deactivation Notification Email Address	The email address where deactivation notification will be automatically sent. A domain is deactivated automatically by the Network Solutions systems six days after the domain has expired. This notification will be a single email for each order that is submitted for deactivation. When a domain is deactivated, the host information is removed from the root zone file, thus the domain will no longer resolve. If the domain is not renewed within 36 days of the expiration date, it will be deleted.
XML API Access Control List	List of IP addresses of up to 20 that the partner can use to

	connect to Partner Protocol.
Ok_To_Contact_Registrant	Partner can choose to allow Network Solutions to contact the Registrant for renewal and deactivation notifications.
Success Notification	Partner can turn on/off success email notifications.
Deactivation Notification	Partner can turn on/off deactivation email notifications.
Default Purchase Period	Default number of years all domain creations. It can be from 1 to 10.
Default Renewal Period	Default number of years all domain renewals. It can be from 1 to 10.
Do Not Queue CreateRegistration/Parking	By default the Partner Protocol Gateway will accept domain registration requests even when the target registry is down. This allows partners to continue to submit registration requests even when the target registry is not available. It accomplishes this by queuing these requests and then asynchronously processing them once the target registry becomes available. However, the partner now has the option of bypassing this feature using a new Partner Preference. If this Partner Preference is set then the Partner Protocol gateway will not queue registration requests but will simply return a failure response in the OrderResponse. The partner would then need to submit the reservation request sometime later when the target registry is available.
Ignore Alternate Name Availability Sources	By default, the Partner Protocol Gateway will attempt to use alternate sources for domain name availability when the registry for a given TLD is not available. Partner Protocol's use of alternate name availability sources allows the Partner Protocol gateway to return an availability response even when the registries are down, which in turn allows our partners to continue to present domain names for sale even when the target registry is not available. The existence of alternate sources of name availability varies by TLD. For the com, net, org, .name, .us, and .info TLDs the only alternate source is Whois. However, other alternate sources will be implemented in the future. But the partner now has the option of disabling this feature using a Partner Preference. If this Partner Preference is set then the Partner Protocol gateway will not check alternate sources for name availability and the AvailabilityResponse will indicate "Failed" for the domains whose registry is not available.

1.6 Aging Rules

There are two email addresses that are maintained per Partner:

- A renewal notification e-mail address (required)
- A deactivation notification e-mail address (optional)

The first email address is the email address that renewal email messages will be sent to. These email notifications cannot be turned off. The emails will be sent to the Partner only. These emails are sent out on the 1st and the 15th of each month, on which date the Partner will be emailed a consolidated list of domain products that are up for renewal. They will be sorted into three categories:

- Products expiring between 45-75 days of the process date
- Products expiring between 15-45 days of the process date
- Products expiring between 0-15 days of the process date

On the expiration date, if the partner has chosen to allow Network Solutions to contact the Registrant, an email will be sent to the Registrant email address, saying that their domain has expired.

Thirty-six (36) days after the expiration date of the domain name, the domain name will be deleted from your Wholesale Account. No e-mail notice will be sent. If we have secured a backorder for the domain name, the domain name may be renewed on behalf of the current registrant and transferred to the new registrant in accordance with the Network Solutions Service Agreement. If no backorder for the domain name has been placed or if the backorder process is not fulfilled, the domain name will be deleted at the registry and may be available for redemption.

Type 1 domain names, those with extensions .BE, .DE, .CO.UK, .ME.UK, and .ORG.UK, .AM, .FM must be renewed 11 days prior to expiration or they will be deleted within the month that the domain name expires. Renewals submitted after 11 days prior to expiration will not be accepted and no redemptions will be possible for these domain names. The Registrant will not receive the Expiration or Deactivation Notices, as no renewals may be submitted beyond 11 days prior to expiration.

Type 2 domain names, those with extensions .AT, .COM.MX, .CO.NZ, .ORG.NZ and .NET.NZ, ES, COM.ES, ORG.ES, NOM.ES must be renewed 42 days prior to expiration or they will be deleted within the month that the domain name expires. Renewals submitted after 42 days prior to expiration will not be accepted and no redemptions will be possible for these domain names. The Registrant will not receive the Expiration or Deactivation Notices, as no renewals may be submitted beyond 42 days prior to expiration.

2.0 XML COMMAND SEQUENCE

As previously mentioned the XML transaction structure is made up of a request/response pair. The Request contains the action or operation that the Partner wants the XML server to execute and the Response contains the execution status and/or the data requested by the Partner.

This section focuses on the XML request structure including the transaction operation, XML format, the required versus optional fields, and the corresponding response.

Some XML requests may utilize data elements from previously completed transactions. The data element (UserID for example) from a previous response may be used as input to a secondary request to create an overall operation. For example, a Partner who wishes to acquire a domain for a brand new customer must first create the user (via the CreateIndividual or CreateBusiness request) and then issue the create domain command (via the CreateRegistration or CreateParking request). The remainder of this section describes each XML request in greater detail.

Most common data elements utilized in more than one type of request, like TechContact, AdminContact, and ProductID are defined in the CommonDataElements DTD. Appendix E describes these common data elements in detail. Appendix D contains the validation rules and sample values for each field.

2.1 User Request (UserRequest.dtd)

The actions contained in the UserRequest.dtd consist of:

- CreateIndividual,
- CreateBusiness,
- ModifyIndividual, and
- ModifyBusiness.

The following element from the UserRequest.dtd defines the Body:

```
<!ELEMENT Body (( CreateIndividual | CreateBusiness | ModifyIndividual | ModifyBusiness ) )
>
```

Table 8 contains a detailed explanation of the Body fields.

Note: *Only one action may be submitted in a single request.*

Table 8. Body for User Request.

Field	Required	Multiple	Notes
CreateIndividual	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
CreateBusiness	Conditional	NO	
ModifyIndividual	Conditional	NO	
ModifyBusiness	Conditional	NO	

2.1.1 Create Request

The following requests should be issued to create a new user.

2.1.1.1 CreateIndividual

The following element from the UserRequest.dtd defines the CreateIndividual action:

```
<!ELEMENT CreateIndividual (LoginName?, Password, FirstName, MiddleName?,
  LastName, Address, Phone, Fax?, Email, AuthQuestion, AuthAnswer,
  SupplementalRegistryData? ) >
```

The **CreateIndividual** Request creates a new individual user. The new user can purchase domains and other products. The new user can also be used as a technical, billing or admin contact for one or more domains and can be a Host Coordinator for one or more hosts.

Table 9 contains a detailed explanation of each field in the request.

Table 9. Required and Optional Fields for the CreateIndividual Request.

Field	Required	Multiple	Notes
-------	----------	----------	-------

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 8 for details.
CreateIndividual	Required	NO	
LoginName	Optional	NO	If possible this will set the login name of the user to this value. If that name is taken, then the login name will be initially set to the UserID. It can be changed later.
Password	Required	NO	The initial password for the new user. It can be changed later if desired.
FirstName	Required	NO	
MiddleName	Optional	NO	
LastName	Required	NO	
Address	Required	NO	See Appendix E for details.
Phone	Required	NO	
Fax	Optional	NO	
Email	Required	NO	
AuthQuestion	Required	NO	If Network Solutions needs to verify the identity because of a lost password these fields will be used.
AuthAnswer	Required	NO	
SupplementalRegistryData	Optional	NO	See Appendix E for Details. Required to be set for any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with the OrderRequest containing a ccTLD domain purchase.

Example - CreateIndividual Request

```

<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <CreateIndividual>
      <LoginName>SampleUser1</LoginName>
      <Password>customer</Password>
      <FirstName>John</FirstName>
      <MiddleName>Hank</MiddleName>
      <LastName>Doe</LastName>
      <Address>
        <AddressLine1>123 Any Street</AddressLine1>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
      </Address>
      <Phone>703-555-0001</Phone>
      <Fax>703-555-0002</Fax>
      <Email>john@example.com</Email>
      <AuthQuestion>Does this work?</AuthQuestion>
      <AuthAnswer>Yes.</AuthAnswer>
    </CreateIndividual>
  </Body>
</UserRequest>

```

```
</CreateIndividual>
```

```
</Body>
```

```
</UserRequest>
```

Expected Response - CreateIndividual

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">
```

```
<UserResponse>
```

```
<ResponseHeader>
```

```
<VERSION_5_0></VERSION_5_0>
```

```
<ReferenceNumber>91148615</ReferenceNumber>
```

```
<Timestamp>2006-08-01T12:33:56EDT</Timestamp>
```

```
</ResponseHeader>
```

```
<Body>
```

```
<Status>
```

```
<StatusCode>5700</StatusCode>
```

```
<Description>User created successfully</Description>
```

```
</Status>
```

```
<UserID>789</UserID>
```

```
<LoginName>SampleUser1</LoginName>
```

```
</Body>
```

```
</UserResponse>
```

Example - CreateIndividual Request (new .EU)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<UserRequest>
```

```
<RequestHeader>
```

```
<VERSION_5_0></VERSION_5_0>
```

```
<Authentication>
```

```
<PartnerID>123</PartnerID>
```

```
<PartnerPassword>partner</PartnerPassword>
```

```
</Authentication>
```

```
</RequestHeader>
```

```
<Body>
```

```
<CreateIndividual>
```

```
<LoginName>SampleUser1</LoginName>
```

```
<Password>customer</Password>
```

```
<FirstName>John</FirstName>
```

```
<MiddleName>Hank</MiddleName>
```

```
<LastName>Doe</LastName>
```

```
<Address>
```

```
<AddressLine1>123 Any Street</AddressLine1>
```

```
<City>Herndon</City>
```

```
<State>VA</State>
```

```
<PostalCode>20171</PostalCode>
```

```
<CountryCode>FR</CountryCode>
```

```
</Address>
```

```
<Phone>703-555-0001</Phone>
```

```
<Fax>703-555-0002</Fax>
```

```
<Email>john@example.com</Email>
```

```
<AuthQuestion>Does this work?</AuthQuestion>
```

```
<AuthAnswer>Yes.</AuthAnswer>
```

```
<SupplementalRegistryData>
```

```
<DotEUSupplementalData CustomerType="ResidentInEU" NaturalPerson="True" />
```

```
</SupplementalRegistryData>
```

```

</CreateIndividual>
</Body>
</UserRequest>
Expected Response - CreateIndividual (new .EU)
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">

<UserResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91148615</ReferenceNumber>
    <Timestamp>2006-08-01T17:37:21EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5700</StatusCode>
      <Description>User created successfully</Description>
    </Status>
    <UserID>789</UserID>
    <LoginName>SampleUser1</LoginName>
  </Body>
</UserResponse>

```

2.1.1.2 CreateBusiness

The following element from the *UserRequest.dtd* defines the *CreateBusiness* Request:

```
<!ELEMENT CreateBusiness ( LoginName?, Password, CompanyName, Address, Phone,
  Fax?, Email, LegalContact, AuthQuestion, AuthAnswer, SupplementalRegistryData? ) >
```

The **CreateBusiness** Request creates a new business user. This new user can purchase domains and other products. It can also be used as a technical, billing or admin contact for one or more domains and can be a Host Coordinator for one or more hosts. A business has two addresses, the main address, and a legal contact address.

Table 10 contains a detailed explanation of each field in the request.

Table 10. Required and Optional Fields for the CreateBusiness Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 8 for details.
CreateBusiness	Conditional	NO	
LoginName	Optional	NO	If possible this will set the login name of the user to this value. If that name is taken, then the login name will be initially set to the UserID. It can be changed later.
Password	Required	NO	The initial password for the new user. It can be changed later if desired.
CompanyName	Required	NO	
Address	Required	NO	See Appendix E for details.
Phone	Required	NO	
Fax	Optional	NO	
Email	Required	NO	
LegalContact	Required	NO	See Appendix E for details.
AuthQuestion	Required	NO	If Network Solutions needs to verify the identity
AuthAnswer	Required	NO	because of a lost password these fields will be used.

Field	Required	Multiple	Notes
SupplementalRegistryData	Optional	NO	See Appendix E for Details. Required to be set for any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with the OrderRequest containing a ccTLD domain purchase.

Example - CreateBusiness Request

```
<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <CreateBusiness>
      <LoginName>ExampleCompany</LoginName>
      <Password>customer</Password>
      <CompanyName>Example Company</CompanyName>
      <Address>
        <AddressLine1>123 Any Street</AddressLine1>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
      </Address>
      <Phone>703-555-0044</Phone>
      <Fax>703-555-4444</Fax>
      <Email>admin@example.com</Email>
      <LegalContact>
        <FirstName>Jane</FirstName>
        <LastName>Doe</LastName>
        <Email>jane@example.com</Email>
      </LegalContact>
      <AuthQuestion>Does this work?</AuthQuestion>
      <AuthAnswer>Yes.</AuthAnswer>
    </CreateBusiness>
  </Body>
</UserRequest>
```

Expected Response - CreateBusiness

```
<?xml version="1.0"?>
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">
<UserResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278490</ReferenceNumber>
    <Timestamp>2006-08-01T17:41:12EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
```

```

    <StatusCode>5700</StatusCode>
    <Description>User created successfully</Description>
  </Status>
  <UserID>789</UserID>
  <LoginName>ExampleCompany</LoginName>
</Body>
</UserResponse>

```

Example - CreateBusiness Request (new .EU)

```

<?xml version="1.0" encoding="UTF-8" ?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <CreateBusiness>
      <LoginName>ExampleCompany</LoginName>
      <Password>customer</Password>
      <CompanyName>Example Company</CompanyName>
      <Address>
        <AddressLine1>123 Any Street</AddressLine1>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
      </Address>
      <Phone>703-555-0044</Phone>
      <Fax>703-555-4444</Fax>
      <Email>admin@example.com</Email>
      <LegalContact>
        <FirstName>Jane</FirstName>
        <LastName>Doe</LastName>
        <Email>jane@example.com</Email>
      </LegalContact>
      <AuthQuestion>Does this work?</AuthQuestion>
      <AuthAnswer>Yes.</AuthAnswer>
      <SupplementalRegistryData>
        <DotEUSupplementalData CustomerType="OrganizationIncorporatedInEU"/>
      </SupplementalRegistryData>
    </CreateBusiness>
  </Body>
</UserRequest>

```

Expected Response - CreateBusiness (new .EU)

```

<?xml version="1.0" ?>
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">
<UserResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278490</ReferenceNumber>
    <Timestamp>2006-08-01T17:43:47EDT</Timestamp>

```

```

    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5700</StatusCode>
      <Description>User created successfully</Description>
    </Status>
    <UserID>789</UserID>
    <LoginName>ExampleCompany</LoginName>
  </Body>
</UserResponse>

```

2.1.2 Modify Request

The following requests should be issued to modify an existing account.

2.1.2.1 *ModifyIndividual*

The following element from the *UserRequest.dtd* defines the *ModifyIndividual* request:

```

<!ELEMENT ModifyIndividual (AuthorizedAgent, CustomerID, NewLoginName?,
NewPassword?, FirstName?, MiddleName?, LastName?, Address?, Phone?, Fax?, Email?,(
AuthQuestion, AuthAnswer )?, SupplementalRegistryData? ) >

```

The **ModifyIndividual** Request modifies an existing individual account. This individual must be previously entered in the Network Solutions system, and the CustomerID must be used to identify the individual (not the LoginName).

Optional fields that are provided but are empty will result in that field being “emptied out” or cleared. Optional fields that are not provided (the element is not sent as part of the request) will result in no change to that field.

Table 11 contains a detailed explanation of each field in the request.

Table 11. Required and Optional Fields for the *ModifyIndividual* Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 8 for details.
ModifyIndividual	Conditional	NO	
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Required	NO	See Appendix E for details.
NewLoginName	Optional	NO	
NewPassword	Optional	NO	
FirstName	Optional	NO	
MiddleName	Optional	NO	
LastName	Optional	NO	
Address	Optional	NO	See Appendix E for details.
Phone	Optional	NO	
Fax	Optional	NO	
Email	Optional	NO	
AuthQuestion	Optional	NO	If Network Solutions needs to verify the identity because of a lost password these fields will be used.
AuthAnswer	Optional	NO	
SupplementalRegistryData	Optional	NO	See Appendix E for Details. Required to be set for any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with the OrderRequest containing a .ccTLD domain purchase.

Example - ModifyIndividual Request

```
<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <ModifyIndividual>
      <AuthorizedAgent SystemUserType="customer">
        <SystemUser>
          <UserID>456</UserID>
          <Password>customer</Password>
        </SystemUser>
      </AuthorizedAgent>
      <CustomerID>456</CustomerID>
      <Email>jdoe@exampleco.com</Email>
    </ModifyIndividual>
  </Body>
</UserRequest>
```

Expected Response - ModifyIndividual

```
<?xml version="1.0"?>
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">
<UserResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>78362981</ReferenceNumber>
    <Timestamp>2006-08-01T17:45:51EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5701</StatusCode>
      <Description>User modified successfully</Description>
    </Status>
    <UserID>456</UserID>
  </Body>
</UserResponse>
```

2.1.2.2 ModifyBusiness

The following element from the *UserRequest.dtd* defines the *ModifyBusiness* Request:

```
<!ELEMENT ModifyBusiness (AuthorizedAgent, CustomerID, NewLoginName?,
NewPassword?, CompanyName?, Address?, Phone?, Fax?, Email?, LegalContact?,
(AuthQuestion, AuthAnswer)?, SupplementalRegistryData?) >
```

The **ModifyBusiness** Request should be issued to modify an existing business account. This business must be previously entered in the Network Solutions system, and the CustomerID must be used to identify the business (not LoginName).

Optional fields that are provided but are empty will result in that field being "emptied out" or cleared. Optional fields that are not provided (the element is not sent as part of the request) will result in no change to that field.

Table 12 contains a detailed explanation of each field in the request.

Table 12. Required and Optional Fields for the ModifyBusiness Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body			See Table 8 for details.
ModifyBusiness	Required	NO	
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Required	NO	
NewLoginName	Optional	NO	
NewPassword	Optional	NO	
CompanyName	Optional	NO	
Address	Optional	NO	See Appendix E for details.
Phone	Optional	NO	
Fax	Optional	NO	
Email	Optional	NO	
LegalContact	Optional	NO	See Appendix E for details.
AuthQuestion	Optional	NO	If Network Solutions needs to verify the identity because of a lost password these fields will be used.
AuthAnswer	Optional	NO	
SupplementalRegistryData	Optional	NO	See Appendix E for Details. Required to be set for any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with the OrderRequest containing a ccTLD domain purchase.

Example - ModifyBusiness Request

```
<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <ModifyBusiness>
      <AuthorizedAgent SystemUserType="customer">
        <SystemUser>
          <UserID>789</UserID>
          <Password>customer</Password>
        </SystemUser>
      </AuthorizedAgent>
      <CustomerID>789</CustomerID>
      <LegalContact>
        <FirstName>Jennifer</FirstName>
        <LastName>Smith</LastName>
      </LegalContact>
    </ModifyBusiness>
  </Body>
</UserRequest>
```

Expected Response - ModifyBusiness

```
<?xml version="1.0"?>
```

```
<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">
<UserResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278493</ReferenceNumber>
    <Timestamp>2006-08-01T17:57:49EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5701</StatusCode>
      <Description>User modified successfully</Description>
    </Status>
    <UserID>789</UserID>
  </Body>
</UserResponse>
```

2.2 Order Request (*OrderRequest.dtd*)

The actions contained in the *OrderRequest.dtd* consist of:

- CreateRegistration,
- CreateParking,
- CreateHost,
- CreatePrivateRegistration,
- ModifyRegistration,
- ModifyPrivateRegistration,
- ModifyParking,
- ModifyHost,
- DeleteRegistration,
- DeletePrivateRegistration,
- DeleteParking,
- DeleteHost,
- RenewRegistration,
- RenewParking,
- RenewDomainsOnly,
- CascadeRenewDomain,
- CascadeDeleteDomain,
- ConvertToRegistration,
- ConvertToParking,
- TransferInternalToRegistration,
- TransferInternalToParking,
- ModifyCustomerName
- InboundTransferToRegistration,

- InboundTransferToParking,
- RetrieveAuthCode

The following element from the OrderRequest.dtd defines the Body:

```
<!ELEMENT Body ( AuthorizedAgent, CustomerID, Billing?, Product ) >
```

The `AutoRenew` flag in the `Billing` element may be set to "True" so a domain will be automatically renewed by the Network Solutions system, or it may be set to "False" (default value) so a domain will not be automatically renewed by the Network Solutions system. If the flag is not provided, then the Partner Protocol system will set the value to "False". If the `AutoRenew` flag is set as true for a domain, the Partner Protocol system will automatically submit a renew request for that domain 15 days prior to the expiration date for the domain. If a Partner no longer wishes for a domain to be auto renewed, the Partner must submit a `Modify` request *before* the 15th day prior to the expiration date for the domain, setting the value of `AutoRenew` to "False".

The `IsPricingRequest` flag in the `Billing` element may be set to true in order to simply get the price for an order. If the `IsPricingRequest` flag is set then the order will not be submitted for fulfillment. Instead the price of the order and the price of each order line item will be calculated and returned in the `OrderResponse`.

Table 13 contains a detailed explanation of the Body fields.

Note: *Only one action may be submitted in a single request.*

Table 13. Body for Order Request.

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Required	NO	
Billing	Optional	NO	
DiscountCode	Optional	NO	Tag ignored for processing at this time.
AutoRenew	Optional	NO	Default value is 'False'. See above for more details.
IsPricingRequest	Optional	NO	The <code>IsPricingRequest</code> flag in the <code>Billing</code> element may be set to true in order to simply get the price for an order. If the <code>IsPricingRequest</code> flag is set then the order will not be submitted for fulfillment. Instead the price of the order and the price of each order line item will be calculated and returned in the <code>OrderResponse</code>
Product	Required	NO	
CreateRegistration	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
CreateParking	Conditional	NO	
CreateHost	Conditional	NO	
CreatePrivateRegistration	Conditional	NO	
ModifyRegistration	Conditional	NO	
ModifyPrivateRegistration	Conditional	NO	
ModifyParking	Conditional	NO	
ModifyHost	Conditional	NO	
DeleteRegistration	Conditional	NO	
DeletePrivateRegistration	Conditional	NO	
DeleteParking	Conditional	NO	
DeleteHost	Conditional	NO	
RenewRegistration	Conditional	NO	
RenewPrivateRegistration	Conditional	NO	
RenewParking	Conditional	NO	
RenewDomainsOnly	Conditional	NO	
CascadeRenewDomain	Conditional	NO	
CascadeDeleteDomain	Conditional	NO	
ConvertToRegistration	Conditional	NO	

Field	Required	Multiple	Notes
ConvertToParking	Conditional	NO	
TransferInternalToRegistration	Conditional	NO	
TransferInternalToParking	Conditional	NO	
TransferInternalToRegistration	Conditional	NO	
ModifyCustomerName	Conditional	NO	
InboundTransferToRegistration	Conditional	YES	
InboundTransferToParking	Conditional	YES	

2.2.1 Create Request

The following requests should be issued to create a domain.

Note: This request is processed asynchronously. For an explanation of our synchronous and asynchronous processing, please refer to section 1.4 and section 1.3.2 of this document. For most domains may take up to 48 hours before the domain is fully configured and resolves. For some ccTLDs, it may take up to 14 days before the domain is fully configured and resolves.

2.2.1.1 CreateRegistration

The following element from the OrderRequest.dtd defines the CreateRegistration Request:

```
<!ELEMENT CreateRegistration ( DomainName+, PurchasePeriod, TechContact?,
AdminContact?, BillContact?, BulkWhoisOptOut?, Hosts?, SupplementalRegistryData? ) >
```

The **CreateRegistration** Request should be issued to create a domain. This domain is one that will be hosted completely separately from Network Solutions. The domain must have valid host information for the domain to properly resolve. CreateRegistration should not be used to change host information. The Partner may submit an existing host, or create a new one, however an existing host cannot be modified. To modify an existing host name or IP address, use the ModifyHost Request. Table 14 contains a detailed explanation of each field in the request.

Note: Both *HostIP* and *HostName* are required for hosts where the TLD of the host is not a TLD that the Partner Protocol currently supports. Please refer to section 1.4 for a detailed list of supported TLDs.

Table 14. Required and Optional Fields for the CreateRegistration Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
CreateRegistration			
DomainName	Required	YES	A maximum of 1000 domain names are allowed per request.
PurchasePeriod	Required	NO	
TechContact	Optional	NO	See Appendix E for details. If AdminContact and/or
AdminContact	Optional	NO	TechContact are not provided, the default value is the customer.
BillContact	Optional	NO	If BillContact is not provided it defaults to the Partner's billing information. Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
Hosts	Optional	NO	If no valid hosts are provided, the domain will not resolve. Please see note above. See Appendix E for details.
SupplementalRegistryData	Optional	NO	See Appendix E for details. Required to be set for any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with a CreateIndividual, CreateBusiness, ModifyIndividual, or ModifyBusiness.

Field	Required	Multiple	Notes
			If provided here, with a CreateRegistration, then the data provided will be saved and associated with the Individual or Business identified by the CustomerID associated with this request. If the CreateRegistration request contains a .ccTLD domain name and this supplemental data is not already set for the indicated CustomerID and is not provided here then the ccTLD domain names in this request will not be accepted.
PrivateRegistrationDomains	Optional	YES	Current domain name(s) being registered

Example – CreateRegistration Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateRegistration>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.com</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.example.com</HostName>
              <HostIP1>2.3.4.6</HostIP1>
            </Host>
          </Host2>
        </Hosts>
      </CreateRegistration>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - CreateRegistration

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
```

```
<ResponseHeader>
  <VERSION_5_0></VERSION_5_0>
  <ReferenceNumber>14278502</ReferenceNumber>
  <Timestamp>2006-08-01T18:05:29EDT</Timestamp>
</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278502</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>1000309149</OrderID>
    <OrderAction>Create</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>17.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns1.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>2</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns2.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>3</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <ProductID>WN.D.226564377</ProductID>
      <DomainName>example.com</DomainName>
      <LinItemDetail>
        <DomainLinItemDetail>
          <DomainReservationStatus DomainReservationStatusType="Queued"/>
        </DomainLinItemDetail>
      </LinItemDetail>
      <Price>17.0</Price>
    </LinItem>
  <OrderSubmissionDate>2006-08-01T18:06:00EDT</OrderSubmissionDate>
```

```

        <NumberOfRecordsReturned>3</NumberOfRecordsReturned>
        <NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – CreateRegistration Request (.us ccTLD)

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateRegistration>
        <DomainName>example.us</DomainName>
        <PurchasePeriod>2</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.us</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.example.us</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host2>
        </Hosts>
        <SupplementalRegistryData>
          <DotUSSupplementalData CustomerType="OrganizationIncorporatedInUS"
            AnticipatedUsage="ForProfit" />
        </SupplementalRegistryData>
      </CreateRegistration>
    </Product>
  </Body>
</OrderRequest>

```

Expected Response - CreateRegistration (.us ccTLD)

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>

```

```
<ResponseHeader>
  <VERSION_5_0></VERSION_5_0>
  <ReferenceNumber>25668532</ReferenceNumber>
  <Timestamp> 2006-08-01T18:10:10EDT</Timestamp>
</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>25668532</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>135049604</OrderID>
    <OrderAction>Create</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>20.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns1.example.us</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>2</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns2.example.us</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>3</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <ProductID>WN.D.174963827</ProductID>
      <DomainName>example.us</DomainName>
      <LinItemDetail>
        <DomainLinItemDetail>
          <DomainReservationStatus DomainReservationStatusType="Queued"/>
        </DomainLinItemDetail>
      </LinItemDetail>
      <Price>20.0</Price>
    </LinItem>
  <OrderSubmissionDate>2006-08-01T18:10:37EDT</OrderSubmissionDate>
```

```

        <NumberOfRecordsReturned>3</NumberOfRecordsReturned>
        <NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – CreateRegistration (IsPricingRequest)

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Billing>
      <IsPricingRequest></IsPricingRequest>
    </Billing>
    <Product>
      <CreateRegistration>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.com</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.example.com</HostName>
              <HostIP1>2.3.4.6</HostIP1>
            </Host>
          </Host2>
        </Hosts>
      </CreateRegistration>
    </Product>
  </Body>
</OrderRequest>

```

Expected Response – CreateRegistration (IsPricingRequest)

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>

```

```
<VERSION_5_0></VERSION_5_0>
<ReferenceNumber>14278527</ReferenceNumber>
<Timestamp>2006-08-01T18:16:24EDT</Timestamp>
</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278527</ReferenceNumber>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <OrderAction>Create</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>17.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns1.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>2</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns2.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>3</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <DomainName>example.com</DomainName>
      <Price>17.0</Price>
    </LinItem>
    <NumberOfRecordsReturned>3</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>
```

Example – CreateRegistration Request (with Private Registration)

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateRegistration>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.com</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.example.com</HostName>
              <HostIP1>2.3.4.6</HostIP1>
            </Host>
          </Host2>
        </Hosts>
        <PrivateRegistrationDomains>example.com</PrivateRegistrationDomains>
      </CreateRegistration>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response – CreateRegistration (with Private Registration)

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278527</ReferenceNumber>
    <Timestamp>2006-08-16T18:25:36EDT</Timestamp>
  </ResponseHeader>
  <Body>
```

```
<Order>
  <ReferenceNumber>14278527</ReferenceNumber>
  <Status>
    <StatusCode>7000</StatusCode>
    <Description>Order submitted successfully and is in pending status</Description>
  </Status>
  <OrderID>186028266</OrderID>
  <OrderAction>Create</OrderAction>
  <Customer>
    <UserID>789</UserID>
  </Customer>
  <Price>0.0</Price>
  <TieredPrice Selection="False"/>
  <LinItem>
    <LinItemID>1</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Host</ProductType>
    <Name>ns1.example.com</Name>
    <Price>0.0</Price>
  </LinItem>
  <LinItem>
    <LinItemID>2</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Host</ProductType>
    <Name>ns2.example.com</Name>
    <Price>0.0</Price>
  </LinItem>
  <LinItem>
    <LinItemID>3</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Registration</ProductType>
    <ProductID>WN.D.186941239</ProductID>
    <DomainName>example.com</DomainName>
    <LinItemDetail>
      <DomainLinItemDetail>
        <DomainReservationStatus DomainReservationStatusType="Queued"/>
      </DomainLinItemDetail>
    </LinItemDetail>
    <Price>0.0</Price>
  </LinItem>
  <LinItem>
    <LinItemID>4</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Private Registration</ProductType>
```

```
<ProductID>186941240</ProductID>
<Name>example.com</Name>
<LinItemDetail>
  <DomainLinItemDetail>
    <DomainReservationStatus DomainReservationStatusType="Queued"/>
  </DomainLinItemDetail>
</LinItemDetail>
<Price>20.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T18:25:44EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>4</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>4</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>
```

Example – CreateRegistration Request (new .EU)

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>password</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>password</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateRegistration>
        <DomainName>example.eu</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.eu</HostName>
              <HostIP1>192.178.10.142</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.example.eu</HostName>
              <HostIP1>192.5.200.8</HostIP1>
            </Host>
          </Host2>
        </Hosts>
        <SupplementalRegistryData>
```

```
<DotEUSupplementalData CustomerType="ResidentInEU" NaturalPerson="True"/>
</SupplementalRegistryData>
</CreateRegistration>
</Product>
</Body>
</OrderRequest>
```

Expected Response – CreateRegistration (new .EU)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14411739</ReferenceNumber>
    <Timestamp>2006-08-16T18:29:32EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14411739</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000676624</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>ns1.example.eu</Name>
        <Price>0.0</Price>
      </LineItem>
      <LineItem>
        <LineItemID>2</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>ns2.example.eu</Name>
        <Price>0.0</Price>
      </LineItem>
      <LineItem>
        <LineItemID>3</LineItemID>
```

```

<Status>
  <StatusCode>7004</StatusCode>
  <Description>Line item submitted successfully</Description>
</Status>
<ProductType>Registration</ProductType>
<ProductID>WN.D.226974198</ProductID>
<DomainName>example.eu</DomainName>
<LinItemDetail>
  <DomainLinItemDetail>
    <DomainReservationStatus DomainReservationStatusType="Queued"/>
  </DomainLinItemDetail>
</LinItemDetail>
<Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T18:29:39EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>3</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.1.2 CreatePrivateRegistration

The following element from the PrivateRegistrationProduct.dtd defines the CreatePrivateRegistration Request:

```
<!ELEMENT CreatePrivateRegistration ( DomainName+,PurchasePeriod ) >
```

The **CreatePrivateRegistration** Request should be issued to activate Private Registration for a domain previously registered with Network Solutions.

Table 15. Required and Optional Fields for the CreatePrivateRegistration Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
CreatePrivateRegistration			
DomainName	Required	YES	A maximum of 1000 domain names are allowed.
PurchasePeriod	Required	NO	

Example – CreatePrivateRegistration

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>

```

```
<Password>customer</Password>
</SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <CreatePrivateRegistration>
    <DomainName>example.com</DomainName>
    <PurchasePeriod>1</PurchasePeriod>
  </CreatePrivateRegistration>
</Product>
</Body>
</OrderRequest>
```

Expected Response – CreatePrivateRegistration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278520</ReferenceNumber>
    <Timestamp>2006-08-16T18:32:25EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>78363038</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>186032988</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>38734840</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <Lineltem>
        <LineltemID>1</LineltemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Private Registration</ProductType>
        <ProductID>186944969</ProductID>
        <Name>example.com</Name>
        <LineltemDetail>
          <DomainLineltemDetail>
            <DomainReservationStatus DomainReservationStatusType="Queued"/>
          </DomainLineltemDetail>
        </LineltemDetail>
        <Price>0.0</Price>
      </Lineltem>
      <Lineltem>
        <LineltemID>2</LineltemID>
        <Status>
          <StatusCode>7004</StatusCode>
```

```

    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.186944966</ProductID>
  <DomainName>example.com</DomainName>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Queued"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T18:32:33EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.1.3 CreateParking

The following element from the OrderRequest.dtd defines the CreateParking Request:

```

<!ELEMENT CreateParking ( DomainName+, PurchasePeriod, TechContact?, AdminContact?,
BillContact?, BulkWhoisOptOut?, SupplementalRegistryData?, WebForwardingProduct*,
ADNS* ) >

```

The **CreateParking** Request should be issued to create a domain that will be hosted and managed by Network Solutions. This domain will resolve to an "Under Construction" page until the domain owner modifies the domain by either converting it to another type of domain, or by adding value-added products. Although TechContact is optional as of Version 3.1, this field should not be included in the request. Any data submitted for TechContact will be ignored, and Network Solutions will be assigned as the technical contact.

ADNS

The Action tag specified with the ANDS tag is limited to "Add" and "Delete". The values "Add" and "Delete" specifies the action to perform on the resource record.

Note: An ADNS domain cannot have more than a combined total of 10 resource records (Total CName Records + Total A Records + Total MX Records).

Table 16. Required and Optional Fields for the CreateParking Request. contains a detailed explanation of each field in the request.

Table 16. Required and Optional Fields for the CreateParking Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
CreateParking			
DomainName	Required	YES	A maximum of 1000 domain names are allowed.
PurchasePeriod	Required	NO	
TechContact	Optional	NO	Network Solutions will be the technical contact.
AdminContact	Optional	NO	If AdminContact is not provided it defaults to the UserID of the customer.
BillContact	Optional	NO	If BillContact is not provided it defaults to the Partner's billing information.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
	Optional	NO	See Appendix E for Details. Required to be set for

Field	Required	Multiple	Notes
SupplementalRegistryData			any Individual or Business that may be a registrant/owner of a domain in the .us tld. This supplemental data may alternatively be provided with a CreateIndividual, CreateBusiness, ModifyIndividual, or ModifyBusiness. If provided here, with a CreateParking, then the data provided will be saved and associated with the Individual or Business identified by the CustomerID associated with this request. If the CreateParking request contains a ccTLD domain name and this supplemental data is not already set for the indicated CustomerID and is not provided here then the ccTLD domain names in this request will not be accepted.
PrivateRegistrationDomains	Optional	YES	Current domain name(s) being registered
WebForwardingProduct	Optional	YES	See Appendix E for Details
ADNS	Optional	YES	See Appendix E for Details

Example – CreateParking Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateParking>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
      </CreateParking>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response – CreateParking

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278510</ReferenceNumber>
    <Timestamp>2006-08-16T18:40:08EDT</Timestamp>
```

```

</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278510</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>1000309150</OrderID>
    <OrderAction>Create</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>17.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <ProductID>WN.D.226564378</ProductID>
      <DomainName>example.com</DomainName>
      <LinItemDetail>
        <DomainLinItemDetail>
          <DomainReservationStatus DomainReservationStatusType="Queued"/>
        </DomainLinItemDetail>
      </LinItemDetail>
      <Price>17.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>2</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Parking</ProductType>
      <ProductID>WN.H.226564379</ProductID>
      <DomainName>example.com</DomainName>
      <Price>0.0</Price>
    </LinItem>
    <OrderSubmissionDate>2006-08-16T18:40:16EDT</OrderSubmissionDate>
    <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – CreateParking Request (with Private Registration)

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>

```

```
<RequestHeader>
  <VERSION_5_0></VERSION_5_0>
  <Authentication>
    <PartnerID>123</PartnerID>
    <PartnerPassword>partner</PartnerPassword>
  </Authentication>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <Product>
    <CreateParking>
      <DomainName>example.com</DomainName>
      <PurchasePeriod>1</PurchasePeriod>
      <PrivateRegistrationDomains>example.com</PrivateRegistrationDomains>
    </CreateParking>
  </Product>
</Body>
</OrderRequest>
```

Expected Response – CreateParking (with Private Registration)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278510</ReferenceNumber>
    <Timestamp>2006-08-16T18:45:21EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278510</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>186028270</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
```

```
<ProductType>Registration</ProductType>
<ProductID>WN.D.186941299</ProductID>
<DomainName>example.com</DomainName>
<LinItemDetail>
  <DomainLinItemDetail>
    <DomainReservationStatus DomainReservationStatusType="Queued"/>
  </DomainLinItemDetail>
</LinItemDetail>
<Price>0.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Parking</ProductType>
  <ProductID>WN.H.186941300</ProductID>
  <DomainName>example.com</DomainName>
  <Price>0.0</Price>
</LinItem>
<LinItem>
  <LinItemID>3</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Private Registration</ProductType>
  <ProductID>186941301</ProductID>
  <Name>example.com</Name>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Queued"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>20.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T18:45:28EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>3</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>
```

Example – CreateParking Request (with WebForwarding)

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
```

```
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>123</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <Product>
    <CreateParking>
      <DomainName>example.com</DomainName>
      <PurchasePeriod>1</PurchasePeriod>
      <PrivateRegistrationDomains>example.com</PrivateRegistrationDomains>
      <WebForwardingProduct>
        <DomainName>example.com</DomainName>
        <DestURL>http://www.microsoft.com</DestURL>
        <Masking>true</Masking>
      </WebForwardingProduct>
    </CreateParking>
  </Product>
</Body>
</OrderRequest>
```

Expected Response – CreateParking (with WebForwarding)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145513</ReferenceNumber>
    <Timestamp>2006-08-16T18:53:52EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91145513</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000843496</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>39425258</UserID>
      </Customer>
      <Price>25.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191276044</ProductID>
        <DomainName>EXAMPLE.COM</DomainName>
```

```
<LinItemDetail>
  <DomainLinItemDetail>
    <DomainReservationStatus DomainReservationStatusType="Queued"/>
  </DomainLinItemDetail>
</LinItemDetail>
<Price>17.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Parking</ProductType>
  <ProductID>WN.H.191276045</ProductID>
  <DomainName>EXAMPLE.COM</DomainName>
  <Price>0.0</Price>
</LinItem>
<LinItem>
  <LinItemID>3</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Private Registration</ProductType>
  <ProductID>191276046</ProductID>
  <Name>EXAMPLE.COM</Name>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Queued"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>5.0</Price>
</LinItem>
<LinItem>
  <LinItemID>4</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191276047</ProductID>
  <Name>EXAMPLE.COM</Name>
  <Price>3.0</Price>
</LinItem>
<LinItem>
  <LinItemID>5</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191276047</ProductID>
  <Name>EXAMPLE.COM</Name>
  <Price>0.0</Price>
</LinItem>
```

```

<LinItem>
  <LinItemID>6</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191276047</ProductID>
  <Name>EXAMPLE.COM</Name>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T18:54:01EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>6</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>6</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – CreateParking Request (with ADNS)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword> pass </PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>123</UserID>
        <Password>pass</Password>
      </SystemUser>
      </AuthorizedAgent>
      <CustomerID>789</CustomerID>
    <Product>
      <CreateParking>
        <DomainName>example.com</DomainName>
        <PurchasePeriod> 2 </PurchasePeriod>
        <ADNS>
          <DomainName>example.com</DomainName>
          <ResourceRecords>
            <Action>Add</Action>
            <A>
              <RecordName>math.example.com </RecordName>
              <IPAddress>3.4.6.6</IPAddress>
            </A>
            <CNAME>
              <RecordName>undergradMathDepartment.example.com </RecordName>
              <CanonicalName>math.example.com</CanonicalName>
            </CNAME>
          </ResourceRecords>
        </ADNS>
      </CreateParking>
    </Product>
  </Body>
</OrderRequest>

```

```
<MX>
  <RecordName>example.com</RecordName>
  <Preference>10</Preference>
  <Exchange>main.example.com</Exchange>
</MX>
</ResourceRecords>
</ADNS>
</CreateParking>
</Product>
```

```
</Body>
```

```
</OrderRequest>
```

Expected Response – CreateParking (with ADNS)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
```

```
<OrderResponse>
```

```
<ResponseHeader>
```

```
<VERSION_5_0></VERSION_5_0>
```

```
<ReferenceNumber>91147611</ReferenceNumber>
```

```
<Timestamp>2006-08-16T18:57:05EDT</Timestamp>
```

```
</ResponseHeader>
```

```
<Body>
```

```
<Order>
```

```
<ReferenceNumber>91147611</ReferenceNumber>
```

```
<Status>
```

```
<StatusCode>7000</StatusCode>
```

```
<Description>Order submitted successfully and is in pending status</Description>
```

```
</Status>
```

```
<OrderID>1000920711</OrderID>
```

```
<OrderAction>Create</OrderAction>
```

```
<Customer>
```

```
<UserID>789</UserID>
```

```
</Customer>
```

```
<Price>34.0</Price>
```

```
<TieredPrice Selection="False"/>
```

```
<LinItem>
```

```
<LinItemID>1</LinItemID>
```

```
<Status>
```

```
<StatusCode>7004</StatusCode>
```

```
<Description>Line item submitted successfully</Description>
```

```
</Status>
```

```
<ProductType>Registration</ProductType>
```

```
<ProductID>WN.D.191338658</ProductID>
```

```
<DomainName>EXAMPLE.COM</DomainName>
```

```
<LinItemDetail>
```

```
<DomainLinItemDetail>
```

```
<DomainReservationStatus DomainReservationStatusType="Queued"/>
```

```
</DomainLinItemDetail>
```

```
</LinItemDetail>
```

```
<Price>34.0</Price>
```

```
</LinItem>
```

```
<LinItem>
```

```
<LinItemID>2</LinItemID>
```

```
<Status>
```

```
<StatusCode>7004</StatusCode>
```

```

    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Advanced DNS</ProductType>
  <ProductID>WN.H.191338659</ProductID>
  <Name>EXAMPLE.COM</Name>
  <Price>0.0</Price>
</LineItem>
<OrderSubmissionDate>2006-08-16T18:57:14EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.1.4 Create WebForwarding

The following element from the OrderRequest.dtd defines the Create Webforwarding action:

```
<!ELEMENT CreateWebForwarding ( WebForwardingProduct+, PurchasePeriod ) >
```

The **CreateWebForwarding** Request should be issued to forward an existing domain to a URL specified in the DestURL tag. Allowable protocols for the URL are http, https, and ftp. Table 17. Required and Optional Fields for the CreateWebForwarding Request. contains a detailed explanation of each field in the request.

Table 17. Required and Optional Fields for the CreateWebForwarding Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
CreateWebForwarding			
WebForwardingProduct	Required	YES	See WebForwardingProduct DTD for details
PurchasePeriod	Required	NO	

Example – CreateWebForwarding Request

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6 />
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>

```

```
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <CreateWebForwarding>
    <WebForwardingProduct>
      <DomainName>example.com</DomainName>
      <DestURL>http://www.microsoft.com</DestURL>
      <Masking>true</Masking>
    </WebForwardingProduct>
    <PurchasePeriod>1</PurchasePeriod>
  </CreateWebForwarding>
</Product>
</Body>
</OrderRequest>
```

Expected Response - CreateWebForwarding

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91148496</ReferenceNumber>
    <Timestamp>2006-08-16T19:01:14EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91148496</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000965388</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>3.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191358256</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LineItem>
      <LineItem>
        <LineItemID>2</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
```

```

</Status>
<ProductType>Advanced DNS</ProductType>
<ProductID>WN.H.191358257</ProductID>
<Name>example.com</Name>
<Price>0.0</Price>
</LineItem>
<LineItem>
  <LineItemID>3</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191358258</ProductID>
  <Name>example.com</Name>
  <Price>3.0</Price>
</LineItem>
<LineItem>
  <LineItemID>4</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191358258</ProductID>
  <Name>example.com</Name>
  <Price>0.0</Price>
</LineItem>
<LineItem>
  <LineItemID>5</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Web Forwarding</ProductType>
  <ProductID>WN.F.191358258</ProductID>
  <Name>example.com</Name>
  <Price>0.0</Price>
</LineItem>
<OrderSubmissionDate>2006-08-16T19:01:22EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>5</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>5</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.1.5 CreateHost

The following element from the *OrderRequest.dtd* defines the Create Host action:

```
<!ELEMENT CreateHost ( NewHost+ ) >
```

The **CreateHost** Request should be issued to create a host on a domain owned by the requestor. This host will be able to resolve DNS queries. Table 18 contains a detailed explanation of each field in the request.

Deprecation Note: The HostCoordinator element found within the NewHost element has been deprecated as of version 3.3. Please do not use this element. It will be removed in a future version.

Table 18. Required and Optional Fields for the CreateHost Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
CreateHost			
NewHost	Required	YES	
HostName	Required	NO	
HostIP1	Required	NO	
HostIP2	Optional	NO	
HostIP3	Optional	NO	
HostIP4	Optional	NO	
HostIP5	Optional	NO	

Example – CreateHost Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateHost>
        <NewHost>
          <HostName>dns.example.com</HostName>
          <HostIP1>1.2.3.4</HostIP1>
        </NewHost>
      </CreateHost>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - CreateHost

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25668594</ReferenceNumber>
    <Timestamp>2006-08-16T19:03:45EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
```

```

<ReferenceNumber>25668594</ReferenceNumber>
<Status>
  <StatusCode>7000</StatusCode>
  <Description>Order submitted successfully and is in pending status</Description>
</Status>
<OrderID>135049615</OrderID>
<OrderAction>Create</OrderAction>
<Customer>
  <UserID>789</UserID>
</Customer>
<Price>0.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Host</ProductType>
  <Name>dns.example.com</Name>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-16T19:03:52EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.1.6 CreateSSL

The following element from the *OrderRequest.dtd* defines the Create SSL action:

```
<ELEMENT CreateSSL (SSLProductName, (CertProduct+ | SealProduct+), PurchasePeriod, SubscribersAgreement, AccuracyAgreement?, EVSubscriber?)>
```

The **CreateSSL** Request should be issued to create Secured Socket Layer products (certificates or site confirm seal). Up to 20 similar products can be created per request.

Prior to submitting the CreateSSL request, partner must agree to Network Solutions Purchase Agreement. Please refer <ftp://ftp.networksolutions.com/partners/Agreements/> site for the latest copies.

For Server Selection ID, please see Appendix F for the complete list of ID and corresponding server software name.

For detailed description of error codes, please see section B5 OrderRequest DTD.

For detailed description of validation error related to the SSL Product, please see section B10.3 Validation Codes.

Table 18. Required and Optional Fields for the CreateSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
CreateSSL			
SSLProductName	Required	NO	Selection: Basic, Pro ,WildCard, SiteConfirmSeal.

	Required	Multiple	Notes
			Note: if the product selected is Basic, the CSR entered should be for a basic domain, and not for a wild card domain.
CertProduct	Optional	YES	
CSRData	Required	NO	
CSR		NO	Certificate Signing Request w/ domain name and company name.
ServerSelectionID	Required	NO	Enter the ID of the selected software. See Appendix F for a list of server ID and corresponding software name.
Contacts	Required	NO	
ValidationContact	Required	NO	Contact information that will be used for verification
EntityType	Required	NO	Selection: Individual, OrganizationCompany, EducationalInstitution or GovernmentOrganization Selection for EV services: OrganizationCompany, EducationalInstitution or GovernmentOrganization. An error code will return if "Individual" is selected for EV services.
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	
PostalCode	Required	NO	
CountryCode	Required	NO	
Department	Optional	NO	Entering data in this field will overwrite the department name (orgUnitName) in the CSR.
CompanyName	N/A	NO	If purchasing a Certificate, then this field should be provided in the CSR.
ContactFirstName	Required	NO	
ContactLastName	Required	NO	
ContactPhone	Required	NO	
DUNS	Optional	NO	Company Registration Name, limited to 20 characters
AssumedName	Optional	NO	Assumed name
JurisdictionOfIncorporation	Required for EV	NO	
City	Optional	NO	
State	Optional	NO	Required if country is US
CountryCode	Required for EV	NO	
IncorporatingAgency	Required for EV	NO	
RegistrationNumber	Optional	NO	
MainPhoneNumber	Required for EV	NO	
OrderContact	Required	NO	Partner contact information.
FirstName	Required	NO	
LastName	Required	NO	
Phone	Required	NO	

	Required	Multiple	Notes
Email	Required	NO	
CertificateRequester			Certificate Request information is required for EV Services
FirstName	Required	NO	
LastName	Required		
Title	Required		
RelationshipToApplicant	Required		Relationship to applicant
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	Valid US State if Country Code is US
PostalCode	Required	NO	
CountryCode	Required	NO	
Phone	Required	NO	
Email	Required	NO	
CertificateApprover			Certificate Approver information is required for EV Services
FirstName	Required	NO	
LastName	Required		
Title	Required		
RelationshipToApplicant	Required		Relationship to applicant
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	Valid US State if Country Code is US
PostalCode	Required	NO	
CountryCode	Required	NO	
Phone	Required	NO	
Email	Required	NO	
ContractSigner			Contract Signer information is required for EV Services
FirstName	Required	NO	
LastName	Required		
Title	Required		
RelationshipToApplicant	Required		Relationship to applicant
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	Valid US State if Country Code is US
PostalCode	Required	NO	
CountryCode	Required	NO	
Phone	Required	NO	
Email	Required	NO	

	Required	Multiple	Notes
SealProduct	Optional	YES	
DomainName	Required		Domain name, URL or IP Address
ValidationContact	Required	NO	Contact information that will be used for verification
EntityType	Required	NO	Selection for Site Confirm Seals: Individual, OrganizationCompany, EducationalInstitution or GovernmentOrganization
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	
PostalCode	Required	NO	
CountryCode	Required	NO	
Department	Optional	NO	
CompanyName	Required	NO	Required for Seal product for all Entity types.
ContactFirstName	Required	NO	
ContactLastName	Required	NO	
ContactPhone	Required	NO	
DUNS	Optional	NO	Company Registration Name
OrderContact	Required	NO	Partner contact information.
FirstName	Required	NO	
LastName	Required	NO	
Phone	Required	NO	
Email	Required	NO	
PurchasePeriod	Required	NO	1,2 3, or 4 years for SSL Certificates or Site Confirm Seals 1,2 years for EV Certificates
SubscribersAgreement	Required	NO	True or False
AccuracyAgreement	Optional	NO	Required for EV Certificates True or False
EVSubscriber	Optional		Required for EV Certificates
FirstName	Required	NO	
LastName	Required	NO	

Example 1: WildCard Certs Pricing Request - CreateSSL

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234</PartnerID>
      <PartnerPassword>pass</PartnerPassword>
    </Authentication>
    <SystemInfo>Netsol-PartnerPortal-[11.1.0]</SystemInfo>
    <RequestProperty>
```

```

    <PropertyName>EncryptedPassword</PropertyName>
    <PropertyValue>False</PropertyValue>
  </RequestProperty>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
<UserID>1234</UserID>
<Password>pass</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>40280168</CustomerID>
  <Billing>
    <IsPricingRequest/>
  </Billing>
  <Product>
    <CreateSSL>
<SSLProductName SSLProductName="Wildcard"/>
<CertProduct>
    <CSRData>
      <CSR>
MIICFTCCAX4CAQAwwgaoxCzAJBgNVBAYTAnVzMQswCQYDVQQIEwJuYzESMBAGA1UE
BxMJaGlnaHBvaW50MRwwGgYDVQQKEExNmZWdjZXJ0LXdpcGRjYXJkIGNvMQ4wDAYD
VQQLEwVzYWxlczEmMCQGA1UEAxQdKi5mZWdjZXJ0LXdpcGRjYXJkLTAxMDktMS5j
b20xJDAiBkgqhkiG9w0BCQEFWFhlgVuQGZIZ2NlcnR0ZXN0LmNvbTCBnzANBgnkq
hkiG9w0BAQEFAAOBjQAwgYkCgYEAAnlpxmFjXNL/v9pAzL5HpcjJnvsoOfX0xtbrp
mIL8IMRnmC4IAmkVZhxzTOM2CY72tpctFIKc0iT5b4F8nbPC9/vEo8P5FScF7fN
0VJv6WJTRKIJxjCdtSWLcJUzujwLyhjNhFvh+9I5clvs8e/WFQ1IHMaPYGzVx7
CSov8s0CAwEAaAqMBMGCSqGSib3DQEJAjEGEwR0ZXN0MBMGCSqGSib3DQEJBzEG
EwR0ZXN0MA0GCSqGSib3DQEBBQUAA4GBAANlu9y3sMmuxDAWeN5xOfis2BZS8wic
8pOQoBp3tmOpb1iYdzeGTYX47SG71Hm5pffnXkSGKMQwsEGmBrsSxgbLYZk+mtwx
ueo44WeLKPL3yagWO83yIMrFY0nqlhZDJBIVxcaCRYeud8vGkaKgOSzlp/CRNNK
r7NwHgabMTdS
      </CSR>
    </CSRData>
    <ServerSelectionID>2</ServerSelectionID>
  </Contacts>
  <ValidationContact>
    <EntityType Selection="OrganizationCompany"/>
    <AddressLine1>1234 main st</AddressLine1>
    <POBox>200</POBox>
    <City>Herndon</City>
    <State>VA</State>
    <PostalCode>20171</PostalCode>
    <CountryCode>US</CountryCode>
    <Department>Advertising</Department>
    <ContactFirstName>validation contact first</ContactFirstName>
    <ContactLastName> validation contact last</ContactLastName>
    <ContactPhone>7030000000</ContactPhone>
    <DUNS>9978</DUNS>
  </ValidationContact>
  <OrderContact>
    <FirstName>partners contact first name</FirstName>
    <LastName> partners contact last name</LastName>
    <Phone>7030000000</Phone>
    <Email>contact@example-name.com</Email>
  </OrderContact>
</Body>
</RequestHeader>

```

```
</OrderContact>
</Contacts>
</CertProduct>
<PurchasePeriod>2</PurchasePeriod>
<SubscribersAgreement Selection="True"/>
</CreateSSL>
</Product>
</Body>
</OrderRequest>
```

Example 1 Expected Response - Wild Card Certs Pricing Request - CreateSSL

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790847</ReferenceNumber>
    <Timestamp>2007-01-18T14:26:28EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790847</ReferenceNumber>
      <Status>
        <StatusCode>7012</StatusCode>
        <Description>Command completed successfully</Description>
      </Status>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>799.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>SSL</ProductType>
        <DomainName>*.fegcert-wildcard-0109-1.com</DomainName>
        <Name>SSL_WILDCARD</Name>
        <Price>799.0</Price>
      </LineItem>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>
```

Example 2: Site Confirm Seals purchase

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID> 1234 </PartnerID>
      <PartnerPassword> pass </PartnerPassword>
    </Authentication>
    <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="partner">
      <SystemUser>
        <UserID> 1234 </UserID>
        <Password> pass </Password>
      </SystemUser>
    </AuthorizedAgent>

    <CustomerID>40280168 </CustomerID>
    <Billing>
      <AutoRenew>false</AutoRenew>
    </Billing>
    <Product>
    <CreateSSL>
      <SSLProductName SSLProductName="SiteConfirmSeal"/>
      <SealProduct>
        <DomainName> www.fegseal-011-1.net</DomainName>
        <Contacts>
          <ValidationContact>
            <EntityType Selection="EducationalInstitution"/>
            <AddressLine1>1334 Main St.</AddressLine1>
            <AddressLine2/>
            <AddressLine3/>
            <POBox>109</POBox>
            <City>Herndon</City>
            <State>VA</State>
            <PostalCode>22041</PostalCode>
            <CountryCode>US</CountryCode>
            <Department>English</Department>
            <CompanyName>Union College</CompanyName>
            <ContactFirstName>validation contact first</ContactFirstName>
            <ContactLastName> validation contact last</ContactLastName>
            <ContactPhone>7030000000</ContactPhone>
            <DUNS>9978</DUNS>
          </ValidationContact>
          <OrderContact>
            <FirstName>partners contact first name</FirstName>
            <LastName> partners contact last name</LastName>
            <Phone>7031234567</Phone>
            <Email>contact@example-name.com</Email>
          </OrderContact>
        </Contacts>
      </SealProduct>
    </CreateSSL>
  </Product>
</Body>
</OrderRequest>
```

```
    </Contacts>
  </SealProduct>
  <PurchasePeriod>1</PurchasePeriod>
  <SubscribersAgreement Selection="True"/>
</CreateSSL>
</Product>
</Body>
</OrderRequest>
```

Example 2 expected response : Site Confirm Seal s purchase

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790848</ReferenceNumber>
    <Timestamp>2007-01-18T14:37:02EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790848</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>217813355</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>79.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>SSL</ProductType>
        <ProductID>WN.S.198888105</ProductID>
        <DomainName>www.fegseal-011-1.net</DomainName>
        <Name>SSL_SEAL</Name>
        <Price>79.0</Price>
      </LineItem>
      <OrderSubmissionDate>2007-01-18T14:37:05EST</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>
```

Example 3 – EV Certificate Request

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1475448 </PartnerID>
    <PartnerPassword> intTest </PartnerPassword>
  </Authentication>

  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>

</RequestHeader>

<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1475448 </UserID>
      <Password> intTest </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>50262727 </CustomerID>
  <Billing>
    <AutoRenew>>false</AutoRenew>
    <IsPricingRequest/>
  </Billing>
  <Product>
<CreateSSL>
  <SSLProductName SSLProductName="EV"/>
  <CertProduct>
    <CSRData>
      <CSR>
MIIB+DCCAWECAQAwY0xCzAJBgNVBAYTAnVzMQswCQYDVQQIEwJ2YTEQMA4GA1UE
BxMHaGVybmRvbjEPMA0GA1UEChMGZmVnIjAwNy5uZXQxIjAgBgkqhkiG9w0BCQEW
E2h1bGVuQHNv
bWVlbWFpbC5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK+DKhZLrkaU
YqJpCAA8ceV4Iyn/PbOBQit1oXUie9IemiFIQYdOzetrWQcWQfWIPhCgVb2HYXFf
v3a2X3nnpik2SqnyucYuTb4UjClxh5Exbmzkl5gQvmatJwNtDYijxbY8u5IXPMzR
3B1YOoKURWzNA/M8ri0Tccw/iGS9DbhTAgMBAAGgKjATBgkqhkiG9w0BCQIxhME
dGVzdDATBgkqhkiG9w0BCQcxhMEdGVzdDANBgkqhkiG9w0BAQUFAAOBgQBqaUNg
kJfNbU+SxRJOFlx5Q+NWMWwktEpsHhciGGN07Ncq6NSGPJ+AzUBC2ID0Dnu9dLU2
m3V5yKwToF5CPsC6Tz+03U5eyoUJz8SsAYfzPrR33P3M4DsPz8kHaj1MzUlzCmOq
BJu38A64WZp1Lxk49+mjf8rPCHvHHQF+KYqqLQ==
      </CSR>
      <ServerSelectionID>20</ServerSelectionID>
    </CSRData>
  <Contacts>
    <ValidationContact>
      <EntityType Selection="EducationalInstitution"/>
      <AddressLine1>1334 mclearen rd</AddressLine1>
    </ValidationContact>
  </Contacts>
</CreateSSL>
</Product>
</Body>

```

```
<AddressLine2/>
<AddressLine3/>
<POBox>400</POBox>
<City>herndon</City>
<State>VA</State>
<PostalCode>22071</PostalCode>
<CountryCode>US</CountryCode>
<Department>English</Department>
<ContactFirstName>contactFirstName</ContactFirstName>
<ContactLastName>contactLastName</ContactLastName>
<ContactPhone>222-668-4749</ContactPhone>
<DUNS>88888</DUNS>
<AssumedName>EDU</AssumedName>
<JurisdictionOfIncorporation>
  <City>Naples</City>
  <State>FL</State>
  <CountryCode>US</CountryCode>
</JurisdictionOfIncorporation>
<IncorporatingAgency>EDU1</IncorporatingAgency>
<RegistrationNumber>1234</RegistrationNumber>
<MainPhoneNumber>000-111-2222</MainPhoneNumber>
</ValidationContact>
<OrderContact>
  <FirstName>ordercontactFirstName</FirstName>
  <LastName>ordercontactLastName</LastName>
  <Phone>703111-2222</Phone>
  <Email>ordercontact@netsol.com</Email>
</OrderContact>
<CertificateRequester>
  <FirstName>crFirstName</FirstName>
  <LastName>crLastName</LastName>
  <Title>big cheese</Title>
  <RelationshipToApplicant> partner</RelationshipToApplicant>
  <AddressLine1>13200 mcclarend rd</AddressLine1>
  <AddressLine2/>
  <AddressLine3/>
  <POBox>500</POBox>
  <City>herndon</City>
  <State>va</State>
  <PostalCode> 20071</PostalCode>
  <CountryCode>US</CountryCode>
  <Phone>000-111-2222</Phone>
  <Email>cr@someemail.com</Email>
</CertificateRequester>
<CertificateApprover>
  <FirstName>caFirstName</FirstName>
  <LastName>caLastName</LastName>
  <Title>big cheese</Title>
  <RelationshipToApplicant> partner</RelationshipToApplicant>
  <AddressLine1>13200 mcclarend rd</AddressLine1>
  <AddressLine2/>
  <AddressLine3/>
  <POBox>500</POBox>
  <City>herndon</City>
  <State>va</State>
  <PostalCode> 20071</PostalCode>
```

```

    <CountryCode>US</CountryCode>
    <Phone>000-111-2222</Phone>
    <Email>ca@someemail.com</Email>
  </CertificateApprover>
  <ContractSigner>
    <FirstName>csFirstName</FirstName>
    <LastName>csLastName</LastName>
    <Title>big cheese</Title>
    <RelationshipToApplicant> partner</RelationshipToApplicant>
    <AddressLine1>13200 mcclarend rd</AddressLine1>
    <AddressLine2/>
    <AddressLine3/>
    <POBox>500</POBox>
    <City>herndon</City>
    <State>va</State>
    <PostalCode> 20071</PostalCode>
    <CountryCode>US</CountryCode>
    <Phone>000-111-2222</Phone>
    <Email>cs@someemail.com</Email>
  </ContractSigner>
</Contacts>
</CertProduct>
<PurchasePeriod>2</PurchasePeriod>
<SubscribersAgreement Selection="True"/>
<AccuracyAgreement Selection="True"/>
<EVSubscriber>
  <FirstName>EVFirstname </FirstName>
  <LastName>EVLastname </LastName>
</EVSubscriber>
</CreateSSL>
  </Product>
</Body>
</OrderRequest>
```

Example 3 Expected Response - EV Certificate Request

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14415444</ReferenceNumber>
    <Timestamp>2007-03-09T14:01:27EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14415444</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000723401</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>50262727</UserID>
      </Customer>
      <Price>1098.0</Price>
```

```

<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>SSL</ProductType>
  <ProductID>WN.C.227036182</ProductID>
  <DomainName>feg-ev-02202007.net</DomainName>
  <Name>SSL_EV</Name>
  <Price>1098.0</Price>
</LinItem>
<OrderSubmissionDate>2007-03-09T14:03:23EST</OrderSubmissionDate>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.2 Modify Request

The following requests should be issued to modify a domain.

2.2.2.1 ModifyRegistration

The following element from the *OrderRequest.dtd* defines the *Modify Registration* action:

```

<!ELEMENT ModifyRegistration ( DomainName+, TechContact?, AdminContact?,
BulkWhoisOptOut?, BillContact?, Hosts? ) >

```

The **ModifyRegistration** Request should be issued to modify an existing domain registration. Only those fields provided in XML will be modified. Any field not provided will not be modified.

Table 19. Required and Optional Fields for the *ModifyRegistration* Request. contains a detailed explanation of each field in the request.

Note: Both *HostIP* and *HostName* are required for hosts where the TLD of the host is not a TLD that the Partner Protocol currently supports. Please refer to section 1.4 for a detailed list of supported TLDs.

Table 19. Required and Optional Fields for the ModifyRegistration Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
ModifyRegistration			
DomainName	Required	YES	
TechContact	Optional	NO	See Appendix E for details.
AdminContact	Optional	NO	
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
Hosts	Optional	NO	If this tag is present in the request but is empty, then all of the current hosts will be removed from the domain. In

Field	Required	Multiple	Notes
			other words, an empty list of hosts is interpreted as a request to remove all hosts from the domain. However, if this tag is not present at all in the request, then the hosts will not be modified. In other words, not sending the Hosts tag is interpreted as an indication to not modify the domain's hosting. Please see note above. See Appendix E for details.

Example – ModifyRegistration Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <ModifyRegistration>
        <DomainName>example.com</DomainName>
        <AdminContact>
          <UserID>456</UserID>
        </AdminContact>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns3.example.com</HostName>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns4.example.com</HostName>
            </Host>
          </Host2>
        </Hosts>
      </ModifyRegistration>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - ModifyRegistration

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278632</ReferenceNumber>
```

```
<Timestamp>2005-06-30T15:01:03EDT</Timestamp>
</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278632</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>1000309163</OrderID>
    <OrderAction>Modify</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>0.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns3.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>2</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Host</ProductType>
      <Name>ns4.example.com</Name>
      <Price>0.0</Price>
    </LinItem>
    <LinItem>
      <LinItemID>3</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <ProductID>WN.D.94291390</ProductID>
      <DomainName>EXAMPLE.COM</DomainName>
      <Price>0.0</Price>
    </LinItem>
    <OrderSubmissionDate>2006-08-16T19:11:53EDT</OrderSubmissionDate>
    <NumberOfRecordsReturned>3</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>
```

2.2.2.2 *ModifyPrivateRegistration*

The following element from the PrivateRegistrationProduct.dtd defines the ModifyPrivateRegistration Request:

<!ELEMENT ModifyPrivateRegistration (DomainName+)>

The **ModifyPrivateRegistration** Request should be issued to set the auto renew flag for Private Registration of a domain name registered with Network Solutions.

Table 20. Required and Optional Fields for the ModifyPrivateRegistration Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
ModifyPrivateRegistration			
DomainName	Required	YES	A maximum of 1000 domain names are allowed.

Example – ModifyPrivateRegistration Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Billing>
      <AutoRenew>true</AutoRenew>
    </Billing>
    <Product>
      <ModifyPrivateRegistration>
        <DomainName>example.com</DomainName>
      </ModifyPrivateRegistration>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response – ModifyPrivateRegistration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278520</ReferenceNumber>
    <Timestamp>2006-08-16T19:24:38EDT</Timestamp>
```

```

</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278520</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>186033136</OrderID>
    <OrderAction>Modify</OrderAction>
    <Customer>
      <UserID>38734840</UserID>
    </Customer>
    <Price>0.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Private Registration</ProductType>
      <ProductID>186944969</ProductID>
      <Name>thisdoexample.com</Name>
      <LinItemDetail>
        <DomainLinItemDetail>
          <DomainReservationStatus DomainReservationStatusType="Queued"/>
        </DomainLinItemDetail>
      </LinItemDetail>
      <Price>0.0</Price>
    </LinItem>
    <OrderSubmissionDate>2006-08-16T19:24:45EDT</OrderSubmissionDate>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.2.3 *ModifyParking*

The following element from the OrderRequest.dtd defines the ModifyParking action:

```
<!ELEMENT ModifyParking ( DomainName+, AdminContact?, BillContact?,
BulkWhoisOptOut? ) >
```

The **ModifyParking** Request should be issued to modify an existing Parked domain. Only those fields provided in XML will be modified. Any field not provided will not be modified. A Parked domain is one that is hosted by Network Solutions, therefore Network Solutions is the technical contact, so hosts and technical contact information may not be modified. Table 21. Required and Optional Fields for the ModifyParking Request.contains a detailed explanation of each field in the request.

Table 21. Required and Optional Fields for the ModifyParking Request.

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.

Field	Required	Multiple	Notes
ModifyParking			
DomainName	Required	YES	
AdminContact	Optional	NO	See Appendix E for details.
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
ModifiedParking	Optional	YES	See Appendix E for details

Example – ModifyParking Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <ModifyParking>
        <DomainName>example.com</DomainName>
        <AdminContact>
          <UserID>456</UserID>
        </AdminContact>
      </ModifyParking>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - ModifyParking

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91147695</ReferenceNumber>
    <Timestamp>2006-08-17T11:39:01EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91147695</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
      </Status>
    </Order>
  </Body>
</OrderResponse>
```

```

    <Description>Order submitted successfully and is in pending status</Description>
  </Status>
  <OrderID>1000927894</OrderID>
  <OrderAction>Modify</OrderAction>
  <Customer>
    <UserID>789</UserID>
  </Customer>
  <Price>0.0</Price>
  <TieredPrice Selection="False"/>
  <LineItem>
    <LineItemID>1</LineItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Registration</ProductType>
    <ProductID>WN.D.191338658</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </LineItem>
  <OrderSubmissionDate>2006-08-17T11:39:17EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – ModifyParking Request (modifying ADNS)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>123</UserID>
        <Password>partner</Password>
      </SystemUser>
      </AuthorizedAgent>
      <CustomerID>789</CustomerID>
    <Product>
      <ModifyParking>
        <DomainName>example.com</DomainName>
        <AdminContact><UserID>567</UserID></AdminContact>
        <ModifiedParking>
          <DomainName>example.com</DomainName>
          <UnderConstructionPage>Standard</UnderConstructionPage>
        </ModifiedParking>
      </ModifyParking>
    </Product>
  </Body>
</OrderRequest>

```

```
    </ModifiedParking>
  </ModifyParking>
</Product>
</Body>
</OrderRequest>
```

Expected Response - ModifyParking

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91147696</ReferenceNumber>
    <Timestamp>2006-08-17T11:52:10EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91147696</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000927903</OrderID>
      <OrderAction>Modify</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191338658</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>2</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Advanced DNS</ProductType>
        <ProductID>WN.H.191338659</ProductID>
        <Name>11example.com</Name>
        <Price>0.0</Price>
      </LinItem>
      <OrderSubmissionDate>2006-08-17T11:52:17EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
```

```

</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.2.4 ModifyHost

The following element from the OrderRequest.dtd defines the ModifyHost action:

```
<!ELEMENT ModifyHost ( Host+ ) >
```

The **ModifyHost** Request should be issued to change IP address information on an existing Host or to change the Host Coordinator. A host may be modified using HostName to identify the host to be modified. When IP addresses are submitted, a complete replacement of all IP addresses is done in the system. In other words, it is not possible to simply delete or add a single IP address. For example, if ns1.foo.com has HostIP1 and HostIP2 currently associated with it, and a third, valid IP address is to be added, then HostIP1-3 must be submitted.

Table 22. Required and Optional Fields for the ModifyHost Request. contains a detailed explanation of each field in the request.

Note 1: In order to modify a particular host, the customer specified in the XML must be either the owner of that host.

Deprecation Note: The HostID and HostCoordinator elements found within the Host element were deprecated as of version 3.3. Please do not use these elements. They will be removed in a future version.

Table 22. Required and Optional Fields for the ModifyHost Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
ModifyHost	Required	NO	
Host	Required	YES	
HostName	Conditional	NO	
HostIP1	Optional	NO	
HostIP2	Optional	NO	
HostIP3	Optional	NO	
HostIP4	Optional	NO	
HostIP5	Optional	NO	

Example – ModifyHost Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>

```

```

<AuthorizedAgent SystemUserType="customer">
  <SystemUser>
    <UserID>789</UserID>
    <Password>customer</Password>
  </SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <ModifyHost>
    <Host>
      <HostName>dns.example.com</HostName>
      <HostIP1>2.3.4.6</HostIP1>
    </Host>
  </ModifyHost>
</Product>
</Body>
</OrderRequest>

```

Expected Response - ModifyHost

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25668607</ReferenceNumber>
    <Timestamp>2006-08-23T16:28:46EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>25668607</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>135049619</OrderID>
      <OrderAction>Modify</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>dns.example.com</Name>
        <Price>0.0</Price>
      </LineItem>
      <OrderSubmissionDate>2006-08-23T16:29:01EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  </Body>
</OrderResponse>

```

```

    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

2.2.3 Delete Request

The following requests should be issued to delete a domain.

2.2.3.1 DeleteRegistration

The following element from the OrderRequest.dtd defines the DeleteRegistration action:

```
<!ELEMENT DeleteRegistration ( DomainName+ ) >
```

The **DeleteRegistration** Request should be issued to delete domain(s) from the Network Solutions system. It will delete an existing domain from the Network Solutions database. Table contains a detailed explanation of each field in the request.

Note: This will only delete domains purchased as CreateRegistration, it will not work on Parked domains.

Table 23. Required and Optional Fields for the DeleteRegistration Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
DeleteRegistration	Required	NO	
DomainName	Required	YES	

Example – DeleteRegistration Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <DeleteRegistration>
        <DomainName>example.com</DomainName>
      </DeleteRegistration>
    </Product>
  </Body>
</OrderRequest>

```

Expected Response - DeleteRegistration

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278679</ReferenceNumber>
    <Timestamp>2006-08-17T11:36:13EDT</Timestamp>

```

```

</ResponseHeader>
<Body>
  <Order>
    <ReferenceNumber>14278679</ReferenceNumber>
    <Status>
      <StatusCode>7000</StatusCode>
      <Description>Order submitted successfully and is in pending status</Description>
    </Status>
    <OrderID>1000309175</OrderID>
    <OrderAction>Delete</OrderAction>
    <Customer>
      <UserID>789</UserID>
    </Customer>
    <Price>0.0</Price>
    <TieredPrice Selection="False"/>
    <LinItem>
      <LinItemID>1</LinItemID>
      <Status>
        <StatusCode>7004</StatusCode>
        <Description>Line item submitted successfully</Description>
      </Status>
      <ProductType>Registration</ProductType>
      <ProductID>WN.D.160296583</ProductID>
      <DomainName>example.com</DomainName>
      <Price>0.0</Price>
    </LinItem>
    <OrderSubmissionDate>2006-08-17T11:36:20EDT</OrderSubmissionDate>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.3.2 DeletePrivateRegistration

The following element from the PrivateRegistrationProduct.dtd defines the Delete Request:

```
<!ELEMENT DeletePrivateRegistratoin ( DomainName+ ) >
```

The **DeletePrivateRegistration** Request should be issued to remove Private Registration from a domain name that has Private Registration.

Table 24. Required and Optional Fields for the DeletePrivateRegistration Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
DeletePrivateRegistration			
DomainName	Required	YES	

Example – DeletePrivateRegistration Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>

```

```
<Authentication>
  <PartnerID>123</PartnerID>
  <PartnerPassword>partner</PartnerPassword>
</Authentication>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <Product>
<DeletePrivateRegistration>
  <DomainName>example.com</DomainName>
  </DeletePrivateRegistration>
</Product>
</Body>
</OrderRequest>
```

Expected Response – DeletePrivateRegistration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
```

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>12345678</ReferenceNumber>
    <Timestamp>2006-08-23T17:28:40EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>12345678</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>186044125</OrderID>
      <OrderAction>Delete</OrderAction>
      <Customer>
        <UserID>38734840</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Private Registration</ProductType>
        <ProductID>186950325</ProductID>
        <Name>example.com</Name>
        <LinItemDetail>
          <DomainLinItemDetail>
```

```

    <DomainReservationStatus DomainReservationStatusType="Queued"/>
  </DomainLinItemDetail>
</LinItemDetail>
  <Price>0.0</Price>
</LinItem>
  <OrderSubmissionDate>2006-08-23T17:28:49EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.3.3 DeleteParking

The following element from the OrderRequest.dtd defines the DeleteParking action:

```
<!ELEMENT DeleteParking ( DomainName+ ) >
```

The **DeleteParking** Request should be issued to delete Parked domain(s) from the Network Solutions system. It will delete an existing domain from the Network Solutions database.

Table contains a detailed explanation of each field in the request.

Note: This will only delete domains purchased as CreateParking, it will not work on Registrations domains.

Table 25. Required and Optional Fields for the DeleteParking Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
DeleteParking			
DomainName	Required	YES	

Example – DeleteParking Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <DeleteParking>
        <DomainName>example</DomainName>
      </DeleteParking>
    </Product>
  </Body>
</OrderRequest>

```

Expected Response - DeleteParking

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278681</ReferenceNumber>
    <Timestamp>2006-08-17T11:59:40EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278681</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309176</OrderID>
      <OrderAction>Delete</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.94291390</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>2</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Parking</ProductType>
        <ProductID>WN.H.226564405</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LinItem>
      <OrderSubmissionDate>2006-08-17T11:59:48EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

2.2.3.4 DeleteHost

The following element from the OrderRequest.dtd defines the DeleteHost action:

<!ELEMENT DeleteHost (HostName+) >

The **DeleteHost** Request should be issued to remove a host from the Network Solutions system. The Partner can use HostName to identify the host to delete.

Table contains a detailed explanation of each field in the request.

Note 1: This host must have NO domains dependent on it. If it does the delete will fail.

Deprecation Note: The HostID element found within the DeleteHost element has been deprecated as of version 3.3. Please do not use this element. It will be removed in a future version.

Table 26. Required and Optional Fields for the DeleteHost Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
DeleteHost			
HostName	Conditional	YES	

Example – DeleteHost Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <DeleteHost>
        <HostName>dns.example.com</HostName>
      </DeleteHost>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - DeleteHost

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25668608</ReferenceNumber>
    <Timestamp>2006-08-23T17:19:44EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>25668608</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
    </Order>
  </Body>
</OrderResponse>
```

```

<OrderID>135049620</OrderID>
<OrderAction>Delete</OrderAction>
<Customer>
  <UserID>789</UserID>
</Customer>
<Price>0.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Host</ProductType>
  <Name>dns.example.com</Name>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-23T17:19:53EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.3.5 CascadeDeleteDomain

The following element from the OrderRequest.dtd defines the CascadeDeleteDomain action:

```
<!ELEMENT CascadeDeleteDomain ( DomainName+ ) >
```

The **CascadeDeleteDomain** Request should be issued to delete any type of domain (Registration or Parking). Table contains a detailed explanation of each field in the request.

Table 27. Required and Optional Fields for the CascadeDeleteDomain Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
CascadeDeleteDomain			
DomainName	Required	YES	

Example – CascadeDeleteDomain Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>

```

```
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <CascadeDeleteDomain>
    <DomainName>example.com</DomainName>
  </CascadeDeleteDomain>
</Product>
</Body>
</OrderRequest>
```

Expected Response - CascadeDeleteDomain

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278671</ReferenceNumber>
    <Timestamp>2006-08-23T17:32:06EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278671</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309172</OrderID>
      <OrderAction>Delete</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <Lineltem>
        <LineltemID>1</LineltemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.160296583</ProductID>
        <DomainName>EXAMPLE.COM</DomainName>
        <Price>0.0</Price>
      </Lineltem>
      <Lineltem>
        <LineltemID>2</LineltemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Parking</ProductType>
        <ProductID>WN.H.160296584</ProductID>
        <DomainName>EXAMPLE.COM</DomainName>
        <Price>0.0</Price>
      </Lineltem>
      <OrderSubmissionDate>2006-08-23T17:32:17EDT</OrderSubmissionDate>
```

```

    <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
  </Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.4 Renew Request

The following requests should be issued to renew a domain.

2.2.4.1 RenewRegistration

The following element from the OrderRequest.dtd defines the RenewRegistration action:

```
<!ELEMENT RenewRegistration ( DomainName+, PurchasePeriod ) >
```

The **RenewRegistration** Request should be issued to renew a domain registration. Table 28. Required and Optional Fields for the RenewRegistration Request. contains a detailed explanation of each field in the request.

Note: *This will only renew domains purchased as CreateRegistration, it will not work for Parked domains*

Table 28. Required and Optional Fields for the RenewRegistration Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
RenewRegistration			
DomainName	Required	YES	
PurchasePeriod	Required	NO	Number of years to add to the current expiration date.

Example – RenewRegistration Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <RenewRegistration>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
      </RenewRegistration>
    </Product>
  </Body>
</OrderRequest>

```

Expected Response - RenewRegistration

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278596</ReferenceNumber>
    <Timestamp>2006-08-17T11:34:20EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278596</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309157</OrderID>
      <OrderAction>Renew</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>26.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.94291390</ProductID>
        <DomainName>EXAMPLE.COM</DomainName>
        <Price>26.0</Price>
      </LinItem>
      <OrderSubmissionDate>2006-08-17T11:34:27EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

2.2.4.2 RenewPrivateRegistration

The following element from the PrivateRegistration Product.dtd defines the RenewPrivateRegistration Request:

```
<!ELEMENT RenewPrivateRegistration ( DomainName+, PurchasePeriod ) >
```

The **RenewPrivateRegistration** Request should be issued to renew the Private Registration service for a domain name registration. Table 29. Required and Optional Fields for the RenewPrivateRegistration Request. contains a detailed explanation of each field in the request.

Table 29. Required and Optional Fields for the RenewPrivateRegistration Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
RenewPrivateRegistration			
DomainName	Required	YES	
PurchasePeriod	Required	NO	Number of years to add to the current expiration date.

Example – RenewPrivateRegistration Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <RenewPrivateRegistration>
        <DomainName>example.com</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
      </RenewPrivateRegistration>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - RenewPrivateRegistration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>78363182</ReferenceNumber>
    <Timestamp>2006-08-23T17:27:00EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>78363182</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
```

```

</Status>
<OrderID>186044204</OrderID>
<OrderAction>Renew</OrderAction>
<Customer>
  <UserID>38734840</UserID>
</Customer>
<Price>0.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Private Registration</ProductType>
  <ProductID>186950879</ProductID>
  <Name>example.com</Name>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Queued"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-23T17:27:10EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.4.3 RenewParking

The following element from the OrderRequest.dtd defines the RenewParking action:

```
<!ELEMENT RenewParking ( DomainName+, PurchasePeriod ) >
```

The **RenewParking** Request should be issued to renew a Parked domain.

Table contains a detailed explanation of each field in the request.

Note: *This will only renew domains purchased as CreateParking, it will not work for Registrations domains.*

Table 30. Required and Optional Fields for the RenewParking Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
RenewParking			
DomainName	Required	YES	
PurchasePeriod	Required	NO	Number of years to add to the current expiration date.

Example – RenewParking Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
  <Authentication>
    <PartnerID>123</PartnerID>

```

```

    <PartnerPassword>partner</PartnerPassword>
  </Authentication>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <Product>
    <RenewParking>
      <DomainName>example.com</DomainName>
      <PurchasePeriod>1</PurchasePeriod>
    </RenewParking>
  </Product>
</Body>
</OrderRequest>

```

Expected Response – RenewParking

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278609</ReferenceNumber>
    <Timestamp>2006-08-17T12:01:58EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278609</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309160</OrderID>
      <OrderAction>Renew</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>17.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.160296583</ProductID>
        <DomainName>example.com</DomainName>
        <Price>17.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>2</LinItemID>

```

```

    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Parking</ProductType>
    <ProductID>WN.H.160296584</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </LinItem>
  <OrderSubmissionDate>2006-08-17T12:01:58EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.4.4 CascadeRenewDomain

The following element from the OrderRequest.dtd defines the CascadeRenewDomain action:

```
<!ELEMENT CascadeRenewDomain ( DomainName+, PurchasePeriod ) >
```

The **CascadeRenewDomain** Request should be issued to renew any type of domain (Registration or Parking).

Table 31. Required and Optional Fields for the CascadeRenewDomain Request contains a detailed explanation of each field in the request.

Table 31. Required and Optional Fields for the CascadeRenewDomain Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
CascadeRenewDomain			
DomainName	Required	YES	
PurchasePeriod	Required	NO	Number of years to add to the current expiration date.

Example – CascadeRenewDomain Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CascadeRenewDomain>

```

```
<DomainName>example.com</DomainName>
<PurchasePeriod>1</PurchasePeriod>
</CascadeRenewDomain>
</Product>
</Body>
</OrderRequest>
```

Expected Response - CascadeRenewDomain

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278673</ReferenceNumber>
    <Timestamp>2006-08-23T17:30:30EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278673</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309173</OrderID>
      <OrderAction>Renew</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>17.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.150626296</ProductID>
        <DomainName>example.com</DomainName>
        <Price>17.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>2</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Parking</ProductType>
        <ProductID>WN.H.150626297</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LinItem>
      <OrderSubmissionDate>2006-08-23T17:30:50EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
    </Order>
```

```

    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

2.2.4.5 RenewSSL

The following element from the OrderRequest.dtd defines the RenewSSL action:

```

<ELEMENT RenewSSL (SSLProductName, SSLProductToRenew+, PurchasePeriod, SubscribersAgreement,
AccuracyAgreement?, EVSubscriber?)+>

```

The **RenewSSL** Request should be used to renew the Secured Socket Layer products (certificates or seals). Up to 20 similar products can be renewed per request (all products should be of the same type).

Prior to submitting the **RenewSSL** request, partner must agree to Network Solutions Purchase Agreement. Please refer <ftp://ftp.networksolutions.com/partners/Agreements/> site for the latest copies.

Partners can renew a SSL product that is (1) within 180 days of expiration, (2) which has expired already and has not been renewed, (3) has not been revoked or (4) isn't in the process of being renewed or reissued.

Partners can use the FindSSLProductsForPartner or FindAllSSLProductsForPartner queries in PartnerManagerRequest.dtd to find the status of each product to see if the product is renewable.

The renewed product response will return a new product ID. Please retain this product ID for further management requests in the future.

For Server Selection ID, please see Appendix F for the complete list of ID and corresponding server software name.

For detailed description of error codes, please see section B5 OrderRequest DTD.

For detailed description of validation error related to the SSL Product, please see section B10.3 Validation Codes.

Table 32. Required and Optional Fields for the RenewSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
RenewSSL			
SSLProductName	Required	NO	Indicate the name of product to renew. This is new for 4.6 Values: Basic ,Pro ,WildCard , SiteConfirmSeal ,EV
SSLProductToRenew	Required	YES	
ProductID	Optional	YES	Original product's ID. This ID is returned with the CreateSSL response or can be retrieved using the FindAllSSLProductsForPartner or FindSSLProductsForPartner queries.
DomainName	Optional	NO	Applicable to Site Confirm Seal only. If provided in the request, domain name must match the original product's. If domain name is not provided, Network Solutions will look up the domain name.
CSRData	Optional	NO	
CSR	Optional	NO	Certificate Signing Request w/ domain name and company name. CSR must be the same as the original product.

	Required	Multiple	Notes
			If not present, Network Solutions will look up the original CSR.
ServerSelectionID	Optional	NO	See table XX for a list of server selections
Contacts	Required	NO	
ValidationContact	Required	NO	Contact information that will be used for verification
EntityType	Required	NO	Selection: Individual, OrganizationCompany, EducationalInstitution or GovernmentOrganization
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	
PostalCode	Required	NO	
CountryCode	Required	NO	
Department	Optional	NO	Entering data in this field will overwrite the department name (orgUnitName) in the CSR.
CompanyName	See notes	NO	If renewing a Certificate, then this field should be provided in the CSR. If renewing a Site Confirm Seal, then this field is required.
ContactFirstName	Required	NO	
ContactLastName	Required	NO	
ContactPhone	Required	NO	
DUNS	Optional	NO	Company Registration Name, limited to 20 characters
OrderContact	Required	NO	Partner contact information. Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
CertificateRequester	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
CertificateApprover	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
ContractSigner	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
PurchasePeriod	Required	NO	1,2,3 or 4 years 1,2 years fro EV Services
SubscribersAgreement	Required	NO	True or False
AccuracyAgreement	Optional	NO	Required for EV Certificates True or False
EVSubscriber	Optional		Required for EV Certificates
FirstName	Required	NO	
LastName	Required	NO	

Example 1: Renewing a seal product

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1234 </PartnerID>
    <PartnerPassword> some password </PartnerPassword>
  </Authentication>
  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1234 </UserID>
      <Password> some password </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>40280168 </CustomerID>
  <Billing>
  </Billing>
  <Product>
<RenewSSL>
  <SSLProductName SSLProductName=" SiteConfirmSeal">
  <SSLProductToRenew>
    <ProductID> WN.S.198888105</ProductID>
    <Contacts>
      <ValidationContact>
        <EntityType Selection="EducationalInstitution"/>
        <AddressLine1>1334 Main St.</AddressLine1>
        <AddressLine2/>
        <AddressLine3/>
        <POBox>109</POBox>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>22041</PostalCode>
        <CountryCode>US</CountryCode>
        <Department>Math</Department>
        <CompanyName>Union College</CompanyName>
        <ContactFirstName>renewed contact first</ContactFirstName>
        <ContactLastName> renewed contact last</ContactLastName>
        <ContactPhone>7030000000</ContactPhone>
        <DUNS>9978</DUNS>
      </ValidationContact>
      <OrderContact>
        <FirstName>contact-firstname</FirstName>
        <LastName>contact-lastname</LastName>
        <Phone>contact-phone</Phone>
        <Email>contactFirstName@netsol.com</Email>
      </OrderContact>
    </Contacts>
  </SSLProductToRenew>
  <PurchasePeriod>2</PurchasePeriod>

```

Specify which product to renew

```

<SubscribersAgreement Selection="True"/>
  </RenewSSL>
</Product>
</Body>
</OrderRequest>

```

Example 1 expected response : Renewing a seal product

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790849</ReferenceNumber>
    <Timestamp>2007-01-18T15:46:59EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790849</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>217813471</OrderID>
      <OrderAction>Renew</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>129.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>SSL</ProductType>
        <ProductID>WN.S.198888268</ProductID>
        <DomainName>www.fegseal-011-1.net</DomainName>
        <Name>SSL_SEAL</Name>
        <Price>129.0</Price>
      </LineItem>
      <OrderSubmissionDate>2007-01-18T15:47:00EST</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

New Product ID is returned

Example 2 – Renewing an EV Service

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1475448 </PartnerID>
    <PartnerPassword> intTest </PartnerPassword>
  </Authentication>
  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
</RequestHeader>

<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1475448 </UserID>
      <Password> intTest </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>40336623 </CustomerID>
  <Billing>
  </Billing>
  <Product>
</Product>
<RenewSSL>
  <SSLProductName SSLProductName="EV"/>
  <SSLProductToRenew>
    <ProductID> WN.C.198957202 </ProductID>
    <CSRData>
      <CSR>
MIIB9DCCA0CAQAwYkxCzAJBgNVBAYTAnVzMQswCQYDVQQIEwJ2YTEQMA4GA1UE
BxMHaGVybmRvbjEQMA4GA1UEChMHZmVnIGxsYzEMMAoGA1UECjxMDZW5nMR0wGwYD
VQQDEjRmZWctZXYtMDIxNTlwMDcuaW5mbzEcMBoGCSqGSIb3DQEJARYNaGVsZW5A
ZmVnLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAwYkCgYEAu4weNFObz5FvMuOK
Anvn+hSoYgKGaAUM12qtMvz2O7ByQHMfGDPDYop39lqUGL+NejLgXRHaBKbWcQa2
+84xJAH8OacdLqVBW3u9OI4SREMSKypP42rlucOyuGkKqoJQopMXrAq+RXYK4iZZ
ttIJ6SxaEI6BaL0u9//R4WVYh8CAwEAAsAqMBMGCSqGSIb3DQEJAJEGEwR0ZXN0
MBMGCSqGSIb3DQEJBJzEGEwR0ZXN0MA0GCSqGSIb3DQEBBQUAA4GBAltQn+SGCniR
8YekXoGDmk93UDswZfiY0MoWtuApoR50m1/Du63+Tz4dRrELj1hTluluvoU65g8x
ryUGxU+XHoUoNXz7Vcw2lq8OtLQZrA/G66DPI75fJOSpHLzTtgUdsTPto4/7sdAe
YCx56zpj9xUpXaNpAtDHs4TBNUOPlwXT
      </CSR>
    <ServerSelectionID>5</ServerSelectionID>
  </CSRData>
  <Contacts>
    <ValidationContact>
      <EntityType Selection="EducationalInstitution"/>
      <AddressLine1>vc 1334 renewd rd</AddressLine1>
      <AddressLine2/>
      <AddressLine3/>
      <POBox>vcissued400</POBox>
      <City>miami</City>
      <State>FL</State>
      <PostalCode>22071</PostalCode>
    </ValidationContact>
  </Contacts>
</Body>
</OrderRequest>

```

```
<CountryCode>US</CountryCode>
<Department>dept from validation contact</Department>
<ContactFirstName>vcrenewd first name</ContactFirstName>
<ContactLastName>vcrenewd last name</ContactLastName>
<ContactPhone>703-668-4749</ContactPhone>
<DUNS>88888</DUNS>
<AssumedName>vcrenewd assumedname</AssumedName>
<JurisdictionOfIncorporation>
  <City>miami</City>
  <State>FL</State>
  <CountryCode>US</CountryCode>
</JurisdictionOfIncorporation>
<IncorporatingAgency>renewdia1</IncorporatingAgency>
<RegistrationNumber>1234</RegistrationNumber>
<MainPhoneNumber>000-123-4444</MainPhoneNumber>
</ValidationContact>
<OrderContact>
  <FirstName>ocrenewd contact-FirstName</FirstName>
  <LastName>ocrenewd contact-LastName</LastName>
  <Phone>000-111-3333</Phone>
  <Email>ocrenewd@netsol.com</Email>
</OrderContact>
<CertificateRequester>
  <FirstName>crrenewd FirstName</FirstName>
  <LastName>crrenewd LastName</LastName>
  <Title>big cheese</Title>
  <RelationshipToApplicant>crrenewd partner</RelationshipToApplicant>
  <AddressLine1>13200 renewdd rd</AddressLine1>
  <AddressLine2/>
  <AddressLine3/>
  <POBox>500</POBox>
  <City>miami</City>
  <State>va</State>
  <PostalCode>20071</PostalCode>
  <CountryCode>US</CountryCode>
  <Phone>000-111-2222</Phone>
  <Email>crrenewd@someemail.com</Email>
</CertificateRequester>
<CertificateApprover>
  <FirstName>caFirstName</FirstName>
  <LastName>caLastName</LastName>
  <Title>big cheese</Title>
  <RelationshipToApplicant> partner</RelationshipToApplicant>
  <AddressLine1>13200 mclarend rd</AddressLine1>
  <AddressLine2/>
  <AddressLine3/>
  <POBox>500</POBox>
  <City>herndon</City>
  <State>va</State>
  <PostalCode> 20071</PostalCode>
  <CountryCode>US</CountryCode>
  <Phone>000-111-2222</Phone>
  <Email>cr@someemail.com</Email>
</CertificateApprover>
<ContractSigner>
  <FirstName>csrenewdFirstName</FirstName>
```

```

        <LastName>csrenewdLastName</LastName>
        <Title>big cheese</Title>
        <RelationshipToApplicant>csrenewdpartner</RelationshipToApplicant>
        <AddressLine1>13200 renewdd rd</AddressLine1>
        <AddressLine2/>
        <AddressLine3/>
        <POBox>502</POBox>
        <City>miami</City>
        <State>va</State>
        <PostalCode>20071</PostalCode>
        <CountryCode>US</CountryCode>
        <Phone>000-111-2222</Phone>
        <Email>csrenewd@someemail.com</Email>
    </ContractSigner>
</Contacts>
</SSLProductToRenew>
<PurchasePeriod>2</PurchasePeriod>
<SubscribersAgreement Selection="True"/>
<AccuracyAgreement Selection="True"/>
    <EVSubscriber>
        <FirstName> FirstNameevfirstname</FirstName>
        <LastName> FirstNameevlastname </LastName>
    </EVSubscriber>
</RenewSSL>
</Product>
</Body>
</OrderRequest>

```

Example 2 Expected Response – Renewing an EV Service

```

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>173199116</ReferenceNumber>
    <Timestamp>2007-03-09T03:30:04EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>173199116</ReferenceNumber>
      <Status>
        <StatusCode>7012</StatusCode>
        <Description>Command completed successfully</Description>
      </Status>
      <OrderAction>Renew</OrderAction>
      <Customer>
        <UserID>41557053</UserID>
      </Customer>
      <Price>1098.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
      </LineItem>
    </Order>
  </Body>
</OrderResponse>

```

```

    <ProductType>SSL</ProductType>
    <ProductID>WN.C.236306959</ProductID>
    <DomainName>REGDONEVWHL.COM</DomainName>
    <Name>SSL_EV</Name>
    <Price>1098.0</Price>
  </LinItem>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.5 Convert Request

The following requests should be issued to convert a domain.

2.2.5.1 ConvertToRegistration

The following element from the *OrderRequest.dtd* defines the *ConvertToRegistration* action:

```
<!ELEMENT ConvertToRegistration ( DomainName+, TechContact?, AdminContact?,
BillContact?, BulkWhoisOptOut?, Hosts? ) >
```

The **ConvertToRegistration** Request should be issued to convert a Parking to Registration. A registration must have at least one host to properly resolve. Table 33. Required and Optional Fields for the ConvertToRegistration Request contains a detailed explanation of each field in the request.

Note: Both *HostIP* and *HostName* are required for hosts where the TLD of the host is not a TLD that the Partner Protocol currently supports. Please refer to section 1.4 for a detailed list of supported TLDs.

Table 33. Required and Optional Fields for the ConvertToRegistration Request

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
ConvertToRegistration			
DomainName	Required	YES	This is the domain being modified.
TechContact	Optional	NO	See Appendix E for details. If AdminContact and/or
AdminContact	Optional	NO	TechContact are not provided, the default value is the customer.
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
Hosts	Optional	NO	Please see note above. See Appendix E for details.

Example – ConvertToRegistration Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">

```

```

<SystemUser>
  <UserID>789</UserID>
  <Password>customer</Password>
</SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <ConvertToRegistration>
    <DomainName>example.com</DomainName>
    <Hosts>
      <Host1>
        <Host>
          <HostName>ns1.example.com</HostName>
        </Host>
      </Host1>
      <Host2>
        <Host>
          <HostName>ns2.example.com</HostName>
        </Host>
      </Host2>
    </Hosts>
  </ConvertToRegistration>
</Product>
</Body>
</OrderRequest>

```

Expected Response - ConvertToRegistration

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278669</ReferenceNumber>
    <Timestamp>2006-08-24T12:52:31EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278669</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309170</OrderID>
      <OrderAction>Convert</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
      </LineItem>
    </Order>
  </Body>
</OrderResponse>

```

```

    <Name>ns2.example.com</Name>
    <Price>0.0</Price>
  </Lineltem>
  <Lineltem>
    <LineltemID>2</LineltemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Host</ProductType>
    <Name>ns1.example.com</Name>
    <Price>0.0</Price>
  </Lineltem>
  <Lineltem>
    <LineltemID>3</LineltemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Registration</ProductType>
    <ProductID>WN.D.150626296</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </Lineltem>
  <Lineltem>
    <LineltemID>4</LineltemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Parking</ProductType>
    <ProductID>WN.H.150626297</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </Lineltem>
  <OrderSubmissionDate>2006-08-24T12:52:44EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>4</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>4</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.5.2 ConvertToParking

The following element from the *OrderRequest.dtd* defines the *ConvertToParking* action:

```
<!ELEMENT ConvertToParking ( DomainName+, AdminContact?, BillContact?,
BulkWhoisOptOut? ) >
```

The **ConvertToParking** Request should be issued to convert a Registration product to a Parking product. Table 34. Required and Optional Fields for the *ConvertToRegistration* Request contains a detailed explanation of each field in the request.

Table 34. Required and Optional Fields for the ConvertToRegistration Request

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.

Field	Required	Multiple	Notes
Body			See Table 13 for details.
ConvertToParking			
DomainName	Required	YES	This is the domain being modified.
AdminContact	Optional	NO	See Appendix E for details. If AdminContact and/or
BillContact	Optional	NO	TechContact are not provided, the default value is the customer.
BulkWhoisOptOut	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.

Example – ConvertToParking Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <ConvertToParking>
        <DomainName>example.com</DomainName>
      </ConvertToParking>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response – ConvertToParking Response

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278670</ReferenceNumber>
    <Timestamp>2006-08-24T12:55:31EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278670</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309171</OrderID>
      <OrderAction>Convert</OrderAction>
      <Customer>
```

```

    <UserID>789</UserID>
  </Customer>
  <Price>0.0</Price>
  <TieredPrice Selection="False"/>
  <LinItem>
    <LinItemID>1</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Registration</ProductType>
    <ProductID>WN.D.94291390</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </LinItem>
  <LinItem>
    <LinItemID>2</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Parking</ProductType>
    <ProductID>WN.H.226564405</ProductID>
    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </LinItem>
  <OrderSubmissionDate>2006-08-24T12:55:40EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.6 RNCA Request

The following requests should be issued to initiate the process of transferring a domain between two different owners within the Network Solutions Registrar.

2.2.6.1 *TransferInternalToRegistration*

The following element from the *OrderRequest.dtd* defines the *TransferInternalToRegistration* action:

```
<!ELEMENT TransferInternalToRegistration ( DomainName+, TechContact?, AdminContact?,
BillContact?, BulkWhoisOptOut?, Hosts?, SupplementalRegistryData) >
```

The **TransferInternalToRegistration** Request should be issued to initiate the process of transferring a domain from the current owner within the Network Solutions Registrar to a different owner also within the Network Solutions Registrar. Once the proper authorization steps are completed successfully, this request will result in a domain Registration product that is owned by the new owner. Neither the ProductID nor the expiration date of the domain will change. If the domain transfer request does not provide valid host information for the domain, then the existing hosts, as listed in a whois lookup, will be used. See Table 35. Required and Optional Fields for the TransferInternalToRegistration Request for more options when handling Hosts. If a TransferInternalToRegistration request is issued for a domain that is currently a Parking domain product then the domain will be converted to a Registration product. Duplicate domains names within a given request will be filtered out, resulting in only the first occurrence of the domain name actually being submitted for an internal transfer.

Table contains a detailed explanation of each field in the request.

Table 35. Required and Optional Fields for the TransferInternalToRegistration Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
TransferInternalToRegistration			
DomainName	Required	YES	A maximum of 1000 domain names are allowed per request.
TechContact	Optional	NO	If TechContact is not provided it defaults to the UserID of the customer.
AdminContact	Optional	NO	If AdminContact is not provided it defaults to the UserID of the customer.
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
Hosts	Optional	NO	If this tag is present in the request but is empty then all of the current hosts will be removed from the domain. In other words an empty list of hosts is interpreted as a request to remove all hosts from the domain. However, if this tag is not present at all in the request then the hosts will not be modified. In other words, not sending the Hosts tag is interpreted as an indication to not modify the domain's hosting. Please see note above. See Appendix E for details.

Example – TransferInternalToRegistration Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <TransferInternalToRegistration>
        <DomainName>example.com</DomainName>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.example.com</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
        </Hosts>
      </TransferInternalToRegistration>
    </Product>
  </Body>
</OrderRequest>

```

```
</Host1>
<Host2>
  <Host>
    <HostName>ns2.example.com</HostName>
    <HostIP1>2.3.4.6</HostIP1>
  </Host>
</Host2>
</Hosts>
</TransferInternalToRegistration>
</Product>
</Body>
</OrderRequest>
```

Expected Response – TransferInternalToRegistration

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>43343463</ReferenceNumber>
    <Timestamp>2006-08-24T12:59:51EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>43343463</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>150496179</OrderID>
      <OrderAction>InternalTransfer</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>ns1.example.com</Name>
      </LineItem>
      <LineItem>
        <LineItemID>2</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>ns2.example.com</Name>
      </LineItem>
      <LineItem>
        <LineItemID>3</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
```

```

    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.164207528</ProductID>
  <DomainName>example.com</DomainName>
</LineItem>
<OrderSubmissionDate>2006-08-24T13:00:01EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>3</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.6.2 TransferInternalToParking

The following element from the *OrderRequest.dtd* defines the *TransferInternalToParking* action:

```

<!ELEMENT TransferInternalToParking ( DomainName+, AdminContact?, BillContact?,
BulkWhoisOptOut?) >

```

The **InternalTransferToParking** Request should be issued to initiate the process of transferring a domain from the current owner within the Network Solutions Registrar to a different owner also within the Network Solutions Registrar. Once the proper authorization steps are completed successfully, this request will result in a domain Parking product that is associated with the new owner and dis-associated with the previous owner. Neither the ProductID nor the expiration date of the domain will change. The *TransferInternalToParking* request will not accept Registration domain products. Only domains that are currently Parked domains may be the subject of an *InternalTransferToParking*.

Duplicate domain names within a given request will be filtered out, resulting in only the first occurrence of the domain name actually being submitted for an internal transfer. Table contains a detailed explanation of each field in the request.

Table 36. Required and Optional Fields for the TransferInternalToParking

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
TransferInternalToParking			
DomainName	Required	YES	A maximum of 1000 domain names are allowed per request.
AdminContact	Optional	NO	If AdminContact is not provided it defaults to the UserID of the customer.
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
DomainProtect	Optional	NO	Default is True.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
ADNS	Optional	YES	See Appendix E for details.

Example – TransferInternalToParking Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">

```

```
<SystemUser>
  <UserID>789</UserID>
  <Password>customer</Password>
</SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <TransferInternalToParking>
    <DomainName>example.com</DomainName>
  </TransferInternalToParking>
</Product>
</Body>
</OrderRequest>
```

Expected Response - TransferInternalToParking

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>43343480</ReferenceNumber>
    <Timestamp>2006-08-24T13:06:53EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>43343480</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>150496180</OrderID>
      <OrderAction>InternalTransfer</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.412634</ProductID>
        <DomainName>example.com</DomainName>
        <Price>0.0</Price>
      </LineItem>
      <LineItem>
        <LineItemID>2</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Parking</ProductType>
        <ProductID>WN.H.412635</ProductID>
```

```

    <DomainName>example.com</DomainName>
    <Price>0.0</Price>
  </LineItem>
  <OrderSubmissionDate>2006-08-24T13:07:03EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.6.3 ModifyCustomerName

The following element from the OrderRequest.dtd defines the ModifyCustomerName action:

```

<!ELEMENT ModifyCustomerName( (FirstName, MiddleName?, LastName) | CompanyName)
>

```

The **ModifyCustomerName** Request should be issued to modify the legal name of an Individual or Business. This request is an OrderRequest rather than a UserRequest due to the fact that it requires some additional authorization steps and status tracking before it can be completely fulfilled.

Table 37. Required and Optional Fields for the ModifyCustomerName Request

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
ModifyCustomerName			
FirstName	Conditional	NO	Required if the target customer is an Individual. See Appendix E for details.
MiddleName	Optional	NO	Optional if the target customer is an Individual. See Appendix E for details.
LastName	Conditional	NO	Required if the target customer is an Individual. See Appendix E for details.
CompanyName	Conditional	NO	Required if the target customer is a Business. See Appendix E for details.

Example – ModifyCustomerName Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>456</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>456</CustomerID>
    <Product>
      <ModifyCustomerName>

```

```
<FirstName>XML</FirstName>
<MiddleName>API</MiddleName>
<LastName>Customer</LastName>
</ModifyCustomerName>
</Product>
</Body>
</OrderRequest>
```

Expected Response – ModifyCustomerName

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278638</ReferenceNumber>
    <Timestamp>2006-08-24T13:04:18EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278638</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309164</OrderID>
      <OrderAction>LegalNameChange</OrderAction>
      <Customer>
        <UserID>456</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Individual</ProductType>
        <Price>0.0</Price>
      </LinItem>
      <OrderSubmissionDate>2006-08-24T13:04:28EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>
```

2.2.7 CRA Request

The following requests should be issued to initiate the process of transferring a domain into the Network Solutions Registrar.

2.2.7.1 *InboundTransferToRegistration*

The following element from the *OrderRequest.dtd* defines the *InboundTransferToRegistration* action:

```
<!ELEMENT InboundTransferToRegistration ( ToCustomer, DomainName+, TechContact?,
AdminContact?, BillContact?, BulkWhoisOptOut?, Hosts?, AuthCodeInfo*,
SupplementalRegistryData?, PrivateRegistrationDomains* ) >
```

The **InboundTransferToRegistration** Request should be issued to initiate the process of transferring a domain into the Network Solutions Registrar. Once the proper authorization steps are completed successfully, this request will result in a domain that is registered with Network Solutions but hosted separately from Network Solutions. If the domain transfer request does not provide valid host information for the domain, then the existing hosts, as listed in a whois lookup, will be used.

As indicated in the *OrderRequest* DTD, *OrderRequests* may contain multiple *InboundTransferToRegistration* elements. This is allowed in order to support “cross customer” transfer requests. As a result, each *InboundTransferToRegistration* element may specify a different value for *ToCustomer*, and each *InboundTransferToRegistration* element will result in a separate *Order* element returned in the resulting *OrderResponse* message. When performing “cross customer” transfer requests, the *AuthorizedAgent* *SystemUserType* element must be set to “partner”. However, if it is set to “customer”, then only the *InboundTransferToRegistration* elements whose *ToCustomer* element has the same value as the *UserID* element in *AuthorizedAgent* will be processed.

It is also important to note that dependencies on “inline host creation” are only supported within a given *InboundTransferToRegistration* element. Therefore, hosts created as a result of one *InboundTransferToRegistration* element cannot be specified as hosts for a domain in a different *InboundTransferToRegistration* element within the same *OrderRequest*.

Duplicate domain names within a given request will be filtered out, resulting in only the first occurrence of the domain name actually being submitted for transfer. It is recommended to limit inbound transfers to 100 domain names per *OrderRequest*.

The Network Solutions Partner Protocol now supports automated transfers of the following tlds: **.biz, .org, .info, .us, .tv, .cc, .bz, .mobi, .us.com, .uk.com, .eu.com, .de.com, .la, .br.com, .cn.com, .gb.com, .gb.net, .hu.com, .jpn.com, .kr.com, .no.com, .qc.com, .ru.com, .sa.com, .se.com, .se.net, .uk.net, .uy.com, .za.com**

An *AuthCode* is required for the tlds listed prior.

Table 38. Required and Optional Fields for the *InboundTransferToRegistration* Request contains a detailed explanation of each field in the request.

Table 38. Required and Optional Fields for the *InboundTransferToRegistration* Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
InboundTransferToRegistration			
ToCustomer	Required	NO	The customer ID to which the domain should be transferred.
DomainName	Required	YES	A maximum of 1000 domain names are allowed per request.
TechContact	Optional	NO	If <i>TechContact</i> is not provided it defaults to the <i>UserID</i> of the customer.
AdminContact	Optional	NO	If <i>AdminContact</i> is not provided it defaults to the <i>UserID</i> of the customer.
BillContact	Optional	NO	Although <i>BillContact</i> may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
BulkWhoisOptOut	Optional	NO	Default value is ‘N’.

Field	Required	Multiple	Notes
DomainProtect	Optional	NO	Default is True.
Hosts	Optional	NO	If this tag is present in the request but is empty then all of the current hosts will be removed from the domain. In other words an empty list of hosts is interpreted as a request to remove all hosts from the domain. However, if this tag is not present at all in the request then the hosts will not be modified. In other words, not sending the Hosts tag is interpreted as an indication to not modify the domain's hosting. Please see note above. See Appendix E for details.
AuthCodeInfo	NO	YES	This tag is used to associate the domain name to the domain password (AuthCode). See usage below.
SupplementalRegistryData	NO	NO	
PrivateRegistrationDomains	NO	YES	

Example – InboundTransferToRegistration Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID></CustomerID>
  </Product>
  <InboundTransferToRegistration>
    <ToCustomer>
      <UserID>789</UserID>
    </ToCustomer>
    <DomainName>example.com</DomainName>
    <AuthCodeInfo>
      <DomainName>example.com</DomainName>
      <AuthCode>password</AuthCode>
    </AuthCodeInfo>
  </InboundTransferToRegistration>
</Product>
</Body>
</OrderRequest>
```

Expected Response - InboundTransferToRegistration

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145700</ReferenceNumber>
```

```

    <Timestamp>2006-08-24T13:11:35EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91145700</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000858215</OrderID>
      <OrderAction>InboundTransfer</OrderAction>
      <Customer>
        <UserID>39425258</UserID>
      </Customer>
      <Price>17.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191293329</ProductID>
        <DomainName>EXAMPLE.COM</DomainName>
        <Price>17.0</Price>
      </LineItem>
      <OrderSubmissionDate>2006-08-24T13:11:43EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

Example – InboundTransferToRegistration Request (.org and AuthCode Success)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="partner">
      <SystemUser>
        <UserID>123</UserID>
        <Password>partner</Password>
      </SystemUser>
    </AuthorizedAgent>
  </Body>
</OrderRequest>

```

```

    <CustomerID>789</CustomerID>
  <Product>
    <InboundTransferToRegistration>
      <ToCustomer>
        <UserID>789</UserID>
      </ToCustomer>
      <DomainName>example.org</DomainName>
    <AuthCodeInfo>
      <DomainName>example.org</DomainName>
      <AuthCode>a@b#cpad</AuthCode>
    </AuthCodeInfo>
  </InboundTransferToRegistration>
</Product>
</Body>
</OrderRequest>

```

Expected Response - InboundTransferToRegistration

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

```

```

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91146142</ReferenceNumber>
    <Timestamp>2006-08-24T13:22:52EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91146142</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000864163</OrderID>
      <OrderAction>InboundTransfer</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>17.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191300814</ProductID>
        <DomainName>EXAMPLE.ORG</DomainName>
        <Price>17.0</Price>
      </LineItem>
      <OrderSubmissionDate>2006-08-24T13:23:02EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  </Body>
</OrderResponse>

```

```

    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

Example – InboundTransferToRegistration Request (.org and AuthCode Failure)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="partner">
      <SystemUser>
        <UserID>123</UserID>
        <Password>partner</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <InboundTransferToRegistration>
        <ToCustomer>
          <UserID>789</UserID>
        </ToCustomer>
        <DomainName>example.org</DomainName>
        <AuthCodeInfo>
          <DomainName>example1.org</DomainName>
          <AuthCode>a@b#cpad</AuthCode>
        </AuthCodeInfo>
      </InboundTransferToRegistration>
    </Product>
  </Body>
</OrderRequest>

```

DomainName tag should match the DomainName tag in the AuthCodeInfo tag. This will result a Missing AuthCode error

Expected Response - InboundTransferToRegistration

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91147365</ReferenceNumber>
    <Timestamp>2006-08-24T13:19:48EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91147365</ReferenceNumber>
      <Status>
        <StatusCode>-7016</StatusCode>
        <Description>All products are invalid. Order was not submitted</Description>
      </Status>
    </Order>
  </Body>
</OrderResponse>

```

```

<OrderAction>InboundTransfer</OrderAction>
<Customer>
  <UserID>39425258</UserID>
</Customer>
<Price>0.0</Price>
<TieredPrice Selection="False"/>
<LineItem>
  <Status>
    <StatusCode>-2320</StatusCode>
    <Description>Invalid line item, validation error.</Description>
    <ValidationError>
      <PathName>/OrderRequest/Body/Product/AuthCodeInfo/AuthCode/</PathName>
      <StatusCode>-2027</StatusCode>
      <Description>Transfer Failed - Missing Auth Code</Description>
    </ValidationError>
  </Status>
  <ProductType>Registration</ProductType>
  <DomainName>EXAMPLE.ORG</DomainName>
  <Price>0.0</Price>
</LineItem>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.7.2 InboundTransferToParking

The following element from the *OrderRequest.dtd* defines the *InboundTransferToParking* action:

```
<!ELEMENT InboundTransferToParking ( ToCustomer, DomainName+, AdminContact?,
BillContact?, BulkWhoisOptOut?) >
```

The **InboundTransferToParking** Request should be issued to initiate the process of transferring a domain into the Network Solutions Registrar that will also be hosted by Network Solutions. Once the proper authorization steps are successful, this request will result in a domain that is hosted by Network Solutions.

As indicated in the *OrderRequest* DTD, *OrderRequests* may contain multiple *InboundTransferParking* elements. This is allowed in order to support "cross customer" transfer requests. As a result, each *InboundTransferToParking* element may specify a different value for *ToCustomer*, and each *InboundTransferToParking* element will result in a separate *Order* element returned in the resulting *OrderResponse* message. When performing "cross customer" transfer requests, the *AuthorizedAgent SystemUserType* element must be set to "partner". However, if it is set to "customer" then only the *InboundTransferToParking* elements whose *ToCustomer* element has the same value as the *UserID* element in *AuthorizedAgent* will be processed.

Duplicate domain names within a given request will be filtered out, resulting in only the first occurrence of the domain name actually being submitted for transfer. Table 39 contains a detailed explanation of each field in the request.

Table 39. Required and Optional Fields for the InboundTransferToParking

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 13 for details.
InboundTransferToParking			
ToCustomer	Required	NO	The customer ID to which the domain should be transferred.
DomainName	Required	YES	A maximum of 1000 domain names are allowed per request.

Field	Required	Multiple	Notes
AdminContact	Optional	NO	If AdminContact is not provided it defaults to the UserID of the customer.
BillContact	Optional	NO	Although BillContact may be modified, this is only a cosmetic change. The Partner will still receive the invoice and renewal notices.
DomainProtect	Optional	NO	Default is True.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.

Example – InboundTransferToParking Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>456</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID></CustomerID>
    <Product>
      <InboundTransferToParking>
        <ToCustomer>
          <UserID>456</UserID>
        </ToCustomer>
        <DomainName>example.net</DomainName>
        <AuthCodeInfo>
          <DomainName>example.net</DomainName>
          <AuthCode>password</AuthCode>
        </AuthCodeInfo>
      </InboundTransferToParking>
    </Product>
  </Body>
</OrderRequest>
```

Expected Response - InboundTransferToParking

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278646</ReferenceNumber>
    <Timestamp> 2006-08-24T13:38:59EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278646</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
```

```

</Status>
<OrderID>1000309166</OrderID>
<OrderAction>InboundTransfer</OrderAction>
<Customer>
  <UserID>456</UserID>
</Customer>
<Price>6.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.226564396</ProductID>
  <DomainName>example.net</DomainName>
  <Price>6.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Parking</ProductType>
  <ProductID>WN.H.226564397</ProductID>
  <DomainName>example.net</DomainName>
  <Price>0.0</Price>
</LinItem>
<OrderSubmissionDate> 2006-08-24T13:39:09EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example – InboundTransferToParking Request (with BZ and AuthCode)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123 </PartnerID>
      <PartnerPassword>password</PartnerPassword>
    </Authentication>
  </RequestHeader>

  <Body>
    <AuthorizedAgent SystemUserType="partner">
      <SystemUser>

```

```
<UserID>123</UserID>
  <Password>password</Password>
</SystemUser>
</AuthorizedAgent>
<CustomerID>789</CustomerID>
<Product>
  <InboundTransferToParking>
    <ToCustomer>
      <UserID>789</UserID>
    </ToCustomer>
    <DomainName>example.bz</DomainName>
    <AuthCodeInfo>
      <DomainName>example.bz</DomainName>
      <AuthCode>password</AuthCode>
    </AuthCodeInfo>
  </InboundTransferToParking>
</Product>
</Body>
</OrderRequest>
```

Expected Response - InboundTransferToParking

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
```

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91147446</ReferenceNumber>
    <Timestamp> 2006-08-24T13:41:26EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91147446</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000916893</OrderID>
      <OrderAction>InboundTransfer</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>26.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191337042</ProductID>
        <DomainName>EXAMPLE.BZ</DomainName>
        <Price>26.0</Price>
      </LineItem>
    </Order>
  </Body>
</OrderResponse>
```

```

<LinItemID>2</LinItemID>
<Status>
  <StatusCode>7004</StatusCode>
  <Description>Line item submitted successfully</Description>
</Status>
<ProductType>Advanced DNS</ProductType>
<ProductID>WN.H.191337043</ProductID>
<Name>EXAMPLE.BZ</Name>
<Price>0.0</Price>
</LinItem>
<OrderSubmissionDate> 2006-08-24T13:41:35EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.8 RetrieveAuthCode Request

The following request should be issued to trigger an email to the primary contact of a domain with the domain AuthCode. The following TLDs are supported for this request, (cc, tv, bz, tw (including all regional), cn (including all regional), com, net, biz, org, info, us, name(third level only), mobi). The maximum domains allowed per request is 100.

The following element from the OrderRequest.dtd defines the RetrieveAuthCode Request action:

```
<!ELEMENT RetrieveAuthCode (DomainName+)>
```

Table 40. Required and Optional Fields for RetrieveAuthCode Request

Field	Required	Multiple	Notes
RetrieveAuthCode	Required	NO	See Appendix E for details.
Domain	Required	YES	

Example – AuthoCode Request

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partnerpass</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>custpass</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>

```

```
<RetrieveAuthCode>
  <DomainName>example.CC</DomainName>
  <DomainName>example.TV</DomainName>
  <DomainName>example.TW</DomainName>
  <DomainName>example.COM</DomainName>
  <DomainName>example.biz</DomainName>
  <DomainName>example.org</DomainName>
  <DomainName>example.info</DomainName>
  <DomainName>example.us</DomainName>
  <DomainName>example.bz</DomainName>
  <DomainName>example.net</DomainName>
  <DomainName>example.cn</DomainName>
  <DomainName>john.johnson.name</DomainName>
</RetrieveAuthCode>
</Product>
</Body>
</OrderRequest>
```

Expected Response - RetrieveAuthCode

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
```

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105786938</ReferenceNumber>
    <Timestamp>2006-08-24T13:57:51EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105786938</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>217296623</OrderID>
      <OrderAction>AuthCodeRequest</OrderAction>
      <Customer>
        <UserID>40249165</UserID>
      </Customer>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>AuthCodeRequest</ProductType>
        <Name> example.cc</Name>
      </LineItem>
      <LineItem>
        <LineItemID>2</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>AuthCodeRequest</ProductType>
```

```
<Name> example.tv</Name>
</LineItem>
<LineItem>
  <LineItemID>3</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.tw</Name>
</LineItem>
<LineItem>
  <LineItemID>4</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.com</Name>
</LineItem>
<LineItem>
  <LineItemID>5</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.biz</Name>
</LineItem>
<LineItem>
  <LineItemID>6</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.org</Name>
</LineItem>
<LineItem>
  <LineItemID>7</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.info</Name>
</LineItem>
<LineItem>
  <LineItemID>8</LineItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.us</Name>
</LineItem>
```

```
<Lineltem>
  <LineltemID>9</LineltemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.bz</Name>
</Lineltem>
<Lineltem>
  <LineltemID>10</LineltemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name> example.net</Name>
</Lineltem>
<Lineltem>
  <LineltemID>11</LineltemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name>example.cn</Name>
</Lineltem>
<Lineltem>
  <LineltemID>12</LineltemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>AuthCodeRequest</ProductType>
  <Name>john.johnson.name</Name>
</Lineltem>
<OrderSubmissionDate>2006-08-24T13:58:06EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>12</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>12</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>
```

2.2.9 Secured Socket Layer Product Management (SSL)

2.2.9.1 ReissueSSL

The following element from the OrderRequest.dtd defines the ReissueSSL action:

```
<ELEMENT ReissueSSL (ProductID, DomainName?, CSRData?, Contacts, AccuracyAgreement?)>
```

Partner can use the ReissueSSL for the following 3 cases:

1. The ReissueSSL request allows partners to reissue an existing (issued) certificate or seal.

- To find out if the certificate or seal is issued, use the FindSSLProductsForPartner and search by SSLProductStatus of "Issued" for a given partner or use the FindDetailedSSLProduct for the specific product.
2. If the Certificate is in "validation" state and has not been issued, partners can change the CSR information and/or contacts by doing the following steps:
1. Contact Customer Service Department and ask them to put the product into the "Provide CSR" state.
 2. Send the ReissueSSL request with the revised CSR and contacts information.
3. If the Seal is in "validation" state and has not been issued, partners can change the domain name/URL and/or contacts by doing the following steps:
- Contact Customer Service Department and ask them to put the product into the "Provide Validation" status if changing contact information, or "Provide Domain" if changing both the domain name and contact information.
 - Send the ReissueSSL request with the revised domain name/URL and contacts information.

For Server Selection ID, please see Appendix F for the complete list of ID and corresponding server software name.

For detailed description of error codes, please see section B5 OrderRequest DTD.

For detailed description of validation error related to the SSL Product, please see section B10.3 Validation Codes.

Table 41. Required and Optional Fields for the RenewSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
ReissueSSL			
ProductID	Required	NO	Original product's ID. This ID is returned with the CreateSSL response or can be retrieved using the FindAllSSLProductsForPartner or FindSSLProductsForPartner queries.
DomainName	See Notes	NO	Required for case (3) above - If the product is SiteConfirmSeal and the partner would like to change the domain name. Not required for case (1) – If the product is SiteConfirmSeal and product has been issued.
CSRData	See Notes	NO	Required for case (2) above - If the product is a certificate (Basic, Pro or WildCard) and the partner would like to change the domain name, company name and other CSR related data. Not required for case (1) – If the product is a certificate (Basic, Pro or WildCard) and product has been issued.
CSR		NO	Certificate Signing Request w/ domain name and company name. CSR must be the same as the original product.
ServerSelectionID	Required	NO	See table XX for a list of server selections
Contacts	Required	NO	
ValidationContact	Required	NO	Contact information that will be used for verification
EntityType	Required	NO	Selection: Individual, OrganizationCompany, EducationalInstitution or

	Required	Multiple	Notes
			GovernmentOrganization
AddressLine1	Required	NO	
AddressLine2	Optional	NO	
AddressLine3	Optional	NO	
POBox	Optional	NO	
City	Required	NO	
State	Required	NO	
PostalCode	Required	NO	
CountryCode	Required	NO	
Department	Optional	NO	Entering data in this field will overwrite the department name (orgUnitName) in the CSR.
CompanyName	See notes	NO	Not required if product is certificate. Required if product is site confirm seal and request is for case (3).
ContactFirstName	Required	NO	
ContactLastName	Required	NO	
ContactPhone	Required	NO	
DUNS	Optional	NO	Company Registration Name, limit 20 characters
OrderContact	Required	NO	Partner contact information. See CreateSSL in section 2.2.1.6 for attribute descriptions.
CertificateRequester	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
CertificateApprover	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
ContractSigner	Optional		Required for EV services Please see CreateSSL in section 2.2.1.6 for attribute descriptions.
AccuracyAgreement	Optional	NO	Required for EV Certificates True or False

Example 1 : Reissue an existing issued Organizationally Validated Certificate(case 1)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1234 </PartnerID>
    <PartnerPassword> some password </PartnerPassword>
  </Authentication>
  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1234 </UserID>
      <Password> some password </Password>
    </SystemUser>
  </AuthorizedAgent>
</Body>
</OrderRequest>
```

```

</AuthorizedAgent>

  <CustomerID>40280168 </CustomerID>
  <Billing>
  </Billing>
  <Product>
  <ReissueSSL>
    <ProductID>WN.C.198860257</ProductID>
    <CSRData>
      <CSR>
MIIB/zCCAwwCAQAwgZQxCzAJBgNVBAYTAnVzMQswCQYDVQQIEwJ2YTEQMA4GA1UE
BxMHaGVybmlRbWVjEMMAoGA1UEChMDZW5nMRswGQYDVQQLEwJmZWdjZXJ0LTAxMDM0t
Mi5jb20xGzAZBgNVBAMTEmZlZ2NlcnQtMDEwMy0yLmNvbTEeMBwGCsQGS1b3DQEJ
ARYPdGVzdEBuZXRzb2wuY29tMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCm
0+XaV5SMbFMw39q5UtttmxBsDu/g5FpU/3oybjpIXkh909bQ+30gPTUDzInG4300
l1LHbe0BfejSk/t8a30Y9+UuH5JvaSFFRFVMS2apxymcrqJPkMu5wmrpriuvPcGA
kpfPY/DNg3H9vzxeN6Wbi3E8RGIotn0KmpFJG1QhNQIDAQABoCowEwYJKoZihvcN
AQkCMQYTBHRlc3QwEwYJKoZihvcNAQkHMQYTBHRlc3QwDQYJKoZihvcNAQEFBQAD
gYEAfAlunbpeJ/pnrOFk9u3A2AR0T6XgCgpyHJ0HLhTDGIR+z7+xZT2MNdPBaMLL
CLc6SgvS7DG11U/RjqyVzJb349iRV/T9jTqdyH5V2i9epIpQ82oXMO5/g4h/U+U0
JaD87dT72Q3cDi3oTbXXEVxtXVSF3Pthop6ruUh16zQDhfQ=
      </CSR>
    </CSRData>
    <Contacts>
      <ValidationContact>
        <EntityType Selection="EducationalInstitution"/>
        <AddressLine1>13200 Woodland Park Dr</AddressLine1>
        <AddressLine2>Ste 120</AddressLine2>
        <POBox>212</POBox>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
        <ContactFirstName>firstName</ContactFirstName>
        <ContactLastName>lastName</ContactLastName>
        <ContactPhone>7030000000</ContactPhone>
        <DUNS>2212</DUNS>
      </ValidationContact>
      <OrderContact>
        <FirstName>reissue-firstName</FirstName>
        <LastName>reissue-lastName</LastName>
        <Phone>reissue-phone</Phone>
        <Email>contact@example-name.com</Email>
      </OrderContact>
    </Contacts>
  </ReissueSSL>
</Product>
</Body>
</OrderRequest>

```

Example 1 expected response: Reissue an existing issued Organizationally Validated Certificate (case 1)

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

```

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790859</ReferenceNumber>
    <Timestamp>2007-01-22T10:51:52EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790859</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>217842531</OrderID>
      <OrderAction>ReissueSSL</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>SSL</ProductType>
        <ProductID>WN.C.198860257</ProductID>
        <DomainName>fegcert-0103-2.com</DomainName>
        <Name>SSL_BASIC</Name>
        <Price>0.0</Price>
      </LineItem>
      <OrderSubmissionDate>2007-01-22T10:52:00EST</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>
```

2.2.9.2 RevokeSSL

The following element from the *OrderRequest.dtd* defines the *RevokeSSL* action:

```
<ELEMENT RevokeSSL (ServiceInstID, (RevokeReason | OtherReason))>
```

Partner can use the *RevokeSSL* to revoke a particular instance of the Secured Socket Layer product, by specifying the *ServiceInstID*.

If the revoked instance is the only instance, then the product is also revoked.

Revoke only applies to Certificates, therefore, all seal revoke requests will result in an error. For detailed description of error codes, please see section B5 *OrderRequest* DTD.

Table 42. Required and Optional Fields for the RevokeSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
RevokeSSL			
ServiceInstID	Required	NO	The SSL instance for the certificate that will be revoked. To get the ServiceInstID: First, Use the high level queries FindAllSSLProductsForPartner or FindSSLProductsForPartner to get the Product ID then use the FindDetailedSSLProduct to get ServiceInstID.
RevokeReason	Optional	NO	Selection: (PrivateKeyCompromised AffiliationChanged Superseded CessationOfOperations)
Other Reason	Optional	NO	Other reasons not listed above.

Example 1: RevokeSSL request

```

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1234 </PartnerID>
    <PartnerPassword> some password </PartnerPassword>
  </Authentication>

  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>

</RequestHeader>

<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1234 </UserID>
      <Password> some password </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>40280168 </CustomerID>
  <Billing>
    <AutoRenew>>false</AutoRenew>
    <IsPricingRequest/>
  </Billing>
  <Product>
<RevokeSSL>

```

```
<ServiceInstID>9120659</ServiceInstID>
  <RevokeReason Selection="PrivateKeyCompromised"/>
</RevokeSSL>
</Product>
</Body>
</OrderRequest>
```

Example 1 expected response: RevokeSSL

```
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790863</ReferenceNumber>
    <Timestamp>2007-01-22T11:15:00EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790863</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>217842535</OrderID>
      <OrderAction>-1</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>SSL</ProductType>
        <ProductID>WN.C.198860258</ProductID>
        <DomainName>FEGCERT-0103-4.COM</DomainName>
        <Name>SSL_BASIC</Name>
        <Price>0.0</Price>
      </LineItem>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>
```

2.2.9.3 DownloadCertificate

The following element from the OrderRequest.dtd defines the DownloadCertificate action:

```
<ELEMENT DownloadCertificate (ServiceInstID, CertificateFormat)>
```

Partner can use the Download Certificate to download a specific instance of the certificate that has been issued. The certificate can then be installed on the customer's website. Please refer to the Help section at www.partners.networksolutions.com for additional installation instructions.

The response is defined in OrderResponse.dtd. The Certificate will be returned in the [CDATA element since it contains special characters (see Example 1 expected response).

For detailed description of error codes, please see section B5 OrderRequest DTD.

Table 43. Required and Optional Fields for the RevokeSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
DownloadCertificate			
ServiceInstID	Required	NO	The SSL instance that will be revoked. To get the ServiceInstID: <ol style="list-style-type: none"> 1. Use the high level queries FindAllSSLProductsForPartner or FindSSLProductsForPartner to get the Product ID. 2. use the FindDetailedSSLProduct to get ServiceInstID.
CertificateFormat	Required	NO	Selection (UTNAddTrustServerCA AddTrustExternalCA NetworkSolutionsCA BundledCert DomainSpecific EVRootCA IntermediateSSLCA) The EVRootCA and IntermediateSSLCA selections applies to the EV Service. The following formats will return a single certificate: <ul style="list-style-type: none"> • UTNAddTrustServerCA • AddTrustExternalCA • NetworkSolutionsCA • DomainSpecific The BundledCert format will return 2 certificates: UTNAddTrustServerCA and NetworkSolutionsCA. These certificates can be used to install on web servers such as Apache/Plesk .

Example 1: Downloading a certificate in AddTrustExternalCA format

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
```

```

<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1234 </PartnerID>
    <PartnerPassword> some password </PartnerPassword>
  </Authentication>
  <SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1234 </UserID>
      <Password> some password </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>40244930 </CustomerID>
  <Billing>
    <AutoRenew>false</AutoRenew>
  </Billing>
  <Product>
<DownloadCertificate>
  <ServiceInstID>9122866</ServiceInstID>
  <CertificateFormat Selection="AddTrustExternalCA"/>
</DownloadCertificate>
  </Product>
</Body>
</OrderRequest>

```

Example 1 expected response : Downloading a certificate in AddTrustExternalCA format

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790851</ReferenceNumber>
    <Timestamp>2007-01-19T13:59:34EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105790851</ReferenceNumber>
      <Status>
        <StatusCode>7012</StatusCode>
        <Description>Command completed successfully</Description>
      </Status>
      <OrderID>0</OrderID>
      <OrderAction>DownloadCertificate</OrderAction>
      <Customer>
        <UserID>40244930</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>

```

Certificate is returned between the <![CDATA[-----BEGIN CERTIFICATE--- -- and the -----END CERTIFICATE-----]]> tag.

```

<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7005</StatusCode>
    <Description>Order line item has completed</Description>
  </Status>
  <ProductType>SSL</ProductType>
  <ProductID>WN.C.198862846</ProductID>
  <Price>0.0</Price>
  <SSLDetail>
    <Certificate><![CDATA[-----BEGIN CERTIFICATE-----
MIENjCCAx6gAwIBAgIBATANBgkqhkiG9w0BAQUFADBvMQswCQYDVQQGEwJTRTEU
MBIGA1UEChMLQWRkVHJ1c3QgQUlXJjAkBgNVBAAsTHUFkZFRydXN0IEV4dGVybmFs
IFRUUCBOZXR3b3JrMSIwIAYDVQQDExiBZGRUcnVzdCBFeHRlcm5hbCBDQSB290
MB4XDTEwMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUzMDUz
FDASBgNVBAoTC0FkZFRydXN0IEFCMSYwJAYDVQQLEw1BZGRUcnVzdCBFeHRlcm5h
bCBUVFAGTmV0d29yazEiMCAGA1UEAxMZQWRkVHJ1c3QgRXh0ZXJuYWwgQ0EgUm9v
dDCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALf3GjPm8gAELTngTlvt
H7xsD821+iO2zt6bETOXpCImfZOvfUq8k+0DGuOPz+VtUFRWlymUWoCwSXrbLpX9
uMq/NzgtHj6RQa1wVsfwTz/oMp50ysiQVOnGXw94nZpAPA6sYapeFI+eh6FqUNzX
mk6vBbOmcZScbnNQYArHE504B4YCqOmoaSYyKtMsE8jqzpPhNjfp/haW+710LX
a0Tx63ubUfclpxCDezeWWkWaCUN/cALw3CknLa0Dhy2xSoRcRdKn23tNbE7qzN
E0S3ySvdQwAl+mG5aWpYlxG3pzOPVnVZ9c0p10a3CitltnCbxWyuHv77+ldU9U0
WicCAwEAAaOB3DCB2TAdBgNVHQ4EFgQUrb2Yejs0Jvf6xCZU7wO94CTLVBowCwYD
VR0PBAQDAgEGMA8GA1UdEwEB/wQFMAMBAf8wgZkGA1UdlwSBkTCBjoAUrb2Yejs0
Jvf6xCZU7wO94CTLVBqhc6RxBG8xCzAJBgNVBAYTAINFMRQwEgYDVQQKEw1BZGRU
cnVzdCBBBQJEmMCQGA1UECXMdQWRkVHJ1c3QgRXh0ZXJuYWwgVFRQIE5ldHdvcmsx
ljAgBgNVBAMTGUFkZFRydXN0IEV4dGVybmFsIENBIFJvb3SCAQEwDQYJKoZIhvcN
AQEFBQADggEBALCb4IUlwtYj4g+WBpKdQZic2YR5gdkeWxQHlzZl7DYd7usQWxH
YINRSPkyPef89iYTx4AWpb9a/IfPeHmJlZriTAcKhjW88t5RxNKWt9x+Tu5w/Rw5
6wwCURQtjr0W4MHfRnXnJK3s9EK0hZNwEGE6nQY1ShjTK3rMUUKhemPR5ruhXsvC
Nr4TDea9Y355e6cJDUcRat2PisP29owaQgVR1EX1n6dilWgVIEM8med8vSTYqZEX
c4g/VhsxOBI0cQ+azcgOno4uG+GMmIPLHzHxRezGBHNJdmAPx/i9F4BrLunMTA5a
mnkPIAou1Z5jJh5VkpTYghdae9C8x49OhgQ=
-----END CERTIFICATE-----]]></Certificate>
  </SSLDetail>
</LinItem>
  <OrderSubmissionDate>-1</OrderSubmissionDate>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
  <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.2.9.4 GetSealCode

The following element from the *OrderRequest.dtd* defines the *GetSealCode* action:

```
<ELEMENT GetSealCode (ServiceInstID, SealType, SealColor, SealShape, PublishingType)>
```

Partner can use the *GetSealCode* to download the seal code associated with the Basic, Pro or WildCard certificate, as well as the Site Confirm Seal.

The response is defined in *OrderResponse.dtd*. The java script response will be returned as CDATA element since it contains special characters.

For detailed description of error codes, please see section B5 OrderRequest DTD.

Table 44. Required and Optional Fields for the RevokeSSL Request

	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body			See Table 13 for details.
GetSealCode			
ServiceInstID	Required	NO	The SSL instance to get the seal code for. To get the ServiceInstID: <ol style="list-style-type: none"> 1. Use the high level queries FindAllSSLProductsForPartner or FindSSLProductsForPartner to get the Product ID. 2. use the FindDetailedSSLProduct to get ServiceInstID.
SealType	Required	NO	Selection (Basic ProWildCard SiteConfirmSeal EV) If the product is Basic, then selection must be Basic. If the product is Pro or Wildcard, selection can be Basic or ProWildCard. If the product is EV , then selection must be EV. If the product is SiteConfirmSeal, then selection must be SiteConfirmSeal. An error will be returned if the selection does not match the product associated with the ServInstId.
SealColor	Required	NO	Selection (Green Blue)
SealShape	Required	NO	Selection (Square Rectangular)
PublishingType	Required	NO	Selection (HTTP HTTPS)

Example 1: GetSealCode

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderRequest SYSTEM "OrderRequest.dtd">

<OrderRequest>
<RequestHeader>
  <VERSION_4_6/>
  <Authentication>
    <PartnerID> 1234 </PartnerID>
```

```
<PartnerPassword> some password </PartnerPassword>
</Authentication>
<SystemInfo>Verisign-PartnerPortal-[8.0]</SystemInfo>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="partner">
    <SystemUser>
      <UserID> 1234 </UserID>
      <Password> some password </Password>
    </SystemUser>
  </AuthorizedAgent>

  <CustomerID>40280168 </CustomerID>
  <Billing>
    <AutoRenew>>false</AutoRenew>
  </Billing>
  <Product>
<GetSealCode>
  <ServiceInstID>9125359</ServiceInstID>
  <SealType Selection="SiteConfirmSeal"/>
  <SealColor Selection="Green"/>
  <SealShape Selection="Square"/>
  <PublishingType Selection="HTTP"/>
</GetSealCode>
  </Product>
</Body>
</OrderRequest>
```

Example 1 expected response : GetSealCode

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14414801</ReferenceNumber>
    <Timestamp>2007-01-19T14:13:00EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14414801</ReferenceNumber>
      <Status>
        <StatusCode>7012</StatusCode>
        <Description>Command completed successfully</Description>
      </Status>
      <OrderID>0</OrderID>
      <OrderAction>GetSealCode</OrderAction>
      <Customer>
        <UserID>40280168</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
```

```

<LinItemID>1</LinItemID>
<Status>
  <StatusCode>7005</StatusCode>
  <Description>Order line item has completed</Description>
</Status>
<ProductType>SSL</ProductType>
<ProductID>WN.S.198865674</ProductID>
<Name>SSL_SEAL</Name>
<Price>0.0</Price>
<SSLDetail>
  <SealHeader><![CDATA[<script language="JavaScript"
src="http://seal.networksolutions.com/siteseal/javascript/siteseal.js"
type="text/javascript"></script>
]]></SealHeader>
  <SealBody><![CDATA[<!--
SiteSeal Html Builder Code:
Shows the logo at URL http://seal.networksolutions.com/images/basicsgreen.gif
Logo type is ("NETSB")
/-->
<script language="JavaScript" type="text/javascript">
SiteSeal("http://seal.networksolutions.com/images/basicsgreen.gif", "NETSB", "none");</script>
]]></SealBody>
</SSLDetail>
</LinItem>
<OrderSubmissionDate>-1</OrderSubmissionDate>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Javascript for seal code is returned in the <![CDATA[]]> tag.

2.3 Lookup Request (LookupRequest.dtd)

The actions contained in the LookupRequest.dtd consist of:

- LookupHost, and
- LookupDomain.

The following element from the LookupRequest.dtd defines the Body:

```
<!ELEMENT Body ( AuthorizedAgent, CustomerID?, ( LookupDomain | LookupHost ) ) >
```

If the CustomerID is provided, the system will give an error message if any of the following happens:

- If 'Super-User' authentication is used and the customer does not belong to the partner;
- If 'Customer-Level' authentication is used and this CustomerId is not the same as the one in the AuthorizedAgent;
- The domain or host does not belong to the customer.

Table contains a detailed explanation of the Body fields.

Note: Only one action may be submitted in a single request.

Table 45. Body for Lookup Request

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Optional	NO	

Field	Required	Multiple	Notes
LookupDomain	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
LookupHost	Conditional	NO	

2.3.1 LookupHost

The following element from the LookupRequest.dtd defines the LookupHost action:

```
<!ELEMENT LookupHost (HostName)>
```

The **LookupHost** Request should be issued to gather all information about an existing Host.

Deprecation Note: *The HostID and HostIP elements found within the LookupHost element were deprecated as of version 3.3. Please do not use these elements. They will be removed in a future version.*

Table 46. Required and Optional Fields for the LookupHost Request. contains a detailed explanation of each field in the request.

Table 46. Required and Optional Fields for the LookupHost Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 45. Body for Lookup Request for details.
LookupHost			
HostName	Conditional	NO	Host can be identified by HostName

Example – LookupHost Request

```
<?xml version="1.0" encoding="UTF-8"?>
<LookupRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <LookupHost>
      <HostName>dns.example.com</HostName>
    </LookupHost>
  </Body>
</LookupRequest>
```

Expected Response – LookupHost

```
<?xml version="1.0"?>
<!DOCTYPE LookupResponse SYSTEM "LookupResponse.dtd">
<LookupResponse>
  <ResponseHeader>
    <VERSION_4_6></VERSION_4_6>
    <ReferenceNumber>25668605</ReferenceNumber>
    <Timestamp>2006-08-24T14:17:59EDT</Timestamp>
  </ResponseHeader>
  <Body>
```

```

    <HostData>
      <HostID></HostID>
      <HostName>dns.example.com</HostName>
      <HostIP1>1.2.3.4</HostIP1>
    </HostData>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
  </Body>
</LookupResponse>

```

2.3.2 LookupDomain

The following element from the LookupRequest.dtd defines the LookupDomain action:

```
<!ELEMENT LookupDomain ( ProductID+ | DomainName+ ) >
```

The **LookupDomain** Request should be issued to gather all information about one or more existing domains. LookupDomain will retrieve data for a Registration or Parked domain. Table contains a detailed explanation of each field in the request.

Table 47. Required and Optional Fields for the LookupDomain Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 45. Body for Lookup Request for details.
LookupDomain			
ProductID	Conditional	YES	The domain can be identified by either ProductID or DomainName.
DomainName	Conditional	YES	

Example – LookupDomain Request

```

<?xml version="1.0" encoding="UTF-8"?>
<LookupRequest>
  <RequestHeader>
    <VERSION_4_6</VERSION_4_6
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <LookupDomain>
      <DomainName>example.com</DomainName>
    </LookupDomain>
  </Body>
</LookupRequest>

```

Expected Response – LookupDomain

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LookupResponse SYSTEM "LookupResponse.dtd">

<LookupResponse>

```

```

<ResponseHeader>
  <VERSION_5_0></VERSION_5_0>
  <ReferenceNumber>78364088</ReferenceNumber>
  <Timestamp>2006-08-24T14:18:41EDT</Timestamp>
</ResponseHeader>
<Body>
  <Domain>
    <ProductID>WN.D.186987916</ProductID>
    <ProductType>Registration</ProductType>
    <DomainNameProperties encoded="false">AFT-E3YZRB102.COM</DomainNameProperties>
    <ProductStatus>ACTIVE</ProductStatus>
    <DomainProtect Selection="True"/>
    <LegalLock Selection="False"/>
    <CustomerID>38721539</CustomerID>
    <AdminContact>
      <UserID>38721539</UserID>
    </AdminContact>
    <TechContact>
      <UserID>38721539</UserID>
    </TechContact>
    <BillContact>
      <UserID>1234</UserID>
    </BillContact>
    <BulkWhoisOptOut>N</BulkWhoisOptOut>
    <AutoRenew>N</AutoRenew>
    <PurchaseDate>08.21.2006</PurchaseDate>
    <ExpirationDate>08.21.2007</ExpirationDate>
    <NicHandle>ISKUEURPBD</NicHandle>
    <HostData>
      <HostID></HostID>
      <HostName>NS1.WORLDDNIC.COM</HostName>
      <HostIP1>216.168.225.131</HostIP1>
      <HostIP2>216.168.228.3</HostIP2>
      <HostIP3>216.168.231.11</HostIP3>
    </HostData>
    <HostData>
      <HostID></HostID>
      <HostName>NS2.WORLDDNIC.COM</HostName>
      <HostIP1>216.168.225.240</HostIP1>
      <HostIP2>216.168.231.21</HostIP2>
    </HostData>
    <PrivateRegistrationActive>N</PrivateRegistrationActive>
  </Domain>
  <Status>
    <StatusCode>11702</StatusCode>
    <Description>Product lookup successful</Description>
  </Status>
</Body>
</LookupResponse>

```

Example – LookupDomain Request (Private RegistrationActive)

```

<?xml version='1.0' encoding='UTF-8'?>
<LookupRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>

```

```
<PartnerID>1234567</PartnerID>
<PartnerPassword>partner</PartnerPassword>
</Authentication>
<SystemInfo>VERISIGN-PARTNERPORTAL</SystemInfo>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType='customer'>
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <LookupDomain>
    <DomainName>example.com.COM</DomainName>
  </LookupDomain>
</Body>
</LookupRequest>
```

Expected Response – LookupDomain Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE LookupResponse SYSTEM "LookupResponse.dtd">
```

```
<LookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>78364042</ReferenceNumber>
    <Timestamp>2006-08-24T14:20:13EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Domain>
      <ProductID>WN.D.186986783</ProductID>
      <ProductType>Registration</ProductType>
      <DomainNameProperties encoded="false">example.com</DomainNameProperties>
      <ProductStatus>ACTIVE</ProductStatus>
      <DomainProtect Selection="True"/>
      <LegalLock Selection="False"/>
      <CustomerID>789</CustomerID>
      <AdminContact>
        <UserID>345</UserID>
      </AdminContact>
      <TechContact>
        <UserID>387</UserID>
      </TechContact>
      <BillContact>
        <UserID>147</UserID>
      </BillContact>
      <BulkWhoisOptOut>N</BulkWhoisOptOut>
      <AutoRenew>N</AutoRenew>
      <PurchaseDate>01.14.2005</PurchaseDate>
      <ExpirationDate>01.14.2007</ExpirationDate>
      <NicHandle>TPCNNXTBAD</NicHandle>
      <HostData>
        <HostID></HostID>
        <HostName>NS1.WORLDDNIC.COM</HostName>
        <HostIP1>1.68.25.1</HostIP1>
```

```

    <HostIP2>1.68.25.2</HostIP2>
    <HostIP3>1.68.25.3</HostIP3>
  </HostData>
  <HostData>
    <HostID></HostID>
    <HostName>NS2.WORLDDNIC.COM</HostName>
    <HostIP1>1.68.25.4</HostIP1>
    <HostIP2>1.68.25.5</HostIP2>
  </HostData>
  <PrivateRegistrationActive>Y</PrivateRegistrationActive>
</Domain>
<Domain>
  <ProductID>186986784</ProductID>
  <ProductType>PrivateRegistration</ProductType>
  <DomainNameProperties encoded="false">example.com</DomainNameProperties>
  <ProductStatus>ACTIVE</ProductStatus>
  <DomainProtect Selection="True"/>
  <LegalLock Selection="False"/>
  <CustomerID>789</CustomerID>
  <AutoRenew>N</AutoRenew>
  <PurchaseDate>08.21.2006</PurchaseDate>
  <ExpirationDate>08.21.2007</ExpirationDate>
</Domain>
<Status>
  <StatusCode>11702</StatusCode>
  <Description>Product lookup successful</Description>
</Status>
</Body>
</LookupResponse>

```

Example – LookupDomain Request (With WebForwarding)

```

<?xml version='1.0' encoding='UTF-8'?>
<LookupRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234567</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType='customer'>
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <LookupDomain>
      <DomainName>example.com</DomainName>
    </LookupDomain>
  </Body>
</LookupRequest>

```

Expected Response – LookupDomain Request

```

<?xml version="1.0" encoding="UTF-8"?>

```

```
<!DOCTYPE LookupResponse SYSTEM "LookupResponse.dtd">
```

```
<LookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105786953</ReferenceNumber>
    <Timestamp>2006-08-24T14:29:31EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Domain>
      <ProductID>WN.D.198719895</ProductID>
      <ProductType>Registration</ProductType>
      <DomainNameProperties encoded="false">EXAMPLE.COM</DomainNameProperties>
      <ProductStatus>ACTIVE</ProductStatus>
      <DomainProtect Selection="True"/>
      <LegalLock Selection="False"/>
      <CustomerID>40247872</CustomerID>
      <AdminContact>
        <UserID>40247872</UserID>
      </AdminContact>
      <TechContact>
        <UserID>40247872</UserID>
      </TechContact>
      <BillContact>
        <UserID>40217631</UserID>
      </BillContact>
      <BulkWhoisOptOut>N</BulkWhoisOptOut>
      <AutoRenew>N</AutoRenew>
      <PurchaseDate>08.24.2006</PurchaseDate>
      <ExpirationDate>08.24.2007</ExpirationDate>
      <NicHandle>GYLWKCOQHD</NicHandle>
      <HostData>
        <HostID></HostID>
        <HostName>NS29.WORLDDNIC.COM</HostName>
        <HostIP1>21.16.22.15</HostIP1>
      </HostData>
      <HostData>
        <HostID></HostID>
        <HostName>NS30.WORLDDNIC.COM</HostName>
        <HostIP1>21.16.22.16</HostIP1>
      </HostData>
      <PrivateRegistrationActive>N</PrivateRegistrationActive>
    </Domain>
    <Domain>
      <ProductID>WN.H.198719902</ProductID>
      <ProductType>Parking</ProductType>
      <DomainNameProperties encoded="false">example.com</DomainNameProperties>
      <ProductStatus>ACTIVE</ProductStatus>
      <DomainProtect Selection="True"/>
      <LegalLock Selection="False"/>
      <CustomerID>40247872</CustomerID>
      <AutoRenew>N</AutoRenew>
      <PurchaseDate>08.24.2006</PurchaseDate>
      <ExpirationDate>12.31.2007</ExpirationDate>
    </Domain>
  </Body>
</LookupResponse>
```

```

<ProductID>WN.F.198719903</ProductID>
<ProductType>Web Forwarding</ProductType>
<DomainNameProperties encoded="false">example.com</DomainNameProperties>
<ProductStatus>ACTIVE</ProductStatus>
<DomainProtect Selection="True"/>
<LegalLock Selection="False"/>
<CustomerID>40247872</CustomerID>
<AutoRenew>N</AutoRenew>
<PurchaseDate>08.24.2006</PurchaseDate>
<ExpirationDate>08.24.2007</ExpirationDate>
<WebForwardingProduct>
  <DomainName>example.com</DomainName>
  <DestURL>http://www.microsoft.com</DestURL>
  <Masking>True</Masking>
</WebForwardingProduct>
</Domain>
<Status>
  <StatusCode>11702</StatusCode>
  <Description>Product lookup successful</Description>
</Status>
</Body>
</LookupResponse>

```

2.4 Partner Manager Request (*PartnerManagerRequest.dtd*)

The actions contained in the *PartnerManagerRequest.dtd* consist of:

- FindAllCustomersForPartner
- FindAllDomainsForCustomer
- FindAllDomainsForPartner
- FindAllDomainsWithPrivateRegistration
- FindAllDomainsWithWebForwarding
- ConsolidateAllCustomerProducts
- ConsolidateSelectedCustomerProducts
- GetPartnerData

The following element from the *PartnerManagerRequest.dtd* defines the Body:

```

<!ELEMENT Body ( FindAllCustomersForPartner | FindAllDomainsForCustomer |
FindAllDomainsForPartner | FindDomainsWithPrivateRegistration |
ConsolidateAllCustomerProducts | ConsolidateSelectedCustomerProducts ) >

```

Table 48. Body for PartnerManager Request contains a detailed explanation of the Body fields.

Note: *Only one action may be submitted in a single request.*

Table 48. Body for PartnerManager Request

Field	Required	Multiple	Notes
FindAllCustomersForPartner	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
FindAllDomainsForCustomer	Conditional	NO	
FindAllDomainsForPartner	Conditional	NO	
FindDomainsWithPrivateRegistration	Conditional	NO	
FindDomainsWithWebForwarding	Conditional	NO	
ConsolidateAllCustomerProducts	Conditional	NO	

Field	Required	Multiple	Notes
ConsolidateSelectedCustomerProducts	Conditional	NO	
GetPartnerData	Conditional	NO	

Table 49. Common Required and Optional Fields for PartnerManager Requests outlines those fields in common with a PartnerManager Request. The sections that follow detail each request individually.

Table 49. Common Required and Optional Fields for PartnerManager Requests

Field	Required	Multiple	Notes
DateRange	Optional	NO	See Appendix E for details.
CreatedDateRange	Optional	NO	Date the customer was created.
DateRange	Required	NO	
PurchaseDateRange	Optional	NO	Date the product was purchased.
DateRange	Required	NO	
ExpirationDateRange	Optional	NO	Date the product expires.
DateRange	Required	NO	

2.4.1 FindAllCustomersForPartner

The following element from the PartnerManagerRequest.dtd defines the FindAllCustomersForPartner action:

```
<!ELEMENT FindAllCustomersForPartner ( CreatedDateRange? ) >
```

The **FindAllCustomersForPartner** Request should be issued to retrieve a list of all customers created by this Partner. Table contains a detailed explanation of each field in the request.

Note: If there are a large number of Partner customers, this request may take a long time to complete. A maximum of 100,000 records will be returned by the system at one time.

Table 50. Required and Optional Fields for the FindAllCustomersForPartner Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindAllCustomersForPartner			
CreatedDateRange	Optional	NO	See Table 49. Common Required and Optional Fields for PartnerManager Requests for details.

Example - FindAllCustomersForPartner Request

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <FindAllCustomersForPartner>
      <CreatedDateRange>
        <DateRange>
          <BeginDate>
            <Date>2006-08-01T00:00:00EDT</Date>
          </BeginDate>
          <EndDate>
            <Date>2006-08-13T00:00:00EDT</Date>
          </EndDate>
        </DateRange>
      </CreatedDateRange>
    </FindAllCustomersForPartner>
  </Body>
</PartnerManagerRequest>
```

```

    </FindAllCustomersForPartner>
  </Body>
</PartnerManagerRequest>

```

Expected Response – FindAllCustomersForPartner

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">

```

```

<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145551</ReferenceNumber>
    <Timestamp>2006-08-24T14:42:24EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5702</StatusCode>
      <Description>User lookup successful</Description>
    </Status>
    <Customer>
      <Individual>
        <UserID>12345678</UserID>
        <NicHandle>12345678P</NicHandle>
        <LoginName>USER1</LoginName>
        <FirstName>User1</FirstName>
        <LastName>User1</LastName>
      </Individual>
    </Customer>
    <Customer>
      <Individual>
        <UserID>23456789</UserID>
        <NicHandle>23456789P</NicHandle>
        <LoginName>USER2</LoginName>
        <FirstName>User2</FirstName>
        <LastName>User2</LastName>
      </Individual>
    </Customer>
    <DateRange>
      <BeginDate>
        <Date>2006-08-01T00:00:00EDT</Date>
      </BeginDate>
      <EndDate>
        <Date>2006-08-13T00:00:00EDT</Date>
      </EndDate>
    </DateRange>
    <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
  </Body>
</PartnerManagerResponse>

```

2.4.2 FindAllDomainsForCustomer

The following element from the PartnerManagerRequest.dtd defines the FindAllDomainsForCustomer action:

```

<!ELEMENT FindAllDomainsForCustomer ( CustomerID, ( PurchaseDateRange? |
ExpirationDateRange? ) )>

```

The **FindAllDomainsForCustomer** Request should be issued to retrieve a list of all domains owned by a given customer. Table 51. Required and Optional Fields for the FindAllDomainsForCustomer Request contains a detailed explanation of each field in the request.

Note: *If there are a large number of domains for this customer, this request may take a long time to complete. A maximum of 100,000 records will be returned by the system at one time.*

Table 51. Required and Optional Fields for the FindAllDomainsForCustomer Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindAllDomainsforCustomer			
CustomerID	Required	NO	
PurchaseDateRange	Optional	NO	See Table 49. Common Required and Optional
ExpirationDateRange	Optional	NO	Fields for PartnerManager Requests for details.

Example - FindAllDomainsForCustomer Request

```
<?xml version="1.0" encoding="UTF-8"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <FindAllDomainsForCustomer>
      <CustomerID>789</CustomerID>
      <PurchaseDateRange>
        <DateRange>
          <BeginDate>
            <Date>2006-08-01T00:00:00EDT</Date>
          </BeginDate>
          <EndDate>
            <Date>2006-08-13T00:00:00EDT</Date>
          </EndDate>
        </DateRange>
      </PurchaseDateRange>
    </FindAllDomainsForCustomer>
  </Body>
</PartnerManagerRequest>
```

Expected Response - FindAllDomainsForCustomer

```
<?xml version="1.0"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">
<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145624</ReferenceNumber>
    <Timestamp>2006-08-24T14:42:38EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
```

```

<Domain>
  <ProductID>WN.D.191241501</ProductID>
  <DomainName encoded="false">EXAMPLE.COM</DomainName>
  <ProductType>Registration</ProductType>
  <DomainProtect Selection="True"/>
  <LegalLock Selection="False"/>
  <AutoRenew>N</AutoRenew>
  <PurchaseDate>
    <Date>2006-08-07T09:49:59EDT</Date>
  </PurchaseDate>
  <ExpirationDate>
    <Date>2007-08-07T09:49:59EDT</Date>
  </ExpirationDate>
  <BulkWhoisOptOut>N</BulkWhoisOptOut>
  <CustomerID>39425258</CustomerID>
</Domain>
<Domain>
  <ProductID>WN.D.191241509</ProductID>
  <DomainName encoded="false">EXAMPLE1.COM</DomainName>
  <ProductType>Registration</ProductType>
  <DomainProtect Selection="True"/>
  <LegalLock Selection="False"/>
  <AutoRenew>N</AutoRenew>
  <PurchaseDate>
    <Date>2006-08-07T09:55:34EDT</Date>
  </PurchaseDate>
  <ExpirationDate>
    <Date>2007-08-07T09:55:34EDT</Date>
  </ExpirationDate>
  <BulkWhoisOptOut>N</BulkWhoisOptOut>
  <CustomerID>39425258</CustomerID>
</Domain>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Body>
</PartnerManagerResponse>

```

2.4.3 FindAllDomainsForPartner

The following element from the PartnerManagerRequest.dtd defines the FindAllDomainsForPartner action:

```
<!ELEMENT FindAllDomainsForPartner ( PurchaseDateRange? | ExpirationDateRange? ) >
```

The **FindAllDomainsForPartner** Request should be issued to retrieve a list of domains managed by all customers of the Partner. Table contains a detailed explanation of each field in the request.

Note: *If there are a large number of domains for the Partner, this request may take a long time to complete. A maximum of 100,000 records will be returned by the system at one time.*

Table 52. Required and Optional Fields for the FindAllDomainsForPartner Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindAllDomainsForPartner			
PurchaseDateRange	Optional	NO	See Table 49. Common Required and Optional
ExpirationDateRange	Optional	NO	Fields for PartnerManager Requests for details.

Example - FindAllDomainsForPartner Request

```
<?xml version="1.0" encoding="UTF-8"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>50240628</PartnerID>
      <PartnerPassword>devTest</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <FindAllDomainsForPartner>
      <PurchaseDateRange>
        <DateRange>
          <BeginDate>
            <Date>2005-01-01T00:00:00EST</Date>
          </BeginDate>
          <EndDate>
            <Date>2005-06-13T00:00:00EST</Date>
          </EndDate>
        </DateRange>
      </PurchaseDateRange>
    </FindAllDomainsForPartner>
  </Body>
</PartnerManagerRequest>
```

Expected Response - FindAllDomainsForPartner

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">

<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14403859</ReferenceNumber>
    <Timestamp>2006-08-24T14:29:31EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
    <Domain>
      <ProductID>WN.D.198692263</ProductID>
      <DomainName encoded="false">example1.net</DomainName>
      <ProductType>Registration</ProductType>
      <DomainProtect Selection="True"/>
      <LegalLock Selection="False"/>
      <AutoRenew>N</AutoRenew>
      <PurchaseDate>
        <Date>2006-08-03T07:44:05EDT</Date>
      </PurchaseDate>
      <ExpirationDate>
        <Date>2007-08-03T07:44:05EDT</Date>
      </ExpirationDate>
      <BulkWhoisOptOut>N</BulkWhoisOptOut>
      <CustomerID>40217632</CustomerID>
```

```

</Domain>
<Domain>
  <ProductID>WN.D.198692248</ProductID>
  <DomainName encoded="false">example1.cn</DomainName>
  <ProductType>Registration</ProductType>
  <DomainProtect Selection="True"/>
  <LegalLock Selection="False"/>
  <AutoRenew>N</AutoRenew>
  <PurchaseDate>
    <Date>2006-08-03T07:42:44EDT</Date>
  </PurchaseDate>
  <ExpirationDate>
    <Date>2007-08-03T07:42:44EDT</Date>
  </ExpirationDate>
  <BulkWhoisOptOut>N</BulkWhoisOptOut>
  <CustomerID>40217632</CustomerID>
</Domain>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Body>
</PartnerManagerResponse>

```

2.4.4 FindDomainsWithPrivateRegistration

The following element from the PartnerManagerRequest.dtd defines the FindDomainsWithPrivateRegistration action:

```
<!ELEMENT FindDomainsWithPrivateRegistration (PrivateRegistrationActive, StartRecord,MaximumRecords) >
```

The FindDomainsWithPrivateRegistration Request should be issued to retrieve a list of all domains with Private Registration active.

Table 53. Required and Optional Fields for the FindDomainsWithPrivateRegistration Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindDomainsWithPrivateRegistration			
PrivateRegistrationActive	Required	NO	Empty Tag
StartRecord	Required	NO	Index: default is 0 (First Record)
MaximumRecords	Required	NO	Number of records to retrieve

Example - FindDomainsWithPrivateRegistration Request

```

<?xml version='1.0' encoding='UTF-8'?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>12345678</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <FindDomainsWithPrivateRegistration>
      <PrivateRegistrationActive Selection='True'/>

```

```

    <StartRecord>1</StartRecord>
    <MaximumRecords>5</MaximumRecords>
  </FindDomainsWithPrivateRegistration>
</Body>
</PartnerManagerRequest>
Expected Response – FindDomainsWithPrivateRegistration
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">

<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145698</ReferenceNumber>
    <Timestamp>2006-08-24T14:51:31EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
    <DomainName>example1.com</DomainName>
    <DomainName>example2.com</DomainName>
    <DomainName>example3.com</DomainName>
    <DomainName>example4.com</DomainName>
    <DomainName>example5.com</DomainName>
    <NumberOfRecordsReturned>5</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>5</NumberOfRecordsAvailable>
  </Body>
</PartnerManagerResponse>

```

2.4.5 FindDomainsWithWebForwarding

The following element from the PartnerManagerRequest.dtd defines the FindDomainsWithWebForwarding action:

```
<!ELEMENT FindDomainsWithWebForwarding (WithWebFwd, StartRecord,MaximumRecords) >
```

The FindDomainsWithWebForwarding Request should be issued to retrieve a list of all domains with WebForwarding on.

Table 54. Required and Optional Fields for the FindDomainsWithWebForwarding Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindDomainsWithWebForwarding			
WithWebFwd	Required	NO	Empty Tag
StartRecord	Required	NO	Index: default is 0 (First Record)
MaximumRecords	Required	NO	Number of records to retrieve

Example - FindDomainsWithWebForwarding Request

```

<?xml version='1.0' encoding='UTF-8'?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>

```

```

    <Authentication>
      <PartnerID>12345678</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <FindDomainsWithWebForwarding>
      <WithWebFwd Selection='True'/>
      <StartRecord>1</StartRecord>
      <MaximumRecords>5</MaximumRecords>
    </FindDomainsWithWebForwarding>
  </Body>
</PartnerManagerRequest>

```

Expected Response – FindDomainsWithWebForwarding

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">

<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145696</ReferenceNumber>
    <Timestamp>2006-08-24T14:52:16EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
    <DomainName>example1.com</DomainName>
    <DomainName>example2.com</DomainName>
    <DomainName>example3.com</DomainName>
    <DomainName>example4.com</DomainName>
    <DomainName>example5.com</DomainName>
    <NumberOfRecordsReturned>5</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>5</NumberOfRecordsAvailable>
  </Body>
</PartnerManagerResponse>

```

2.4.6 ConsolidateAllCustomerProducts

The following element from the PartnerManagerRequest.dtd defines the ConsolidateAllCustomerProducts action:

```
<!ELEMENT ConsolidateAllCustomerProducts ( FromCustomer, ToCustomer ) >
```

The **ConsolidateAllCustomerProducts** Request should be issued to move all products owned by one customer to another customer. Table 55. Required and Optional Fields for the ConsolidateAllCustomerProducts Request contains a detailed explanation of each field in the request.

Note: This is a synchronous request. If there are a large number of products for the FromCustomer, this request may take a long time to complete.

For this request to be successful the following criteria must be met:

The FromCustomer and ToCustomer must be of the same type, either Individual or Business.

The FromCustomer and ToCustomer must have the same name. If they are Businesses, then the CompanyName field of both customers must match. If they are both Individual customers, then their FirstName and LastName fields must match. The MiddleName is not used during this comparison. These comparisons are not case sensitive.

None of the domains owned by the FromCustomer may be in the following TLDs: .us, .co.uk, .org.uk, .me.uk, .de, .com.mx, .com.br, .co.nz., .at, .be. In other words, product consolidation is not currently supported for customers owning domains under these TLDs.

The FromCustomer must own no more than 1000 products.

Table 55. Required and Optional Fields for the ConsolidateAllCustomerProducts Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
ConsolidateAllCustomerProducts			
FromCustomer	Required	NO	
ToCustomer	Required	NO	

Example - ConsolidateAllCustomerProducts Request

```
<?xml version="1.0" encoding="UTF-8"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <ConsolidateAllCustomerProducts>
      <FromCustomer>
        <UserID>456</UserID>
      </FromCustomer>
      <ToCustomer>
        <UserID>321</UserID>
      </ToCustomer>
    </ConsolidateAllCustomerProducts>
  </Body>
</PartnerManagerRequest>
```

Expected Response – ConsolidateAllCustomerProducts

```
<?xml version="1.0"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">
<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145687</ReferenceNumber>
    <Timestamp> 2006-08-24T14:44:06EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <CustomerProductConsolidationResult>
      <FromCustomer>
        <UserID>456</UserID>
      </FromCustomer>
      <ToCustomer>
        <UserID>321</UserID>
      </ToCustomer>
    </CustomerProductConsolidationResult>
  </Body>
</PartnerManagerResponse>
```

2.4.7 ConsolidateSelectedCustomerProducts

The following element from the PartnerManagerRequest.dtd defines the ConsolidateSelectedCustomerProducts action:

```
<!ELEMENT ConsolidateSelectedCustomerProducts ( FromCustomer, ToCustomer,
(DomainName+ | ProductID+) ) >
```

The **ConsolidateSelectedCustomerProducts** Request should be issued to move a select set of products owned by one customer to another customer. Table 56. Required and Optional Fields for the ConvertSelectedCustomerProducts Request contains a detailed explanation of each field in the request.

Note: This is a synchronous request. If there are a large number of products, this request may take a long time to complete.

For this request to be successful the following criteria must be met:

The FromCustomer and ToCustomer must be of the same type, either Individual or Business.

The FromCustomer and ToCustomer must have the same name. If they are Businesses then the CompanyName field of both customers must match. If they are both Individual customers, then their FirstName and LastName fields must match. The MiddleName is not used during this comparison. These comparisons are not case sensitive.

None of the domains provided in the list of products to be consolidated may be in the following TLDs: .us, .co.uk, .org.uk, .me.uk, .de, .com.mx, .com.br, .co.nz, .at, .be. In other words, product consolidation is not currently supported for customers owning domains under these TLDs.

The list of products to consolidate must contain no more than 1000 products.

Table 56. Required and Optional Fields for the ConvertSelectedCustomerProducts Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
ConsolidateSelectedCustomerProducts			
FromCustomer	Required	YES	
ToCustomer	Required	YES	
DomainName	Required	YES	
ProductID	Required	YES	

Example - ConsolidateSelectedCustomerProducts Request

```
<?xml version="1.0" encoding="UTF-8"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <ConsolidateSelectedCustomerProducts>
      <FromCustomer>
        <UserID>456</UserID>
      </FromCustomer>
      <ToCustomer>
        <UserID>321</UserID>
      </ToCustomer>
      <DomainName>example.com</DomainName>
      <DomainName>example.net</DomainName>
    </ConsolidateSelectedCustomerProducts>
  </Body>
</PartnerManagerRequest>
```

Expected Response – ConsolidateSelectedCustomerProducts

```

<?xml version="1.0"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">
<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145693</ReferenceNumber>
    <Timestamp> 2006-08-24T14:45:27EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <CustomerProductConsolidationResult>
      <FromCustomer>
        <UserID>456</UserID>
      </FromCustomer>
      <ToCustomer>
        <UserID>321</UserID>
      </ToCustomer>
      <ProductConsolidationResult>
        <DomainName encoded="false">example.com</DomainName>
        <Status>
          <StatusCode>7014</StatusCode>
          <Description>Product and all its children products consolidated.</Description>
        </Status>
      </ProductConsolidationResult>
      <ProductConsolidationResult>
        <DomainName encoded="false">example.net</DomainName>
        <Status>
          <StatusCode>7014</StatusCode>
          <Description>Product and all its children products consolidated.</Description>
        </Status>
      </ProductConsolidationResult>
    </CustomerProductConsolidationResult>
  </Body>
</PartnerManagerResponse>

```

2.4.8 GetPartnerData

The following element from the PartnerManagerRequest.dtd defines the GetPartnerData action:

```
<!ELEMENT GetPartnerData EMPTY >
```

The **GetPartnerData** Request should be issued to retrieve that status of the partner account (active, suspended, etc), and the partner's prepaid or credit balance. No AccountBalance element will be returned if the partner does not maintain a credit or prepaid balance.

Table 57. Required and Optional Fields for the GetPartnerData Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
GetPartnerData			The element is empty.

Example - GetPartnerData Request

```
<?xml version="1.0" encoding="UTF-8"?>
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <GetPartnerData></GetPartnerData>
  </Body>
</PartnerManagerRequest>
```

Expected Response – GetPartnerData

```
<?xml version="1.0"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">
<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91145695</ReferenceNumber>
    <Timestamp>2006-08-24T14:57:41EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <PartnerData AccountStatus="ACTIVE">
      <AccountBalance>10000.0</AccountBalance>
    </PartnerData>
  </Body>
</PartnerManagerResponse>
```

2.4.9 FindAllSSLProductsForPartner

The following element from the PartnerManagerRequest.dtd defines the FindAllSSLProductsForPartner action:

```
<!ELEMENT FindAllSSLProductsForPartner EMPTY >
```

The FindAllSSLProductsForPartner Request should be issued to retrieve all the Security Socket Layer products for a partner ID. The request returns a high level of information, including the product ID, customer ID etc. Please see the PartnerManagerResponse.dtd for element definitions.

Table 58. Required and Optional Fields for the FindAllSSLProductsForPartner Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindAllSSLProductsForPartner			The element is empty.

Example - FindAllSSLProductsForPartner Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerRequest SYSTEM "PartnerManagerRequest">
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234</PartnerID>
      <PartnerPassword> some password </PartnerPassword>
    </Authentication>
    <Comments>pcdata</Comments>
  </RequestHeader>
  <Body>
    <FindAllSSLProductsForPartner></FindAllSSLProductsForPartner>
  </Body>
</PartnerManagerRequest>
```

Expected Response - FindAllSSLProductsForPartner Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">
<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105790437</ReferenceNumber>
    <Timestamp>2007-01-09T10:01:41EST</Timestamp>
    <Comments>pcdata</Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <Product>
      <ProductID>WN.C.198857016</ProductID>
      <ProductType>SSL_CERTS</ProductType>
      <Name>SiteSafe(TM) SSL Certificate - Basic</Name>
      <DomainName>JIGARPARIKH.COM</DomainName>
      <PurchaseDate>
        <Date>2006-12-27T11:26:15EST</Date>
      </PurchaseDate>
      <ExpirationDate>
        <Date>2006-12-27T14:42:04EST</Date>
      </ExpirationDate>
      <CustomerID>15641500</CustomerID>
      <Status>Domain Configured</Status>
    </Product>
    <Product>
      <ProductID>WN.C.198857268</ProductID>
      <ProductType>SSL_CERTS</ProductType>
      <Name>SiteSafe(TM) SSL Certificate - Basic</Name>
      <DomainName>JIGARPARIKH.COM</DomainName>
```

```

<PurchaseDate>
  <Date>2006-12-28T13:51:58EST</Date>
</PurchaseDate>
<ExpirationDate>
  <Date>2007-01-02T15:50:20EST</Date>
</ExpirationDate>
<CustomerID>15641500</CustomerID>
<Status>Domain Configured</Status>
</Product>
</Product>
<DateRange>
  <BeginDate>
    <LongDate>0</LongDate>
  </BeginDate>
  <EndDate>
    <LongDate>1168354908206</LongDate>
  </EndDate>
</DateRange>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>0</NumberOfRecordsAvailable>
</Body>

```

2.4.10 FindSSLProductsForPartner

The following element from the PartnerManagerRequest.dtd defines the FindSSLProductsForPartner action:

```
<ELEMENT FindSSLProductsForPartner (CustomerID | PurchaseDateRange | ExpirationDateRange | WildCardSearch | DomainName | SSLProductStatus)>
```

The FindSSLProductsForPartner Request allows the partner to narrow the search based on selected criterias. The response will be the same as the FindAllSSLProductsForPartner above. See PartnerManagerResponse.dtd for more details.

Table 59. Required and Optional Fields for the FindSSLProductsForPartner Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindSSLProductsForPartner			Selection : CustomerID PurchaseDateRange ExpirationDateRange WildCardSearch DomainName SSLProductStatus.
CustomerID	Optional	NO	Search based on customer id.
PurchaseDateRange			Search based on purchase date range.
ExpirationDateRange			Search based on expiration date range.
WildCardSearch			Search based on wild card.
DomainName			Search based on domain name.
SSLProductStatus			Search based on product status. Selection (ProvideDomain DomainConfigured TermStarted Issued BeingReissued Expired BeingRenewed Renewed).

Example 1: FindAllSSLProductsForPartner based on Wildcard search “contains”

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerRequest SYSTEM "PartnerManagerRequest">
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234</PartnerID>
      <PartnerPassword> some password </PartnerPassword>
    </Authentication>
    <Comments>pCDATA</Comments>
  </RequestHeader>
  <Body>
    <FindSSLProductsForPartner>
      <WildcardSearch>
        <Contains>fegcert</Contains>
      </WildcardSearch>
    </FindSSLProductsForPartner>
  </Body>
</PartnerManagerRequest>
```

Example 2: FindAllSSLProductsForPartner based on Wildcard search “endsWith”

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerRequest SYSTEM "PartnerManagerRequest">
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234</PartnerID>
      <PartnerPassword> some password </PartnerPassword>
    </Authentication>
    <Comments>pCDATA</Comments>
  </RequestHeader>
  <Body>
    <FindSSLProductsForPartner>
      <WildcardSearch>
        <EndsWith>parikh</EndsWith>
      </WildcardSearch>
    </FindSSLProductsForPartner>
  </Body>
</PartnerManagerRequest>
```

Example 3: FindAllSSLProductsForPartner based on product status

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerRequest SYSTEM "PartnerManagerRequest">
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID>1234</PartnerID>
      <PartnerPassword> some password </PartnerPassword>
    </Authentication>
    <Comments>pCDATA</Comments>
  </RequestHeader>
```

```

<Body>
  <FindSSLProductsForPartner>
    <SSLProductStatus Selection="Issued"/>
  </FindSSLProductsForPartner>
</Body>
</PartnerManagerRequest>

```

2.4.11 FindDetailedSSLProduct

The following element from the PartnerManagerRequest.dtd defines the FindDetailedSSLProduct action:

```
<!ELEMENT FindDetailedSSLProduct (ProductID)>
```

The FindDetailedSSLProduct Request allows the partner to get detailed information given a product ID.

The product ID is returned when the product is first purchased (CreateSSL) or when the product is renewed (RenewSSL).

The information returned by the FindDetailedSSLProduct like ServiceInstID returned in the SSLInstance element can be used for requests such as RevokeSSL, DownloadCertificate and GetSealCode. Other information includes "statuses" or "actions" that can be taken against the SSL instance, such as "IsRevocable" or "IsReissueable".

Table 60. Required and Optional Fields for the FindDetailedSSLProduct Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 48. Body for PartnerManager Request for details.
FindDetailedSSLProduct			ProductID

Example : FindDetailedSSLProduct Request

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerRequest SYSTEM "PartnerManagerRequest">
<PartnerManagerRequest>
  <RequestHeader>
    <VERSION_4_6/>
    <Authentication>
      <PartnerID> 1234 </PartnerID>
      <PartnerPassword> some password </PartnerPassword>
    </Authentication>
    <Comments>pcdata</Comments>
  </RequestHeader>
  <Body>
    <FindDetailedSSLProduct>
      <ProductID> WN.C.198865414 </ProductID>
    </FindDetailedSSLProduct>
  </Body>
</PartnerManagerRequest>

```

Expected response: FindDetailedSSLProduct

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PartnerManagerResponse SYSTEM "PartnerManagerResponse.dtd">

```

```

<PartnerManagerResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14414803</ReferenceNumber>
    <Timestamp>2007-01-19T14:59:56EST</Timestamp>
    <Comments>pcdata</Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>7012</StatusCode>
      <Description>Command completed successfully</Description>
    </Status>
    <Product>
      <ProductID>WN.C.198865414</ProductID>
      <ProductType>SSL_CERTS</ProductType>
      <Name>SiteSafe(TM) SSL Certificate - Wildcard</Name>
      <PurchasePeriod>2</PurchasePeriod>
      <ExpirationDate>
        <Date>2009-04-09T12:42:32EDT</Date>
      </ExpirationDate>
      <Status>The domain name or IP address has been assigned to the Certificate or Seal.</Status>
      <IsRenewable>N</IsRenewable>
      <IsReissueable>N</IsReissueable>
      <HasEverBeenIssued>N</HasEverBeenIssued>
      <SSLInstance>
        <ServiceInstID>9125217</ServiceInstID>

```

Actions that can be taken
against this SSL instance

```

<CSR><![CDATA[MIICFTCCAX4CAQAwgaoxGzAJBgNVBAYTAnVzMQswCQYDVQQIEwJuYzESMB
AGA1UE
BxMJaGlnaHBvaW50MRwwGgYDVQQKEExNmZWdjZXJ0LXdpcGRjYXJkIGNvMQ4wDAYD
VQQLEwVzYWxlczEmMCQGA1UEAxQdKi5mZWdjZXJ0LXdpcGRjYXJkLTAxMDktMS5j
b20xJDAiBkgqhkiG9w0BCQEFWFhlibGVuQGZIZ2NlcnR0ZXN0LmNvbTCBnzANBgkq
hkiG9w0BAQEFAAOBjQAwgYkCgYEAAnlpxmFjXNL/v9pAzL5HpcjJnvsoOfX0xtbrp
mIL8IMRnmC4IAmkVZhxzTOM2CY72tpctFIKc0iT5b4F8nbPC9/vEo8P5FScF7fN
0VJv6WJTRKIjxjCdtSWLcJUzujwLyhjNhFvh+9I5clvs8e/WFQ1IHMaPYGzVx7
CSov8s0CAwEAAaAqMBMGCSqGSIb3DQEJAJEGEwR0ZXN0MBMGCSqGSIb3DQEBBzEG
EwR0ZXN0MA0GCSqGSIb3DQEBBQUAA4GBAANlu9y3sMmuxDAWeN5xOfis2BZS8wic
8pOQoBp3tmOpb1iYdzeGTyX47SG71Hm5pffnXkSGKMQwsEGmBrsSxgbLYzk+mtwx
ueo44WeLKPL3yagWO83yIMrFY0nqihZDJBIVVxcaCRYeud8vGkaKgOSzlp/CRNNK
r7NwHgabMTdS]]></CSR>
  <ServerSelectionID>2</ServerSelectionID>
  <DomainName>*.CERT-WILDCARD-0109-1.COM</DomainName>
  <IsRevocable>N</IsRevocable>
  <IsDownloadable>N</IsDownloadable>
  <ValidationContact>
    <EntityType>Organization</EntityType>
    <AddressLine1>1234 Main St</AddressLine1>
    <POBox>515</POBox>
    <City>Herndon</City>
    <State>VA</State>
    <PostalCode>20171</PostalCode>
    <CountryCode>US</CountryCode>
    <Department>QA</Department>
    <CompanyName>Example</CompanyName>
    <ContactFirstName>contactsfirstname</ContactFirstName>
    <ContactLastName>contactslastname</ContactLastName>

```

```

    <ContactPhone>7030000000</ContactPhone>
    <DUNS/>
  </ValidationContact>
  <OrderContact>
    <FirstName>firstname</FirstName>
    <LastName>lastname</LastName>
    <Phone>7039998888</Phone>
    <Email>basic@netsol.com</Email>
  </OrderContact>
</SSLInstance>
</Product>
<DateRange>
  <BeginDate>
    <LongDate>0</LongDate>
  </BeginDate>
  <EndDate>
    <LongDate>1169236797071</LongDate>
  </EndDate>
</DateRange>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>0</NumberOfRecordsAvailable>
</Body>
</PartnerManagerResponse>

```

2.5 Available Request (AvailableRequest.dtd)

The actions contained in the LookupRequest.dtd consist of:

- VerifyDomainName,
- VerifyDomains,
- VerifyDomainsAdvanced,
- VerifyDomainsAll,
- GenerateDomains,
- GenerateDomainsAdvanced,
- GenerateDomainsAll,
- IsDomainCandidateForInboundTransfer,
- IsDomainCandidateForInternalTransfer

Note: *The Partner must send all available requests to port 8010 (see Section 1.2.1 for details).*

The following element from the AvailableRequest.dtd defines the Body:

```

<!ELEMENT Body ( VerifyDomainName+ | VerifyDomains | GenerateDomains |
VerifyDomainsAdvanced | GenerateDomainsAdvanced | VerifyDomainsAll |
GenerateDomainsAll | IsDomainCandidateForInboundTransfer+,
IsDomainCandidateForInternalTransfer+ ) >

```

Table contains a detailed explanation of the Body fields.

Table 61. Body for AvailableRequest

Field	Required	Multiple	Notes
VerifyDomainName	Conditional	YES	One of the "Conditional" elements in this table must be submitted.
VerifyDomains	Conditional	NO	
VerifyDomainsAdvanced	Conditional	NO	
VerifyDomainsAll	Conditional	NO	

Field	Required	Multiple	Notes
GenerateDomains	Conditional	NO	
GenerateDomainsAdvanced	Conditional	NO	
GenerateDomainsAll	Conditional	NO	
IsDomainCandidateForInboundTransfer	Conditional	YES	
IsDomainCandidateForInternalTransfer	Conditional	YES	

2.5.1 VerifyDomainName

The following element from the AvailableRequest.dtd defines the VerifyDomainName action:

```
<!ELEMENT VerifyDomainName (#PCDATA) >
```

The **VerifyDomainName** Request should be issued to verify that a domain is available for purchase. Multiple VerifyDomainName requests can be issued sequentially with this command. Table contains a detailed explanation of each field in the request.

Note: *The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).*

Table 62. Required and Optional Fields for the VerifyDomainName Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
VerifyDomainName	Required	YES	

Example - VerifyDomainName Request

```
<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <VerifyDomainName>example.com</VerifyDomainName>
    <VerifyDomainName>example.org</VerifyDomainName>
    <VerifyDomainName>example.net</VerifyDomainName>
  </Body>
</AvailableRequest>
```

Expected Response – VerifyDomainName

```
<?xml version="1.0"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2006-08-24T14:59:56EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName> example.com </DomainName>
      <RespondingService ServiceName="Registry"/>
    </Available>
    <NotAvailable>
      <DomainName> example.org</DomainName>
```

```

    <RespondingService ServiceName="Registry"/>
    <DomainName> example.net </DomainName>
    <RespondingService ServiceName="Registry"/>
  </NotAvailable>
</Body>
</AvailableResponse>

```

2.5.2 VerifyDomains

The following element from the AvailableRequest.dtd defines the VerifyDomains action:

```
<!ELEMENT VerifyDomains (#PCDATA) >
```

The **VerifyDomains** Request should be issued to check the availability .com, .net, .org, .info, .biz, .us.com, .mobi, .tv TLDs that start with the domain in question. For example, <VerifyDomains> word </VerifyDomains> will check word.com, word.net, word.info, word.biz, word.us.com, word.mobi, and word.tv

Table 63. *Required and Optional Fields for the VerifyDomains Request* contains a detailed explanation of each field in the request.

Note: *The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).*

Table 63. Required and Optional Fields for the VerifyDomains Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
VerifyDomains	Required	NO	

Example - VerifyDomains Request

```

<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    < VERSION_5_0></ VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <VerifyDomains>example</VerifyDomains>
  </Body>
</AvailableRequest>

```

Expected Response – VerifyDomains

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">

<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-07T16:05:58EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>

```

```

    <DomainName>example.us.com</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.tv</DomainName>
    <RespondingService ServiceName="Registry"/>
  </Available>
  <NotAvailable>
    <DomainName>example.net</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.org</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.biz</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.info</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.mobi</DomainName>
    <RespondingService ServiceName="Registry"/>
    <DomainName>example.com</DomainName>
    <RespondingService ServiceName="Registry"/>
  </NotAvailable>
</Body>
</AvailableResponse>

```

2.5.3 VerifyDomainsAdvanced

The following element from the AvailableRequest.dtd defines the VerifyDomainsAdvanced action:

```
<!ELEMENT VerifyDomainsAdvanced ( Domain, TLD+ ) >
```

The **VerifyDomainsAdvanced** Request should be issued to check the availability .com, .net, .org., info, .biz , .us.com , .mobi, and .tv TLDs and one or more of the specified TLDs that start with the domain in question. Table contains a detailed explanation of each field in the request.

Note: The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).

Table 64. Required and Optional Fields for the VerifyDomainsAdvanced Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
VerifyDomainsAdvanced	Required	NO	
Domain	Required	NO	
TLD	Required	YES	

Example - VerifyDomainsAdvanced Request

```

<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    < VERSION_5_0></ VERSION_5_0>
  </RequestHeader>
  <Authentication>
    <PartnerID>123</PartnerID>
    <PartnerPassword>partner</PartnerPassword>
  </Authentication>
  <Comments></Comments>
</AvailableRequest>

```

```
</RequestHeader>
<Body>
  <VerifyDomainsAdvanced>
    <Domain>example</Domain>
    <TLD>biz</TLD>
    <TLD>ws</TLD>
  </VerifyDomainsAdvanced>
</Body>
</AvailableRequest>
```

Expected Response – VerifyDomainsAdvanced

```
<?xml version="1.0"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-07T16:55:27EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName>example.tv</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.us.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </Available>
    <NotAvailable>
      <DomainName>example.biz</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.info</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.mobi</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.net</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.org</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.ws</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </NotAvailable>
  </Body>
</AvailableResponse>
```

2.5.4 VerifyDomainsAll

The following element from the AvailableRequest.dtd defines the VerifyDomainsAll action:

```
<!ELEMENT VerifyDomainsAll (#PCDATA) >
```

The **VerifyDomainsAll** Request should be issued to check the availability of all the following TLDs that start with the domain in question.

.com, .net, .org, .info, .biz, .us.com, .mobi, .tv, .am, .at, .be, .br.com, .bz, .cn, .com.cn, net.cn, .org.cn, .cc, .de, .de.com, .eu, .eu.com, .fm, .gb.com, .gb.net, .gs, .hu.com, .jpn.com, .kr.com, .la, .ms, .no.com, .com.mx, .co.nz, .net.nz, .org.nz, .qc.com, .ru.com, .sa.com, .se.com, .se.net, .tc, .tw, .com.tw, .org.tw, .co.uk, .org.uk, .me.uk, .uk.com, .uk.net, .us, .uy.com, .vg, .ws, .za.com, .es, .com.es, .org.es, .nom.es, .in, .co.in, .net.in, .org.in, .firm.in, .gen.in, .ind.in, .ah.cn, .bj.cn, .cq.cn, .fj.cn, .gd.cn, .gs.cn, .gx.cn, .ha.cn, .hb.cn, .he.cn, .hi.cn, .hk.cn, .hl.cn, .hn.cn, .jl.cn, .js.cn, .jx.cn, .ln.cn, .mo.cn, .nm.cn, .nx.cn, .qh.cn, .sc.cn, .sd.cn, .sh.cn, .sn.cn, .sx.cn, .tj.cn, .xj.cn, .xz.cn, .yn.cn, .zj.cn, .ar.com, .cx, .gd

Table 65. Required and Optional Fields for the VerifyDomainsAll Request contains a detailed explanation of each field in the request.

Note: The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).

Table 65. Required and Optional Fields for the VerifyDomainsAll Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
VerifyDomainsAll	Required	NO	

Example - VerifyDomainsAll Request

```
<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <VerifyDomainsAll>example</VerifyDomainsAll>
  </Body>
</AvailableRequest>
```

Expected Response – VerifyDomainsAll

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-08T11:38:36EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName>example.us.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.tv</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.am</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.br.com</DomainName>
```

```
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.cc</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.de.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.eu.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.fm</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gb.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gb.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gs</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hu.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.jpn.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.kr.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.la</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ms</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.no.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net.nz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.nz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.qc.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ru.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sa.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.se.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.se.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.tc</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.me.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uk.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uk.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uy.com</DomainName>
```

```
<RespondingService ServiceName="Registry"/>
<DomainName>example.vg</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.za.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.nom.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.firm.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gen.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ind.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ah.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.bj.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.cq.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.fj.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gd.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gs.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gx.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ha.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hb.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.he.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hi.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hk.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hl.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.hn.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.jl.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.js.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.jx.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ln.cn</DomainName>
```

```
<RespondingService ServiceName="Registry"/>
<DomainName>example.mo.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.nm.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.nx.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.qh.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sc.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sd.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sh.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sn.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.sx.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.tj.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.xj.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.xz.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.yn.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.zj.cn</DomainName>
<RespondingService ServiceName="Registry"/>
</Available>
<NotAvailable>
<DomainName>example.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.biz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.info</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.mobi</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.at</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.be</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.bz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.de</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.mx</DomainName>
```

```

<RespondingService ServiceName="Registry"/>
<DomainName>example.co.nz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.co.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.us</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ws</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.co.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com</DomainName>
<RespondingService ServiceName="Registry"/>
</NotAvailable>
<Invalid>
  <DomainName>example.eu</DomainName>
</Invalid>
</Body>
</AvailableResponse>

```

2.5.5 GenerateDomains

The following element from the AvailableRequest.dtd defines the GenerateDomains action:

```
<!ELEMENT GenerateDomains (#PCDATA) >
```

The **GenerateDomains** Request should be issued to generate a list of variations on the domain using com TLDs that contain the domain in question. Availability will be performed for the following TLDs: com net org biz info us.com mobi tv.

Table contains a detailed explanation of each field in the request.

Note: *The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).*

Table 66. Required and Optional Fields for the GenerateDomains Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
GenerateDomains	Required	NO	

Example - GenerateDomains Request

```

<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>

```

```
<Body>
  <GenerateDomains>example</GenerateDomains>
</Body>
</AvailableRequest>
```

Expected Response – GenerateDomains

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
```

```
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-17T11:38:49EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName>example.us.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.tv</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>freeexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleessays.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>pagesexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleformat.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplereport.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplebroker.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>urlexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>fastexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </Available>
    <NotAvailable>
      <DomainName>example.net</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.org</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.biz</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.info</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.mobi</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplepaper.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exemple.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </NotAvailable>
  </Body>
</AvailableResponse>
```

```
</NotAvailable>
</Body>
</AvailableResponse>
```

2.5.6 GenerateDomainsAdvanced

The following element from the AvailableRequest.dtd defines the GenerateDomainsAdvanced action:
 <ELEMENT GenerateDomainsAdvanced (Domain, TLD+, GetMisspelledListOnly?)>

The **GenerateDomainsAdvanced** should be used to get alternative domain names for the following TLDs: com.

If GetMisspelledListOnly is True, return the misspelled list only

Availability will also be returned for the requested TLDs as well as the following TLDs: com net org biz info us.com mobi tv

In addition, if the TLDs fall into the group below, additional availability checks will be performed (see table below).

Requested TLD	Additional availability will be returned for these TLDs
cn	cn com.cn org.cn net.cn
tw	tw com.tw org.tw
de	de de.com
eu	eu eu.com hu.com no.com ru.com se.com se.net
uk	uk uk.com uk.net gb.com gb.net qc.com za.com
es	es com.es org.es nom.es
in	in co.in net.in org.in firm.in gen.in ind.in

Note: The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).

Table 67. Required and Optional Fields for the GenerateDomainsAdvanced Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
GenerateDomainsAdvanced	Required	NO	
Domain	Required	NO	
TLD	Required	YES	
GetMisspelledListOnly	Optional	NO	If True, return alternative misspelled domain names

Example - GenerateDomainsAdvanced Request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableRequest SYSTEM "AvailableRequest.dtd">
<AvailableRequest>
<RequestHeader>
  <VERSION_5_0/>
<Authentication>
  <PartnerID> 1475448 </PartnerID>
  <PartnerPassword> intTest </PartnerPassword>
</Authentication>
```

```
</RequestHeader>
```

```
<Body>
```

```
  <GenerateDomainsAdvanced >
```

```
    <Domain> example </Domain>
```

```
    <TLD> AM</TLD>
```

```
    <GetMisspelledListOnly Selection = "True"/>
```

```
  </GenerateDomainsAdvanced >
```

```
</Body>
```

```
</AvailableRequest>
```

Expected Response – GenerateDomainsAdvanced with Misspelled option set to TRUE

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
```

```
<AvailableResponse>
```

```
  <ResponseHeader>
```

```
    <VERSION_5_0></VERSION_5_0>
```

```
    <ReferenceNumber>002</ReferenceNumber>
```

```
    <Timestamp>2008-01-17T11:16:24EST</Timestamp>
```

```
  </ResponseHeader>
```

```
  <Body>
```

```
    <Available>
```

```
      <DomainName>example.am</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.tv</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.us.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>exomple.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>exmp1e.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>examp1e.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>examp1.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>examp1.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>examp1e.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>examp1e.com</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
    </Available>
```

```
    <NotAvailable>
```

```
      <DomainName>example.biz</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.info</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.mobi</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.net</DomainName>
```

```
      <RespondingService ServiceName="Registry"/>
```

```
      <DomainName>example.org</DomainName>
```

```
<RespondingService ServiceName="Registry"/>
<DomainName>example.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exemple.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampl3.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>examplee.com</DomainName>
<RespondingService ServiceName="Registry"/>
</NotAvailable>
</Body>
</AvailableResponse>
```

Expected Response – GenerateDomainsAdvanced with Misspelled option set to FALSE

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">

<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-17T11:28:27EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName>example.am</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.tv</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.us.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>businessexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplebusiness.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplevideo.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>graphicexample.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleter.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleterm.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleinc.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>examplefamily.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>exampleletters.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </Available>
    <NotAvailable>
      <DomainName>example.biz</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.info</DomainName>
      <RespondingService ServiceName="Registry"/>
    </NotAvailable>
  </Body>
</AvailableResponse>
```

```

<DomainName>example.mobi</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exemple.com</DomainName>
<RespondingService ServiceName="Registry"/>
</NotAvailable>
</Body>
</AvailableResponse>

```

2.5.7 GenerateDomainsAll

The following element from the AvailableRequest.dtd defines the GenerateDomainsAll action:

```
<!ELEMENT GenerateDomainsAll (#PCDATA) >
```

The **GenerateDomainsAll** Request should be issued to generate a list of variations on the domain.

Alternative domains will be returned for the following TLDs: com

Availability will be returned for the following TLDs: am at be br.com bz cn com.cn net.cn org.cn cc de de.com eu eu.com fm gb.com gb.net gs hu.com jpn.com kr.com la ms no.com com.mx co.nz net.nz org.nz qc.com ru.com sa.com se.com se.net tc tw com.tw org.tw co.uk org.uk me.uk uk.com uk.net us uy.com vg ws za.com es com.es org.es nom.es in co.in net.in org.in firm.in gen.in ind.in

The Table below contains a detailed explanation of each field in the request.

Note: *The Partner must send all name availability requests to port 8010 (see Section 1.2.1 for details).*

Table 68. Required and Optional Fields for the GenerateDomainsAll Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
GenerateDomainsAll	Required	NO	

Example - GenerateDomainsAll Request

```

<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <GenerateDomainsAll>example</GenerateDomainsAll>
  </Body>
</AvailableRequest>

```

Expected Response – GenerateDomainsAll

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">

```

```
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2008-01-17T11:52:00EST</Timestamp>
  </ResponseHeader>
  <Body>
    <Available>
      <DomainName>example.us.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.tv</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.am</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.br.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.org.cn</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.cc</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.de.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.eu.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.fm</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.gb.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.gb.net</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.gs</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.hu.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.jpn.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.kr.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.la</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.ms</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.no.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.net.nz</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.org.nz</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.qc.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.ru.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.sa.com</DomainName>
      <RespondingService ServiceName="Registry"/>
      <DomainName>example.se.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </Available>
  </Body>
</AvailableResponse>
```

```
<DomainName>example.se.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.tc</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.me.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uk.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uk.net</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.uy.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.vg</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.za.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.nom.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.firm.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.gen.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ind.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampleletter.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampleessay.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>sayexample.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampleshining.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>shiningexample.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>affiliateexample.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampleterm.com</DomainName>
<RespondingService ServiceName="Registry"/>
</Available>
<NotAvailable>
  <DomainName>example.net</DomainName>
  <RespondingService ServiceName="Registry"/>
  <DomainName>example.org</DomainName>
  <RespondingService ServiceName="Registry"/>
```

```
<DomainName>example.biz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.info</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.mobi</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.at</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.be</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.bz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.net.cn</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.de</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com.mx</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.co.nz</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.tw</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.co.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.org.uk</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.us</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.ws</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.es</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.co.in</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>example.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exemple.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>exampledomain.com</DomainName>
<RespondingService ServiceName="Registry"/>
<DomainName>testexample.com</DomainName>
<RespondingService ServiceName="Registry"/>
</NotAvailable>
<Invalid>
  <DomainName>example.eu</DomainName>
</Invalid>
</Body>
</AvailableResponse>
```

2.5.8 IsDomainCandidateForInboundTransfer

The following element from the AvailableRequest.dtd defines the IsDomainCandidateForInboundTransfer action:

```
<!ELEMENT IsDomainCandidateForInboundTransfer (#PCDATA) >
```

The **IsDomainCandidateForInboundTransfer** Request should be issued to check the “transferability” of a domain name before including it in an InboundTransfer Order Request. A domain name is **not** a candidate for inbound transfer to the Network Solutions registrar if (1) it is already managed by the Network Solutions registrar, or (2) is not already registered, or (3) is not a well formed domain name, or (4) is not a domain name in the com, net, or org TLD. For example, `<IsDomainCandidateForInboundTransfer>word.com</IsDomainCandidateForInboundTransfer>` will verify that word.com is already registered and that it is not already managed by the Network Solutions registrar. If this is the case, the response will return `MaybeTransferable` for this domain; otherwise it will return `NotTransferable`. Table contains a detailed explanation of each field in the request.

Note: *The Partner must send all such requests to port 8010 (see Section 1.2.1 for details).*

Table 69. Required and Optional Fields for the IsDomainCandidateForInboundTransfer Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
IsDomainCandidateForInboundTransfer	Required	YES	

Example - IsDomainCandidateForInboundTransfer Request

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
```

```
<IsDomainCandidateForInboundTransfer>example.com</IsDomainCandidateForInboundTransfer>
</Body>
```

```
</AvailableRequest>
```

Expected Response – IsDomainCandidateForInboundTransfer

```
<?xml version="1.0"?>
```

```
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
```

```
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2006-08-24T15:12:31EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <MaybeTransferable>
      <DomainName>example.com</DomainName>
      <TransferDetail TransferType="Inbound">
    </MaybeTransferable>
  </Body>
```

</AvailableResponse>

2.5.9 IsDomainCandidateForInternalTransfer

The following element from the AvailableRequest.dtd defines the IsDomainCandidateForInternalTransfer action:

```
<!ELEMENT IsDomainCandidateForInternalTransfer (#PCDATA) >
```

The IsDomainCandidateForInternalTransfer Request should be issued to check the “transferability” of a domain name before including it in a TransferInternal Order Request. A domain name is not a candidate for internal transfer if (1) it is not a domain managed by the Network Solutions registrar, or (2) is not registered, or (3) is not a well formed domain name, or (4) the domain is in a Hold, Lock, or Special status, or (5) the domain is unpaid, or (6) the domain is already the subject of another pending transfer request, or (7) the domain has not been owned by the current owner for at least 60 days.

For example, <IsDomainCandidateForInboundTransfer> word.com </IsDomainCandidateForInboundTransfer> will verify that word.com adheres to all these conditions. If this is the case, the response will return MaybeTransferrable for this domain; otherwise it will return NotTransferrable. Table contains a detailed explanation of each field in the request.

Note: *The Partner must send all such requests to port 8010 (see Section 1.2.1 for details).*

Table 70. Required and Optional Fields for the IsDomainCandidateForInboundTransfer Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 61. Body for AvailableRequest for details.
IsDomainCandidateForInternalTransfer	Required	YES	

Example - IsDomainCandidateForInternalTransfer Request

```
<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <IsDomainCandidateForInternalTransfer>example.com
  </IsDomainCandidateForInternalTransfer>
  </Body>
</AvailableRequest>
```

Expected Response – IsDomainCandidateForInternalTransfer

```
<?xml version="1.0"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2006-08-24T15:13:14EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <MaybeTransferable>
      <DomainName>example.com</DomainName>
      <TransferDetail TransferType="Internal">
        <InternalTransferDetail>
```

```

    <InternalTransferCategory
InternalTransferCategorySelection="WholesaleToWholesaleInPartner"/>
    <ProductType>Registration</ProductType>
    </InternalTransferDetail>
  </TransferDetail>
</MayBettransferable>
</Body>
</AvailableResponse>

```

2.6 Order Status Request (OrderStatusRequest.dtd)

The action contained in the OrderStatusRequest.dtd consists of the following:

- Order

The following element describes the OrderStatusRequest.dtd Body:

```
<!ELEMENT Body ( AuthorizedAgent, CustomerID?, Order ) >
```

Table contains a detailed explanation of the Body fields.

Deprecation Note: *The use of CustomerID is no longer necessary and will be removed in a future version of this DTD. Please avoid using this item.*

Table 71. Body for OrderStatus Request

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Optional	NO	The use of CustomerID is no longer necessary and will be removed in a future version of this DTD. Please avoid using this item.
Order	Required	NO	

2.6.1 Order

The following element from the OrderStatusRequest.dtd defines the Order Request action:

```
<!ELEMENT Order ( ReferenceNumber | OrderID | Customer | Product | Domain |
OrderStatusInfo | OrderSubmissionDate ) >
```

The Order Request should be issued to check on the status of any asynchronously fulfilled orders. As the order is processed Network Solutions will update the status of each domain within that request. Partners can then use the Order Request to check on that status. The status should return either success, pending, failed or partial. Table contains a detailed explanation of each field in the request.

Table 72. Required and Optional Fields for the Order Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 71. Body for OrderStatus Request for details.
Order			
ReferenceNumber	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
OrderID	Conditional	NO	
Customer	Conditional	NO	
UserID	Required	NO	
OrderSubmissionDate	Optional	NO	
Product	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ProductID	Required	NO	
OrderSubmissionDate	Optional	NO	
Domain	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
DomainName	Required	NO	

Field	Required	Multiple	Notes
OrderSubmissionDate	Optional	NO	
OrderStatusInfo	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
OrderStatus	Required	NO	
OrderStatusType	Required	NO	The value can be 'success', 'pending', 'failed' or 'partial'.
OrderSubmissionDate	Optional	NO	
OrderSubmissionDate	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
DateRange	Required	NO	See Appendix E for details.
AuthCode	Conditional	NO	Is present for the following tlds (.biz, .org, .info, .us, .tv, .cc, .bz) if the order being queried is an Inbound transfer.

Example - OrderStatus Request

```
<?xml version="1.0" encoding="UTF-8"?>
<OrderStatusRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Order>
      <OrderID>1000309149</OrderID>
    </Order>
  </Body>
</OrderStatusRequest>
```

Expected Response – OrderStatus

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278540</ReferenceNumber>
    <Timestamp>2006-08-24T15:15:49EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278502</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309149</OrderID>
```

```

<OrderAction>Create</OrderAction>
<Customer>
  <UserID>789</UserID>
</Customer>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Host</ProductType>
  <Name>NS1.EXAMPLE.COM</Name>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Host</ProductType>
  <Name>NS2.EXAMPLE.COM</Name>
</LinItem>
<LinItem>
  <LinItemID>3</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.226564377</ProductID>
  <DomainName>EXAMPLE.COM</DomainName>
</LinItem>
<OrderSubmissionDate>2006-08-24T13:39:09EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>3</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
</Order>
<Status>
  <StatusCode>7007</StatusCode>
  <Description>Order Status Lookup Successful</Description>
</Status>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

Example - OrderStatus Request (for an Inbound Transfer of an org domain)

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderStatusRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>

```

```

<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <Order>
    <OrderID>1000123456</OrderID>
  </Order>
</Body>
</OrderStatusRequest>

```

Expected Response – OrderStatus (for an Inbound Transfer of an org domain)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
```

```

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>91146144</ReferenceNumber>
    <Timestamp>2006-08-24T15:16:34EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>91146142</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000123456</OrderID>
      <OrderAction>InboundTransfer</OrderAction>
      <Customer>
        <UserID>39425258</UserID>
      </Customer>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7008</StatusCode>
          <Description>Awaiting authorization from the current Administrative Contact. Transfer request is
being processed</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.191300814</ProductID>
        <DomainName>EXAMPLE.ORG</DomainName>
        <AuthCode>a@b#cpad</AuthCode>
      </LineItem>
      <OrderSubmissionDate>2006-08-24T13:39:09EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <Status>
      <StatusCode>7007</StatusCode>
      <Description>Order Status Lookup Successful</Description>
    </Status>

```

```

    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

The response to an OrderStatus Request is identical in format to the OrderResponse. The OrderStatus checks the progress of the request and reports back what has been completed and what is still pending.

2.7 User Lookup Request (UserLookupRequest.dtd)

The actions contained in the UserLookupRequest.dtd consist of:

- LookupUser
- CheckLoginNameAvailability

The following element describes the UserLookupRequest.dtd Body:

```

<!ELEMENT Body ( AuthorizedAgent, CustomerID?, (LookupUser |
CheckLoginNameAvailability) ) >

```

Table contains a detailed explanation of the Body fields.

Table 73. Body for UserLookup Request

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID?	Optional	NO	
LookupUser	Conditional	NO	
CheckLoginNameAvailability	Conditional	NO	

2.7.1 LookupUser

The following element from the UserLookupRequest.dtd defines the LookupUser action:

```

<!ELEMENT LookupUser ( UserID+ | NicHandle+ | DomainName | ProductID | LoginName+ )
>

```

A Partner should issue the **LookupUser** Request to gather all the information about one or more customers of a Partner, or one or more technical or admin contact for a Partner. A customer may issue a **LookupUser** Request to gather all of the information about him/herself. Table contains a detailed explanation of each field in the request.

Table 74. Required and Optional Fields for the LookupUser Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 73. Body for UserLookup Request for details.
LookupUser	Required	NO	
UserID	Conditional	YES	Must provide one of the following: UserID, NicHandle, DomainName, ProductID, or LoginName.
NicHandle	Conditional	YES	
DomainName	Conditional	NO	
ProductID	Conditional	NO	
LoginName	Conditional	YES	LoginName can be used to identify a customer instead of a CustomerID, allowing a customer to provide an identifier that is much easier to remember than their CustomerID.

Example - LookupUser Request

```

<?xml version="1.0" encoding="UTF-8"?>
<UserLookupRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
  <Authentication>

```

```
<PartnerID>123</PartnerID>
<PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>customer</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <LookupUser>
    <UserID>789</UserID>
  </LookupUser>
</Body>
</UserLookupRequest>
```

Expected Response – LookupUser

```
<?xml version="1.0"?>
<!DOCTYPE UserLookupResponse SYSTEM "UserLookupResponse.dtd">
<UserLookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278496</ReferenceNumber>
    <Timestamp>2006-08-24T15:18:04EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5702</StatusCode>
      <Description>User lookup successful</Description>
    </Status>
    <User>
      <Business>
        <UserID>789</UserID>
        <NicHandle>7890</NicHandle>
        <LoginName>EXAMPLECOMPANY</LoginName>
        <CompanyName>Example Company</CompanyName>
        <Address>
          <AddressLine1>123 Any Street</AddressLine1>
          <City>Herndon</City>
          <State>VA</State>
          <PostalCode>20171</PostalCode>
          <CountryCode>US</CountryCode>
        </Address>
        <Phone>703-555-0044</Phone>
        <Fax>703-555-4444</Fax>
        <Email>admin@example.com</Email>
        <LegalContact>
          <FirstName>Jennifer</FirstName>
          <LastName>Smith</LastName>
          <Address>
            <AddressLine1>123 Any Street</AddressLine1>
            <City>Herndon</City>
```

```

        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
    </Address>
    <Phone>703-555-0044</Phone>
    <Fax>703-555-4444</Fax>
    <Email>jane@example.com</Email>
</LegalContact>
<AuthQuestion>Does this work?</AuthQuestion>
<AuthAnswer>Yes.</AuthAnswer>
</Business>
</User>
</Body>
</UserLookupResponse>

```

2.7.2 CheckLoginNameAvailability

The following element from the UserLookupRequest.dtd defines the CheckLoginNameAvailability action:

```
<!ELEMENT CheckLoginNameAvailability ( LoginName) >
```

A Partner should issue the **CheckLoginNameAvailability** Request to determine whether a given LoginName is available for use. A LoginName can be used to identify a customer instead of a CustomerID, allowing a customer to provide an identifier that is much easier to remember than their CustomerID. Table contains a detailed explanation of each field in the request.

Table 75. Required and Optional Fields for the CheckLoginNameAvailability Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 73. Body for UserLookup Request for details.
CheckLoginNameAvailability	Required	NO	
LoginName	Required	NO	

Example - CheckLoginNameAvailability Request

```

<?xml version="1.0" encoding="UTF-8"?>
<UserLookupRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <CheckLoginNameAvailability>
      <LoginName>ExampleUser</LoginName>
    </CheckLoginNameAvailability>
  </Body>
</UserLookupRequest>

```

Expected Response – CheckLoginNameAvailability

```
<?xml version="1.0"?>
<!DOCTYPE UserLookupResponse SYSTEM "UserLookupResponse.dtd">
<UserLookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278492</ReferenceNumber>
    <Timestamp>2006-08-24T15:18:57EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>5705</StatusCode>
      <Description>User login name available</Description>
    </Status>
  </Body>
</UserLookupResponse>
```

2.8 Email Forwarding Request (EmailForwardingRequest.dtd)

Note: Actions in Section 2.8 apply ONLY to .name domains.

The actions contained in the EmailForwardingRequest.dtd consist of:

- CreateEmailForwardingAndDomain
- ModifyEmailForwarding
- DeleteEmailForwardingAndDomain
- RenewEmailForwardingAndDomain

The following element from the EmailForwardingRequest.dtd defines the Body:

```
<!ELEMENT Body ( AuthorizedAgent, CustomerID, Billing?, Product ) >
```

Table contains a detailed explanation of the Body fields.

Note: Only one action may be submitted in a single request.

Table 76. Body for EmailForwarding Request

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
CustomerID	Required	NO	
Billing	Optional	NO	
DiscountCode	Optional	NO	Tag ignored for processing at this time.
AutoRenew	Optional	NO	Default value is 'False'. See below for more details.
Product	Required	NO	
CreateEmailForwardingAndDomain	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ModifyEmailForwarding	Conditional	NO	
DeleteEmailForwarding	Conditional	NO	
RenewEmailForwarding	Conditional	NO	

Note: Currently, the AutoRenew flag may be set to "true" so a domain will be automatically renewed by the Network Solutions system, or it may be set to "false" (default value) so a domain will not be automatically renewed by the Network Solutions system. If the flag is not provided, then the Partner Protocol system will set the value to "false".

2.8.1 CreateEmailForwardingAndDomain

The following element from the EmailForwardingRequest.dtd defines the CreateEmailForwardingAndDomain action:

```
<!ELEMENT CreateEmailForwardingAndDomain(CreateRegistration | CreateParking) >
```

The **CreateEmailForwardingAndDomain** Request creates a new .name domain and email address, as well as defines the forwarding email address. For example, if foo.bar.name is purchased, then the foo@bar.name email address will automatically be created by the .name Registry. There is not an actual mailbox for foo@bar.name; therefore a forwarding email address must be submitted with the order. The forwarding email address should be a valid, active email box. For example, the forwarding email address for foo@bar.name could be foobar@hotmail.com. Table contains a detailed explanation of each field in the request.

Note: The accepted PurchasePeriod values for the .name TLD are 1, 2 and 10 years.

Note: If a .name domain is purchased and the ForwardingEmailAddress field is not included (just registered as a CreateRegistration or CreateParking), then email forwarding CANNOT be added later.

Note: If multiple domains are listed in one of the create actions, the same forwarding email address will be assigned to ALL of the domains.

Table 77. Required and Optional Fields for the CreateEmailForwardingAndDomain Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
CreateEmailForwardingAndDomain	Required	NO	
CreateRegistration	Conditional	NO	One of the "Conditional" elements in this table must be submitted. Note that the accepted PurchasePeriod values for the .name TLD are 1, 2 and 10 years.
CreateParking	Conditional	NO	

2.8.1.1 CreateRegistration

The following element from the EmailForwardingRequest.dtd defines the CreateRegistration action:

```
<!ELEMENT CreateRegistration ( DomainName+, PurchasePeriod, TechContact?,
AdminContact?, BillContact?, BulkWhoisOptOut?, Hosts?, ForwardingEmailAddress ) >
```

The **CreateRegistration** Request should be issued to create a .name domain and matching email address, as well as to define the forwarding email address. This domain is one that will be hosted completely separately from Network Solutions. The domain must have valid host information for the domain to properly resolve. CreateRegistration should not be used to change host information. The Partner may submit an existing host, or create a new one, however an existing host cannot be modified. To modify an existing host name or IP address, use the ModifyHost Request. Table 78. Required and Optional Fields for the CreateRegistration Request contains a detailed explanation of each field in the request.

Note: Both HostIP and HostName are required for hosts where the TLD of the host is not a TLD that the Partner Protocol currently supports. Please refer to section 1.4 for a detailed list of supported TLDs.

Note: Network Solutions processes this request asynchronously. Please refer to section 1.4 and section 1.3.2 of this document. It may take up to 48 hours before the domain is fully configured and resolves.

Table 78. Required and Optional Fields for the CreateRegistration Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
CreateEmailForwardingAndDomain	Required	NO	
CreateRegistration			
DomainName	Required	YES	A maximum of 2500 domain names are allowed per request.
PurchasePeriod	Required	NO	
TechContact	Optional	NO	See Appendix E for details. If AdminContact and/or TechContact are not provided, the default value is the customer.
AdminContact	Optional	NO	
BillContact	Optional	NO	If BillContact is not provided it defaults to the Partner's billing information.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
Hosts	Optional	NO	If no valid hosts are provided, the domain will

Field	Required	Multiple	Notes
			not resolve. Please see note above. See Appendix E for details.
DomainProtect	Optional	NO	Default is True.
ForwardingEmailAddress	Required	NO	This is the actual mailbox for the domain. It should be a valid, active mailbox.

Example – CreateEmailForwardingAndDomain Request

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Billing>
      <AutoRenew>true</AutoRenew>
    </Billing>
    <Product>
      <CreateEmailForwardingAndDomain>
        <CreateRegistration>
          <DomainName>john.doe.name</DomainName>
          <PurchasePeriod>2</PurchasePeriod>
          <ForwardingEmailAddress>John@example.com</ForwardingEmailAddress>
        </CreateRegistration>
      </CreateEmailForwardingAndDomain>
    </Product>
  </Body>
</EmailForwardingRequest>
```

Expected Response – CreateEmailForwardingAndDomain

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">

<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>105786988</ReferenceNumber>
    <Timestamp>2006-08-24T15:20:25EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>105786988</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
    </Order>
  </Body>
</OrderResponse>
```

```

<OrderID>217298144</OrderID>
<OrderAction>Create</OrderAction>
<Customer>
  <UserID>40247872</UserID>
</Customer>
<Price>36.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.198720080</ProductID>
  <DomainName>john.dioe.name</DomainName>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Queued"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>20.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>EmailForwarding</ProductType>
  <ProductID>198720081</ProductID>
  <DomainName>john.dioe.name</DomainName>
  <Name>john@dioe.name</Name>
  <Price>16.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-24T15:20:34EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.8.1.2 CreateParking

The following element from the *EmailForwardingRequest.dtd* defines the *CreateParking* action:

```
<!ELEMENT CreateParking ( DomainName+, PurchasePeriod, TechContact?, AdminContact?,
BillContact?, BulkWhoisOptOut?, ForwardingEmailAddress ) >
```

The **CreateParking** Request should be issued to create a .name domain that will be hosted and managed by Network Solutions, and to define the forwarding email address. This domain will resolve to an "Under Construction" page until the domain owner modifies the domain by either converting it to another type of domain, or by adding value-added products. Although TechContact is optional in Version 3.0, this field should not be included in the request. Any data submitted for TechContact will be ignored, and Network Solutions will be assigned as the technical contact. Table contains a detailed explanation of each field in the request.

Note: *Network Solutions processes this request asynchronously. Please refer to section 1.4 and section 1.3.2 of this document for details. It may take up to 48 hours before the domain is fully configured and resolves.*

Deprecation Note: Network Solutions will always be the TechContact so the use of TechContact is not necessary, and it will be removed in a future version of this DTD. Please avoid using this item.

Table 79. Required and Optional Fields for the CreateParking Request

Field	Required	Multiple	Notes
RequestHeader			See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
CreateEmailForwardingAndDomain	Required	NO	
CreateParking			
DomainName	Required	YES	A maximum of 2500 domain names are allowed.
PurchasePeriod	Required	NO	
TechContact	Optional	NO	Network Solutions will be the technical contact.
AdminContact	Optional	NO	If AdminContact is not provided it defaults to the UserID of the customer.
BillContact	Optional	NO	If BillContact is not provided it defaults to the Partner's billing information.
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
DomainProtect	Optional	NO	Default is True.
ForwardingEmailAddress	Required	NO	This is the actual mailbox for the domain. It should be a valid, active mailbox.
WebForwardingProduct	Optional	YES	See Appendix E for details.
ADNS	Optional	YES	See Appendix E for details.

Example – CreateParking Request

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Billing>
      <AutoRenew>False</AutoRenew>
    </Billing>
    <Product>
      <CreateEmailForwardingAndDomain>
        <CreateParking>
          <DomainName>mary.doe.name</DomainName>
          <PurchasePeriod>2</PurchasePeriod>
          <ForwardingEmailAddress>mdoe@example.com</ForwardingEmailAddress>
        </CreateParking>
      </CreateEmailForwardingAndDomain>
    </Product>
```

```
</Body>
</EmailForwardingRequest>
Expected Response – CreateParking
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278703</ReferenceNumber>
    <Timestamp>2006-08-24T15:27:17EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278703</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309178</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>36.0</Price>
      <TieredPrice Selection="False"/>
      <LinItem>
        <LinItemID>1</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Registration</ProductType>
        <ProductID>WN.D.226564412</ProductID>
        <DomainName>mary.doe.name</DomainName>
        <LinItemDetail>
          <DomainLinItemDetail>
            <DomainReservationStatus DomainReservationStatusType="Queued"/>
          </DomainLinItemDetail>
        </LinItemDetail>
        <Price>20.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>2</LinItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Parking</ProductType>
        <ProductID>WN.H.226564413</ProductID>
        <DomainName>mary.doe.name</DomainName>
        <Price>0.0</Price>
      </LinItem>
      <LinItem>
        <LinItemID>3</LinItemID>
        <Status>
```

```

    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>EmailForwarding</ProductType>
  <ProductID>226564414</ProductID>
  <DomainName>mary.doe.name</DomainName>
  <Name>mary@doe.name</Name>
  <Price>16.0</Price>
</LineItem>
<OrderSubmissionDate>2006-08-24T15:27:27EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>3</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>3</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.8.2 ModifyEmailForwarding

The following element from the EmailForwardingRequest.dtd defines the ModifyEmailForwarding action:

```
<!ELEMENT ModifyEmailForwarding ( EmailForwardingProduct+, ForwardingEmailAddress )
>
```

The **ModifyEmailForwarding** Request should be issued to modify the forwarding email address of a .name domain. Table contains a detailed explanation of each field in the request.

Note: *If multiple domains are listed for EmailForwardingProduct, then the same forwarding email address will be assigned to ALL of the domains.*

Table 80. Required and Optional Fields for the ModifyEmailForwarding Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
ModifyEmailForwarding	Required	NO	
EmailForwardingProduct	Required	YES	
EmailAddress	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ProductID	Conditional	NO	
ForwardingEmailAddress	Required	NO	

Example – ModifyEmailForwarding Request

```

<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>456</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
  </Body>
</EmailForwardingRequest>

```

```

    <CustomerID>456</CustomerID>
  <Product>
    <ModifyEmailForwarding>
      <EmailForwardingProduct>
        <EmailAddress>john@doe.name</EmailAddress>
      </EmailForwardingProduct>
      <ForwardingEmailAddress>jdoe@hotmail.com</ForwardingEmailAddress>
    </ModifyEmailForwarding>
  </Product>
</Body>
</EmailForwardingRequest>
Expected Response – ModifyEmailForwarding
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd"> <OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278563</ReferenceNumber>
    <Timestamp>2006-08-24T15:28:50EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278563</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309155</OrderID>
      <OrderAction>Modify</OrderAction>
      <Customer>
        <UserID>789</UserID>
      </Customer>
      <Price>0.0</Price>
      <TieredPrice Selection="False"/>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>EmailForwarding</ProductType>
        <ProductID>171404730</ProductID>
        <Name>john@doe.name</Name>
        <Price>0.0</Price>
      </LineItem>
      <OrderSubmissionDate>2006-08-24T15:33:50EDT</OrderSubmissionDate>
      <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```

2.8.3 DeleteEmailForwardingAndDomain

The following element from the EmailForwardingRequest.dtd defines the DeleteEmailForwardingAndDomain action:

<!ELEMENT DeleteEmailForwardingAndDomain (DomainName+ | EmailForwardingProduct+) >

The **DeleteEmailForwardingAndDomain** Request should be issued to delete the email forwarding product and the associated .name domain. Table contains a detailed explanation of each field in the request.

Table 81. Required and Optional Fields for the DeleteEmailForwardingAndDomain Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
DeleteEmailForwardingAndDomain	Required	NO	
DomainName	Conditional	YES	
EmailForwardingProduct	Conditional	YES	
EmailAddress	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ProductID	Conditional	NO	

Example – DeleteEmailForwardingAndDomain Request

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>123</UserID>
        <Password>partner</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <DeleteEmailForwardingAndDomain>
        <DomainName>John.Doe.NAME</DomainName>
      </DeleteEmailForwardingAndDomain>
    </Product>
  </Body>
</EmailForwardingRequest>
```

Expected Response – DeleteEmailForwardingAndDomain

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278611</ReferenceNumber>
    <Timestamp>2006-08-24T15:29:33EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278611</ReferenceNumber>
      <Status>
```

```

    <StatusCode>7000</StatusCode>
    <Description>Order submitted successfully and is in pending status</Description>
  </Status>
  <OrderID>1000309161</OrderID>
  <OrderAction>Delete</OrderAction>
  <Customer>
    <UserID>456</UserID>
  </Customer>
  <Price>0.0</Price>
  <TieredPrice Selection="False"/>
  <LinItem>
    <LinItemID>1</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>Registration</ProductType>
    <ProductID>WN.D.171404728</ProductID>
    <DomainName>John.Doe.NAME</DomainName>
    <Price>0.0</Price>
  </LinItem>
  <LinItem>
    <LinItemID>2</LinItemID>
    <Status>
      <StatusCode>7004</StatusCode>
      <Description>Line item submitted successfully</Description>
    </Status>
    <ProductType>EmailForwarding</ProductType>
    <ProductID>171404730</ProductID>
    <DomainName>John.Doe.NAME</DomainName>
    <Name>john@doe.name</Name>
    <Price>0.0</Price>
  </LinItem>
  <OrderSubmissionDate>2006-08-24T15:34:33EDT</OrderSubmissionDate>
  <NumberOfRecordsReturned>2</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.8.4 RenewEmailForwardingAndDomain

The following element from the EmailForwardingRequest.dtd defines the RenewEmailForwardingAndDomain action:

```
<!ELEMENT RenewEmailForwardingAndDomain ( (DomainName+ |
EmailForwardingProduct+)), PurchasePeriod >
```

The **RenewEmailForwardingAndDomain** Request should be issued to renew the email forwarding product and the associated .name domain. Table contains a detailed explanation of each field in the request.

Table 82. Required and Optional Fields for the RenewEmailForwardingAndDomain Request.

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding

Field	Required	Multiple	Notes
			Request for details.
RenewEmailForwardingAndDomain	Required	NO	
DomainName	Conditional	YES	
EmailForwardingProduct	Conditional	YES	
EmailAddress	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ProductID	Conditional	NO	
PurchasePeriod	Required	NO	See Appendix E for details.

Example – RenewEmailForwardingAndDomain Request

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>456</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>456</CustomerID>
    <Billing>
      <AutoRenew>true</AutoRenew>
    </Billing>
    <Product>
      <RenewEmailForwardingAndDomain>
        <DomainName>john.doe.name</DomainName>
        <PurchasePeriod>1</PurchasePeriod>
      </RenewEmailForwardingAndDomain>
    </Product>
  </Body>
</EmailForwardingRequest>
```

Expected Response – RenewEmailForwardingAndDomain

```
<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278607</ReferenceNumber>
    <Timestamp>2006-08-24T15:30:20EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>14278607</ReferenceNumber>
      <Status>
        <StatusCode>7000</StatusCode>
        <Description>Order submitted successfully and is in pending status</Description>
      </Status>
      <OrderID>1000309158</OrderID>
```

```

<OrderAction>Renew</OrderAction>
<Customer>
  <UserID>789</UserID>
</Customer>
<Price>18.0</Price>
<TieredPrice Selection="False"/>
<LinItem>
  <LinItemID>1</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.171404728</ProductID>
  <DomainName>john.doe.name</DomainName>
  <Price>18.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>EmailForwarding</ProductType>
  <ProductID>171404730</ProductID>
  <DomainName>john.doe.name</DomainName>
  <Name>john@doe.name</Name>
  <Price>8.0</Price>
</LinItem>
<OrderSubmissionDate>2006-08-24T15:35:20EDT</OrderSubmissionDate>
<NumberOfRecordsReturned>2</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>2</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

2.9 Email Forwarding Lookup Request (*EmailForwardingLookupRequest.dtd*)

Note: Actions in Section 2.9 apply ONLY to .name domains.

The actions contained in the *EmailForwardingLookupRequest* consist of:

- `LookupEmailForwarding`

The following element from the *EmailForwardingLookupRequest.dtd* defines the `Body`:

```
<!ELEMENT Body ( AuthorizedAgent, LookupEmailForwarding ) >
```

Table contains a detailed explanation of the `Body` fields.

Table 83. Body for EmailForwardingLookup Request

Field	Required	Multiple	Notes
AuthorizedAgent	Required	NO	See Appendix E for details.
LookupEmailForwarding	Required	NO	

2.9.1 LookupEmailForwarding

The following element from the *EmailForwardingLookupRequest.dtd* defines the `LookupEmailForwarding` action:

<!ELEMENT LookupEmailForwarding (EmailAddress | ProductID) >

The **LookupEmailForwarding** Request should be issued to retrieve the forwarding email address being used for a .name domain. Table contains a detailed explanation of each field in the request.

Note: *EmailAddress in this instance means the email address that was assigned to the .name domain when the CreateEmailForwardingAndDomain Request was submitted. For example, if foo.bar.name was purchased, the associated EmailAddress would be foo@bar.name.*

Table 84. Required and Optional Fields for the LookupEmailForwarding Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 76. Body for EmailForwarding Request for details.
LookupEmailForwarding	Required	NO	
EmailAddress	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
ProductID	Conditional	NO	

Example – LookupEmailForwarding Request

```
<?xml version="1.0" encoding="UTF-8"?>
<EmailForwardingLookupRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <LookupEmailForwarding>
      <ProductID>171404730</ProductID>
    </LookupEmailForwarding>
  </Body>
</EmailForwardingLookupRequest>
```

Expected Response – LookupEmailForwarding

```
<?xml version="1.0"?>
<!DOCTYPE EmailForwardingLookupResponse SYSTEM
"EmailForwardingLookupResponse.dtd">
<EmailForwardingLookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278568</ReferenceNumber>
    <Timestamp>2006-08-24T15:30:59EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>11702</StatusCode>
      <Description>Product lookup successful</Description>
    </Status>
    <EmailForwarding>
      <ProductID>171404730</ProductID>
      <ProductStatus>ACTIVE</ProductStatus>
    </EmailForwarding>
  </Body>
</EmailForwardingLookupResponse>
```

```

    <CustomerID>789</CustomerID>
    <PurchaseDate>12.01.2003</PurchaseDate>
    <ExpirationDate>12.01.2003</ExpirationDate>
    <EmailAddress>JOE@DOE.NAME</EmailAddress>
    <ForwardingEmailAddress>JOE@HOTMAIL.COM</ForwardingEmailAddress>
  </EmailForwarding>
</Body>
</EmailForwardingLookupResponse>

```

2.10 Security Request (SecurityRequest.dtd)

The actions contained in the SecurityRequest.dtd consist of:

- AuthenticatePartner,
- AuthenticatePartnerAndGetPartnerInfo,
- AuthenticateCustomer,
- AuthenticateCustomerAndGetCustomerInfo,
- AuthorizePartner, and
- AuthorizeCustomer.

The following element from the SecurityRequest.dtd defines the Body:

```

<!ELEMENT Body ( ( AuthenticatePartner |AuthenticatePartnerAndGetPartnerInfo |
AuthenticateCustomer | AuthenticateCustomerAndGetCustomerInfo | AuthorizePartner |
AuthorizeCustomer ) ) >

```

Table contains a detailed explanation of the Body fields.

Table 85. Body for Security Request

Field	Required	Multiple	Notes
AuthenticatePartner	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
AuthenticatePartnerAndGetPartnerInfo	Conditional	NO	
AuthenticateCustomer	Conditional	NO	
AuthenticateCustomerAndGetCustomerInfo	Conditional	NO	
AuthorizePartner	Conditional	NO	
AuthorizeCustomer	Conditional	NO	

2.10.1 AuthenticatePartner

The following element from the SecurityRequest.dtd defines the AuthenticatePartner action:

```

<!ELEMENT AuthenticatePartner ( Partner ) >

```

The **AuthenticatePartner** Request should be issued to validate that a Partner's UserID or NicHandle matches the provided password. Table contains a detailed explanation of each field in the request.

Table 86. Required and Optional Fields for the AuthenticatePartner Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthenticatePartner	Required	NO	
Partner	Required	NO	See Appendix E for details.

Example – AuthenticatePartner Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>

```

```

<Authentication>
  <PartnerID>123</PartnerID>
  <PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <AuthenticatePartner>
    <Partner>
      <UserID>123</UserID>
      <Password>partner</Password>
    </Partner>
  </AuthenticatePartner>
</Body>
</SecurityRequest>

```

Expected Response – AuthenticatePartner

```

<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278517</ReferenceNumber>
    <Timestamp>2006-08-24T15:32:10EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2000</StatusCode>
      <Description>Partner Authenticated</Description>
    </Status>
  </Body>
</SecurityResponse>

```

2.10.2 AuthenticatePartnerAndGetPartnerInfo

The following element from the SecurityRequest.dtd defines the AuthenticatePartnerAndGetPartnerInfo action:

```
<!ELEMENT AuthenticatePartnerAndGetPartnerInfo ( Partner ) >
```

The **AuthenticatePartnerAndGetPartnerInfo** Request should be issued to validate that a Partner's UserID or NicHandle matches the provided password, and it returns either the organization name or the first and last name of the Partner associated with the data provided.

Table contains a detailed explanation of each field in the request.

Table 87. Required and Optional Fields for the AuthenticatePartnerAndGetPartnerInfo Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthenticatePartnerAndGetPartnerInfo	Required	NO	
Partner	Required	NO	See Appendix E for details.

Example – AuthenticatePartnerAndGetPartnerInfo Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>

```

```

<Authentication>
  <PartnerID>123</PartnerID>
  <PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <AuthenticatePartnerAndGetPartnerInfo>
    <Partner>
      <UserID>123</UserID>
      <Password>partner</Password>
    </Partner>
  </AuthenticatePartnerAndGetPartnerInfo>
</Body>
</SecurityRequest>

```

Expected Response – AuthenticatePartnerAndGetPartnerInfo

```

<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278519</ReferenceNumber>
    <Timestamp>2006-08-24T15:33:09EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2000</StatusCode>
      <Description>Partner Authenticated</Description>
    </Status>
    <UserData>
      <OrgName>Example Company</OrgName>
    </UserData>
  </Body>
</SecurityResponse>

```

2.10.3 AuthenticateCustomer

The following element from the SecurityRequest.dtd defines the AuthenticateCustomer action:

```
<!ELEMENT AuthenticateCustomer ( Customer ) >
```

The **AuthenticateCustomer** Request should be issued to validate that a customer's UserID or NicHandle matches the provided password. Table 88 contains a detailed explanation of each field in the request.

Table 88. Required and Optional Fields for the AuthenticateCustomer Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthenticateCustomer	Required	NO	
Customer	Required	NO	See Appendix E for details.

Example – AuthenticateCustomer Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>

```

```

<Authentication>
  <PartnerID>123</PartnerID>
  <PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <AuthenticateCustomer>
    <Customer>
      <UserID>456</UserID>
      <Password>customer</Password>
    </Customer>
  </AuthenticateCustomer>
</Body>
</SecurityRequest>

```

Expected Response – AuthenticateCustomer

```

<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278523</ReferenceNumber>
    <Timestamp>2006-08-24T15:34:36EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2002</StatusCode>
      <Description>User Authenticated</Description>
    </Status>
  </Body>
</SecurityResponse>

```

2.10.4 AuthenticateCustomerAndGetCustomerInfo

The following element from the SecurityRequest.dtd defines the AuthenticateCustomerAndGetCustomerInfo action:

```
<!ELEMENT AuthenticateCustomerAndGetCustomerInfo ( Customer ) >
```

The **AuthenticateCustomerAndGetCustomerInfo** Request should be issued to validate that a customer's UserID or NicHandle matches the provided password, and it returns either the organization name or the first and last name of the customer associated with the data provided.

Table contains a detailed explanation of each field in the request.

Table 89. Required and Optional Fields for the AuthenticateCustomerAndGetCustomerInfo Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthenticateCustomerAndGetCustomerInfo	Required	NO	
Customer	Required	NO	See Appendix E for details.

Example – AuthenticateCustomerAndGetCustomerInfo Request

```

<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>

```

```

<Authentication>
  <PartnerID>123</PartnerID>
  <PartnerPassword>partner</PartnerPassword>
</Authentication>
<Comments></Comments>
</RequestHeader>
<Body>
  <AuthenticateCustomerAndGetCustomerInfo>
    <Customer>
      <UserID>456</UserID>
      <Password>customer</Password>
    </Customer>
  </AuthenticateCustomerAndGetCustomerInfo>
</Body>
</SecurityRequest>

```

Expected Response – AuthenticateCustomerAndGetCustomerInfo

```

<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278507</ReferenceNumber>
    <Timestamp>2006-08-24T15:35:51EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2002</StatusCode>
      <Description>User Authenticated</Description>
    </Status>
    <UserData>
      <OrgName>Example Company</OrgName>
    </UserData>
  </Body>
</SecurityResponse>

```

2.10.5 AuthorizePartner

The following element from the SecurityRequest.dtd defines the AuthorizePartner action:

```
<!ELEMENT AuthorizePartner ( Partner, CustomerData ) >
```

The **AuthorizePartner** Request should be issued to validate that a customer or product is associated with a Partner. For example, if a Partner wants to modify account information or a customer, it may be necessary to first make sure that the customer was created under the Partner's UserID. Table contains a detailed explanation of each field in the request.

Table 90. Required and Optional Fields for the AuthorizePartner Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthorizePartner	Required	NO	
Partner	Required	NO	See Appendix E for details.
CustomerData	Required	NO	

Field	Required	Multiple	Notes
CustomerID	Required	NO	
ProductID	Optional	NO	

Example – AuthorizePartner Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <AuthorizePartner>
      <Partner>
        <UserID>456</UserID>
        <Password>partner</Password>
      </Partner>
      <CustomerData>
        <CustomerID>456</CustomerID>
      </CustomerData>
    </AuthorizePartner>
  </Body>
</SecurityRequest>
```

Expected Response – AuthorizePartner

```
<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278508</ReferenceNumber>
    <Timestamp>2006-08-24T15:38:01EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2001</StatusCode>
      <Description>Partner Authorized</Description>
    </Status>
  </Body>
</SecurityResponse>
```

2.10.6 AuthorizeCustomer

The following element from the SecurityRequest.dtd defines the AuthorizeCustomer action:

```
<!ELEMENT AuthorizeCustomer ( Customer, ProductID ) >
```

The **AuthorizeCustomer** Request should be issued to validate that a product is associated with a customer. For example, if there is a question about which customer account was used to purchase a domain, the AuthorizeCustomer Request will allow the Partner to see if the domain is associated with a given customer. Table contains a detailed explanation of each field in the request.

Table 91. Required and Optional Fields for the AuthorizeCustomer Request

Field	Required	Multiple	Notes
RequestHeader	Required	NO	See Appendix E for details.
Body	Required	NO	See Table 85. Body for Security Request for details.
AuthorizeCustomer	Required	NO	
Customer	Required	NO	See Appendix E for details.
ProductID	Required	NO	

Example – AuthorizeCustomer Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SecurityRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <AuthorizeCustomer>
      <Customer>
        <UserID>456</UserID>
        <Password>customer</Password>
      </Customer>
      <ProductID>WN.D.123456789</ProductID>
    </AuthorizeCustomer>
  </Body>
</SecurityRequest>
```

Expected Response – AuthorizeCustomer

```
<?xml version="1.0"?>
<!DOCTYPE SecurityResponse SYSTEM "SecurityResponse.dtd">
<SecurityResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>14278513</ReferenceNumber>
    <Timestamp>2006-08-24T15:38:22EDT</Timestamp>
    <Comments></Comments>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>2003</StatusCode>
      <Description>User Authorized</Description>
    </Status>
  </Body>
</SecurityResponse>
```

3.0 RESPONSES

As previously mentioned the XML transaction structure is made up of a request/response pair. The request contains the action or operation that the Partner wants the XML server to execute and the response contains the execution status and or the data requested by the Partner.

This section focuses on the XML response structure including the XML format, and the required versus optional fields. Examples of specific responses can be found within Section 2.0 with the corresponding request. The remainder of this section describes each XML response in greater detail and provides a link to the corresponding request section so an example may be easily located. Appendix E also contains some of the common data elements found in various DTDs. Appendix D contains the validation rules and sample values for each field.

3.1 User Response (*UserResponse.dtd*)

User Responses are returned when a Partner submits a UserRequest in order to create or modify a business account or individual account. The following element from the UserResponse.dtd defines the Body of the User Response:

```
<!ELEMENT Body (Status, UserID?, NicHandle?, LoginName?) >
```

Section 2.1 contains examples of User Responses. Table contains a detailed explanation of each field in the response.

Table 92. Required and Optional Fields for the User Response.

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Status	Required	NO	See Appendix E for details.
UserID	Optional	NO	
NicHandle	Optional	NO	
LoginName	Optional	NO	

3.2 Order Response (*OrderResponse.dtd*)

Order Responses are returned for two different requests, OrderRequest and OrderStatusRequest. The following element from the OrderResponse.dtd defines the Body of the Order Response:

```
<!ELEMENT Body ( Order*, Status?, NumberOfRecordsReturned?,
NumberOfRecordsAvailable?) >
```

Sections 2.2 and 2.6 contain examples of Order Responses. Table contains a detailed explanation of each field in the response.

Table 93. Required and Optional Fields for the Order Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body			
Order	Conditional	YES	0 or more Orders will appear depending on input request type.
ReferenceNumber	Required	NO	The unique tracking number for the request used internally by Network Solutions.
Status	Required	NO	See Appendix E for details.
OrderID	Conditional	NO	OrderID is generated when an order is successfully submitted and may be used to track the order.
OrderAction	Required	NO	Examples are 'Create', 'Modify', etc
Customer	Required	NO	
Price	Optional	NO	This is the cost of the order.
LineItem	Conditional	YES	There will be zero or more LineItem(s) for

Field	Required	Multiple	Notes
			each product requested.
LineItemID	Optional	NO	
Status	Required	NO	See Appendix E for details.
ProductType	Required	NO	
ProductID	Optional	NO	
DomainName	Optional	NO	
AuthCode	Optional	NO	Domain password
Name	Optional	NO	
LineItemDetail	Optional	NO	Contains any additional characteristics or statuses for the given line item type.
DomainLineItemDetail	Required	NO	Only domain line items contain additional detail as of this release.
DomainReservationStatus	Required	NO	For a domain creation request, this indicates whether the domain was synchronously reserved in the registry or queued for registration. The possible values are Reserved or Queued. This status will only be returned by the OrderRequest and will not be returned on a subsequent OrderStatusRequest.
DomainReservationStatusType	Required	NO	
Reserved	Conditional	NO	
Queued	Conditional	NO	
SSLDetail	Optional	NO	
Certificate	Conditional	NO	If request type is DownLoadCertificate, then the result will be returned as CDATA in this element.
SealHeader	Conditional	NO	If request type is GetSealCode, then the java script for the header will be returned as CDATA in this element.
SealBody	Conditional	NO	If request type is GetSealCode, then the java script for the body will be returned as CDATA in this element.
Price	Optional	NO	This is the extended cost of the line item (line item unit price multiplied by the term).
OrderSubmissionDate	Optional	NO	
NumberOfRecordsReturned	Optional	NO	The number of LineItems returned in this response for this Order. If the OrderResponse is returned in response to an OrderRequest, then this number will always equal the NumberOfRecordsAvailable. But if returned in response to an OrderStatusRequest, then a maximum of 25,000 line items will be returned, so this number will never be larger than 25,000.

Field	Required	Multiple	Notes
NumberOfRecordsAvailable	Optional	NO	The number of Lineltems in this response for the given Order. The maximum number of line items returned in a response to an OrderStatusLookup is 25,000. So if the lookup result set was larger than this, then this number will represent the total size of the result set and will be larger than the NumberOfRecordsReturned.
Status	Optional	NO	See Appendix E for details.
NumberOfRecordsReturned	Optional	NO	The number of Orders returned in this response.
NumberOfRecordsAvailable	Optional	NO	The number of Orders in the entire result set. On large query result sets, this number may be larger than the NumberOfRecordsReturned.

3.3 Lookup Response (LookupResponse.dtd)

Lookup Responses will vary depending on the type of Lookup Request action. The Lookup Response supports the following types of lookup actions:

- Domain
- Host

The following element from the LookupResponse.dtd defines the Body of the Lookup Response:

```
<!ELEMENT Body (( Domain* | HostData* ), Status) >
```

Section 2.3 contains examples of Lookup Responses. Table contains a detailed explanation of common elements for both types of Lookup Responses.

Table 94. Required and Optional Fields for All Lookup Responses

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Domain	Conditional	YES	
HostData	Conditional	YES	
Status	Required	NO	See Appendix E for details.

3.3.1 Lookup Response for Domain

The following element from the LookupResponse.dtd defines what is returned when retrieving information for a domain:

```
<!ELEMENT Domain ( ProductID, ProductType, DomainNameProperties, ProductStatus, CustomerID, AdminContact?, TechContact?, BillContact?, BulkWhoisOptOut?, PurchaseDate, ExpirationDate, A*, CNAME*, MX*, NicHandle?, HostData* ) >
```

Table contains a detailed explanation of each field in the response.

Table 95. Required and Optional Fields for Domain Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Domain	Conditional	YES	
ProductID	Required	NO	
ProductType	Required	NO	
DomainNameProperties	Required	NO	
ProductStatus	Required	NO	ACTIVE or INACTIVE.

Field	Required	Multiple	Notes
DomainProtect	Required	NO	TRUE or FALSE
LegalLock	Required	NO	TRUE or FALSE
CustomerID	Required	NO	
AdminContact	Optional	NO	See Appendix E for details.
TechContact	Optional	NO	
BillContact	Optional	NO	
BulkWhoisOptOut	Optional	NO	
PurchaseDate	Required	NO	Default value is 'N'.
ExpirationDate	Required	NO	
A	Optional	YES	See Appendix E for details.
CNAME	Optional	YES	
MX	Optional	YES	
NicHandle	Optional	NO	
HostData	Optional	YES	See Appendix E for details.

3.3.2 Lookup Response for Host

The following element from the LookupResponse.dtd defines what is returned when retrieving information for a host:

```
<!ELEMENT HostData ( HostID, HostName, HostCoordinator?, HostIP1?, HostIP2?, HostIP3?, HostIP4?, HostIP5? ) >
```

Deprecation Note: *The HostID and HostCoordinator elements found within the HostData element were deprecated as of version 3.3. Please do not use these elements. They will be removed in a future version.*

Table contains a detailed explanation of each field in the response.

Table 96. Required and Optional Fields for Host Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
HostData	Conditional	YES	See Appendix E for details.
HostName	Required	NO	
HostIP1	Optional	NO	
HostIP2	Optional	NO	
HostIP3	Optional	NO	
HostIP4	Optional	NO	
HostIP5	Optional	NO	

3.4 Partner Manager Response (PartnerManagerResponse.dtd)

PartnerManager Responses will vary depending on the action type in the PartnerManager Request. PartnerManager Response will return one or a list of the following elements:

- Domain, returned by FindAllDomainsForCustomer
- Customer, returned by FindAllCustomersForPartner
- CustomerProductConsolidationResult, returned by ConsolidateAllCustomerProducts or ConsolidateSelectedCustomerProducts
- PartnerData, returned by GetPartnerData
- SSLInstance, returned by FindDetailedSSLProduct

The following element from the PartnerManagerResponse.dtd defines the Body of the PartnerManager Response:

```
<!ELEMENT Body ( Status, ( Domain* | Customer* | CustomerProductConsolidationResult | PartnerData), DateRange?, NumberOfRecordsReturned?, NumberOfRecordsAvailable? ) >
```

Note: *NumberOfRecordsAvailable* is the number of records in the Network Solutions system matching the data requested. *NumberOfRecordsReturned* is the number of records the system was able to return in a timely manner in order to fulfill the request. If the *NumberOfRecordsReturned* is less than the *NumberOfRecordsAvailable*, then the search parameters should be narrowed until the numbers are equal. A maximum of 100,000 records will be returned by the system at one time.

Section 2.4 contains examples of PartnerManager Responses.

Table 97. Common Fields for All PartnerManager Responses shows the common field information for all PartnerManager Responses.

Table 97. Common Fields for All PartnerManager Responses

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Status	Required	NO	See Appendix E for details.
Domain	Conditional	YES	0 or more depending on request type.
Customer	Conditional	YES	0 or more depending on request type.
CustomerProductConsolidationResult	Conditional	NO	
PartnerData	Conditional	NO	
DateRange	Optional	NO	See Appendix E for details.
NumberOfRecordsReturned	Optional	NO	See Note above.
NumberOfRecordsAvailable	Optional	NO	

3.4.1 PartnerManager Responses for FindAllDomainsForCustomer

The following element from PartnerManager Response defines what is returned for FindAllDomainsForCustomer and FindAllDomainsForPartner actions:

```
<!ELEMENT Domain ( ProductID?, DomainName, ProductType, NicHandle?, AutoRenew,
PurchaseDate, ExpirationDate, BulkWhoisOptOut?, CustomerID? ) >
```

Currently, the **AutoRenew** flag may be set to "true" so a domain will be automatically renewed by the Network Solutions system, or it may be set to "false" (default value) so a domain will not be automatically renewed by the Network Solutions system, however it should be noted that Network Solutions is not currently processing any automatic renewals. Once this feature is turned on, these flags will be recognized, and automatic renewals will be processed accordingly.

Section 2.4 contains examples of PartnerManager Responses. Table 98. Required and Optional Fields for the FindAllDomains Response contains a detailed explanation of each field in the response.

Table 98. Required and Optional Fields for the FindAllDomains Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Domain	Conditional	YES	If any domains exist for the customer or Partner then there will be one entry per domain.
ProductID	Optional	NO	
DomainName	Required	NO	
ProductType	Required	NO	
DomainProtect	Required	NO	TRUE or FALSE
LegalLock	Required	NO	TRUE or FALSE
NicHandle	Optional	NO	
AutoRenew	Required	NO	Default value is 'False'. See above for more details.
PurchaseDate	Required	NO	
DateRange	Required	NO	
ExpirationDate	Required	NO	
DateRange	Required	NO	
BulkWhoisOptOut	Optional	NO	Default value is 'N'.
CustomerID	Optional	NO	

3.4.2 PartnerManager Response for FindAllCustomersForPartner

The following element from PartnerManager Response defines what is returned for FindAllCustomersForPartner action:

```
<!ELEMENT Customer ( Individual | Business ) >
```

Section 2.4 contains examples of PartnerManager Responses. Table 99. Required and Optional Fields for the FindAllCustomersForPartner Response contains a detailed explanation of each field in the response.

Table 99. Required and Optional Fields for the FindAllCustomersForPartner Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Customer	Conditional	YES	If any customers exist for the Partner then there will be one entry per customer.
Individual	Conditional	YES	
UserID	Required	NO	
NicHandle	Optional	NO	
LoginName	Optional	NO	
FirstName	Required	NO	
MiddleName	Optional	NO	
LastName	Required	NO	
Business	Conditional	YES	
UserID	Required	NO	
NicHandle	Optional	NO	
LoginName	Optional	NO	
CompanyName	Required	NO	
LegalContactFirstName	Required	NO	See Appendix E for details.
LegalContactMiddleName	Optional	NO	
LegalContactLastName	Required	NO	

3.4.3 PartnerManager Responses for GetPartnerData

The following element from PartnerManager Response defines what is returned for FindAllDomainsForCustomer and FindAllDomainsForPartner actions:

```
<!ELEMENT PartnerData ( AccountBalance? ) >
```

```
<!ATTLIST PartnerData AccountStatus (ACTIVE | SUSPENDED) #REQUIRED >
```

The optional `AccountBalance` element indicates the current credit or prepaid balance for the partner's account. If the partner maintains neither a credit balance nor a prepaid balance then the `AccountBalance` element is not applicable and will not be returned. The `AccountStatus` element indicates whether the partner's account is `SUSPENDED` or `ACTIVE`.

Normally a partner account is `ACTIVE`, but if a `PARTNER` account is `SUSPENDED` then the following types of requests will be rejected:

EmailForwardingRequest:
 CreateRegistration
 CreateParking
 RenewRegistration
 RenewParking
 CascadeRenewDomain
 InboundTransferToRegistration
 InboundTransferToParking
 TransferInternalToRegistration
 TransferInternalToParking

ConvertToRegistration
ConvertToParking

EmailForwardingRequest:
CreateEmailForwardingAndDomain
RenewEmailForwardingAndDomain

PartnerManagerRequest:
ConsolidateAllCustomerProducts
ConsolidateSelectedCustomerProducts

Section 2.4 contains examples of PartnerManager Responses. Table contains a detailed explanation of each field in the response.

Table 100. Required and Optional Fields for the GetPartnerData Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
PartnerData	Conditional	NO	
AccountBalance	Optional	NO	Will not be returned if the partner does not maintain a credit balance or prepaid balance.
AccountStatus	Required	NO	ACTIVE, or SUSPENDED

3.4.4 PartnerManager Responses for FindAllSSLProductsForPartner and FindSSLProductsForPartner

The following element from PartnerManager Response defines what is returned for FindAllSSLProductsForPartner and FindSSLProductsForPartner

```
<ELEMENT Product (ProductID, ProductType, Name, DomainName?, PurchasePeriod?, PurchaseDate?, ExpirationDate, CustomerID?, Status, IsRenewable?, IsReissueable?, HasEverBeenIssued?, SSLInstance*)>
```

Table 98. Required and Optional Fields for the FindAllDomains Response contains a detailed explanation of each field in the response.

Table 101. Required and Optional Fields for the FindAllSSLProductsForPartner and FindSSLProductForPartner Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Product	Conditional	YES	If any SSL product exist for the customer or Partner then there will be one entry per SSL product.
ProductID	Optional	NO	
ProductType	Required	NO	All SSL products will have the value of "SSL_CERTS"
Name	Required	NO	Name of product, possible values are: SiteSafe(TM) SSL Certificate – Basic SiteSafe(TM) SSL Certificate – Wildcard SiteSafe(TM) SSL Certificate – Pro
DomainName	Required	NO	Domain name of certificate or site confirm seal.
PurchaseDate	Required	NO	Purchase date
ExpirationDate	Required	NO	Expiration date

CustomerID	Required	NO	
Status	Required	NO	Status of the SSL product.

3.4.5 PartnerManager Response for FindDetailedSSLProduct

The following element from PartnerManager Response defines what is returned for FindDetailedSSLProduct

<IELEMENT Product (ProductID, ProductType, Name, DomainName?, PurchasePeriod?, PurchaseDate?, ExpirationDate, CustomerID?, Status, IsRenewable?, IsReissuable?, HasEverBeenIssued?, SSLInstan

Table 102. contains a detailed explanation of each field in the response.

Table 102. Required and Optional Fields for the FindDetailedSSLProduct Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Product	Conditional	YES	If any SSL product exist for the customer or Partner then there will be one entry per SSL product.
ProductID	Optional	NO	
ProductType	Required	NO	All SSL products will have the value of "SSL_CERTS"
Name	Required	NO	Name of product, possible values are: SiteSafe(TM) SSL Certificate – Basic SiteSafe(TM) SSL Certificate – Wildcard SiteSafe(TM) SSL Certificate – Pro SiteSafe(TM) SSL Certificate - Extended Validation
DomainName	Required	NO	Domain name of certificate or site confirm seal.
PurchaseDate	Required	NO	Purchase date
ExpirationDate	Required	NO	Expiration date
CustomerID	Required	NO	
Status	Required	NO	Status of the SSL product.
SSLInstance	Optional	NO	If any SSL instance exist for the product.
ServiceInstID	Required	NO	Instance ID for the Certificate or Site Confirm Seal
CSR	Optional	NO	Certificate Signing Request – will be returned as CDATA.
ServerSelectionID	Optional	NO	Server selection id
DomainName	Required	NO	Domain name for the Certificate or Domain name/URL for Site Confirm Seal.
IssuedDate	Optional	NO	If the instance has been issued, then the issued date will be returned.
IsRevocable	Optional	NO	Indicates if the instance can be revoked. Possible values are 'Y' or 'N'.
IsDownloadable	Optional	NO	Indicates if the instance can be downloaded. Possible values are 'Y' or 'N'.
ValidationContact	Required	NO	Validation contact data. Please see CreateSSL request for description of elements.
OrderContact	Required	NO	Order contact data. Please see CreateSSL in section 2.2.1.6 for description of elements.

CertificateRequester	Optional	NO	Certificate Requester data. Applies to EV service only - see CreateSSL in section 2.2.1.6 for description of elements.
CertificateApprover	Optional	NO	Certificate Approver data. Applies to EV service only - see CreateSSL in section 2.2.1.6 for description of elements.
ContractSigner	Optional	NO	Contract Signer data. Applies to EV service only see CreateSSL in section 2.2.1.6 for description of elements.

3.5 Available Response (*AvailableResponse.dtd*)

The status returned with a domain in an Available Response will be one of the following:

- Available,
- NotAvailable,
- Premium,
- Invalid,
- Failed,
- MaybeTransferrable, or
- NotTransferrable.

The following element from the AvailableResponse.dtd defines the Body of the Available Response:

```
<!ELEMENT Body ( Available?, NotAvailable?, Premium?, Invalid?, Failed?,
MaybeTransferrable?, NotTransferrable? ) >
```

Section 3.5 contains examples of Available Responses. Table contains a detailed explanation of each field in the response.

Table 103. Required and Optional Fields for the VerifyDomainName Response

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Available	Conditional	YES	
NotAvailable	Conditional	YES	
Premium	Conditional	YES	This is a premium domain. (See "price" below).
Invalid	Conditional	YES	This domain contains invalid syntax.
Failed	Conditional	YES	This domain failed because the Registry was not available.
MaybeTransferrable	Conditional	YES	This response will only be returned in response to an IsDomainCandidateForInboundTransfer request. A domain name may be a candidate for inbound transfer if it is already registered, and if it is not already managed by the Network Solutions registrar.
TransferDetail	Required	NO	
TransferType	Conditional	NO	This element indicates the type of transfer that would be appropriate for the domain name. The possible values are Inbound or

			Internal.
InternalTransferDetail	Optional	NO	
InternalTransferCategory	Required	NO	This element indicates the category of Internal transfer that would result if a TransferInternalToRegistration/Parking request was initiated for a domain. The possible values are: RetailToWholesale, WholesaleToWholesaleInPartner, WholesaleToWholesaleCrossPartner.
ProductType	Required	NO	This elements indicates the type of domain product. The possible values are Registration or Parking.
NotTransferrable	Conditional	YES	This response will only be returned in response to an IsDomainCandidateForInboundTransfer request. A domain name is not a candidate for inbound transfer if it is not already registered, and if it is already managed by the Network Solutions registrar.
NotTransferrableReasons	Required	YES	This element will be provided in response to IsDomainCandidateForInternalTransfer.
NotTransferrableReason	Required	YES	The statusCode and Description contained in this element indicate why a domain name may not be in a state that allows an Internal Transfer (RNCA) to be initiated. Please refer to Appendix B for the possible statusCode values and their meaning.
StatusCode	Required	NO	See Appendix E for details.
Description	Required	NO	See Appendix E for details.
DomainName	Required	YES	
Encoded	Optional	NO	For international domains.
Price	Optional	NO	For premium domains, this will list the price of the domain.
RespondingService	Required	YES	The contained ServiceName element contains the name of the service from which the domain availability response was received.
ServiceName	Conditional	NO	The name of the service from which the domain availability response was received. The possible values are: Registry or Whois or ZoneFiles or LocalCache or NotIndicated.

3.6 Order Status Response

There is no OrderStatus Response. The response to an OrderStatus Request is an OrderResponse. See Section 3.2.

3.7 User Lookup Response (UserLookupResponse.dtd)

UserLookup Response will return data for the existing individual or business customer specified in the UserLookup Request. The following element defines the Body of the UserLookup Response:

```
<!ELEMENT Body ( Status, User? ) >
```

Section 3.7 contains examples of UserLookup Responses. Table contains a detailed explanation of each field in the response.

Table 104. Required and Optional Fields for the UserLookup Response.

Field	Required	Multiple	Notes
-------	----------	----------	-------

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Status	Required	NO	See Appendix E for details.
User	Required	NO	Will be either Individual or Business.
Individual	Conditional	NO	
UserID	Required	NO	
NicHandle	Optional	NO	
LoginName	Optional	NO	
FirstName	Required	NO	
MiddleName	Optional	NO	
LastName	Required	NO	
Address	Required	NO	See Appendix E for details.
Phone	Required	NO	
Fax	Optional	NO	
Email	Required	NO	
AuthQuestion	Required	NO	
AuthAnswer	Required	NO	
SupplementalRegistryData	Optional	NO	See Appendix E for details.
Business	Conditional	NO	
UserID	Required	NO	
NicHandle	Optional	NO	
LoginName	Optional	NO	
CompanyName	Required	NO	
Address	Required	NO	See Appendix E for details.
Phone	Required	NO	
Fax	Optional	NO	
Email	Optional	NO	
LegalContact	Required	NO	See Appendix E for details.
AuthQuestion	Required	NO	
AuthAnswer	Required	NO	
SupplementalRegistryData	Optional	NO	See Appendix E for details.

3.8 Email Forwarding Response

There is no EmailForwarding Response. The response to an EmailForwarding Request is an Order Response. See Section 3.2.

3.9 Email Forwarding Lookup Response (EmailForwardingLookupResponse.dtd)

EmailForwardingLookup Response will return data for a .name domain with email forwarding.

The following element defines the Body of the EmailForwardingLookup Response:

```
<!ELEMENT Body ( Status, EmailForwarding? ) >
```

Section 1.3.9 contains examples of EmailForwardingLookup Responses. Table contains a detailed explanation of each field in the response.

Table 105. Required and Optional Fields for the EmailForwardingLookup Response.

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Status	Required	NO	See Appendix E for details.

Field	Required	Multiple	Notes
EmailForwarding	Optional	NO	
ProductID	Required	NO	
ProductStatus	Required	NO	ACTIVE or INACTIVE.
CustomerID	Required	NO	
PurchaseDate	Required	NO	
ExpirationDate	Required	NO	
EmailAddress	Required	NO	
ForwardingEmailAddress	Required	NO	

3.10 Security Response (*SecurityResponse.dtd*)

Security Response will return data that authenticates or authorizes a Partner or customer.

The following element defines the Body of the Security Response:

```
<!ELEMENT Body ( Status, UserData?) >
```

Section 1.3.10 contains examples of Security Responses. Table contains a detailed explanation of each field in the response.

Table 106. Required and Optional Fields for the Security Response.

Field	Required	Multiple	Notes
ResponseHeader	Required	NO	See Appendix E for details.
Body	Required	NO	
Status	Required	NO	See Appendix E for details.
UserData	Optional	NO	
OrgName	Optional	NO	
FirstName	Optional	NO	
MiddleName	Optional	NO	
LastName	Optional	NO	

4.0 ERROR HANDLING

The Partner Protocol provides the Partner with detailed information whenever an error occurs. There are three categories of errors:

- Malformed XML errors,
- Validation Errors, and
- Data Errors.

This section describes each type of error, and provides an example.

4.1 Malformed XML

When the Partner Protocol server receives XML that it cannot parse (due to missing tags, extra "<" or ">", or other reasons), then it will respond with an Error Response. This response will not contain the original request, and the <OriginalData> tag will be empty. Table 107 contains a detailed explanation of each field in the response.

Table 107. Fields for the Error Response.

Field	Notes
ResponseHeader	
ReferenceNumber	Unique Transaction reference number.
Timestamp	Time the response was sent. Uses ISO8601; Format is YYYY-MM-DDThh:mm:ssTZD
Comments	Optional comment text.
Body	
StatusCode	Code identifying the status or error. See Appendix A.
Description	Message describing the status or error. See Appendix A.

Field	Notes
OriginalData	The original XML request received.

Below is an example of an invalid request, and the corresponding response.

REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
<UserRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
    <Comments></Comments>
  </RequestHeader>
  <Body>
    <CreateIndividual>
      <LoginName>SampleUser</LoginName>
      <Password>customer</Password>
      <FirstName>John</FirstName>
      <MiddleName>Hank</MiddleName>
      <LastName>Doe</LastName>
      <Address>
        <AddressLine1>123 Any Street</AddressLine1>
        <City>Herndon</City>
        <State>VA</State>
        <PostalCode>20171</PostalCode>
        <CountryCode>US</CountryCode>
      </Address>
      <Phone>703-555-0001</Phone>
      <Fax>703-555-0002</Fax>
      <Email>john@example.com</Email>
      <AuthQuestion>Does this work?</AuthQuestion>
      <AuthAnswer>Yes.</AuthAnswer>
    </CreateBusiness>
  </Body>
</UserRequest>
```

RESPONSE:

```
<?xml version="1.0"?>
<!DOCTYPE ErrorResponse SYSTEM "ErrorResponse.dtd">
<ErrorResponse>
  <ResponseHeader>
    <ReferenceNumber>222700</ReferenceNumber>
    <Timestamp>2006-08-24T15:46:27EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <StatusCode>-12000</StatusCode>
    <Description>XML is malformed</Description>
    <OriginalData>
      <ValidationError>
        <PathName>/UserRequest</PathName>
        <StatusCode>NV-001</StatusCode>
        <Description><![CDATA[XML Malformed. Parser error: Expecting </CreateIndividual>
got </CreateBusiness> ]]></Description>
      </ValidationError>
    </OriginalData>
  </Body>
</ErrorResponse>
```

Tags mismatch.



```

    <PathName>/UserRequest/Body</PathName>
    <StatusCode>NV-001</StatusCode>
    <Description><![CDATA[XML Malformed. Parser error: Expecting </CreateIndividual>
got </Body> ]]></Description>
  </ValidationError>
</ValidationError>
  <PathName>/UserRequest/Body/CreateIndividual</PathName>
  <StatusCode>NV-001</StatusCode>
  <Description><![CDATA[XML Malformed. Parser error: Expecting </CreateIndividual>
got </UserRequest> ]]></Description>
    </ValidationError>
  </OriginalData>
</Body>
</ErrorResponse>

```

4.2 Validation Errors

Starting with release 3.0 the Partner Protocol provides significantly enhanced support for request validation. The following improvements have been made:

- The PathName element in the ValidationError element will now consistently contain the fully qualified name of the DTD element that failed validation and will adhere to the XQL standard path name notation. This improvement will also be evident for 2.0 and 2.1-based requests. An example of a typical PathName is:

```
<PathName>/OrderRequest/Body/Product/CreateRegistration/DomainName[0]</PathName>
```

PathNames of the following form will no longer appear in response messages: <PathName> state: vv </PathName>. Please refer to the response examples in this section for further clarification.

- Validation rules are now applied more thoroughly and consistently, resulting in more deterministic responses.
- OrderResponse messages will now consistently allocate any Product/LineItem validation errors to the ValidationError elements located in the LineItem's Status element rather than the Order's Status element. Any ValidationError elements found in the Order's Status element will apply only to Order level attributes like CustomerID. Please refer to the response examples in this section for further clarification.
- ValidationError codes and associated status response codes have been made more consistent. Please refer to Appendix B for the updated list of validation status codes that may be returned. For example, the mixed use of the generic Status Code "-12001 XML is invalid" and the more precise "-245 Data validation error" has been consolidated so that only "-245 Data validation error" will be returned in the Status of the Body element if validation errors were encountered. Similarly, the mixed use of "-2305 Input data failed pattern validation" and "-2319 Invalid value" has been simplified by consolidating on the use of only "-2319 Invalid value". Please refer to the response examples in this section for further clarification.
- Request validation no longer stops and returns prior to discovering all validation errors in the request. The complete request is subjected to validation and all validation errors are returned. This prevents the need to submit a single request multiple times to discover and

rectify multiple validation errors. Please refer to the response examples in this section for further clarification.

Prior to processing each request, the Partner Protocol server validates that each data element in the request adheres to the validation rules listed in Appendix D. For example, when validation errors are discovered in a UserRequest (e.g. CreateBusiness, CreateIndividual, ModifyBusiness, ModifyIndividual), the Partner Protocol server will return the validation error status code –245 along with the set of one or more ValidationError elements. Each ValidationError element will precisely identify (1) the fully qualified name of the DTD element that failed validation, (2) the error code of the validation rule that was violated, and (3) the text message associated with the error code.

The fully qualified name of the DTD element that failed validation will be located in the PathName element of the ValidationError. The value in the PathName element adheres to the XML Query Language (XQL) standard for specifying fully qualified DTD element names. And in keeping with the XQL standard, when a value in the PathName refers to an individual item within a node-set the first item is referred to as item "0" (zero), not item "1". For example:

<PathName>/OrderRequest/Body/Product/CreateRegistration/DomainName[0]</PathName>

refers to the first item in the list of DomainNames, and

<PathName>/OrderRequest/Body/Product/CreateRegistration/DomainName[1]</PathName>

refers to the second item in a node-set .

In addition to the PathName of the invalid element, one of the validation error codes found in Appendix B will be returned in the StatusCode element in the ValidationError, along with the Description element that describes the type of validation error that the StatusCode represents. If a request results in a validation error, please examine the value passed in for the invalid element(s) to verify proper formatting. The StatusCode descriptions in Appendix B, in concert with the data validation requirements in Appendix D, and the DTD should provide precise guidance in identifying the possible causes of the validation error. Table 108 outlines the fields that describe the validation error response.

Table 108. Fields that Highlight Validation Errors.

Field	Required	Multiple	Notes
StatusCode	Required	NO	Error code identifying the error. See Appendix A
Description	Required	NO	Error message describing the error. See Appendix A.
ValidationError	Optional	YES	Optional list of validation errors.
PathName	Required	NO	Fully qualified path to the element in error.
StatusCode	Required	NO	Type of validation error.
Description	Required	NO	Description of the type of validation error.

The following example highlights some data validation errors.

REQUEST:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<UserRequest>
```

```
<RequestHeader>
```

```
<VERSION_4_6 />
```

```
<Authentication>
```

```
<PartnerID>123</PartnerID>
```

```
<PartnerPassword>partner</PartnerPassword>
```

```
</Authentication>
```

```
</RequestHeader>
```

```
<Body>
```

```
<CreateIndividual>
```

```
<LoginName>ExampleJenny</LoginName>
```

```
<Password>Example</Password>
```

```
<FirstName>Jenny</FirstName>
```

```
<LastName>Smith</LastName>
```

```
<Address>
```

```
<AddressLine1>12345 The Street With The Very Long Name Avenue</AddressLine1>
```

```

<City>Herndon</City>
<State>VA</State>
<PostalCode>20171</PostalCode>
<CountryCode>US</CountryCode>
</Address>
<Phone>703-867-5309</Phone>
<Email>jenny@example</Email>
<AuthQuestion>Does this work?</AuthQuestion>
<AuthAnswer>Yes.</AuthAnswer>
</CreateIndividual>
</Body>
</UserRequest>

```

RESPONSE:

```

<!DOCTYPE UserResponse SYSTEM "UserResponse.dtd">

```

```

<UserResponse>
  <ResponseHeader>
    <VERSION_4_6></VERSION_4_6 >
    <ReferenceNumber>25501488</ReferenceNumber>
    <Timestamp>2006-08-24T15:49:29EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>-245</StatusCode>
      <Description>Data Validation Error</Description>
      <ValidationError>
        <PathName>/UserRequest/Body/CreateIndividual/Address/AddressLine1</PathName>
        <StatusCode>-2319</StatusCode>
        <Description>Invalid value</Description>
      </ValidationError>
      <ValidationError>
        <PathName>/UserRequest/Body/CreateIndividual/Email</PathName>
        <StatusCode>-2319</StatusCode>
        <Description>Invalid value</Description>
      </ValidationError>
    </Status>
  </Body>
</UserResponse>

```

The overall Status Code.

Address Error.

Email Error.

REQUEST:

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>

```

Invalid character
in domain name.

```

<CustomerID>789</CustomerID>
<Product>
  <CreateRegistration>
    <DomainName>errorexample.com</DomainName>
    <DomainName>error!example.net</DomainName>
    <DomainName>errorexample.org</DomainName>
  <PurchasePeriod>1</PurchasePeriod>
  <Hosts>
    <Host1>
      <Host>
        <HostName>ns1.errorexample.org</HostName>
        <HostIP1>141.156.4.82</HostIP1>
      </Host>
    </Host1>
    <Host2>
      <Host>
        <HostName>ns2.errorexample.org</HostName>
        <HostIP1>141.156.4.X</HostIP1>
      </Host>
    </Host2>
  </Hosts>
</CreateRegistration>
</Product>
</Body>
</OrderRequest>

```



Invalid IP address.

RESPONSE:

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25501587</ReferenceNumber>
    <Timestamp>2006-08-24T15:49:52EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>25501587</ReferenceNumber>
      <Status>
        <StatusCode>7001</StatusCode>
        <Description>Part of requested order submitted and is being fulfilled</Description>
      </Status>
      <OrderID>130121665</OrderID>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>456</UserID>
      </Customer>
      <Price>34.0</Price>
      <LineItem>
        <LineItemID>1</LineItemID>
        <Status>
          <StatusCode>7004</StatusCode>
          <Description>Line item submitted successfully</Description>
        </Status>
        <ProductType>Host</ProductType>
        <Name>ns1.errorexample.org</Name>
      </LineItem>
    </Order>
  </Body>
</OrderResponse>

```

```
<Price>0.0</Price>
</LinItem>
<LinItem>
  <Status>
    <StatusCode>-2320</StatusCode>
    <Description>Invalid line item, validation error.</Description>
    <ValidationError>

<PathName>/OrderRequest/Body/Product/CreateRegistration/Hosts/Host2/Host/HostIP1</Path
Name>
    <StatusCode>-2319</StatusCode>
    <Description>Invalid value</Description>
    </ValidationError>
  </Status>
  <ProductType>Host</ProductType>
  <Name>ns2.errorexample.org</Name>
  <Price>0.0</Price>
</LinItem>
<LinItem>
  <LinItemID>2</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
  <ProductID>WN.D.174989427</ProductID>
  <DomainName>errorexample.com</DomainName>
  <LinItemDetail>
    <DomainLinItemDetail>
      <DomainReservationStatus DomainReservationStatusType="Reserved"/>
    </DomainLinItemDetail>
  </LinItemDetail>
  <Price>17.0</Price>
</LinItem>
<LinItem>
  <Status>
    <StatusCode>-2320</StatusCode>
    <Description>Invalid line item, validation error.</Description>
    <ValidationError>

<PathName>/OrderRequest/Body/Product/CreateRegistration/DomainName[1]</PathName>
    <StatusCode>-2312</StatusCode>
    <Description>Invalid domain</Description>
    </ValidationError>
  </Status>
  <ProductType>Registration</ProductType>
  <DomainName>error!example.net</DomainName>
  <Price>0.0</Price>
</LinItem>
<LinItem>
  <LinItemID>3</LinItemID>
  <Status>
    <StatusCode>7004</StatusCode>
    <Description>Line item submitted successfully</Description>
  </Status>
  <ProductType>Registration</ProductType>
```

```

    <ProductID>WN.D.174989434</ProductID>
    <DomainName>errorexample.org</DomainName>
    <LinItemDetail>
      <DomainLinItemDetail>
        <DomainReservationStatus DomainReservationStatusType="Reserved"/>
      </DomainLinItemDetail>
    </LinItemDetail>
    <Price>17.0</Price>
  </LinItem>
  <NumberOfRecordsReturned>5</NumberOfRecordsReturned>
  <NumberOfRecordsAvailable>5</NumberOfRecordsAvailable>
</Order>
<NumberOfRecordsReturned>1</NumberOfRecordsReturned>
<NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
</Body>
</OrderResponse>

```

REQUEST:

```

<?xml version="1.0" encoding="UTF-8"?>
<OrderRequest>
  <RequestHeader>
    <VERSION_5_0></VERSION_5_0>
    <Authentication>
      <PartnerID>123</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <AuthorizedAgent SystemUserType="customer">
      <SystemUser>
        <UserID>789</UserID>
        <Password>customer</Password>
      </SystemUser>
    </AuthorizedAgent>
    <CustomerID>789</CustomerID>
    <Product>
      <CreateRegistration>
        <DomainName>errorexample.com</DomainName>
        <DomainName>error!example.net</DomainName>
        <DomainName>errorexample.org</DomainName>
        <PurchasePeriod>11</PurchasePeriod>
        <Hosts>
          <Host1>
            <Host>
              <HostName>ns1.errorexample.net</HostName>
              <HostIP1>1.2.3.4</HostIP1>
            </Host>
          </Host1>
          <Host2>
            <Host>
              <HostName>ns2.errorexample.org</HostName>
              <HostIP1>2.3.4</HostIP1>
            </Host>
          </Host2>
        </Hosts>
      </CreateRegistration>
    </Product>
  </Body>
</OrderRequest>

```

Invalid domain name.

Invalid Purchase
Period.

Invalid IP address.

```

</Product>
</Body>
</OrderRequest>

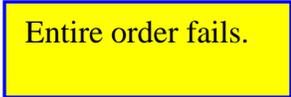
```

RESPONSE:

```

<?xml version="1.0"?>
<!DOCTYPE OrderResponse SYSTEM "OrderResponse.dtd">
<OrderResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25501805</ReferenceNumber>
    <Timestamp>2006-08-24T15:50:12EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Order>
      <ReferenceNumber>25501805</ReferenceNumber>
      <Status>
        <StatusCode>-7016</StatusCode>
        <Description>Order not Submitted</Description>
        <ValidationError>
          <PathName>/OrderRequest/Body/Product/CreateRegistration/PurchasePeriod</PathName>
          <StatusCode>-7021</StatusCode>
          <Description>Purchase period is invalid</Description>
        </ValidationError>
      </Status>
      <OrderAction>Create</OrderAction>
      <Customer>
        <UserID>456</UserID>
      </Customer>
      <Price>0.0</Price>
      <NumberOfRecordsReturned>0</NumberOfRecordsReturned>
      <NumberOfRecordsAvailable>0</NumberOfRecordsAvailable>
    </Order>
    <NumberOfRecordsReturned>1</NumberOfRecordsReturned>
    <NumberOfRecordsAvailable>1</NumberOfRecordsAvailable>
  </Body>
</OrderResponse>

```


4.3 Data Errors

If the XML is well formed and the data passes all the validation rules, there are still reasons the Partner Protocol Server may not accept a request. These reasons include, but are not limited to:

- The UserID and passwords do not match for either the Partner or the Customer;
- The product that is being modified does not exist; or
- The domain name is already registered.

In these cases the request will fail, and the response will contain the reason. The following example highlights a security error.

REQUEST:

```

<?xml version="1.0" encoding="UTF-8"?>
<UserLookupRequest>
  <RequestHeader>
    <VERSION_4_6 />
    <Authentication>
      <PartnerID>123</PartnerID>

```

```

    <PartnerPassword>partner</PartnerPassword>
  </Authentication>
</RequestHeader>
<Body>
  <AuthorizedAgent SystemUserType="customer">
    <SystemUser>
      <UserID>789</UserID>
      <Password>BADpassword</Password>
    </SystemUser>
  </AuthorizedAgent>
  <CustomerID>789</CustomerID>
  <LookupUser>
    <UserID>789</UserID>
  </LookupUser>
</Body>
</UserLookupRequest>
RESPONSE:
<?xml version="1.0"?>
<!DOCTYPE UserLookupResponse SYSTEM "UserLookupResponse.dtd">
<UserLookupResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>25501898</ReferenceNumber>
    <Timestamp>2006-08-24T15:50:36EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <Status>
      <StatusCode>-2002</StatusCode>
      <Description>User not authenticated</Description>
    </Status>
  </Body>
</UserLookupResponse>

```

The password is wrong.

4.4 XML Status Codes

XML Status Codes are contained in Appendix B at the end of this document. These codes are presented in two separate tables that provide the following information to the user:

- **Code** – This code uniquely identifies the type of code as either a status or an error. It can be used as a reference for the Status Code and Error Code tables in Appendix B. A status code has a positive value and provides additional information regarding the status of a request. An error code has a negative value and provides additional information regarding the particular error(s) in the request.
- **XML Description**– This is the textual response that will be displayed.

5.0 USER TROUBLESHOOTING

This section describes simple troubleshooting procedures that can be used when a Partner has trouble connecting to the Partner Protocol. This section is by no means exhaustive, and should be used as a starting place for a Partner's troubleshooting procedures.

The examples cited in this section are specific to running from a Unix platform, but the commands are available from all platforms. The specific syntax might be slightly different, and the specific response from a command might be slightly different, but the overall sequence of steps should be the same regardless of the platform.

5.1 Connectivity

In situations where basic connectivity is in question, perform the following steps:

- Ping the Partner Protocol server,
- Telnet to the Partner Protocol port, and
- Submit a valid request to the Partner Protocol port.

It is important to perform each of these steps from the machine that is thought to have connectivity trouble. Different machines in a network can have different connectivity to the network, so performing these steps on a different machine will only confirm that the particular machine tested has the connectivity. Also, the Partner Protocol server may be configured to only accept incoming connections from a known list of IP addresses. If any of the following tests fail, please contact a Technical Support Specialist.

5.1.1 Ping

To perform a ping, simply type `ping <Partner ProtocolServerName>`, where *Partner ProtocolServerName* is the name of the machine that supports the HTTPS proxy (see Figure 1).

Test Environment Example – Ping

```
ping partners.ote.networksolutions.com
```

Test Environment Expected Response

```
partners.ote.networksolutions.com is alive
```

This response indicates that machine from which this command was executed can connect to the Partner Protocol server. It does not mean that anything is necessarily running on the correct port.

Note: To ping the production environment substitute “*partners.networksolutions.com*” for “*partners.ote.networksolutions.com*”.

5.1.2 Telnet

To confirm that a process is listening on a given port, it is necessary to telnet to the actual port. In Partner Protocol, two ports are used, 8010 and 8020. It is necessary to telnet to both of these ports. Partner Protocol uses HTTPS (i.e. HTTP over SSL). Using telnet, which does not support SSL, a Partner can determine if the HTTPS proxy is listening on the two ports (8010 and 8020).

To telnet to a specific port, simply do:

```
telnet <serverName> <portNumber>.
```

Test Environment Example – Telnet

```
telnet partners.ote.networksolutions.com 8010
```

Test Environment Expected Response

```
Trying xxx.xxx.xxx.xxx...
Connected to partners.ote.networksolutions.com.
Escape character is '^]'.
Any text followed by a carriage return
Connection closed by foreign host.
```

The key here is to look for:

“Connected to partners.ote.networksolutions.com”.

This indicates that a connection was established to the requested port. Because the connection is not an SSL connection, any data sent via this connection (triggered by a carriage return in telnet) will cause the server-socket to close the connection, as indicated by “Connection closed by foreign host.”

If there is no process listening on the requested port, the connection is not accepted.

Note: The example above is specific to a Unix Telnet client. If you use a Windows telnet client, the response “Connected to partners.ote.networksolutions.com” does not appear. To determine if the Windows telnet client has connected, click on the “connect” menu. Telnet successfully established a connection if the “Disconnect” option is available.

Any responses other than the two mentioned above should be considered a failure.

Note: To test the production environment substitute “*partners.networksolutions.com*” for “*partners.ote.networksolutions.com*”.

5.1.3 Submit Request

The final step in verifying proper connectivity is to submit a valid request to the desired port. Since Partner Protocol uses standard HTTP over a SSL connection, the reader will need a telnet tool that supports using SSL connections. One such freeware package is OpenSSL. The tool itself is called `s_client` (in the Unix variant). The specific HTTP request was documented in section 1.2. In this section, the minimum amount of information that is required will be used, to minimize the amount of typing that is required. The request with the least required fields is the AvailableRequest.

Example – Submit Request

```
s_client -connect partners.ote.networksolutions.com:8010 -quiet
```

```
POST /invoke/vpp/AvailabilityService HTTP/1.0
```

```
Content-Type: text/xml
```

```
Content-length: 300
```

```
<?xml version="1.0" encoding="UTF-8"?>
<AvailableRequest>
  <RequestHeader>
    <VERSION_4_6 />
    <Authentication>
      <PartnerID>1234567</PartnerID>
      <PartnerPassword>partner</PartnerPassword>
    </Authentication>
  </RequestHeader>
  <Body>
    <VerifyDomainName>example.com</VerifyDomainName>
  </Body>
</AvailableRequest>
```

Expected Response

```
HTTP/1.1 200 OK
```

```
Content-Type: text/xml; charset=UTF8
```

```
Connection: Close
```

```
Content-Length: 448
```

```
<?xml version="1.0"?>
<!DOCTYPE AvailableResponse SYSTEM "AvailableResponse.dtd">
<AvailableResponse>
  <ResponseHeader>
    <VERSION_5_0></VERSION_5_0>
    <ReferenceNumber>002</ReferenceNumber>
    <Timestamp>2006-08-24T15:51:45EDT</Timestamp>
  </ResponseHeader>
  <Body>
    <NotAvailable>
      <DomainName>example.com</DomainName>
      <RespondingService ServiceName="Registry"/>
    </NotAvailable>
  </Body>
</AvailableResponse>
```

There are several points to note in this example.

The line beginning with “POST” is the first line that needs to be entered at the command prompt. The string that follows POST is the specific service that is being invoked. This is specific to the HTTP server that one is connecting to. The

required string for the Partner Protocol server is in the Partner Protocol configuration file. In general, this string is case sensitive. The "HTTP/1.0" at the end of this line is required by the HTTP.

"Content-length" is the final line in the HTTP header. This line specifies the number of characters (bytes) in this HTTP post. Unfortunately, the size of the post is required before any information is actually typed. An easy way to determine this appropriate size is to save the Partner Protocol-request in a text file, and then look at the size of the file. One can then also cut-and-paste from this file to submit the actual request. Content-length must then be followed by two carriage returns, and then the request text.

When submitting a request in this manner, it is actually safer to over-estimate the content-length, rather than to make content-length too small. Content-length is the number of bytes that will be read off the stream before the HTTP server starts to process the request. If this length is too small, even by one byte, most likely a malformed XML request will be interpreted. If content-length is too large, one will have to enter additional carriage returns to "pad" the request. Once "content-length" bytes have been entered onto the stream, the HTTP server will process the request.

If the expected response is received from this request, then the Partner Protocol system is working as expected. If the Partner's production system is not connecting as expected, the problem is then most likely a configuration issue specific to that system. This test confirms that the Partner's site can connect, submit, and receive proper Partner Protocol commands.

6.0 HELP DESK INFORMATION

Network Solutions assigns a Technical Support Specialist (TSS) to each new Partner. The TSS assignment is part of the new Partner integration procedure. The TSS helps the Partner integrate the Partner Protocol into their infrastructure and helps resolve the Partner's technical issues.

Appendix A Document Type Definitions (DTD)s

A.1 AvailableRequest DTD

```

<!-- ===== -->
<!-- ===== Domain Availability Request Message and Body Declaration. ===== -->
<!-- ===== -->
<!ELEMENT AvailableRequest (RequestHeader, Body)>
<!ELEMENT Body (VerifyDomainName+ | VerifyDomains | GenerateDomains | VerifyDomainsAdvanced |
GenerateDomainsAdvanced | VerifyDomainsAll | GenerateDomainsAll |
IsDomainCandidateForInboundTransfer+ | IsDomainCandidateForInternalTransfer+)>
<!-- ===== -->
<!-- ===== Domain Availability Action Declarations. ===== -->
<!-- ===== -->
<!ELEMENT VerifyDomainName (#PCDATA)>
<!-- VerifyDomainName - Check availability of a domain name -->
<!ELEMENT VerifyDomains (#PCDATA)>
<!-- VerifyDomains - Check availability of a second level domain name in -->
<!-- the com, net, org, info, biz us.com, mobi, tv top level domains. -->
<!ELEMENT GenerateDomains (#PCDATA)>
<!-- GenerateDomains - Check availability of a second level domain name in -->
<!-- the com, net, org, info, biz us.com, mobi, tv top level domains and
generated additional domain names under the com TLD. -->
<!ELEMENT VerifyDomainsAdvanced (Domain, TLD+)>
<!-- VerifyDomainsAdvanced - Check availability of a second level domain name in -->
<!-- the com, net, org, info, biz us.com, mobi, tv top level domains and any additional TLDs
specified. -->
<!ELEMENT GenerateDomainsAdvanced (Domain, TLD+, GetMisspelledListOnly?)>
<!-- GenerateDomainsAdvanced - Generate alternative domain names on
the com as well as TLDs passed in and check their availability.
If GetMisspelledListOnly is True, return the misspelled list only.-->
<!ELEMENT GenerateDomainsAll (#PCDATA)>
<!-- GenerateDomainsAll -Generate alternative domain names on all supported TLDs
and check their availability.-->
<!ELEMENT VerifyDomainsAll (#PCDATA)>
<!-- VerifyDomainsAll - Check availability of a second level domain name in -->
<!-- all supported top level domains. -->
<!ELEMENT IsDomainCandidateForInboundTransfer (#PCDATA)>
<!-- IsDomainCandidateForInboundTransfer - Assess whether a domain might be a -->
<!-- candidate for Inbound Transfer. -->
<!ELEMENT IsDomainCandidateForInternalTransfer (#PCDATA)>
<!-- IsDomainCandidateForInternalTransfer - Assess whether a domain might be a -->
<!-- candidate for an Internal Customer to Customer Transfer. -->
<!-- ===== -->
<!-- ===== Base element declarations ===== -->
<!-- ===== -->
<!-- Domain name or word without a tld -->
<!ELEMENT Domain (#PCDATA)>
<!-- TLD, Top Level Domain name -->
<!ELEMENT TLD (#PCDATA)>
<!ELEMENT GetMisspelledListOnly EMPTY>
<!ATTLIST GetMisspelledListOnly
    Selection (True | False) #REQUIRED
>
<!-- ===== -->
<!-- ===== Inclusion of externally defined elements. ===== -->
<!-- ===== -->
<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;
<!-- sccsID "%TC-INFO%" -->

```

A.2 AvailableResponse DTD

```

<!-- ===== -->
<!-- ===== Domain Availability Response Message and Body Declaration. ===== -->
<!-- ===== -->
<!ELEMENT AvailableResponse (
    ResponseHeader,
    Body ) >
<!ELEMENT Body (
    Available?,
    NotAvailable?,
    Premium?,

```

```

    Invalid?,
    Failed?,
    MaybeTransferrable?,
    NotTransferrable? ) >

<!-- Domains that are available -->
<!ELEMENT Available (DomainName, RespondingService)* >

<!-- Domains that are not available -->
<!ELEMENT NotAvailable (DomainName, RespondingService)* >

<!-- Premium domains -->
<!ELEMENT Premium (DomainName, RespondingService)* >

<!-- Domains that failed due to temporary unavailability of the target registry -->
<!ELEMENT Failed (DomainName)* >

<!-- Domains that may be transferrable -->
<!ELEMENT MaybeTransferrable (DomainName, TransferDetail)* >

<!-- Domains that are not transferrable -->
<!ELEMENT NotTransferrable (DomainName, NotTransferrableReasons)* >

<!-- Domains with invalid syntax -->
<!ELEMENT Invalid (DomainName)* >

<!ELEMENT DomainName (#PCDATA) >
<!ATTLIST DomainName price CDATA #IMPLIED encoded CDATA "false" >

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

<!-- The service that provided the name availability response -->
<!ELEMENT RespondingService EMPTY >
<!ATTLIST RespondingService ServiceName (
    Registry |
    Whois |
    ZoneFiles |
    LocalCache |
    NotIndicated ) #REQUIRED >

<!-- The details about the transfer candidate -->
<!ELEMENT TransferDetail (InternalTransferDetail?) >
<!ATTLIST TransferDetail TransferType (Inbound | Internal) #REQUIRED >

<!-- The details about the internal transfer candidate -->
<!ELEMENT InternalTransferDetail (InternalTransferCategory, ProductType) >

<!-- The internal transfer candidate category -->
<!ELEMENT InternalTransferCategory EMPTY >
<!ATTLIST InternalTransferCategory InternalTransferCategorySelection (
    RetailToWholesale |
    WholesaleToWholesaleInPartner |
    WholesaleToWholesaleCrossPartner) #REQUIRED >

<!-- The reason(s) why a domain is not a candidate for transfer -->
<!ELEMENT NotTransferrableReasons (NotTransferrableReason+) >
<!ELEMENT NotTransferrableReason (StatusCode, Description) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.3 Common Declarations DTD

```

<!-- ===== -->
<!-- ==== Common element declarations ===== -->
<!-- ===== -->
<!-- ===== -->
<!-- ===== User and Customer Related Common element declarations ===== -->
<!-- ===== -->
<ELEMENT AuthorizedAgent (SystemUser)>
<ATTLIST AuthorizedAgent
    SystemUserType (customer | partner) #REQUIRED
>
<ELEMENT SystemUser ((UserID | NicHandle), Password)>
<ELEMENT UserID (#PCDATA)>
<ELEMENT CustomerID (#PCDATA)>
<ELEMENT NicHandle (#PCDATA)>
<ELEMENT Password (#PCDATA)>
<ELEMENT OrgName (#PCDATA)>
<ELEMENT FirstName (#PCDATA)>
<ELEMENT MiddleName (#PCDATA)>
<ELEMENT LastName (#PCDATA)>
<ELEMENT LoginName (#PCDATA)>
<ELEMENT CompanyName (#PCDATA)>
<ELEMENT Phone (#PCDATA)>
<ELEMENT Fax (#PCDATA)>
<ELEMENT Email (#PCDATA)>
<ELEMENT AuthQuestion (#PCDATA)>
<ELEMENT AuthAnswer (#PCDATA)>
<ELEMENT Address (AddressLine1, AddressLine2?, AddressLine3?, AddressLine4?, AddressLine5?,
City?, State?, PostalCode?, CountryCode)>
<ELEMENT AddressLine1 (#PCDATA)>
<ELEMENT AddressLine2 (#PCDATA)>
<ELEMENT AddressLine3 (#PCDATA)>
<ELEMENT AddressLine4 (#PCDATA)>
<ELEMENT AddressLine5 (#PCDATA)>
<ELEMENT City (#PCDATA)>
<ELEMENT State (#PCDATA)>
<ELEMENT PostalCode (#PCDATA)>
<ELEMENT CountryCode (#PCDATA)>
<ELEMENT POBox (#PCDATA)>
<!-- ===== -->
<!-- ==== Product Related Common element declarations ===== -->
<!-- ===== -->
<ELEMENT ProductID (#PCDATA)>
<!-- ===== ServiceInstID is an ID for a specific instance of product (i.e. an SSL Certificate or
Seal) ===== -->
<ELEMENT ServiceInstID (#PCDATA)>
<ELEMENT PurchasePeriod (#PCDATA)>
<ELEMENT BulkWhoisOptOut (#PCDATA)>
<ELEMENT ProductStatus (#PCDATA)>
<ELEMENT Billing (DiscountCode?, AutoRenew?, IsPricingRequest?)>
<ELEMENT DiscountCode (#PCDATA)>
<ELEMENT AutoRenew (#PCDATA)>
<ELEMENT TechContact (UserID | NicHandle)>
<ELEMENT AdminContact (UserID | NicHandle)>
<ELEMENT BillContact (UserID | NicHandle)>
<ELEMENT LegalContact (FirstName, MiddleName?, LastName, Address?, Phone?, Fax?, Email?)>
<ELEMENT LegalContactFirstName (#PCDATA)>
<ELEMENT LegalContactMiddleName (#PCDATA)>
<ELEMENT LegalContactLastName (#PCDATA)>
<!-- Name of the product instance. e.g. domain name, host name, email name -->
<ELEMENT Name (#PCDATA)>
<!-- The following two elements are used to refer to customers as part of transfer requests -->
<ELEMENT FromCustomer (UserID)>
<ELEMENT ToCustomer (UserID)>
<!-- Name of the product type. e.g. Registration, Host, Email -->
<ELEMENT ProductType (#PCDATA)>
<!-- If DomainProtect is set to True on a domain then outbound transfer is not allowed -->
<ELEMENT DomainProtect EMPTY>
<ATTLIST DomainProtect
    Selection (True | False) #REQUIRED

```

```

>
<!ELEMENT LegalLock EMPTY>
<!ATTLIST LegalLock
  Selection (True | False) #REQUIRED
>
<!ELEMENT URL (#PCDATA)>
<!-- Indicates that the request should just be priced and not submitted. -->
<!ELEMENT IsPricingRequest EMPTY>
<!-- Subscriber's agreement -->
<!ELEMENT SubscribersAgreement EMPTY>
<!ATTLIST SubscribersAgreement
  Selection (True | False) #REQUIRED
>
<!-- ===== -->
<!-- ==== TLD Specific Customer Data Element Declarations ===== -->
<!-- ===== -->
<!ELEMENT SupplementalRegistryData (DotUSSupplementalData?, DotDESupplementalData?,
DotAUSupplementalData?, DotEUSupplementalData?)>
<!-- Classification of a domain names registrant, and the possible usage of the domain. -->
<!-- Used for: .us -->
<!ELEMENT DotUSSupplementalData EMPTY>
<!ATTLIST DotUSSupplementalData
  CustomerType (PersonUSCitizen | PersonPermanentUSResident | OrganizationIncorporatedInUS |
OrganizationWithActiveUSPresence | OrganizationWithUSPresence) #REQUIRED
  AnticipatedUsage (ForProfit | NotForProfit | Personal | Educational | Governmental) #REQUIRED
>
<!ELEMENT DotDESupplementalData (UseDefaultRegistrantAddress)>
<!ELEMENT DotAUSupplementalData (LegalIdentificationNumber)>
<!-- Legal registration or identification number of a business or -->
<!-- legal identification number of an individual. -->
<!ELEMENT LegalIdentificationNumber (#PCDATA)>
<!-- Indicates that a default, in-country registrant address should be used. -->
<!ELEMENT UseDefaultRegistrantAddress EMPTY>
<!ATTLIST UseDefaultRegistrantAddress
  Selection (True | False) #REQUIRED
>
<!-- EU NEXUS elements -->
<!ELEMENT DotEUSupplementalData EMPTY>
<!ATTLIST DotEUSupplementalData
  CustomerType (RegisteredOfficeInEU | OrganizationIncorporatedInEU | ResidentInEU) #REQUIRED
  NaturalPerson CDATA #FIXED "True"
>
<!-- ===== -->
<!-- ==== Request and Order Related Common element declarations ===== -->
<!-- ===== -->
<!ELEMENT Status (StatusCode, Description, ValidationError*)>
<!ELEMENT StatusCode (#PCDATA)>
<!ELEMENT Comments (#PCDATA)>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT ValidationError (PathName, StatusCode, Description)>
<!ELEMENT PathName (#PCDATA)>
<!ELEMENT ReferenceNumber (#PCDATA)>
<!ELEMENT OrderID (#PCDATA)>
<!ELEMENT LineItemID (#PCDATA)>
<!-- Number of records returned in a response. This is a signed 64-bit integer -->
<!ELEMENT NumberOfRecordsReturned (#PCDATA)>
<!-- Total number of records in the system. If the total number of records -->
<!-- returned is less than the number of records available, narrow the -->
<!-- date parameters until values are equal. This is a signed 64-bit integer -->
<!ELEMENT NumberOfRecordsAvailable (#PCDATA)>
<!-- ===== -->
<!-- ==== General Common element declarations ===== -->
<!-- ===== -->
<!-- Begin and end dates for a date range -->
<!ELEMENT DateRange (BeginDate, EndDate)>
<!ELEMENT BeginDate (Date | LongDate)>
<!ELEMENT EndDate (Date | LongDate)>
<!-- Date format is Month, Day, Year, minutes, -->
<!-- seconds, and milliseconds in the format of -->
<!-- Month 01 through 12, Day 01 through 31, -->
<!-- Year 1970 through Present year as a four digit year, -->

```



```

<!ELEMENT Host5 ( Host ) >
<!ELEMENT Host6 ( Host ) >
<!ELEMENT Host7 ( Host ) >
<!ELEMENT Host8 ( Host ) >
<!ELEMENT Host9 ( Host ) >
<!ELEMENT Host10 ( Host ) >
<!ELEMENT Host11 ( Host ) >
<!ELEMENT Host12 ( Host ) >
<!ELEMENT Host13 ( Host ) >

<!ELEMENT Host ( ( HostID | HostName ), HostCoordinator?,
    HostIP1?, HostIP2?, HostIP3?, HostIP4?, HostIP5? ) >

<!ELEMENT NewHost ( HostName, HostCoordinator?,
    HostIP1, HostIP2?, HostIP3?, HostIP4?, HostIP5? ) >

<!ELEMENT HostID (#PCDATA) >
<!ELEMENT HostName (#PCDATA) >
<!ELEMENT HostCoordinator ( UserID | NicHandle ) >
<!ELEMENT HostIP1 (#PCDATA) >
<!ELEMENT HostIP2 (#PCDATA) >
<!ELEMENT HostIP3 (#PCDATA) >
<!ELEMENT HostIP4 (#PCDATA) >
<!ELEMENT HostIP5 (#PCDATA) >
<!ELEMENT Action (#PCDATA)>
<!-- sccsID "%TC-INFO%" -->

```

A.5 *DomainProduct DTD*

```

<!-- ===== -->
<!-- ===== Domain Registration product actions ===== -->
<!-- ===== -->

<!ELEMENT CreateRegistration
(DomainName+,
PurchasePeriod,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
Hosts?,
SupplementalRegistryData?,
PrivateRegistrationDomains*)>

<!ELEMENT ModifyRegistration
(DomainName+,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
Hosts?)>

<!ELEMENT ConvertToRegistration

```

```
(DomainName+,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
Hosts?)>

<!ELEMENT DeleteRegistration (DomainName+)>

<!ELEMENT RenewRegistration (DomainName+, PurchasePeriod)>

<!ELEMENT TransferInternalToRegistration
(DomainName+,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
Hosts?,
SupplementalRegistryData?)>

<!ELEMENT InboundTransferToRegistration
(ToCustomer,
DomainName+,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
Hosts?,
AuthCodeInfo*,
SupplementalRegistryData?,
PrivateRegistrationDomains*)>

<!-- ==== Domain Parking product actions ===== -->
<!ELEMENT CreateParking
(DomainName+,
PurchasePeriod,
TechContact?,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
SupplementalRegistryData?,
PrivateRegistrationDomains*,
WebForwardingProduct*,
ADNS*)>
<!ELEMENT ModifyParking
(DomainName+,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
```

```
ModifiedParking*)>

<!ELEMENT ConvertToParking (DomainName+, AdminContact?, BillContact?, BulkWhoisOptOut?, ADNS*)>

<!ELEMENT DeleteParking (DomainName+)>

<!ELEMENT RenewParking (DomainName+, PurchasePeriod)>

<!ELEMENT TransferInternalToParking
(DomainName+,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
ADNS*,
SupplementalRegistryData?)>

<!ELEMENT InboundTransferToParking
(ToCustomer,
DomainName+,
AdminContact?,
BillContact?,
BulkWhoisOptOut?,
DomainProtect?,
AuthCodeInfo*,
SupplementalRegistryData?,
PrivateRegistrationDomains*,
WebForwardingProduct*)>

<!-- ==== PreRegistration product actions ===== -->
<!ELEMENT PreRegister
(Phase1?,
Phase2?,
OpenRegistration?,
LanguageSelection?,
TechContact?,
AdminContact?,
BillContact?,
DotEUSupplementalData,
EUExhibit)>

<!ELEMENT Phase1
(PublicBodyDomains?,
TradeMarkNationalDomains?,
TradeMarkComIntlDomains?,
GeoDooDomains?)>

<!ELEMENT Phase2 (CompIDDomains?, UnRegTMDomains?, TitlesArtDomains?, OtherRightsDomains?)>
<!-- ==== PreRegistration categories ===== -->
<!ELEMENT EUDomain (DomainName, PriorRightOnName, PriorRightCountryCode)>
<!ELEMENT PublicBodyDomains (EUDomain+)>
<!ELEMENT TradeMarkNationalDomains (EUDomain+)>
<!ELEMENT TradeMarkComIntlDomains (EUDomain+)>
```

```
<!ELEMENT GeoDooDomains (EUDomain+)>
<!ELEMENT CompIDDomains (EUDomain+)>
<!ELEMENT UnRegTMDomains (EUDomain+)>
<!ELEMENT TitlesArtDomains (EUDomain+)>
<!ELEMENT OtherRightsDomains (EUDomain+)>
<!ELEMENT OpenRegistration (DomainName+)>
<!ELEMENT EUExhibit EMPTY>
<!ATTLIST EUExhibit Selection (True | False) #REQUIRED>
<!ELEMENT PriorRightOnName (#PCDATA)>
<!ELEMENT PriorRightCountryCode (#PCDATA)>

<!-- ===== Language Selection must be 1 of the
following:Czech,Danish,Dutch,English,Estonian,Finnish,French,German == -->
<!-- =====
Greek,Hungarian,Italian,Latvian,Lithuanian,Maltese,Polish,Portuguese,Slovak,Slovenian,Spanish,Swe
dish == -->
<!ELEMENT LanguageSelection (#PCDATA)>

<!-- ===== Domain product actions applicable to any type of domain product == -->
<!-- Delete Parking, or Registration, or SelfManaged product. -->
<!ELEMENT CascadeDeleteDomain (DomainName+)>
<!-- Renew Parking, or Registration, or SelfManaged. -->
<!ELEMENT CascadeRenewDomain (DomainName+, PurchasePeriod)>
<!-- Renew domains only. -->
<!ELEMENT RenewDomainsOnly (DomainName+, PurchasePeriod)>

<!-- ===== -->
<!-- ===== Product Lookup queries and Element returned in lookup reply. === -->
<!-- ===== -->
<!ELEMENT LookupDomain (ProductID+ | DomainName+)>
<!ELEMENT Domain
(ProductID,
ProductType,
DomainNameProperties,
ProductStatus,
DomainProtect,
LegalLock,
CustomerID,
AdminContact?,
TechContact?,
BillContact?,
BulkWhoisOptOut?,
AutoRenew,
PurchaseDate,
ExpirationDate,
A*,
CNAME*,
MX*,
NicHandle?,
HostData*,
PrivateRegistrationActive?,
WebForwardingProduct?)>

<!-- ===== -->
```

```

<!-- ==== Domain AuthCode request Element ===== -->
<!-- === This request will result in the domain(s) AuthCode being sent == -->
<!-- === via email to the domains Primary contact ===== -->
<!ELEMENT RetrieveAuthCode (DomainName+)>
<!-- ===== -->
<!-- ===== Base element declarations ===== -->
<!-- ===== -->
<!-- For backward compatibility DomainNameProperties is defined here -->
<!ELEMENT DomainNameProperties (#PCDATA)>
<!ATTLIST DomainNameProperties encoded CDATA "false">
<!ELEMENT ResourceRecords (Action, A*, CNAME*, MX*)>
<!ELEMENT DomainName (#PCDATA)>
<!ELEMENT A (RecordName, IPAddress)>
<!ELEMENT CNAME (RecordName, CanonicalName)>
<!ELEMENT MX (RecordName, Preference, Exchange)>
<!ELEMENT RecordName (#PCDATA)>
<!ELEMENT IPAddress (#PCDATA)>
<!ELEMENT CanonicalName (#PCDATA)>
<!ELEMENT Preference (#PCDATA)>
<!ELEMENT Exchange (#PCDATA)>
<!ELEMENT Hosts (Host1, (Host2, (Host3, (Host4, (Host5, (Host6, (Host7, (Host8, (Host9, (Host10,
(Host11, (Host12, (Host13)?)?)?)?)?)?)?)?)?)?>
<!ELEMENT Host1 (Host)>
<!ELEMENT Host2 (Host)>
<!ELEMENT Host3 (Host)>
<!ELEMENT Host4 (Host)>
<!ELEMENT Host5 (Host)>
<!ELEMENT Host6 (Host)>
<!ELEMENT Host7 (Host)>
<!ELEMENT Host8 (Host)>
<!ELEMENT Host9 (Host)>
<!ELEMENT Host10 (Host)>
<!ELEMENT Host11 (Host)>
<!ELEMENT Host12 (Host)>
<!ELEMENT Host13 (Host)>
<!ELEMENT PrivateRegistrationDomains (#PCDATA)>
<!ELEMENT PrivateRegistrationActive (#PCDATA)>
<!-- ===== 2Q 2005 Release ===== -->
<!ELEMENT AuthCodeInfo (DomainName, AuthCode)>
<!ELEMENT AuthCode (#PCDATA)>
<!ELEMENT ModifiedParking (DomainName, (ResourceRecords* | UnderConstructionPage))>
<!ELEMENT Action (#PCDATA)>
<!ELEMENT ADNS (DomainName, ResourceRecords?)>
<!ELEMENT UnderConstructionPage (#PCDATA)>
<!-- ===== -->
<!-- ===== Inclusion of externally defined elements. ===== -->
<!ENTITY % WebForwardingProduct SYSTEM "WebForwardingProduct.dtd">
%WebForwardingProduct;
<!-- ===== -->
<!--none -->
<!-- sccsID "%TC-INFO%" -->

```

A.6 *EmailForwardingLookupRequest DTD*

```

<!-- ===== -->
<!-- ==== This DTD remains for backward compatibility. ===== -->
<!-- ===== -->

<!ELEMENT EmailForwardingLookupRequest ( RequestHeader, Body ) >

<!ELEMENT Body ( AuthorizedAgent, LookupEmailForwarding ) >

<!-- ===== -->
<!-- ==== Product Lookup queries and Element returned in lookup reply. === -->
<!-- ===== -->

<!ELEMENT LookupEmailForwarding (EmailAddress | ProductID ) >
<!ELEMENT EmailAddress (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.7 *EmailForwardingLookupResponse DTD*

```

<!-- ===== -->
<!-- ==== This DTD remains for backward compatibility. ===== -->
<!-- ===== -->

<!ELEMENT EmailForwardingLookupResponse ( ResponseHeader, Body ) >

<!ELEMENT Body ( Status, EmailForwarding? ) >

<!ELEMENT EmailForwarding (
    ProductID,
    ProductStatus,
    CustomerID,
    PurchaseDate,
    ExpirationDate,
    EmailAddress,
    ForwardingEmailAddress ) >

<!ELEMENT EmailAddress (#PCDATA)>
<!ELEMENT ForwardingEmailAddress (#PCDATA)>
<!ELEMENT PurchaseDate (#PCDATA)>
<!ELEMENT ExpirationDate (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->

```

```
<!-- ===== -->
```

```
<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;
```

```
<!-- sccsID "%TC-INFO%" -->
```

A.8 *EmailForwardingRequest DTD*

```
<!-- ===== -->
```

```
<!-- ===== This DTD remains for backward compatibility. ===== -->
```

```
<!-- ===== -->
```

```
<!ELEMENT EmailForwardingRequest (
    RequestHeader,
    Body ) >
```

```
<!ELEMENT Body (
    AuthorizedAgent,
    CustomerID,
    Billing?,
    Product ) >
```

```
<!ELEMENT Product (
    CreateEmailForwardingAndDomain |
    ModifyEmailForwarding |
    DeleteEmailForwardingAndDomain |
    RenewEmailForwardingAndDomain ) >
```

```
<!ELEMENT CreateEmailForwardingAndDomain (
    CreateRegistration |
    CreateParking ) >
```

```
<!ELEMENT CreateRegistration (
    DomainName+,
    PurchasePeriod,
    TechContact?,
    AdminContact?,
    BillContact?,
    BulkWhoisOptOut?,
    DomainProtect?,
    Hosts?,
    ForwardingEmailAddress ) >
```

```
<!ELEMENT CreateParking (
    DomainName+,
    PurchasePeriod,
    TechContact?,
    AdminContact?,
    BillContact?,
    BulkWhoisOptOut?,
    DomainProtect?,
    ForwardingEmailAddress ) >
```

```

<!ELEMENT ModifyEmailForwarding (
    EmailForwardingProduct+,
    ForwardingEmailAddress ) >

<!ELEMENT EmailForwardingProduct (
    ( EmailAddress | ProductID ) ) >

<!ELEMENT EmailAddress (#PCDATA) >
<!ELEMENT ForwardingEmailAddress (#PCDATA) >

<!ELEMENT DeleteEmailForwardingAndDomain
    ( DomainName+ | EmailForwardingProduct+ ) >

<!ELEMENT RenewEmailForwardingAndDomain
    ( (DomainName+ | EmailForwardingProduct+),
    PurchasePeriod ) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;
<!ENTITY % DomainData SYSTEM "DomainData.dtd">
%DomainData;

<!-- sccsID "%TC-INFO%" -->

```

A.9 *ErrorResponse DTD*

```

<!-- ===== -->
<!-- ==== Error Response Declarations ===== -->
<!-- ===== -->

<!ELEMENT ErrorResponse (ResponseHeader, Body ) >
<!ELEMENT ResponseHeader (ReferenceNumber, Timestamp, Comments? ) >
<!ELEMENT Body (StatusCode, Description, OriginalData ) >

<!ELEMENT OriginalData ANY >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % CommonDeclarations SYSTEM "CommonDeclarations.dtd">
%CommonDeclarations;

<!-- sccsID "%TC-INFO%" -->

```

A.10 *HostProduct DTD*

```

<!-- ===== -->

```

```
<!-- ==== Product actions ===== -->
<!-- ===== -->

<!ELEMENT CreateHost (NewHost+)>
<!ELEMENT ModifyHost (Host+)>
<!ELEMENT DeleteHost ((HostID | HostName)+)>

<!-- ===== -->
<!-- ==== Product Lookup queries and Element returned in lookup reply. === -->
<!-- ===== -->

<!ELEMENT LookupHost (HostName+ | HostID | HostIP ) >

<!-- The ability to lookup a host using HostID and HostIP have been deprecated. -->
<!-- Please do not use these elements to lookup hosts. -->

<!ELEMENT HostData (
    HostID,
    HostName,
    HostCoordinator?,
    HostIP1?,
    HostIP2?,
    HostIP3?,
    HostIP4?,
    HostIP5? ) >

<!-- HostID and HostCoordinator elements have been deprecated. Please do not use these elements -
->

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

<!ELEMENT NewHost (
    HostName,
    HostCoordinator?,
    HostIP1,
    HostIP2?,
    HostIP3?,
    HostIP4?,
    HostIP5? ) >

<!ELEMENT Host (
    (HostID | HostName),
    HostCoordinator?,
    HostIP1?,
    HostIP2?,
    HostIP3?,
    HostIP4?,
    HostIP5? ) >

<!-- HostID and HostCoordinator elements have been deprecated. Please do not use these elements -
->
```

```

<!ELEMENT HostName (#PCDATA) >
<!ELEMENT HostIP1 (#PCDATA) >
<!ELEMENT HostIP2 (#PCDATA) >
<!ELEMENT HostIP3 (#PCDATA) >
<!ELEMENT HostIP4 (#PCDATA) >
<!ELEMENT HostIP5 (#PCDATA) >
<!ELEMENT HostIP (#PCDATA) >
<!ELEMENT HostID (#PCDATA) >
<!ELEMENT HostCoordinator (UserID | NicHandle) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!--none -->

<!-- sccsID "%TC-INFO%" -->

```

A.11 *LookupRequest DTD*

```

<!-- ===== -->
<!-- ==== Product Lookup Request Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT LookupRequest ( RequestHeader, Body ) >

<!ELEMENT Body (
  AuthorizedAgent,
  CustomerID?,
  (LookupDomain |
  LookupHost) ) >

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

<!-- For backward compatibility ExpirationDate and PurchaseDate are defined here -->
<!ELEMENT ExpirationDate (#PCDATA)>
<!ELEMENT PurchaseDate (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;
<!ENTITY % DomainProduct SYSTEM "DomainProduct.dtd">
%DomainProduct;
<!ENTITY % HostProduct SYSTEM "HostProduct.dtd">

```

```
%HostProduct;
```

```
<!-- sccsID "%TC-INFO%" -->
```

A.12 *LookupResponse DTD*

```
<!-- ===== -->
<!-- ==== Product Lookup Response Declaration. ===== -->
<!-- ===== -->

<!ELEMENT LookupResponse (ResponseHeader, Body ) >

<!-- ===== -->
<!-- ==== Products returned from the lookup request. ===== -->
<!-- ===== -->

<!ELEMENT Body (
  (Domain* |
   HostData*),
  Status ) >

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

<!-- For backward compatibility ExpirationDate and PurchaseDate are defined here -->
<!ELEMENT ExpirationDate (#PCDATA ) >
<!ELEMENT PurchaseDate (#PCDATA ) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;
<!ENTITY % DomainProduct SYSTEM "DomainProduct.dtd">
%DomainProduct;
<!ENTITY % HostProduct SYSTEM "HostProduct.dtd">
%HostProduct;

<!-- sccsID "%TC-INFO%" -->
```

A.13 *OrderRequest DTD*

```
<!-- ===== -->
<!-- ==== Product Order Request Message and Body Declaration. ===== -->
<!-- ===== -->
<!ELEMENT OrderRequest (RequestHeader, Body)>
<!ELEMENT Body (AuthorizedAgent, CustomerID, Billing?, Product)>
<!-- ===== -->
```

```

<!-- ===== Catalog of supported Product Actions. ===== -->
<!-- ===== -->
<ELEMENT Product (CreateRegistration | ModifyRegistration | ConvertToRegistration | DeleteRegistration |
RenewRegistration | TransferInternalToRegistration | InboundTransferToRegistration+ | CreateParking | DeleteParking |
RenewParking | ModifyParking | TransferInternalToParking | InboundTransferToParking+ | ConvertToParking |
CascadeDeleteDomain | CascadeRenewDomain | CreateHost | ModifyHost | DeleteHost | ModifyCustomerName |
CreatePrivateRegistration | RenewPrivateRegistration | DeletePrivateRegistration | ModifyPrivateRegistration |
RenewDomainsOnly | CreateWebForwarding | ModifyWebForwarding | DeleteWebForwarding | RenewWebForwarding |
PreRegister | RetrieveAuthCode | CreateSSL | RenewSSL | ReissueSSL | RevokeSSL | DownloadCertificate |
GetSealCode)>
<!-- ===== -->
<!-- ===== Base element declarations ===== -->
<!-- ===== -->
<!-- For backward compatibility ExpirationDate and PurchaseDate are defined here -->
<ELEMENT ExpirationDate (#PCDATA)>
<ELEMENT PurchaseDate (#PCDATA)>
<ELEMENT ModifyCustomerName ((FirstName, MiddleName?, LastName) | CompanyName)>
<!-- ===== -->
<!-- ===== Inclusion of externally defined elements. ===== -->
<!-- ===== -->
<ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;
<ENTITY % DomainProduct SYSTEM "DomainProduct.dtd">
%DomainProduct;
<ENTITY % HostProduct SYSTEM "HostProduct.dtd">
%HostProduct;
<ENTITY % PrivateRegistrationProduct SYSTEM "PrivateRegistrationProduct.dtd">
%PrivateRegistrationProduct;
<ENTITY % SSLProduct SYSTEM "SSLProduct.dtd">
%SSLProduct;
<!-- sccsID "%TC-INFO%" -->

```

A.14 OrderResponse DTD

```

<!-- ===== -->
<!-- ===== Product Order Response Message and Body Declaration. ===== -->
<!-- ===== -->
<ELEMENT OrderResponse (ResponseHeader, Body)>
<ELEMENT Body (Order*, Status?, NumberOfRecordsReturned?, NumberOfRecordsAvailable?)>
<!-- ===== -->
<!-- ===== Status of an Order and its contained LinelItems. ===== -->
<!-- ===== -->
<ELEMENT Order (ReferenceNumber, Status, OrderID?, OrderAction, Customer, Price?, TieredPrice?, LinelItem*,
OrderSubmissionDate?, NumberOfRecordsReturned?, NumberOfRecordsAvailable?)>
<ELEMENT LinelItem (LinelItemID?, Status, ProductType, ProductID?, DomainName?, AuthCode?, Name?,
LinelItemDetail?, Price?, SSLDetail?)>
<!-- ===== -->
<!-- ===== Base element declarations ===== -->
<!-- ===== -->
<!-- OrderAction examples, Create, Modify, etc. -->
<ELEMENT OrderAction (#PCDATA)>
<!-- The date the order was submitted. Date format: YYYY-MM-DDThh:mm:ssTZD -->
<ELEMENT OrderSubmissionDate (#PCDATA)>
<ELEMENT DomainName (#PCDATA)>
<ELEMENT AuthCode (#PCDATA)>
<ELEMENT Customer (UserID)>
<ELEMENT LinelItemDetail (DomainLinelItemDetail)>
<ELEMENT DomainLinelItemDetail (DomainReservationStatus)>
<ELEMENT DomainReservationStatus EMPTY>
<ATTLIST DomainReservationStatus
    DomainReservationStatusType (Reserved | Queued) #REQUIRED
>
<!-- The total price of a product line item or an order. -->
<ELEMENT Price (#PCDATA)>
<ELEMENT TieredPrice EMPTY>

```

```

<!ATTLIST TieredPrice
  Selection (True | False) #REQUIRED
>
<!ELEMENT SSLDetail (Certificate?, SealHeader?, SealBody?)>
<!ELEMENT Certificate (#PCDATA)>
<!ELEMENT SealHeader (#PCDATA)>
<!ELEMENT SealBody (#PCDATA)>
<!-- ===== -->
<!-- ===== Inclusion of externally defined elements. ===== -->
<!-- ===== -->
<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;
<!-- sccsID "%TC-INFO%" -->

```

A.15 *OrderStatusRequest DTD*

```

<!-- ===== -->
<!-- ===== Order Lookup Request Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT OrderStatusRequest (RequestHeader, Body)>

<!-- Deprecation: The CustomerID here is deprecated and may be removed in -->
<!-- future versions. Please avoid using it -->
<!ELEMENT Body (AuthorizedAgent, CustomerID?, Order)>

<!-- ===== -->
<!-- ===== List of order query criteria. ===== -->
<!-- ===== -->

<!ELEMENT Order ((
  ReferenceNumber |
  OrderID |
  Customer |
  Product |
  Domain |
  OrderStatusInfo |
  OrderSubmissionDate),TransferPaging?)>

<!-- ===== -->
<!-- ===== Base element declarations ===== -->
<!-- ===== -->

<!ELEMENT Customer (UserID, OrderSubmissionDate?)>
<!ELEMENT Product (ProductID, OrderSubmissionDate?)>
<!ELEMENT Domain (DomainName+, OrderSubmissionDate?)>
<!ELEMENT OrderStatusInfo (OrderStatus, OrderSubmissionDate?)>

<!ELEMENT OrderStatus EMPTY>
<!ATTLIST OrderStatus OrderStatusType (
  success | pending | failed | partial) #REQUIRED >

<!ELEMENT OrderSubmissionDate (DateRange)>

```

```

<!ELEMENT DomainName (#PCDATA)>
<!ELEMENT TransferPaging (StartRecord, MaximumRecords)>
<!ELEMENT StartRecord (#PCDATA)>
<!ELEMENT MaximumRecords (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.16 PartnerManagerRequest DTD

```

<!-- ===== -->
<!-- ==== Partner Manager Request Message and Body Declaration. ===== -->
<!-- ===== -->
<!ELEMENT PartnerManagerRequest (RequestHeader, Body)>
<!ELEMENT Body (FindAllCustomersForPartner | FindAllDomainsForCustomer | FindAllDomainsForPartner |
FindDomainsWithPrivateRegistration | FindDomainsWithWebForwarding | ConsolidateAllCustomerProducts |
ConsolidateSelectedCustomerProducts | GetPartnerData | FindPreRegisteredDomainsForPartner |
FindAllSSLProductsForPartner | FindSSLProductsForPartner | FindDetailedSSLProduct)>
<!-- ===== -->
<!-- ==== Supported Partner Manager Actions. ===== -->
<!-- ===== -->
<!-- Find all customers for a parnter, created within the date range -->
<!ELEMENT FindAllCustomersForPartner (CreatedDateRange?)>
<!-- Find all domains for parnter, purchased within the date range -->
<!-- or having an expiration date within the date range -->
<!ELEMENT FindAllDomainsForPartner (PurchaseDateRange? | ExpirationDateRange?)>
<!-- Find all domains owned by a customer, purchased within the -->
<!-- date range or having an expiration date within the date range -->
<!ELEMENT FindAllDomainsForCustomer (CustomerID, (PurchaseDateRange? | ExpirationDateRange?))>
<!-- Move all products owned by one customer to another customer -->
<!ELEMENT ConsolidateAllCustomerProducts (FromCustomer, ToCustomer)>
<!-- Move the specified products from one customer to another customer -->
<!ELEMENT ConsolidateSelectedCustomerProducts (FromCustomer, ToCustomer, (DomainName+ | ProductID+))>
<!-- Get information about the partner's account -->
<!ELEMENT GetPartnerData EMPTY>
<!-- Get domains with active or not active private registration products -->
<!-- indicated by the PrivateRegistrationActive flag. The value of -->
<!-- PrivateRegistrationActive can be true or false -->
<!ELEMENT FindDomainsWithPrivateRegistration (PrivateRegistrationActive, StartRecord, MaximumRecords)>
<!-- Get webforwarding domains or domains that are not a webforwarding product -->
<!-- indicated by the WithWebFwd flag. The value of -->
<!-- WithWebFwd can be true or false -->
<!ELEMENT FindDomainsWithWebForwarding (WithWebFwd, StartRecord, MaximumRecords)>
<!-- Get PreRegistered domains , TLD will default to .eu for this release -->
<!ELEMENT FindPreRegisteredDomainsForPartner EMPTY>
<!-- High level SSL Products query -->
<!ELEMENT FindAllSSLProductsForPartner EMPTY>
<!ELEMENT FindSSLProductsForPartner (CustomerID | PurchaseDateRange | ExpirationDateRange | WildCardSearch |
DomainName | SSLProductStatus)>
<!-- Detailed query for a specific SSL product -->
<!ELEMENT FindDetailedSSLProduct (ProductID)>
<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->
<!ELEMENT TLD (#PCDATA)>
<!ELEMENT StartRecord (#PCDATA)>
<!ELEMENT MaximumRecords (#PCDATA)>

```

```

<!ELEMENT PrivateRegistrationActive EMPTY>
<!ATTLIST PrivateRegistrationActive
  Selection (True | False) #REQUIRED
>
<!ELEMENT WithWebFwd EMPTY>
<!ATTLIST WithWebFwd
  Selection (True | False) #REQUIRED
>
<!-- For backward compatibility ExpirationDate, PurchaseDate, DomainName are defined here -->
<!ELEMENT CreatedDateRange (DateRange)>
<!ELEMENT PurchaseDateRange (DateRange)>
<!ELEMENT ExpirationDateRange (DateRange)>
<!ELEMENT DomainName (#PCDATA)>
<!ELEMENT WildCardSearch (StartsWith | Contains | EndsWith)>
<!ELEMENT StartsWith (#PCDATA)>
<!ELEMENT Contains (#PCDATA)>
<!ELEMENT EndsWith (#PCDATA)>
<!ELEMENT SSLProductStatus EMPTY>
<!ATTLIST SSLProductStatus
  Selection (ProvideDomain | DomainConfigured | TermStarted | Issued | BeingReissued | Expired | BeingRenewed |
Renewed) #REQUIRED
>
<!-- ===== -->
<!-- ===== Inclusion of referenced common elements. ===== -->
<!-- ===== -->
<ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;
<!-- sccsID "%TC-INFO%" -->

```

A.17 PartnerManagerResponse DTD

```

<!-- ===== -->
<!-- ===== Partner Manager Response Message and Body Declaration. ===== -->
<!-- ===== -->
<!ELEMENT PartnerManagerResponse (ResponseHeader, Body)>
<!ELEMENT Body (Status, (Domain* | Customer* | CustomerProductConsolidationResult |
PartnerData | DomainName* | PreRegisteredDomain* | Product*), DateRange?,
NumberOfRecordsReturned?, NumberOfRecordsAvailable?)>
<!-- ===== -->
<!-- ===== Elements returned from the partner manager request. ===== -->
<!-- ===== -->
<!ELEMENT Domain (ProductID?, DomainName, ProductType, DomainProtect, LegalLock,
NicHandle?, AutoRenew, PurchaseDate, ExpirationDate, BulkWhoisOptOut?, CustomerID?)>
<!ELEMENT PreRegisteredDomain (ProductID?, DomainName, ProductType, (Phase? | Phase1? |
Phase2?), LanguageSelection?, PreRegStatus?, RequestedDate?, AccountID?, AccountHolder?,
OrgName?, FirstName?, LastName?)>
<!ELEMENT Customer (Individual | Business)>
<!ELEMENT Individual (UserID, NicHandle?, LoginName?, FirstName, MiddleName?, LastName)>
<!ELEMENT Business (UserID, NicHandle?, LoginName?, CompanyName, LegalContactFirstName,
LegalContactMiddleName?, LegalContactLastName)>
<!ELEMENT CustomerProductConsolidationResult (FromCustomer, ToCustomer,
ProductConsolidationResult*)>
<!ELEMENT ProductConsolidationResult ((DomainName | ProductID), Status)>
<!ELEMENT PartnerData (AccountBalance?)>
<!ATTLIST PartnerData
  AccountStatus (ACTIVE | SUSPENDED) #REQUIRED
>
<!ELEMENT Product (ProductID, ProductType, Name, DomainName?, PurchasePeriod?,
PurchaseDate?, ExpirationDate, CustomerID?, Status, IsRenewable?, IsReissueable?,
HasEverBeenIssued?, SSLInstance*)>
<!ELEMENT SSLInstance (ServiceInstID, CSR?, ServerSelectionID?, DomainName?, IssuedDate?,
IsRevocable?, IsDownloadable?, ValidationContact, OrderContact, CertificateRequester?,
CertificateApprover?, ContractSigner?)>
<!-- ===== -->

```

```

<!-- ===== Base element declarations ===== -->
<!-- ===== -->
<!-- For backward compatibility ExpirationDate, PurchaseDate, DomainName are defined here -
->
<!ELEMENT PurchaseDate (Date)>
<!ELEMENT ExpirationDate (Date)>
<!ELEMENT DomainName (#PCDATA)>
<!ATTLIST DomainName
    encoded CDATA "false"
>
<!ELEMENT AccountBalance (#PCDATA)>
<!-- PreRegistration query -->
<!ELEMENT Phase (#PCDATA)>
<!ELEMENT Phase1 (#PCDATA)>
<!ELEMENT Phase2 (#PCDATA)>
<!ELEMENT LanguageSelection (#PCDATA)>
<!ELEMENT PreRegStatus (#PCDATA)>
<!ELEMENT RequestedDate (#PCDATA)>
<!ELEMENT AccountID (#PCDATA)>
<!ELEMENT AccountHolder (#PCDATA)>
<!ELEMENT IsRevocable (#PCDATA)>
<!ELEMENT IssuedDate (#PCDATA)>
<!ELEMENT IsDownloadable (#PCDATA)>
<!ELEMENT IsRenewable (#PCDATA)>
<!ELEMENT IsReissueable (#PCDATA)>
<!ELEMENT HasEverBeenIssued (#PCDATA)>
<!-- ===== -->
<!-- ===== Inclusion of externally defined elements. ===== -->
<!-- ===== -->
<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;
<!ENTITY % SSLProduct SYSTEM "SSLProduct.dtd">
%SSLProduct;
<!-- sccsID "%TC-INFO%" -->

```

A.18 PrivateRegistrationProduct DTD

```

<!-- ===== -->
<!-- ===== Private Registrationproduct actions ===== -->
<!-- ===== -->

<!-- ===== Create Private Registrationproducts only ===== -->
<!ELEMENT CreatePrivateRegistration ( DomainName+,
                                     PurchasePeriod ) >

<!-- ===== Delete Private Registrationproducts only ===== -->
<!ELEMENT DeletePrivateRegistration ( DomainName+ ) >

<!-- ===== Renew Private Registrationproducts only ===== -->
<!ELEMENT RenewPrivateRegistration ( DomainName+,
                                     PurchasePeriod ) >

<!-- ===== Modify Private Registrationproducts only ===== -->
<!ELEMENT ModifyPrivateRegistration ( DomainName+ ) >

```

A.19 RequestHeader DTD

```

<!-- ===== -->
<!-- ===== Message Request Header Declarations ===== -->

```

```

<!-- ===== -->
<!ELEMENT RequestHeader (VERSION_4_6, Authentication, Comments?, SystemInfo?, RequestProperty*)>
<!ELEMENT Authentication (PartnerID, PartnerPassword)>
<!ELEMENT PartnerID (#PCDATA)>
<!ELEMENT PartnerPassword (#PCDATA)>
<!ELEMENT SystemInfo (#PCDATA)>
<!ELEMENT RequestProperty (PropertyName, PropertyValue)>
<!ELEMENT PropertyName (#PCDATA)>
<!ELEMENT PropertyValue (#PCDATA)>
<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->
<!ENTITY % CommonDeclarations SYSTEM "CommonDeclarations.dtd">
%CommonDeclarations;
<!-- sccsID "%TC-INFO%" -->

```

A.20 ResponseHeader DTD

```

<!-- ===== -->
<!-- ==== Message Response Header Declarations ===== -->
<!-- ===== -->

<!ELEMENT ResponseHeader (VERSION_4_6, ReferenceNumber, Timestamp, Comments?)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % CommonDeclarations SYSTEM "CommonDeclarations.dtd">
%CommonDeclarations;

<!-- sccsID "%TC-INFO%" -->

```

A.21 SecurityRequest DTD

```

<!-- ===== -->
<!-- ==== Security Request Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT SecurityRequest (RequestHeader, Body)>
<!ELEMENT Body (
    (AuthenticatePartner |
    AuthenticatePartnerAndGetPartnerInfo |
    AuthenticateCustomer |
    AuthenticateCustomerAndGetCustomerInfo |
    AuthorizePartner |
    AuthorizeCustomer ) ) >

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

```

```

<!ELEMENT AuthenticatePartner (Partner)>
<!ELEMENT AuthenticatePartnerAndGetPartnerInfo (Partner)>
<!ELEMENT AuthenticateCustomer (Customer)>
<!ELEMENT AuthenticateCustomerAndGetCustomerInfo (Customer)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.22 SecurityResponse DTD

```

<!-- ===== -->
<!-- ==== Security Response Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT SecurityResponse (ResponseHeader, Body) >
<!ELEMENT Body (Status, UserData?) >

<!ELEMENT UserData (
    OrgName?,
    FirstName?,
    MiddleName?,
    LastName? ) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % ReponseHeader SYSTEM "ResponseHeader.dtd">
%ReponseHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.23 UserLookupRequest DTD

```

<!-- ===== -->
<!-- ==== User Lookup Request Message and Body Declaration. ===== -->
<!-- ===== -->

```

```

<!ELEMENT UserLookupRequest (RequestHeader, Body)>

<!ELEMENT Body (
  AuthorizedAgent,
  CustomerID?,
  (LookupUser | CheckLoginNameAvailability ) ) >

<!ELEMENT LookupUser (
  UserID+ |
  NicHandle+ |
  DomainName |
  ProductID |
  LoginName+ ) >

<!ELEMENT CheckLoginNameAvailability ( LoginName ) >

<!-- ===== -->
<!-- ==== Base element declarations ===== -->
<!-- ===== -->

<!-- For backward compatibility DomainName is defined here -->
<!ELEMENT DomainName (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.24 *UserLookupResponse DTD*

```

<!-- ===== -->
<!-- ==== User Lookup Response Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT UserLookupResponse (ResponseHeader, Body)>
<!ELEMENT Body (Status, User*)>
<!ELEMENT User (Individual | Business)>

<!ELEMENT Individual (
  UserID,
  NicHandle?,
  LoginName?,
  FirstName,
  MiddleName?,
  LastName,
  Address,
  Phone,
  Fax?,
  Email,

```

```

    AuthQuestion,
    AuthAnswer,
    SupplementalRegistryData? ) >

<!ELEMENT Business (
    UserID,
    NicHandle?,
    LoginName?,
    CompanyName,
    Address,
    Phone,
    Fax?,
    Email?,
    LegalContact,
    AuthQuestion,
    AuthAnswer,
    SupplementalRegistryData? ) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;

<!-- sccsID "%TC-INFO%" -->

```

A.25 *UserRequest DTD*

```

<!-- ===== -->
<!-- ==== User Request Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT UserRequest (RequestHeader, Body)>
<!ELEMENT Body (
    ( CreateIndividual |
      CreateBusiness |
      ModifyIndividual |
      ModifyBusiness |
      IndividualNewInfo |
      BusinessNewInfo) ) >

<!ELEMENT CreateIndividual (
    LoginName?,
    Password,
    FirstName,
    MiddleName?,
    LastName,
    Address,
    Phone,
    Fax?,
    Email,

```

```
    AuthQuestion,
    AuthAnswer,
    SupplementalRegistryData? ) >

<!ELEMENT ModifyIndividual (
    AuthorizedAgent,
    CustomerID,
    NewLoginName?,
    NewPassword?,
    FirstName?,
    MiddleName?,
    LastName?,
    Address?,
    Phone?,
    Fax?,
    Email?,
    (AuthQuestion, AuthAnswer)?,
    SupplementalRegistryData? ) >

<!ELEMENT CreateBusiness (
    LoginName?,
    Password,
    CompanyName,
    Address,
    Phone,
    Fax?,
    Email,
    LegalContact,
    AuthQuestion,
    AuthAnswer,
    SupplementalRegistryData? ) >

<!ELEMENT ModifyBusiness (
    AuthorizedAgent,
    CustomerID,
    NewLoginName?,
    NewPassword?,
    CompanyName?,
    Address?,
    Phone?,
    Fax?,
    Email?,
    LegalContact?,
    (AuthQuestion, AuthAnswer)?,
    SupplementalRegistryData? ) >

<!-- Remove all old information and replace with new. -->
<!ELEMENT IndividualNewInfo (
    AuthorizedAgent,
    CustomerID,
    FirstName,
    MiddleName?,
    LastName,
```

```
    Address,
    Phone,
    Fax?,
    Email ) >

<!-- Remove all old information and replace with new. -->
<!ELEMENT BusinessNewInfo (
    AuthorizedAgent,
    CustomerID,
    CompanyName,
    Address,
    Phone,
    Fax?,
    Email,
    LegalContact ) >

<!ELEMENT NewLoginName (#PCDATA)>
<!ELEMENT NewPassword (#PCDATA)>

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % RequestHeader SYSTEM "RequestHeader.dtd">
%RequestHeader;

<!-- sccsID "%TC-INFO%" -->
```

A.26 *UserResponse DTD*

```
<!-- ===== -->
<!-- ==== User Response Message and Body Declaration. ===== -->
<!-- ===== -->

<!ELEMENT UserResponse (ResponseHeader, Body)>

<!ELEMENT Body (
    Status,
    UserID?,
    NicHandle?,
    LoginName? ) >

<!-- ===== -->
<!-- ==== Inclusion of externally defined elements. ===== -->
<!-- ===== -->

<!ENTITY % ResponseHeader SYSTEM "ResponseHeader.dtd">
%ResponseHeader;

<!-- sccsID "%TC-INFO%" -->
```

A.27 *WebForwardingProduct DTD*

```

<!-- ===== -->
<!-- ==== Web Forwarding product actions ===== -->
<!-- ===== -->
<!-- ==== Create Web Forwarding for existing domain ==== -->
<!ELEMENT CreateWebForwarding (WebForwardingProduct+, PurchasePeriod)>
<!-- ==== Modify Web Forwarding products only ===== -->
<!ELEMENT ModifyWebForwarding (WebForwardingProduct+)>
<!-- ==== Delete Web Forwarding products only ===== -->
<!ELEMENT DeleteWebForwarding (DomainName+)>
<!-- ==== Renew Web Forwarding products only ===== -->
<!ELEMENT RenewWebForwarding (DomainName+, PurchasePeriod)>
<!-- ==== Webforwarding elements ===== -->
<!ELEMENT WebForwardingProduct (DomainName, DestURL?, Masking?)>
<!ELEMENT DestURL (#PCDATA)>
<!ELEMENT Masking (#PCDATA)>

```

A.28 *SSLProduct DTD*

```

<!-- ===== -->
<!-- ==== SSL product ===== -->
<!-- ===== -->
<!-- ==== Common commands for both Certificate & Seal products ===== -->
<!-- ==== For Certificate product, CSRData is required for create, renew ===== -->
<!ELEMENT CreateSSL (SSLProductName, (CertProduct+ | SealProduct+), PurchasePeriod,
SubscribersAgreement, AccuracyAgreement?, EVSubscriber?)>
<!ELEMENT RenewSSL (SSLProductName, SSLProductToRenew+, PurchasePeriod, SubscribersAgreement,
AccuracyAgreement?, EVSubscriber?)>
<!ELEMENT ReissueSSL (ProductID, DomainName?, CSRData?, Contacts, AccuracyAgreement?)>
<!ELEMENT RevokeSSL (ServiceInstID, (RevokeReason | OtherReason))>
<!-- ==== Certificate specific commands ===== -->
<!ELEMENT DownloadCertificate (ServiceInstID, CertificateFormat)>
<!-- ==== Seal specific commands ===== -->
<!ELEMENT GetSealCode (ServiceInstID, SealType, SealColor, SealShape, PublishingType)>
<!-- ===== -->
<!-- ===== -->
<!ELEMENT CertProduct (CSRData, Contacts)>
<!ELEMENT SealProduct (DomainName, Contacts)>
<!ELEMENT SSLProductToRenew (ProductID, CSRData?, Contacts)>
<!ELEMENT SSLProductName EMPTY>
<!ATTLIST SSLProductName
    SSLProductName (Basic | Pro | WildCard | SiteConfirmSeal | EV) #REQUIRED
>
<!-- ==== CSR Data ===== -->
<!-- ===== -->
<!ELEMENT CSRData (CSR, ServerSelectionID)>
<!ELEMENT CSR (#PCDATA)>
<!-- Please see XML documentation for a list of server selections -->
<!ELEMENT ServerSelectionID (#PCDATA)>
<!-- Certificate format for downloading certificate-->
<!ELEMENT CertificateFormat EMPTY>
<!ATTLIST CertificateFormat
    Selection (UTNAddTrustServerCA | AddTrustExternalCA | NetworkSolutionsCA | BundledCert |
DomainSpecific | EVRootCA | IntermediateSSLCA) #REQUIRED
>
<!ELEMENT SealType EMPTY>
<!ATTLIST SealType
    Selection (Basic | ProWildCard | SiteConfirmSeal | EV) #REQUIRED
>
<!ELEMENT SealColor EMPTY>
<!ATTLIST SealColor
    Selection (Green | Blue) #REQUIRED

```

```

>
<!ELEMENT SealShape EMPTY>
<!ATTLIST SealShape
  Selection (Square | Rectangular) #REQUIRED
>
<!ELEMENT PublishingType EMPTY>
<!ATTLIST PublishingType
  Selection (HTTP | HTTPS) #REQUIRED
>
<!-- ==== POC Details =====>
<!-- ===== -->
<ELEMENT Contacts (ValidationContact, OrderContact, CertificateRequester?, CertificateApprover?,
ContractSigner?)>
<!-- ==== Validation Contact ===== -->
<ELEMENT ValidationContact (EntityType, AddressLine1, AddressLine2?, AddressLine3?, POBox?,
City, State, PostalCode, CountryCode, Department?, CompanyName?, ContactFirstName,
ContactLastName, ContactPhone, DUNS?, AssumedName?, JurisdictionOfIncorporation?,
IncorporatingAgency?, RegistrationNumber?, MainPhoneNumber?)>
<!-- ==== CertificateRequester ===== -->
<ELEMENT CertificateRequester (FirstName, LastName, Title, RelationshipToApplicant,
AddressLine1, AddressLine2?, AddressLine3?, POBox?, City, State, PostalCode, CountryCode, Phone,
Email)>
<!-- ==== CertificateApprover ===== -->
<ELEMENT CertificateApprover (FirstName, LastName, Title, RelationshipToApplicant, AddressLine1,
AddressLine2?, AddressLine3?, POBox?, City, State, PostalCode, CountryCode, Phone, Email)>
<!-- ==== ContractSigner ===== -->
<ELEMENT ContractSigner (FirstName, LastName, Title, RelationshipToApplicant, AddressLine1,
AddressLine2?, AddressLine3?, POBox?, City, State, PostalCode, CountryCode, Phone, Email)>
<!-- ==== Order Contact ===== -->
<ELEMENT OrderContact (FirstName, LastName, Phone, Email)>
<!-- Entity Type -->
<ELEMENT EntityType EMPTY>
<ATTLIST EntityType
  Selection (Individual | OrganizationCompany | EducationalInstitution | GovernmentOrganization)
#REQUIRED
>
<ELEMENT ContactFirstName (#PCDATA)>
<ELEMENT ContactLastName (#PCDATA)>
<ELEMENT ContactPhone (#PCDATA)>
<ELEMENT Department (#PCDATA)>
<!-- Company Registration Number (optional) -->
<ELEMENT DUNS (#PCDATA)>
<!-- ===== -->
<ELEMENT RevokeReason EMPTY>
<ATTLIST RevokeReason
  Selection (PrivateKeyCompromised | AffiliationChanged | Superseded | CessationOfOperations)
#REQUIRED
>
<ELEMENT OtherReason (#PCDATA)>
<ELEMENT EVSubscriber (FirstName, LastName)>
<ELEMENT JurisdictionOfIncorporation (City?, State?, CountryCode)>
<ELEMENT AssumedName (#PCDATA)>
<ELEMENT Title (#PCDATA)>
<ELEMENT RelationshipToApplicant (#PCDATA)>
<ELEMENT IncorporatingAgency (#PCDATA)>
<ELEMENT RegistrationNumber (#PCDATA)>
<ELEMENT MainPhoneNumber (#PCDATA)>
<!-- ==== Inclusion of externally defined elements. ===== -->
<!--none -->
<!-- sccsID "%TC-INFO%" -->

```

Appendix B XML Status Codes

B.1 UserRequest DTD

Request	Code	External Message	Level
CreateIndividual			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5700	User created successfully.	/UserResponse/Body/Status/StatusCode
	5703	User created successfully, but login name not available; please change later.	/UserResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserResponse/Body/Status/StatusCode
	-201	System Error	/UserResponse/Body/Status/StatusCode
	-245	Data Validation Error	/UserResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2319	Invalid value.	/UserResponse/Body/Status/ValidationError/StatusCode
	-5706	Invalid postal code.	/UserResponse/Body/Status/ValidationError/StatusCode
	-5719	City is required for US address.	/UserResponse/Body/Status/ValidationError/StatusCode
	-5720	State is required for US address.	/UserResponse/Body/Status/ValidationError/StatusCode
	-5721	Postal code is required for US address.	/UserResponse/Body/Status/ValidationError/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/UserResponse/Body/Status/ValidationError/StatusCode
-12018	Unexpected element, missing element, or element out of sequence.	/UserResponse/Body/Status/ValidationError/StatusCode	

Request	Code	External Message	Level
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ModifyIndividual	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5701	User modified successfully.	/UserResponse/Body/Status/StatusCode
	5704	User modified successfully, but login name not available; please change later.	/UserResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserResponse/Body/Status/StatusCode
	-201	System Error	/UserResponse/Body/Status/StatusCode
	-245	Data Validation Error	/UserResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/UserResponse/Body/Status/StatusCode
	-5715	User not found.	/UserResponse/Body/Status/StatusCode
	-2022	Invalid Authorized Agent	/UserResponse/Body/Status/StatusCode
	-2307	Validation error.	/UserResponse/Body/Status/Validation/Status Code
	-2319	Invalid value.	/UserResponse/Body/Status/Validation/Status Code
	-5700	Invalid user type.	/UserResponse/Body/Status/Validation/Status Code
	-5735	User not a individual user.	/UserResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
-12012	Element not found in DTD	/UserResponse/Body/Status/Validation/Status Code	
-12018	Unexpected element, missing element, or element out of sequence.	/UserResponse/Body/Status/Validation/Status Code	

Request	Code	External Message	Level
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
CreateBusiness	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5700	User created successfully.	/UserResponse/Body/Status/StatusCode
	5703	User created successfully, but login name not available; please change later.	/UserResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserResponse/Body/Status/StatusCode
	-201	System Error	/UserResponse/Body/Status/StatusCode
	-245	Data Validation Error	/UserResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2307	Validation error.	/UserResponse/Body/Status/Validation/Status Code
	-2319	Invalid value.	/UserResponse/Body/Status/Validation/Status Code
	-5706	Invalid postal code.	/UserResponse/Body/Status/Validation/Status Code
	-5719	City is required for US address.	/UserResponse/Body/Status/Validation/Status Code
	-5720	State is required for US address.	/UserResponse/Body/Status/Validation/Status Code
	-5721	Postal code is required for US address.	/UserResponse/Body/Status/Validation/Status Code
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/UserResponse/Body/Status/Validation/Status Code
-12018	Unexpected element, missing element, or element out of sequence.	/UserResponse/Body/Status/Validation/Status Code	

Request	Code	External Message	Level
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ModifyBusiness	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5701	User modified successfully.	/UserResponse/Body/Status/StatusCode
	5704	User modified successfully, but login name not available; please change later.	/UserResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserResponse/Body/Status/StatusCode
	-201	System Error	/UserResponse/Body/Status/StatusCode
	-245	Data Validation Error	/UserResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/UserResponse/Body/Status/StatusCode
	-5715	User not found.	/UserResponse/Body/Status/Validation/Status Code
	-2022	Invalid Authorized Agent	/UserResponse/Body/Status/StatusCode
	-2307	Validation error.	/UserResponse/Body/Status/Validation/Status Code
	-2319	Invalid value.	/UserResponse/Body/Status/Validation/Status Code
	-5700	Invalid user type.	/UserResponse/Body/Status/StatusCode
	-5736	User not a business user.	/UserResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
-12012	Element not found in DTD	/UserResponse/Body/Status/Validation/Status Code	
-12018	Unexpected element, missing element, or element out of sequence.	/UserResponse/Body/Status/Validation/Status Code	

Request	Code	External Message	Level
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode

B.2 LookupUserRequest.DTD

Request	Code	External Message	Level
UserLookup			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5702	User lookup successful.	/UserLookupResponse/Body/Status/StatusCode
	5706	User lookup partially successful	/UserLookupResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserLookupResponse/Body/Status/StatusCode
	-201	System Error	/UserLookupResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2002	User not Authenticated	/UserLookupResponse/Body/Status/StatusCode
	-2004	User not Authorized	/UserLookupResponse/Body/Status/StatusCode
	-2022	Invalid Authorized Agent	/UserLookupResponse/Body/Status/StatusCode
	-2023	Invalid Authorized Agent	/UserLookupResponse/Body/Status/StatusCode
	-5700	Invalid user type.	/UserLookupResponse/Body/Status/StatusCode
	-5715	User not found.	/UserLookupResponse/Body/Status/StatusCode
	-7057	Request too large.	/UserLookupResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserLookupResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/UserLookupResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/UserLookupResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
CheckLoginName Availability			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5705	Login Name Available	/UserLookupResponse/Body/Status/StatusCode
	-200	System Unavailable.	/UserLookupResponse/Body/Status/StatusCode
	-201	System Error	/UserLookupResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-5737	Login Name not available	/UserLookupResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/UserLookupResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/UserLookupResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/UserLookupResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode

B.3 SecurityRequest.DTD

Request	Code	External Message	Level
AuthenticatePartner			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	2000	Partner Authenticated	/SecurityResponse/Body/Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/Status/StatusCode
	-201	System Error	/SecurityResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	- 12001	XML is invalid.	/SecurityResponse/Body/ Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Sta tusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Sta tusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Sta tusCode
	- 12012	Element not found in DTD	/SecurityResponse/Body/ Status/ValidationError/Sta tusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/ Status/ValidationError/Sta tusCode
	- 12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Sta tusCode
AuthenticatePartner GetPartnerInfo	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Sta tusCode
	2000	Partner Authenticated	/SecurityResponse/Body/ Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/ Status/StatusCode
	-201	System Error	/SecurityResponse/Body/ Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Sta tusCode
	- 12000	XML is malformed.	/ErrorResponse/Body/Sta tusCode
	- 12001	XML is invalid.	/SecurityResponse/Body/ Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Sta tusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Sta tusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Sta tusCode
	- 12012	Element not found in DTD	/SecurityResponse/Body/ Status/ValidationError/Sta tusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/ Status/ValidationError/Sta tusCode
	- 12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Sta tusCode
Authenticate Customer	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Sta tusCode
	2002	User Authenticated	/SecurityResponse/Body/ Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/ Status/StatusCode
	-201	System Error	/SecurityResponse/Body/ Status/StatusCode

Request	Code	External Message	Level
	- 12000	XML is malformed.	/ErrorResponse/Body/Status/StatusCode
	- 12001	XML is invalid.	/SecurityResponse/Body/Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status/StatusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status/StatusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status/StatusCode
	- 12012	Element not found in DTD	/SecurityResponse/Body/Status/Validation/Status/StatusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/Status/Validation/Status/StatusCode
	- 12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode
Authenticate CustomerAnd GetCustomerInfo			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status/StatusCode
	2002	User Authenticated	/SecurityResponse/Body/Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/Status/StatusCode
	-201	System Error	/SecurityResponse/Body/Status/StatusCode
	-2002	User not Authenticated	/ErrorResponse/Body/Status/StatusCode
	- 12000	XML is malformed.	/ErrorResponse/Body/Status/StatusCode
	- 12001	XML is invalid.	/SecurityResponse/Body/Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status/StatusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status/StatusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status/StatusCode
	- 12012	Element not found in DTD	/SecurityResponse/Body/Status/Validation/Status/StatusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/Status/Validation/Status/StatusCode
	- 12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode
AuthorizePartner			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status/StatusCode
	2001	Partner Authorized	/SecurityResponse/Body/Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-201	System Error	/SecurityResponse/Body/Status/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/Status/StatusCode
	-12001	XML is invalid.	/SecurityResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status/StatusCode
	-12012	Element not found in DTD	/SecurityResponse/Body/Status/ValidationError/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/Status/ValidationError/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode
AuthorizeCustomer	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status/StatusCode
	2003	User Authorized	/SecurityResponse/Body/Status/StatusCode
	-200	System Unavailable.	/SecurityResponse/Body/Status/StatusCode
	-201	System Error	/SecurityResponse/Body/Status/StatusCode
	-2003	User not Authorized	/SecurityResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/Status/StatusCode
	-12001	XML is invalid.	/SecurityResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status/StatusCode
	-12012	Element not found in DTD	/SecurityResponse/Body/Status/ValidationError/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/SecurityResponse/Body/Status/ValidationError/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode

B.4 AvailableRequest.DTD

Request	Code	External Message	Level
VerifyDomainName			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	-200	System Unavailable.	/AvailableResponse/Body/Status/StatusCode
	-201	System Error	/AvailableResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/AvailableResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/AvailableResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body/Status/Validation/StatusCode
	-12026	Request should be sent to Available Service	/ErrorResponse/Body/StatusCode
VerifyDomains			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	-200	System Unavailable.	/AvailableResponse/Body/Status/StatusCode
	-201	System Error	/AvailableResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/AvailableResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/ValidationError/St atusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/ValidationError/St atusCode
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Sta tusCode
GenerateDomains	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Sta tusCode
	-200	System Unavailable.	/AvailableResponse/Body /Status/StatusCode
	-201	System Error	/AvailableResponse/Body /Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Sta tusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/Sta tusCode
	- 12000	XML is malformed.	/ErrorResponse/Body/Sta tusCode
	- 12001	XML is invalid.	/AvailableResponse/Body /Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Sta tusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Sta tusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Sta tusCode
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/ValidationError/St atusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/ValidationError/St atusCode
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Sta tusCode
	VerifyDomains Avanced	209	Front-End Gateway temporarily unavailable
-200		System Unavailable.	/AvailableResponse/Body /Status/StatusCode
-201		System Error	/AvailableResponse/Body /Status/StatusCode
-2000		Partner not Authenticated	/ErrorResponse/Body/Sta tusCode
-2001		Partner not Authorized	/ErrorResponse/Body/Sta tusCode
- 12000		XML is malformed.	/ErrorResponse/Body/Sta tusCode
- 12001		XML is invalid.	/AvailableResponse/Body /Status/StatusCode

Request	Code	External Message	Level
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status Code
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status Code
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/ValidationError/Status Code
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/ValidationError/Status Code
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Status Code
GenerateDomains Advanced	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status Code
	-200	System Unavailable.	/AvailableResponse/Body /Status/StatusCode
	-201	System Error	/AvailableResponse/Body /Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status Code
	-2001	Partner not Authorized	/ErrorResponse/Body/Status Code
	- 12000	XML is malformed.	/ErrorResponse/Body/Status Code
	- 12001	XML is invalid.	/AvailableResponse/Body /Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status Code
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status Code
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/ValidationError/Status Code
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/ValidationError/Status Code
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Status Code
VerifyDomainsAll	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status Code
	-200	System Unavailable.	/AvailableResponse/Body /Status/StatusCode
	-201	System Error	/AvailableResponse/Body /Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status Code
	-2001	Partner not Authorized	/ErrorResponse/Body/Status Code

Request	Code	External Message	Level
	- 12000	XML is malformed.	/ErrorResponse/Body/Status statusCode
	- 12001	XML is invalid.	/AvailableResponse/Body /Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status statusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status statusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status statusCode
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/Validation/Status statusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/Validation/Status statusCode
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Status statusCode
GenerateDomainsAll			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status statusCode
	-200	System Unavailable.	/AvailableResponse/Body /Status/StatusCode
	-201	System Error	/AvailableResponse/Body /Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status statusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/Status statusCode
	- 12000	XML is malformed.	/ErrorResponse/Body/Status statusCode
	- 12001	XML is invalid.	/AvailableResponse/Body /Status/StatusCode
	- 12002	XML contains invalid root tag.	/ErrorResponse/Body/Status statusCode
	- 12003	XML contains invalid header.	/ErrorResponse/Body/Status statusCode
	- 12004	XML contains invalid tag.	/ErrorResponse/Body/Status statusCode
	- 12012	Element not found in DTD	/AvailableResponse/Body /Status/Validation/Status statusCode
	- 12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body /Status/Validation/Status statusCode
	- 12026	Request should be sent to Available Service	/ErrorResponse/Body/Status statusCode
IsDomainCandidate ForInboundTransfer			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status statusCode
	-200	System Unavailable.	/AvailableResponse/Body /Status/StatusCode
	-201	System Error	/AvailableResponse/Body /Status/StatusCode

Request	Code	External Message	Level
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2312	Invalid domain.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7030	The requested action is not supported for the designated products(s).	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7046	Domain name is already registered with Network Solutions	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-11636	Domain already registered	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/AvailableResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/AvailableResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body/Status/ValidationError/StatusCode
	-12026	Request should be sent to Available Service	/ErrorResponse/Body/StatusCode
IsDomainCandidateForInternalTransfer	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	-200	System Unavailable.	/AvailableResponse/Body/Status/StatusCode
	-201	System Error	/AvailableResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2312	Invalid domain.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7030	The requested action is not supported for the designated products(s).	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7045	Another transfer request is already in process for this domain. Transfer process cannot be initiated.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8017	Domain not found.	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8029	Domain in hold state	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8030	Domain in locked state	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8031	Name change or domain transfer pending	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8033	Domain in unpaid state	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8034	Domain in special state	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode

Request	Code	External Message	Level
	-8037	Domain already owned by customer	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8038	Domain not registered with Network Solutions	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-8039	Domain in expired state	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-11636	Domain already registered	/AvailableResponse/Body/NotTransferrable/NotTransferrableReasons/NotTransferrableReason/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/AvailableResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/AvailableResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/AvailableResponse/Body/Status/ValidationError/StatusCode
	-12026	Request should be sent to Available Service	/ErrorResponse/Body/StatusCode

B.5 OrderRequest.DTD

Request	Code	External Message	Level
CreateRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-11607	Domain already registered	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12006	Incorrect value of attribute	/OrderResponse/Body//Status/ValidationError/StatusCode
	-12008	Input data is longer than max length	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body//Status/ValidationError/StatusCode
	-12022	Missing attribute or element	/OrderResponse/Body/Order/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ModifyRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2019	User not Authenticated for Product	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/Validation/ValidationError/StatusCode
	-8019	Unable to perform action. Incorrect product type	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8508	Not enough hosts.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8510	Duplicate hosts detected	/OrderResponse/Body/Order/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
RenewRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
			/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2019	User not Authenticated for Product	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2028	Your account credit is not sufficient. The requested action will not be processed	
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/Status/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode

Request	Code	External Message	Level
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11700	Product unpaid.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
TransferInternalToRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2025	Transfer Failed - Invalid Auth Code	/OrderResponse/Body/Order/Status/StatusCode
	-2027	Transfer Failed - Missing Auth Code	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2319	Invalid value.	OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7030	The requested action is not supported for the designated products(s).	OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-7032	Duplicate request property.	ErrorResponse/Body/StatusCode
	-7033	Reference Number not owned by partner.	ErrorResponse/Body/StatusCode
	-7034	Unknown request property	ErrorResponse/Body/StatusCode
	-7036	Reference Number invalid value.	ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7057	Request too large.	/OrderResponse/Body/Order/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8017	Domain not found.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8029	Domain in hold state	OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8030	Domain in locked state	OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8031	Name change or domain transfer pending	/OrderResponse/Body/Order/Status/StatusCode
	-8034	Domain in special state	OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8037	Domain already owned by customer	/OrderResponse/Body/Order/Status/StatusCode
	-8038	Domain not registered with Network Solutions	OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8039	Domain in expired state	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8508	Not enough hosts.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8510	Duplicate hosts detected	/OrderResponse/Body/Order/Status/StatusCode
	-11636	Domain already registered	OrderResponse/Body/Order/LinItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode

Request	Code	External Message	Level
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode
InboundTransfer ToRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/Status/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2025	Transfer Failed - Invalid Auth Code	/OrderResponse/Body/Order/Status/StatusCode
	-2027	Transfer Failed - Missing Auth Code	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/Validation/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7030	The requested action is not supported for the designated products(s).	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-7032	Duplicate request property.	/ErrorResponse/Body/StatusCode
	-7033	Reference Number not owned by partner.	/ErrorResponse/Body/StatusCode
	-7034	Unknown request property	/ErrorResponse/Body/StatusCode
	-7036	Reference Number invalid value.	/ErrorResponse/Body/StatusCode
	-7046	Domain name is already registered with Network Solutions	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-8508	Not enough hosts.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8510	Duplicate hosts detected	/OrderResponse/Body/Order/Status/StatusCode
	-11636	Domain already registered	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12006	Incorrect value of attribute	/OrderResponse/Body/Status/Validation/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
DeleteRegistration	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-7032	Duplicate request property.	/ErrorResponse/Body/StatusCode
	-7033	Reference Number not owned by partner.	/ErrorResponse/Body/StatusCode
	-7034	Unknown request property	/ErrorResponse/Body/StatusCode
	-7036	Reference Number invalid value.	/ErrorResponse/Body/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-8019	Unable to perform action. Incorrect product type	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
CreateParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11607	Domain already registered	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ModifyParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11000		/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
RenewParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2019	User not Authenticated for Product	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/Status/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-8019	Unable to perform action. Incorrect product type	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11000		/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11700	Product unpaid.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
TransferInternalToParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-8017	Domain not found.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8029	Domain in hold state	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8030	Domain in locked state	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8031	Name change or domain transfer pending	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8033	Domain in unpaid state	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8034	Domain in special state	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8037	Domain already owned by customer	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8038	Domain not registered with Network Solutions	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11000	Domain is not a parking product.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11636	Domain already registered	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
InboundTransfer To	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-7036	Reference Number invalid value.	/ErrorResponse/Body/StatusCode
	-7046	Domain name is already registered with Network Solutions	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11636	Domain already registered	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12006	Incorrect value of attribute	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ConvertToParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7032	Duplicate request property.	/ErrorResponse/Body/StatusCode
	-7033	Reference Number not owned by partner.	/ErrorResponse/Body/StatusCode
	-7034	Unknown request property	/ErrorResponse/Body/StatusCode
	-7036	Reference Number invalid value.	/ErrorResponse/Body/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-8019	Unable to perform action. Incorrect product type	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12006	Incorrect value of attribute	/OrderResponse/Body/Status/Validation/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
DeleteParking	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/Validation/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11700	Product unpaid.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
CasacadeDelete Domain	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/Validation/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-8011	Domain does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
LookupDomain	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	11702	Product lookup successful.	/OrderResponse/Body/Order/Status/StatusCode
	11703	Product lookup partially successful	/OrderResponse/Body/Order/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-7057	Request too large.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-11702	Domain name does not have associated service.	/OrderResponse/Body/Order/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
CreateHost	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/Validation/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Validation/Status/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/Status/StatusCode

Request	Code	External Message	Level
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8506	Host already exists.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8508	Not enough hosts.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/Validation/Validation/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/Status/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/Validation/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/Validation/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status/StatusCode
ModifyHost	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/ValidationError/StatusCode
	-8507	Host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8516	No new data found to modify host.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode

Request	Code	External Message	Level
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
DeleteHost	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7020	Parent domain of host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-8507	Host does not exist.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-8509	Host not found	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
LookupHost	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	11702	Product lookup successful.	/OrderResponse/Body/Order/Status/StatusCode
	11703	Product lookup partially successful	/OrderResponse/Body/Order/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-7057	Request too large.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11702	Domain name does not have associated service.	/OrderResponse/Body/Order/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/StatusCode

Request	Code	External Message	Level
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
Create WebForwarding	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LineItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2304	Type is invalid.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2317	Value not unique	/OrderResponse/Body/Order/LineItem/Status/Validation/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/Validation/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/Validation/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8028	Resource records exceed max allowed.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8042	Web Forwarding has not yet been added to this domain	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8045	The domain already has Web Forwarding	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8046	Invalid Web Forwarding URL	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-8048	Invalid Webforwarding with unsupported protocol	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-11607	Domain already registered	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
PreRegister	-2324	Invalid language selection	/OrderResponse/Body/Order/Status/StatusCode
	-2325	EU Exhibit value must be TRUE	/OrderResponse/Body/Order/Status/StatusCode
	-2326	NaturalPerson attribute required for ResidentInEU Selection	/OrderResponse/Body/Order/Status/StatusCode
	-5744	Customer country code is not a valid member	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
CreateSSL		Note: The following error codes are new and specific to CreateSSL. Other errors returned are similar to Order requests .	

Request	Code	External Message	Level
	-2300	Attribute length is invalid (DUNS is limited to 20 characters)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-5703	Invalid State (This error will be returned if the state is not a valid US State if country code is US – This applies to Validation Contact, Certificate Requester, Certificate Approver and Contract Signer elements).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11900	Partner does not have purchase agreement. (Partner must agree to the purchase agreement before issuing the CreateSSL request. See CreateSSL section of the OrderRequest DTD)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11901	Subscriber's agreement must be true for this action.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11902	Invalid server selection. (See Appendix F for a list of valid server selections).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11903	Required fields are missing (For Certificates, CSR must contain domain name and company name. For Site Confirm Seal, DomainName element and ValidationContact/CompanyName element must be filled out).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11904	CSR validation failed. (This error will be returned if the data in the CSR fails validation. Additional status codes and description may be returned in the ValidationError element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11905	CSR parsing failed. (This error will be returned if the CSR format has an error and cannot be read).	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
	-11906	Domain parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain parsing fails).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11907	Seal parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain/URL has invalid format. Additional status codes and description may be returned in the Validation Error element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11908	Invalid SSL Product. (This error will be returned if the information provided for the product does not correspond to the selected product type – Basic, Pro, WildCard or Site Confirm Seal).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11915	Cannot override domain name or company name for Cert product. (This error will be returned if the ValidationContact/CompanyName is filled out for the Basic, Pro or Wildcard product. CompanyName should be provided in the CSR).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11918	Accuracy agreement must be true for this action (Required for EV Service)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11919	EVSubscriber is required and must contain first and lastname. (Required for EV Service)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11920	Selected EntityType is not valid for this product. (EV Service cannot select "Individual" as a Validation Contact Entity type)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11921	State is required if country code is US (State is required and must be a valid US State if country code is US).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11924	Required fields are missing for SSL EV (This error will be returned if required fields for EV are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11925	Required fields are missing for SSL EV Contact (This error will be returned if required fields for EV Contacts are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11926	The CSR that you submitted has been rejected because the Debian Linux operating system used to generate it is producing keys that are not secure	/OrderResponse/Body/Order/LineItem/Status/StatusCode

Request	Code	External Message	Level
RenewSSL		Note: The following error codes are new and specific to RenewSSL	
	-2300	Attribute length is invalid (DUNS is limited to 20 characters)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-5703	Invalid State (This error will be returned if the state is not a valid US State if country code is US – This applies to Validation Contact, Certificate Requester, Certificate Approver and Contract Signer elements).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11702	Product not found (This error will be returned if the product does not exist in our database).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	- 11708	Product is not renewable. (This error will be returned if the expiration date is not within the renewable period).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11900	Partner does not have purchase agreement. (Partner must agree to the purchase agreement before issuing the ReneweSSL request. See ReneweSSL section of the OrderRequest DTD)	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11901	Subscriber's agreement must be true for this action.	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11902	Invalid server selection. (See Appendix F for a list of valid server selections).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11903	Required fields are missing. (For Certificates, CSR must contain domain name and company name. For Site Confirm Seal, DomainName element and ValidationContact/CompanyName element must be filled out).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
	-11904	CSR validation failed. (This error will be returned if the data in the CSR fails validation. Additional status codes and description may be returned in the ValidationError element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LineItem/Status/StatusCode
-11905	CSR parsing failed. (This error will be returned if the CSR format has an error and cannot be read).	/OrderResponse/Body/Order/LineItem/Status/StatusCode	

Request	Code	External Message	Level
	-11906	Domain parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain parsing fails).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11907	Seal parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain/URL has invalid format. Additional status codes and description may be returned in the Validation Error element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11908	Invalid SSL Product. (This error will be returned if the information provided for the product does not correspond to the selected product type – Basic, Pro, WildCard or Site Confirm Seal).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11909	Domain name does not match original product's (This error will be returned if the domain name provided in the CSR does not match with the original domain name given by the ProductID)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11910	Domain name change is not allowed for repurchased product (This error will be returned if the product has been renewed once already).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11915	Cannot override domain name or company name for Cert product. (This error will be returned if the ValidationContact/CompanyName is filled out for the Basic, Pro or Wildcard product. CompanyName should be provided in the CSR).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11918	Accuracy agreement must be true for this action (Required for EV Service)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11919	EVSubscriber is required and must contain first and lastname. (Required for EV Service)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11920	Selected EntityType is not valid for this product. (EV Service cannot select "Individual" as a Validation Contact Entity type)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11921	State is required if country code is US (State is required and must be a valid US State if country code is US).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11924	Required fields are missing for SSL EV (This error will be returned if required fields for EV are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LinItem/Status/StatusCode

Request	Code	External Message	Level
ReissueSSL	-11925	Required fields are missing for SSL EV Contact (This error will be returned if required fields for EV Contacts are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11926	The CSR that you submitted has been rejected because the Debian Linux operating system used to generate it is producing keys that are not secure	/OrderResponse/Body/Order/LinItem/Status/StatusCode
		Note: The following error codes are new and specific to ReissueSSL.	
	-2300	Attribute length is invalid (DUNS is limited to 20 characters)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-5703	Invalid State (This error will be returned if the state is not a valid US State if country code is US – This applies to Validation Contact, Certificate Requester, Certificate Approver and Contract Signer elements).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11709	Product is not reissueable	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11702	Product not found (This error will be returned if the product does not exist in our database).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11902	Invalid server selection. (See Appendix F for a list of valid server selections).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11903	Required fields are missing. (For Certificates, CSR must contain domain name and company name. For Site Confirm Seal, DomainName element and ValidationContact/CompanyName element must be filled out).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11904	CSR validation failed. (This error will be returned if the data in the CSR fails validation. Additional status codes and description may be returned in the ValidationError element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
-11905	CSR parsing failed. (This error will be returned if the CSR format has an error and cannot be read).	/OrderResponse/Body/Order/LinItem/Status/StatusCode	

Request	Code	External Message	Level
	-11906	Domain parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain parsing fails).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11907	Seal parsing failed. (This error is applicable to Site Confirm Seal. It will be returned if the domain/URL has invalid format. Additional status codes and description may be returned in the Validation Error element. Please see the Validation Error section for details).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11908	Invalid SSL Product. (This error will be returned if the information provided for the product does not correspond to the selected product type – Basic, Pro, WildCard or Site Confirm Seal).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11913	Domain name change is not allowed product has been issued. (This error will be returned if the request is attempting to change the domain name but the product has been issued).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11914	Reissue is not allowed, product must be in Provide CSR or Provide Domain/Validation status. (This error will be returned if the ReissueSSL request is sent while the product is “In Validation” state. See case (2) and case (3) descriptions in the ReissueSSL section for appropriate steps).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11918	Accuracy agreement must be true for this action (Required for EV Service)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11919	EVSubscriber is required and must contain first and lastname. (Required for EV Service)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11920	Selected EntityType is not valid for this product. (EV Service cannot select “Individual” as a Validation Contact Entity type)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11921	State is required if country code is US (State is required and must be a valid US State if country code is US).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11924	Required fields are missing for SSL EV (This error will be returned if required fields for EV are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LinItem/Status/StatusCode

Request	Code	External Message	Level
RevokeSSL	-11925	Required fields are missing for SSL EV Contact (This error will be returned if required fields for EV Contacts are missing. Refer to CreateSSL in section 2.2.1.6)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11926	The CSR that you submitted has been rejected because the Debian Linux operating system used to generate it is producing keys that are not secure	/OrderResponse/Body/Order/LinItem/Status/StatusCode
		Note: The following error codes are new and specific to RevokeSSL.	
	-11702	Product not found (This error will be returned if the product does not exist in our database).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11710	Product is not revokeable. (This error will be returned if the partner is attempting to revoke a seal product.)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
		Note: The following error codes are new and specific to DownloadCertificate	
DownloadCertificate	-11702	Product not found (This error will be returned if the product does not exist in our database).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11711	Product has not been issued. (This error will be returned if partner attempts to download a certificate that has not been issued).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11911	Certificate not found. (This error will be returned if the selected certificate cannot be found)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11912	Failed to create bundled certificate. (This error will be returned if the selected BundledCert format cannot be created.)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11922	Requested cert format does not match product code. (This error will be returned if the requested CertificateFormat is EVRootCA or IntermediateSSLCA but the is not an EV service but the ServiceInstID is not associated with an EV service)	/OrderResponse/Body/Order/LinItem/Status/StatusCode
		Note: The following error codes are new and specific to DownloadCertificate	
GetSealCode			

Request	Code	External Message	Level
	-11702	Product not found (This error will be returned if the product does not exist in our database).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11711	Product has not been issued. (This error will be returned if partner attempts to download a certificate that has not been issued).	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-11916	Seal code not found (This error will be returned if the seal code does not exist in our database).	
	-11917	Requested seal code type does not match product. (This error will be returned if the product type does not match requested seal code selection. i.e. Basic product can only download Basic seal etc.. See GetSealCode request in the OrderRequest DTD section).	/OrderResponse/Body/Order/LinItem/Status/StatusCode

B.7 EmailForwardingRequest.DTD

Request	Code	External Message	Level
CreateEmail Forwarding AndDomain			
CreateRegistration			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2003	User not Authorized	/OrderResponse/Body/Order/LinItem/Status/StatusCode

Request	Code	External Message	Level
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
CreateParking	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2003	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2313	Invalid host.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode

Request	Code	External Message	Level
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
ModifyEmail Forwarding	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-11702	Domain name does not have associated service.	/OrderResponse/Body/Order/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
RenewEmail ForwardingAnd Domain	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode /OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode

Request	Code	External Message	Level
	-2312	Invalid domain.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7021	Purchase period is invalid.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-11702	Domain name does not have associated service.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
DeleteEmail ForwardingAnd Domain	7000	Order submitted successfully and is in pending status.	/OrderResponse/Body/Order/Status/StatusCode
	7001	Part of the requested order submitted and is being fulfilled.	/OrderResponse/Body/Order/Status/StatusCode
	7004	Line item submitted to back office successfully.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-200	System Unavailable.	/OrderResponse/Body/Order/Status/StatusCode
	-201	System Error	/OrderResponse/Body/Order/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2004	User not Authorized	/OrderResponse/Body/Order/Status/StatusCode
	-2022	Invalid Authorized Agent	/OrderResponse/Body/Order/Status/StatusCode
	-2312	Invalid domain.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2317	Value not unique.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/Status/ValidationError/StatusCode
	-2319	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-2320	Invalid value.	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7016	All products are invalid. Order was not submitted.	/OrderResponse/Body/Order/Status/StatusCode
	-7023	Line item is valid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-7024	Line item is invalid. Order was not submitted	/OrderResponse/Body/Order/LinItem/Status/StatusCode
	-8014	Unicode name failed during encoding or decoding.	/OrderResponse/Body/Order/LinItem/Status/ValidationError/StatusCode
	-11702	Domain name does not have associated service.	/OrderResponse/Body/Order/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
-12001	XML is invalid.	/OrderResponse/Body/Status/StatusCode	
-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode	

Request	Code	External Message	Level
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/OrderResponse/Body/Status/Validation/Status/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode

B.8 EmailForwardingLookupRequest.DTD

Request	Code	External Message	Level
LookupEmail Forwarding			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	11702	Product lookup successful.	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-200	System Unavailable.	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-201	System Error	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-2022	Invalid Authorized Agent	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-11702	Domain name does not have associated service.	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/EmailForwardingLookupResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode

Request	Code	External Message	Level
	-12012	Element not found in DTD	/EmailForwardingResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/EmailForwardingResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode

B.9 PartnerManagerRequest.DTD

Request	Code	External Message	Level
FindAllCustomersFor Partner			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	5702	User lookup successful.	/PartnerManagerResponse/Body/Status/StatusCode
	-200	System Unavailable.	/PartnerManagerResponse/Body/Status/StatusCode
	-201	System Error	/PartnerManagerResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/PartnerManagerResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
-12023	Undefined element	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode	
-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode	
FindAllDomainsFor Customer			
	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	11702	Product lookup successful.	/PartnerManagerResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-200	System Unavailable.	/PartnerManagerResponse/Body/Status/StatusCode
	-201	System Error	/PartnerManagerResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-7027	Date is formatted incorrectly.	/PartnerManagerResponse/Body/Status/StatusCode
	-7028	Invalid date, Date must be later than Jan 1st, 1990	/PartnerManagerResponse/Body/Status/StatusCode
	-11702	Domain name does not have associated service.	/PartnerManagerResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/StatusCode
	-12001	XML is invalid.	/PartnerManagerResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/StatusCode
	-12003	XML contains invalid header.	/ErrorResponse/Body/StatusCode
	-12004	XML contains invalid tag.	/ErrorResponse/Body/StatusCode
	-12012	Element not found in DTD	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12023	Undefined element	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
FindAllDomainsFor Partner	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode
	11702	Product lookup successful.	/PartnerManagerResponse/Body/Status/StatusCode
	-200	System Unavailable.	/PartnerManagerResponse/Body/Status/StatusCode
	-201	System Error	/PartnerManagerResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/StatusCode
	-2001	Partner not Authorized	/ErrorResponse/Body/StatusCode
	-7028	Invalid date, Date must be later than Jan 1st, 1990	/PartnerManagerResponse/Body/Status/StatusCode
	-11702	Domain name does not have associated service.	/PartnerManagerResponse/Body/Status/StatusCode

Request	Code	External Message	Level
	-12000	XML is malformed.	/ErrorResponse/Body/Status Code
	-12001	XML is invalid.	/PartnerManagerResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status Code
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status Code
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code
	-12012	Element not found in DTD	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12023	Undefined element	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status Code
ConsolidateAllCustomerProducts	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status Code
	7012	Command completed successfully	/PartnerManagerResponse/Body/Status/StatusCode
	-200	System Unavailable.	/PartnerManagerResponse/Body/Status/StatusCode
	-201	System Error	/PartnerManagerResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status Code
	-2001	Partner not Authorized	/ErrorResponse/Body/Status Code
	-2003	User not Authorized	/PartnerManagerResponse/Body/Status/StatusCode
	-7030	The requested action is not supported for the designated products(s).	/PartnerManagerResponse/Body/Status/StatusCode
	-7056	Customer name does not match.	/PartnerManagerResponse/Body/Status/StatusCode
	-7054	FromCustomer and ToCustomer are the same	/PartnerManagerResponse/Body/Status/StatusCode
	-11702	Domain name does not have associated service.	/PartnerManagerResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/Status Code
	-12001	XML is invalid.	/PartnerManagerResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status Code
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status Code

Request	Code	External Message	Level
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code
	-12012	Element not found in DTD	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12023	Undefined element	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/Status Code
ConsolidateSelected CustomerProducts	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/Status Code
	7012	Command completed successfully	/PartnerManagerResponse/Body/Status/StatusCode
	7014	Product and all its children products consolidated	/PartnerManagerResponse/Body/ProductConsolidationResult/Status/StatusCode
	-200	System Unavailable.	/PartnerManagerResponse/Body/Status/StatusCode
	-201	System Error	/PartnerManagerResponse/Body/Status/StatusCode
	-2000	Partner not Authenticated	/ErrorResponse/Body/Status Code
	-2001	Partner not Authorized	/ErrorResponse/Body/Status Code
	-2019	User not Authenticated	/PartnerManagerResponse/Body/ProductConsolidationResult/Status/StatusCode
	-2320	Invalid line item. Validation error	/PartnerManagerResponse/Body/ProductConsolidationResult/Status/StatusCode
	-7030	The requested action is not supported for the designated products(s).	/PartnerManagerResponse/Body/Status/StatusCode
	-7053	Command failed.	/PartnerManagerResponse/Body/Status/StatusCode
	-7056	Customer name does not match.	/PartnerManagerResponse/Body/Status/StatusCode
	-8011	Domain does not exist.	/PartnerManagerResponse/Body/Status/StatusCode
	-12000	XML is malformed.	/ErrorResponse/Body/Status Code
	-12001	XML is invalid.	/PartnerManagerResponse/Body/Status/StatusCode
	-12002	XML contains invalid root tag.	/ErrorResponse/Body/Status Code
	-12003	XML contains invalid header.	/ErrorResponse/Body/Status Code
	-12004	XML contains invalid tag.	/ErrorResponse/Body/Status Code

Request	Code	External Message	Level
	-12012	Element not found in DTD	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12018	Unexpected element, missing element, or element out of sequence.	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12023	Undefined element	/PartnerManagerResponse/Body/Status/ValidationError/StatusCode
	-12027	Request should be sent to Transaction Service	/ErrorResponse/Body/StatusCode
FindAllDomains AssociatedWith Customer	209	Front-End Gateway temporarily unavailable	/ErrorResponse/Body/StatusCode

B.10 XML Status Codes

The code returned by the Partner Protocol uniquely identifies the type of code as either a status or an error. It can be used as a reference for the Status Code and Error Code tables below. A status code has a positive value and provides additional information regarding the status of a request. An error code has a negative value and provides additional information regarding the particular error(s) in the request.

Positive numbers returned by the Partner Protocol are referred to as Status Codes. The status codes are grouped numerically by status type. Each status code is accompanied by additional information found in the XML Description column of the Status Codes Table below.

B.10.1 Status Codes

Code	XML Description
209	Front-End Gateway temporarily unavailable
2000	Partner Authenticated
2001	Partner Authorized
2002	User Authenticated
2003	User Authorized
5700	User created successfully.
5701	User modified successfully.
5702	User lookup successful.
5703	User created successfully, but login name not available; please change later.
5704	User modified successfully, but login name not available; please change later.
5705	Login Name Available
5706	User lookup partially successful
7000	Order submitted successfully and is in pending status.
7001	Part of the requested order submitted and is being fulfilled.
7002	Order has been successfully fulfilled.
7003	Order has been partially fulfilled.
7004	Line item submitted to back office successfully.
7005	Order line item has completed.
7006	Order line item is pending.
7007	Order status lookup successful.
7012	Command completed successfully
7014	Product and all its children products consolidated
7015	Order status lookup partially successful.
11702	Product lookup successful.
11703	Product lookup partially successful

B.10.2 Error Codes

Negative numbers returned by the Partner Protocol are referred to as Error Codes. The error codes are grouped numerically by error type. Each error code is accompanied by additional information found in the XML Description column of the Error Codes Table below.

Code	XML Description
-200	System Unavailable.
-201	System Error
-205	System Error
-245	Data Validation Error
-750	Order submitted successfully but status is not available
-2000	Partner not Authenticated
-2001	Partner not Authorized
-2002	User not Authenticated
-2003	User not Authorized
-2004	User not Authorized
-2010	Partner exceeded usage limit. Please retry later.
-2019	User not Authenticated
-2019	User not Authenticated for Product
-2022	Invalid Authorized Agent
-2023	Invalid Authorized Agent
-2025	Transfer Failed – Invalid Auth Code
-2027	Transfer Failed – Missing Auth Code
-2304	Type is invalid
-2307	Validation error
-2312	Invalid domain
-2313	Invalid host
-2316	Invalid number format
-2317	Value not unique
-2319	Invalid value
-2320	Invalid line item. Validation error
-2320	Invalid value
-2323	This XML request has been retired
-3004	CNAME cannot be the same as base domain name
-5700	Invalid user type
-5706	Invalid postal code
-5715	User not found
-5719	City is required for US address
-5720	State is required for US address
-5721	Postal code is required for US address
-5735	User not a individual user
-5736	User not a business user
-5737	Login Name not available
-7012	Order was aborted
-7016	All products are invalid. Order was not submitted
-7017	Order(s) not found
-7020	Parent domain of host does not exist
-7021	Purchase period is invalid
-7022	Invalid preference type
-7023	Line item is valid. Order was not submitted

Code	XML Description
-7024	Line item is invalid. Order was not submitted
-7025	Order line item failed
-7027	Date is formatted incorrectly
-7028	Invalid date, Date must be later than Jan 1st, 1990
-7030	The requested action is not supported for the designated products(s)
-7032	Duplicate request property
-7033	Reference Number not owned by partner
-7034	Unknown request property
-7036	Reference Number invalid value
-7045	Another transfer request is already in process for this domain. Transfer process cannot be initiated.
-7046	Domain name is already registered with Network Solutions
-7047	Domain name is currently not registered and is available for purchase. It cannot be transferred
-7053	Command failed
-7054	FromCustomer and ToCustomer are the same
-7056	Customer name does not match
-7057	Request too large
-7058	Transfer Failed – AccountTransferLocked
-7059	Consolidation Failed – Account Transfer Lock
-7060	Error - Consolidation not allowed for (us de com.au net.au co.uk org.uk me.uk be at com.mx co.nz net.nz org.nz)
-8011	Domain does not exist
-8014	Unicode name failed during encoding or decoding
-8017	Domain not found
-8019	Unable to perform action. Incorrect product type
-8028	Resource records exceed max allowed
-8029	Domain in hold state
-8030	Domain in locked state
-8031	Name change or domain transfer pending
-8033	Domain in unpaid state
-8034	Domain in special state
-8037	Domain already owned by customer
-8038	Domain not registered with Network Solutions
-8039	Domain in expired state
-8042	Web Forwarding has not yet been added to this domain
-8044	Under Construction page is not valid
-8045	The domain already has Web Forwarding
-8046	Invalid Web Forwarding URL
-8047	Invalid Web Forwarding
-8048	Invalid Webforwarding with unsupported protocol
-8506	Host already exists
-8507	Host does not exist
-8508	Not enough hosts
-8509	Host not found
-8510	Duplicate hosts detected
-8516	No new data found to modify host.
-11000	Domain is not a parking product.
-11001	Unable to switch Under Construction Page
-11607	Domain already registered
-11636	Domain already registered
-11700	Product unpaid
-11702	Domain name does not have associated service.

Code	XML Description
-11708	Product is not renewable
-11709	Product is not reissueable
-11710	Product is not revocable
-11711	Product has not been issued
-12000	XML is malformed
-12001	XML is invalid
-12002	XML contains invalid root tag
-12003	XML contains invalid header
-12004	XML contains invalid tag
-12006	Incorrect value of attribute
-12008	Input data is longer than max length
-12012	Element not found in DTD
-12018	Unexpected element, missing element, or element out of sequence
-12022	Missing attribute or element
-12023	Undefined element
-12026	Request should be sent to Available Service
-12027	Request should be sent to Transaction Service
-11900	Partner does not have purchase agreement
-11901	Subscriber's agreement must be true for this action
-11902	Invalid server selection
-11903	Required fields are missing
-11904	CSR validation failed
-11905	CSR parsing failed
-11906	Domain parsing failed
-11907	Seal parsing failed
-11908	Invalid SSL product
-11909	Domain name does not match original product's
-11910	Domain name change is not allowed for repurchased product
-11911	Certificate not found
-11912	Failed to create bundled certificate
-11913	Domain name change is not allowed product has been issued
-11914	Reissue is not allowed, product must be in Provide CSR or Provide Domain/Validation status
-11915	Cannot override domain name or company name for Cert product
-11916	Seal code not found
-11917	Requested seal code type does not match product
-11918	Accuracy agreement must be true for this action.
-11919	EVSubscriber is required and must contain first and lastname.
-11920	Selected EntityType is not valid for this product
-11921	State is required if country code is US
-11922	Requested cert format does not match product code
-11923	Product type does not match the requested ProductName
-11924	Required fields are missing for SSL EVV
-11925	Required fields are missing for SSL EV Contacts

B.10.3 Validation Codes

The following validation codes will be returned if validation error occurs.

Code	XML Description
- 13900	Invalid Domain Name (applicable to SSL Product)
- 13901	Not a wild card domain (applicable to SSL Product)
- 13902	Invalid organization name (applicable to SSL Product)

Code	XML Description
- 13903	Invalid country code (applicable to SSL Product)
- 13904	Invalid IP Address (applicable to SSL Product)
- 13905	Invalid parent domain name (applicable to SSL Product)
- 13906	Invalid TLD
- 13907	Domain must have at least one dot
- 13908	Domain cannot start and end with a dot
- 13909	Is a wild card domain (applicable to SSL Product)
- 13910	Invalid key size
- 13911	Private IP Address not allowed (applicable to SSL Product)

Appendix C Country Codes

UNITED STATES, UNITED STATES FOREIGN PRESENCE DESIGNATION CODES

Location	Code	Location	Code
ALABAMA	AL	NEW HAMPSHIRE	NH
ALASKA	AK	NEW JERSEY	NJ
AMERICAN SAMOA	AS	NEW MEXICO	NM
ARIZONA	AZ	NEW YORK	NY
ARKANSAS	AR	NORTH CAROLINA	NC
CALIFORNIA	CA	NORTH DAKOTA	ND
COLORADO	CO	NORTHERN MARIANA ISLANDS	MP
CONNECTICUT	CT	OHIO	OH
DELAWARE	DE	OKLAHOMA	OK
DISTRICT OF COLUMBIA	DC	OREGON	OR
FEDERATED STATES OF MICRONESIA	FM	PALAU	PW
FLORIDA	FL	PENNSYLVANIA	PA
GEORGIA	GA	PUERTO RICO	PR
GUAM	GU	RHODE ISLAND	RI
HAWAII	HI	SOUTH CAROLINA	SC
IDAHO	ID	SOUTH DAKOTA	SD
ILLINOIS	IL	TENNESSEE	TN
INDIANA	IN	TEXAS	TX
IOWA	IA	UTAH	UT
KANSAS	KS	VERMONT	VT
KENTUCKY	KY	VIRGIN ISLANDS	VI
LOUISIANA	LA	VIRGINIA	VA
MAINE	ME	WASHINGTON	WA
MARSHALL ISLANDS	MH	WEST VIRGINIA	WV
MARYLAND	MD	WISCONSIN	WI
MASSACHUSETTS	MA	WYOMING	WY
MICHIGAN	MI	ARMED FORCES AFRICA	AE
MINNESOTA	MN	ARMED FORCES AMERICAS	AA
MISSISSIPPI	MS	ARMED FORCES CANADA	AE
MISSOURI	MO	ARMED FORCES EUROPE	AE
MONTANA	MT	ARMED FORCES MIDDLE EAST	AE
NEBRASKA	NE	ARMED FORCES PACIFIC	AP
NEVADA	NV		

FOREIGN COUNTRY DESIGNATION CODES

Location	Code	Location	Code
United States	US	Anguilla	AI
Afghanistan	AF	Antarctica	AQ
Albania	AL	Antigua And Barbuda	AG
Algeria	DZ	Argentina	AR
American Samoa	AS	Armenia	AM
Andorra	AD	Aruba	AW
Angola	AO	Australia	AU
Austria	AT	Egypt	EG
Azerbaijan	AZ	El Salvador	SV
Bahamas	BS	England	GB
Bahrain	BH	Equatorial Guinea	GQ
Bangladesh	BD	Eritrea	ER
Barbados	BB	Estonia	EE
Belarus	BY	Ethiopia	ET
Belgium	BE	Falkland Islands (Malvinas)	FK
Belize	BZ	Faroe Islands	FO
Benin	BJ	Fiji	FJ
Bermuda	BM	Finland	FI
Bhutan	BT	France	FR
Bolivia	BO	French Guiana	GF
Bosnia Hercegovina	BA	French Polynesia	PF
Botswana	BW	French Southern Territories	TF
Bouvet Island	BV	Gabon	GA
Brazil	BR	Gambia	GM
British Indian Ocean Territory	IO	Georgia	GE
Brunei Darussalam	BN	Germany	DE
Bulgaria	BG	Ghana	GH
Burkina Faso	BF	Gibraltar	GI
Burundi	BI	Great Britain	GB
Byelorussian SSR	BY	Greece	GR
Cambodia	KH	Greenland	GL
Cameroon	CM	Grenada	GD
Canada	CA	Guadeloupe	GP
Cape Verde	CV	Guam	GU
Cayman Islands	KY	Guatemala	GT
Central African Republic	CF	Guernsey	GG
Chad	TD	Guinea	GN
Chile	CL	Guinea-Bissau	GW
China	CN	Guyana	GY
Christmas Island	CX	Haiti	HT
Cocos (Keeling) Islands	CC	Heard and McDonald Islands	HM
Colombia	CO	Honduras	HN
Comoros	KM	Hong Kong	HK
Congo	CG	Hungary	HU
Congo, The Democratic Republic Of	CD	Iceland	IS
Cook Islands	CK	India	IN
Costa Rica	CR	Indonesia	ID
Cote D'Ivoire	CI	Iran (Islamic Republic Of)	IR

Location	Code	Location	Code
Croatia	HR	Iraq	IQ
Cuba	CU	Ireland	IE
Cyprus	CY	Isle Of Man	IM
Czech Republic	CZ	Israel	IL
Czechoslovakia	CS	Italy	IT
Denmark	DK	Jamaica	JM
Djibouti	DJ	Japan	JP
Dominica	DM	Jersey	JE
Dominican Republic	DO	Jordan	JO
East Timor	TP	Kazakhstan	KZ
Ecuador	EC	Kenya	KE
Kiribati	KI	Oman	OM
Korea, Democratic People's Republic Of	KP	Pakistan	PK
Korea, Republic Of	KR	Palau	PW
Kuwait	KW	Panama	PA
Kyrgyzstan	KG	Papua New Guinea	PG
Lao People's Democratic Republic	LA	Paraguay	PY
Latvia	LV	Peru	PE
Lebanon	LB	Philippines	PH
Lesotho	LS	Pitcairn	PN
Liberia	LR	Poland	PL
Libyan Arab Jamahiriya	LY	Portugal	PT
Liechtenstein	LI	Puerto Rico	PR
Lithuania	LT	Qatar	QA
Luxembourg	LU	Reunion	RE
Macau	MO	Romania	RO
Macedonia	MK	Russian Federation	RU
Madagascar	MG	Rwanda	RW
Malawi	MW	Saint FirstNamea	SH
Malaysia	MY	Saint Kitts And Nevis	KN
Maldives	MV	Saint Lucia	LC
Mali	ML	Saint Pierre and Miquelon	PM
Malta	MT	Saint Vincent and The Grenadines	VC
Marshall Islands	MH	Samoa	WS
Martinique	MQ	San Marino	SM
Mauritania	MR	Sao Tome and Principe	ST
Mauritius	MU	Saudi Arabia	SA
Mayotte	YT	Senegal	SN
Mexico	MX	Seychelles	SC
Micronesia, Federated States Of	FM	Sierra Leone	SL
Moldova, Republic Of	MD	Singapore	SG
Monaco	MC	Slovakia	SK
Mongolia	MN	Slovenia	SI
Montserrat	MS	Solomon Islands	SB
Morocco	MA	Somalia	SO
Mozambique	MZ	South Africa	ZA
Myanmar	MM	South Georgia and The Sandwich Islands	GS
Namibia	NA	Spain	ES
Nauru	NR	Sri Lanka	LK

Location	Code	Location	Code
Nepal	NP	Sudan	SD
Netherlands	NL	Suriname	SR
Netherlands Antilles	AN	Svalbard and Jan Mayen Islands	SJ
Neutral Zone	NT	Swaziland	SZ
New Caledonia	NC	Sweden	SE
New Zealand	NZ	Switzerland	CH
Nicaragua	NI	Syrian Arab Republic	SY
Niger	NE	Taiwan, Province Of China	TW
Nigeria	NG	Tajikista	TJ
Niue	NU	Syrian Arab Republic	SY
Norfolk Island	NF	Syrian Arab Republic	SY
Northern Mariana Islands	MP	Taiwan, Province Of China	TW
Norway	NO	Tajikista	TJ
Tanzania, United Republic Of	TZ	USSR	SU
Thailand	TH	Uzbekistan	UZ
Togo	TG	Vanuatu	VU
Tokelau	TK	Vatican City State	VA
Tonga	TO	Venezuela	VE
Trinidad and Tobago	TT	Vietnam	VN
Tunisia	TN	Virgin Islands (British)	VG
Turkey	TR	Virgin Islands (U.S.)	VI
Turkmenistan	TM	Wallis and Futuna Islands	WF
Turks and Caicos Islands	TC	Western Sahara	EH
Tuvalu	TV	Yemen, Republic of	YE
Uganda	UG	Yugoslavia	YU
Ukraine	UA	Zaire	ZR
United Arab Emirates	AE	Zambia	ZM
United Kingdom	GB	Zimbabwe	ZW
United States	US		
United States Minor Outlying Islands	UM		
Uruguay	UY		

Appendix D XML Default Field Validation Rules

The first table gives the default field validation rules. If there are TLD specific field validation rules, those will override the default field validation rules. The TLD specific field validation rules are listed in the table following the default values.

XML Default Field Validation Rules Table

Field	Min /Max Length	Regular Expression	Sample Values
AddressLine1	1/41	All alphanumeric characters plus: ., ' # * @ / & + - and space PATTERN=^[-. ,/#@+&\'* a-zA-Z0-9]+\$	Regional Office
AddressLine2	1/41	All alphanumeric characters plus: ., ' # * @ / & + - and space PATTERN=^[-. ,/#@+&\'* a-zA-Z0-9]+\$	20/F Citicorp Center
AddressLine3	1/41	All alphanumeric characters plus: ., ' # * @ / & + - and space PATTERN=^[-. ,/#@+&\'* a-zA-Z0-9]+\$	18 Whitfield Road
AddressLine4	1/41	All alphanumeric characters plus: ., ' # * @ / & + - and space PATTERN=^[-. ,/#@+&\'* a-zA-Z0-9]+\$	Causeway Bay
AddressLine5	1/41	1-41 alphanumeric characters plus: ., ' # * @ / & + - and space PATTERN=^[-. ,/#@+&\'* a-zA-Z0-9]+\$	Hong Kong
AuthAnswer	1/255	All printable characters	Green
AuthCode	6/16	All printable characters	a@c#pad
AuthQuestion	1/255	All printable characters	What color is Kermit the Frog?
AutoRenew	1/1	Valid Values: True or False in any case; Default value is False. PATTERN=^([Tt][Rr][Uu][Ee]) ([Ff][Aa][Ll][Ss][Ee])\$	TRUE
BulkWhoisOptOut	1/1	Valid Values: "Y", "y", "N", "n" Default value is "N". PATTERN=^[yY][nN]\$	Y N
CanonicalName		See Hostname	wormhole.movie.edu
City	1/30	All alphanumeric characters plus: /-, ' and space PATTERN=^[/-. , a-zA-Z0-9]+\$	Los Angeles
CompanyName	1/50	All alphanumeric characters plus: -, ()V@&! and space PATTERN=^[a-zA-Z0-9-.,()V@&!]+\$	Network Solutions, Inc.
Comments		All printable characters	Creating new customer
CountryCode	2/3	All alpha characters (see also Foreign Country Designation Code in) PATTERN=^[a-zA-Z]+\$	US
CustomerID	1/10	All numeric characters PATTERN=^[0-9]+\$	15574315
Date		ISO 8601 Standard format: YYYY-MM-DDThh:mm:ssTZD	2001-10-30T17:36:54EST
Domain	4/67	All alphanumeric characters plus: - PATTERN=^[a-zA-Z0-9-]+\$	foo.bar

Field	Min /Max Length	Regular Expression	Sample Values
DomainName	2/67	All alphanumeric characters plus: -. PATTERN=^[a-zA-Z0-9-]+\$	Network Solutions.com
Email	7/128	All alphanumeric characters plus: ! % & = \ ' + _ . - followed by a @ then a domain name (see also DomainName validation) PATTERN=^[!%&=\'+_a-zA-Z0-9-]@\([a-zA-Z0-9-]\.\)+([a-zA-Z-])+\$	internetsales@Network Solutions.com
EmailAddress	7/128	All alphanumeric characters plus: ! % & = \ ' + _ . - followed by a @ then a domain name (see also DomainName validation) PATTERN=^[!%&=\'+_a-zA-Z0-9-]@\([a-zA-Z0-9-]\.\)+([a-zA-Z-])+\$	foo@bar.name
Exchange	4/128	All alphanumeric characters plus: -. See also RFC 1035, page 17 PATTERN=^[a-zA-Z0-9-]+\$	foo.bar.net
Fax	1/25	PATTERN=^[.,+() / a-zA-Z0-9-*] +\$	703-555-1212
FirstName	1/30	PATTERN=^[a-zA-Z0-9-.,()\@&'] +\$	Peter
ForwardingEmailAddress	6/128	All alphanumeric characters plus: ! % & = \ ' + _ . - followed by a @ then a domain name (see also DomainName validation) PATTERN=^[!%&=\'+_a-zA-Z0-9-]@\([a-zA-Z0-9-]\.\)+([a-zA-Z-])+\$	foobar@Network Solutions.com
GenerateDomains	6/67	All alphanumeric characters plus: -. PATTERN=^[a-zA-Z0-9-]+\$	testOnline
HostIP		See IPAddress	216.168.252.86
HostIP1			
HostIP2			
HostIP3			
HostIP4			
HostIP5			
HostName	2/80	All alphanumeric characters plus: . _ - May not start or end in a dash. Please see .name table below for more validation rules for .name HostNames. Cannot have two constructive dots like “..” PATTERN=^[a-zA-Z0-9-] +\$	h33.foobar.com
IPAddress	7/15	An IP Address shall contain exactly 3 dots. Cannot start or end with dot. An IP Address shall not have two or more dots in sequence. Each node in IP address shall contain between 1 and 3 numeric characters The numeric value of each node in an IP Address shall be between 0 and 255. IP Addresses on a hostname min number is 1 For .INFO min count is 1 and max count is 13 For .BIZ min count is 1 and max count is 1 For .US min count is 1 and max count is 1	216.168.252.86

Field	Min /Max Length	Regular Expression	Sample Values
LastName	1/30	All alphanumeric characters plus: -,()\@&' and space PATTERN=^[a-z A-Z0-9-.,()\@&']+\$	Smith
LegalContactFirstName	1/30	All alphanumeric characters plus: -,()\@&' and space PATTERN=^[a-z A-Z0-9-.,()\@&']+\$	Edward
LegalContactLastName	1/30	All alphanumeric characters plus: -,()\@&' and space PATTERN=^[a-z A-Z0-9-.,()\@&']+\$	Johnson
LegalContactMiddleName	1/30	All alphanumeric characters plus: -,()\@&' and space PATTERN=^[a-z A-Z0-9-.,()\@&']*+\$	Brandon
LoginName	8/32	All alphanumeric characters plus: .-@ PATTERN=^[a-zA-Z0-9@.-_&,'/]+\$	eJohnson22
LongDate		Number of milliseconds since January 1, 1970, 00:00:00 GMT	1004482456262
MiddleName	1/30	All alphanumeric characters plus: -,()\@&'* and space PATTERN=^[a-z A-Z0-9-.,()\@&'*]+\$	Andrew
NewLoginName	1/32	All alphanumeric characters plus: .-@ PATTERN=^[.a-zA-Z0-9@]+)\$	eJohnson23
NewPassword	4/8	All Printable PATTERN=^[^\S]{4,8}\$	aSecret2
NicHandle	1/32	All alphanumeric characters plus: -* and space PATTERN=^[a-zA-Z0-9-]*+\$	LXB244
OrderID	1/10	All numeric characters PATTERN=^[0-9]+\$	49161682
PartnerID	1/10	All numeric characters PATTERN=^[0-9]+\$	1471234
PartnerPassword	4/8	All Printable PATTERN=^[^\S]{4,8}\$	aSecret5
Password	4/8	All Printable PATTERN=^[^\S]{4,8}\$	aSecret6
Phone	1/25	1-25 alphanumeric characters plus: .,+()/- and space PATTERN=^[.,+()/- a-zA-Z0-9-]*+\$	703-555-1212
PostalCode	US: 5/10 Non-US: 1/15	US address: 9 numeric and – PATTERN=^[0-9]{5}\$^[0-9]{5}-[0-9]{4}\$ excluding codes starting from 00001 to 00004 and 99996 to 99999 Non-US address: 1-15 alphanumeric plus: -* and space PATTERN=^[a-z A-Z0-9-]*+\$ PATTERN=^[()\\.,/#;+&' a-zA-Z0-9-]*+\$	20170-5139 20171 106-0041
Preference	1/2	All Numeric PATTERN=^[0-9]+\$	20
ProductID	1/32	All alphanumeric characters plus: . PATTERN=^[.a-zA-Z0-9-]*+\$	WN.D.68651332

Field	Min /Max Length	Regular Expression	Sample Values
PurchasePeriod	1/2	All Numeric PATTERN=^[0-9]+\$	10
RecordName	4/128	See hostname	foobar.net
ReferenceNumber	1/10	All numeric PATTERN=^[0-9]+\$	29543
State	1/40	Required for US Address Must be a valid US State. PATTERN=^[-, ' a-zA-Z0-9]+\$	FL Florida Armed Forces Europe
TLD	Varies by TLD	All alpha PATTERN=^[a-zA-Z]+\$	com
UserID	1/9	All numeric PATTERN=^[0-9]+\$	15534713
VerifyDomainName		See DomainName	foobar.org
VerifyDomains			foobar
VerifyDomainsAll			

Additional Validation Rules for Specific TLDs

Field	Validation Rules	Sample Values
NAME		
DomainName	All DomainName validation rules from above apply, plus: -The DomainName must have exactly 3 nodes with a dot between each node. -Last name cannot be all numeric. -Last name must be at least 3 characters.	foo.bar.name
HostName	All HostName validation rules from above apply, plus: -The HostName must have at least 3 nodes.	ns.foo.bar.name
CC		
DomainName	All DomainName validation rules from above apply, plus: -The DomainName cannot be a combination of all numeric and dashes.	foo.cc

Override Default Validation Rules for Specific TLDs

Field	Min /Max Length	Regular Expression	Sample Values
INFO			
DomainName	8/67	PATTERN=^[a-zA-Z.0-9]+\$	Example.info
WS			
DomainName	7/67	PATTERN=^[a-zA-Z.0-9]+\$	Example.ws
ORG.UK			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
DomainName	10/67	PATTERN=^[a-zA-Z.0-9]+\$	Example.org.uk
CO.UK			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
DomainName	9/67	PATTERN=^[a-zA-Z.0-9]+\$	Example.co.uk
ME.UK			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
DomainName	9/67	PATTERN=^[a-zA-Z.0-9]+\$	Example.me.uk
COM.MX			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74

Field	Min /Max Length	Regular Expression	Sample Values
Phone	1/25	PATTERN= ^(52\s{0,1}\d{2,3}\s{0,1}\d{3}\s{0,1}\d{4})\$	
State	1/40	Must be valid MX state	Sinaloa
DomainName	10/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.com.mx
DE			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
PostalCode	5/5	PATTERN=^[0-9 -]+\$	12345
Phone	1/25	PATTERN= ^\(+49\s{0,1}\d{2}\s{0,1}\d{4}\s{0,1}\d{4})\$	
DomainName	6/66	PATTERN=^[a-zA-Z.0-9-]+\$	Example.de
AT			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
DomainName	6/66	PATTERN=^[a-zA-Z.0-9-]+\$	Example.at
BE			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
Phone	1/25	PATTERN=^\+(\d){11,15}\$	
DomainName	6/66	PATTERN=^[a-zA-Z.0-9-]+\$	Example.be
CO.NZ, NET.NZ, ORG.NZ			
IP Address	7/15	Maximum of 1 IP Address	216.168.225.74
DomainName	9/66	PATTERN=^[a-zA-Z.0-9-]+\$	Example.co.nz
MOBI			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	8/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.mobi
US.COM UK.COM EU.COM DE.COM, BR.COM, CN.COM, GB.COM, GB.NET, HU.COM, JPN.COM, KR.COM, NO.COM, QC.COM, RU.COM, SA.COM, SE.COM, SE.NET, UK.NET, UY.COM, ZA.COM , COM.ES, ORG.ES, NOM.ES, NET.IN, ORG.IN, GEN.IN, IND.IN, AR.COM			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	10/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.us.com
LA IN ES			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	6/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.la
AM FM			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	10/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.am
CO.IN			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	9/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.co.in
FIRM.IN			
IP Address	2/13	Maximum of 1 IP Address	216.168.225.74
DomainName	11/67	PATTERN=^[a-zA-Z.0-9-]+\$	Example.firm.in

Appendix E Common Data Elements

Field	Required	Multiple	Notes
Address			
AddressLine1	Required	NO	Only AddressLine1 is required.
AddressLine2	Optional	NO	AddressLine2-5 may be used for those addresses requiring additional space.
AddressLine3	Optional	NO	
AddressLine4	Optional	NO	
AddressLine5	Optional	NO	
City	Conditional	NO	City is only required when the country code is US.
State	Conditional	NO	State is only required when the country code is US.
PostalCode	Conditional	NO	Postal code is only required when the country code is US.
CountryCode	Required	NO	See Appendix D for a list of valid country codes.
AdminContact			
UserID	Conditional	NO	Either UserID or NicHandle must be used to identify AdminContact.
NicHandle	Conditional	NO	
AuthorizedAgent			
SystemUser	Required	NO	This defines who is issuing the Request.
UserID	Conditional	NO	Either UserID or NicHandle must be used to identify AuthorizedAgent.
NicHandle	Conditional	NO	
Password	Required	NO	User password.
SystemUserType	Required	NO	Type of user being authorized.
customer	Conditional	NO	The SystemUserType must be either customer or partner.
partner	Conditional	NO	
BillContact			
UserID	Conditional	NO	Either UserID or NicHandle must be used to identify BillContact.
NicHandle	Conditional	NO	
Customer			
UserID	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
NicHandle	Conditional	NO	
Password	Required	NO	
DateRange			
BeginDate	Required	NO	
Date	Conditional	NO	Either Date or LongDate must be provided. Date is ISO 8601 standard format: YYYY-MM-DDThh:mm:ssTZD
LongDate	Conditional	NO	Either Date or LongDate must be provided. LongDate is the number of milliseconds since January 1, 1970, 00:00:00 GMT.
EndDate	Required	NO	

Field	Required	Multiple	Notes
Date	Conditional	NO	Either Date or LongDate must be provided. Date is ISO 8601 standard format: YYYY-MM-DDThh:mm:ssTZD
LongDate	Conditional	NO	Either Date or LongDate must be provided. LongDate is the number of milliseconds since January 1, 1970, 00:00:00 GMT.
DomainProtect			This element is passed into an OrderRequest as an attribute of a domain product to indicate that the domain should be protected against outbound transfers. If this flag is not provided then it will default to TRUE. For any domain product for which this attribute is TRUE, any outbound transfer requests will be denied. In order to allow an outbound transfer this attribute must be set or modified to FALSE. This element is also returned by product lookup requests to indicate the current setting.
LegalLock			This element is returned by product lookup requests to indicate whether or not the domain is in a legal lock status. The possible values are TRUE or FALSE.
FromCustomer			
UserID	Required	NO	Customer from which the product is being transferred.
LegalContact			Businesses have a legal contact Network Solutions uses to contact the business.
FirstName	Required	NO	
MiddleName	Optional	NO	
LastName	Required	NO	
Address	Optional	NO	If Address is not provided, it defaults to the business address.
Phone	Optional	NO	If Phone is not provided, it defaults to the business phone.
Fax	Optional	NO	If Fax is not provided, it defaults to the business fax.
Email	Optional	NO	If Email is not provided, it defaults to the business email.
Host			This element is used when modifying a hosts properties using ModifyHost in OrderRequest. It is also used when creating or modifying the host information for a domain as part of an OrderRequest.
HostName	Conditional	NO	
HostIP1	Optional	NO	IP addresses associated with the specified host.
HostIP2	Optional	NO	
HostIP3	Optional	NO	

Field	Required	Multiple	Notes
HostIP4	Optional	NO	
HostIP5	Optional	NO	
HostData			This element is returned in response to LookupHost in LookupRequest. It is also returned as part of the Domain element in response to a LookupDomain in LookupRequest.
HostName	Required	NO	IP addresses associated with the specified host.
HostIP1	Optional	NO	
HostIP2	Optional	NO	
HostIP3	Optional	NO	
HostIP4	Optional	NO	
HostIP5	Optional	NO	
Hosts			Max of 4 hosts allowed, must have at least one for domain to resolve.
Host1	Optional	NO	
Host2	Optional	NO	
Host3	Optional	NO	
Host4	Optional	NO	
NewHost			This element is used when creating a new host using CreateHost in OrderRequest.
Hostname	Required	NO	IP addresses associated with the specified.
HostIP1	Optional	NO	
HostIP2	Optional	NO	
HostIP3	Optional	NO	
HostIP4	Optional	NO	
HostIP5	Optional	NO	
Partner	Required	NO	One of the "Conditional" elements in this table must be submitted.
UserID	Conditional	NO	
NicHandle	Conditional	NO	
Password	Required	NO	
RequestHeader			Required for all Requests except AvailableRequest.
VERSION_4_6	Required	NO	This defines the version of the DTD and is used for routing the request.
Authentication	Required	NO	
PartnerID	Required	NO	
PartnerPassword	Required	NO	
Comments	Optional	NO	An arbitrary string that may be passed in and will be returned in the Comments element of the Response.
SystemInfo	Optional	NO	For Network Solutions internal use only. No input from Partner required.
RequestProperty	Optional	YES	
PropertyName	Required	NO	Name of a supported request property.
PropertyValue	Required	NO	Value of the named request property.
ResourceRecords			

Field	Required	Multiple	Notes
A	Optional	YES	The DNS A record for applicable domains.
RecordName	Required	NO	This is the RecordName of the A record being added.
IPAddress	Required	NO	This is the IPAddress being added.
CNAME	Optional	YES	The DNS CNAME record for applicable domains.
RecordName	Required	NO	This is the RecordName of the CNAME record being added.
CanonicalName	Required	NO	This is the Canonical Name where the above reference resolves.
MX	Optional	YES	The DNS MX record for applicable domains.
RecordName	Required	NO	This is the RecordName of the MX record.
Preference	Required	NO	This is the Preference of the mail exchange. Lower values are used first.
Exchange	Required	NO	This is the Mail Exchange to forward all mail sent to the RecordName.
SupplementalRegistryData			Additional registrant/owner data to be used when registering domains in some domain name registries. This supplemental data may alternatively be provided with an OrderRequest to purchase a domain.
DotUSSupplementalData	Optional	NO	Additional registrant data associated with the .us registry. This data is required to be set for any Individual or Business that will be a registrant/owner of a domain in the .us TLD. This supplemental data can be provided with UserRequests when creating or modifying a User or with OrderRequests when purchasing a domain in the .us TLD.
CustomerType	Required	NO	Classification of the registrant of a .us domain.
PersonUSCitizen	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
PersonPermanentUSResident	Conditional	NO	
OrganizationIncorporatedInUS	Conditional	NO	
OrganizationWithActiveUSPresence	Conditional	NO	
OrganizationWithUSPresence	Conditional	NO	
AnticipatedUsage	Required	NO	Probable usage of one or more of the .us domains owned by the registrant.
ForProfit	Conditional	NO	One of the "Conditional" elements in this table must be submitted.
NotForProfit	Conditional	NO	
Personal	Conditional	NO	
Educational	Conditional	NO	
Governmental	Conditional	NO	
DotDESupplementalData	Optional	NO	Additional registrant data associated with the .de registry. This data may be set for any Individual or Business that will be a registrant/owner of a domain in the .de TLD. This supplemental data can be provided with UserRequests when

Field	Required	Multiple	Notes
			creating or modifying a User or with OrderRequests when purchasing a domain in the .de TLD.
UseDefaultRegistrantAddress	Required	NO	A boolean (True False) flag indicating that the registrant/owner of a .de domain wishes to use a default, Network Solutions address in Germany as the registrant/owner address when registering the .de domain with the .de registry. The .de registry requires that the registrant/owner address be an address in Germany. Using a default German address will allow registrants/ owners to register .de domains without having their own German address. If set to True, all subsequent registrations of .de domains for this customer will use the default German registrant address. If set to False, all subsequent registrations of .de domains for this customer will use the customers address (which must be a German address or the registration will fail). Changes to this preference will not affect any previously register .de domains.
ResponseHeader			Required for all Responses.
VERSION_4_6	Required	NO	This defines the version of the DTD and is used for routing the request.
ReferenceNumber	Required	NO	A unique ID returned by Network Solutions.
TimeStamp	Required	NO	TimeStamp is ISO8601 standard format: YYYY-MM-DDThh:mm:ssTZD
Comments	Optional	NO	
TechContact			
UserID	Conditional	NO	Either UserID or NicHandle must be used to identify TechContact.
NicHandle	Conditional	NO	
ToCustomer			The customer to which a product will be transferred.
UserID	Required	NO	
Status			
StatusCode	Required	NO	
Description	Required	NO	
ValidationError	Optional	YES	If the request has validation errors, they are listed here.
PathName	Required	NO	The fully qualified path to an element in an XML request that failed a validation rule. Notation adheres to the XQL standard.
StatusCode	Required	NO	Numeric value identifying a specific validation error.
Description	Required	NO	A readable description of the specific

Field	Required	Multiple	Notes
			validation error.

Appendix F Server Selection Software for Secured Socket Layer Products

Server Selection ID	ServerSoftware Name
1	AOL
2	Apache-SSL (Ben-SSL, not Stronghold)
3	Apache/ModSSL
4	C2Net Stronghold
5	Cobalt Raq
6	Covalent Server Software
7	Ensim
8	H-Sphere
9	IBM HTTP Server
10	IBM Internet Connection Server
11	Java Web Server (Javasoft / Sun)
12	Lotus Domino
13	Lotus Domino Go!
14	Microsoft IIS 1.x to 4.x
15	Microsoft IIS 5.x and later
16	Netscape Enterprise Server
17	Netscape FastTrack
18	Novell Web Server
19	OTHER
20	Oracle
21	Plesk
22	Quid Pro Quo
23	R3 SSL Server
24	Raven SSL
25	RedHat Linux
26	SAP Web Application Server
27	Tomcat
28	WHM/cPanel
29	WebStar 4.x and later
30	WebTen (from Tenon)
31	Website Professional
32	Zeus Web Server
33	iPlanet