Backorder Reference

Version 1.0
July 4, 2011

DISCLAIMER AND LIMITATION OF LIABILITY

We made efforts to ensure the accuracy and completeness of all information in this document. However, we make no warranties of any kind (whether express, implied or statutory) with respect to the information herein. We assume no liability to any party for loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, we assume 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. 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.

We reserve the right to make changes to any information herein without further notice.

NOTICE AND CAUTION concerning Patent or Trademark Rights

The inclusion in this document, the associated on-line file, or the associated software of any information covered by any patent, trademark, or service mark rights shall not constitute nor imply a grant of, or authority to exercise, any right or privilege protected by such patent, trademark, or service mark. All such rights and privileges 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.

Backorder Manual 1.0
July 4, 2011

Table of Contents

1 INTRODUCTION

Domain backordering is one of the fastest growing and most lucrative sectors in the domain industry. With thousands of value domains expiring every day, the only way to obtain the most value of these is through a backorder system.

This Quickstart guide will explain the definitions, process-flows, billing aspects, and API details for a reseller to integrate with the system. Please refer to this manual for all backorder related issues.

1.1 What are Domain Backorders

Everyday tens of thousands of previously registered domains become available for general registration again, once the registry controlling the domain name releases it (known as a drop, dropped, or deleted domain). Many of these dropping domains are very valuable. To increase the chances of obtaining domains that drop, a backorder can be used.

A domain backorder triggers a registration system to attempt to register a domain name the instant it becomes generally available from the registry. The backorder system monitors dropping domains and in less than a second of its availability, the system attempts to register the domain on behalf of a client who placed a backorder request. Typically backorder requests have to be placed at least 24 hours prior to the drop date and time.

Though a backorder system increases the chances of obtaining a dropping domain name, there is no guarantee. Typically the registration system competes with other registration systems for those expiring domains that are valuable.

1.2 Prepaid Deposits and Refunds for Backorders

When a backorder is successful, this means the registration system was able to register the domain on behalf of a client. And since there is always a possibility that any backorder will result in a domain registration, three (3) days before the deletion date of a domain, funds are reserved as a deposit.

Please ensure account balances are funded to cover all pending backorders otherwise some or all backorder requests will not be processed.

If the registration system is unable to register a backordered domain, reserved deposits are then automatically returned to their respective account balances.

1.3 Higher Registration Costs for Domain Backorders

Registration of a domain through a backorder system requires the use of additional systems and technology. As a result, whenever a backorder is successfully registered the cost of the registration is typically higher. As always, it is free to place a backorder and a payment is only required if a backorder is successfully registered.

Registration of a domain through a backorder is for the minimum registration period, which is one (1) year for most TLDs and two (2) years of more for specialty TLDs.

Registration period included in successful backorder (examples):
- .COM, .NET, .ORG includes one (1) year registration
- .CO.UK includes two (2) year registration

1.4 Multiple Domain Backorders for the Same Domain Name

When two or more clients backorder the same domain name and the domain is successfully registered, the domain is then auctioned off to the highest bidder via a private auction at PremiumSale.com. Please note that a private auction means that only PremiumSale.com will only invite the original clients, who placed a backorders.

1.5 PremiumSale.com for Domain Auctions

PremiumSale.com manages all invitations, registrations, auction proceedings, payment, and even notifications for the reseller. For more information on private auctions and on our auction partner company please visit www.premiumsale.com.

1.6 Processing of a Domain Backorder After Successful Registration

Domain backorders with a single request that are successfully registered by the backorder system, will be placed in the reseller's account automatically. Those backorders requiring an auction for settling ownership, the domain will be placed in the reseller's account or the reseller's customer at the conclusion of the auction.

2 BACKORDER APPLICATION WORKFLOW AND PROCESS

Backordering for any domain name is allowed for any TLD as long as the respective TLD supports the backorder function.

The relation - PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION must be defined.

2.1 Finding Backorder Domain Names

Most TLDs that support backordering have lists of domain name that are about to drop. These lists are called the 'pending delete lists' and can be queried using the QueryDomainPendingDeleteList command.

COMMAND - QueryDomainPendingDeleteList

2.1.1 Pending Delete Filtering and Scoring System

With tens of thousands of domains becoming available everyday, the list of pending deletes is often too large to manually process. The API provides functionality to filter and sort pending delete domains by a basic keyword scoring system. Please note that the score is only a hint of how close it matches keyword filters and does not provide reference to a domain's value. Use of the tools is as is without any guarantee.

2.1.2 Pending Delete Filtering and Scoring System

The full list of deleting domains supported by the system can also be downloaded here:

http://www.hexonet.net/files/domain-backordering/pending_delete_domain_list.csv.zip

2.1.3 TLDs without Pending Delete Lists and Scoring System

Some TLDs, for example like .DE, do not have a pending delete list. However, backorders can still be placed for these TLDs. Please contact support to find out which TLDs, without pending delete lists, can accept a backorder application.

Domains that are not expiring in near future can also be backordered. Obviously, processing of these backorders will not commence until they expire and are ready to be deleted by their respective registry.

2.2 Creating a New Backorder Application

To create a new backorder application, simply use the command below:

COMMAND - AddDomainApplication

Parameters used for this command are very similar to the parameters used for creating a new registration via AddDomain, however, in the case of an backorder application, most parameters are optional as well as the periods are ignored. The backorder system always uses the minimum period for registrations.

RECOMMENDATION:
Please provide at least a valid OWNERCONTACT to the AddDomainApplication command some TLDs don't support automated ownerchanges, which can result in additional fees or additional forms. Additionally, if an application goes to an auction, the provided contact information will be used by the auction provider to send out private the auction invitations as well as preload an auction account.

A newly created backorder application will have the status REQUESTED.

RESPONSE - REQUESTED

2.3 Securing Deposits for Backorder Application

Three days before a domain is about to drop (FINALIZATIONDATE) the system will attempt to secure a deposit for the backorder. This deposit is defined by the following relation:

The relation - PRICE_CLASS_DOMAIN_TLD_BACKORDER_DEPOSIT

2.3.1 Automated Deposit Refunds

- Backorder application is deleted before its finalization date (drop date)
- Backorder fails
- Backorder goes into auction

2.3.2 Unable to Secure Deposit

If the user's account has insufficient funds or credit to secure a deposit for upcoming backorder applications, then the backorder application is completely ignored and the system will not try to register the domain name.

If a deposit is successfully secure, the backorder application status will change

STATUS - ACTIVE

2.4 Processing Backorder Applications

On the finalization date or drop date, the status of the Backorder Application changes from ACTIVE to PROCESSING. This status is only visible for a couple of hours depending on the TLD and the respective registry's drop procedures.

STATUS - PROCESSING

Backorder Applications that had insufficient funds in the reseller's account, will not have an ACTIVE status and thus will not be processed on the finalization date.

2.5 Backorder Application Failed

Failed Backorder Applications are a result of either the domain not being processed for insufficient funds or if the domain was registered by another backorder system. Within seven (7) days the Backorder Application will automatically be deleted from the system. Additionally, all deposits secured for the domain application are refunded automatically.

EVENT - DOMAINAPPLICATION :: BACKORDER_FAILED

2.6 Single Backorder Application Successful

When only one Backorder Application is present and the respective domain is successfully registered, the domain is immediately assigned to the account for the application object (application account). Then the application account is charged according to the relation PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION.

STATUS – SUCCESSFUL RELATION - PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION EVENT - BACKORDER_SUCCESSFUL

2.7 Multiple Backorder Application Successful

When multiple Backorder Applications are made for the same domain and the respective domain is also successfully registered, the domain will then be assigned to a private auction for deciding who gets the backordered domain. All reserved backorder deposits are automatically refunded to the individual user accounts.

STATUS – AUCTION-PENDING EVENT - DOMAINAPPLICATION :: BACKORDER_AUCTION_PENDING

2.8 Auctions via Auction Provider PremiumSale.com

All backorder auctions are managed by the online auction systems at PremiumSale.com and are all private, meaning only the original users who placed a backorder for the auctioned domain will be invited (closed to

everyone else). All invitations, auctioning, and payment processing is handled by PremiumSale.com seamlessly.

Resellers and sub-Resellers are able to resell services as the auction platform and processes are not only fully automated but also white-labeled.

Auction Information:

Auctions start at 20:00 UTC on the same day the domain is dropped. Auctions which are scheduled to start after 19:45 UTC on a given day will be rescheduled to start at 20:00 UTC on the next day. Please note that private auctions run for only three (3) days.

2.9 Private Auction Invitation Email

All auctions resulting from multiple backorders of a domain are private. Only the original applicants who placed backorder for the auctioned domain will be invited (closed to everyone else).

The system sends an auction invitation email to each applicant, which contains a secret cookie. If a valid OWNERCONTACT was provided during the backorder process then the invitation is sent to this contact, otherwise, the email will be sent to the email on file of the customer account that placed the backorder application. Also if the OWNERCONTACT is present this information is seamlessly carried over to the PremiumSale.com. This means that signup and account information is provided to PremiumSale.com to expedite signup and auction participation.

Your backorder of XXXXX.TLD through your Domain Service Provider has been successfully registered:

Domain Name: XXXXX.TLD
Registration Date: 2011-04-26
Backorder Status: PENDING AUCTION

However, prior to registration of this domain, multiple backorders for this domain were submitted along with yours. As a result, this backordered domain will be auctioned off between you and the other backordering clients.

PremiumSale.com has been authorized by your Service Provider to conduct a PRIVATE auction for ONLY you and the other original backorder clients of this domain. The auction details are provided as follows:

Auction Provider: PremiumSale.com
Auction Start Date: 2011-05-06 20:00:00 UTC
Auction Opening Bid: USD 69.00
Max. Auction Participants: 4

To participate in this auction please click on the web link below:
http://www.premiumsale.com/aftermarket/?sale=873-ADF4E-Y38S-TEST

If this is your first auction at PremiumSale.com, please register for an account with us prior to participating in the auction.

2.10 Winning a Private Auction and Reseller Commission

When an auction successfully closes and the auction winner makes a payment within 14 days, then the domain name is automatically assigned to the auction winner's customer account. The application status will change to AUCTION-WON, and the system creates a DOMAINAPPLICATION :: BACKORDER_AUCTION_WON event. Additionally, if the original backorder was placed through a reseller of class commission rated class, then commissions are paid out to the reseller account placing the original backorder application.

STATUS – AUCTION-WON EVENT - DOMAINAPPLICATION :: BACKORDER_AUCTION_WON

2.11 Losing a Private Auction

Losing an auction is the result of the backorder applicant not responding to the auction invite, not bidding, or being outbid by other backorder applicants of the same domain name. The status of the application will change to AUCTION-LOST, and the system creates DOMAINAPPLICATION :: BACKORDER_AUCTION_LOST event.

STATUS – AUCTION-LOST EVENT - DOMAINAPPLICATION :: BACKORDER_AUCTION_LOST

2.12 Non-Participation in Auction

Whenever an auction for a domain, that originally had multiple backorders, fails to garner any auction bids by the end of the auction, the domain is then assigned on a first-come-first-serve basis (the first active backorder application made on the system). The system bills and allocates this type of backorder no differently than a domain, which had only a single backorder request on it.

2.13 Failure to Pay Auctioned Backorder

If payment is not received for a domain won at auction within 14 days of the auction's close, the auction is voided and a new auction is started. The same domain will be re-auctioned with the original backorder applicants less

the one applicant who failed to make payment. And the backorder application for the user that failed to pay will change to a FAILED status.

Additional auctions only occur if two or more active backorder applications remain after removing the application of the user who failed to pay. If only a single backorder application remains, then the system bills and allocates the backorder no differently than a domain, which had only a single backorder request on it.

2.14 TLD Registries Not Supporting Domain Auctions

Some registries allow backordering of deleted domains, as well as, the auction of backorders whenever there is contention between multiple backorder applicants. However, some TLDs only allow backorder on a first-come-first-serve basis (FCFS).

Backorders for TLDs where FCFS is applicable will be handled in the same manner as a single backorder. The system bills and allocates the backorder no differently than a domain, which had only a single backorder request on it.

3 RESELLER BILLING, PAYMENT, and COMMISSIONS

Backordered domains like all objects in the system have a price class associated with them. However, backorders also pay commissions back to the reseller if a successful backorder is process through a resellers account.

Not all resellers qualify for these commissions. Please contact your service provider to ask about backorder commissions. This section details backorder billing and commissions if applicable

NOTE:

In the relation examples below, the TLD marker any of the strings needs to be replaced by the appropriate domain class (ie. COM or NET)

EXAMPLE - PRICE_CLASS_DOMAIN**_TLD_**BACKORDER_CURRENCY

USAGE - PRICE_CLASS_DOMAIN**_COM_**BACKORDER_CURRENCY

3.1 Backorder Currency

The relation for backorder currency is:

PRICE_CLASS_DOMAIN_TLD_BACKORDER_CURRENCY

This is the currency in which successful backorders should be charged for the respective domain class (COM, NET, …). If this relation in not defined then the system will use the standard currency for registering regular domains of the respective class (PRICE_CLASS_DOMAIN_TLD_CURRENCY). Additionally, if the standard currency is not defined then the active currency setting of the customer's account will be used.

3.2 Backorder Application

The relation for a backorder application is:

PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION

This is the price for a successful backorder for the respective domain class (COM, NET, …) when it is sold directly and does not process through a backorder auction. This relation is also the reserved deposit that the system takes before the domain's deletion date if a TLD backorder deposit is undefined. Also, if the backorder goes to auction or the backorder fails, this value is the amount that is automatically refunded back to the customer's account. Please note that customers who do not have this relation setting are prevented from making backorder.

3.3 Backorder Deposit

The relation for a backorder deposit is:

PRICE_CLASS_DOMAIN_TLD_BACKORDER_DEPOSIT

This is the backorder reserved deposit for a domain class (COM, NET, …) and overrides the PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION relation when defined. The system processes backorder deposits three (3) days before a domain's finalization (drop) date.

3.4 Payment for Successful Single Backorders via Account Balance

When only one backorder application is made on an expiring domain and the system successfully registers the respective domain name, the reseller system handles the backorder billing by appropriately charging the reseller and sub-resellers, related to the sale, in accordance to the following relations:

PRICE_CLASS_DOMAIN_TLD_BACKORDER_APPLICATION PRICE_CLASS_DOMAIN_TLD_BACKORDER_CURRENCY

Nothing is charged if the backorder domain cannot be registered. If there is a successful registration then the domain is automatically assigned into the customer's account.

3.5 Payment for Backorder Won at Auction

Backorders going to auction due to multiple backorder applications are no longer billed in the system. The auction is handled by PremiumSale.com as well as payments post auction close.

3.6 Commissions on Backorders – If Applicable

Commissions are paid to successful single backorders, as well as, backorders that go to auction. Depending on the backorder commission class, the backorder system will automatically process commissions based on the percentage commission rate defined in the following relation:

PRICE_CLASS_DOMAIN_TLD_BACKORDER_COMMISSION

Commissions will be calculated for all resellers and sub-resellers, related to the successful sale of a backorder, according to their individually defined commission relation between one another. Additionally, commissions are paid out at the end of a given month. And no commissions are paid on losing applications, failed backorder registrations, failed auctions, or any backorder application that fails to make payment.

4 API REFERENCE

This API command reference below details the API commands for the Backordering Service. The API itself is NOT object oriented but is objected related and can be encapsulated into various Object-Oriented languages. he reseller's account or the reseller's customer at the conclusion of the auction.

Please adhere to the following BOX COLORS:

GREEN – Request and Response Parameters
BLUE – Example Command Use
GREY – EPP Request and Response if Applicable

4.1 COMMAND - Create a Backorder Application

Request Parameters:

Required:

COMMAND = AddDomainApplication
CLASS = BACKORDER
DOMAIN = <DOMAIN NAME>

Optional:

AUTH = AUTH code for the domain name **(note 1)**
OWNERCONTACT0 = contact handle ID of registrant **(note 2)**
ADMINCONTACT0 = contact handle ID of the administrative contact
TECHCONTACT0 = contact handle ID of the technical contact
BILLINGCONTACT0 = contact handle ID of the billing contact
NAMESERVER0 = nameserver #1
NAMESERVER1 = nameserver #2
...
NAMESERVER12 = nameserver #13

Note:

(1) If the application has been completed successfully (BACKORDER_SUCCESSFUL or BACKORDER_AUCTION_WON) this AUTH code will be set for the domain name.
(2) If not given the contact details of the user the application is assigned.

Response Parameters:

Parameters:

CLASS = application class (syntax: <TLD>_BACKORDER)
APPLICATION = unique application ID
STATUS = application status **(NOTE 1)**
DOMAIN = domain name
USER = user id the domain application is assigned to
PEERUSER = User ID hierarchy (tree) that domain application is assigned
CREATEDDATE = application created date
UPDATEDDATE = last application updated date
AUTH = AUTH code for the domain name **(NOTE 2)**

Note:

(1) Possible values are:
'INCOMPLETE','REQUESTED','ACTIVE','PROCESSING','SUCCESSFUL','FAILED'
,'DELETED','AUCTION- PENDING','AUCTION-WON','AUCTION-LOST'.

(2) If the application has been completed successfully (BACKORDER_SUCCESSFUL or BACKORDER_AUCTION_WON) this AUTH code will be set for the domain name.

Example:

- To create .COM backorder (only the domain name is given)

RRP Request/Response:

```
[COMMAND]
command=AddDomainApplication
class=BACKORDER
domain=hexotest123.com
EOF
[RESPONSE]
PROPERTY[CREATEDDATE][0]=2011-06-08 12:38:07
PROPERTY[CLASS][0]=COM_BACKORDER
PROPERTY[PEERUSER][0]=
PROPERTY[APPLICATION][0]=2962
PROPERTY[STATUS][0]=REQUESTED
PROPERTY[UPDATEDDATE][0]=2011-06-08 12:38:07
PROPERTY[UPDATEBY][0]=SYSTEM
PROPERTY[CREATEDBY][0]=SYSTEM
PROPERTY[DOMAIN][0]=hexotest123.com
PROPERTY[AUTH][0]=t30elV1QUE
PROPERTY[USER][0]=test.user
PROPERTY[PERIOD][0]=1
DESCRIPTION=Command completed successfully
CODE=200
```

EPP:

EPP request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <extension>
      <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key='COMMAND' value='AddDomainApplication' />
            <keyvalue:kv key='CLASS' value='BACKORDER' />
            <keyvalue:kv key='DOMAIN' value='hexotest123.com' />
         </keyvalue:extension>
    </extension>
</epp>
```

EPP response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1001">
         <msg>Command completed successfully; action pending</msg>
         <extValue>
            <value xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">
               <epp:undef/>
            </value>
            <reason>200 Command completed successfully</reason>
         </extValue>
      </result>
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="APPLICATION" value="2962"/>
            <keyvalue:kv key="AUTH" value="t30elV1QUE"/>
            <keyvalue:kv key="CLASS" value="COM_BACKORDER"/>
            <keyvalue:kv key="CREATEDBY" value="SYSTEM"/>
            <keyvalue:kv key="CREATEDDATE" value="2011-06-08 12:38:07"/>
            <keyvalue:kv key="DOMAIN" value="hexotest123.com"/>
            <keyvalue:kv key="PEERUSER" value=""/>
            <keyvalue:kv key="PERIOD" value="1"/>
            <keyvalue:kv key="STATUS" value="REQUESTED"/>
            <keyvalue:kv key="UPDATEBY" value="SYSTEM"/>
            <keyvalue:kv key="UPDATEDDATE" value="2011-06-08 12:38:07"/>
            <keyvalue:kv key="USER" value="test.user"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <svTRID>RW-599-1307537399864592</svTRID>
      </trID>
   </response>
</epp>
```

4.2 COMMAND - Delete a Backorder Application

Request Parameters:

Required:
COMMAND = DeleteDomainApplication
CLASS = BACKORDER
APPLICATION = unique application ID

Response Parameters:

Parameters:
DOMAIN = domain name

Example:

- Delete backorder application with ID 2963 (domain "hexotest124.com")

RRP Request/Response:

```
[COMMAND]
command=DeleteDomainApplication
application=2963
EOF
[RESPONSE]
PROPERTY[DOMAIN][0]=hexotest124.com
DESCRIPTION=Command completed successfully
CODE=200
```

EPP:

EPP request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <extension>
      <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key='COMMAND' value='DeleteDomainApplication' />
            <keyvalue:kv key='APPLICATION' value='2963' />
        </keyvalue:extension>
    </extension>
</epp>
```

EPP response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1000">
         <msg>Command completed successfully</msg>
         <extValue>
            <value xmlns:epp="urn:ietf:params:xml:ns:epp-1.0">
               <epp:undef/>
            </value>
            <reason>200 Command completed successfully</reason>
         </extValue>
      </result>
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="DOMAIN" value="hexotest124.com"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <svTRID>RW-599-1307537812178544</svTRID>
      </trID>
   </response>
</epp>
```

4.3 COMMAND - Query List All Backorder Applications

Request Parameters:

Required:

COMMAND = QueryDomainApplication
CLASS = BACKORDER

Optional:

WIDE = 1
STATUS = filter the list by STATUS value **(NOTE 1)**
ORDERBY = order by parameter, suffix "DESC" is descending order **(NOTE 2)**
USERDEPTH = filter the list by choosing user depth **(NOTE 3)**

Note:

(1) Allowed values: 'INCOMPLETE','REQUESTED','ACTIVE','PROCESSING','SUCCESSFUL','FAILED','DELETED','AUCTION-PENDING','AUCTION-WON','AUCTION-LOST').

(2) Allowed values: CLASS, CLASSDESC, DOMAIN, DOMAINDESC, USER, USERDESC, CREATEDDATE, CREATEDDATEDESC, UPDATEDDATE, UPDATEDDATEDESC.

(3) SELF (only own application) | SUBUSER (only applications of subusers) | ALL (all applications)

Response Parameters:

Common List of Parameters:

Returned as an Array with Only One Element

TOTAL = number of total list entries
FIRST = number of first returned list entry
COUNT = number of returned entries
LAST = number of last returned list entry
LIMIT = limitation of returned list entries

List Entries Returned as an Array

CLASS = application class (syntax: <TLD>_BACKORDER)
APPLICATION = unique application ID
STATUS = application status **(NOTE 1)**
DOMAIN = domain name
USER = user id the domain application is assigned to
PEERUSER = User ID hierarchy (tree) that domain application is assigned
PARENTUSER = parent user id the domain application is assigned to
CREATEDDATE = application created date
UPDATEDDATE = last application updated date
AUTH = AUTH code for the domain name **(NOTE 2)**
OWNERCONTACT = contact handle ID of registrant **(NOTE 3)**
ADMINCONTACT = contact handle ID of the administrative contact
TECHCONTACT = contact handle ID of the technical contact
BILLINGCONTACT = contact handle ID of the billing contact

EXPIRATIONDATE = TBD
X-DOMAIN-ROID = TBD
MISSINGPARAMETER = TDB
INVALIDPARAMETER = TDB

Note:

(1) Allowed values: 'INCOMPLETE','REQUESTED','ACTIVE','PROCESSING','SUCCESSFUL','FAILED','DELETED','AUCTION- PENDING','AUCTION-WON','AUCTION-LOST'.
(2) If the application has been completed successfully (BACKORDER_SUCCESSFUL or BACKORDER_AUCTION_WON) this AUTH code will be set for the domain name.
(3) If not given the contact details of the user the application are assigned

Example:
- To query a detailed list of all backorder applications.

RRP Request/Response:

```
[COMMAND]
command=QueryDomainApplicationList
class=BACKORDER
wide=1
EOF
[RESPONSE]
PROPERTY[CREATEDDATE][0]=2011-06-09 13:28:11
PROPERTY[PEERUSER][0]=
PROPERTY[BILLINGCONTACT][0]=P-ABH91365
PROPERTY[TECHCONTACT][0]=P-ABH91365
PROPERTY[APPLICATION][0]=2972
PROPERTY[STATUS][0]=FAILED
PROPERTY[X-DOMAIN-ROID][0]=
PROPERTY[OWNERCONTACT][0]=P-ABH91365
PROPERTY[MISSINGPARAMETER][0]=
PROPERTY[COUNT][0]=1
PROPERTY[LAST][0]=0
PROPERTY[USER][0]=test.user
PROPERTY[INVALIDPARAMETER][0]=
PROPERTY[LIMIT][0]=10000
PROPERTY[ADMINCONTACT][0]=P-ABH91365
PROPERTY[CLASS][0]=COM_BACKORDER
PROPERTY[PARENTUSER][0]=demo.hexonet.net
PROPERTY[EXPIRATIONDATE][0]=
PROPERTY[UPDATEDDATE][0]=2011-06-09 13:55:03
PROPERTY[TOTAL][0]=1
PROPERTY[FIRST][0]=0
PROPERTY[DOMAIN][0]=hexotest123.com
DESCRIPTION=Command completed successfully
CODE=200
```

4.4 COMMAND - Query List of Deleting Domains

Downloadable CSV List:

http://www.hexonet.net/files/domain-backordering/pending_delete_domain_list.csv.zip

Request Parameters:

Required:

COMMAND = QueryDomainPendingDeleteList

Optional:

KEYWORD# = filter by list of keywords **(NOTE 1)**
KEYWORDCATENATION = keyword catenation with "AND" or "OR"
KEYWORDMATCH = keyword must match **(NOTE 2)**
ZONE# = filter by list of zones **(NOTE 3)**
DELETIONDAY# = filter by list of deletion days **(NOTE 4)**
FINALIZATIONDATE# = filter by list of finalization dates **(NOTE 5)**
LENGTH# = filter by list of length of domain name (range: 1-20) **(NOTE 6)**
SCORE# = filter by list of scores of domain name (range: 1-10) **(NOTE 7)**
FILTER# = filter by domain name criteria **(NOTE 8)**
ORDERBY = order by parameter, suffix "DESC" is descending order **(NOTE 9)**
LIMIT = limitation of maximum number of returned entries
FIRST = number of first returned entry

Note:

(1) Specify as KEYWORD0=..., KEYWORD1=..., ...
(2) Begin of domain name ("BEGIN"), End of domain name ("END"), Any part of the domain name ("ANY")
(3) Allowed specified with ZONE0=..., ZONE1=... (valid zones: COM, NET,..)
(4) Specify as DELETIONDAY0=..., DELETIONDAY1=..., ...
(5) Specify as FINALIZATIONDATE0=..., FINALIZATIONDATE1=..., ...
(6) Specify as LENGTH0=..., LENGTH1=..., ...
(7) Specify as SCORE0=..., SCORE1=..., ...
(8) "NOHYPHENS" for domain names with no hyphens, "NONUMBERS" for no number domain names, "NOIDNS" for no IDN domain names are a few examples
(9) Allowed values: NAME | DELETIONDAY | FINALIZATIONDATE | SCORE | NAMEDESC | DELETIONDAYDESC | FINALIZATIONDATEDESC | SCOREDESC)

Response Parameters:

Common List of Parameters:

Returned as an Array with Only One Element

TOTAL = number of total list entries
FIRST = number of first returned list entry
COUNT = number of returned entries
LAST = number of last returned list entry
LIMIT = limitation of returned list entries

List Entries Returned as an Array

DOMAIN = domain name
SCORE = Domain Score
FINALIZATIONDATE = backorder finalization date

Example:

- TOP10 .COM|.NET backorder domain list
- Query a list with the first 10 .COM|.NET backorder domains ordered by Domain Score (descending)

RRP Request/Response:

```
[COMMAND]
command=QueryDomainPendingDeleteList
zone0=com
zone1=net
orderby=SCOREDESC
limit=10
EOF
```

```
[RESPONSE]

PROPERTY[FINALIZATIONDATE][0]=2011-06-21 17:00:00
PROPERTY[FINALIZATIONDATE][1]=2011-06-17 17:00:00
PROPERTY[FINALIZATIONDATE][2]=2011-06-20 17:00:00
PROPERTY[FINALIZATIONDATE][3]=2011-06-20 17:00:00
PROPERTY[FINALIZATIONDATE][4]=2011-06-19 17:00:00
PROPERTY[FINALIZATIONDATE][5]=2011-06-20 17:00:00
PROPERTY[FINALIZATIONDATE][6]=2011-06-16 17:00:00
PROPERTY[FINALIZATIONDATE][7]=2011-06-16 17:00:00
PROPERTY[FINALIZATIONDATE][8]=2011-06-19 17:00:00
PROPERTY[FINALIZATIONDATE][9]=2011-06-18 17:00:00
PROPERTY[SCORE][0]=10
PROPERTY[SCORE][1]=10
PROPERTY[SCORE][2]=10
PROPERTY[SCORE][3]=10
PROPERTY[SCORE][4]=10
PROPERTY[SCORE][5]=10
PROPERTY[SCORE][6]=10
PROPERTY[SCORE][7]=10
PROPERTY[SCORE][8]=10
PROPERTY[SCORE][9]=10
PROPERTY[TOTAL][0]=403908
PROPERTY[FIRST][0]=0
PROPERTY[DOMAIN][0]=ameridream.net
PROPERTY[DOMAIN][1]=anglegrinders.net
PROPERTY[DOMAIN][2]=auto-site.net
PROPERTY[DOMAIN][3]=barcode-scanners.net
PROPERTY[DOMAIN][4]=desktop-wallpapers.net
PROPERTY[DOMAIN][5]=envirocheck.net
PROPERTY[DOMAIN][6]=event-management.net
PROPERTY[DOMAIN][7]=freeconsultation.net
PROPERTY[DOMAIN][8]=glyde.net
PROPERTY[DOMAIN][9]=gpower.net
PROPERTY[COUNT][0]=10
PROPERTY[LAST][0]=9
PROPERTY[LIMIT][0]=10
DESCRIPTION=Command completed successfully
CODE=200
```

4.5 COMMAND - Backorder Events

Poll a List of All Available Events

RRP Request/Response:

```
[COMMAND]
command=QueryEventList
EOF
[RESPONSE]
PROPERTY[TOTAL][0]=3
PROPERTY[FIRST][0]=0
PROPERTY[EVENT][0]=225780
PROPERTY[EVENT][1]=225784
PROPERTY[EVENT][2]=225788
PROPERTY[COUNT][0]=3
PROPERTY[LAST][0]=2
PROPERTY[LIMIT][0]=100
DESCRIPTION=Command completed successfully
CODE=200
```

RRP Request/Response:

```
[COMMAND]
command=StatusEvent
event=225780
EOF
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_AUCTION_LOST
property[data][0] = application:2989
property[data][1] = domain:hexonet124.com
property[info][0] =
This%20application%20did%20not%20win%20the%20auction%2e
```

EPP: Event Polling

EPP request:

EPP request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <command>
         <poll op='req' />
      <clTRID>POLL-REQUEST123</clTRID>
    </command>
</epp>
```

4.6 STATUS - BACKORDER_SUCCESSFUL

4.6.1 Single Backorder

RRP StatusEvent Response:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_SUCCESSFUL
property[data][0] = application:2967
property[data][1] = domain:hexonet124.com
property[info][0] =
The%20backorder%20for%20this%20domain%20name%20was%20successful
%2e
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="7" id="226268">
         <qDate>2011-06-08T13:40:06.0Z</qDate>
         <msg>BACKORDER_SUCCESSFUL</msg>
      </msgQ>
```

```
     <extension>
        <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
           <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
           <keyvalue:kv key="EVENTINFO"
value="The%20backorder%20for%20this%20domain%20name%20was%20successful%2e"/>
           <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_SUCCESSFUL"/>
           <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
           <keyvalue:kv key="OBJECTID" value="2967"/>
        </keyvalue:extension>
     </extension>
     <trID>
        <clTRID>POLL-REQUEST123</clTRID>
        <svTRID>RO-599-1307627449706611</svTRID>
     </trID>
  </response>
</epp>
```

4.6.2 Backorder with Multiple Applications

RRP Response:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_SUCCESSFUL
property[data][0] = application:2966
property[data][1] = domain:hexonet124.com
property[info][0] =
The%20backorder%20for%20this%20domain%20name%20was%20successful
%2e%20%28No%20winner%20at%20auction%2c%20first%20application%29
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="6" id="226272">
         <qDate>2011-06-08T13:40:07.0Z</qDate>
         <msg>BACKORDER_SUCCESSFUL</msg>
      </msgQ>
```

```
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
            <keyvalue:kv key="EVENTINFO"
value="The%20backorder%20for%20this%20domain%20name%20was%20successful%2e%20
%28No%20winner%20at%20auction%2c%20first%20application%29"/>
            <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_SUCCESSFUL"/>
            <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
            <keyvalue:kv key="OBJECTID" value="2966"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <clTRID>POLL-REQUEST123</clTRID>
         <svTRID>RO-599-1307627545902318</svTRID>
      </trID>
   </response>
</epp>
```

4.7 STATUS - BACKORDER_FAILED

Single and Multiple Backorders

RRP StatusEvent Response:

```
[RESPONSE]
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_FAILED
property[data][0] = application:2973
property[data][1] = domain:hexonet124.com
property[info][0] =
The%20backorder%20for%20this%20domain%20name%20was%20not%20succe
ssful%2e"
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="13" id="226653">
         <qDate>2011-06-09T14:20:06.0Z</qDate>
         <msg>BACKORDER_FAILED</msg>
      </msgQ>
```

```
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
            <keyvalue:kv key="EVENTINFO"
value="The%20backorder%20for%20this%20domain%20name%20was%20not%20successful%2e"/
>
            <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_FAILED"/>
            <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
            <keyvalue:kv key="OBJECTID" value="2973"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <clTRID>POLL-REQUEST123</clTRID>
         <svTRID>RO-1626-1307629662554443</svTRID>
      </trID>
   </response>
</epp>
```

4.8 STATUS - BACKORDER_AUCTION_PENDING

For Multiple Backorders

RRP StatusEvent Response:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_AUCTION_PENDING
property[data][0] = application:2966
property[data][1] = domain:hexonet124.com
property[info][0] =
Since%20multiple%20backorders%20for%20this%20domain%20were%20placed
%2c%20an%20auction%20will%20be%20held%20to%20equitably%20allocate%
20this%20domain%2e%2c%20an%20auction%20will%20be%20held%20to%20e
quitably%20allocate%20this%20domain%2e
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="8" id="226264">
         <qDate>2011-06-08T13:40:06.0Z</qDate>
         <msg>BACKORDER_AUCTION_PENDING</msg>
      </msgQ>
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
            <keyvalue:kv key="EVENTINFO"
value="Since%20multiple%20backorders%20for%20this%20domain%20were%20placed%2c%20
an%20auction%20will%20be%20held%20to%20equitably%20allocate%20this%20domain%2e"/>
            <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_AUCTION_PENDING"/>
            <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
            <keyvalue:kv key="OBJECTID" value="2966"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <clTRID>POLL-REQUEST123</clTRID>
         <svTRID>RO-599-1307627327310251</svTRID>
      </trID>
   </response>
</epp>
```

4.9 STATUS - BACKORDER_AUCTION_WON

For Multiple Backorders

RRP StatusEvent Response:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_AUCTION_WON
property[data][0] = application:2982
property[data][1] = domain:hexonet124.com
property[info][0] =
This%20backorder%20application%20won%20the%20auction%21
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="1" id="226701">
         <qDate>2011-06-09T14:20:11.0Z</qDate>
         <msg>BACKORDER_AUCTION_WON</msg>
      </msgQ>
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
            <keyvalue:kv key="EVENTINFO"
value="This%20backorder%20application%20won%20the%20auction%21"/>
            <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_AUCTION_WON"/>
            <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
            <keyvalue:kv key="OBJECTID" value="2982"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <clTRID>POLL-REQUEST123</clTRID>
         <svTRID>RO-1626-1307629751077034</svTRID>
      </trID>
   </response>
</epp>
```

4.10 STATUS - BACKORDER_AUCTION_LOST

For Multiple Backorders

RRP StatusEvent Response:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2011-06-09 15:00:20.0
property[class][0] = DOMAINAPPLICATION
property[subclass][0] = BACKORDER_AUCTION_LOST
property[data][0] = application:2982
property[data][1] = domain:hexonet124.com
property[info][0] =
This%20application%20did%20not%20win%20the%20auction%2e
```

EPP Poll Response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0 epp-1.0.xsd">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="1" id="226797">
         <qDate>2011-06-09T15:00:20.0Z</qDate>
         <msg>BACKORDER_AUCTION_LOST</msg>
      </msgQ>
      <extension>
         <keyvalue:extension xmlns:keyvalue="http://schema.ispapi.net/epp/xml/keyvalue-1.0"
xsi:schemaLocation="http://schema.ispapi.net/epp/xml/keyvalue-1.0 keyvalue-1.0.xsd">
            <keyvalue:kv key="EVENTCLASS" value="DOMAINAPPLICATION"/>
            <keyvalue:kv key="EVENTINFO"
value="This%20application%20did%20not%20win%20the%20auction%2e"/>
            <keyvalue:kv key="EVENTSUBCLASS" value="BACKORDER_AUCTION_LOST"/>
            <keyvalue:kv key="OBJECTCLASS" value="APPLICATION"/>
            <keyvalue:kv key="OBJECTID" value="2989"/>
         </keyvalue:extension>
      </extension>
      <trID>
         <clTRID>POLL-REQUEST123</clTRID>
         <svTRID>RO-599-1307691384002440</svTRID>
      </trID>
   </response>
</epp>
```