DEERFIELD.COM

DNS2Go Update API

DNS2Go Update API

DEERFIELD.COM PRODUCT DOCUMENTATION

DNS2Go Update API

© Deerfield.com
4241 Old U.S. 27 South
Gaylord, MI 49686
Phone 989.732.8856
Email sales@deerfield.com
http://www.deerfield.com/

Table of Contents

Welcome

What is DNS?

The Domain Name System or Service (DNS) is a process that maps hostnames (how humans address servers) to IP addresses (how computers address servers).When you visted www.dns2go.com, your computer resolved that hostname as 65.117.119.140. This translation happens every time you access a website, FTP server, or your email.

What is DNS2Go?

Sometimes you want to be followed.

For example, if you've ever considered hosting a web or FTP site on a computer with a broadband or dial-up Internet connection, chances are you would want a domain name so that friends and family could use "yourdomain.com" to reach you. But this usually isn't possible simply because a domain name must point to a static IP address, and there's a good chance (if you have a broadband or dialup connection) that yours changes. It could change daily, hourly, weekly, whatever.

What you need is for your domain name to follow your dynamic, ever-changing IP address so that other users on the Internet can easily reach you.

DNS2Go is the solution to this problem. DNS2Go allows anyone to associate a friendly, easy-to-remember domain name with an Internet-connected computer, even if its Internet connection uses dynamic IP address assignment. DNS2Go is just like directory assistance for your Internet connection! Imagine if your Phone number changed once a week, once a day, or once a month! Nobody would ever know what number you are using. Your Internet connection works exactly like this! A DNS2Go domain name will, in fact, follow your current IP address to the ends of the earth, even through routers and firewalls.

What is an API?

The DNS2Go Application Programming Interface (API) is a series of programming commands that allow you to perform actions in lieu of an online interface. The DNS2Go Update API is used to update the IP addresses of dynamic DNS hostnames subscribed to the DNS2Go service.

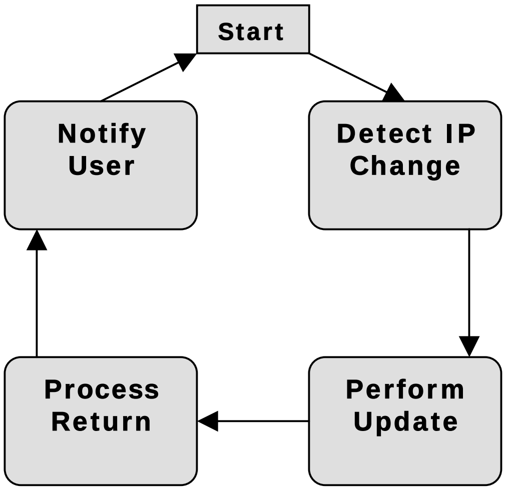
Dynamic DNS Update Process

The DNS client's purpose is to keep the hostname up to date with a user's current IP address. This is done in as follows:

Detect IP Chang*e* – Check for changes to the current IP address
Perform Update – If the IP address has changed or a user updates any setting
Process Return – Parse return code
Notify User – Perform logging; notify the user (if necessary), shutdown on fatal errors

Flow Chart

The flow chart provides an overview of the behavior required from the clients.



Detect IP Change

Determining whether a client should update requires it has a reliable method of determining what its current IP address is, so it can compare to the last updated address.

Directly Connected

The client must determine, whether automatically or through the input of the user, if it is connected directly to the Internet. A direct connection to the Internet would mean the device has a publicly addressable IP address. Direct connections often rely on API methods from the network stack of the parent operating system.

Internet IP Detection

A client may determine, automatically or through the input of the user, that it is not directly connected to the Internet. The client is on a machine with a private IP. In this case, the optimal method is to use Internet IP detection.

Clients with private IPs are typically behind a NAT, requiring the setup of port forwarding.

Complications

Multiple Interfaces

Some clients can be installed on devices with multiple network interfaces, such as a gateway or router. If this is the case, the developer needs to base a basic assumption or the user needs to be given the option as to which interface to use for IP detection.

IpChecker

DNS2Go offers a free web IP detection tool for use with DNS2Go services. The IpChecker will return the remote socket's IP address.

Hostname:
http://update.dns2go.com/ipcheck.aspx or https://update.dns2go.com/ipcheck.aspx

Call and Answer

An HTTP request to the DNS2Go IpChecker will result in a response similar to the following:

```
HTTP/1.1  200  OK

Content-Type: text/html

Server: DNS2Go Updater

Connection: close

Cache-Control: no-cache

Pragma: no-cache

Content-Length:  80

<html><head><title>DNS2Go
IpCheck</title></head><body>63.149.6.103</body></html>
```

Policies

- Send a well-formed user-agent including company name, model number, and software build revision.
- If possible, use the Operating System for IP detection
- Checks must be performed at least 15 minutes apart to help reduce server load
- In the case of an error while accessing IpCheckerDee, the client should not send an update

Perform Update

The client should perform an update when a change in IP address is detected or if a user alters any of their settings. All updates are sent using a specifically formatted HTTP request. The DNS2Go Updater will pass back a return code that the client needs to parse.

The HTTP Request

Updates can be performed over HTTP or securely via SSL encrypted HTTPS, both GET and POST requests are acceptable.

Update hostname:	update.dns2go.com/update.aspx
HTTP update port:	80
HTTPS update port:	443

Update Parameters

PARAMETER	REQUIRED	DEFAULT VALUE	SAMPLE VALUE
FQDN	Yes	None	mydomain.dns2go.com
KEY	Yes	None	001153-UYREN-YMMKB-10000
COMMAND	No	1	1
IP	No	Client IP Address	62.12.6.13

FQDN – your DNS2Go domain name
KEY – your DNS2Go key
COMMAND – 1=Online (default), 0=Offline
IP – The IP address to associate with the domain (default is the connecting client IP)

Examples

Note: Online and Offline configuration of domains, including custom MX and other DNS records is performed via the DNS2Go Account Manager at: https://www.dns2go.com/am Configure the online and/or offline settings before using the DNS2Go Updater.

Note: Taking a domain online or offline via the DNS2Go Updater will cause all domains in the same DNS2Go Domain Group to be taken online or offline. DNS2Go Domain Groups are configured via the DNS2Go Account Manager.

Example 1: HTTP GET Request taking the domain online with the client IP

http://update.dns2go.com/update.aspx?fqdn=mydomain.dns2go.com&key=001153-UYREN-YMMKB-10000

Example 2: HTTP GET Request taking the domain online with the specified IP

http://update.dns2go.com/update.aspx?fqdn=mydomain.dns2go.com&key=001153-UYREN-YMMKB-10000&ip=12.45.89.78

Example 3: HTTP GET Request taking the domain offline

http://update.dns2go.com/update.aspx?fqdn=mydomain.dns2go.com&key=001153-UYREN-YMMKB-10000&command=0

Example 4: Programmatic GET Request (varies with client host operating system and programming language)

```
GET update.aspx?fqdn=mydomain.dns2go.com&key=001153-UYREN-YMMKB-10000
HTTP/1.0
Host: update.dns2go.com
User-Agent: Company - Device - Version Number
```

Example 5: HTTP POST Request form

Create a new text document named d2gupdate.htm and copy the following HTML to it. Open the document with a web browser, fill in the appropriate fields and submit the POST request.

```
<html>
<body>
<form name="d2gupdate" action="http://update.dns2go.com/update.aspx"
method="post">
        <table cellpadding="0" cellspacing="0" border="1">
                <tr>
                        <td>FQDN:</td>
                        <td><input type="text" name="fqdn" value="" size="50"></td>
                </tr>
                <tr>
                        <td>Key:</td>
                        <td><input type="text" name="key" value="" size="50">  </td>
                </tr>
                <tr>
                        <td>IP:</td>
                        <td><input type="text" name="ip" value="" size="50"></td>
                </tr>
                <tr>
                        <td>Command:</td>
                        <td><select name="command">
                                <option value="1" selected>Online</option>
                                <option value="0">Offline</option>
                              </select>
                        </td>
                </tr>
                <tr>
                        <td colspan="2" align="right">
                                <input type="submit" value="Submit">
                        </td>
        </table>
</form>
</body>
</html>
```

Return Codes

When updating a hostname, the response to the update syntax will be one of the return codes. If there is an error, clients should communicate to the user a brief description of the problem that the return code indicates.

Update Types

Successful Updates

There is only one return code that indicates a successful update. Code: 0 indicates that the update was successful and that the IP address was changed in our system.

Fatal Updates

Any failed update attempt is fatal which means that all further updates will also fail until the user has taken some sort of corrective action. Any failed update should cause the client to be disabled until the situation is corrected and the client re-enabled by the user.

Additionally, because the update may fail for a number of different reasons, the client needs to provide some method of communicating with the user that the update has failed and why. Some suggestions include:

- Communicating to an external process running on the users desktop
- Generating an error message to an email address configured by the user
- Logging a message to a log window specific to the DNS2Go client
- Logging a message in the general log window for the router (assuming it has one)

Many of these errors involve configuration mistakes within the client or inconsistencies between the client configuration and the user's account status. In all of these cases, the client must stop updating until the user has corrected the problem. Retrying the update automatically in those cases is abuse and may result in a client block.

HTTP Response

An HTTP request to the DNS2Go Updater will result in a response similar to the following:

```
HTTP/1.1   200  OK

Content-Type: text/html

Server: DNS2Go Updater

Connection: close

Cache-Control: no-cache

Pragma: no-cache

Content-Length:  80

<html><head><title>DNS2Go Update</title></head><body>Code:0
Data:mydomain.dns2go.com;12.45.89.78;1</body></html>
```

The format of the response is as follows:

Code: - A code indicating a successful or failed update (see return codes table)
Data: - Useful information that further explains the code

Return Codes

The following codes could be returned by the DNS2Go Updater service, note that only a Code:0 indicates a successful update. The Data: returned by a Code:0 includes the FQDN (DNS2Go Domain) that was updated, the IP Address that was associated with the domain and the applicable command (1=Online, 0=Offline).

Note: If custom records have been defined for the domain within the DNS2Go Account Manager, the returned values may not be accurate.

CODE	DATA	DESCRIPTION
Code:0	Data:FQDN;IP;COMMAND	Successful Update
Code:1	Data:FQDN_INVALID	The submitted FQDN was invalid or not found in the DNS2Go system
Code:2	DATA:KEY_INVALID	The submitted KEY was invalid or not found in the DNS2Go system
Code:3	DATA:IP_INVALID	The submitted IP Address was invalid
Code:4	DATA:ONLINE_FAILURE	The system was unable to take the domain online
Code:5	DATA:OFFLINE_FAILURE	The system was unable to take the domain offline
Code:6	DATA:USERID_INVALID	The UserID portion of the key is invalid
Code:7	DATA:DOMAINID_INVALID	The DomainID associated with the submitted FQDN is invalid
Code:8	DATA:AUTH_FAILURE	The submitted FQDN and KEY does not authenticate in the system
Code:9	DATA:KEY_MISMATCH	The submitted FQDN and KEY results in a UserID mismatch
Code:10	DATA:NOT_IMPLEMENTED	The submitted COMMAND is invalid or has not been implemented

Policies

To ensure fair use of our systems, client developers must ensure their clients follow these policies. Clients not adhering to these policies are at risk of being blocked from accessing our systems.

Required Client Behavior

- Send a well-formed user-agent including company name, model number, and software build revision.
- Check that all input is in valid form before updating
- Check that any IP obtained through web-based IP detection is a valid dotted quad numeric IP (eg: 1.2.3.4) before sending it in an update
- Only update when the IP address is different from the IP of the last update

Unacceptable Client Behavior

- Send requests to or access anything other than ?fqdn=&key= at the update host
- Reverse engineer web requests to our web site to create or delete hostnames
- Hardcode the IP address of any DNS2Go server
- Attempt to update after receiving a failed return code without user intervention
- Perform DNS queries to determine whether the client IP requires updating
- Access the DNS2Go IpChecker more than once every 15 minutes

Guidelines

These guidelines serve as our recommendations on how to handle various issues involved in client development to ensure that the client does not violate our policies.

POST vs GET

We highly encourage the use of GET as opposed to POST because of the simplified debugging process.

System Startup and DHCP Leases

Clients should store the IP in a manner that a system restart or power cycling does not cause the client to update unless the IP has changed. Clients must not update every time a DHCP lease is renewed. Instead they must check the new IP and determine if it differs from the old IP before updating.

DNS Queries

Clients should not perform DNS queries to determine whether it is necessary to update.

The danger is that the ISP's DNS server will be caching the old IP for a few minutes, leading the client to conclude the update failed and causing a loop.

HTTP Headers

The HTTP headers returned may be status 200 (ok), 401 (Authorization Required) or 500 (Internal Server Error). The response body should be parsed for return codes no matter what the status is.

User Input

Users need to enter a Domain and Registration Key. It is important the Domain field be long enough to support up to 200 characters.

Denying Clients

Clients and developers not following the DNS2Go Update API may become blocked. If a client poses a serious threat to the DNS2Go system and a developer ignores a report of the problem, DNS2Go may take necessary action to block or disable those clients. If a client is denied, DNS2Go will work with the developer as follows:

- Contact the developer/company to inform them of the problem
- Work to develop a fix as a firmware patch or future release
- Promote new software revision to encourage adoption

Contacting Deerfield.com / DNS2Go

Feedback, suggestions, comments

Deerfield.com greatly values feedback of all types from customers that are actively using our software solutions. Your suggestions and comments are an important part of the ongoing product development process, and help to shape the future functionality of our entire product line.

Please contact us with your thoughts at feedback@deerfield.com. We look forward to hearing from you.

Getting technical support

Deerfield.com offers a range of technical support options, available to both trial and registered product users. For a complete and current listing of these support options, please visit our customer support system on the web at http://www.deerfield.com/support/.

About Deerfield.com

Deerfield.com is a rural company situated in a small Northern Michigan community. This is important because it defines not only who we are as a group of people, but the challenges we face and overcome every day to effectively service our local, national, and international customers. It is this experience that connects us to our customers and qualifies us to recommend products that solve real world problems.

To learn more about Deerfield.com, please visit our corporate website.