

ABA Routing Number Validation Object Library and Database

Version 2.02 for Perl , PHP, and C++

Quentin Sager Consulting
20429 Ring Neck Road
Altoona, FL 32702
USA

www.quentinsagerconsulting.com
support@quentinsagerconsulting.com

The ABA Routing Number Validation Library is a simple object oriented class library and database that provides methods to perform basic validation and lookup of an ABA Routing Number.

The library is available in the Perl, PHP, and C++ programming languages. All versions share a common definition and naming convention for the validation object therefore we use a single description for each method.

The Perl version is designed for and requires Perl version 5 or above. It includes one source file, *aba.pm* and the associated data file *abadata.csv*. This is a simple Perl module that requires no specific installation. It may be used with any script by placing it in the same directory as your Perl script and create the object.

The PHP version is designed for and requires PHP version 4 or above. It includes one source file, *aba.class.php* and the associated data file *abadata.csv*. To use the object simply include the class file in your particular script and create the object.

The C++ version includes two source files, *claba.h* and *claba.cpp* and the associated data file *abadata.csv*. This version is delivered as source only. You must include and compile it with your particular application. It uses no compiler specific values or setting and may be compiled with virtually any C++ compiler on any platform.

::clAba()

This method is used to instantiate and initialize the validation object. For the Perl version only, the object is instantiated by creating a new instance of the ssn module, which equates to the PHP and C++ clAba object constructor.

Parameters:

none

Examples:

Perl

```
# we assume the aba module is in the current directory
use aba;

my $aba = new aba;
```

PHP

```
<?php
require_once("./aba.class.php");

$aba =& new clAba();
?>
```

C++

```
#include "claba.h"

clAba aba;           // static allocation
clAba *aba = new clAba; // dynamic allocation
```

:: IsValid(number)

This method is the heart of the object and must be called before any subsequent *Get* methods. The number may be passed with or without common delimiters. The number is first checked for valid characters and length. Next, the number's checksum is verified to determine if it could be a valid routing number. If the number passes checksum validation the database is then searched to find the associated bank identified by the number.

Parameters:

number
the routing number to validate.

Returns:

- 0 - the number is valid and matching institution found
- 1 - the number is valid but the institution could not be identified
- 2 - the number has not been validated
- 3 - the routing number contains invalid characters
- 4 - the routing number length is too short
- 5 - the routing number length is too long
- 6 - the routing number length is invalid
- 7 - the routing number checksum is incorrect
- 8 - the routing number is invalid

Examples:

Perl

```
# we assume the aba module is in the current directory
use aba;

my $aba = new aba;

    if ($aba->IsValid($number) >= 1)
    {
        # the number passed validation
    }
```

PHP

```
<?php
require_once("./aba.class.php");

    $aba =& new cAba();

    if ($aba->IsValid($number) >= 1)
    {
        // the number passed validation
    }
?>
```

C++

```
#include "claba.h"

cAba *aba = new cAba; // dynamic allocation

if (aba->IsValid(number))
{
    // the number passed basic validation
}
```

::Get Methods

These methods are called after a successful validation to return the formatted results that may or may not identify the banking institution identified by the ABA routing number. The amount of actual information returned may vary.

Parameters:

none

Returns:

String copy of the validation information or empty string. For C++ version information is returned via const char pointer.

::GetAba()

Returns the currently validated ABA routing number.

::GetRoutingSymbol()

Returns the routing symbol identifier extracted from the currently validated ABA routing number.

::GetRoutingInstitution()

Returns the institution symbol extracted from the currently validated ABA routing number.

::GetName()

Returns the name of the bank or financial institution for the currently validated ABA routing number.

::GetAddress()

Returns the address of the bank or financial institution for the currently validated ABA routing number.

::GetCity()

Returns the city of the bank or financial institution for the currently validated ABA routing number.

::GetState()

Returns the state of the bank or financial institution for the currently validated ABA routing number.

::GetZip()

Returns the formatted zip code of the bank or financial institution for the currently validated ABA routing number.

::GetPhone()

Returns the formatted telephone number of the bank or financial institution for the currently validated ABA routing number.

Examples:

Perl

```
# we assume the aba module is in the current directory
use aba;

my $aba = new aba;

if ($aba->IsValid($number) == 0)
{
    # the number passed validation and data record found
    # display the bank name and address

    print $aba->GetName() . "\n";
    print $aba->GetAddress() . "\n";
    print $aba->GetCity() . "\n";
    print $aba->GetState() . "\n";
    print $aba->GetZip() . "\n";
    print $aba->GetPhone() . "\n";
}
```

PHP

```
<?php
require_once("./aba.class.php");

$aba =& new clAba();

if ($aba->IsValid($number) == 0)
{
    // the number passed validation and data record found
    // display the bank name and address

    print $aba->GetName() . "\n";
    print $aba->GetAddress() . "\n";
    print $aba->GetCity() . "\n";
    print $aba->GetState() . "\n";
    print $aba->GetZip() . "\n";
    print $aba->GetPhone() . "\n";
}
```

C++

```
?>
#include "claba.h"

clAba *aba = new clAba; // dynamic allocation

if (aba->IsValid(number))
{
    // the number passed basic validation
    printf("%s\n",aba->GetAba());
    printf("%s\n",aba->GetRoutingSymbol());
    printf("%s\n",aba->GetRoutingInstitution());
    printf("%s\n",aba->GetName());
    printf("%s\n",aba->GetAddress());
    printf("%s\n",aba->GetCity());
    printf("%s\n",aba->GetState());
    printf("%s\n",aba->GetZip());
    printf("%s\n",aba->GetPhone());
}
```

ABA Routing Number Database

This database contains approximately 28,000 records. For all versions of the validation object it is supplied as an external file which should be placed in the same working directory as the object script or application.

The database is delivered in standard ASCII CSV (comma separated value) format with each line (record) terminated with a single carriage return and line feed combination.

Routing

The nine digit ABA routing transit number of the bank or financial institution.

Bank

The name of the bank, credit union, or other financial institution.

MainAddress

The mailing or physical address of the main branch of the bank or financial institution.

MainCity

The city of the main branch of the bank or financial institution.

MainState

The state location of the main branch of the bank or financial institution.

MainZip

The five-digit zip code of the main branch of the bank or financial institution.

MainPhone

The complete telephone number including area code of the main branch of the bank or financial institution.