CHILKAT C++ EXAMPLES

Chilkat • HOME • Android™ • Classic ASP • C • C++ • C# • Mono C# • .NET Core C# • C# UWP/WinRT • DataFlex • Delphi ActiveX • Delphi DLL • Visual FoxPro • Java • Lianja • MFC • Objective-C • Perl • PHP ActiveX • PHP Extension • PowerBuilder • PowerShell • PureBasic • CkPython • Chilkat2-Python • Ruby • SQL Server • Swift 2 • Swift 3,4,5... • Tcl • Unicode C • Unicode C++ • Visual Basic 6.0 • VB.NET • VB.NET UWP/WinRT • VBScript • Xojo Plugin • Node.js • Excel • Go

C++ Examples

Web API Categories

ASN.1
Amazon EC2
Amazon Glacier
Amazon S3
Amazon S3 (new)
Amazon SES
Amazon SNS
Amazon SQS
Async
Azure Cloud Storage
Azure Service Bus
Azure Table Service
Base64
Bounced Email
Box
CAdES
CSR
CSV
Certificates
Compression
DKIM / DomainKey
DSA
Diffie-Hellman
Digital Signatures
Dropbox
Dynamics CRM
ECC
Ed25519
Email Object
Encryption
FTP
FileAccess
Firebase
GMail REST API
GMail
SMTP/IMAP/POP
Geolocation
Google APIs
Google Calendar
Google Cloud SQL
Google Cloud Storage
Google Drive
Google Photos
Google Sheets
Google Tasks

Gzip
HTML-to-XML/Text
HTTP
HTTP Misc
IMAP
JSON
JSON Web Encryption (JWE)
JSON Web Signatures (JWS)
JSON Web Token (JWT)
Java KeyStore (JKS)
MHT / HTML Email
MIME
MS Storage Providers
Microsoft Graph
NTLM
OAuth1
OAuth2
Office365
OneDrive
OpenSSL
Outlook
PEM
PFX/P12
POP3
PRNG
REST
REST Misc
RSA
SCP
SFTP
SMTP
SSH
SSH Key
SSH Tunnel
SharePoint
Socket/SSL/TLS
Spider
Stream
Tar Archive
Upload
WebSocket
XAdES
XML
XML Digital Signatures
XMP
Zip
curl

(C++) Getting Started Spidering a Site

This is a very simple "getting started" example for spidering a web site. As you'll see in future examples, the Chilkat Spider library can be used to c single site.

Chilkat C/C++ Library Downloads

- MS Visual C/C++
- Linux/CentOS C/C++
- Alpine Linux C/C++
- MAC OS X C/C++
- armhf/aarch64 C/C++
- C++ Builder
- iOS C/C++
- Android C/C++
- Win Mobile 5.0/Pocket PC 2003

```
#include <CkSpider.h>

void ChilkatSample(void)
    {
    CkSpider spider;

    //  The spider object crawls a single web site at a time.  As you'll see
    //  in later examples, you can collect outbound links and use them to
    //  crawl the web.  For now, we'll simply spider 10 pages of chilkatsoft.com
    spider.Initialize("www.chilkatsoft.com");

    //  Add the 1st URL:
    spider.AddUnspidered("http://www.chilkatsoft.com/");

    //  Begin crawling the site by calling CrawlNext repeatedly.
    int i;
    for (i = 0; i <= 9; i++) {
        bool success;
        success = spider.CrawlNext();
        if (success == true) {
            //  Show the URL of the page just spidered.
            std::cout << spider.lastUrl() << "\r\n";
            //  The HTML is available in the LastHtml property
        }
        else {
            //  Did we get an error or are there no more URLs to crawl?
            if (spider.get_NumUnspidered() == 0) {
                std::cout << "No more URLs to spider" << "\r\n";
            }
            else {
                std::cout << spider.lastErrorText() << "\r\n";
            }

        }

        //  Sleep 1 second before spidering the next URL.
        spider.SleepMs(1000);
    }


    }
```