

Microsoft IIS HttpModule (on Windows)

This uses a managed language and uses the managed language implementation of the VCDiff engine. An ISAPI version written in a native language would be better – expected to be faster.

Microsoft are now trying to make ISAPI unattractive to developers and even turning off some of the functionality that has been abused (and slowed down IIS), and instead at IIS7 is encouraging developers to use HttpModule and provide examples how to create them with native code: <http://msdn.microsoft.com/en-us/library/ms689320>.

Microsoft IIS ISAPI (on Windows)

This is an IIS ISAPI extension wrapper written in native language (Free Pascal) on Windows and uses the native language VCDiff engine. It uses the following IIS ISAPI notifications:

- PreProcHeaders
- SendResponse
- SendRawData
- EndOfNetSession.

Compliance

<i>Feature</i>	<i>Implemented</i>
Chunked Encoding provider	No
Content-Length provider	Yes
Chunked Encoding consumer	Yes
Content-Length consumer	No
IIS 5.1 (Windows XP)	Yes
IIS 6 (Windows 2003 Server)	Yes
IIS 7 (Windows 2008 Server)	not known
SDCH	Yes
jetSTREAM	Yes

The ISAPI supports both SDCH and jetSTREAM in a tightly coupled way.

ALB ISAPI (on Linux)

This development required some integration with ALB and in particular the ALB ISAPI interface.

ALB implements a variant of ISAPI that supports the following notifications:

- PreProcHeader
- SendRawData
- EndOfRequest

PreProcHeader and EndOfRequest have no return value and SendRawData returns true if the method was successful and false if not.

ALB ISAPI passes data to SendRawData in 16K blocks. this is because of receive buffer size configuration.

The output of SendRawData is rendered if cbInData is set to a value greater than 0 (and pvInData contains some text – it is a string).

Client Components

The client is responsible for making a jetSTREAM or SDCH request with SHA256's of dictionaries it has and then correctly processing the response and rendering a web page. There are several components in this area

HttpClient

This was the first client component developed and was developed in a managed language (C#) and was developed to complement the server-side HttpModule(also in C#) and uses the managed implementation of the Core VCDiff engine

Decoder API

This is developed as a native language DLL on Windows. It uses the native language VCDiff engine. It was designed as a wrapper that would provide decoding features for a third party developer to use to implement a browser helper object or plug-in on Windows. This implements a documented Decoder API that can decode both jetSTREAM and SDCH.

DictionaryAPI

This was developed as another native language DLL on Windows. It was designed as a wrapper that would provide dictionary management features for a third party developer to use to implement a browser helper object or plug-in. This implements a documented Decoder API that can manage both jetSTREAM and SDCH dictionaries.

ClientAPI

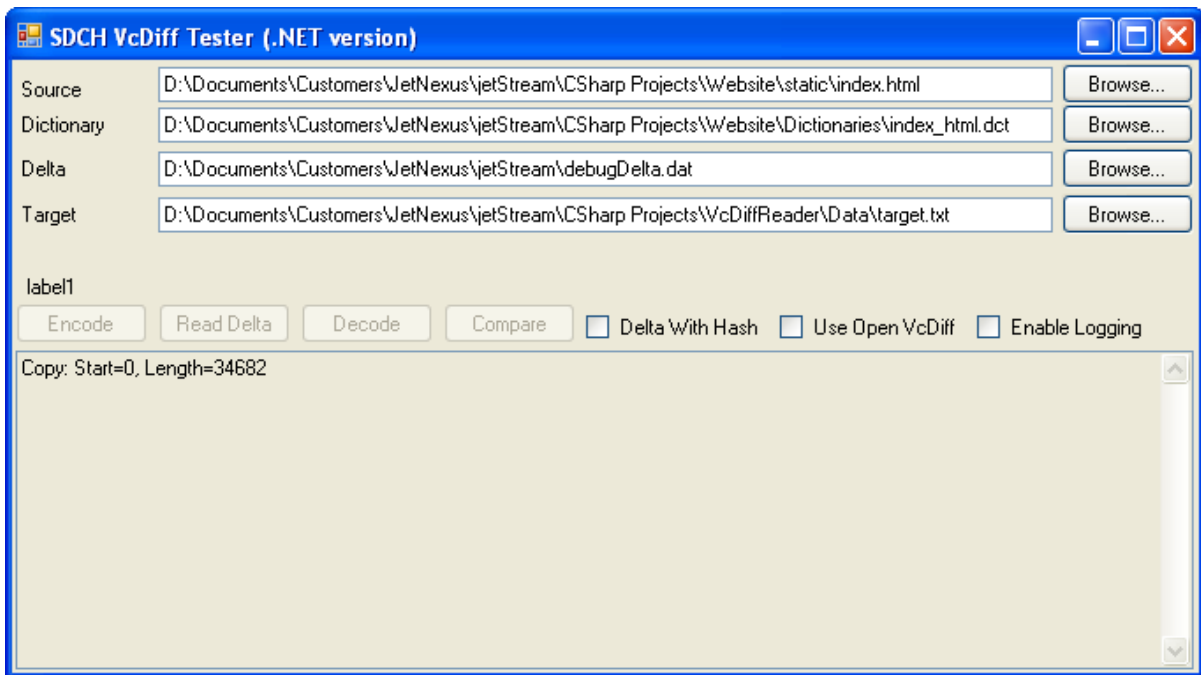
It quickly became clear that the Decoder API would need to talk to the "Dictionary API" to retrieve the dictionary needed to decode a compressed webpage. We combined the two interfaces into a single DLL, so that they could communicate internally. This became the "Client API", which is a composite of both the Decoder API and the Dictionary API.

Other documents give a full explanation of the API.

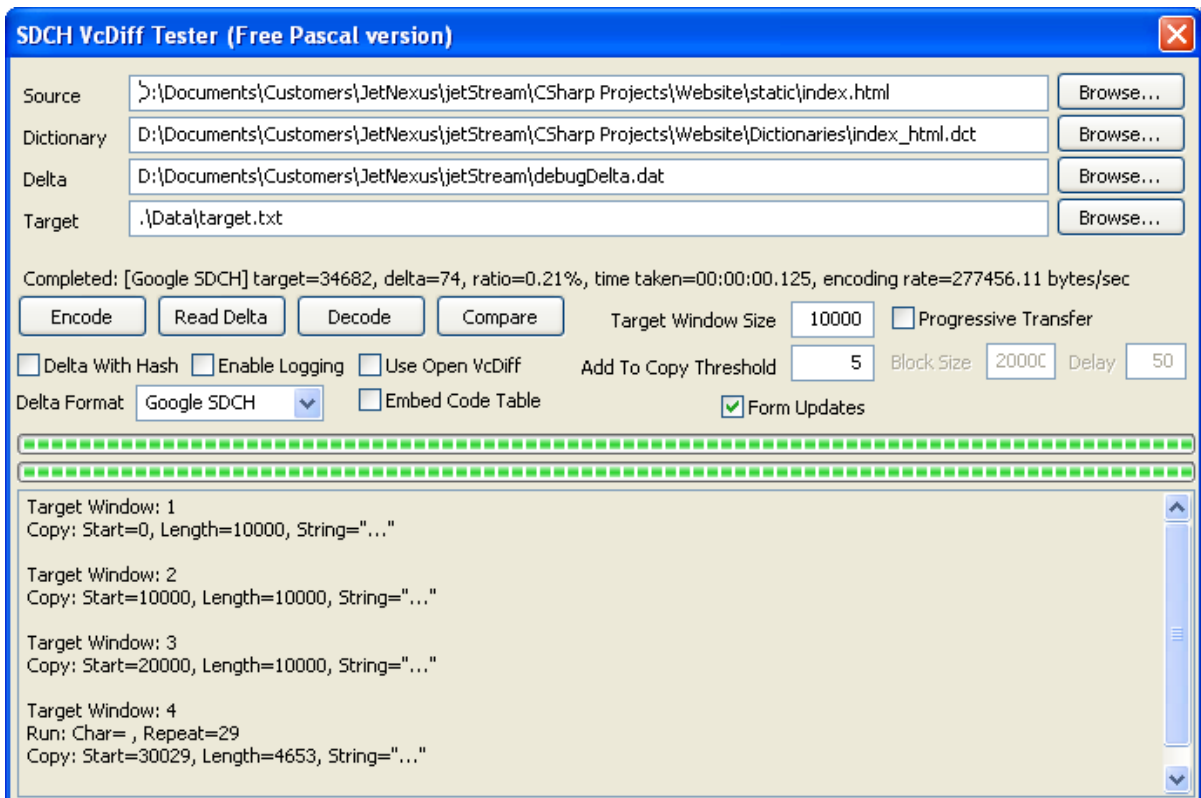
Test Components

VCDIFF Reader

This was first developed as a managed code form based application (in C#) using the managed code VCDiff engine.



This was later rewritten as a native code form based application (in Free Pascal) using the native code VCDiff engine (in fact the figure above shows that the managed version does not support the many options available in the native version).



Both are still available, although the managed version possibly is missing some features of the native version.

ALB ISAPI Host

This is a native code forms based application which was developed to host, debug and exercise the jetSTREAM ALB ISAPI extension.

The screenshot displays the 'ALB ISAPI Host' application window. The interface is divided into several sections:

- Request Headers:** Shows headers for the request, including 'Host: localhost', 'Accept-Encoding: deflate,gzip,jetstream', 'X-jetStream: 1.0', and 'X-jetStream-Dictionary-Hashes: 8wrVzGBO'.
- Response Headers:** Shows headers for the response, including 'HTTP/1.1 200 OK', 'Server: Microsoft-IIS/5.1', 'Date: Tue, 24 May 2011 20:08:24 GMT', 'X-Powered-By: ASP.NET', 'X-AspNet-Version: 2.0.50727', 'Accept-Ranges: bytes', 'Cache-Control: public', 'Expires: Fri, 03 Jun 2011 20:08:24 GMT', 'Last-Modified: Sat, 14 May 2011 20:08:24 GMT', 'ETag: "1CA7F1045C19800"', 'Content-Type: text/html', 'Cache-Control: no-cache', 'Content-Encoding: jetstream', 'X-jetStream:: 1.1', and 'Transfer-Encoding: chunked'.
- Uncompressed Page (Text):** Displays the raw HTML content of the response, including DOCTYPE, meta tags, title, description, keywords, and CSS/JavaScript links.
- Compressed (Hex):** Shows the hexadecimal representation of the compressed response data, including headers, chunked data, and a delta.

TestHttpClient

This is a managed code form based client with a browser embedded within it. This is the only currently available jetSTREAM client. It was originally developed to only use the managed code decoder (HttpClient), but has since been extended to use the native code ClientApi (Decoder and Dictionary). The engine is selected using a dropdown at the top of the form.

The client will work with SDCH and jetSTREAM and will interwork with these protocols using the IIS ISAPI implemented (on an IIS) or will interwork with the ALB ISAPI implementation (on an ALB).