OneTimeDownload-5M

Link Management for ASP.Net

Version 2

for Visual Studio .NET

Desaware, Inc.

Rev 2.0.0 (06/06)

Desaware, Inc.
Software License

Please read this agreement. If you do not agree to the terms of this license, promptly return the product and all accompanying items to the place from which you obtained them.

This software is protected by United States copyright laws and international treaty provisions.

This program will be licensed to you for use only on a single computer. If you wish to install it on additional computers, you must purchase additional software licenses. You may (and should) make archival copies of the software for backup purposes.

You may not make copies of this software for other people. Companies or schools interested in multiple copy licenses or site licenses should contact Desaware, Inc. directly at (408) 404-4760.

You have a royalty-free right to incorporate any of the sample code provided into your own applications with the stipulation that you agree that Desaware, Inc. has no warranty, obligation or liability, real or implied, for its performance.

Licensing: This software is licensed for use on a single computer.

File Descriptions: You may not redistribute any of the files included with the OneTimeDownload-5M software.

Source Files: If you have purchased a source code license, the following applies: You may rebuild modified versions of the software provided subject to the restrictions listed. You may not use this source code to develop or distribute components and tools that provide functionality similar to all or part of the functionality provided by the OneTimeDownload-5M software or any of its components except for use in your own applications. Modified assemblies and namespaces must be renamed – you may not use Desaware in the assembly name or any namespace. However Desaware's copyright notice must be prominently displayed in any location where your own copyright notice is present. Source code may not be published or distributed, and may be used or accessed only by the individuals and at the locations covered by the source code license. You may distribute modified versions of the Desaware.Onetimedownload.Dll file only for use by your applications. Modified versions of other components may only be used by the individuals and at the locations covered by the source code license.

Limited Warranty

Desaware, Inc. warrants the physical CD and physical documentation enclosed herein to be free of defects in materials and workmanship for a period of sixty days from the date of purchase.

The entire and exclusive liability and remedy for breach of this Limited Warranty shall be limited to replacement of defective CD(s) or documentation and shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data or use of the software, or special, incidental or consequential damages or other similar claims, even if Desaware, Inc. has been specifically advised of the possibility of such damages. In no event will Desaware, Inc.'s liability for any damages to you or any other person ever exceed the suggested list price or actual price paid for the license to use the software, regardless of any form of the claim.

DESAWARE, INC. SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Specifically, Desaware, Inc. makes no representation or warranty that the software is fit for any particular purpose and any implied warranty of merchantability is limited to the sixty-day duration of the Limited Warranty covering the physical CD and documentation only (not the software) and is otherwise expressly and specifically disclaimed.

This limited warranty gives you specific legal rights. You may have others, which vary from state to state.

This License and Limited Warranty shall be construed, interpreted and governed by the laws of the State of California, and any action hereunder shall be brought only in California. If any provision is found void, invalid or unenforceable it will not affect the validity of the balance of this License and Limited Warranty, which shall remain valid and enforceable according to its terms.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of Commercial Computer Software - Restricted Rights at 48 CFR 52.227-19, as applicable. Contractor/Manufacturer is Desaware, Inc., 3510 Charter Park Drive, Suite 48, San Jose, CA 95136.

Table of Contents

OneTimeDownload-5M –Link Manager for ASP.Net

Introduction

OneTimeDownload-5M is a component that allocates, implements and manages temporary links. These are URL's that are active for a limited amount of time, and are often uniquely associated with individual users. Typical scenarios for this type of link include:

- Implementing special offers or discounts – where each customer receives a unique link that expires after a set time.

- Software or document distribution – where a link expires after a pre-determined time, or a limited time after first use.

- Select one file or document – where a group of links are provided, and all other links become invalid once one is selected.

- Email campaign tracking – where each email has a unique link, making it easy to track responses to specific outgoing messages.

Features

- Create unlimited unique links, each with a specified expiration date.

- Specify the time a link remains active after first use.

- Links are associated with a URL, file path, and optional description or user defined data.

- Links can be grouped. Use of one link invalidates the others in the group.

- Highly secure. 128 bit link identifiers virtually eliminate the possibility of guessing a link.

- Specify IP limits to increase security. Once a link is used by the specified number of IP addresses, only those IP addresses can continue to access it.

- Document/file download links. Supports direct download for common file types.

As with all software in Desaware's 5 Minute Software line, the OneTimeDownload-5M component is easy to learn, use and deploy. Many developers will find the base component sufficient for most tasks, however a source code license is available for those who need additional customization.

Learning to Use OneTimeDownload-5M

The Desaware.OneTimeDownload.Dll component contains a single class named Desaware.OneTimeDownload.LinkManager.

Installation

Use the OneTimeDownload11.msi or OneTimeDownload20.msi installation software to install the OneTimeDownload software on your system.

Samples

The FreeEBookExample sample application is the one we used for a Desaware promotion. We sent a link for a free eBook. The link directed people to a page where they could choose from among the available ebooks. Once an ebook was chosen, the links to the other ebooks became invalid.

This application demonstrates most of the features of the OneTimeDownload software, and includes source code.

To install the sample application, click on the Start Menu and select the FreeEbookExample installation link. Refer to the FreeEbookExample section of this document or the readme file for information about configuring the sample program.

Licensing

The OneTimeDownload component is licensed for each machine. Once licensed on a system, you can use the component on as many web applications as you wish on that system.

LinkManager Class Quick Start

The LinkManager class is designed to be used directly by any ASP.Net web application or web service. The class is easy to use – the only part that can be tricky is creating the database correctly.

The LinkManager class constructor takes two parameters: a connection string and an enumeration indicating the database provider to use (oledb, sql or ODBC). You must ensure that the connection string is correct and can be used by ASP.Net. The LinkManager class includes the method, VerifyDB, which verifies the connection string, and confirms that the database has been initialized correctly for use by the LinkManager (creating the necessary table if it is not present).

This document assumes that you already know how to configure a database for use with ASP.Net. For those who are not familiar with this task, a later section will detail the process of creating an Access database, which is adequate for smaller sites.

Creating the LinkManager Object

Create an instance of the LinkManager class by passing it a connection string and a member of the DatabaseProvider enumeration. The following code illustrates how to create an instance of the object using OleDb:

[VB]

```
Dim otd As LinkManager

otd = New LinkManager(connstring, DatabaseProvider.OleDb)
```


[C#]

```
LinkManager otd;


otd = new LinkManager(connstring, DatabaseProvider.OleDb);
```

Creating a URL or File Download Link

To create a single link to a specified file or URL, use the CreateLink method as shown here:

[VB]

```
Dim link As String

link = otd.CreateLink(FileOrUrl, Description, ExpirationDate,
ActiveHours, MaxIps)
```

[C#]

```
string link;

link = otd.CreateLink(FileOrUrl, Description, ExpirationDate,
ActiveHours, MaxIps);
```

ActiveHours is the number of hours a link remains valid after first use. MaxIps is the number of unique IP addresses allowed to access the link.

Creating a Mutually Exclusive Set of Links

To create a group of links, where use of one link will invalidate the others, first create a parent link. Then create a group of links that specify that parent as shown here:

[VB]

```
Dim mainlink, childlink1 As String

mainlink = otd.CreateLink(FileOrUrl, Description, ExpirationDate,
ActiveHours, MaxIps)

childlink1 = otd.CreateLink(FirstChoice, Description, ExpirationDate,
ActiveHours, MaxIps, mainlink, ExclusionSet)
```

[C#]

```
string mainlink, childlink1;

mainlink = otd.CreateLink(FileOrUrl, Description, ExpirationDate,
ActiveHours, MaxIps);

childlink1 = otd.CreateLink(FirstChoice, Description, ExpirationDate,
ActiveHours, MaxIps, mainlink, ExclusionSet);
```

The mainlink parameter defines the parent link for this set. The ExclusionSet parameter defines the set of links from which to choose.

As soon as you access a link, all other links that share the same mainlink and ExclusionSet values become invalid.

Creating a URL From a Link Identifier

The CreateLink method returns a link identifier (in the form of a GUID string). There are three methods you can use to obtain a URL for this link identifier. Of those, the GetSimpleLink function is the easiest.

[VB]

```
url = GetSimpleLink(LinkId, BaseURL)
```

[C#]

```
url = GetSimpleLink(LinkId, BaseURL);
```

LinkId is the value returned from the CreateLink method. BaseURL is the base URL for the web application. If the BaseUrl parameter ends with ".asp", ".aspx", ".htm" or ".html", the link is build directly using the link ID.

For example:

```
BaseUrl = "http://www.5MinuteSoftware.com/linkhandler.aspx"

LinkId = "bff0d253-b924-4653-afa7-ac9341eab17e"

Result = "http://www.5MinuteSoftware.com/linkhandler.aspx?LinkID=
bff0d253-b924-4653-afa7-ac9341eab17e"
```

If the BaseUrl parameter ends with any other extension, the link will be build to the otd.aspx web page (included with the product).

```
BaseUrl = "http://www.5MinuteSoftware.com"

LinkId = "bff0d253-b924-4653-afa7-ac9341eab17e"

Result = "http://www.5MinuteSoftware.com/otd.aspx?LinkID=bff0d253-b924-
4653-afa7-ac9341eab17e"
```

The GetALink method works the same way, except that it also accepts a Description parameter and it returns the URL in the form of a <A> reference tag.

The GetHyperLink method also accepts a Description parameter, and it returns the URL in the form of a System.Web.UI.WebControls object.

Validating and Retrieving the File or URL For a Link

When your ASP.Net application receives a request, you can retrieve the File or URL for that link as follows:

[VB]

```
Dim ls As LinkStatus
Try
    link = Request.Item("LinkID").ToString()
    filepath = otd.GetFilePath(link, Request.UserHostAddress, ls)
Catch
    ' Handle the error here
End Try
```

[C#]

```
LinkStatus ls;
try
{
    link = Request["LinkID"].ToString();
    filepath = otd.GetFilePath(link, Request.UserHostAddress, ref ls);
}
catch (Exception ex)
{
    // Handle the error here
}
```

The LinkStatus parameter is set to one of the following values:

```
Valid = 0
Expired = 1
Excluded = 2
Invalid = 3
IpLimited = 4
Exception = 5
```

Be sure to pass the UserHostAddress as the second parameter in order to allow IP tracking to work.

Finding the Child Links for a Link

Sometimes a link refers to a specific URL, in which case the preceding example applies. However, sometimes a link directs users to a page which contains additional links. In this case you'll need to obtain information about the child links that are available for a parent. This is accomplished as follows:

[VB]

```
Dim ChildLinks As DataSet

Dim ls As LinkStatus

ChildLinks = otd.GetChildLinks(parentlink, Request.UserHostAddress, ls)
```

[C#]

```
DataSet ChildLinks;

LinkStatus ls;

ChildLinks = otd.GetChildLinks(parentlink, Request.UserHostAddress, ref
ls);
```

The returned DataSet contains a single table, with a row for each valid child link. The columns are as follows:

LinkID	The link identifier for the child link.
Path	The File or URL for the child.
Description	The description of the child link.

Refer to the Database Schema appendix for details on additional columns.

The GetChildLinks method counts as a request against the parent link (i.e., it records the first request, the IP, etc.), but does NOT count as a request against the child links.

Sending a File

The easiest way to handle links is to use the otd.aspx page provided with this product (source code for the page is included as part of the FreeEbook sample project).

[VB]

```
Try

    mainlink = Request.Item("LinkID").ToString()


    Dim otd As New LinkManager(connstring, linktype)


    Dim ls As LinkStatus

    Dim filepath As String

    filepath = otd.GetFilePath(mainlink, Request.UserHostAddress, ls)
```

```vbnet
        If Not ls = LinkStatus.Valid Then

            ErrorTransfer(ls)      End If


        otd.SendTheFile(filepath, Response)
    Catch


    End Try



    Private Sub ErrorTransfer(ByVal ls As LinkStatus)
        Session.Item("OTDLinkStatus") = ls
        Server.Transfer("LinkInvalidChoice.aspx", False)
    End Sub
```

[C#]
```csharp
try
{

    mainlink = Request["LinkID"].ToString();


    LinkManager otd = new LinkManager(connstring, linktype);


    LinkStatus ls;
    string filepath;
    filepath = otd.GetFilePath(mainlink, Request.UserHostAddress, ref
ls);


    if (ls != LinkStatus.Valid)
    {
            ErrorTransfer(ls);
    }
    otd.SendTheFile(filepath, Response);
}
catch (Exception ex)
{
}


private void ErrorTransfer(LinkStatus ls)
```

```
{
        Session.Item("OTDLinkStatus") = ls;

        Server.Transfer("LinkInvalidChoice.aspx", false);
}
```

Configuring the FreeEbookExample Sample Program

The FreeEbookExample project demonstrates most of the features of the OneTimeDownload component.

We gave some thought into whether to completely automate the installation, but decided it would be a better experience to actually require you to build and explicitly configure the web application.

Demo Installation

To install and configure the FreeEbookExample sample program, do the following:

1. Be sure your ASP.Net compatible web server is running.

2. Using the Program/Start menu, install the FreeEbookExample.msi installer file.

3. Open the project file. Typically http://LocalHost/FreeEbookExample/FreeEbookExample.vbproj.

4. Edit the web.config file. The DataSource field in the connection string should include the full path to the linkdb.mdb database. You can specify an alternate connection string if you wish.

5. From Windows Explorer, right click on the DB directory and select the security tab. Add the ASPNET user (Use NETWORK SERVICE if no ASPNET user is found on XP systems, use the IIS_WPG group on Windows 2003 server).

6. You should now be able to run the application. Set the default page when prompted to CreateNewLink.asmx.

7. Though it is not required for this application, we do recommend you secure the CreateNewLink.asmx web service by either adding authentication, or using an IP filter.

Things to Try:

1. Run the project. This will bring up the CreateNewLink page. Click on the GetBookLink hyperlink, then Invoke the method. You should receive a result that looks something like this:
   ```
   <string xmlns="http://OneTimeDownload.desaware.com/"><a
   href="http://localhost/freeEbookExample/Bookchoice.aspx?LinkID=
   4e29f19e-8312-4fec-b552-4812f1d67907">4e29f19e-8312-4fec-
   b552-4812f1d67907</a></string>
   ```

2. Copy the link URL, in this case
   `http://localhost/freeEbookExample/Bookchoice.aspx?LinkID=4e29f19e-8312-4fec-b552-4812f1d67907`
   and paste it into the web browser address box.

3. Experiment with the site navigation. See what happens when you click on a sample book, navigate back, try to reuse the link, or enter an invalid link.

Reference

The following is detailed reference information for the Desaware OneTimeDownload classes. Properties and methods of base classes are not listed unless they are overridden.

Note on thread safety: Unless otherwise noted, as is common in the .NET Framework, static methods of objects are thread safe. Instance methods are not.

Enum DatabaseProvider

This enumeration is used to indicate the .NET provider to use with a connection string.

0	SqlClient
1	OleDB
2	ODBC

Generally speaking, the ODBC option will only be valid on the 1.1 .NET framework.

Enum LinkStatus

This enumeration is used by several methods to return the current status of a link.

Value	Status	Definition
0	Valid	The link is still valid.
1	Expired	The link has expired.
2	Excluded	The link is blocked because another link with the same parent link and exclusion group has already been used.
3	Invalid	The link is not valid.
4	IpLimited	The maximum number of unique IP addresses allowed to use this link has been exceeded.
5	Exception	An exception occurred while processing the request.

Class LinkManager

This class implements the functionality of the OneTimeDownload-5M software. It is used to define, manage and process links.

Fields:

MimeMap	**[VB]** MimeMap As String(,)

	[C#] `string[,] MimeMap` This is a two dimensional array defining MIME types and extensions used by the SendTheFile method. The first column is the extension in the form ".xxx". The second column is the mimetype. For example: {".jpg", "image/jpeg"} You can replace the default list.

Methods:

Constructor	**[VB]** `Sub New(ByVal ConnectionString As String, ByVal ConnectionType As DatabaseProvider)` **[C#]** `void LinkManager (string ConnectionString, DatabaseProvider ConnectionType)` The constructor for the LinkManager class requires a valid database connection string and a DatabaseProvider enumeration that specifies the type of database provider to use. Exceptions: Licensing exceptions will be raised if the component is not licensed for this system.
VerifyDB	**[VB]** `Sub VerifyDB()` **[C#]** `void VerifyDB()` This method verifies that the database can be accessed and that the Links table is defined. It will attempt to create the Link table if it is not yet present. Exceptions: Exceptions raised by the database provider will be forwarded to the caller.
CreateLink	**[VB]** `Function CreateLink(ByVal FilePath As String, ByVal Description As String, ByVal ExpirationDate As Date, ByVal ActiveHours As Integer, ByVal MaxIPs As Integer) As String`

Page 15

	[C#] `string CreateLink (string FilePath, string Description, DateTime ExpirationDate, int ActiveHours, int MaxIPs)` This function creates a new link and returns the link identifier string. Parameters: FilePath – Typically the path of the file or URL associated with the link. This can actually be any string of up to 255 characters that you wish to associate with the link. Description – A description of the link. This can actually be any string of up to 255 characters that you wish to associate with the link. ExpirationDate – The absolute date at which this link expires. ActiveHours – The number of hours the link remains active after the first time it is used. MaxIPs – The maximum number of unique IP addresses that can access this link. Exceptions: Exceptions raised by the database provider will be forwarded to the caller.
CreateLink	**[VB]** `Function CreateLink(ByVal FilePath As String, ByVal Description As String, ByVal ExpirationDate As Date, ByVal ActiveHours As Integer, ByVal MaxIPs As Integer, ByVal ParentID As String, ByVal ExclusionSet As Integer) As String` **[C#]** `string CreateLink (string FilePath, string Description, DateTime ExpirationDate, int ActiveHours, int MaxIPs, string ParentID, int ExclusionSet)` This overload is identical to the previous one, with the following additional parameters: ParentID – The link identifier of a valid link which will act as a "parent" to this one. Functions are available to obtain a list of child links for a parent, so this provides a way to group links. ExclusionSet – If greater than zero, this link belongs to an exclusion set. As soon as one link in an exclusion set is accessed, all other links in the same exclusion set that share the same parent link become invalid and will return a link status of

	excluded.
	Exceptions will be raised if an invalid or expired parent is specified, or if a parent is specified that itself has a parent (nested parents are not allowed).
GetALink	**[VB]** `Function GetALink(ByVal LinkId As String, ByVal Description As String, ByVal BaseURL As String) As String` **[C#]** `string GetALink (string LinkId, string Description, string BaseURL)` This function formats a link identifier into an <a> tag. The result will be roughly in the form: *Description* The BaseURL is the URL for the link. If empty, the function will use otd.aspx. If BaseURL does not reference an ASP, ASPX, HTM or HTML page, the function will append otd.aspx to the URL. The description will be HTML encoded if necessary.
GetSimpleLink	**[VB]** `Function GetSimpleLink(ByVal LinkId As String, ByVal BaseURL As String) As String` **[C#]** `string GetSimpleLink (string LinkId, string BaseURL)` Returns a simple link in the form: *BaseURL?LinkID=linkid* The BaseURL parameter is handled as described for the GetALink method.
GetHyperLink	**[VB]** `Function GetHyperLink(ByVal LinkId As String, ByVal Description As String, ByVal BaseURL As String) As HyperLink` **[C#]** `string GetHyperLink (string LinkId, string Description, string BaseURL)` This method is identical to the GetALink method, except that it

	returns the link as a System.Web.UI.WebControls.HyperLink object.
GetFilePath	**[VB]** `Function GetFilePath(ByVal LinkId As String, ByVal Ip As String, ByRef ls As LinkStatus) As String` **[C#]** `string GetFilePath (string LinkId, string Ip, ref LinkStatus ls)` This method retrieves the FilePath specified when the link was created. You should pass the calling IP as the IP parameter in order for IP tracking to occur. The ls parameter is set to the status of this link. The FilePath will only be returned if the link status is valid. This method counts as a request for the link. The first time this method is called for a link, the request date is recorded. The expiration date is updated based on the number of hours the link is supposed to be active. If this link has a parent, and its exclusion set is greater than zero, all other links with the same parent and exclusion set immediately become blocked.
SendTheFile	**[VB]** `Sub SendTheFile(ByVal path As String, ByVal response As Web.HttpResponse)` **[C#]** `void SendTheFile (string path, System.Web.HttpResponse response)` This method sends the file to the Response object provided. For document types specified in the MimeMap field (.jpg, .doc, .zip, .tif, .bmp, .rtf, .pdf), calling this function sends the file and stops processing the page. The method uses the MimeMap field to set the Response ContentType.
GetChildLinks	**[VB]** `Function GetChildLinks(ByVal ParentId As String, ByVal RequestingIP As String, ByRef ls As LinkStatus) As DataSet` **[C#]** `DataSet GetChildLinks (string ParentId, string RequestingIP, ref LinkStatus ls)`

	This method returns a dataset containing information about any child links for the specified parent link. The returned DataSet contains a single table, with a row for each valid child link. The columns are as follows: LinkID – The link identifier for the child link Path – The File or URL for the child Description – The description of the child link Refer to the Database Schema appendix for details on additional columns. This method counts as a request for the parent link. The first time this method is called for a link, the request date is recorded. The expiration date is updated based on the number of hours the link is supposed to be active. If this link has a parent, and its exclusion set is greater than zero, all other links with the same parent and exclusion set immediately become blocked. This method does not count as a request for the child links.
DeleteExpiredLinks	**[VB]** `Sub DeleteExpiredLinks()` **[C#]** `void DeleteExpiredLinks()` This method deletes any expired links from the database. You should call this periodically to prevent the database from becoming cluttered with old data.

Security

The LinkManager component is implemented in managed code. It will support partial trust components.

Partial trust components must have the following permissions:

- The appropriate database access permissions.
- File access permissions if required by the database.

Licensing

The Desaware OneTimeDownload-5M toolkit is available in two different licenses.

- The Standard license requires a separate license for each system on which the component is installed.

- The Source license includes a site license for all developers in a single geographical location (one street address). The following terms apply:

  o You may modify the source code and distribute modified components for use in installing your own applications subject to the limitations that follow.

  o You may not sell, market or relicense modified components, or components or tools that incorporate our source code, for use by other developers to develop their own installations. In other words – you can't modify our code, then sell a product that competes with this toolkit in any way. You can only use modified versions of our code to facilitate the installation of your own, non competing, products.

  o If you modify our components, you must change the assembly name and all namespaces so that they do not include the word Desaware.

  o Your assembly.vb file must be modified to your own product name.

  o Your assembly.vb file must, as part of its copyright notice, include "Portions of this code are copyright ©2006 by Desaware Inc."

Unlike other Desaware components, OneTimeDownload-5M does not include a redistribution license. If you are a consultant or solution provider who wishes to license this component as part of a complete solution, please contact Desaware.

Appendix

Database Schema

The following schema is used for the OneTimeDownload-5M database. You won't need this information for normal use, however it can be useful if you wish to generate reports of link activity.

The database contains a single table named Links

Column Name	Description
ID	Autonumber field. Not used by the component.
LinkID	Link identifier string. Takes the form of a standard GUID string.
Path	The URL or file path associated with the links. Up to 255 characters.
ParentID	The identifier string of the parent if any. Will be null if there is no parent.
Description	A text description of the link. Up to 255 characters.

ExclusionSet	The number of the exclusion set to which this link belongs. Ignored if less than or equal to zero.
Excluded	True if this link has been excluded due to being part of an exclusion set in which another link has been used.
ValidUntil	The expiration date for this link.
ActiveTime	The number of hours a link is valid after it is used for the first time.
FirstRequest	The time and date of the first request for this link.
RequestingIPs	A semicolon delimited list of IP addresses used to request this link.
MaxIPs	The maximum number of IP addresses that may be used to request this link.