

Trusted Proxy Server  
Installation Guide

Version 2.0.1

[1 Installation 2](#_Toc415068035)

[1.1 Overview 2](#_Toc415068036)

[2 Deploying TPS - checklist 3](#_Toc415068037)

[2.1 Completing the TPS Configuration form 3](#_Toc415068038)

[2.2 TPS Customer Configuration Form 4](#_Toc415068039)

[3 Sample Scripts 5](#_Toc415068040)

[3.1 JSP 6](#_Toc415068041)

[3.2 PERL 7](#_Toc415068042)

[3.3 PHP 8](#_Toc415068043)

[3.4 ColdFusion 9](#_Toc415068044)

[3.5 ASP 10](#_Toc415068045)

[3.6 ASP.NET Version 1 VB 11](#_Toc415068047)

[3.7 ASP.NET Version 1 C# 12](#_Toc415068049)

[3.8 ASP.NET Version 2 VB 13](#_Toc415068050)

[3.9 ASP.NET Version 2 C# 14](#_Toc415068052)

[4 Installing the code and deploying the link 15](#_Toc415068053)

[5 Technical notes 16](#_Toc415068054)

[5.1 TPS Link and Parameters 16](#_Toc415068055)

[5.1.1 Linking to publication homepages via DOI 16](#_Toc415068056)

[5.1.2 Linking to other pages 17](#_Toc415068057)

[5.1.3 Domain Parameter 18](#_Toc415068058)

[5.2 Hosting environments 18](#_Toc415068059)

This guide aims to guide webmasters through the installation of the Wiley Online Library Trusted Proxy Server (TPS) authentication script.

# Installation

Deploying TPS into a Society or Institution (Client) website involves inserting a small script that handles the request and response process.

It is the responsibility of the customer to integrate script into their web site.

Wiley will provide support, and advise the customer during the planning and deployment process.

## Overview

Example code is provided for the following environments:

• JSP

* Perl
* PHP
* Coldfusion
* ASP
* ASP.NET Version 1 VB
* ASP.NET Version 1 C#
* ASP.NET Version 2 VB
* ASP.NET Version 2 C#

Other environments can be supported. The security request and response process can be scripted in any common programming language.

The example code provides the methodology of the process, and can be customised as appropriate.

Code may need to be tailored to ﬁt the server environment and website set-up, but the process will be the same.

# Deploying TPS - checklist

1. Complete TPS Customer Configuration form and return to Wiley (Sections 2.1 and 2.2)
2. Customize sample code to suit your hosting environment and target publication (Section 3)
3. Save customized code to your website (Section 4)
4. Create link from members area to execute the script
5. Await confirmation from Wiley that your details have been entered into the system
6. Test the link

## Completing the TPS Configuration form

The customer needs to complete the following steps:

* Find the appropriate page / URL in their site - this is a page that is only presented to authorised members.
* Decide which page on Wiley Online Library you would like to direct your users to (for example, the Journal homepage)
  + Note the page DOI (or URL if not a publication homepage) in the Target Page / DOI section on the TPS Request form (see Section 5.1.1 for more information on the DOI)
* Note your Website domain in the Domain Parameter on the TPS Request form (see Section 5.1.3 for more information on the Domain Parameter)
* Determine / create a default error page for your website, where users will be directed if the TPS process fails due to technical or configuration problems.
  + Note the URL of the default error page and enter it in the Error URL section on the TPS Request form

**Note:** The Error URL should not be any page on Wiley Online Library, as this may confuse users who think they should have access, when the authentication has failed

* Create a test Login ID and Password for the members area on your website, so that Wiley can test the TPS setup to ensure that the pass-through is working as expected.
* Complete the TPS request form and return it to Wiley. *(Section 2.2)*

## TPS Customer Configuration Form

Please complete this form, and return to your contact at Wiley to initiate the setup of the TPS service.

|  |  |
| --- | --- |
| Requested By |  |
| Email Address |  |
| Telephone |  |
| Date |  |

|  |  |
| --- | --- |
| Customer Name |  |
| Server IP address range  This will be the IP address of the server or firewall that sends the TPS access request to Wiley |  |
| Your Website URL |  |
| Target Page / DOI |  |
| Domain Parameter  This must match EXACTLY what you enter in your TPS script as the domain parameter |  |
| Error URL |  |
| Member Website URL  If different from Website URL |  |
| Site Test Username |  |
| Site Test Password |  |

Please provide a test login ID and Password to the members area on your website for testing purposes.

This will enable Wiley to login as a member of your website to test the TPS link.

Use the above parameters to customize the TPS script suitable for your environment.

# Sample Scripts

Replace the code in **orange** with the DOI of the page you wish to direct your users to

Replace the domain(**green**) with the domain parameter in your script

Replace the code in **blue** with the URL or your error page.

* JSP
* Perl
* PHP
* Coldfusion
* ASP
* ASP.NET Version 1 VB
* ASP.NET Version 1 C#
* ASP.NET Version 2 VB
* ASP.NET Version 2 C#

**Common issues when creating scripts:**

* - All scripts(except PERL)

Please ensure that there are no gaps or spaces in the long ‘ticket’ or ‘GET’ URL string:

e.g.

http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http:// onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com

This is one complete string, with no spaces

* - All Scripts

Please ensure the DOI parameter is in the correct format

journal/10.1111/(ISSN)1467-6281

**See section 4 (page 15) for details on how to deploy your customised script**

## JSP

<%@ page language="java" contentType="text/html" import="java.net.\*" import="java.io.\*"%>

<%

  String requestTicketURL = "http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com";

  String str;

  URL url = new URL(requestTicketURL);

  try

  {

    BufferedReader in = new BufferedReader(new InputStreamReader(url.openStream()));

    StringBuffer result = new StringBuffer();

    while ((str = in.readLine()) != null) {

      result.append(str);

    }

    in.close();

    out.println("<html><head><meta http-equiv=\"Refresh\" content=\"0; url=" + result.toString() + "\"></head></html>");

  }

  catch (Exception e)

  {

    out.println("<html><head><meta http-equiv=\"Refresh\" content=\"0; url=http://www.error.com?err=" + e.getMessage() + "\"></head></html>");

  }

%>

## PERL

#!perl

use CGI;

use HTTP::Request;

use LWP::UserAgent;

my $HOST = "http://onlinelibrary.wiley.com";

my $RETURN\_URL="http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281";

my $TICKET\_REQUEST = "$HOST/login-proxy-tps?domain=www.domain.com";

my $URL\_ERROR\_CONNECT = "http://www.error.com";

my $query = new CGI;

my $ua = new LWP::UserAgent;

$TICKET\_REQUEST .= "&targetURL=" . $RETURN\_URL;

my $request = HTTP::Request->new(GET =>$TICKET\_REQUEST);

my $response = $ua->request($request);

print "Content-Type: text/html\n\n";

if ($response->is\_success)

{

  my $url=$response->content;

  print "<html>\n";

  print "<head>\n";

  print "<meta http-equiv=\"Refresh\" content=\"0; url=$url\"/>\n";

  print "</head>\n";

  print "</html>\n";

}

else

{

  my $url=$response->status\_line;

  print "<html>\n";

  print "<head>\n";

  print "<meta http-equiv=\"Refresh\" content=\"0; url=$URL\_ERROR\_CONNECT?err=Error $url\"/>\n";

  print "</head>\n";

  print "</html>\n";

}

## PHP

<?php

function scrrefcustomerror($errornumber, $errormessage, $errorfile, $errorrow)

{

  echo '<meta http-equiv="refresh" content="0;url=http://www.error.com?err=Error code '.$errornumber.' - '.$errormessage.'">';

}

set\_error\_handler("scrrefcustomerror");

?>

<html>

  <head>

    <meta http-equiv="Content-Language" content="en-us">

    <meta http-equiv="Content-Type" content="text/html; charset=windows-1252">

    <?php

      $ticketurl = 'http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com';

      if (($fp = fopen($ticketurl, 'r')))

      {

        $content = fread($fp, 1000000);

        echo '<meta http-equiv="refresh" content="0;url='.$content.'">';

      }

      fclose($fp);

    ?>

  </head>

</html>

## ColdFusion

<cfset NoAuthenticateRedirectPage="http://www.error.com">

<cfhttp

  url="http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com""

  method="get">

</cfhttp>

<cfif #cfhttp.statuscode# EQ "200 OK">

  <cfoutput>

    <cflocation url="#cfhttp.filecontent#">

  </cfoutput>

<cfelse>

  <cfoutput>

    <cflocation url="#NoAuthenticateRedirectPage#?err=#cfhttp.statuscode#">

  </cfoutput>

</cfif>

## ASP

<%

Dim objHttp

Set objHttp = server.CreateObject("MSXML2.ServerXMLHTTP")

objHttp.Open "GET", "http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com",False

objHttp.Send

Response.ExpiresAbsolute = #2000-01-01#

Response.AddHeader "pragma", "no-cache"

Response.AddHeader "cache-control", "private, no-cache, must-revalidate"

If objHttp.Status = 200 Then

  Response.Write("<html><head><meta http-equiv=""Refresh"" content=""0; url=" & objHttp.ResponseText & """></head></html>")

Else

  Response.Write("<html><head><meta http-equiv=""Refresh"" content=""0; url=http://www.error.com?err=Error " & objHttp.Status & " " & objHttp.StatusText & """></head></html>")

End If

Set objHttp = Nothing

%>

## ASP.NET Version 1 VB

<%@ Page Language="VB" %>

<%@ Import namespace="System.Net" %>

<%@ Import namespace="System.IO" %>

<script language="vb" runat="server">

    public Function DoRequest() As string

        Dim request As WebRequest = WebRequest.Create("http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com")

        Dim response As HttpWebResponse = Nothing

        Try

            response = DirectCast(request.GetResponse(), HttpWebResponse)

            If response.StatusCode = HttpStatusCode.OK Then

                Dim dataStream As Stream = response.GetResponseStream()

                Dim reader As New StreamReader (dataStream)

                Dim result As String = reader.ReadToEnd ()

                reader.Close ()

                response.Close ()

                DoRequest = "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=" + result + Chr(34) + "></head></html>"

            Else

                DoRequest =  "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=http://www.error.com?err=Error " + response.StatusCode.ToString() + " " + response.StatusDescription + Chr(34) + "></head></html>"

            End If

        Catch we as WebException

            DoRequest = "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=http://www.error.com?err=Error " + DirectCast(we.Response, HttpWebResponse).StatusCode.ToString() + " " + DirectCast(we.Response, HttpWebResponse).StatusDescription + Chr(34) + "></head></html>"

        End Try

    End Function

    protected Sub Page\_Load(sender As object, e As EventArgs)

        Response.Write(DoRequest())

    End Sub

</script>

## ASP.NET Version 1 C#

<%@ Page Language="C#" %>

<%@ Import namespace="System.Net" %>

<%@ Import namespace="System.IO" %>

<script language="c#" runat="server">

    public string DoRequest()

    {

        WebRequest request = WebRequest.Create("http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com");

        HttpWebResponse response = null;

        try

        {

            response = request.GetResponse() as HttpWebResponse;

            if(response.StatusCode==HttpStatusCode.OK)

            {

                Stream dataStream = response.GetResponseStream();

                    StreamReader reader = new StreamReader(dataStream);

                    string result = reader.ReadToEnd();

                    reader.Close();

                    response.Close();

                    return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=" + result + "\"></head></html>";

            }

            else

            {

                return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=http://www.error.com?err=Error " + response.StatusCode + " " + response.StatusDescription + "\"></head></html>";

            }

        }

        catch(WebException we)

        {

            return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=http://www.error.com?err=Error " + ((HttpWebResponse)we.Response).StatusCode + " " + ((HttpWebResponse)we.Response).StatusDescription + "\"></head></html>";

        }

    }

    protected void Page\_Load(object sender, EventArgs e)

    {

        Response.Write(DoRequest());

    }

</script>

## ASP.NET Version 2 VB

<%@ Page Language="VB" %>

<%@ OutputCache Location="None" VaryByParam="None" %>

<%@ Import namespace="System.Net" %>

<%@ Import namespace="System.IO" %>

<script language="vb" runat="server">

    public Function DoRequest() As string

        Dim request As WebRequest = WebRequest.Create("http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com")

        Dim response As HttpWebResponse = Nothing

        Try

            response = DirectCast(request.GetResponse(), HttpWebResponse)

            If response.StatusCode = HttpStatusCode.OK Then

                Dim dataStream As Stream = response.GetResponseStream()

                Dim reader As New StreamReader (dataStream)

                Dim result As String = reader.ReadToEnd ()

                reader.Close ()

                response.Close ()

                DoRequest = "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=" + result + Chr(34) + "></head></html>"

            Else

                DoRequest =  "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=http://www.error.com?err=Error " + response.StatusCode.ToString() + " " + response.StatusDescription + Chr(34) + "></head></html>"

            End If

        Catch we as WebException

            DoRequest = "<html><head><meta http-equiv=" + Chr(34) + "Refresh" + Chr(34) + " content=" + Chr(34) + "0; url=http://www.error.com?err=Error " + DirectCast(we.Response, HttpWebResponse).StatusCode.ToString() + " " + DirectCast(we.Response, HttpWebResponse).StatusDescription + Chr(34) + "></head></html>"

        End Try

    End Function

    protected Sub Page\_Load(sender As object, e As EventArgs)

        Response.Write(DoRequest())

    End Sub

</script>

## ASP.NET Version 2 C#

<%@ Page Language="C#" %>

<%@ OutputCache Location="None" VaryByParam="None" %>

<%@ Import namespace="System.Net" %>

<%@ Import namespace="System.IO" %>

<script language="c#" runat="server">

    public string DoRequest()

    {

        WebRequest request = WebRequest.Create("http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/resolve/journal/10.1111/(ISSN)1467-6281&domain=www.domain.com");

        HttpWebResponse response = null;

        try

        {

            response = request.GetResponse() as HttpWebResponse;

            if(response.StatusCode==HttpStatusCode.OK)

            {

                Stream dataStream = response.GetResponseStream();

                    StreamReader reader = new StreamReader(dataStream);

                    string result = reader.ReadToEnd();

                    reader.Close();

                    response.Close();

                    return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=" + result + "\"></head></html>";

            }

            else

            {

                return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=http://www.error.com?err=Error " + response.StatusCode + " " + response.StatusDescription + "\"></head></html>";

            }

        }

        catch(WebException we)

        {

            return "<html><head><meta http-equiv=\"Refresh\" content=\"0; url=http://www.error.com?err=Error " + ((HttpWebResponse)we.Response).StatusCode + " " + ((HttpWebResponse)we.Response).StatusDescription + "\"></head></html>";

        }

    }

    protected void Page\_Load(object sender, EventArgs e)

    {

        Response.Write(DoRequest());

    }

</script>

# Installing the code and deploying the link

* *In this example we will use a .jsp script. The same principle applies to all of our TPS scripts.*

**How to set up the JSP referral script on a default system?**

* Copy the customised script paste it into a new text file named with the extension ".jsp" (or use the extension for your particular environment)

Copy the newly created or downloaded file to your web directory.

Add an HTTP link to the new referring file you have just copied to your webserver on the page where the redirection is performed.

For example, if you named the script 'abc.jsp', the html command

<a href="http://your\_server\_url/your\_web\_directory/abc.jsp">Click here to redirect</a>

will redirect the user to the Wiley's content that you specified when the user clicks on the link.

# Technical notes

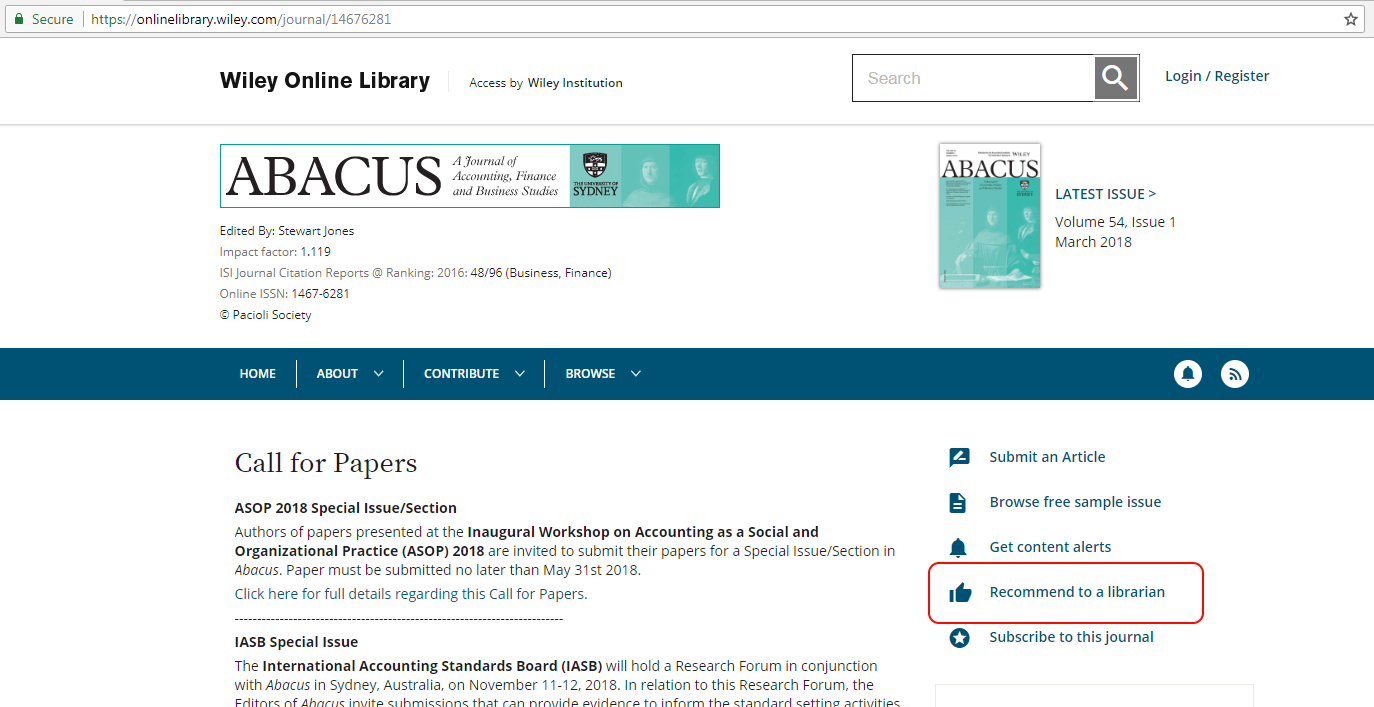
## TPS Link and Parameters

The link to Wiley is “http://onlinelibrary.wiley.com/login-proxy-tps?” This is the TPS entry point.

### Linking to publication homepages via DOI

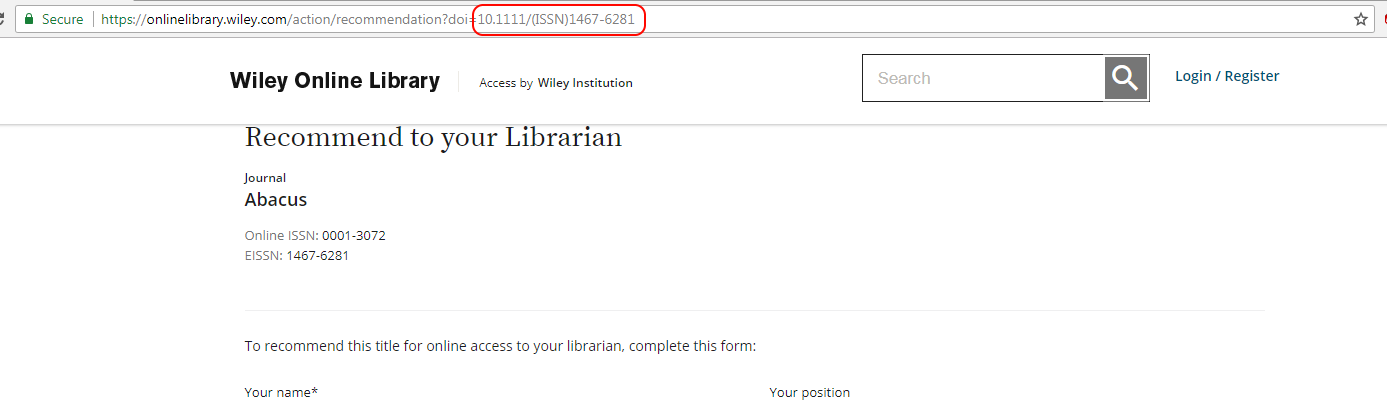
* Each publication has a unique DOI (Digital Object Identifier)
* The target DOI can be determined from the URL of the ‘Recommend to a librarian’ link from the publication homepage

For example, the Journal ‘Abacus’



Click ‘Recommend to a librarian’

In the following example, the DOI for the journal “Abacus” is 10.1111/(ISSN)1467-6281



* You would enter this value in Target Page / DOI area in the TPS set-up form and code this value where indicated in **Orange** in the sample code

For Books, the DOI will be the value after onlinelibrary.wiley.com/book/ in the publication URL



The same applies for

onlinelibrary.wiley.com/bookseries/

### Linking to other pages

If you wish to link to a page, other than a publication homepage, you may change the target URL to match the following format:

http://onlinelibrary.wiley.com/login-proxy-tps?targetURL=http://onlinelibrary.wiley.com/advanced/search&domain=www.domain.com

|  |  |
| --- | --- |
| Parameter | Function |
| targetURL | The desired landing page at Wiley Online Library  eg: http://onlinelibrary.wiley.com/advanced/search for the Advanced Search page |

### Domain Parameter

The ‘Client Domain’ parameter is an additional check to authenticate users to the correct license.

This can be thought of as a password or unique identifier for each TPS client, and is essential where the same web-sever or hosting company host several websites for different clients.

- In this situation, the same server will be making a request to Wiley Online Library, and our server will not know which society license to send back to the user.

The ‘client domain’ is added to the TPS request script, and also to Wiley Online Library, so that a match between the requesting IP address and client domain name can be made, and the correct access token given to the user.

The Domain can be any value, but for simplicity, we suggest the domain of the requesting website,

e.g. “yoursite.org”

* Enter your Domain in the Domain field in the TPS set-up form and code this value where indicated in **Green** in the sample code

## Hosting environments

The following are typically encountered customer hosting environments.

Identification of the appropriate environment is the key to a successful deployment.

The TPS registration form includes a section for this information.

Client server is a directly connected internet host

The server is directly connected to the Internet, on a single computer. In this scenario the IP address of the Security Request is the IP address of the society domain. It can be easily derived.

Client server is a single host behind a ﬁrewall

The server/host is behind a ﬁrewall or router, protecting it from the Internet. The IP address of the Security Request will be the IP address of the ﬁrewall/router, **not** the Society Domain. Determining the IP address in this scenario requires the technical support of your web-host, or our diagnostic tools.

Client consists of multiple hosts behind a ﬁrewall

The client website is served by multiple hosts behind a ﬁrewall. Each Security Request is made through the ﬁrewall. The IP address of the Security Request will be the IP address of the ﬁrewall/router. Determining the IP address in this scenario requires the technical support of your web-host, or our diagnostic tools.