

Juniper Networks Secure Access

**Content Intermediation Engine
Best Practices**

Release 6.5

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net

Part Number: 65A080309

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986–1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by The Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, The Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

Juniper Networks, the Juniper Networks logo, NetScreen, NetScreen Technologies, the NetScreen logo, NetScreen-Global Pro, ScreenOS, and GigaScreen are registered trademarks of Juniper Networks, Inc. in the United States and other countries.

The following are trademarks of Juniper Networks, Inc.: ERX, E-series, ESP, Instant Virtual Extranet, Internet Processor, J2300, J4300, J6300, J-Protect, J-series, J-Web, JUNOS, JUNOScope, JUNOScript, JUNOSe, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, M-series, MMD, NetScreen-5GT, NetScreen-5XP, NetScreen-5XT, NetScreen-25, NetScreen-50, NetScreen-204, NetScreen-208, NetScreen-500, NetScreen-5200, NetScreen-5400, NetScreen-IDP 10, NetScreen-IDP 100, NetScreen-IDP 500, NetScreen-Remote Security Client, NetScreen-Remote VPN Client, NetScreen-SA 1000 Series, NetScreen-SA 3000 Series, NetScreen-SA 5000 Series, NetScreen-SA Central Manager, NetScreen Secure Access, NetScreen-SM 3000, NetScreen-Security Manager, NMC-RX, SDX, Stateful Signature, T320, T640, T-series, and TX Matrix. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. All specifications are subject to change without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Copyright © 2009, Juniper Networks, Inc.
All rights reserved. Printed in USA.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Year 2000 Notice

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

Software License

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller.

3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- Customer shall use the Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller, unless the applicable Juniper documentation expressly permits installation on non-Juniper equipment.
- Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees.
- Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Software on non-Juniper equipment where the Juniper documentation does not expressly permit installation on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; or (k) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Interface Information. To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. Third Party Software. Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentés confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Creating CIE-Compatible Web Applications

One of the core technologies that the Instant Virtual Network (IVE) offers is the Content Intermediation Engine (CIE), a highly advanced parser and rewriter. The CIE retrieves Web-based content from internal Web servers and changes URL references and Java socket calls so that all network references point to the IVE.

For instance, when an authenticated user clicks a link, the request goes to the IVE. The IVE performs *intermediation* by parsing the incoming link to determine the internal destination server and then forwarding the request to that internal server on behalf of the end-user. In other words, the IVE acts as the internal server to the end-user and acts as an end-user to the internal server. This intermediation process provides protection and clear separation between end-users and internal resources.

In order to successfully intermediate Web applications, the CIE must successfully locate all links within a page and rewrite them accurately. This document provides guidelines to Web application developers and user interface designers for creating Web applications that the CIE can successfully intermediate. The document provides general recommendations, lists the content-types that the IVE supports, the level of support that the IVE provides for each of the content types, and the language constructs to avoid.



NOTE: The Content Intermediation Engine does not intermediate all types of links. For instance, it does not intermediate `ftp`, `rtsp`, `mms`, and `mailto` links.

This document contains the following sections:

- “Content Types Supported Through the CIE” on page 2
- “Content Types Supported Through Pass Through Proxy” on page 16
- “Determining When to Use the CIE vs. Pass Through Proxy” on page 16

Content Types Supported Through the CIE

The Content Intermediation Engine fully supports Web applications written in standard HTML, JavaScript, VBScript, and Java. There are a few corner cases, however, in which these content types are sensitive to intermediation and parsing (as outlined in the sections that follow). If this document does not contain information about a content type, the Content Intermediation Engine does not officially support it, but the content type may still work through the IVE.

This section includes support information and restrictions related to the following content types:

- “HTML” on page 2
- “JavaScript” on page 5
- “Framed Toolbar Support” on page 9
- “CSS” on page 10
- “Java” on page 11
- “VBScript” on page 12
- “ActiveX” on page 13
- “Flash” on page 13
- “XML” on page 14
- “PDF” on page 15
- “Streaming Media and Video Content” on page 16

HTML

The Content Intermediation Engine fully supports native HTML 4.0. When creating HTML content, however, please adhere to the guidelines in the following sections.

Use Well-Formed HTML

We recommend that you run your HTML through an HTML syntax checker to ensure that the HTML is well-formed. This process eliminates the possibility of poorly formed HTML with missing information such as end tags and right brackets. Although the Content Intermediation Engine is powerful enough to successfully intermediate invalid HTML, it is safer to write valid and well-structured HTML.

Use Standard HTML

We recommend that you use standard HTML in your Web pages. For example, use the standard format:

```
<A href="www.servername.com:portNo"> Click Here </A>
```

instead of the more rare format:

```
<A href="www.servername.com" port="portNo"> Click Here </A>
```

Specify the Correct Content Type

The Content-Type header in your Web page should match the actual content of the document. For example, do not send a content type of `text/html` if the content is XML.

Construct URLs Using RFC Standards

Follow the URL specification available at <http://www.faqs.org/rfcs/rfc1738.html> when constructing URLs in HTML pages. Avoid using HTML escape codes in the URLs. Use forward slashes (`/`) in URLs instead of backward slashes (`\`).

Use a Supported HTTP header

The Content Intermediation Engine supports HTTP/1.1 and 1.0 when communicating to the browser, but only supports HTTP/1.0 when communicating to the back-end server. Make sure that all HTTP headers adhere to the HTTP specification for the version that you are employing.

The Content Intermediation Engine does not pass all headers from the internal Web servers to the browser. Avoid using custom headers because the Content Intermediation Engine may not pass those headers back to the browsers. Instead use the standard headers defined by HTTP.

Set Character Encoding Through META Tags

Specify the character set in the META tag to avoid problems relating to character encoding. For example, to set the character encoding of a document to EUC-JP, include the following META declaration in the document:

```
<META http-equiv="Content-Type" content="text/html; charset=EUC-JP">
```

Avoid Browser-Specific Code

Avoid writing HTML that is browser-specific. If most commercially-available browsers support a construct, the IVE probably supports it too. For example, the IVE supports the following code snippet that uses layers:

```
<style type="text/css">
<!--
#Layer1 {left: 55px; top: 120px;}
#Layer2 {left: 300px; top: 120px;}
-->
</style>
</head>
<body>
```

```
<div id="Layer1"><a href="http://www.google.com">Google</a> </div>
<div id="Layer2"><a href="http://www.yahoo.com">Yahoo</a> </div>
</body>
```

Microsoft Word and Microsoft Power Point can generate Internet Explorer-specific HTML for embedded drawings and figures. The applications embed these tags within comments so that non-Internet Explorer browsers cannot render the tags. The Content Intermediation Engine might not rewrite these conditional comments appropriately.

Do Not Use Multiple BASE Tags

You should only place **BASE** tags in the **HEAD** element of your HTML pages—The IVE ignores any **BASE** tags that appear inside of the **BODY** tag of a document. This standard is included in the specifications for HTML 3.2 and later and is enforced by Internet Explorer 7.0 and later. Additionally, you should only include one **BASE** tag per document, as required by the HTML 3.2 standard and later.

If you have a page that contains multiple **BASE** tags or **BASE** tags outside of the **HEAD** element, you may encounter broken image links or anchors that do not navigate to the proper locations.

Do Not Embed an “<a href” String Within an “<a href” Tag

Do not embed the `<a href` string within an `<a href` tag. For example, the following HTML code does not work:

```
document.write("<a href='javascript:top.foo('\<a href=alink>label link</a>\'')");
```

Instead, use variables as shown in this example:

```
document.write("<script> var str='\<a href=alink>label link</a>\'";</scr"+"ipt>
<a href='\&quot;javascript:top.foo(str)\&quot;'>");
```

Miscellaneous

In addition to the issues outlined in the previous sections, also keep the following guidelines in mind when creating HTML content:

- Avoid complex nesting and escaping of quotes.
- In HTML tags, do not use null `src` attributes.
- Avoid using in-line server-side script tags, typically marked by `<% ... %>`. The server usually processes these tags before they reach the client. Occasionally, when a server does not process the server-side tags, however, the scripts remain on the client page (which can cause problems).
- When writing `<OBJECT>` and `<APPLET>` tags, make sure `codebase` and `cabbage` are present.
- For best performance, we recommend that you limit the amount of text between angle brackets `< >` to less than 10,000 characters. For example:

```
tagName name="To improve performance, break up a very large string
here."></tagname>.
```

- For performance reasons, we recommend that you write pages that contain no more than 4 frames. Exceeding 4 frames can adversely impact Web rewriting performance.

JavaScript

The Content Intermediation Engine handles complex uses of JavaScript, including menu animation, field validation, pop-up windows, frame manipulation, and calendar functions. In addition, the Content Intermediation Engine also supports standard and advanced JavaScript functions such as `setTimeout`, `setInterval`, and `insertAdjacentHTML`. When creating JavaScript content, however, please adhere to the guidelines in the following sections.



NOTE: JavaScript 1.0 and later and DOM APIs that are level 1 and later are supported. Microsoft specific extensions of DOM and DHTML for IE 6 and later are also supported.

Use Straightforward JavaScript

Even though the Content Intermediation Engine is sophisticated enough to handle complex constructs in JavaScript, it may have trouble processing code whose purpose is obscured by multiple levels of indirection. We recommend that you write your code in a straightforward fashion in order to enable the Content Intermediation Engine to capture all the URL references.

Usage of `document.write`

The Content Intermediation Engine supports the use of `document.write`. We recommend the following guidelines when using `document.write`:

- Do not use base href's in `document.write`.
- Avoid writing nested script tags in `document.write`. If you must write nested script tags in a `document.write`, break the string “`<script ...`” into “`<scr`” + “`ipt....`”
- Tags created partially in static HTML and partially in JavaScript are not supported. For example, the following code snippet is not supported. The “`textarea`” tag is written dynamically using the JavaScript `document.write` function while the rest is written using static HTML

```
<script>
  document.write("<Textarea> Tag contents");
</script>
</Textarea>
```

Avoid Complicated Constructs in the eval() Function

The server cannot intermedate JavaScript code that dynamically generates and executes on the browser such as the `eval()` function. Instead, the IVE inserts a client-side JavaScript parser into the rewritten page in order to parse and rewrite the dynamically generated code. However, the client-side parser is not as sophisticated as the server-side intermediation engine. As a result, the IVE sometimes accurately rewrites code inside a `<script>` tag but might not handle the same strings when you pass them through an `eval()` function. Therefore, complicated constructs within an `eval()` function may not work as you expect. For example, `window.open()` within the `eval()` function works, but accessing the Document Object Model (DOM) in an `eval()` function might not work.

Do Not Use Common JavaScript Functions on the Left Hand Side of an Assignment Statement

If the javascript code contains an assignment statement where a function call is in the left hand side of an assignment statement then it is not supported through the CIE engine. For example, `foo.setAttribute("bar") = "false"` will not work through the CIE engine.

Do Not Assume Element Numbers or Positions in a DOM

The Content Intermediation Engine supports pages that use the Document Object Model (DOM). When traversing the DOM, however, do not assume the number of elements or the position of the elements. Instead, use criteria such as the ID field of the element to access specific DOM elements (since the IVE may add content to the intermediated page, thereby invalidating the original number of DOM elements). Web pages that assume the number of elements or position of an element may trample upon or use content added by the IVE.

For example, if you include five elements on a page, the IVE may add a sixth element to the DOM. When the Web application then attempts to access and display the last element of the page, it displays the element inserted by the IVE, which was not the desired intent.

This guideline is especially relevant when using the IVE toolbar.

Use One Scripting Language Per Page

Use only one scripting language in one page—do not mix JavaScript and VBScript in the same page. If possible, use JavaScript instead of VBScript since VBScript has no published standard that we can recommend at this time.

In relation to scripting languages, keep in mind that using an empty type attribute in a script tag does not work. For example:

```
<script language="javaScript" type="">
    Some JavaScript Code
</script>
```

Limit usage of the with command

Limit the use of the `with` command. Excessive usage could lead to incorrectly rewritten pages. For example, instead of using:

```
with (doc){
```

```
        location=http://...;
    }
```

use:

```
doc.location=http://...
```

Even though the Content Intermediation Engine supports the `with` statement, we recommend that you avoid such statements and use simpler constructs. The IVE may not properly rewrite more complicated statements such as nested `with` statements since it is difficult to distinguish local variable references from property references on an object.

For example:

```
foo = 1;
```

is a local variable but:

```
with (obj) {
    foo = 1;
}
```

In this example, it is difficult to determine if `foo` is a local variable or a property of `obj`. The IVE uses heuristics to trap the common combinations of objects and properties but this practice obviously does not translate to a general solution. For that reason, we recommend that you avoid the use of `with`.

IFRAME Objects Must Contain an src Attribute

IFRAME objects must contain an `src` attribute to avoid the secure/non-secure warning. For example, the rendering of the following IFRAME results in a secure/non-secure warning.

```
var ifrm = document.createElement("IFRAME");
ifrm.id = foo;
ifrm.height = 100;
ifrm.width = 100;
document.body.insertAdjacentElement("bar", ifrm);
```

Use frames.length Instead of frames[0]

When checking for the existence of frames in a document that may not contain any frames, use `frames.length` instead of `frames[0]`.

Setting a Cookie and Accessing the Cookie Through JavaScript

A cookie is not available through JavaScript unless the HTML body exists in the response to the page where the cookie was set. That is, if you are setting a cookie in an HTML response and want that cookie to be available in JavaScript, the response body must contain some HTML content. For example, the following web page will not work:

1. Set a cookie, `myURL`, on a 302 response.
2. The 302 response does not contain any HTML but contains JavaScript.

3. In the `onunload` function in the JavaScript, access the `myURL` cookie.
4. The cookie is not accessible.

Understand the Number of Cookies You Can Set

Most browsers have an upper bound on the number of cookies that you can set on the client-side through the use of `document.cookie`. You cannot use the maximum number of cookies allowed by the browser, however, since the IVE sets cookies as well.

In most deployments, the IVE manages configuration information by setting up to four cookies. (Depending on the options chosen by the IVE administrator, this number might be smaller.) Therefore, your Web application can set the maximum number of cookies allowed by the browser minus four. Deployments that use the eTrust SiteMinder server, however, must set less cookies, since the IVE sends cookies to the Web browser to enable single sign-on between SiteMinder and the IVE.

Use Ascii Characters

To render pages through the CIE engine correctly, avoid non-ascii characters such as ``` and `ñ` in JavaScript.

Selective Rewriting Resource Policy for a POST URL

If the ACTION URL for a FORM POST is being generated on the client-side in JavaScript, a selective rewriting resource policy for the ACTION URL may not work.

To work around this issue:

1. Change the web application so that the ACTION URL is in static HTML. For example,


```
<FORM method=POST ACTION=http://www.post_server.com>
```
2. Change the POST to a GET.

Comments in Assignment Statements

Comments inserted in the middle of a right hand side assignment statement in javascript are not supported. For example, the following statement is not supported through the CIE engine.

```
foo = foo.replace(/class=*/, '').
    //replace(/<p [^>]*>/, '<p>').
    replace(/ style="\\"/, '');
```

Mixing JavaScript and Static Content

Pages where the OBJECT tag or the APPLET tag is partly written in JavaScript and partly as static content do not function well within the engine. To ensure correct functionality write the complete tag through static text or through JavaScript.

Miscellaneous

In addition to the issues outlined in the previous sections, also keep the following guidelines in mind when creating JavaScript content:

- Avoid using variables that indirectly assign URL references to native JavaScript objects using the array format rather than the regular dot format. For example:

```
document["location"] = "http://www.yahoo.com";
```

and

```
var d = document;
var l = "location";
d[l] = "http://www.yahoo.com";
```

Instead, use:

```
document.location = "http://www.yahoo.com";
```

- Do not use HTML and JavaScript reserved words and built-in functions as object names, function names or variable names in your code. For example, do not define and use variables such as **top**, **location**, **pathname**, and **domain**.
- The IVE occasionally generates its own JavaScript functions that start with the string **Dana**. To avoid conflicts with IVE JavaScript functions, avoid using **DanaXXX** as function and variable names.
- Avoid embedding JavaScript in the **src** attributes of tags. For example:

```
<frame name="f1" src="JavaScript:func();">
```

- The IVE does not support the use of port in **window.location**. For example, the IVE does not support the following JavaScript code:

```
window.location.port = portNo
```

Framed Toolbar Support

The IVE supports two kinds of browsing toolbars, the framed toolbar and the floating toolbar. The framed toolbar displays pertinent information in a frame in the IVE end-user console, whereas the floating toolbar floats over the left or right side of the user's browser (possibly obscuring Web content). These toolbars include links to the IVE end user home page, a configurable home page, and the end user help system. Additionally, end-users can use the toolbars to sign out of their IVE sessions, add bookmarks, and see session expiration information.

If you choose to use the framed toolbar, you must keep certain guidelines in mind when creating your Web application to ensure that the application does not “break” out of the IVE frame. The following usage in your Web application could cause the framed toolbar to disappear and display the floating toolbar instead:

- **Pop-ups**—If your Web application opens up a popup through a **window.open** call, then the pop-up will not contain a frame. The parent window will continue to display the frame.

- **The top variable**—We recommend that you do not use the **top** variable when working with a frame set because after the IVE intermediates the page, **top** might reference a different frame than you intend. This change might make the framed toolbar disappear or could cause your intermediated application to work erratically or incorrectly.

The following example includes a frame set definition that correctly names its frames. The example also shows an example of using **target** to properly reference a named frame.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
  "http://www.w3.org/TR/html4/frameset.dtd">
<HTML>
<HEAD>
<TITLE>A frameset document</TITLE>
</HEAD>
<FRAMESET rows="50%,50%">
  <FRAME name="fixed" src="init_fixed.html">
  <FRAME name="dynamic" src="init_dynamic.html">
</FRAMESET>
</HTML>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">
<HTML>
<HEAD>
<TITLE>A document with BASE with a specific target</TITLE>
<BASE href="http://www.mycom.com/Slides" target="dynamic">
</HEAD>
<BODY>
...beginning of the document...
</BODY>
</HTML>
```

- **The parent variable**—You can use the **parent** variable from within a frame set (see exception that follows), but do not use the **parent** variable if your Web page does not include a frame set. Also, do not use the **parent** variable from a JavaScript function within your topmost frame set. If you do, the application does not behave as you intend. Instead, When the IVE intermediates the page, the variable references the IVE frame set instead of your intended document.

CSS

The Content Intermediation Engine supports cascading style sheets. When using cascading style sheets, make sure to set their content types to **text/css**. If you set an incorrect content type, errors could occur through the Content Intermediation Engine. Note that the IVE does not support JavaScript in cascading style sheets.

Java

Java class files contain compiled Java byte-code which the Java Virtual Machine interprets and executes. When the IVE encounters this byte-code, it rewrites the compiled Java without decompiling it. The IVE's new byte-code redirects all HTTP(s) and socket based network communication to an intermediate proxy server via secure HTTPS tunneling. This approach provides a secure and portable proxy mechanism for Web-based client/server applications that utilize client Java applets. The Java rewriting technology is available on the Sun JVM (version 1.4.1 +) and MS JVM platforms.



NOTE: The process of rewriting Java code may affect performance. To improve the performance of Java applications, we recommend using the **Enable Java instrumentation caching** option in the **Maintenance > System > Options** page of the IVE Web console. For more information, see the *Secure Access Administration Guide*.

Supported Java Classes and Methods

The IVE supports most network related classes and methods through the Java rewriting engine. In general, as long as the Java applet uses TCP and the network traffic is initiated from the client, the IVE supports the applet. The following table lists Java classes and corresponding methods that are supported through the Content Intermediation Engine.

Table 1: Supported Java Classes and Methods

Supported Java class	Corresponding methods
java.applet.Applet	All methods
java.applet.AppletContext	showDocument
javax.swing.JApplet	All methods
java.net.Socket	All methods
java.net.URL	getHost, getPort, getFile, getProtocol, openStream, openConnection, toString
java.net.HttpURLConnection	setRequestProperty
java.net.URLConnection	setRequestProperty
java.net.InetAddress	All methods
java.lang.reflect.Method	Invoke
java.lang.Class	getResource
java.lang.ClassLoader	getResource, getResourceAsStream
netscape.javascript.JSObject	eval, call, removeMember, setSlot, setMember
msxml3.IXMLHttpRequest	Open
javax.net.ssl.SSLSocketFactory	createSocket
javax.swing.JEditorPane	setPage
com.ms.lang.RegKey	getStringValue, getIntValue, getBinaryValue
java.util.ResourceBundle	getBundle

Unsupported Java Functionality

Listed below are Java features that are not supported through the Content Intermediation Engine.

- The IVE may not support class files written in a proprietary format. To prevent Java intermediation problems with the IVE, ensure that all network-related classes conform to the Sun Java specification. If the class files do not contain standard byte code then the IVE cannot intermediate the content.
- The IVE does not support Java applets that include a **checksum** validation verifying that the applet is unaltered. (The IVE cannot support this type of validation since it alters the applet's byte code during intermediation.) Instead, you should use the standard code-signing procedures to secure the applet. See “Code Signing Certificates” on page 12 for more details on how the end-user can be assured that the applet is safe.
- The IVE does not support Java applets connections that initiate from the server. If the applet contains server-initiated connections through the use of the **ServerSocket** class, then the applet does not work through the IVE.
- The IVE does not support Java applets that make UDP connections.
- The IVE does not support Java applets that use Java Remote Method Invocation (RMI) Technology.
- The IVE does not support Java applets that use the Java Web Start architecture (JNLP files).
- The IVE does not support Java applets that are written for Oracle JVM or IBM JVM (For support information about applets written for the Oracle JVM and IBM JVM, see “Content Types Supported Through Pass Through Proxy” on page 16)

Code Signing Certificates

Most commercial Java applets that the IVE intermediates perform privileged tasks. To perform these tasks, the user must accept the certificate that is used to sign the applet. However, since the IVE modifies the byte-code, the original signature is invalid and the IVE must re-sign the applet. The IVE re-signs the applet with a self-signed certificate whose CA is not a trusted root. Due to the use of the self-signed certificate, the browser displays a warning that must be accepted for every launch of the applet. To avoid the frequent security warnings, you need to import a code signing certificate. For instructions, see the *Secure Access Administration Guide*.

VBScript

The Content Intermediation Engine supports VBScript rewriting that complies with the Microsoft VBScript language reference guide. The guide can be found at <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/script56/html/ddfa5183-d458-41bc-a489-070296ced968.asp>. VBScript rewriting fails in the following instances:

- If the VBScript is dynamically generated using **document.write**, **document.writeln**, **eval**, or **execScript** then it fails through the CIE engine. In addition, assignment of VBScript code to **innerHTML** and **outerHTML** is not supported through the CIE engine.

- If the VBScript contains code where a function call that the CIE engine rewrites is in the left hand side of an assignment statement then the code does not work through the CIE engine. For example, `foo.setAttribute("bar") = "false"` does not work through the CIE engine.

Do Not Use Common VBScript Functions on the Left Hand Side of an Assignment Statement

The CIE engine does not support VBScript code where a function call is in the left-hand side of an assignment statement. For example, `foo.setAttribute("bar") = "false"` is not supported. You should rewrite this statement as `foo.setAttribute("bar", "false");`

ActiveX

The Content Intermediation Engine supports Active X programs that do not make network calls (for example, through TCP or HTTP). Active X programs that do make network connections might or might not work through the IVE. Since a standard is not available that states where URLs, port numbers, or hostnames can be defined, the Content Intermediation Engine may not locate these items and modify them.

For instance, an Active X program could choose to define the first parameter as a URL and the second parameter as the username while another Active X program could reverse the order of parameters. The CIE does not have the necessary knowledge to consistently rewrite the connections in all cases due to the lack of standards inherent to ActiveX.

You can, however, create resource policies that specify parameters that you want to rewrite. These policies must specify the exact URLs and hostnames that the Web page passes to the Active X controls. For more information, see the *Secure Access Administration Guide*.

The IVE also supports Active X programs that only contain relative links through the Pass Through Proxy feature. See “Content Types Supported Through Pass Through Proxy” on page 16 for more information.



NOTE: Pages where the OBJECT tag or the APPLETTAG tag is partly written in JavaScript and partly as static content do not function well within the engine. To ensure correct functionality write the complete tag through static text or through JavaScript.

Flash

The Content Intermediation Engine supports Flash versions 5, 6 and 7, including dynamic rewriting of internal Web links during an access request. We support the rewriting of Actionscript in Flash. The calls in Actionscript that are supported are: `load`, `send`, `sendAndLoad`, `loadVariables`, `loadMovie`, `loadVariablesNum`, `loadMovieNum`, `loadClip`, `loadSound`, `apply`, `connect` on classes of `XML`, `Sound`, `MovieClip`, `NetConnection`, and `MovieClipLoader`. The `eval` equivalent of Actionscript is not supported. Therefore we recommend that the above function calls not be embedded in an Actionscript string object. Note, however, that the IVE does not support Flash applications that use the `XMLSocket` object or Flash remoting.

If an assignment statement is used for URLs in objects then it will not work through the rewriter.

For example, the following construct is not supported:

```
xml_connector_obj.URL = "http://...";
xml_connector_obj.trigger();
```

XML

The IVE supports Web applications that use DTDs, XML schemas, and XML islands within an HTML file. When creating XML content, however, please note the following guidelines:

- The IVE does not support referencing style sheets or DTDs on a separate server.
- The IVE does not support using the `document` call.
- The IVE does not support using the CSS extension for the Microsoft alpha image loader when rewriting XSLT style sheets. However, you can use the alpha image loader if you do not invoke XSLT expressions. For example, the CIE does not support the `STYLE` portion of the following code:

```
<!-- This DIV is the target container for the filter. -->
<DIV ID="oDiv" STYLE="position:relative; width:200px; color:gold;
    filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(
        src='/workshop/graphics/earlobe.gif');" >
    The World
</DIV>
```

- The IVE does not support XSL style sheets that use Microsoft WD-XSL and that use XSL expressions to construct hyperlinks. The IVE delivers your page correctly, however, provided all hyperlinks within the page do not use XSL expressions.
- The IVE does not support passing the parameters of ActiveX or Applet objects using XSLT expressions. If you do not use HTML hyperlinks, however, the objects function properly.
- The IVE does not support manipulating DTD, Xlink, XForm, and XInclude using XSLT expressions. If you do not use XSLT expressions when creating these, however, the page functions properly through IVE.
- The IVE does not support rewriting DTDs inside an HTML file.
- The IVE does not support using XSLT expressions to generate HTML hyperlinks inside JavaScript or VBScript statements. As long as XSLT expression is not used to generate a hyper link inside javascript or VBScript, the page functions properly.

PDF

The Content Intermediation Engine supports rewriting PDF files from all Acrobat versions when you enable the **Rewrite links in PDF files** option on the **Users > User Roles > Select Role > Web > Options** page of the IVE Web console. When you select this option, the IVE rewrites absolute URLs (such as <http://www.google.com>) and relative URLs (such as <http://yourcompany.intranet.net/images/./test.gif>). Otherwise, if you do not select this option, the IVE may not properly display PDF files with links.

The IVE supports rewriting normal PDFs and linearized PDFs. A *normal PDF* requires the browser download the entire document before displaying it. A *linearized PDF* enables the browser to download parts of the document separately, thereby allowing the browser to start displaying the document before it is completely downloaded.



NOTE:

- The IVE does not rewrite embedded streams in PDF files.
- The IVE does not modify encrypted or digitally signed PDF files at all.
- Manually edited PDF files that have incorrect byte offsets do not work correctly through the IVE. Even though these files might work through Acrobat 7, they are not supported through the IVE.
- PDF files that contain 2 objects for the same link do not work through the IVE. You can check if the PDF file contains two objects for the same link by doing the following:
 - a. Open the PDF with Acrobat with Notepad or Wordpad and look for the URI for which you would like to determine the object. (Open the PDF file with Notepad or Wordpad instead of the Acrobat Reader.)
 - b. Look for the URI string and find out what is the object number that references this URI. The URI object is in the following format:

```
55 0 obj
<</S URI
...
/URI (http://www.google.com)>>
endobj
```

where, 55 is the object number.

- c. Next check if the file contains another object with the same object number referencing a different URL. If another object with the same number is found then the PDF file cannot be rewritten through the CIE engine.
-

Streaming Media and Video Content

Since streaming media content often contains direct network connections without the use of HTTP, the CIE cannot support it. If you deliver the streaming content through an <OBJECT> tag and one of the attributes of the tag is a URL to which an HTTP connection is made, then the content may work through the IVE.

Content Types Supported Through Pass Through Proxy

Pass Through Proxy is a key component of the Content Intermediation Engine that supports various intermediation-sensitive content types with relative links such as Active X, the IBM JVM, and the Oracle JVM. For information about when to use the Pass Through Proxy feature, see Table 2, “Supported Content Types” on page 16.

Determining When to Use the CIE vs. Pass Through Proxy

Our final recommendation is to test the Web-based content through the IVE Content Intermediation Engine. If a page does not display accurately then this document can provide suggestions on how you can alter your code to ensure compatibility with the IVE.

To summarize, Web applications written in HTML, JavaScript, VBScript, Java, or XML that use the guidelines listed in this document should work seamlessly with the IVE Content Intermediation Engine (CIE). The IVE supports other content types such as Active X through the Pass Through Proxy feature as long as the content only contains relative links.

Table 2: Supported Content Types

Content type	Support level
HTML	Fully supported. (Refer to this document for details.)
JavaScript	Fully supported. (Refer to this document for details.)
VBScript	Fully supported. (Refer to this document for details.)
Java	Fully supported. (Refer to this document for details.)
Flash	Fully supported. (Refer to this document for details.)
ActiveX	Partially supported.*
PDF	Partially supported. (Refer to this document for details.)
XML	Fully supported. (Refer to this document for details.)
Streaming	Very limited support. If your application does not work, please contact your account team about using Network Connect as an alternative.

* To use partially supported formats, you may need to use the Pass Through Proxy option. (Refer to this document for details.) In most cases, the IVE can intermedate the content, and in the few cases where it cannot, you can easily modify the content to a supportable format. If you cannot modify the content, please contact your account team about using the Secure Application Manager as an alternative.