

# Good Mobile Intranet™

---

Wireless Access to Business Applications

## Developer's Guide

Version 1.9.9

## Copyright, trademark and patent information.

©Good Technology, Inc. 2001-2008. All rights reserved. All use subject to license terms posted at [www.good.com/legaldocs](http://www.good.com/legaldocs). Good Technology, Inc. 2001-2006. All rights reserved. Good, Good Technology, the Good logo, Good Mobile Messaging, Good Mobile Intranet, Good Mobile Defense, Good Mobile Application Services, GoodAccess, GoodInfo, GoodLink, and Powered by Good are trademarks of Good Technology, Inc. VeriSign(R) is a registered trademark of VeriSign, Inc. All other trademarks and service marks contained herein are the property of their respective owners. For example, Microsoft, Windows, Windows NT, Exchange and Outlook are trademarks of Microsoft Corporation. RIM, Research in Motion, RIM 950, RIM 957, and BlackBerry are registered trademarks or trademarks of Research in Motion Limited. Mobitex is a trademark of the Swedish Telecommunications Administration that may be registered in some jurisdictions. Datalight is a registered trademark of Datalight, Inc. FlashFX(tm) is a trademark of Datalight, Inc. Cingular, Cingular Wireless, the Cingular Icon, Xpress Mail, and Xpress Mail with GoodLink are trademarks of Cingular Wireless, LLC. All rights reserved.

Some or all of the following notices may apply to portions of the software or documentation provided by Good Technology, Inc.: Outside In®Wireless Export © 2001 Stellent Chicago, Inc. All rights reserved. Copyright 1993-2001 Datalight, Inc., All Rights Reserved. U.S. Patent Office 5,860,082. Code written by John Halleck is used with his permission. This distribution contains executables of the Netscape® Security Service (NSS) and Netscape Portable Runtime (NSPR). You may obtain the source code for these files from [www.mozilla.org](http://www.mozilla.org), which source files are subject to the Mozilla Public License 1.1. Part of the software embedded in this product is eCos - Embedded Configurable Operating System, a trademark of Red Hat. Portions created by Red Hat are Copyright (C) 1998, 1999, 2000 Red Hat, Inc. (<http://www.redhat.com/>). All Rights Reserved. THE SOFTWARE IN THIS PRODUCT WAS IN PART PROVIDED BY RED HAT AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED BY RED HAT. IN NO EVENT SHALL RED HAT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. You may obtain a copy of the source code of the eCos Original Code from <http://www.redhat.com>. You may obtain a copy of source code of Good Technology, Inc.'s Modifications that have been publicly released in Executable form by sending an email to [support@good.com](mailto:support@good.com). The source code of the eCos Original Code and Good Technology, Inc.'s Modifications are subject to the Red Hat eCos Public License Version 1.1 (copy available at <http://www.redhat.com/>).

Some or all of the following notices may also apply to portions of the software or documentation provided by Good Technology, Inc.: ScriptEase(tm) Javascript/ECMAScript interpreter developed by Nombas, Inc. All Rights Reserved. This product includes software developed by the Apache Software Foundation (<http://www.apache.org>). Copyright (c) 2000-2003, The Apache Software Foundation and/or Yves Pigué. All rights reserved. Neither the name of Yves Pigué nor the names of its contributors may be used to endorse or promote products derived from this software without specific

prior written permission. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License. Copyright (c)1999-2001 Dan Adler, 315 E72 St. NY, NY, 10021 USA. mailto: [danadler@rcn.com](mailto:danadler@rcn.com) All rights reserved. The Jetty Package is Copyright Mort Bay Consulting Pty. Ltd. (Australia) and others. Individual files in this package may contain additional copyright notices. The javax.servlet packages are copyright Sun Microsystems Inc. Copyright (c) 1990-2003 Sleepycat Software. All rights reserved. You may obtain a copy of the source code for the DB software from <http://www.sleepycat.com>. You may obtain a copy of source code of Good Technology, Inc.'s Modifications that have been publicly released in Executable form by sending an email to [support@good.com](mailto:support@good.com). Copyright ©1996-1999 Corporation for National Research Initiatives; All Rights Reserved. Copyright (c) 1995-2000 by the Hypersonic SQL Group. All rights reserved. Copyright (c) 2001-2002, The HSQJ Development Group. All rights reserved. Copyright 2002 (C) Nathaniel G. Auvil. All Rights Reserved. Copyright (c) 1998-2000 World Wide Web Consortium (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. Copyright (c) 2001 MX4j. All rights reserved. Copyright 1994-2004 Sun Microsystems, Inc. All Rights Reserved. Copyright 1999,2000 Boris Fomitchev Copyright 1994 Hewlett-Packard Company Copyright 1996, 97 Silicon Graphics Computer Systems, Inc. Copyright 1997 Moscow Center for SPARC Technology.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL ANY COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Good Technology, Inc. may have patents or pending patent applications, trademarks, copyrights or other intellectual property rights covering this subject matter. The software and documentation do not give you any license to these patents, trademarks, copyrights, or other intellectual property rights except as expressly provided in any written license agreement from Good Technology, Inc. The software and documentation may be covered by one or more patents as set forth at <http://www.rim.net/patents> which have been licensed by Research in Motion, Ltd. ("RIM") to Good. RIM is not affiliated with, nor does RIM endorse the operability of, the products or services described herein. Such patent license should not be construed as exhausting RIM's rights to royalties or damages or other compensation or relief or the grant of any express or implied license: (a) in relation to customer's use of third party products (except to the extent that use of third party email applications arises as a direct result of the customer using Good's products or services or the customer uses a third party wireless personal digital assistant or network carrier services in conjunction with Good's products or services); or (b) where customer or the supplier of the wireless personal digital assistant or wireless network services asserts any intellectual property rights against RIM notwithstanding the terms of clause (a) above, and RIM has exercised its right to suspend all or a portion of the licenses granted to Good.

## **Disclaimer**

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Good Technology, Inc. Information in this document is subject to change without notice. This publication could include technical inaccuracies or typographical errors. Good Technology may make improvements or changes in the products or the programs described in this publication at any time.

Good Technology, Inc.  
4250 Burton Drive  
Santa Clara, CA, 95054  
Tel. (408) 327-6000 Fax (408) 327-6001  
Web site: [www.good.com](http://www.good.com)

**Be Good. Be Safe.**  
**Please do not use while driving or engaged in any other activity that requires your full attention.**

# Contents

---

## 1 Overview 9

- Using Transformations 9
- Creating Custom Applications 11
- Understanding the Development Process 11
- Picking Compatible Applications 12
- Setting Up the Good Intranet Properties File 12
- Setting Up the Application Properties File 15

## 2 Using Transformations 17

- Built-In Transformations 17
- Using Debugging Transformations 20
  - Unpooled Transformations 20
- Using Custom Transformations 21
- Testing and Debugging Applications 22
  - Log Support 22
  - Debugging Tips 23
- Using Transformation Debugging Tools 24
- Using the HttpTransformTool 24
- Deploying Good Intranet Applications 26
- Examples for Creating Custom Transformations 26
  - Flight Status Example 26
  - Analyzing the Site 27

Updating the Configuration Properties File	29
Updating the Applications Properties File	29
Creating a Custom Style Sheet	30
Processing Result	32
Customizing Tips	32
Example - Sample Intranet	34
Using Transformations to Control Site Access	38

### **3 Developing Custom Applications 41**

Understanding How Applications Work With Good Intranet	41
Understanding the Benefits of This Approach	42
Identifying the Application Requested	43
Selecting Pages and Providing Transformations	47
Selector Component Syntax	50
Selector Matching and Ranking	51
Using Xpath to Refine a Selection	52
Ignoring Case	54
Setting Application Properties	55
Creating an HTML Form for Uploading Files	55
Restricting File Size	56
Supported HTML Elements	57
Additional Restrictions, Limitations and Notes	66
HTML Color Support	66
Unsupported HTML	69
Image support	69
Audio and Video Support	70
MIME Type Conversion	70
Character Encoding Standards	73
Cache Control	73
Unsupported WML Features	73
Support for JavaScript	76

Launching Good Intranet from a Third-Party  
Application 85

Migrating GoodInfo Files to Good Intranet 87

Using GoodInfo 1.0 System Variables 89

**Index 91**



# 1 Overview

---

Corporate developers can use Good Intranet to adapt and extend applications for use with Good Intranet, since Good Intranet is based on standard web technologies, including XSLT, XML, CGI, and HTTP(S). Custom applications can be accessed through a number of techniques such as Visual Studio .NET programming and .ASP page creation.

The server includes an intelligent XSLT transformation engine for conversion of web page content. You can develop your own applications to take advantage of these built-in transformations or create transformations of your own.

This chapter describes the Good Intranet application development process and integration methods available. It also provides an overview of the application development process.

## Using Transformations

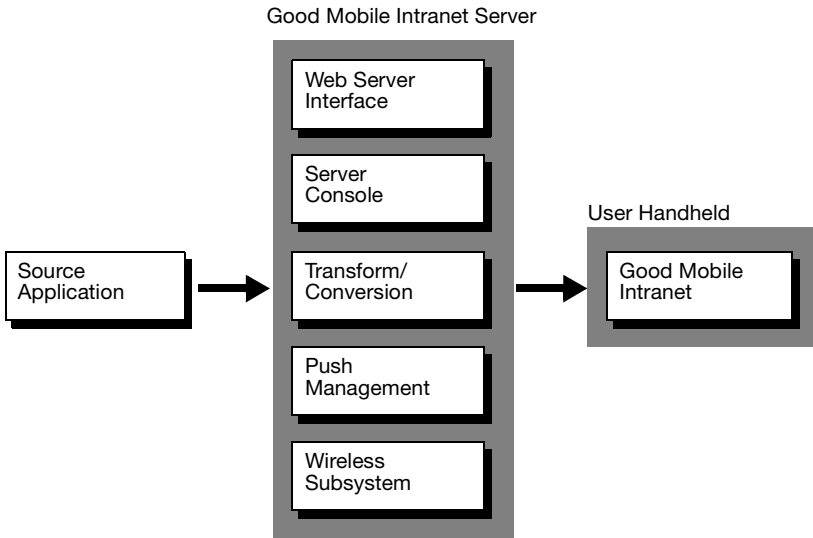
Good Intranet Server includes a set of transformations you can use to integrate applications with Good Intranet. These transformations enable you to take new or existing corporate applications and optimize them for users' handhelds. You can leverage your existing application development investment. You don't have to rewrite applications to work with Good Intranet.

## Overview

Transformations are useful because many web sites do not display well on a handheld's small screen. Transformations enable you to filter information before it reaches the handheld so users get the information they need in an attractive display. Transformations can be used on a specific page, a domain, or all pages in a site.

During transformation, Good Intranet parses through HTML to find specific items, remove items, or rewrite them. For example, Good Intranet can process a web application to remove graphics or simplify table layouts. The transformations defined for Good Intranet control the layout of the output documents and where to access data within the input document.

Transformations reside on the server, so all the heavy-duty processing is done by the server instead of the handheld. This enable more-efficient downloads and faster application processing on the handheld.



For more information about transformations, see "Using Transformations" on page 17.

## Creating Custom Applications

You can build your own custom applications for use with Good Intranet. Application developers can identify custom applications used by Good Intranet handhelds and creating separate configuration files for each custom application. You use the Good Intranet properties file (*config.props*) and application-specific properties files for the other items.

For more information about creating custom applications, see “Developing Custom Applications” on page 41.

## Understanding the Development Process

This sections describes the overall development process for building and debugging application transformations. Good Intranet Server provides a set of default transformations you can use for your application or your can create your own custom transformations using XSLT.

- Identify a key application or web site that mobile users need to access when away from the office. Make a list of key information or program features users need.
- Analyze the site you want to transform. Check the site to determine which pages should be transformed and which tools are best suited for the transformation. See “Picking Compatible Applications” on page 12 for details.
- Set Good Intranet Properties. Properties you set in the *config.props* file determine which URLs are transformed and which application properties files are used. See “Picking Compatible Applications” on page 12 for details.

## Overview

- Create an application properties file. This file contains transformations specific to your application. You can include Good Intranet's built-in transformations or point to a XSLT transformation file. See "Setting Up the Application Properties File" on page 15 for details.
- If necessary, design a custom XSLT transformation for your application. See "Examples for Creating Custom Transformations" on page 26 for details.
- Test and debug the application on a handheld. This chapter includes some tips and techniques for debugging common problems that can occur with transformations. See "Testing and Debugging Applications" on page 22 for details.

## Picking Compatible Applications

Transformations work on public web sites or your own personal corporate application sites. One of the keys to preparing a successful transformation is to pick a compatible site. Sites should include well-formed HTML. Not all public and private web applications meet these requirements. For more information on well-formed HTML, refer to your HTML standards documentation. Good Technology recommends starting simply then building complexity as your experience with transformations grows.

## Setting Up the Good Intranet Properties File

Good Intranet Server contains the following property files:

- *Config.props* is the default property file. It contains the most-common Server properties and defaults. This file also contains some supplemental properties for fine-tuning your server installation. These properties are commented out.

- *Applicationname.props* is the application properties files. These are properties files you create for your applications. Good Intranet includes a generic properties file named *generic.props* and it applies to all applications that do not have an application-specific properties file.
- When you first install Good Intranet Server, the server makes a backup of the server configuration file and saves it as *config.props.bak*. You can use this file to restore the original server defaults if desired.

In addition to the *config.props* file, applications you develop for Good Intranet can have their own properties file to control such things as data transformations and header replacement. Application properties files are located in:

**install\_dir**\app

This directory includes a *generic.props* file that applies to all applications that don't have an application-specific property file. You can use this file as a template for creating custom properties files for your own applications.

For details about the Good Intranet property files, refer to the *Good Intranet Administrator's Guide*.

The server looks for transformation in the *config.props* file. When you are creating or customizing a Good Intranet application, you need to edit the file (located in *install\_dir*\*config.props*) and include the following:

1. Define a selector property for your application.

The server uses a *selector* property to recognize which sites to transform.

Applications can be selected:

- Using regular expressions. For example:

```
app.gtsweb.selectors requestUrl=.*?[gG][tT][sS][wW][eE][bB].*
```

## Overview

The server matches URLs by finding the best match. Be careful when you define a regular expression because the server could match similar expressions incorrectly.

- Using host matching. For example:

```
app.getit.selectors requestUrl=http://w1thinkpad01/.*
```

- Using specific URL matching. For example:

```
app.demo.selectors requestUrl=http://GoodMobileIntranet/doc/addlabor.html
```

### 2. Set the location of the application properties file.

Once a site is selected by the server, the server references a *properties* setting for the site. This setting points to a specific properties file that defines transformations for the site.

The following are some examples of *properties* settings:

```
app.demo.properties app/demo.props  
app.bugzilla.properties app/bugzilla.props
```

The *property* definition for a selector must match the selector exactly. So if you have a *gtsweb* application, you must have both the following properties defined:

```
app.gtsweb.selectors  
app.gtsweb.properties
```

### 3. Choose a reload option.

Optionally, you can have a *reload* property. When this property is set to *true*, you do not have to restart the server after making changes to the application properties file. For example:

```
app.demo.reload true
```

Good Technology recommends keeping this property set to *true*. Otherwise, you may experience unexpected application behavior.

- ### 4. When you've finished editing *config.props*, you should have completed set of property definitions for your application. For example:

```
app.myapp.selectors requestUrl=.*
app.myapp.properties app/myapp.props
app.myapp.reload true
```

For more information on setting properties in *config.props*, see “Server Configuration Parameters” on page 141.

## Setting Up the Application Properties File

After you edit the main Server properties file (*config.props*), you must create a more-specific properties file for your application (for example, *app\myapp.props*). Called an application properties file, this file contains transformation definitions for your application. Definitions can include the server’s built-in transformations and/or pointers to custom XSLTs you have defined to selectively transform pages.

To set up an application properties file:

1. Create a plain text file.
2. Determine which built-in transformations you want to include in the file.
3. If necessary, create custom XSLT transformations and add pointers to these transformations. For more information, see “Examples for Creating Custom Transformations” on page 26.
4. Save the file. Make sure the filename and location matches the value you entered for the *app.name.properties* setting in *config.props*.

For an example of an applications properties file, see the generic application properties file located in the Good Intranet Server installation directory (*install\_dir\app\generic.props*).

## **Overview**

# 2 Using Transformations

---

This chapter describes how to develop and extend applications for use with Good Intranet. It describes how to use and manage transformations and provides some examples of how to use them to optimize your applications.

## Built-In Transformations

Good Intranet Server includes the following built-in transformations you can use in your applications properties file. These transformations are automatically applied to *generic* applications (applications that don't reference any other properties file).

### *Built-In Transformations*

---

<b>Transformation</b>	<b>Description</b>
<i>CleanTables</i>	Removes empty rows, columns, and tables. Basically, removes empty table data cells (defined by <code>&lt;td&gt;</code> tags) from the document.
<i>CompressHtml</i>	Removes HTML content that holds redundant or unnecessary information.

## Using Transformations

### Built-In Transformations

---

Transformation	Description
<i>IndentTables</i> [( <i>style=ol</i> )]	<p>Replaces a table with a bulleted list. For example, a table with this structure:</p> <pre>a0 a1 a2 b0 b1 b2</pre> <p>Is changed to this structure:</p> <pre>a0 * a1 * a2 b0 * b1 * b2</pre> <p>If you specify the optional (<i>style=ol</i>) argument, the result includes a numbered list instead of a bulleted list (an HTML ordered list <code>&lt;ol&gt;</code>, instead of an unordered list <code>&lt;li&gt;</code>).</p>
<i>RemoveImgs</i>	<p>Removes images (<code>&lt;img&gt;</code> tags) that do not have hyperlinks. Also replaces images that are hyperlinked with the value of their <i>alt</i> attribute.</p>
<i>TransformIfLarge</i> ( <i>size=number</i> )	<p>Checks to see if the content is larger than the <b>number</b> of bytes specified. If so, further transformations are done.</p>
<i>TransformNever</i>	<p>No further transformations are done. Because the <i>response.name.transforms</i> property must have at least one transformation specified, you can use <i>TransformNever</i> as a placeholder.</p>

*Built-In Transformations*

Transformation	Description
<i>Truncate</i> ([ <i>size=number</i> ] [ <i>text=&lt;value&gt;</i> ])	<p>Limits the response to no more than <i>number</i> bytes by removing HTML elements. If <i>size=number</i> is specified, a fixed limit of <i>number</i> bytes is used. If <i>size=number</i> is not specified, the limit set within the <i>responsize.&lt;handheld_type&gt;.normal parameter</i> in <i>config.props</i> is used. In other words, the filter truncates the content so it will be delivered to the handheld.</p> <p>If a text <i>value</i> is specified, this value is inserted into the content and appears on the handheld. If a text value is not specified, the default value, <i>Truncated by Good Intranet</i>, is used.</p> <p>Example: <i>Truncate (size=22222 text="&lt;Truncated by Home Office&gt;")</i></p>
<i>UnpooledXslt</i> ( <i>transform="filename"</i> )	<p>Transforms content like the <i>Xslt</i> transformation (described below). This transformation is intended for use during active development of the XSLT. It reads in the XSLT source every time it is used to transform a page.</p>
<i>UnrollTables</i>	<p>Moves the contents of table data cells (defined by <i>&lt;td&gt;</i> tags) outside the table. Inserts <i>&lt;br&gt;</i> tags as appropriate to maintain element spacing. For example, a table with this structure:</p> <pre>a0 a1 a2 b0 b1 b2</pre> <p>Is changed to this:</p> <pre>a0 a1 a2 b0 b1 b2</pre>
<i>UnrollTablesWithTables</i>	<p>For tables within tables, the inner-most table layout is preserved, but all outer tables are unrolled.</p>

## Using Transformations

### *Built-In Transformations*

---

Transformation	Description
<i>AutoScaleImage</i>	Scales images automatically based on the screen size of the handheld. The <code>postData</code> parameter in the selector does not apply here.
<i>Xslt</i> ( <i>transform="filename"</i> )	Enables you to design and reference custom transformations. This command reads the XSLT style sheet specified by <i>filename</i> and transforms the content based on this style sheet. The <i>filename</i> value can be an absolute or relative path. (A relative path is relative to the directory that contains this application properties file.)  Example: <i>Xslt (transform="bugzilla/bzMain.xsl")</i>

## Using Debugging Transformations

In addition to the transformations listed above, Good Intranet includes some debugging transformations you can use to help troubleshoot potential problems. See the `app\generic.props` file for details.

### Unpooled Transformations

Good Intranet Server also supports an *unpooledXSLT* transformation that refreshes automatically when changed.

---

<i>UnpooledXslt</i> ( <i>transform="filename"</i> )	This command reads the XSLT style sheet specified by <i>filename</i> and transforms the content based on this style sheet. This transformation refreshes automatically.  The <i>filename</i> value can be an absolute or relative path. (A relative path is relative to the directory that contains this application properties file.)  Example: <i>UnpooledXslt (transform="myapp/Main.xsl")</i>
--	---

## Using Custom Transformations

If desired, you can create custom transformations and add references to them in your applications properties file. For an example of how to create a custom transformation, see “Examples for Creating Custom Transformations” on page 26.

*Important:* If you create a separate properties files for your application, transformations (and other settings) in the *generic* application properties file no longer apply. If desired, you can copy (and modify) transformation settings from the *generic* application properties file to your application’s properties file.

## Testing and Debugging Applications

Before you deploy your application to users, make sure the HTML syntax of your application is correct. You can check the syntax by opening the application on a handheld then check the application interface and log files to see if errors occur.

This section describes logs, tips, and other tools you can use to help test and debug your applications.

### Log Support

Good Intranet Server includes a set of standard and diagnostic logs you can use to help test your applications:

#### *Types of Log Files*

---

<b>Log Type</b>	<b>Description</b>
<b>Production Logs</b>	
Access	Logs information about HTTP requests and responses.
Management	Records initialization of the server console and errors.
Content	Logs content errors, such as incorrect HTML tags that prevent the browser from displaying a page.
Default	Contains start up, configuration, and shutdown information.
Push	Logs information about the states of push messages and their deliverables.
Stderr	Logs serious error messages that would otherwise be sent to the standard error.
Stdout	Primarily logs serious messages that would otherwise be sent to the standard output.

*Types of Log Files*

---

Log Type	Description
<b>Diagnostic Logs</b>	
Converter	The converter log stores messages related to content conversion.
Diff	Logs caching of new content and sending of differential data.
Redirect	Logs data when server encounters an HTTP redirection response from a web server.
Urlsubst	Logs URL substitutions when the server replaces the path in a URL that follows the destination host name with a different path.

**Debugging Tips**

The following are some tips on debugging transformations:

1. Refer to the stdout log to help debug a problem.

This log contains useful information about how the server is matching your transformations.

- a. Check the Good Intranet Server Console or the *config.props* file to make sure the stdout log is enabled.
  - b. Examine the log entries to see how transformations are being applied. (See example below.)
2. Include a transformation identification stamp in the application file.
  3. In custom XSLT transformations, use lowercase only to match element names and attribute values. For example:

```
<a href="http://www.name.com">Name</a>
```

instead of:

```
<A HREF="http://www.name.com">Name</A>
```

# Using Transformation Debugging Tools

Because of wireless connectivity and provisioning requirements, developing and testing content transformations can be difficult using an actual handheld. Good provides a command-line tool (*HttpTransformTool*) for doing scripted tests to help you test and debug transformations. This tool can be run from a local, networked computer that has access to Good Intranet Server. It shows the following:

- Request made to the origin server
- Response returned from the origin server
- Tidied response (displayed in XHTML)
- Tidied and transformed response (displayed in XHTML)
- How the transformation was selected

To use this tool, you must have the HTTP development port enabled on your Good Intranet server and a web browser (such as Microsoft Internet Explorer, version 1.5) configured for proxying through the tool. Instructions are provided below.

*Important:* Good Technology provides this transformation tool for your convenience only. Further testing and debugging is required for creating transformations that work in a production environment.

## Using the HttpTransformTool

The *HttpTransformTool* is a command line tool you can use for scripted tests.

To use the tool:

1. Open a command window on the local machine.
2. From the local machine, change to the directory you used to install Good Intranet (typically, this is: C:\Program Files\Good Technology\Good Intranet Server):
3. Type the following command:

```
jre\bin\java HttpTransformTool
```

The following text appears. It describes how to use the tool.

*Usage:*

```
HttpTransformTool -url=<url> [options]
```

*Description:*

*Make an HTTP "get" request through the Good Intranet Server, returning information about the request and its transformation*

```
-url=<url>
```

*The web resource to "get". The <url> has the form*

*<http://host/<path>>*

*Options:*

```
-host=<url>
```

*The URL of the development port on the Good Intranet Server.*

*If not given, the value "<http://localhost:8090>" is used.*

4. Follow the on-screen instructions to use the tool.

When you enter the HttpTransformTool command and specify a URL it should connect to the Good Intranet Server and return a multi-part MIME document as a result.

## Deploying Good Intranet Applications

When you have finished test and debugging your Good Intranet application, you're ready to deploy the application to user handhelds.

To deploy a Good Intranet application:

1. Post the application on an HTTP (or HTTPS) site.
2. Email the URL location to mobile users.
3. The mobile user clicks the URL in the email message to access the application.

*Note:* The first time a user accesses the Good Intranet client, the default home page is shown, even if you have changed it. The new page you defined appears thereafter.

## Examples for Creating Custom Transformations

The following examples show how to create a custom transformation for the Flight Status page on *www.yahoo.com* and a sample intranet page. You can use these as models for creating your own custom transformations.

### Flight Status Example

**Important:** Information provided in the chapter is for instructional purposes only. Public web sites are subject to change, and if so, some portions of this example may no longer apply. All Yahoo!® trademarks, logos, service marks, trade dress, slogans, copyrighted designs, or other brand features can be used only as explicitly licensed by Yahoo!.

The main tasks required to create a custom transformation are:

- Analyze the site.
- Update the configuration properties file.
- Update the application properties file to apply a custom XSLT transformation.

*Important:* Do not confuse the application properties file (for example, *install\_dir\app\flightstatus.props*) with the main server properties file (*install\_dir\config.props*).

- Create a custom style sheet for the site.

Each of these tasks are described in detail in the following section.

### Analyzing the Site

The first step in creating a custom transformation is to look at the site and determine the type of transformations you want to perform.

1. Examine the application interface for important information content.

## Using Transformations

You'll notice on the flight status page that most of the relevant information is located on the right, near the center of the page.

Relevant information

Reproduced with permission of Yahoo! Inc. \* 2004 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

## 2. Examine the application source code for key fields and forms.

There are two HTML form definitions that contain the relevant fields in the application.

Corresponds to first form definition

Corresponds to second form definition

Reproduced with permission of Yahoo! Inc. \* 2004 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

- a. This is the form that holds the top part of the application for users who know the airline and flight number for the status check:

```
<form name="status_resv_lhc_us" action="http://
edit.travel.yahoo.com/config/ytravel_checkflight" method=get >
```

- b. This is the form that holds the bottom part of the application for users who need to look up flight information before getting status:

```
<form action="http://edit.travel.yahoo.com/config/
ytravel_checkflight" method="GET" name="RESERVE" >
```

**Tip:** To help search through HTML code for relevant fields and forms, try searching the text for distinguishable keywords such as “form,” “method” or “get.”

## Updating the Configuration Properties File

You need to add the following properties to the *config.props* file (*install\_dir/config.props*). You can use a text editor to edit the file.

1. Open the *config.props* file.
2. Add the following properties:

```
app.flightstatus.selectors requestUrl=http://edit.travel.yahoo.com/*
app.flightstatus.properties app/flightstatus.props
```

3. Save your changes.

## Updating the Applications Properties File

Before you define a custom transformation, make sure the applications properties file (*app/appname.properties*) includes a pointer to the transformation.

1. Open the application properties file. For example:

```
install_dir/app/flightstatus.props
```

2. Add a selector for the page. For example:

## Using Transformations

```
response.yahoolightmain.selectors  
requestUrl=http://edit.travel.yahoo.com/config/ytravel_checkflight
```

### 3. Add a custom transformation:

```
response.yahoolightmain.transforms  
Xslt (transform="flightstatus/custom.xsl")
```

*Important:* The commands shown in the steps above must be entered as a single line in property files. Otherwise, use a command continuation symbol (\) between arguments.

## Creating a Custom Style Sheet

This section describes the custom XSLT style sheet you need to parse out the two forms described above.

1. Begin the style sheet with an XML header statement. This statement is required for all XML files:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

2. Continue with a style sheet definition statement:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
version="1.0">
```

3. Add statements for title, body and template matching criteria:

```
<xsl:template match="/"> _____ This line specifies which  
<html> nodes to process.  
<head>  
<title>Flight Status</title>  
</head>  
<body>  
<xsl:apply-templates select="*"> _____ This line is important. It says  
</body> to apply every applicable  
</html> template in the style sheet.  
</xsl:template>
```

- Define a template that parses the *status\_resv\_lhc\_us* form.

```

<xsl:template match="//
form[@name='status_resv_lhc_us']">
  <form>
  <xsl:copy-of select="* | @*" />
  </form>
</xsl:template>

```

This line searches every node for a form with this name attribute.  
 This line tells the parser to print all information in the form.

- In the example above, the *[@name='status\_resv\_lhc\_us']* statement basically means, “search every node for a form with the *status\_resv\_lhc\_us* name attribute.
- Define a template that parses the RESERVE form.

```

<xsl:template match="//form[@name='RESERVE']">
  <form>
  <xsl:copy-of select="* | @*" />
  </form>
</xsl:template>

```

Again, this line tells the parser to print out all information in the form.

- Define a final template that parses out the remaining page contents.

```

<xsl:template match="* | @* | text()">
  <xsl:apply-templates select="* | @* | text()" />
</xsl:template>

```

The first line matches for contents that hasn't been matched before. The second line recurses. The result is removal of elements that match.

- Add a closing statement for the style sheet.

```

</xsl:stylesheet>

```

- Save your custom transformation and copy the file to the location you set in the main application properties file. For example:

## Using Transformations

`install_dir\app\flightstatus\custom.xsl`

## Processing Result

After you test and debug the application, the following is an example of the how the flight simulator page should appear on a user handheld.

Locate a specific flight

Choose the airline:

Flight number:

---

If you do not know the airline and flight number, use the fields below:

Leaving from:

Going to:

Date of:

Simplified interface on handheld

Reproduced with permission of Yahoo! Inc. \* 2004 by Yahoo! Inc. YAHOO! and the YAHOO! logo are trademarks of Yahoo! Inc.

For more information on deploying and debugging applications, see “Testing and Debugging Applications” on page 22.

## Customizing Tips

The following are some things to remember when you are creating custom transformations:

- Telling the server which URLs to match and transforming pages are two separate processes.
- The `<xsl:apply-templates select="*" />` and `</xsl:template>` statements included in a custom transformation are standard XSLT functions.

- All content is added by default. Use the following code to remove everything you don't want:

```
<xsl:template match="*|@*|text()">
<xsl:apply-templates select="*|@*|text()"/>
</xsl:template>
```

- Keying in on specific attributes is the easiest way to transform:

```
<xsl:template match="//form[@name='RESERVE']">
<form>
<xsl:copy-of select="*|@*"/>
</form>
</xsl:template>
```

- Use lowercase only when adding element names and attribute values. (This is the current Good Intranet Server restriction.) For example:

```
<a href="http://www.name.com">Name</a>
```

instead of:

```
<A HREF="http://www.name.com">Name</A>
```

- Do not strip the content between `<head>....</head>`. This often includes Javascript function definitions or globally-executed Javascript code.
- Do not remove form elements in a form, or be careful about doing so when the Javascript on a page contains references to offsets within that form. For example, if a script refers to an element like the following:

```
window.document.myform.elements[163].value = "foo";
```

and the element has been stripped by a transformation, the page may not work correctly.

- Be careful about removing forms from pages. Sometimes forms are referred to by their offsets. For example:

```
var theForm = document.forms[2];
```

## Using Transformations

- For more details on creating custom transformations or integrating your applications on Good Intranet, contact Good Technology's Professional Services department.

## Example - Sample Intranet

Transformations are used to optimize various sections of the following sample intranet page for viewing on the handheld. In the following example, transformations, the *config.prop* file and properties files are modified in various ways so Good Intranet displays only particular parts of the page on the handheld.

The screenshot shows a desktop view of the ABC Corp Intranet. At the top is a navigation menu with links: Products, Solutions, Customers, Buy, Support, Partners, News, Events, Resources, Company, and Contact. Below the menu is a 'Welcome!' section with a message about the web portal and a 'The Markets' line chart. The chart shows data for days 10 through 4, with values ranging from 10,720 to 10,800. There are also sections for 'Enter Good Intranet™', 'Sample transformations on this page', and 'My Documents'.

ABC Corp Intranet

Products Solutions Customers Buy Support Partners News Events Resources Company Contact

HR Links

Jobs  
Personal  
Information  
Benefits  
FAQ

Contacts

Nancy Liu  
Adam Sim  
Randy Fudd  
Scott New

My Links

Procurement  
Legal  
Marketing

Search

WEB  
CNN.com

SEARCH

**Welcome!**

To ABC Corp On the web  
This page represents a typical web portal home page. Many types of users will interact daily with it and while on your average, wired, desktop browser it looks grand it is really not meant for the smaller form factor commonplace in today's wireless smartphones.

**Enter Good Intranet™**  
Good Intranet delivers business critical applications to the point of business, increasing sales and service productivity, improving customer responsiveness, and increasing business performance. Good Intranet uses the same secure AES encrypted transport layer used by GoodLink™ to connect mobile professionals to enterprise applications residing behind the firewall.

Good Intranet enables the reuse of your current web pages in the devices that you want. By leveraging industry standards such as XML and XSLT (and a little bit of our own magic!), Good Intranet makes it possible to transform your current pages so that your mobile users get the information that they need, with the best user experience possible. The following links show you how to configure your Good Intranet server to transform this very page!

**Sample transformations on this page**  
See the home page optimized!  
See only the search form!  
See only the left panel!  
See only the middle panel!  
See only the document list and the list of contacts!

**My Documents**  
Development Methodology  
Expense Matrix

The Markets  
as of 2:39 PM ET

Day	Value
10	10,750
11	10,780
12	10,800
1	10,760
2	10,740
3	10,730
4	10,720

To optimize this page for viewing on the handheld:

1. Modify the *config.props* file so Good Intranet can identify the pages that need to be transformed.

```
## Application support for the PortalSample transformations
app.PortalSample.selectors requestUrl=.*PortalPage.*
app.PortalSample.properties app\PortalSample\ps.props
app.PortalSample.reload true
```

2. Modify the properties file (*app\PortalSample\ps.props*) and define the default transformation you want to apply.

```
##
# PortalSample transformation - Optimizing the main page
#
# - Unroll tables
# - Custom XSLT
# - Put a "stamp" on the end of the page to show it's been reformatted
#
#
response.all.selectors requestUrl=.*PortalPage.*
response.all.transforms UnrollTables \
UnpooledXslt (transform=ps.xslt) \
Stamp (text="Reformatted by Good Intranet: %transformName%")
```

To view only the Search form on the page:

1. Modify the *config.props* file as follows.

```
## Application support for the PortalSample transformations
app.PortalSample.selectors requestUrl=.*PortalPage.*
app.PortalSample.properties app/PortalSample/ps.props
app.PortalSample.reload true
```

2. Modify the properties file (*app\PortalSample\ps.props*) as follows:

## Using Transformations

```
##
# PortalSample transformation - Optimizing the main page
#
# - Unroll tables
# - Custom XSLT
# - Put a "stamp" on the end of the page to show it's been reformatted
#
#
response.all.selectors requestUrl=.*PortalPage.*
response.all.transforms UnrollTables \
UnpooledXslt (transform=ps.xslt) \
Stamp (text="Reformatted by Good Intranet: %transformName%")
```

To view only the left panel of page:

1. Modify the *config.props* file as follows.

```
## Application support for the PortalSample transformations
app.PortalSample.selectors requestUrl=.*PortalPage.*
app.PortalSample.properties app/PortalSample/ps.props
app.PortalSample.reload true
```

2. Modify the properties file (*app\PortalSample\ps.props*) as follows:

```
##
# PortalSample transformation - displaying only the left pane section
#
# - Compress HTML
# - Custom XSLT
# - Put a "stamp" on the end of the page to show it's been reformatted
#
#
```

```
response.ps_leftpane.selectors requestUrl=.*PortalPage_leftPane.*
response.ps_leftpane.transforms UnpooledXslt
(transform=ps_leftpane.xslt) \
Stamp (text="Reformatted by Good Intranet: %transformName% ")
```

To view only the middle pane of the page:

1. Modify the *config.props* file as follows:

```
## Application support for the PortalSample transformations
app.PortalSample.selectors requestUrl=.*PortalPage.*
app.PortalSample.properties app\PortalSample\ps.props
app.PortalSample.reload true
```

2. Modify the properties file (*app\PortalSample\ps.props*) as follows.

```
##
# PortalSample transformation - displaying only the mid pane section
#
# - Compress HTML
# - Custom XSLT
# - Put a "stamp" on the end of the page to show it's been reformatted
#
#
response.ps_midpane.selectors requestUrl=.*PortalPage_midPane.*
response.ps_midpane.transforms UnpooledXslt
(transform=ps_midpane.xslt) \
Stamp (text="Reformatted by GoodAccess: %transformName% ")
```

3. If necessary, write your custom XSTL

To view list of documents and contacts:

1. Modify the *config.props* file as follows:

```
## Application support for the PortalSample transformations
```

## Using Transformations

```
app.PortalSample.selectors requestUrl=.*PortalPage.*
app.PortalSample.properties app/PortalSample/ps.props
app.PortalSample.reload true
```

2. Modify the properties file (*app\PortalSample\ps.props*) as follows:

```
##
# PortalSample transformation - displaying a combination of elements
#
# - Custom XSLT
# - Put a "stamp" on the end of the page to show it's been reformatted
#
#
response.ps_combo.selectors requestUrl=.*PortalPage_combo.*
response.ps_combo.transforms UnpooledXslt
(transform=ps_combo.xslt) \
Stamp (text="Reformatted by Good Intranet: %transformName% ")
```

You can also write custom XSTL as needed.

## Using Transformations to Control Site Access

If using a proxy server is an option to control user access to unauthorized sites and there is a limited and known quantity of URLs that users need access to while using Good Intranet, use the following transformations:

1. Define the selectors for the known URL's via the following syntax:

```
app.app1.selectors requestUrl=*www.app1.com*
app.app1.properties app/app_handler.props
```

```
app.app1.reload    true
```

...

```
app.appn.selectors    requestUrl=*www.app1n.com*
app.appn.properties  app/app_handler.props
app.appn.reload      true
```

The properties file for these selectors can be included in one file, as the example above, or in individual files. This depends on how you want to configure the Good Intranet Server.

2. For the other sites, use the generic selector that is included with Good Intranet:

```
app.generic.selectors    requestUrl=.*
app.generic.properties  app/generic.props
app.generic.reload      true
```

In this case, the only transformation that should be available in the generic.props should be:

```
response.all.selectors    requestUrl=.*
response.all.transforms  XSLT (text=hideall.xslt)
```

where the content of the file *hideall.xslt* is as follows:

```
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/
XSL/Transform">
<!-- this defines the html document -->
<xsl:template match="/*">
<html>
<head>
<title>Access not allowed</title>
```

## Using Transformations

```
</head>
```

```
<body>
```

*Content was filtered out by Good Intranet. If you believe you need access to this site please contact your Good Intranet administrator.*

```
<!-- This matches all the templates defined below - effectively running all the templates -->
```

```
<xsl:apply-templates select="*" />
```

```
</body>
```

```
</html>
```

```
</xsl:template>
```

# 3 Developing Custom Applications

---

This chapter describes how to develop and extend custom applications for use with Good Intranet.

## Understanding How Applications Work With Good Intranet

When a wireless handheld requests an HTML page, the page sent in response to the request passes through two selection processes and is then configured according to the results of those processes.

- In the first process, Good Intranet Server attempts to identify the specific Web application the page is requested from. It uses the request URL and any POST data in the body of the request to make the identification. If a specific application can't be identified, it assigns the page to a *generic* application.
- In the second process, Good Intranet Server selects the page for transformation based on settings in the application's configuration file.

Once selected, a page can be transformed to reformat, truncate, and otherwise scale it down to make it suitable for viewing on the small screen of wireless handhelds (for example, removing images, unrolling tables, and truncating pages over a certain size).

## Developing Custom Applications

Good Intranet Server includes a configuration file for the generic applications (*generic.props*). In addition, application developers can identify custom applications used by Good Intranet handhelds and creating separate configuration files for each custom application. In practice, this means:

1. Specifying the request URL and/or POST data to select the application.
2. For each application, specifying the request URLs, response URLs and/or POST data to select pages for transformation.
3. Specifying appropriate content transformations for each page.

You use the Good Intranet properties file (*config.props*) for Item 1 listed above and application-specific properties files for the other items.

## Understanding the Benefits of This Approach

The application and page selection options provided by Good Intranet Server are robust. They enable application developers to fine tune transformations for very specific purposes.

For highly-structured applications and pages that include consistent sets of data, you can define and apply simple application selection and transformation criteria.

For robust applications that include a variety of data sets and page designs, you can define and apply highly-customized application and selection criteria.

For example:

- A simple application that looks up email addresses from a corporate directory, might work just fine with the settings already defined in the *generic* application properties file.
- A structured accounting application that extracts spreadsheet data from a single database, might need a customized application

properties file that applies a single set of transformations to all pages in the application.

- A complex sales lead, order tracking system might need multiple application properties files, each with a large set of customized transformations that apply to individual, customized page types.

### Identifying the Application Requested

The Good Intranet Server properties file has one or more property sets to identify the application requested. One set is for a *generic* application—any application not identified specifically. The other sets are for specific applications (one set for each application).

Properties included in each set are *app.name.selectors*, *app.name.properties*, and *app.name.reload*.

Sets of properties are identified by the *name* element they share. For example, *app.generic.selectors*, *app.generic.properties*, and *app.generic.reload* comprise the *generic* application set.

By default, the Good Intranet Server properties file comes with the properties set for *generic* applications and no others (*apps/generic.props*).

Properties to identify requested applications include:

---

Property Name	Default	Description
<code>app.name.selectors</code>	A list of one or more selectors.	<p>A list of one or more selectors for matching against the request URL and any POST data in the request.</p> <p>Each selector has a <i>requestUrl</i> component and a <i>postData</i> component.</p> <p>If one of the selectors in instance #1 of the property is a better match than any selector in any other instance of the property, the application identified by <i>name</i> in instance #1 is specified. Selector #1 is a better match than selector #2 if it is <i>narrower</i>—that is, if it allows for fewer possible matching URLs.</p> <p>For the syntax of selector components and the algorithm for comparing and ranking matches, see “Selector Component Syntax” on page 50.</p>
<code>app.name.properties</code>	<code>app/name.props</code>	Identifies the configuration file for the application identified by <i>name</i> . (The default application directory is <i>install_dir\app</i> .)

Property Name	Default	Description
<code>app.name.reload</code>	<code>false</code>	Determines if properties for the application identified by <i>name</i> are reloaded when they change. If true, for every request that matches a selector in <code>app.name.selectors</code> , the timestamp of the properties file for <i>name</i> is compared to the last-loaded timestamp and the file is reloaded if it is newer.
<code>app.thinclient.properties</code>	<code>app/name.props</code>	Identifies the configuration file for the application identified by <i>name</i> . This is the same properties file used by the generic application profile. (The default application directory is <code>install_dir\app</code> .)

By default, the *generic.app* properties are:

```
app.generic.selectors requestUrl=.*
# app.generic.properties app/generic.props
# app.generic.reload false
```

The value of the `app.generic.selectors` property (`requestUrl=.*`) is a one-component selector equivalent to the following two-component selector:

```
(requestUrl=.* postData=.*)
```

- In the first property, the *postData* component is not required because the default for any component is “`.*`” (matches anything).
- Parentheses are used, when necessary, to group the components of a selector. The selector in the `app.generic.selectors` property doesn’t need parentheses because it contains only one component.

If the alternative selector had no parenthesis, it would be interpreted as a list of two separate selectors (each with one component), with *postData=.\** as the first selector and *requestUrl=.\** as the second.

- The selector *requestUrl=.\** is “generic” in the sense that it is a match for any possible request. That is, each of its components (*explicit requestUrl* and *implicit postData*) match any possible item of its type in a request.
- Both components of the selector have to match for the selector to count as a match. If the selector were defined as *requestUrl=.\*://yourhost.\** and the request URL didn’t include *yourhost*, the selector would not be a match even though the implied *postData=.\** component was a match.
- The *app.name.properties* and *app.name.reload* properties have default values, so they are functional even if they are commented out. If you are creating a property set and the *app.name.properties* and *app.name.reload* properties use default values, you do not have to include them in the set.

The properties *app.name.selectors* and *app.name.properties* for a set work together as follows:

- If the browser’s request URL and POST data match the *requestUrl* value and *postData* value of the property *app.name.selectors* for an application, this match causes the properties in the configuration file for that application (*app.name.properties*) to be applied to the response (i.e., to the page returned in response to the request).
- Because the *requestUrl* and *postData* values defined by *app.generic.props* are the broadest possible values (.\*), all request URLs or POST data will match those values. Therefore, if there is only one, *generic* set of properties defined, the configuration file *generic.props* applies to all requests.

- The properties in *generic.props* will be applied only if there is no match with a selector for the application. For example:
  - If you have provided a set of properties for a specific application,
  - Its selectors are narrower than *.\**, and
  - The browser's request URL and POST data matches one of those selectors.

The properties in the configuration file for that application will be applied instead of the properties in *generic.props*.

## Selecting Pages and Providing Transformations

The configuration file *generic.props*, and any other configuration file you create, has one or more sets of *response.name.selectors* and *response.name.transforms* properties. Each property set is identified by the *name* element they share. These properties define how response pages are transformed.

In the default *generic.props* file, there is one set (*response.all.selectors* and *response.all.transforms*) that apply to all response pages.

Property Name	Default	Description
<i>response.name.selectors</i>	A list of one or more selectors.	<p>A list of one or more selectors for matching against the request URL, any POST data in the request, and the response URL. (The request URL and response URL may be two different URLs.)</p> <p>Each selector has three components: <i>requestUrl</i>, <i>postData</i>, <i>responseUrl</i></p> <p>If one of the selectors in the first instance of the property is a better, narrower match (i.e., allows for fewer possible matching URLs) than selectors in other instances of the property, the response identified by <i>name</i> in the first instance of the property is specified and the transformations listed in <i>response.name.transforms</i> are used.</p> <p>For the syntax of selector components and the algorithm for comparing and ranking matches, see “Selector Component Syntax” on page 50.</p>
<i>response.name.transforms</i>		<p>Lists the content transformations. For a list of possible transformations, see the configuration file, <i>app/generic.props</i>.</p>
<i>response.name.xpath</i>		<p>Provides xpath expressions to further refine a selection. Expressions that contain spaces must be surrounded by double quotes (“”).</p> <p>For more information, see “Using Xpath to Refine a Selection” on page 52.</p>

A set of *response.name.selectors* and *response.name.transforms* properties work together as follows:

- A match is defined by comparing the *requestUrl* value and the URL of the request, the *postData* value and the POST data in the request, and the *responseUrl* value and the URL of the response page.
- If an application has multiple selectors defined in one or more instances of the *response.name.selectors* property, only the narrowest match is applied.
- A match causes the transformations for that response name to be applied to the response (for example, to the page returned in response to the request).

For the *generic.props* file, the *response.all.transforms* property has the following values:

```
TransformIfLarge(size=6000) \  
RemoveImgs \  
CleanTables \  
UnrollTables \  
Truncate \  
Stamp (text="<Reformatted by Good Intranet>")  
AutoScaleImage
```

*TransformIfLarge* is a switch that controls successive transformations listed below it in the file. By default, *TransformIfLarge* triggers these successive transformations only if the page is at least 6000 bytes.

Additional content transformations are listed at the end of the file. By default these transformations are commented out, but you can use any combination of the transformations in *generic.props* to build configuration files for specific applications.

## Selector Component Syntax

Syntax for selector components include:

---

Component Type	Example	Description
requestUrl	<code>requestUrl=.*//yourhost/.*</code>	A regular expression that is matched against the request URL.
postData	<code>postData=postVariable=Post \\+data</code>	A regular expression that is matched against the POST data in the request.  <b>Note:</b> POST data is generally URL form encoded, with spaces replaced by a plus (+) sign. The regular expression sequence is a backslash preceding the plus (\+), however in a properties file, the backslash must be doubled \\+ (literal backslash followed by plus).
responseUrl	<code>responseUrl=.*login.*</code>	A regular expression that is matched against the response URL. (The response URL differs from the request URL if the origin server redirects the request.)
contentType	<code>contentType=.*MSExcel</code>	A regular expression that is matched against the content type of the response.

The property `app.name.selectors` in the Good Intranet properties file uses only the `requestUrl` and `postData` components. The property `response.name.selectors` in the application configuration file uses `requestUrl`, `postData` and `responseUrl` components. The property `mime.name.selectors` uses `responseUrl` and `contentType` components.

Note the following:

- Selector components are separated by a space. For example:

```
(requestUrl=.*URLPATTERN.* postData=.*Submit.*)
```

Another example:

```
(\
requestUrl=.*URLPATTERN.* \
postData=.*Submit.*\
)
```

In the second example, the space between the first component and the backslash (\) at the end of the first line separates the two components.

- A list with multiple selectors can select an application by both its DNS name and its IP address:

```
requestUrl=.*://mydomainname.* \
requestUrl=.*://1\\.2\\.3\\.4/*
```

- If an application is selected but the *response.name.selectors* property in its configuration file provides no matches, no transformations will occur. For this reason, consider copying the property set (*response.all.selectors* and *response.all.transforms*) from *generic.props* into configuration files for specific applications. This will apply the all-purpose generic transformations in these cases.

## Selector Matching and Ranking

The following algorithm is used to choose the best-matching selector:

- For application selectors, the two components are matched against the request URL and POST data. For response transformation selectors, the three components are matched against the request URL, POST data, and the response URL.
- A selector is ranked only when each of its components is a match.

## Developing Custom Applications

- The highest ranking selector is chosen.  
Although a selector can be ranked only if all its components match, ranking is by individual component. *requestUrls* are considered first, and *postData*s considered only if there is a tie in *requestUrls*. *responseUrls* are considered only for ranking response transformation selectors, and only if there is a tie in *postData*s.  
Each component is ranked by narrowest match. Narrowness is determined by counting the number of literal characters in the regular expression: The more literal characters, the narrower the match. For ranking purposes, the regular-expression “character-set” is equivalent to a literal character. Thus *a.\** is narrower than *.\** but the four following expressions are equally narrow:

*“a”, “b.\*”, “.\*c” “\+.\*”, “[a-z].\*”*

*Note: The \+ is the way to specify a literal plus sign when defined in a properties file. Generally, a literal backslash must be doubled in a properties file.*

- A tie in the selectors is broken by choosing the profile with the “alphabetically least” profile name. For example, with the following selectors:

```
app.a.selectors requestUrl=.*  
app.b.selectors requestUrl=.*
```

If the “a” and “b” profiles are the best matches, the “a” profile will always be preferred over “b.”

## Using Xpath to Refine a Selection

In cases where *requestUrl*, *postData* and *responseUrl* all match, you can use an *xpath* property to further refine a transformation selection. *xpath* is an optional property (added to the application profile) that uses one or more *xpath* expressions to search the document object model (DOM) for specific content/elements.

Use the following structure to configure the *xpath* property:

```
response.A.selectors...
```

```

response.A.transforms...
response.A.xpath an_xpath_expression(s)
response.B.selectors...
response.B.transforms...
response.B.xpath another_xpath_expression(s)

```

If the *xpath* property has more than one *xpath* expression, they are separated by a blank space.

*Important:* If your *xpath* expression include spaces, make sure you include the expression in double quotes (“”).

If both *A* and *B* match via selectors, then each of its *xpath* expressions are evaluated. The best matching transformation is chosen by the following criteria:

1. All of the transformation's *xpath* elements must be true (that is, the DOM element exists).
2. The transformation with the most *xpath* elements is the best match.

One of the *xpath* expressions is evaluated. If the expression is true (that is, the DOM element exists), then the selector associated with that *xpath* is chosen. Otherwise, if the other *xpath* expression is true, the other selector is chosen.

Limitations to this method are:

- The *xpath* selector property only accepts a *xpath* expression that returns a node set. Expressions that return a boolean value, numbers, strings, etc. are not supported.

## Developing Custom Applications

The following is an example of xpath configuration:

```
response.A.selectors  requestUrl=.*
response.A.transforms Stamp (text="Transformed by
%transformName% ")
response.B.selectors  requestUrl=.*
response.B.transforms Stamp (text="Transformed by
%transformName% ")
response.B.xpath      //table/tr/form[@class='small']
```

If A's selectors tied in a match with B's selectors, then B's xpath expression will be tested against the DOM. If it is true, the B transformation will be used. Otherwise, the A transformation will be picked (because ties are broken by looking at the transformation names and A is less than B).

### Ignoring Case

If desired, you can set all response transformation selectors defined for a given application to ignore case. You can also set the regular expressions used to select applications to ignore case.

To set all response transformation selectors in an given application to ignore case, add the following property to the properties file for the application:

```
response.selectors.ignorecase true
```

For example, adding this property to the *apps/myapp.props* file causes all response transformation selectors defined for the *myapp* application to be case insensitive.

Similarly, you can set application profile selection on Good Intranet Server to ignore case. When the following property is set to *true* in the Good Intranet *config.props* file, the regular expressions used to select applications are case insensitive.

```
app.selectors.ignorecase true
```

## Setting Application Properties

By default, the Good Intranet Server properties file comes with a single set of properties for generic applications (*apps/generic.props*) and no others. However, because Good Intranet Server can handle different applications with different transformations, Good Technology recommends using application-specific profiles. This means:

- Configuring the *config.props* file to have multiple *app.name.selectors* settings.
- Preparing multiple *apps/name.props* files.

Application-specific profiles are preferable to grouping all transformations into the single *apps/generic.props* file.

## Creating an HTML Form for Uploading Files

You can create an form using HTML and your users can use that form to upload files to the server. When the HTML form is designed to submit files over the server, it includes the attribute with the value as *multipart/form-data*. The HTTP method is always POST in this case.

The value of attribute type is set as *file* for the INPUT element, to specify the file upload:

```
<input type="file" name="fieldname"/>
```

When the user accessed the form from the handheld and the form is filled and submitted, Good Intranet client sends the POST request in a multi-part format to the origin server and the file is uploaded.

The content type of the file being uploaded is generally sent with the request as a header. A fixed content type header is added to all the file parts. Which is, *Content-type: application/octet-stream*. This is a default content type used by most of the browsers. All other input

## Developing Custom Applications

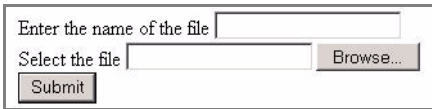
types (text, checkbox, radio and so on) are also sent as part of the MIME message.

### *Example HTML form for uploading files:*

The following example shows an HTML form for users to upload files from their handhelds:

```
<HTML>
<HEAD>
<TITLE>Form enctype Test</TITLE>
</HEAD>
<FORM method="post" action="/test.cgi" >
Enter the name of the file <input type="text" name="name"> </br>
Select the file <input type="file" name="fileInput"></br>
<input type="Submit" name="Submit" value="Submit">
</FORM>
</HTML>
```

On the handheld, the form is displayed to the user as the following:



Enter the name of the file

Select the file

## Restricting File Size

Handhelds are limited by memory, so you can set the maximum size of the file being uploaded. If multiple files are being uploaded, the sum of the size of all of the files is considered. You can specify the maximum file size through the Good Intranet Console or by setting the client. User cannot specify an empty value in this field and the value has to be a numeric value greater than or equal to 0 and less than or equal to 800. The value is specified in KB (kilobytes) only. Server ensures that the value sent to the client is valid.

## Supported HTML Elements

The following table lists HTML Elements and attributes and indicates whether they are supported on Good Intranet handhelds. See also “Additional Restrictions, Limitations and Notes” on page 66 and “HTML Color Support” on page 66.

Element	Attribute	Supported (Y/N)	
A		Y	
		CHARSET	N
		TYPE	N
		NAME	N
		HREF	Y
			Link to phone number supported. For example, <code>&lt;a href="tel:+408-327-6000"&gt;</code>
		HREFLANG	N
		REL	N
		REV	N
		ACCESSKEY	N
		SHAPE	N
		COORDS	N
		ONFOCUS	N
		ONBLUR	N
	ONCLICK	N	
ABBR		Y (always boldface)	
ACRONYM		Y (always boldface)	
ADDRESS		N	
APPLET		N	
AREA		N (includes image maps)	
B		Y	
BASE		N	
	HREF	N	
	TARGET	N	

## Developing Custom Applications

---

Element	Attribute	Supported (Y/N)
BASEFONT		N
	SIZE	N
BDO		N
BIG		Y
BLOCKQUOTE		N
	CITE	N
BODY		Y
	ALINK	N
	BACKGROUND	Y
	BGCOLOR	Y
	LINK	N
	ONLOAD	Y
	ONUNLOAD	N
	TEXT	N
	VLINK	N
BR		Y
	CLEAR	N
BUTTON		Y (mapped to input)
	NAME	Y
	VALUE	Y
	TYPE=(BUTTON   SUBMIT)	Y
	TYPE=RESET	N
	DISABLED	Y
	TABINDEX	N
	ACCESSKEY	N
	ONFOCUS	N
	ONBLUR	N
ONCLICK	Y (mapped to input)	
CAPTION		N
	ALIGN	N
CENTER		Y
CITE		Y (always boldface)

Element	Attribute	Supported (Y/N)
CODE		Y (always boldface)
COL		N
COLGROUP		N
DD		N
DEL		N (renders as normal font)
DFN		Y (always boldface)
DIR		N
DIV	COMPACT	N
		Y
	ALIGN	Y
	BACKGROUND COLOR	Y
DL		N
	COMPACT	N
DT		N
EM		Y (mapped to bold)
FIELDSET		N
FONT		Y
	COLOR	N
	FACE	N
	SIZE	N
	WEIGHT	N
FORM		Y
	ACTION	Y
	METHOD	Y
	ENCTYPE	Y
	ACCEPT	N
	NAME	N
	ONRESET	N
	ONSUBMIT	Y
	ACCEPT- CHARSET	N
FRAME		N

## Developing Custom Applications

---

Element	Attribute	Supported (Y/N)
FRAMESET		N
H1 - H6		Y
	ALIGN	Y
HEAD		Y
	PROFILE	N
HR		Y
	ALIGN	N
	NOSHADE	N
	SIZE (1-5)	Y
	WIDTH	N (non-percentage widths are supported)
HTML		Y
I		Y (mapped to bold)
IFRAME		N
IMG		Y
	ALIGN	N
	ALT	Y
	BORDER	N
	HEIGHT	N
	HSPACE	N (deprecated in HTML 4.01)
	ISMAP	N
	NAME	N
	SRC	Y
	USEMAP	N
	VSPACE	N (deprecated in HTML 4.01)
	WIDTH	N
INPUT	TYPE=button	Y
	NAME	Y
	VALUE	Y

---

Element	Attribute	Supported (Y/N)
INPUT	TYPE=checkbox	Y
	CHECKED	Y
	NAME	Y
	VALUE	Y
	DISABLED	N
	READONLY	N
	ONFOCUS	N
	ONBLUR	N
	ONSELECT	N
	ONCHANGE	Y
	ONCLICK	Y
INPUT	TYPE=file	Y
INPUT	TYPE=hidden	Y
	NAME	Y
	VALUE	Y
	DISABLED	N
	READONLY	N
	ONFOCUS	N
	ONBLUR	N
	ONSELECT	N
	ONCHANGE	N
	ONCLICK	N
INPUT	TYPE=image	(mapped to submit)
	ALIGN	N
	ALIGN	N
	ALT	N
	NAME	Y
	SRC	N
	ONCLICK	Y

## Developing Custom Applications

Element	Attribute	Supported (Y/N)
INPUT	TYPE=password	Y
	MAXLENGTH	Y
	NAME	Y
	SIZE	Y
	VALUE	Y
INPUT	ONCLICK	N
	TYPE=radio	Y
	CHECKED	Y
	NAME	Y
	SIZE	N
INPUT	VALUE	Y
	ONCLICK	N
	TYPE=reset	Y (simple input field supported, but text area is not supported)
	NAME	N
	VALUE	N
INPUT	ONCLICK	N
	TYPE=scribble	N
	NAME	N
	SIZE	N
	VALUE	N
INPUT	ONCLICK	N
	TYPE=submit	Y
	NAME	Y
	VALUE	Y
	ONCLICK	N
INPUT	TYPE=text	Y
	MAXLENGTH	Y
	NAME	Y
	SIZE	Y
	VALUE	Y
	ONCLICK	N

---

Element	Attribute	Supported (Y/N)
	DISABLED	N
INS		Y (renders as normal font)
ISINDEX		N
KBD		Y (always boldface)
LABEL		N
LEGEND		N
LI		Y
	TYPE	N
	VALUE	N
LINK		N
MAP		N
MENU		N
	COMPACT	N
META		Y
	CONTENT	Y
	HTTP-EQUIV	Y
	NAME	Y
	SCHEME	N
NOFRAMES		N
NOSCRIPT		N
OBJECT		N
OL		Y
OPTGROUP		N
OPTION		Y
	SELECTED	Y
	VALUE	Y
P		Y
	ALIGN	Y
	BACKGROUND	Y
PARAM		N
PRE		Y
	WIDTH	N

## Developing Custom Applications

---

Element	Attribute	Supported (Y/N)
Q		N
	CITE	N
S		N (renders as normal font)
SAMP		Y (always boldface)
SCRIPT		Y
	LANGUAGE	N (Javascript only)
	SRC	Y
	TYPE	N
	DEFER	N
SELECT		Y
	MULTIPLE	Y
	NAME	Y
	SIZE	Y
	DISABLED	N
	TABINDEX	N
	ONFOCUS	N
	ONBLUR	N
	ONCHANGE	Y
	ONCLICK	N
SMALL		N
SPAN		Y (some fonts are not supported)
STRIKE		N (renders as normal font)
STRONG		Y
STYLE		N
SUB		N
SUP		N

Element	Attribute	Supported (Y/N)
TABLE		Y
	ALIGN	N
	BGCOLOR	Y
	BORDER	Y (including support for variable border width)
	BORDERCOLOR	Y
	CELLPADDING	N
	CELLSPACING	N
	WIDTH	Y
TBODY		N
TD		Y
	ALIGN	Y
	BGCOLOR	Y
	COLSPAN	Y
	HEIGHT	N
	NOWRAP	N
	ROWSPAN	N
	VALIGN	N
TEXTAREA	WIDTH	Y
		Y
	COLS	Y
	NAME	Y
	ROWS	Y
TEXTAREA	ONCHANGE	Y
		N
TFOOT		N
TH		Y
	ALIGN	Y
	COLSPAN	Y
	HEIGHT	N
	NOWRAP	N
	ROWSPAN	N
	VALIGN	N
WIDTH	Y	

Element	Attribute	Supported (Y/N)
THEAD		N
TITLE		N
TR		Y
	ALIGN	Y
	BGCOLOR	Y
	VALIGN	N
TT		N
U		Y
UL		Y
	COMPACT	N
	TYPE	N
VAR		Y (always boldface)
XMP		N

### Additional Restrictions, Limitations and Notes

In addition to the “Supported HTML Elements” table above, this section includes some additional clarifications about HTML limitations and restrictions. Most restrictions and limitations are the result of handheld-specific designs.

- A break `<BR>` embedded in list of items, treats the break as an additional list item and, in an ordered list, increments the list number.

## HTML Color Support

Good Intranet Server has an HTML preprocessor that converts HTML color-specifying elements and/or attributes into other HTML tags and/or attributes. This “tidied” HTML is then converted into a Good Document Object Model (GDOM) object and is sent to the handheld as serialized GBDOM.

Color values can be named or can be represented in standard HTML RGB format:  $xRrGgBb$  is the color with red= $0xRr$ , green= $0xGg$ , blue= $0xBb$ . Good Intranet supports the standard colors that are supported by Internet Explorer. For a complete list of colors, see:

*<http://msdn.microsoft.com/library/default.asp?url=/workshop/author/dhtml/reference/colors/colors.asp>*.

The following table lists the system color substitution for some color-specifying HTML elements:

*Elements Mapped to New System Color*

---

<b>Element</b>	<b>New Color Value</b>
ActiveBorder	0x990000
ActiveCaption	0x990000
AppWorkspace	0xFFFFFFFF
ButtonFace	0xC0CCD9
ButtonHighlight	0xFFFFFFFF
ButtonShadow	0x808080
CaptionText	0xFFFFFFFF
GrayText	0x808080
Highlight	0x990000
HighlightText	0xFFFFFFFF
InactiveBorder	0xC0C0C0
InactiveCaption	0x808080
InfoBackground	0xCCFFFF
Menu	0xFFFFFFFF
Scrollbar	0xC0C0C0
ThreeDLightShadow	0xFFFFFFFF
Window	0xFFFFFFFF

## Developing Custom Applications

The following table lists the color-specifying HTML elements that may not be mapped to a system color:

---

### HTML

ActiveBorder  
ActiveCaption  
AppWorkspace  
Background  
ButtonFace  
ButtonHighlight  
ButtonShadow  
ButtonText  
CaptionText  
GrayText  
Highlight  
HighlightText  
InactiveBorder  
InactiveCaption  
InactiveCaptionText  
InfoBackground  
InfoText  
Menu  
MenuText  
Scrollbar  
ThreeDDarkShadow  
ThreeDFace  
ThreeDHighlight  
ThreeDLightShadow  
Window  
WindowFrame  
WindowText

The following table shows the currently supported color-specifying HTML elements and attributes and how they are encoded as a

GBDOM element. In each case, the *color* is a supported name or hex value:

---

#### HTML

```
<font color="color">
<div style="color: color">
```

#### GDOM encoding

```
<span color=value>
<div color=value>
```

## Unsupported HTML

Currently, the server does not encode the following HTML elements:

---

#### HTML

```
<body text="color">
<body link="color">
<body vlink="color">
<body alink="color">
```

## Image support

Currently the handheld requests images by specifying the image URL and passing an *Accept* header that includes one or more image MIME types. The following graphics formats are discussed:

---

Acronym	Description	Type
GIF	Graphics Interchange Format	vector
JPEG	Joint Photographic Experts Group	raster
PNG	Portable Network Graphics	vector

*Note:* The colors in PNG graphics may not render correctly on some Palm OS handhelds.

## Developing Custom Applications

The server converts from the source MIME type to the accepted MIME type using the Java Advanced Imaging (JAI) library. This library provides the ability to read/write various graphics formats and apply transformations. The following transformations are performed.

### *Image Transformations*

---

Source MIME Type	Destination MIME Type
BMP	JPEG
GIF	null
JPEG	null
PNG	JPEG
WBMP	JPEG
TIFF	JPEG

## Audio and Video Support

Good Mobile Intranet supports the audio/mpeg and video/x-msvideo mime types. When these mime types are accessed, Good Intranet hands them off to the appropriate native application to open.

To access files with mime types other than these supported mime types (e.g. audio/x-wav, video/avi), use the Mime Type converter module on the Good Intranet Server to convert the mime types to audio/mpeg and video/x-msvideo.

## MIME Type Conversion

Good Intranet Server allows you to override the content type declared by the origin web server. Some times web server sends the incorrect (non-standard) MIME types, which results in content not be handled correctly.

MIME types can be overridden based on the URL being accessed, based on the content type the web server returns, or both.

Include the following settings in the config.props configuration file:

```
mimeconverter.enable  
mime.name.selectors  
mime.name.result
```

The properties are explained below in details:

<b>Property name</b>	<b>Default</b>	<b>Description</b>
mimeconverter.enable	<i>true</i>	Enables the MIME type conversion mechanism module.

## Developing Custom Applications

<code>mime.name.selectors</code>	A list of one or more selectors	<p>A list of one or more selectors used to match against the response URL (The request URL and response URL may be two different URLs.) and content type returned by the web server. Each selector has two components: <code>responseUrl</code>, <code>contentType</code></p> <p>If one of the selectors in the first instance of the property is a better, narrower match (for example, allows for fewer possible matching URLs) than selectors in other instances of the property, the response identified by name in the first instance of the property is specified and MIME type specified in <code>mime.name.result</code> is used. For the syntax of selector components and the algorithm for comparing and ranking matches, see “Selector Component Syntax” on page 50.</p>
<code>mime.name.result</code>		<p>Specifies the value of this property would be used as the content type of the content of the response, ignoring the content type declared by the web server.</p>

By default, Good Intranet server is configured with mappings to correct the non-standard MIME type returned by the server.

The following properties are present in `config.props` by default:

*`mimeconverter.enable true`*

*mime.word.selectors contentType=application/MSWord*  
*mime.word.result application/msword*  
*mime.excel.selectors contentType=application/MSExcel*  
*mime.excel.result application/x-msexcel*  
*mime.ppt.selectors contentType=application/MSPowerpoint*  
*mime.ppt.result application/x-mspowerpoint*

## Character Encoding Standards

Good Intranet supports the following encoding standards:

- Latin-1 (ISO-8859-1)
- UTF-8
- US-ASCII

## Cache Control

For pages that do not include any cache control information (including last-modified), a default expiration is assigned. For user requested pages, the default is 60 seconds. Images is 14 days. Pushed documents is 45 days.

## Unsupported WML Features

With minor exceptions, Good Intranet handhelds support *Wireless Markup Language Specification Version 1.3*. The following table lists features of WML 1.3 that are either unsupported or (as allowed by the specification) unimplemented.

---

Element	Attribute	Unsupported/ Unimplemented	Comment
a	title	Unimplemented	
anchor	title	Unimplemented	

Element	Attribute	Unsupported/ Unimplemented	Comment
card	ordered	Unimplemented	Always treated as <i>true</i> .
do	optional	Unimplemented	Always treated as <i>false</i> .
fieldset		Unimplemented	Handheld ignores the <i>fieldset</i> element but displays all fields specified under <i>fieldset</i> .
go	accept-charset	Unsupported	If the origin server sends a WML deck containing <i>go accept-charset</i> , the handheld ignores it and sends back the list of character sets it supports: UTF-8 Latin 1 (ISO-8859-1) US ASCII

Element	Attribute	Unsupported/ Unimplemented	Comment
img	height	Unimplemented	
	width	Unimplemented	
	map id	Unsupported	
	<i>vspace="percent"</i>	Unimplemented	If you enter <i>vspace</i> as a percentage, the handheld ignores it. If you enter <i>vspace</i> as a number of pixels, the handheld considers it but can not always follow it in displaying the image.
	<i>hspace="percent"</i>	Unimplemented	If you enter <i>hspace</i> as a percentage, the handheld ignores it. If you enter <i>hspace</i> as a number of pixels, the handheld considers it but can not always follow it in displaying the image.
	localsrc	Unsupported	The handheld always uses the URI specified in the <i>img src</i> attribute.
input	tabindex	Unimplemented	
	title=	Unimplemented	
meta	name	Unimplemented	When present, this meta element is ignored.
	forua	Unimplemented	Always treated as if <i>forua="true"</i> .
	scheme	Unimplemented	When present, this meta element is ignored.

Element	Attribute	Unsupported/ Unimplemented	Comment
optgroup		Unimplemented	The handheld ignores the <i>optgroup</i> element but displays all options specified under <i>optgroup</i> .
p	mode	Unimplemented	Paragraph text always wraps, regardless of <i>mode</i> .
select	tabindex	Unimplemented	
small		Unsupported	Appears as plain text in default size.

## Support for JavaScript

The table below lists JavaScript properties and functions that are supported by Good Intranet on the handheld.

### Notation

---

Y	Supported. Can be used in scripts.
I	Ignored. Scripts containing this will run, but the property or function return the empty string.
N	Not supported. Scripts containing this will not run beyond the point where this is used.
*	Support is built in to Nombas' ScriptEase library. Assumes full support for the ECMA Standard 262.
+	Disabled in the Nombas ScriptEase library for PDA handhelds like the Palm.
**	Not possible due to hardware constraints.
a	IE version string is "4.0 (compatible; MSIE 5.5; Windows NT 5.0; T312461)"
b	Handheld type is currently one of: "PalmOS," "PocketPC," "Desktop," or "Unknown."

*JavaScript Support*

Item	Support (Y/N)	Notes
Array	Y*	
Boolean	Y*	
Date	Y*	
Function	Y*	
Math	Y**	Partial: The degree of floating point precision supported by Good's implementation of Javascript depends on the floating support of each device platform. The Pocket PC version of Good Intranet supports up to 16 digits of precision, and the Treo 600 supports up to 9 digits.
Number	Y*	
Object	Y*	
String	Y*	
RegExp	N <sup>+</sup>	
<b>Object events</b>		
onAbort	N	Very rarely used.
onBlur	Y	
onClick	Y	Button, Image, Checkbox, Radio
		Button, Reset, Submit
onDbClick	N	

### JavaScript Support

---

Item	Support (Y/N)	Notes
onChange	Y	Drop Down, Select, Text, Edit, Radio Button, Checkbox
onDragDrop	N**	Very rarely used.
onError	N	
onFocus	N	
onKeyDown	N	
onKeyPress	N	
onKeyUp	N	
onLoad	Y	In HTML <body>.
onMouseDown	N**	
onMouseMove	N**	
onMouseOut	N**	
onMouseOver	N**	
onMouseUp	N**	
onMove	N**	
onReset	N	
onResize	N**	
onSelect	N	
onSubmit	Y	Terminates a form submission if the handler explicitly returned <i>false</i> . If a return value was not specified, submission is allowed.
onUnload	N	
<b>Dialogs</b>		
alert	Y	
prompt	Y	
confirm	Y	

*JavaScript Support*

Item	Support (Y/N)	Notes
<b>Document object</b>		
document.alinkColor	N**	
document.anchors	N	
document.applets	N	
document.bgColor	N**	
document.cookie	N	
document.domain	N	
document.embeds	N	
document.fgColor	N**	
document.forms[]	Y	
document.forms[].length	Y	Number of forms in doc.
document.{form_name}	Y	References form by its name.
document.{form}.action	Y	security: <a href="#">mailto:</a> & <a href="#">news:</a> not allowed.
document.{form}.elements[]	Y	
document.{form}.elements[].length	Y	
document.{form}.{element}.checked	Y	Checkbox, Radio Button
document.{form}.{element}.defaultChecked	I	Checkbox, Radio Button
document.{form}.{element}.defaultValue	Y	Text, Password
document.{form}.{element}.defaultSelected	I	Option
document.{form}.{element}.form	Y	References parent form
document.{form}.{element}.length	Y	Select List or Drop Down
document.{form}.{element}.name	Y	
document.{form}.{element}.options[]	Y*	
document.{form}.{element}.selectedIndex	Y	Select List
document.{form}.{element}.type	Y	

### JavaScript Support

---

Item	Support (Y/N)	Notes
document.{form}.{element}.value	Y	Includes support for all relevant elements. This includes support for Drop Downs and Selects (both single and multi-select).
document.{form}.encoding	N	
document.{form}.method	Y	
document.{form}.name	Y	
document.{form}.target	N	
document.{form}.[elmt_name][] (i.e. document.myform.category[i])	Y	Accesses the <i>ith</i> element having the specified name (for example, <i>category</i> ). Used for Checkbox, Radio Button.
document.images[].border	N	
document.images[].complete	N	
document.images[].height	N	
document.images[].hspace	N	
document.images[].lowsrc	N	
document.images[].name	N	
document.images[].prototype	N	
document.images[].src	N	
document.images[].vspace	N	
document.images[].width	N	
document.lastModified	N	
document.layers	N	
document.linkColor	N**	
document.links[].hash	N	
document.links[].host	N	
document.links[].hostname	N	

*JavaScript Support*

Item	Support (Y/N)	Notes
document.links[].href	N	
document.links[].pathname	N	
document.links[].port	N	
document.links[].protocol	N	
document.links[].search	N	
document.links[].target	N	
document.links[].text	N	
document.plugins	N	
document.referrer	N	
document.title	Y	
document.URL	Y	
document.vLinkColor	N**	
document.captureEvents()	N	
document.close()	N	
document.{form}.{element}.blur()	Y	
document.{form}.{element}.click()	N	
document.{form}.{element}.focus()	Y	
document.{form}.{element}.handleEvent()	N	
document.{form}.{element}.select()	N	
document.{form}.handleEvent()	N	
document.{form}.reset()	Y	
document.{form}.submit()	Y	
document.getSelection()	N	
document.handleEvent()	N	
document.images[].handleEvent()	N	
document.links[].handleEvent()	N	
document.open()	N	
document.releaseEvent()	N	
document.routeEvent()	N	
document.write()	N	
document.writeln()	N	

### JavaScript Support

---

Item	Support (Y/N)	Notes
<b>Browser related objects</b>		
navigator.appCodeName	Y	Returns "Good Intranet"
navigator.appName	Y	Returns "Good Intranet"
navigator.appVersion	Y	Returns IE version. <sup>a</sup>
navigator.language	Y	Returns "en"
navigator.mimeTypes	N	
navigator.platform	Y	Returns handheld type. <sup>b</sup>
navigator.plugins	N	
navigator.userAgent	Y	
navigator.javaEnabled()	Y	Returns <i>false</i> .
navigator.plugins.refresh()	N	
navigator.preference()	I	Disabled for security.
navigator.taintEnabled()	Y	Returns <i>false</i> .
<b>"this" object</b>		
Forms	Y	
Elements	Y	
<b>Window object</b>		
DnEvents	N?	IE only DnEvents.which=13
self	Y	
top	Y	Defaults to main window.
window.closed	Y	
window.defaultStatus	I	Disabled for security.
window.document	Y	
window.frames	Y-	Mapped to 'self'

*JavaScript Support*

Item	Support (Y/N)	Notes
window.history	N	History properties disabled for security.
window.innerHeight	I	
window.innerWidth	I	
window.length	I	
[String] window.location ( <i>accessing window.location as a String</i> )	Y	
window.location.hash	I	
window.location.host	I	
window.location.hostname	I	
window.location.href	Y	
window.location.pathname	I	
window.location.port	I	
window.location.protocol	I	
window.location.search	I	
window.locationbar	I	
window.menubar	I	
window.name	Y-	Read only
window.opener	N	
window.outerHeight	I	
window.outerWidth	I	
window.pageXOffset	I	
window.pageYOffset	I	
window.parent	Y	
window.personalbar	I	
window.scrollbars	I	
window.status	I	Disabled for security
window.statusbar	I	
window.toolbar	I	
window.window	Y	Current window
self.open()	N	

### *JavaScript Support*

---

Item	Support (Y/N)	Notes
top.open()	N	
window.alert()	Y	
window.back()	Y	
window.blur()	I	
window.captureEvents()	I	
window.clearInterval()	I	
window.clearTimeout()	I	
window.close()	Y	
window.confirm()	Y	
window.disableExternalCapture()	I	
window.enableExternalCapture()	I	
window.find()	I	
window.focus()	Y	
window.forward()	Y	
window.handleEvent()	I	
window.history.back()	Y	
window.history.forward()	Y	
window.history.go()	Y-	Only for numerics.
window.home()	I	
window.location.reload()	Y	
window.location.replace()	Y	
window.moveBy()	I	
window.moveTo()	I	
window.open()	N	Currently not implemented. Pop-ups are not supported.
window.print()	I	
window.prompt()	Y	
window.releaseEvents()	I	
window.resizeBy()	I	
window.resizeTo()	I	

### JavaScript Support

---

Item	Support (Y/N)	Notes
window.routeEvent()	I	
window.scroll()	I	
window.scrollBy()	I	
window.scrollTo()	I	
window.setInterval()	Y	Repeating timer.
window.setTimeout()	Y	Timer expiration set to specified value.
window.stop()	I	

*Note:* You cannot set JavaScript events dynamically. For example, you cannot use `document.myForm5.myPasswd.onblur = passonblur;`

If a page contains references to more than ten JavaScript pages, the first ten pages display properly, but the eleventh and subsequent pages do not display. Also, if a bookmark includes JavaScript, the page does not display properly.

## Launching Good Intranet from a Third-Party Application

You can launch Good Intranet client programmatically from either a Palm or Windows Mobile application. When Good Intranet starts, it opens a new window and launches the specified URL. For example, if you have a stock ticker application and it provides a link for users to click to get more news about a company, when the users select the link, Good Intranet will start automatically.

To launch Good Intranet from an application, call this function using the commands provided in the following section.

## Developing Custom Applications

### *Palm applications*

```
SysAppLaunch (cardNo, dbID, 0, SysAppLaunchCmdURL, (void*) "<URL>", &result);
```

### *Windows Mobile applications*

```
GoodAccess.exe -param:url:<URL>
```

### *Considerations*

- If Good Intranet has not been set up on the handheld, the user is prompted to install Good Intranet when the application tries to start Good Intranet.
- If the handheld's radio is not on, the request remains in the queue and Good Intranet will start when the radio is turned on.
- If the handheld has Good Messaging installed but not running, it may take some time for Good Intranet to start.

### *Example*

The following code launches the Good Intranet client and take the user to Google's website:

#### *Palm example*

```
SysAppLaunch (cardNo, dbID, 0, SysAppLaunchCmdURL, (void*) "http://www.google.com", &result);
```

#### *Windows Mobile example:*

```
GoodAccess.exe -param:url:http://www.google.com
```

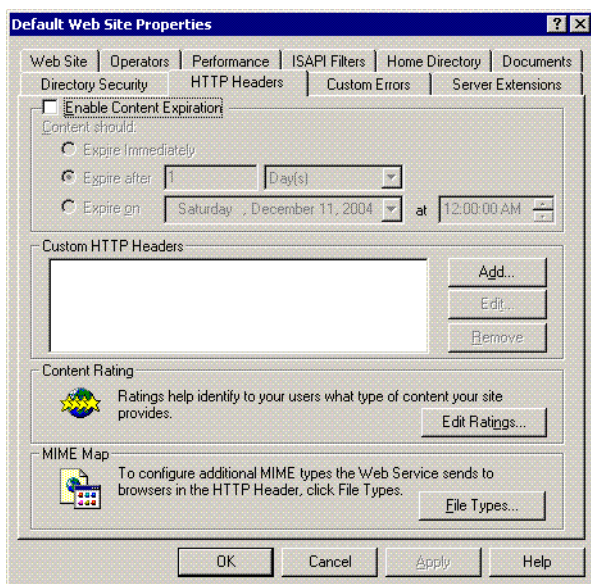
For more information about the *SysAppLaunch* and *SysUIAppSwitch* APIs, see the Palm developer documentation.

## Migrating GoodInfo Files to Good Intranet

You can use GoodInfo (.*gia*) files with Good Intranet without modifying the original files. To use GoodInfo files, you need to place the files in a Web directory and configure an additional MIME type (.*gia*). This file type specification is sent by the Web server to browsers in the HTTP header of the response.

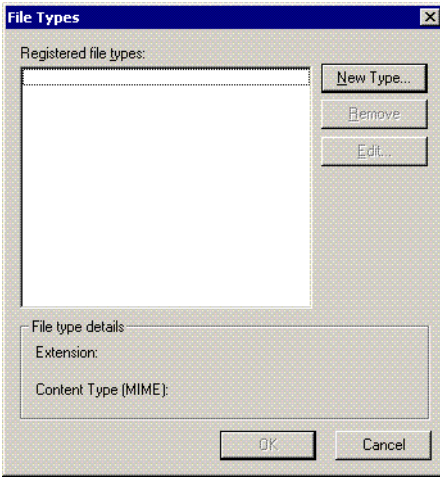
To configure an additional MIME type for GoodInfo files with IIS:

1. From your Windows Server IIS Manager, right-click the Web site and select Properties. The Default Web Properties window appears.
2. Select the HTTP Headers tab.

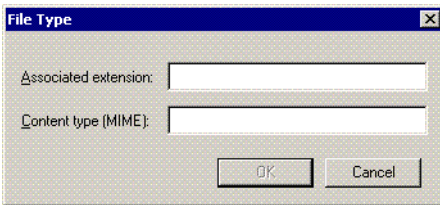


## Developing Custom Applications

3. In the MIME Map section, click File Types.



4. From the File Types window, click New Type.



5. In the Associated Extension field, enter *.gia* or *gia*.
6. In the Content Type field, enter *text/plain*.
7. Click OK.

To configure an additional MIME type for GoodInfo files with an Apache Web Server:

1. Stop your Apache Web Server. A message appears telling you that the server is stopping.
2. Edit the *httpd.conf* file in the *C:\Program Files\Apache Group\Apache2\conf* directory and add the following line to the *AddType* section:  
*AddType text/plain .gia*
3. Save the *httpd.conf* file.
4. Restart to Apache server.

For information about how to install IIS, refer to Windows Help.

## Using GoodInfo 1.0 System Variables

For backward compatibility with GoodInfo 1.0, the following system variables let you submit hidden values from an application to Good Intranet Server. The system variables, which contain information unique to each handheld, are:

- *gi\_emailaddress*—Contains the email address of the handheld's user. You can use this variable to submit the user's email address, so the user does not need to enter it in a separate text field.

For example, you can use the *gi\_emailaddress* variable in an application that allows users to add their address to an email distribution list. The application might contain a menu of distribution lists and a submit button with the label "Add My Name." After the user chooses a list from the menu and then clicks the "Add My Name" button, the application submits the value of the *gi\_emailaddress* variable. (The value might be submitted to a CGI script on the server that manages the distribution lists.)

- *gi\_esn*—Contains the handheld's serial number. You can use this variable to submit the handheld's serial number. For example, you can use this variable to retrieve data on a server based on a handheld's serial number.

## Developing Custom Applications

- The following variables contain the handheld's current time, date, and GMT offset:

*gi\_date\_month* (numeric value between 1 and 12)

*gi\_date\_day* (numeric value between 1 and 31)

*gi\_date\_year* (numeric value starting at 1999) (Set to current year by default.)

*gi\_time\_hour* (numeric value between 0 and 23) (24-hour clock)

*gi\_time\_min* (numeric value between 0 and 59)

*gi\_time\_sec* (numeric value between 0 and 59)

*gi\_time\_gmtoffset* (numeric value representing the number of hours off of GMT) (Can be positive or negative. If negative, the minus sign "-" is not URL-encoded per RFC1738. The same applies to the "." if the value is a fraction.)

For any type of Good Intranet Application, you can use the system variables in an INPUT element with the "hidden" value for the Type attribute. Specify the system variable you want to use in the *\$(var\_name)* syntax for the Value attribute. For example:

```
<input type="hidden" name="email" value="$(gi_emailaddress)" />
```

# Index

---

## A

application development 9, 17  
    customizing tips 32  
    testing and debugging 22  
application properties files 15

## B

bibliography, HTML and XML  
    books 11

## C

case, ignoring 54  
color support 66  
converting MIME types 70

## D

debugging, for application  
    development 23

## G

Good iv  
Good Data Center. *See* Good  
    Operations Center.

## H

HTML books 11  
HTML support 66  
HTML, supported elements 57  
HttpTransformTool command 24

## I

ignoring case, in transformation  
    selectors 54

image support 69

## L

log files  
    debugging applications 22

## M

MIME type conversion 70

## P

properties files  
    for application development 12

## S

style sheet, creating custom 30  
synchronization. *See* wireless  
    synchronization

## T

transformations 47  
    built-in 15, 17  
    custom 21, 26  
    debugging tools 24  
    introduction 9

## W

WML  
    unsupported features 73

## X

XML books 11  
xpath property 48, 52

