

**How-to Guide
SAP NetWeaver '04**



How To Optimize Network Traffic in SAP Enterprise Portal EP 6.0

Version 1.01 – January 2005

Applicable Releases: SAP NetWeaver '04

© Copyright 2005 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

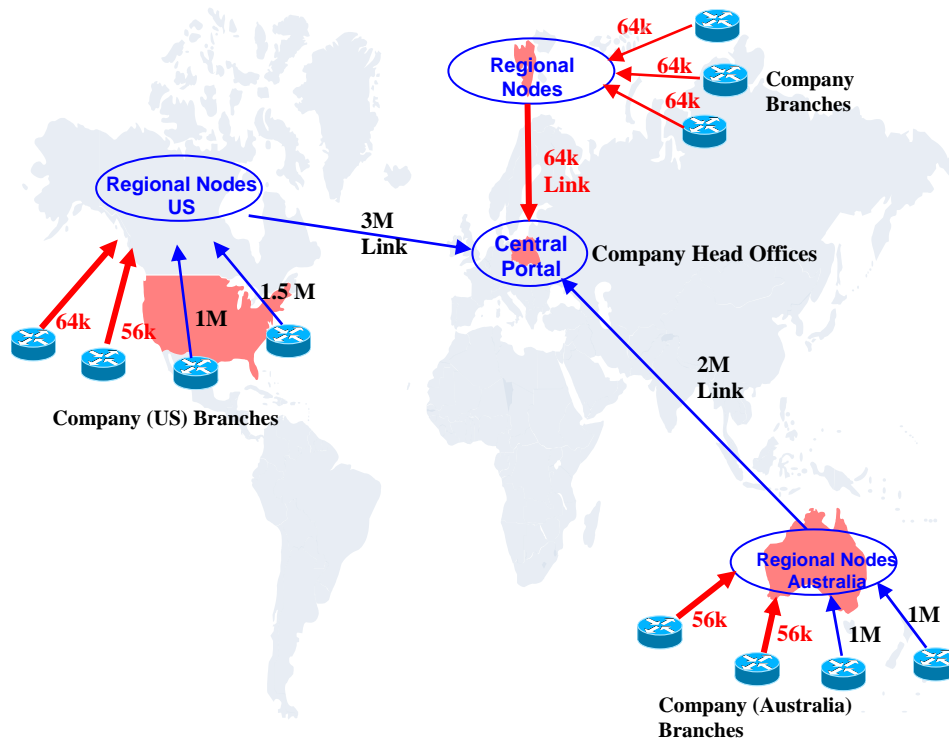
Contents

1	BUSINESS SCENARIO	1
2	INTRODUCTION.....	2
3	PORTAL.....	3
3.1	ISOLATION METHODS	4
3.1.1	<i>Isolation Methods for an iView.....</i>	<i>4</i>
3.1.2	<i>Isolation Methods for a Page.....</i>	<i>6</i>
3.2	PORTAL CACHING MECHANISM	8
3.2.1	<i>iView Caching Properties.....</i>	<i>8</i>
3.2.2	<i>Page Caching Considerations</i>	<i>11</i>
3.2.3	<i>Cache Settings Recommendations</i>	<i>11</i>
3.2.4	<i>Runtime Cache Settings</i>	<i>12</i>
3.3	URL iVIEWS	13
3.4	JAVASCRIPT AND CSS FILE OPTIMIZATION.....	15
3.4.1	<i>Embedded JavaScript</i>	<i>15</i>
3.4.2	<i>Cleaning Up JavaScript and CSS Files</i>	<i>16</i>
3.4.3	<i>Resource Duplication</i>	<i>19</i>
4	J2EE ENGINE.....	20
4.1	HTTP COMPRESSION	20
4.2	CONTENT EXPIRATION	23
5	BROWSER.....	25
5.1	SETTING THE BROWSER FOR COMPRESSION	25
5.2	BROWSER CACHING OPTIONS	26
5.3	TEMPORARY INTERNET FILES OPTIONS.....	27
6	ITS.....	28
6.1	JAVASCRIPT AND CSS FILES OPTIMIZATION	28
6.2	ITS COMPRESSION	28
7	NTLM AUTHENTICATION IMPACT ON NETWORK TRAFFIC.....	29
7.1	COMPRESSION	30
7.2	HTTP ERRORS OPTIMIZATION	31
8	NETWORK ANALYSIS AND TEST SCENARIOS	32
8.1	ANALYSIS TOOLS	32
8.2	OPTIMIZATION TOOLS	32
8.3	TEST SCENARIOS AND EXAMPLES	33
	APPENDIX A: BACKGROUND INFORMATION.....	38
	APPENDIX B: USE CASES.....	40

1 Business Scenario

Many organizations today have distributed systems. Users from different locations (both from inside and outside the country) may try to connect to a central system (see the picture below). Bandwidth and latency between head offices and branches generally differ from site to site, so users connecting through local area networks (LAN) and wide area networks (WAN) can experience serious performance problems caused by heavy network traffic.

And while the number of users connecting to the portal generally increases over time, the infrastructure to handle this growing traffic generally does not. Therefore, it is important to fine-tune the system to take advantage of configuration options that increase portal performance.



2 Introduction

This document describes how to enhance performance of SAP Enterprise Portal 6.0 by fine-tuning the design-time configuration of the portal, the J2EE engine and other related components. Such configuration is designed to speed network performance by reducing the following:

- **Network Traffic:** The volume of data traveling between client and server.
- **Round-Trip Time:** The time required for a network communication to travel from the source to the destination and back.
- **Number of Round-Trips:** The number of times a specific application sends and receives packets of data over the network in a specific period of time.

The recommendations in this document have the added advantage of reducing the load on both the Web application server and the network.

To achieve the goals listed above, portal pages should attempt to do the following:

- Reduce the amount of data exchanged over the network, for example, by using compression or eliminating unnecessary data.
- Cache content that can be reused by the browser and the portal server, instead of retrieving it again from the source.
- Reduce the number of server requests during navigation, for example, by getting information for only those iViews on a page that need to be updated.

This guide deals with both NW '04 and previous releases (for example, EP 6.0 SP2 and J2EE 6.20).

This document does not describe optimizations that are specific to WAN environments.

3 Portal

Before trying to enhance performance by fine-tuning the portal configuration, you should be aware of those events and settings that may affect performance.

Portal performance is significantly affected by navigation within the portal, including round-trip time, number of round-trips, and the size of portal content passing through the network. The initial round-trip – from the client to the server and back to the client – loads the portal framework page and the Welcome page. After that, additional round-trips can be initiated by user action.

There are two types of user action: Navigation actions and Content actions.

Navigation Actions

Navigation actions cause the reloading of the whole Desktop Innerpage, including the navigation panel and content area. The content area includes pages/iViews and their business content.

Content Actions

Content actions are performed within a page/iView and cause data to load, for example, when a user submits a form or selects a link within an iView. Such actions are not meant to trigger navigation; rather, they provide functionality to the user.

The following page/iView properties can affect runtime performance:

Isolation Method

Isolation method refers to how content is gathered for display. There are three isolation methods for pages/iViews: Embedded, URL isolation and pumped. For more information, see [“Isolation Methods”](#).

Cache Settings

Caching stores the most recently accessed information about a page/iView and its content, either in the client's browser or on the server. Afterwards, whenever the same page/iView is launched, the portal first checks the cache for the corresponding content. If the cache has not expired, its content is displayed. Caching improves portal performance and reduces round-trip time. For more information on portal caching, see [“Portal Caching Mechanism”](#).



Do not cache the Desktop Innerpage because each navigation action changes the displayed content in the navigation panel and content area anyway, and caching will have no value.

For more information on isolation methods and cache setting, see help.sap.com → *Documentation* → *SAP NetWeaver* and open the NetWeaver documentation. In the documentation, see *People Integration* → *Portal* → *Administration Guide* → *Content Administration* → *Portal Pages* → *Portal Page at Runtime*.

URL iViews

Data in URL iViews can be fetched in two modes: server side and client side. One should decide whether to use server side or client side fetching based on how heavy the information is (R/3 information, BW iViews), since this may impact performance.

3.1 Isolation Methods

The isolation method can be defined for both a page and an iView. The isolation method determines how the content is retrieved and how it is rendered or incorporated in the surrounding page or framework.

3.1.1 Isolation Methods for an iView

The following describes how the isolation method affects performance of an iViews:

- **URL Isolation** (similar to isolation level 3 in EP5.0): An iView with URL isolation is loaded into an iFrame on the page. The portal first loads the page, and then loads the iView.

This action involves an additional round-trip, and has a performance cost when the page loads. On the other hand, user actions in the iView can trigger content to reload within that iView only, rather than reloading the entire page. Content in other iViews will not reload (so long as there is no client-side eventing between the iViews to trigger reloading).

On pages that are not set to be cached, URL isolation enabled iViews can be cached separately.

- **Embedded Isolation** (new in EP 6.0): An iView with embedded isolation is loaded together with the page. Technically, the iView and its content are part of the HTML table for rendering the page. Therefore, the content of the iView, the iView itself and the page are delivered in the same round-trip. However, user actions in an iView with embedded isolation causes the entire page to reload from the server and, thus, causing all iViews on the page to reload regardless of their isolation method. Another disadvantage is that the browser cannot cache the iView separately; the iView is cached only if the entire page is cached.
- **Pumping Isolation** (isolation level 2 in EP5.0): For an iView with pumped isolation, the content is loaded in the same round-trip as the page, and the iView is located in an iFrame.

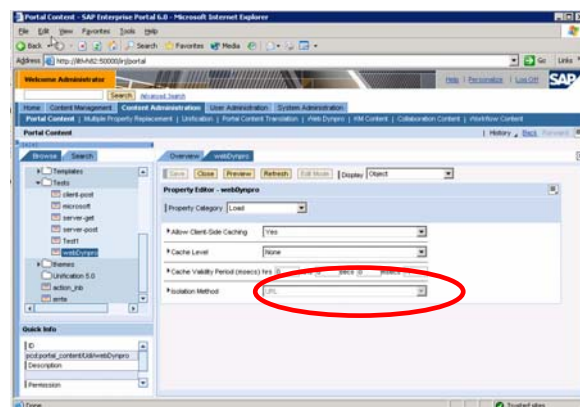
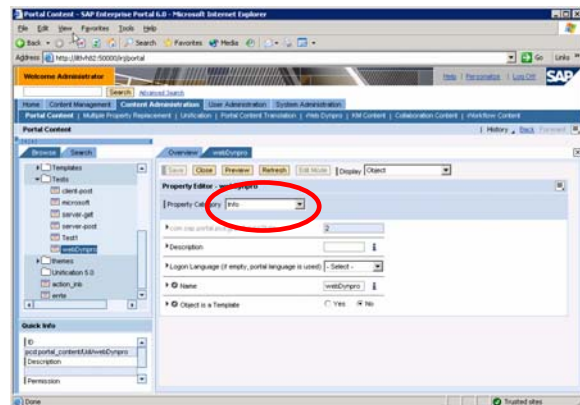
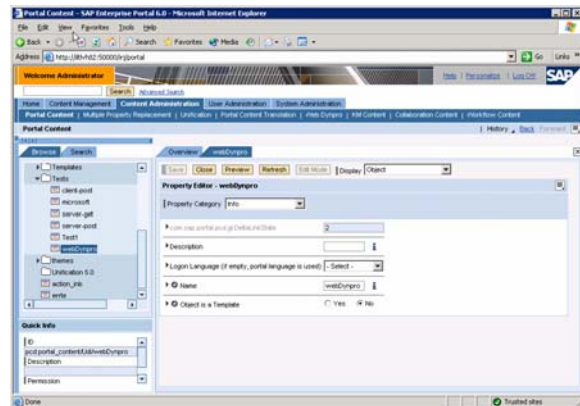


Field experience has proven that the pumped isolation method has limitations, such as potential scripting problems and slow client rendering. For this reason, pumped isolation is not recommended. This feature is deprecated in NW '04. The iView method *Pumped* is no longer supported, although it still appears (as *Pumped-Deprecated*) in the *Isolation Method* iView property menu.

URL iViews always use URL isolation and, therefore, are always displayed within an iFrame in a portal page.

To assign an isolation method to an iView:

1. Open the Property Editor for an iView.
2. From the *Property Category* menu, select *Load*.
3. Select a value for the *Isolation Method* property.



3.1.2 Isolation Methods for a Page

The isolation method of a page determines how the page is included in the content area of the Desktop Innerpage. The effect of the isolation method of a page is similar to that of an iView.

The following describes how the isolation method affects performance of an iViews:

- **URL Isolation for a Page:** A page with URL isolation is contained in an iFrame within the content area. When you navigate to the page, the Desktop Innerpage loads first, and then an additional round-trip is performed to obtain and load the page. On the other hand, content actions do not cause the whole Desktop Innerpage (including the navigation iViews in the navigation panel) to reload; only the page is reloaded. The page can be cached in the browser.

Expect improvement in portal performance when you put iViews with embedded isolation in a page with URL isolation, as the iViews, their content and the page are loaded together in a single round-trip.

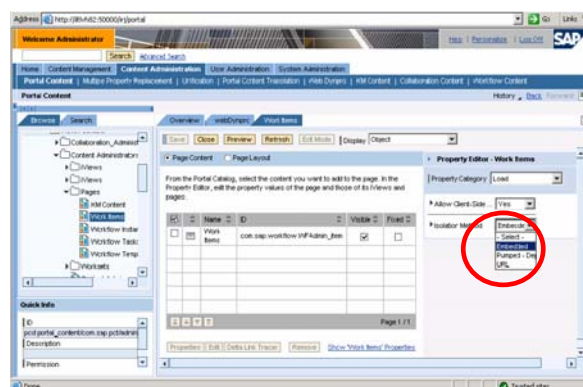
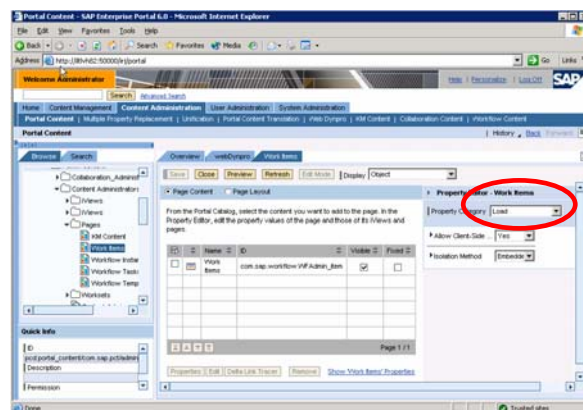
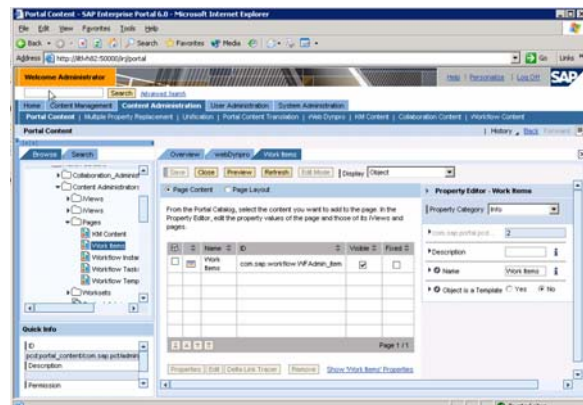
- **Embedded Isolation for a Page:** A page with embedded isolation is included as part of the Desktop Innerpage. Portal performance is improved because each display of the page causes the page to load together with the Desktop Innerpage in a single round-trip. On the other hand, content actions cause the whole Desktop Innerpage to reload, rather than loading only the page. If iViews with URL isolation are placed in this page, any content action in an iView reloads only the content in that iView.



You cannot implement browser caching in the client for a page with embedded isolation. This is due to a technical restriction imposed by the request method used for the Desktop Innerpage (which contains the page).

To assign an isolation method to the page:

1. Open the Property Editor for a page.
2. From the *Property Category* menu, select *Load*.
3. Select a value for the *Isolation Method* property.



General Guidelines

The following are guidelines on when to use which isolation method:

- Either URL or embedded isolation is suitable for iViews with relatively heavy content.
- Use embedded isolation mode for:
 - iViews that are tightly coupled (with POMevents) and need to refresh themselves, through the server, according to mutual client actions.
 - iViews that are presented as navigation nodes outside a page.
- Use URL isolation mode when presenting content external to the portal server, such as in URL iViews. The user will likely navigate within the iView, and it is unnecessary to reload the entire page for this navigation.

After selecting an isolation mode, check the iView functionality in its full portal environment.

There are user interface considerations, as well, for picking the isolation method. For example, iViews with URL isolation have scroll bars when there is too much information to display in the space allotted the iView, while iViews with embedded isolation do not have scroll bars.

3.2 Portal Caching Mechanism

The portal has a caching service that supports content caching for a page/iView on the portal server. In addition, it can implement content caching on the client machine using the browser's caching capabilities. Configure caching properties for pages/iViews at design time. Depending on your configuration, the portal can cache content as follows:

- **On the portal server**

Server-side caching can improve performance in two ways. It reduces processing time of the server (and of other resources) for regenerating the content when the iView is called on subsequent requests (that is, the server does not need to build the content but only needs to fetch it from the cache). It also reduces network traffic to backend systems and other servers that communicate with the portal server.

Information that an iView retrieves from a data source is stored, for a preconfigured amount of time. Whenever the same iView is launched, the server first checks the cache instead of going to the data source. If the cache has not expired, the iView displays the information contained there.

- **On the client browser**

Using the portal caching capabilities on the client minimizes network traffic and displays content faster, as the content is taken from the browser's cache. The browser uses an HTTP conditional request to check whether content in its cache is still valid. If the content in the cache has expired, it is discarded and replaced with content retrieved from the server.

For more information on cache settings, see help.sap.com → *Documentation* → *SAP NetWeaver* and open the NetWeaver documentation. In the documentation, see *People Integration* → *Portal* → *Administration Guide* → *Content Administration* → *Portal Pages* → *Portal Page at Runtime*.

3.2.1 iView Caching Properties

The cache-related properties of iViews determine whether the iView retrieves data from its data source (that is, the iView's `doContent()` method is executed) or from a cached store. Data coming from a cached store decreases the round-trip time for a iView, thus improving the performance and network traffic of a portal page at runtime.

You can activate the following cache settings for iViews:

- **Cache Level**

This property determines whether content is cached or not. There are four cache levels:

None (default)	Content is never cached. Each time the page/iView is launched, the iView is executed to regenerate its content. This level is appropriate for iViews retrieving information that is constantly changing and being updated, such as a news bulletin iView or a ticker to a stock exchange.
Session	Content is cached for the current session. Each new logon empties the cache and the page/iView must be executed.
User	Content is cached per user. New users who launch the page/iView receive content generated by a new instance of the iView. An iView representing a user's daily tasks is appropriate for this cache level.
Shared	Content is cached for all users. This level is suitable for iViews displaying information relevant to all users, such as company announcements.

- **Cache Validity Period**

This property defines the expiration time, in milliseconds, for the iView content stored in the cache. When the cache expires, the iView updates the data in the cache with regenerated content (from the data source).

You can set the cache validity period to -1, which is compatible with all the cache levels except None. It ensures that caching is active at all times. This can be useful for an iView whose content never changes, such as a welcome iView, which does not require updating.



Refreshing an iView from the options menu forces it to retrieve updated content from the source, even if the cache is still valid.

iViews set to Shared cache are not affected since all users receive identical content that cannot be individually updated.

Cache validity is respected, regardless of any particular cache level, except None. For example, if the cache level is Session, and the cache validity is two hours, the iView content will be retrieved from the source the first time a user logs on, and then will come from cache for the next two hours only, even if the session lasts four hours.

Synchronize cache validity and cache level as logically as possible.

- **Allow Client-Side Caching**

The property defines whether the content of a iView is cached in the client's browser, portal's cache or both.

It takes the following values:

True (default): iView/page content is stored in the client's browser cache and portal cache.

False: iView/page content is stored in the portal cache and not in the client's browser cache.

Only browser: iView/page content is stored in the browser's cache and not in the portal cache.

This property is useful in cases where you want to preserve memory resources on the server, and use the user's browser cache; for example, an iView with relatively large content, fast server execution and a cache level set to User.



This property will be available for iViews and pages as of EP 6.0 SP2 Patch 5 and above. Currently, you can activate client-side caching using the global setting for the browser as discussed in the section below.

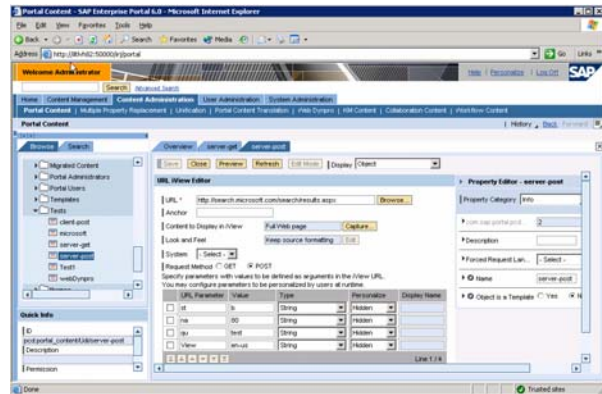
The global client-side caching property provides the portal with extensive control over browser-cache usage by iViews and pages. The default value for this property is **True**, enabling the portal to implement the browser cache for each page/iView that uses the server cache (starting from EP 6.0 SP2 P5, each iView/page has a property that enables client-side caching). If you select **False**, the service is disabled, and the portal does not implement the browser cache at all.

Note that this setting refers only to the generated content for iViews, and not to the static MIMEs (Multi-purpose Internet Mail Extension), such as cascading style sheet (CSS) files, JavaScript (JS) files, and images whose cache is controlled by the J2EE settings.

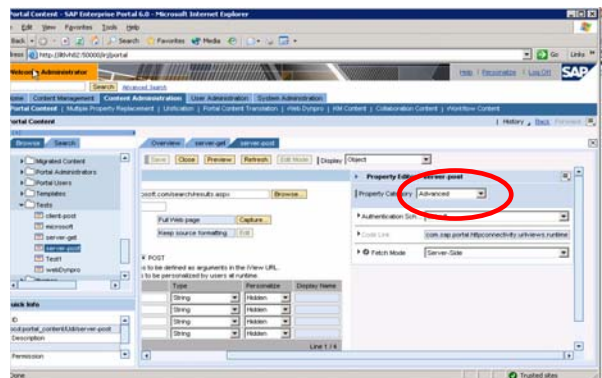
For more information on cache settings, see help.sap.com → *Documentation* → *SAP NetWeaver* and open the NetWeaver documentation. In the documentation, see *People Integration* → *Portal* → *Administration Guide* → *System Administration* → *Service Configuration* → *Administering Application Caching*.

To change iView caching properties (the steps are similar in both EP60 SP2 and NW '04 version but the GUI differs):

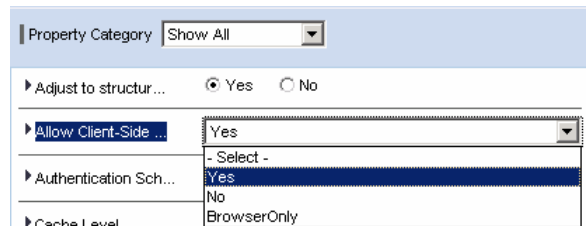
1. Open the Property Editor for an iView.



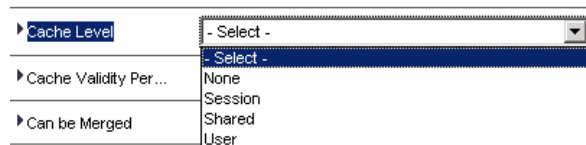
2. From the *Property Category* menu, select *Show All*.



3. Select a value for the *Allow Client Side Caching* property.



4. Select a value for the *Cache Level* property.



5. Select a value for the *Cache Validity Period* property.



3.2.2 Page Caching Considerations

Important considerations when setting page cache:

- Pages with embedded isolation level must not be cached (cache level = none), as some vital code in the `doContent()` phase of the page building process must be executed if a page is embedded in another page.
- Storing a page in the cache is most effective when there are iViews with embedded isolation in the page, and the browser's cache can be used to store the page and its iViews.
- iViews with URL isolation will not be affected when the page is cached, as they are retrieved in an additional request to that of the page.
- iViews with embedded isolation and no caching (cache level = none) behave according to the page settings (i.e., if the page is stored in the cache, the iViews and their content will be stored for the same period of time and scope).
- Caching level and validity period settings for a page should be set such that they can accommodate the settings of these properties for the iViews on the page.

The settings should fit a wide range of cache levels and validity periods for iViews on the page, for example, shared cache level and long validity period. An iView with shorter validity period and narrower cache level, such as User or Session, will have its settings executed for that iView rather than that of the page.

Note that in the server cache, the portal has the ability to combine new content from an iView with embedded isolation with the cached content of another iView with embedded isolation, as long as both iViews are on the same page.

On the client, if the content of an iView with embedded isolation has to be retrieved from the server, the whole page will be retrieved from the server.

3.2.3 Cache Settings Recommendations

Using the portal cache service enhances performance. The following are guidelines for choosing suitable cache settings:

- **Cache Level and Validity Period**

The type of content determines the cache level and validity period settings that are suitable to apply. A cache level with broader scope and longer validity period results in better performance.



Where the type of content allows you to choose between Shared and User cache levels, apply Shared cache level.

Where the type of content allows you to choose between User and Session cache levels, apply User cache.

- **Client-Side Caching**

Use the client's browser caching capabilities whenever possible.



Some content should not be cached in the browser due to security or privacy considerations.

Caching content on the client's browser offers better performance improvement, as the cached content is instantly presented to the user. As long as the content in the browser's cache is valid (based on portal server's calculation), a Not-Modified HTTP304 response will be returned to the browser; otherwise new content is retrieved.

Client Cache Limitations:

For security and privacy reasons, do not cache content with confidential information on client machines.

The portal uses the POST request method for navigation actions, but the HTTP POST request method cannot use the browser cache.

For this reason, a content page/iView with embedded isolation (where the iView and its content are retrieved in the same round-trip as that of the Desktop Innerpage) cannot utilize the client's browser cache.

- **Suitable Server-Side Cache Settings**

Storing the content of iViews in the portal server cache uses system resources (mainly the memory). These resources should be used judiciously.

In addition, the cache has a limited size, so it employs an algorithm that discards and removes old content when new content arrives. This can lead to a situation where all the content is put into the cache, but removed before it is re-used.

For this reason, implementing an efficient cache schema should take into consideration content usage-patterns and hit-rate.

- Set the server's cache memory allocation limit according to the portal cache usage and available server memory. For information on the global property setting, refer to [Runtime Cache Settings](#).
- Preferably, use the server's cache for iViews with Shared cached level, so that they can be available to many users.
- Where iView content loads fast (that is, it displays static content stored in the file system), avoid storing the content in the server's cache in order to preserve the resources of the server.

3.2.4 Runtime Cache Settings

You can configure several global properties to control the portal cache. These properties are in the file *pvtDefault.properties*, located in the portal installation folder

.../j2ee/j2ee_<instance_number>/cluster/server/services/servlet_jsp/work/jspTemp/irj/root/WEB-INF/portal/system/properties.

caching.maxsize=5000	The default maximum size of the cache in KB before it is removed.
enable.http.conditional.request=true	Specifies whether the cache supports client-side (browser) content caching.
caching.cluster.notification=false	Enables the cluster notification.
prt.caching.persistence=true	Specifies whether to keep cached content in the database for all applications using the cache layer.
prt.pc.caching.persistence=true	Specifies whether to also keep cached portal components in the database. This is currently disabled.

Other cache properties can be configured via the portal.runtime service configuration.

To set the cache for the runtime application:

- Open the portal and log on with your administration user.
- From the Top-Level Navigation, choose *System Administration* → *System Configuration*.
- From the *Detailed Navigation*, choose *Service Configuration*.
- In the *Content Catalog*, expand *Applications* → *portal.runtime* → *Settings* → *config*.
- Right-click *config* and choose *Edit* to open the *Service Configuration Editor*.

There are several properties for this application, including the following cache settings:

caching.off=false	Specifies whether PRT caching is enabled
caching.persistence.clean.period=120	Specifies the period for cleaning the cache, in database units

3.3 URL iViews

A URL iView displays data directly from a Web-based content source. The isolation method for URL iViews is always URL and cannot be changed. URL iViews, therefore, are always displayed within an iFrame in a portal page.

The retrieval of data for a URL iView is determined by the value of the iView's property in the component *com.sap.portal.reserved.iView.Redirect* (EP 6.0 SP2) or *Fetch Mode* (in NW '04), which can be either:

- **True** in EP 6.0 SP2
Client-Side in NW '04

The client's browser retrieves the content directly from the source.

- **False** in EP 6.0 SP2
Server-Side in NW '04

The client's browser retrieves the content via the portal server (the browser request is directed to the portal server, which retrieves the content and returns it to the browser). Take advantage of the following when implementing server-side content fetching:

- Portal (server) cache service
- POST request method (which is only available with server-side fetching)



Server-side content fetching may be slower because it involves further processing between the content source and the browser. Use server-side fetching when content caching on the server is really worth it; for instance, shared cache from remote/slow content source.

With both client-side and server-side content fetching, subsequent requests (because of user actions on the displayed Web page) are fetched by the client's browser directly from the source of the content, and not through the portal server.

1. Open the iView Property Editor of the URL iView.

-
- The screenshot shows the SAP NetWeaver Administrator interface. The left pane displays a tree structure with 'outportal_content' selected under 'outportal'. The main area shows the 'Property Editor' for the 'outportal_content' extension. The 'User-Role' property is highlighted with a red circle, and its value is set to 'Select'. Other properties like 'Authentication Sub...' and 'User-Role' are also visible.

3.4 JavaScript and CSS File Optimization

There are two types of JavaScript and CSS (Cascading Style Sheets) optimization scenarios. The first one is for when JS and CSS code is embedded in the HTML files loaded by portal. The second is for when JS and CSS files are contained in separate resource files and loaded by the portal.

JavaScript and CSS files usually contain many comments and much white space in order to make it easier to read for developers. End users don't need the comments or white space, so it is best to remove them and reduce file size by half or even more, resulting in faster downloads to the clients.

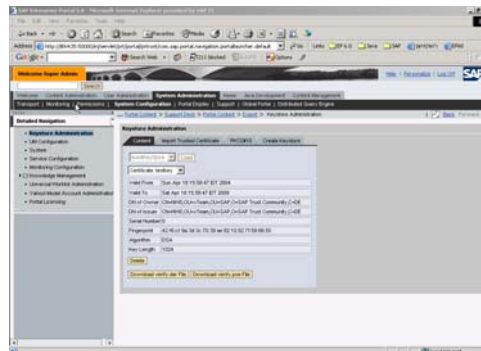
3.4.1 Embedded JavaScript

The client framework includes JavaScript to enable user actions in an HTML page. By default, the client framework embeds JavaScript in HTML files with comments and in an easy-to-read format. You can change the portal settings so that JavaScript comments and unnecessary spaces are eliminated, reducing the size of the page by about 30 percent.

For more information on how to configure the client framework, see help.sap.com/nw04 → *SAP NetWeaver* → *People Integration* → *Portal* → *Administration Guide* → *Content Administration* → *iViews* → *Client Framework And administrators* → *Configuration*.

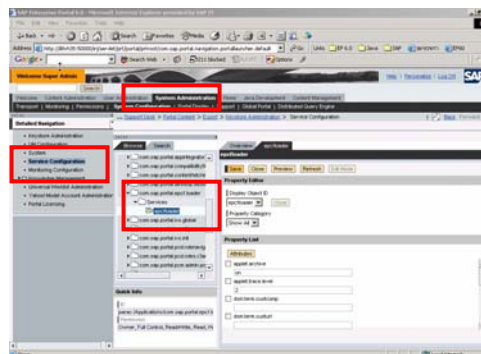
To optimize performance and reduce the size of JavaScript files:

1. Go to *System Administration* → *System Configuration* → *Service Configuration*.

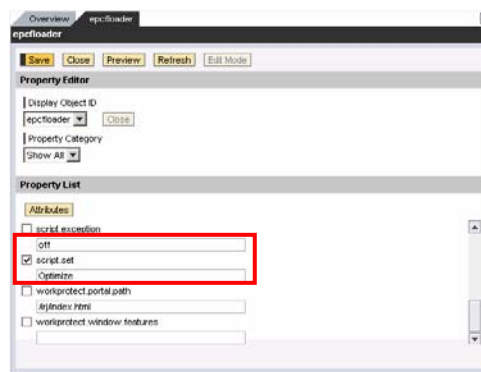


2. In the Portal Catalog, go to *Applications* → *com.sap.portal.epcf.loader* → *Services* → *epcfloader*.

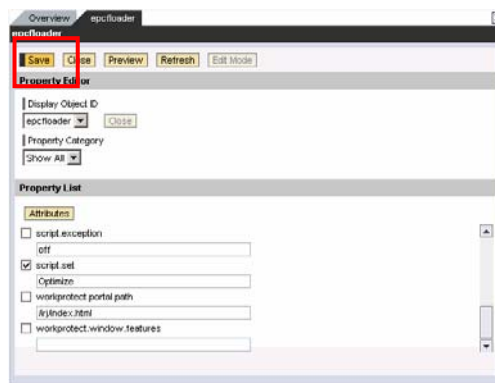
Right-click *epcfloader*, and select *Edit*.



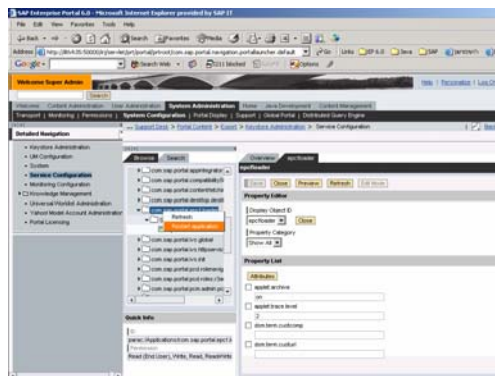
3. Change the *script.set* property to "Optimize".



4. Save your changes



5. Restart the application. Right-click *com.sap.portal.epcf.loader* and choose "Restart Application".



3.4.2 Cleaning Up JavaScript and CSS Files

You can filter JavaScript files with the [JSMIN](#) tool. It is a free utility to compress/scramble JavaScript files. You can compress CSS files with a tool available at <http://flumpcakes.co.uk/css/optimiser>.



Be sure to keep a copy of your original source files. Filtering JavaScript and CSS files is a one-way trip – once done, it cannot be undone. Optimized file can be overwritten by patches/upgrades to the Portal, so it is necessary to check the status of changed files after each portal upgrade.

How-To clean-up JS/CSS files:

There are two steps in cleaning up JS and CSS files. The first step is to find the large JS/CSS files. The second step is cleaning them.

The following steps will guide you on how to find and clean JS and CSS files :

To compress JavaScript and CSS files:

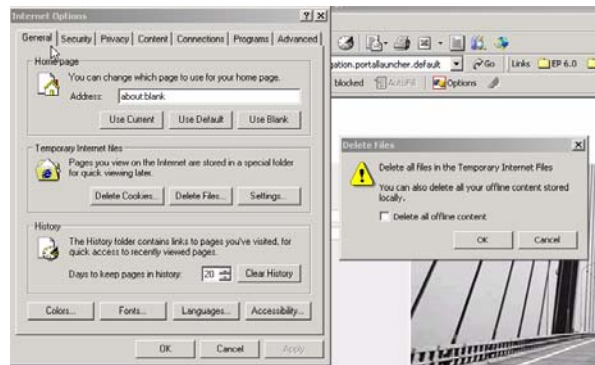
1. Clear the browser cache:

On the Internet Explorer menu, go to *Tools* → *Internet Options*.

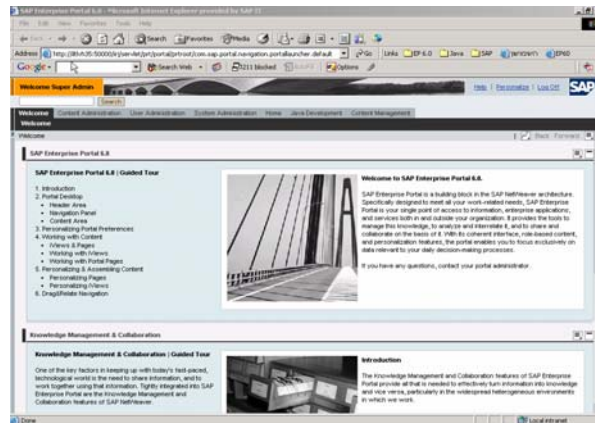
Click *Delete Files*.

Check *Delete all offline files*.

Click *OK*.



2. Log in to the portal.



3. Analyze the files saved in the browser cache:

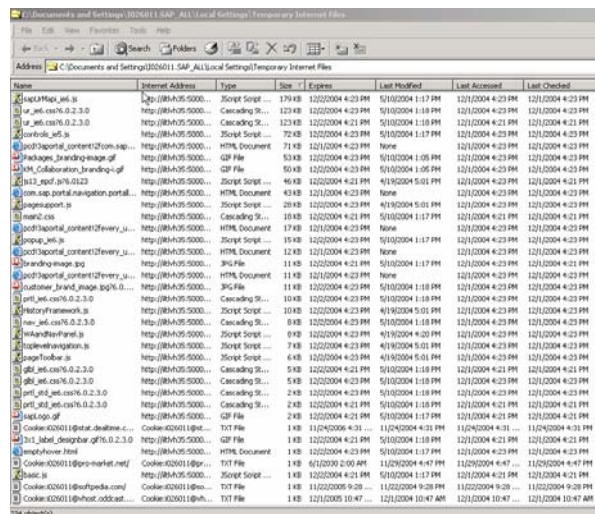
On the Internet Explorer menu, go to *Tools* → *Internet Options*.

Click *Settings*, and then click *View Files*.

Order the files in the cache by size.

In the screen, you can review, among other details, the content expiration, size of the data received for the Welcome page and sources of the files.

Most of the possible problems can be discovered already in this step.



Generally, ignore small files because compression will effectively reduce the time to download them. For example, for a 40K file that is optimized to 25K, you will see only about a 2K network traffic improvement when SAP J2EE HTTP Compression is enabled. Nevertheless, in some cases it can be also useful to pay attention to these files (especially when you have more than five files).

Choose all the files fetched to the browser over 50K. The file locations can be found by looking at their URLs:

EP60 SP2:

Name	Internet Address	Type	Size	Expires	Last Modified	Last Accessed	Last Checked
sapUrMapi_ie6.js	http://iltlvh35:5000...	JScript Script ...	179 KB	12/2/2004 4:23 PM	5/10/2004 1:17 PM	12/1/2004 4:23 PM	12/1/2004 4:23 PM
ur_ie6.css?6.0.2.3.0	http://iltlvh35:5000...	Cascading St...	123 KB	12/2/2004 4:23 PM	5/10/2004 1:18 PM	12/1/2004 4:23 PM	12/1/2004 4:23 PM
ur_ie6.css?6.0.2.3.0	http://iltlvh35:5000...	Cascading St...	123 KB	12/2/2004 4:21 PM	5/10/2004 1:18 PM	12/1/2004 4:21 PM	12/1/2004 4:21 PM
controls_ie5.js	http://iltlvh35:5000...	JScript Script ...	72 KB	12/2/2004 4:23 PM	5/10/2004 1:17 PM	12/1/2004 4:23 PM	12/1/2004 4:23 PM
pcdl3aportal_content!2fcom.sap...	http://iltlvh35:5000...	HTML Document	71 KB	12/1/2004 4:23 PM	None	12/1/2004 4:23 PM	12/1/2004 4:23 PM
Packages_branding-image.gif	http://iltlvh35:5000...	GIF File	53 KB	12/2/2004 4:23 PM	5/10/2004 1:05 PM	12/1/2004 4:23 PM	12/1/2004 4:23 PM

NW '04:

Name	Internet Address	Type	Size	Expires	Last Modified
sapUrMapi_ie6.js	http://iltlvh82:50000/irj/portala...	JScript Script ...	290 KB	12/3/2004 2:11 PM	11/30/2004 2:28 ...
ur_ie6.css?6.0.9.0.0	http://iltlvh82:50000/irj/portala...	Cascading St...	162 KB	12/3/2004 2:11 PM	11/30/2004 2:15 ...
controls_ie5.js	http://iltlvh82:50000/irj/portala...	JScript Script ...	124 KB	12/3/2004 2:11 PM	11/30/2004 2:28 ...
dynamicTree.js	http://iltlvh82:50000/irj/portala...	JScript Script ...	67 KB	12/3/2004 2:12 PM	11/30/2004 2:15 ...
pcdl3aportal_content!2fevery_user!2f...	http://iltlvh82:50000/irj/servlet/...	HTML Document	63 KB	12/2/2004 2:12 PM	None

- For each file, go to the file location and backup the original source file.

Examples:

NW '04

If the URL of the file saved in the cache is:

http://iltlvh82:50000/irj/portalapps/com.sap.portal.htmlb/jslib/sapUrMapi_ie6.js

you can find it in the following location:

```
<Installation Drive>\usr\sap<System Name>\
<Instance Name>\JC00\j2ee\cluster\server<Number>\
apps\sap.com\irj\servlet_jsp\irj\root\portalapps\
com.sap.portal.htmlb\jslib\jslib\sapUrMapi_ie6.js
```

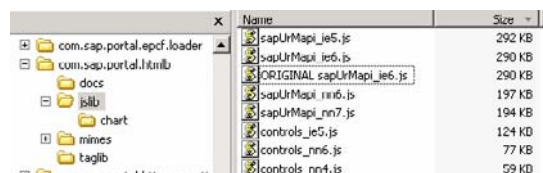
SP2

If the URL of the file saved in the cache is:

http://iltlvh82:50000/irj/portalapps/com.sap.portal.htmlb/jslib/sapUrMapi_ie6.js

you can find it in the following location:

```
<Installation Drive>\usr\sap<System Name>\j2ee\
j2ee_<Instance Number>\cluster\server\services\
servlet_jsp\work\jspTemp\irj\root\portalapps\
com.sap.portal.htmlb\jslib\sapUrMapi_ie6.js
```



Name	Size
sapUrMapi_ie5.js	292 KB
sapUrMapi_ie6.js	290 KB
ORIGINAL sapUrMapi_ie6.js	290 KB
sapUrMapi_nn6.js	197 KB
sapUrMapi_nn7.js	194 KB
controls_ie5.js	124 KB
controls_nn6.js	77 KB
controls_nn4.js	59 KB

5. Process JavaScript files.

At a command prompt, type

```
jsmin <Original.js>New.js
```

```
C:\Nag\nag\J2E\WEB-INF\classes\server\apps\nag.com\ir\servlet_jsp\ir\root\portal\apps\com.nag.portal.html\js\jsmin\jsaplr\aplr_in6.js>temp.js
```

6. Process CSS files.

Go to <http://flumpcakes.co.uk/css/optimiser>.

Copy the CSS content and paste it into the *Style* section.

Click *Go*.

Copy and paste the optimized CSS script back into the original CSS file.



3.4.3 Resource Duplication

Multiple downloads of the same resource can occur in the following cases:

- A resource is referenced from different files but the name of the resource is spelled with different sets of upper- and lower-case letters.
- Two copies of a resource are stored in different folders, and different files reference each one.
- Different frames inside an iView use the same resources (IE bug)

It is important to verify that only one type of browser control file (IE5 or IE6) is downloaded to the client. You can identify them in most cases by their name, for example, controls_IE5 or controls_IE6, or by their location, for example, in folder IE5 or IE6.

4 J2EE Engine

4.1 HTTP Compression

The J2EE Engine contains an HTTP compression feature that enables the compression of files downloaded to the client, thereby reducing network traffic and improving performance.

The use of HTTP compression in conjunction with caching can greatly speed network traffic by dramatically reducing the number of packets transmitted. The number of packets transmitted can be reduced by at least 50 percent, and sometimes by as much as 90 percent. Elapsed time improvement is less dramatic, and strongly depends on your network connection.

There are several J2EE Engine properties that effect network traffic and caching. Optimizing these properties may result in improved performance:

MinimumGZipLength: Specifies the minimum length, in bytes, of a dynamic HTTP response (generated by a servlet or JSP). The default is 8192. This value should be changed to 1024.

The response message is compressed with gzip encoding only if the content length of the response message exceeds the value of the MinimumGZipLength property. Compressing and decompressing short messages can actual slow transmission. Therefore, the Servlet_jsp Service allows the setting of the MinimumGZipLength property. It is recommended to set the value to at least 1024 because the client browser may consume many more resources to decompress the compressed content.

NeverCompressed: Specifies a comma-separated list of servlet response content types. Servlet (or JSP) responses with any of those content types are never compressed. To allow better control over which types of responses are compressed, the HTTP service provides the NeverCompressed and AlwaysCompressed properties. If you want to prevent responses with a specific content type to be compressed, you must add this content type to the list defined by the NeverCompressed property.

When you add a file extension to the list, it must start with an asterisk, for example as **.html*. All other strings are considered to be MIME types (Content-Type header values).

EnableChunkedResponse: Determines if the response to a request is sent as multiple chunks of data. If the response message is bigger than the buffer or the client has flushed the buffers, a persistent connection is maintained by using the [HTTP 1.1](#) chunked encoding. You can enable response chunking by setting the EnableChunkedResponse property of the Servlet_jsp service to true. When chunking is enabled, the container appends the corresponding chunk header to each part of the response message that is being flushed out of the response buffer. Each part is then sent to the client. The connection is not closed and the client keeps waiting for the remaining parts of the response message to be sent to it (according to chunked encoding).

Please refer to SAP Note 746666 if the following symptoms occur after enabling compression:

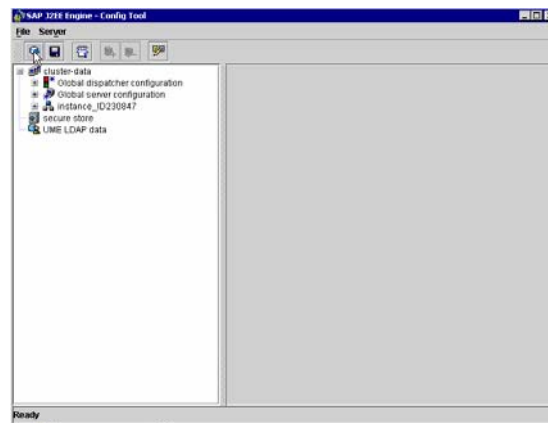
- OutOfMemoryError occurs when the system is under load.
- System response times are unstable under constant load.
- Performance degrades over time.

To optimize HTTP compression:

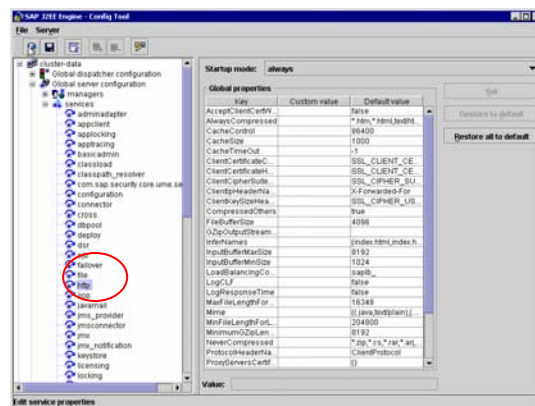
1. Run the J2EE configuration tool.

For J2EE 6.40, run the `/usr/sap/<system name>/j2ee/instance number>/j2ee/configtool/configtool.bat`.

J2EE 6.20, run `/usr/sap/<system name>/j2ee/j2ee_<system number>/configtool/configtool.bat`.



2. Open the HTTP service settings.



3. Change the values of the properties as follows:

J2EE 6.40

MinimumGZipLength = 1024

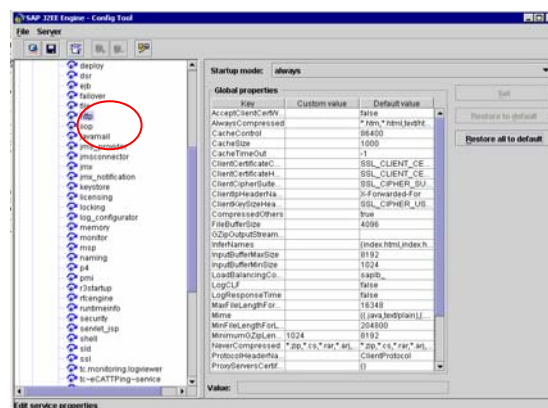
NeverCompressed - To enable script files compression, remove the values marked in bold from this line: `NeverCompressed = *.zip,*.cs,*.rar,*.arj,*.z,*.gz,*.tar,*.lzh,*.cab,*.hqx,*.ace,*.jar,*.ear,*.war,*.css,*.pdf,*.js,*.g zip,*.uue,*.bz2,*.iso,*.sda,*.sar,image,application/x-compressed,application/zip,application/x-gzip,application/pdf,content/unknown, text/JavaScript,[unknown]`

J2EE 6.20

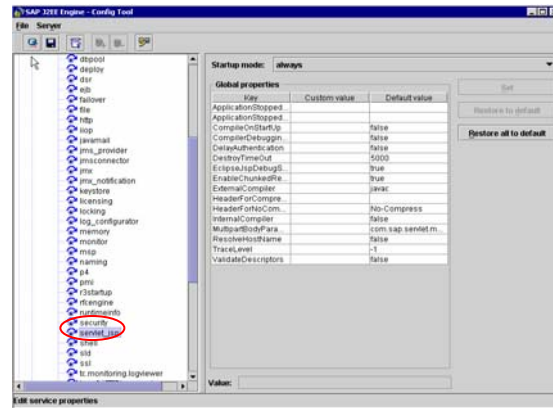
EnableZippedResponse = true

MinimumGZipLength = 1024

NotZippedFiles - To enable script files compression, remove the values marked in bold from this line: `NotZippedFiles = .zip, .cs, .rar, .arj, .z, .gz, .tar, .lzh, .cab, .hqx, .ace, .jar, .ear, .war, .css, .pdf, .js`



- Open the `Servlet_jsp` service settings.



- Change the values of the properties as follows:

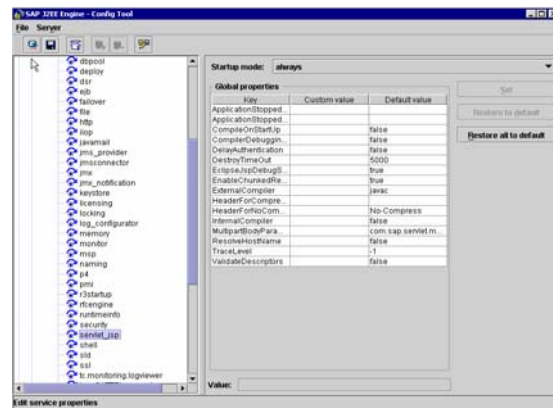
J2EE 6.40

EnableChunkedResponse = true

J2EE 6.20

EnableChunkedResponse = true

MinimumGZipLength = 1024



4.2 Content Expiration

The following properties affect caching in J2EE:

CacheControl: The period, in seconds, for which responses are kept in the client cache. When the value of this property is greater than 0, the HTTP Service sets Cache-Control header to the response message with max-age attribute equal to this value (this is true, if the client is HTTP 1.1 compliant). This means that content will be retrieved from the browser cache until its expiration date

SapCacheControl: The period, in seconds, for which requests are kept in the ICMAN cache. If a second equal request comes up within the specified period, the same response is returned straight from ICMAN (without requesting J2EE Engine again).

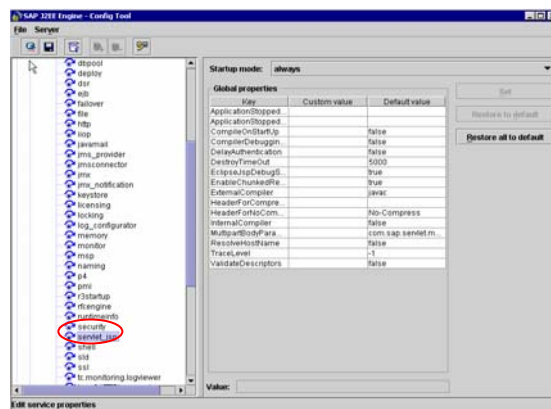
The default value for these properties is one day. The recommendation is to set the value at least for one week (604800 seconds). This means that all the static content will stay in the browser cache on the client for a specified period of time.



Users might experience some problems if the portal components were upgraded and the files that are stored in the browser cache are not updated because they have not expired. In such cases, the cache must be cleared. Either users must manually clear the browser cache, or the IT department should inform all users to clear their cache, or IT should run an automatic script to either clear each user's cache or clear only the files that were changed during the portal upgrade.

To optimize caching:

1. Open *Servlet_jsp* service settings

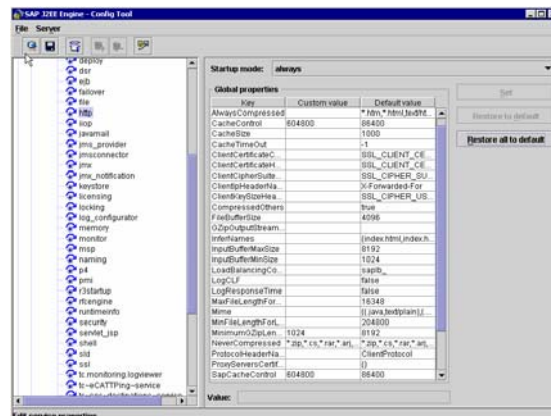


2. Change the values of the properties as follows:

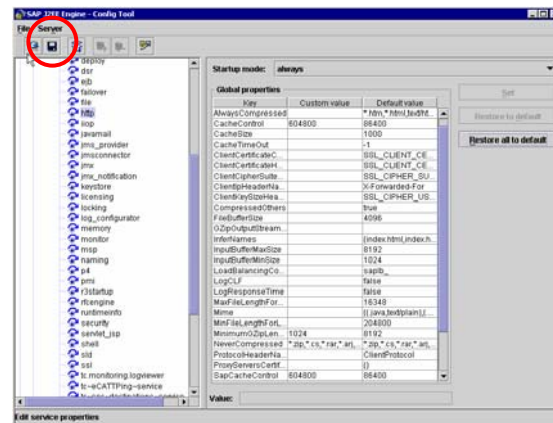
For J2EE 6.40 and J2EE 6.20:

CacheControl=604800

SapCacheControl=604800



3. Save your changes



5 Browser

5.1 Setting the Browser for Compression

Most newer browsers (since 1998) support the HTTP 1.1 standard known as "content-encoding". Essentially, the browser indicates to the server that it can accept "content-encoding" and, if the server is capable, the server compresses the data and transmits it. The browser decompresses the data and renders the page.

Only HTTP 1.1-compliant clients request compressed files. Clients that are not HTTP 1.1-compliant request and receive the files uncompressed and do not benefit from the improved download times of HTTP 1.1-compliant compression. Internet Explorer versions 4 and above, Netscape 4.5 and above, and Windows Explorer are all HTTP 1.1-compliant clients.

When using the Microsoft Internet Explorer, you must activate the HTTP 1.1 setting in order for portal compression to work.

To set the compression in Microsoft Internet Explorer:

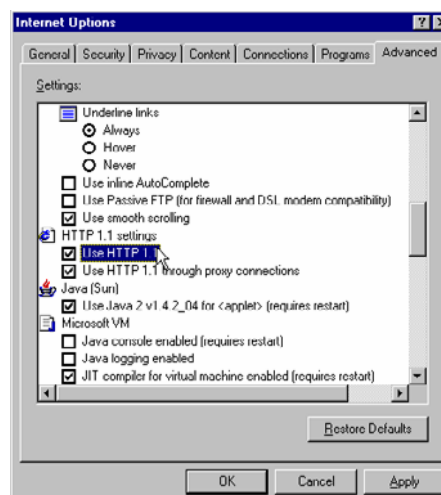
1. From the client browser menu bar, select *Tools* → *Internet Options*.



2. Choose the *Advanced* tab and scroll to *HTTP 1.1 settings*.
3. Make sure the *Use HTTP 1.1* checkbox is selected.

If your network operates with a proxy server, verify that *Use HTTP 1.1 through proxy connections* is also selected.

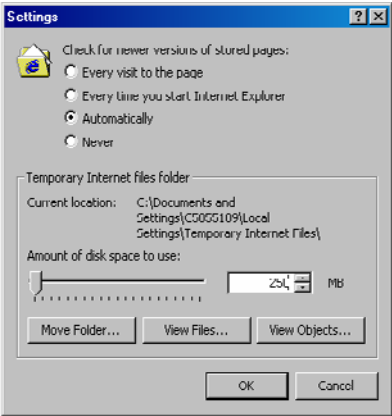
4. Click *OK*.



5.2 Browser Caching Options

Client-side caching is a key method for improving performance over WAN. By caching content in the client, the number of round-trips and network traffic are reduced and, therefore, response time is improved.

In order to use the cache mechanism on the client side, both the [portal's content fetching](#) service and the browser properties should be configured properly.

<i>Microsoft Internet Explorer 5.0, 5.5, 6.0</i>	<p>Browser Caching</p> <p>You can improve portal response time by using browser caching. This enables your browser to cache resource files such as images, CSS, JavaScript and .txt files.</p> <p>For detailed information on browser caching, see <i>Accessing the Enterprise Portal</i> under the topic <i>Setting Internet Browser Options</i> in the <i>End User Guide</i> at: http://help.sap.com → <i>Documentation</i> → <i>SAP NetWeaver</i> → <i>SAP Enterprise Portal</i>.</p> <p>To use the browser caching in Internet Explorer:</p> <ol style="list-style-type: none">1. Select <i>Tools</i> → <i>Internet Options</i>.2. In the <i>General</i> tab, click <i>Settings</i>.3. Under <i>Check for newer versions of stored pages</i> in the <i>Settings</i> dialog box, select either <i>Automatically</i> or <i>Every time you start Internet Explorer</i>. Do not choose <i>Every visit to page</i>.  <ol style="list-style-type: none">4. Click <i>OK</i> twice.
<i>Netscape 6.x</i>	<p>Browser Caching</p> <p>In Netscape 6.x, you can improve portal response time by using browser caching.</p> <p>To use the browser caching in Netscape:</p> <ol style="list-style-type: none">1. From the <i>Edit</i> menu, select <i>Preferences</i>.2. In <i>Category</i>, click <i>Advanced Cache</i>.3. Under <i>Document in cache is compared to document in network</i> in the <i>Cache</i> pane, select either <i>Never</i> or <i>Once per session</i>. Do not choose <i>Every time</i>.4. Click <i>OK</i>.

Make sure to allocate enough disk space for cached files. The default value is 10 MB. Change the value for *Amount of disk space to use* to at least 50 MB.

5.3 Temporary Internet Files Options

Performance may suffer when the browser is configured to delete the cache whenever the browser is closed. When the user opens the browser again and enters the portal, all files configured to use the browser cache will again be transferred from the server because they no longer exist in the browser.

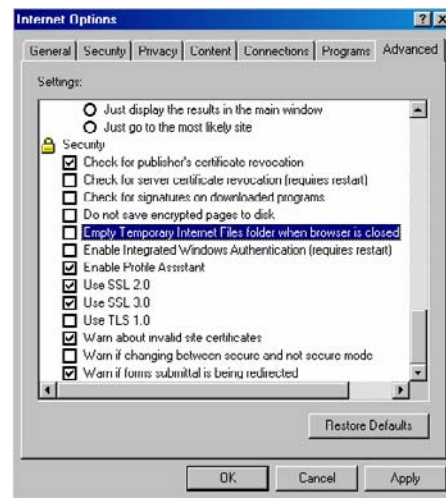
Some company's policy is to clean browser cache each time the browser is closed. Check if company policy can be changed to disable the "*Empty Temporary Internet Files folder when browser is closed*" option.

Avoid deleting the cache frequently. To this end, make sure that the property *Empty Temporary internet files folder when browser is closed* is unchecked.

Microsoft Internet Explorer
5.0, 5.5, 6.0

To prevent the cache from being emptied on browser close, do the following:

1. Select *Tools* → *Internet Options*.
2. In the *Security* section of the *Advanced* tab, de-select the *Empty Temporary Internet Files folder when browser is closed* checkbox, and then click *OK*.



3. Click *OK*.

The only concern about client-side caching is when iViews display sensitive or private content that you would not want to store locally on the browser machine. Consult SAP Performance and SAP Security groups for more information about security and caching.

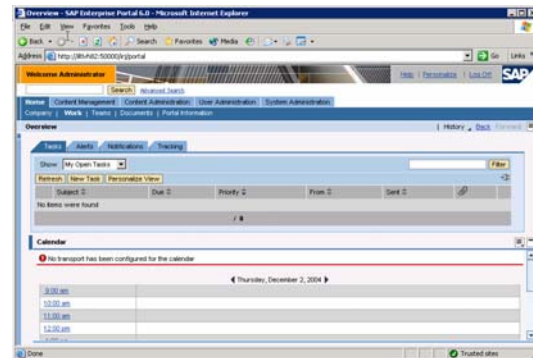
6 ITS

Internet Transaction Server is a gateway between SAP R/3 systems and the Web.

6.1 JavaScript and CSS Files Optimization

Follow the same processes described in *JavaScript and CSS File Optimization* of this document. There is one additional step when testing ITS iViews in the portal.

After logging in to the portal, open a page containing iViews that work through ITS (for example, ESS iViews). This step will load additional files to the browser cache.



6.2 ITS Compression

The ITS currently uses the gzip compression format. ITS will examine HTTP request headers for the "Accept-Encoding" field to verify that the browser supports the gzip format. HTML compression is enabled in ITS by setting the `~http_use_compression` variable in an application's service file to 1. If the variable is set to 0 or is not present, compression will be disabled.

If the browser indicates that it supports gzip decompression and the `~http_use_compression` flag is set to 1 and the response type is text/html, ITS will specify gzip compression in the "Content-Encoding" field in the HTTP response header. The transmitted data will be compressed.

Compression may be configured by use of the `~http_compress_level` variable in the application service file. Allowed values are 1-9. Better compression is achieved as higher values are used at cost of processing speed: 1 represents the fastest speed but performs little compression, while 9 offers the highest compression. The default is 7. This value achieves typical compression rates of approximately 70 percent, with no appreciable improvement for higher compression levels.

Only data that is processed by the AGate can be compressed, that is, all dynamic data that is processed through the WGateURL (default/scripts/wgate). Static data that is called on the MGateURL (default /sap/its/mimes) is not compressed. Additional tools have to be used to achieve compression on mimes. Those tools can either be part of the web server installation or third-party tools. These tools can be hardware or software tools.

Third-Party Compression Tools:

Check out the Apache, iPlanet or Microsoft web sites about compression tools supported for their particular web servers. With IIS 5.0, Microsoft delivers an additional filter called compression.dll. By default, this ISAPI filter is disabled and only compresses files with endings .htm, .html and .txt. To add support for .js and .css, check out MS TechNet:

<http://www.microsoft.com/technet/prodtechnol/windows2000serv/technologies/iis/maintain/featusability/default.mspix>.



Problems can arise if the compression tool compresses the data sent by the AGate a second time. In this case, either turn off compression for WGateURL or set `~http_use_compression` to 0.

IE Browsers rely on [HTTP 1.1](#) to allow compression. For information on required browser settings, see [Setting The Browser For Compression](#).

7 NTLM Authentication Impact on Network Traffic

SAP Enterprise Portal 6.0 can be used with an IIS and Windows Integrated Authentication (NTLM) as the portal authentication method. It is easy for the users to authenticate to the portal because, once they log on to their workstations, they are authenticated to the domain. It would require that they have the IISProxy filter on an IIS web server, and they wouldn't have to logon to access the Portal. Unfortunately, working in this scenario can significantly influence the amount of network traffic between J2EE-IIS-Portal. This section describes a number of ways to optimize IIS and IISProxy for better network utilization.

To use Microsoft's IIS as an intermediary server with the J2EE Engine, the IISProxy plug-in module provided by SAP should be installed. This module replaces the J2EE ISAPI module that was provided with Release 6.20. The IISProxy module contains an ISAPI filter and an ISAPI extension. It consists of the dynamic link library IISProxy.dll and the corresponding configuration file IISProxy.xml. To support the use of SSL, it uses the SAP Cryptographic Library.

NTLM employs a challenge-response mechanism for authentication, in which clients are able to prove their identities without sending a password to the server. It consists of three messages: "negotiation", "challenge" and "authentication". To read more about NTLM, see <http://curl.netmirror.org/rfc/ntlm.html>. This authentication mechanism allows clients to access resources using their Windows credentials, and is typically used within corporate environments to provide single sign-on functionality to intranet sites.

How NTLM HTTP Authentication works:

1. The client requests a protected resource from the server.
2. The server responds with a 401 status, indicating that the client must authenticate.
3. The client resubmits the request with an "authorization" header containing a "negotiation" message parameter.
4. The server replies with a 401 status containing a "challenge" message in the "WWW-Authenticate" header.
5. The client responds to the "challenge" message by resubmitting the request with an "authorization" header.
6. Finally, the server validates the responses in the client's "authentication" message and allows access to the resource. The server replies with status 200 and the content required.

NTLM generally has a negative impact on network traffic. It increases the traffic by as much as 50 percent.

It is worthwhile to do several tests network traffic using NTLM. The basic steps in such a test are (assuming the browser cache is activated and working properly):

1. Test amount of network traffic when working with NTLM + empty browser cache
2. Test amount of network traffic when working with NTLM + full browser cache
3. Test amount of network traffic when working without NTLM + empty browser cache
4. Test amount of network traffic when working without NTLM + full browser cache
5. Compare results 1 and 3, and 2 and 4, and calculate the difference for each pair.

7.1 Compression

If you want to use compression, compress either in IISProxy ISAPI module or in J2EE, but not in both.

- The IISProxy configuration file IISProxy.xml contains by default the compression of the following Mime types: <compress-types>text/html, text/plain, text/css</compress-types>. For Browser versions IE5.5 and older or Netscape 4.78 and older, the compression of the IISProxy ISAPI module has to be deactivated for the MIME type text/css (MS knowledge base 275330).

Problems with Portal Compression

With the portal, an ISAPI filter is installed called *compress.dll*.

The portal may run into problems if you use:

- HTTP keep-alive with IISProxy (keep-alive=true in IISProxy.xml), and;
- EnableChunkedResponse=true on the J2EE engine, and;
- compress.dll portal ISAPI filter

In this case, remove the compress.dll ISAPI filter from the portal or set EnableChunkedResponse=false on the J2EE Engine. The IISProxy ISAPI module also has implemented full compression, so further compression is not needed.

7.2 HTTP Errors Optimization

As described above, authentication and resource retrieval from the server is performed in six steps, not only once but for each thread that requests the information from the server and every time the server detects that the connection to the client has been dropped. This can increase network traffic and the number of round-trips.

During every authentication process, the server replies twice with a 401 error. Not only is the error status transferred to the client, but all the HTML error content. The 401 error response body size is ~4K. The error is transferred at least 10 times when working via NTLM (i.e., 40K of extra network traffic). The error message (HTML) is not seen by the client, so it is not needed.

To reduce error texts transferred to the client:

1. Check the location of the error files in IIS.

Open *IIS Manager* → *Web Sites* → *Custom Errors*

See the location of error file.

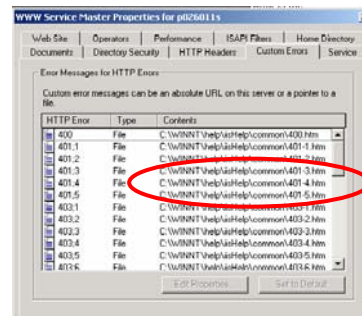
2. Open *Properties* → *Edit* (IIS 5.0) or *Web Sites* (IIS 6.0).

Open *Custom Errors*.

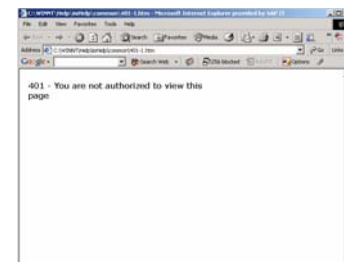
See the location of the error file.

3. Go to the error file and clean up the error message as much as possible.

It is best to leave only one line containing the error number and a short message description.



Before:



After:

8 Network Analysis and Test Scenarios

8.1 Analysis Tools

This section explains the tools for analyzing portal performance.

Network Sniffers

Trace all kinds of network communication directly on the network card: HTTP, LDAP, Database, CIFS/SMB (remote shares), by default, the tool displays just network packets and interprets them according to the used protocol. One common feature is to recombine the packets to the TCP stream again, such that, for example, the HTTP content can be displayed. Due to packet switching, the network sniffer must be active on one of the two communication endpoints of the network communication to be analyzed. With network switches, other hosts will normally not see any IP packets that are not addressed to them.

- **Ethereal** (www.ethereal.com) – open source, underlying winpcap, runs only on single-CPU hosts (i.e., can normally not be used on portal server itself), is used to capture data (maybe directly with filter), find relevant packages, follow TCP stream to display the complete conversation.
- **HttpWatch 3.2** (<http://www.simtec.ltd.uk/buy>) – HttpWatch integrates with Internet Explorer to provide HTTP monitoring without the need for separately configuring proxies or network sniffers. When interacting with a web site, HttpWatch displays a log of requests and responses alongside the web page itself, showing interactions between Internet Explorer and the browser cache. All data can be saved to an XML file for further processing.
- **HttpLook 1.3** (<http://www.httpsniffer.com/>) – HTTPLook provides a way to test and debug internet applications, working over HTTP protocol, such as web sites, web services and their clients, allowing tracing of messages by which these applications communicate. HTTPLook captures, without any modification, data traffic of all internet applications, including encrypted SSL traffic of Internet Explorer. It parses HTTP messages, separates headers and entities, decodes compressed entities and saves entities in the file system. Sorting, filtering and search features are provided.

8.2 Optimization Tools

- **JSMIN** (<http://www.crockford.com/javascript/jsmin.html>) – JSMIN is a filter which removes comments and unnecessary whitespace from [JavaScript](#) files. It typically reduces file size by half. It also encourages a more expressive programming style because it eliminates the download cost of clean, literate self-documentation.
- **CSS Optimizer** (<http://flumpcakes.co.uk/css/optimiser>) – This tool parses a CSS file and outputs a leaner version, removing any redundancies and attributes that are not needed (removes comments and white space, converts multiple background, font, margin, padding, list attributes into a single attribute).

8.3 Test Scenarios and Examples

The factors that influence portal performance are:

- The network bandwidth of the different groups of users. These users can be broken up into two distinct groups for the purposes of these tests.
 - Users with 56 kbps or less bandwidth
 - Users with more than 56 kbps bandwidth
- Client-side caching
- Amount of actual data transferred across the network

This section describes a test plan for measuring the impact of each of these on the overall response time of portal navigation. The tests will measure the amount of data transferred across the network in each of the following scenarios:

Empty Client Cache:

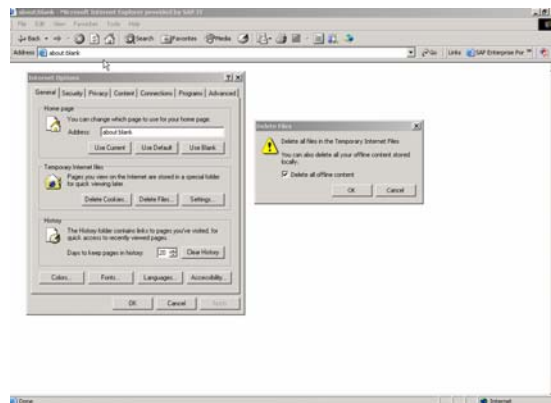
1. Clear browser cache.
2. Record data transferred across network.
3. Open portal login page.
4. Save the data recorded until now (for portal login page).
5. Record data transferred across network.
6. Log into the portal with test users (number of different test users, with different role composition and no role at all, in order to quantify role content impact).
7. Record data transferred across network.

Full Client Cache:

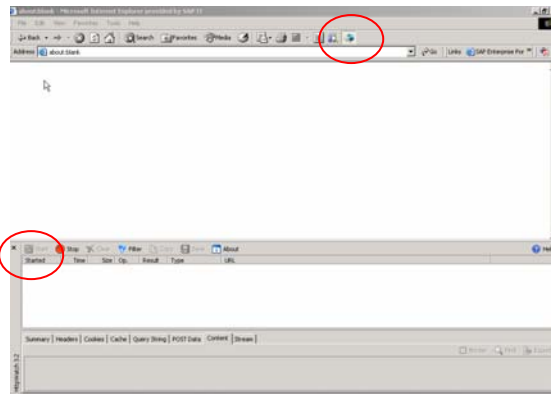
1. Record data transferred across network.
2. Open portal login page.
3. Record data transferred across network.
4. Log into the portal with test users (number of different test users, with different role composition and no role at all, in order to quantify role content impact).
5. Record data transferred across network.

Scenario Example:

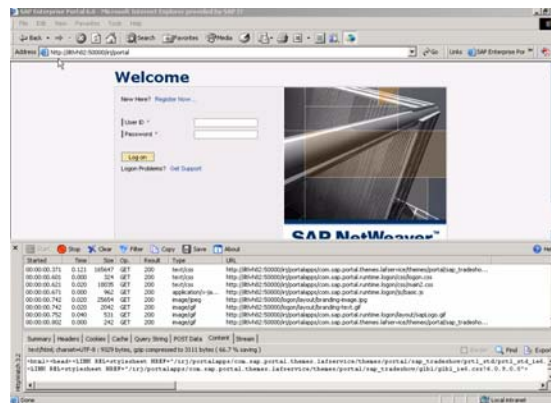
1. Clear browser cache.



- Record data transferred across network.



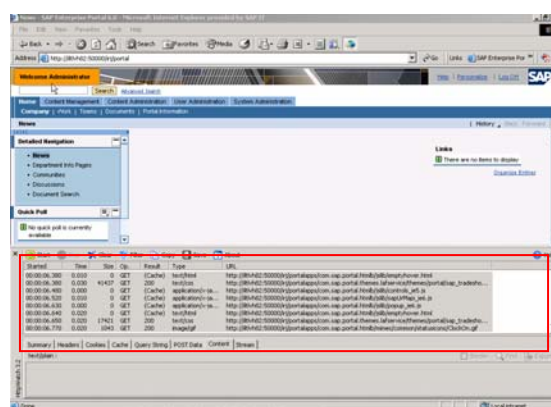
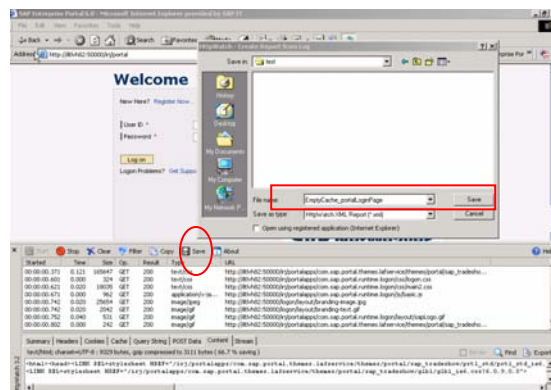
- Open portal login page.



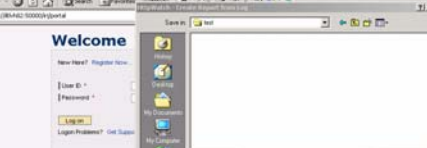
- Save the data recorded until now (for portal login page) and clear the monitor data,

or

select the results and copy and paste it to an XML file for further processing.



- [illegible]

- 

Analyzing results:

1. Open the XML file.

[illegible]

2. Order it by size.

[illegible]





What can we understand from the data received?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	00:00:01.703	0.29	296271	GET		200	application http://ltlwh82:50000/irj/portalapps/com.sap.portal.htmlb/jslib/sapUrMapi_ie6.js							
2	00:00:01.432	0.14	126934	GET		200	application http://ltlwh82:50000/irj/portalapps/com.sap.portal.htmlb/jslib/controls_ie5.js							
3	00:00:04.657	0.11	68200	GET		200	application http://ltlwh82:50000/irj/portalapps/com.sap.portal.ui.uiservice/script/dynamicTree.js							
4	00:00:01.152	0.1	41614	GET		200	application http://ltlwh82:50000/irj/portalapps/com.sap.portal.pagebuilder/scripts/pagesupport.js							

- We can see that the data didn't come from the browser cache because status 200 is written in column E. If the data arrived from the cache, "Cache" would have been written.

Next Step: Rerun the same test without deleting the browser cache. If the data for static files, such as JavaScripts and CSS, do not come from the browser cache, see the configuration in *Content Expiration* chapter.

Next Step: Go to the browser cache and check the file content:

Name	Internet Add...	Type	Size ▾	Expires
 sapUrMapi_ie6.js	http://iltlvh8...	JScript Script ...	290 KB	12/3/2004 7:57 PM
 ur_ie6.css?6.0.9.0.0	http://iltlvh8...	Cascading St...	162 KB	12/3/2004 7:53 PM
 controls_ie5.js	http://iltlvh8...	JScript Script ...	124 KB	12/3/2004 7:57 PM
 dynamicTree.js	http://iltlvh8...	JScript Script ...	67 KB	12/3/2004 7:57 PM

The file is loaded to the cache on 12/02/04 but the expiration date is 12/3/2004, only a one day difference, so it should be changed to a greater value. See *Content Expiration*.

- We can see that the data received is uncompressed because all the JavaScript files size arrived as is (with their original size).

Next Step: See *HTTP Compression*.

- The loaded JavaScript files are too big.

Next Step: See *JavaScript and CSS File Optimization*.

Browser Cache Analysis

Most problems or optimization options can be seen simply by looking at the files saved in the browser cache. In the cache, you can see from where the file loaded (*Internet Address* column), the size of the file (*Size* column), the file expiration date (*Expires* column) and so forth.

It is important to check that the expiration date for all static content (JS , CSS, GIF and other such file types) is big enough. Not only portal content should have proper expiration dates but also content retrieved from other systems (for example, ITS and BW). However, you do not always have access or control over these system. When you notice that some files have bad expiration dates, look at each one of these file's *Internet Address* column, which indicates from where the file was delivered. If you know the source, you can check if it is possible to configure the source server to send its static content with proper expiration dates. Please see *Content Expiration* for how to configure J2EE for sending data with expiration dates.

Appendix A: Background Information

This Appendix provides additional background information on portal performance, including links to related Web sites.

1. Portal Framework Page Organization

When users log on to the portal, they receive the portal framework page, which contains the pages and iViews that they need to operate and navigate in the portal.

The default framework page includes:

- **Header Area**

The header area is the part of the framework page that contains the Masthead, Tools Area and Top-Level Navigation (TLN) iViews. The iViews in this part of the portal are loaded once, when the user logs on. From then on, they remain available until the client's browser is closed.

Content in these iViews does not reload with every navigation action.

- **The Desktop Innerpage**

This area contains the Navigation Panel, the Page Title Bar and the Content Area. Technically, this area is inside an iFrame in the framework page. The Desktop Innerpage reloads with every navigation action.

- **Navigation Panel**

The navigation panel displays navigation iViews with content and links related to your tasks or to the content displayed in the Content Area, including the Detailed Navigation (DTN), Related Links and others.

- **Page Title Bar**

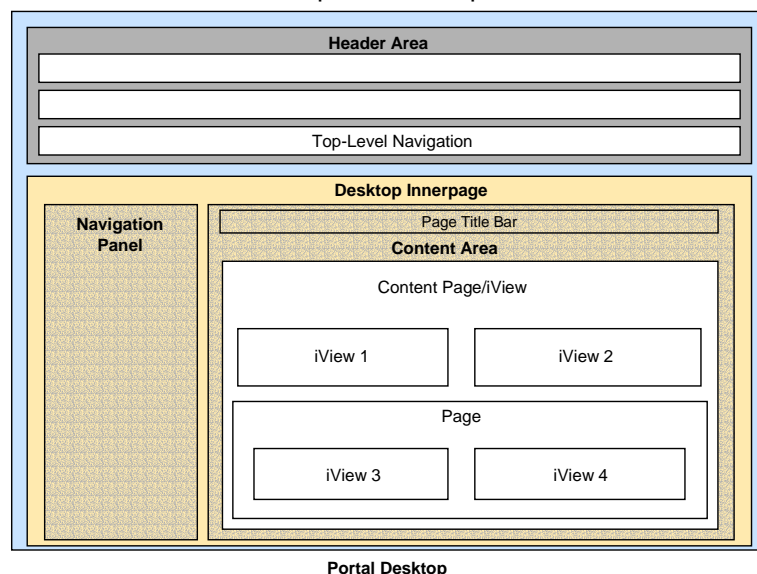
The page title bar is an iView that appears at the top of the content area and contains items related to the identification, personalization and navigation of the currently displayed page in the content area

- **Content Area**

The part of the Desktop Innerpage that displays content pages/iViews, which contain the actual content displayed to the user.

The content is displayed as pages/iViews. A page can contain other iViews or pages. The content in the page/iView can be either embedded or URL isolated.

The following figure illustrates the areas of the portal desktop.



During design time, the portal administrator generates and customizes framework pages to suit different users and scenarios.

2. NTLM Authentication Protocol

For more information on NTLM authentication protocol, go to <http://curl.netmirror.org/rfc/ntlm.html>.

3. HTTP Latency

For more information improving HTTP latency, go to <http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/mogul/HTTPLatency.html>.

4. JVM Settings for NW '04

The information in the notes listed below is relevant if you want to manually configure your portal installation.

Configuration templates are only applicable to newly installed systems based on the J2EE Engine 6.40 SP9 (corresponds to NetWeaver'04 SP Stack 09) or above. For all other systems, configure JVM settings as described below.

Before starting manual configuration, please read the following SAP Notes:

723909	This note describes recommended JVM settings for EP6 SP3 and higher
709140	Recommended JDK and VM Settings for the WebAS630/640
716604	for the Sun JDK (Windows, Linux, Solaris)
716926	for the HP JDK (HP-UX)
722520	for the Tru64 JDK
716927	for the IBM JDK (AIX)
717376	for the IBM JDK (iSeries, OS/400)

Since the above notes only take into account the "default" J2EE installation, some parameters must be set differently for an EP6 installation. The java parameters have to be set via the configtool.

Appendix B: Use Cases

Use-Cases: Setting Isolation Methods and Cache Levels

As previously mentioned in this document, there are many factors that determine isolation methods and cache settings. The Appendix describes several use-cases for isolation and cache settings, based on some of the determining factors.

Case 1: Company News Page in the Portal

This case presents four scenarios describing a portal page that displays company news.

The company news scenario represents content which is mostly static (valid for a longer period of time, for instance over 24 hours).

Content in this page will be viewed several times from the same client machine before its validity expires.

- **Case 1a: Company News Page in the Portal**

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
URL	Server, Browser Browser cache is effective.	Embedded	Server, Browser

Comments

The first navigation to this page involves two round-trips. In subsequent navigation, the content page is fetched from the browser cache, (thus one navigation round-trip and one conditional request in further navigations).

- **Case 1b: Company News Page in the Portal with Scroll Bars**

In this scenario, users can move the displayed information up or down in the display area of the iView. This scenario also covers URL iViews.

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
If page is used, apply Embedded. If iView is used, apply URL.	If page is used, apply None. If iView is used, apply server and browser caches.	URL	Server, Browser

Comments

Due to the need for scroll bars in the iViews, a URL isolation method is applied to the iViews. This enables the use of browser cache for the iViews' content.

To eliminate additional (unnecessary) round-trips for the page, Embedded isolation is applied to it.

- **Case 1c: Company News Page in Portal without Browser Cache**

This scenario is similar to that of Case 1a above. The difference is that the browser cache cannot be used because the content is confidential, or there is no advantage to using the browser cache (the page will not be viewed several times from the same machine before its validity expires, or the cache validity period is short on the server).

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
Embedded	If page is used, apply None. If iView is used, apply cache on server.	Embedded	Server

Comments

The browser cache is not used, and there is no need to isolate any content in an iFrame. All content is embedded, thus navigation to this page involves only one round-trip.

- **Case 1d: Company News Page in the Portal with Scrollbars, no Browser Cache,**

In this scenario the browser cache is not to be used (same reasons as in case 1c). However, scrollbars should be used for the iViews (for the reasons described in case 1b).

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
If page is used, apply Embedded. If iView is used, apply URL.	If page is used, apply None. If iView is used, apply cache on server.	URL	Server

Case 2: Page with Java iViews (Interactive or Application)

This case describes a portal page with content that requires many user-interactions. The iViews in this case are implemented as java iViews (portal components), with their content rendered in the portal server; for example, iViews that load applications, and require user inputs such as submitting data, or performing actions within the content.

In this scenario, there is no need for interaction between the iViews on the server-side, only user actions (no POM events).

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
If page is used, apply Embedded. If iView is used, apply URL.	If page is used, apply None. If iView is used, apply cache on both the server and client, when Shared cache is used. When User or Session cache is used, it is recommended to apply only client cache in order to save server resources.	URL	Server, Browser Apply cache on both the server and client, when Shared cache is used. When User or Session cache is used, it is recommended to apply only client cache in order to save server resources.

Comments

Content is isolated. This drives more round-trips when navigating to the page, but reduces network traffic on additional user interactions within an iView (only the content of the iView is re-loaded, not the entire page, or desktop Innerpage).

Case 3: Page with URL iViews

This case describes a portal page with URL iViews.

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
Embedded	None	URL	According to the type of content

Comments

URL iViews have only URL isolation, so the page should be Embedded. If content caching is essential, then apply Shared cache.

Case 4: Page with iViews' Content Coming from Back-End Applications

This case describes a portal page with iViews that have content coming from back-end applications (based on the AppIntegrator iView template).

The table below shows the isolation methods and cache settings applicable to the page and its iViews:

Page/iView		iView in this Page	
Isolation	Cache	Isolation	Cache
Embedded	None	URL (only option)	None

Comments

AppIntegrator-based components that retrieve content from back-end applications have permanent URL isolation and None cache level settings.

Remarks on Use-Cases:

- Content Area will display either a Page with iViews in it, or an iView that will be the only displayed iView (Full Page Application). Note that sometimes the settings differ between the two.
- When applying isolation methods, usability issues may contradict performance issues. For example, Embedded isolation is preferred for performance reasons, while URL isolation is necessary for scroll bars and height control.
- Set the Cache level (Shared / User / Session) according to the content itself, and whether user personalization is required. You can use a combination of cache levels between the page and its iViews, for example a page with Shared cache, containing iViews with User cache.
- If you use Shared cache level, then disable the personalization options for users.

Portal Welcome Page

The Welcome page serves as the first entry point into the portal for all, or most users. As such, the portal's welcome page is accessed many times more than any other page. For this reason, it has a significant impact on the over-all performance of the portal and in particular on the login time.

The following are some guidelines to use when planning the content of the Welcome page:

- Use static content such as company news and reports, rather than transactional or interactive content.

Consider the fact that each user accesses the Welcome page, and often leaves it very quickly to go to another page. As such, placing iViews that require interaction with applications on the Welcome page causes all the applications to load for all users, although most users will never use them.

- Use public content that can be set with Shared cache level and longer cache validity for content.
- Use browser cache in order to minimize server round-trips. Such enhancement in performance can be realized by using Embedded isolation for the iViews, URL isolation for the page, and setting an appropriate cache level and validity period for both iViews and the page.
- If your page contains URL iViews (which cannot be embedded), use Embedded Isolation for the page without cache (cache level = none).
- Keep the number of iViews relatively small, especially the number of URL isolated iViews, which should not exceed 5.

Constructing Welcome page

When constructing a Welcome page, you can use the following sample:

- Page:
 - Five iViews with Embedded isolation (in-house Java iViews which present company daily news, links to updated reports, the latest CEO vision, and personal links).
 - Page Isolation = URL
 - iView Isolation = Embedded
 - Cache level = Shared for all
 - Validity = between 1 to 7

www.sap.com/netweaver