



# Altio Application Cache Overview

---

## ***Altio***

This document describes the cache of the Altio applet and screen preferences. It focuses upon issues associated with this caching, when and how to clear the cached files.

Document Version -	1.0
AltioLive Version -	5.x
Date -	Oct 2007

# Altio Technical Note – File Caching

---

## 1 Introduction

To improve performance and usability of applications Altio and the environment Altio operates within can be configured to cache certain files. In the majority of circumstances this will never cause a problem to the end user but when testing different versions of the Altio product or applications built using Altio it is possible that problems may occur.

The most common problems that occur are:

- Incorrect version of Altio Applet loaded into browser
- User preferences cause screen corruption in updated Altio application.

This document provides a brief overview of the files that can be cached as part of an Altio configuration, the issues that can occur and recommendations for maintaining the cached files.

# Altio Technical Note – File Caching

## 2 User Preferences

**Summary:** User preferences are a cache of each users screen layout. This feature is enabled or disabled at design time and so may not be available in all Altio applications or windows.

User preference data holds the following information about the users desktop setting; specifically:

- The number of windows (both initial windows and any windows opened while the application is active).
- The state of each window (whether minimized or normal).
- The 'normal' position of each window.

Note that the following information is not saved:

- The layered order (z-order) of windows
- Selection within a window (e.g. selected row) or control (e.g. selected text)
- Scroll position in a list
- Information entered into a form/control that has not been submitted
- The 'active' window

### Problems:

#### *Inconsistent Client Behaviour*

When the application is changed in the designer it is possible that list columns are changed or the controls on a window are removed or modified. Under some circumstances it is then possible users will experience inconsistent screen behaviour e.g. data appearing in the wrong columns of list controls.

**Solution:** After changes to screens in the Altio designer remove the user preferences stored on the server. This will result in all users having their screen layout reset to the default as defined at design time.

User Preferences are stored either on the server as files or in a database store. When preferences are stored in a database table then deleting the rows in the table removes the preferences. Preferences stored on a file server are kept in the directory  
\${Altio installation directory}\\${web app folder}\altio51\WEB-INF\classes\preference,  
e.g. C:\Program Files\AltioLiveStudio5.1.1Professional\tomcat\webapps\altio51\WEB-INF\classes\preference

Deleting all of the files in this folder removes the user preferences.

## 3 Applet

**Summary:** To improve download time of applications it is possible to cache the Java applet. The setting to cache applets is enabled by default when the Java Virtual Machine is installed.

### Problems:

#### *Incompatible versions of Client and Server.*

Occasionally, and often during testing, the same URL is used to access Altio but with different versions of Altio. For example, testing version Altio 5.0 and then Altio 4.4 on the same URL can cause compatibility issues due to cached client files. Because Altio 5.0 will have cached the newer version of the Applet it will be incompatible with the server installation this will present the user with an error message. If testing

# Altio Technical Note – File Caching

Altio 4.4 and then moving to Altio 5.0 there will not be a problem because the Altio 5.0 client applet will have a later date stamp.

The solution to the problem is to remove the client applet cache as described later in this section.

## ***Inconsistent Client Behaviour***

There may be circumstances where the same major version of Altio is tested using different minor versions. This will not cause a license issue but may cause confusion or unexpected results when interacting with a Altio application.

This problem occurs when a newer version of the Altio client is used and then replaced on the server with an earlier version e.g. use Altio 5.1.2 client jars and replace them with Alto 5.1.1. When client Java Applet caching is enabled the date stamp of the client files and the URL where the files came from are compared to the server, if the date on the client is newer than the version on the server then the client version is used. This can lead to testers attempting to verify a bug and cannot reproduce it because the newer version of the Altio client files are being used rather than the older files.

Removing cached applets will rule out caching of client files as the cause of any problems.

## **3.1 Deleting the Applet Cache**

**NOTE: It is important that all internet browsers are closed before deleting the cache otherwise Applets that are in use will remain cached.**

The caching of Applets is controlled through the Java Control Panel, which is accessible on a Windows system through the control panel (Figure 1).



Figure 1 – Control Panel

# Altio Technical Note – File Caching

Once the Java Control Panel is open then select the “Settings” button in the “Temporary Internet Files” section (Figure 2).



Figure 2 – Java Control Panel

The “Temporary Files Settings” window is then displayed, selecting the “Delete File..” button removes all cached Applets and forces a new download of the applet the next time the browser access a page with the Altio applet hosted within it.

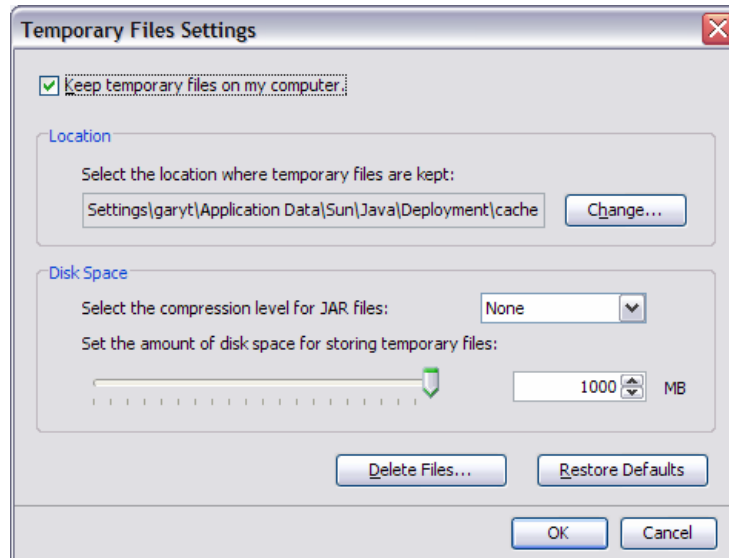


Figure 3 – Temporary Files Settings

# Altio Technical Note – File Caching

---

## 4 Datastores

Whether or not to use datasstores is set by a flag in the host page: `var cacheDatastores = "N";` This defaults to Y.

When enabled, the client attempts to restore a portion the Altio Applet's local data cache if a user moves away from the Applet page and back again, or reconnects with a still valid session. This is useful for reducing bandwidth so that the data is not re-requested from the server, and also tries to preserve the state of the session. How much local data is preserved depends on the number of inactive datastore pools configured, default of 1. This is configured by another host page setting: `var maxInactiveDataStores = 1;`

Datasstores are flushed when a client's session expires.

These are fairly advanced settings and not documented, but turning off datasstores can sometimes avoid unexpected behaviour when reloading an application.

## 5 Cookies

The Altio product does not need or use client side cookies. As a requirement of a system the Altio product may be operating within then cookies may be required. This means that it will not normally be necessary to clear web browser caches.

## 6 Bibliography

1. Applet Caching  
[http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer\\_guide/applet\\_caching.html](http://java.sun.com/j2se/1.5.0/docs/guide/plugin/developer_guide/applet_caching.html)

## 7 Appendix – Altio User Preferences

Extracted from the Altio online help

### Setup User Preferences

---

User Preferences remember the 'state' of the user interface and make it possible for a user to configure the desktop during a session and have it preserved for their next session.

User Preferences are specific to each Application configuration definition. If the same user logs in to separate Applications, their preferences are specific to each Application.

The saving of user preferences can be turned off for an application; this will lessen the demand on the server and the amount of data to transmit from the client. It can also be turned off for individual windows.

#### What is saved

User preference data holds the following information about the users desktop setting; specifically:

- The number of windows (both initial windows and any windows opened while the application is active).
- The state of each window (whether minimized or normal).
- The 'normal' position of each window.

Note that the following information is not saved:

- The layered order (z-order) of windows
- Selection within a window (e.g. selected row) or control (e.g. selected text)
- Scroll position in a list
- Information entered into a form/control that has not been submitted
- The 'active' window

Also note that if the layout of the application has been changed since a user last accessed it, only the preferences which are still valid in the new layout will be restored - any others will be ignored. For example, if a window has been removed, any preference information saved for it will be ignored when the user logs in again. Also, if a new window has been added, it will start in its default position.

#### When preferences are saved

You can set how often the preferences are saved in the Application Properties. It is recommended that you set this value to the same as the Keep-alive interval to avoid continual update of the preference store when the user drags windows around and so forth.

Changes are saved when the user closes the browser. Changes are also saved when the browser page is changed by Back/Forward/Refresh-type actions.

#### How preferences are saved

Preferences are saved as XML data. The Preference store is case insensitive for the user ID and Application name. Preferences can be saved locally in the specified Presentation Server preference directory (this is the simplest implementation) or externally to the Presentation Servers in the Application environment. This entails using either a database or a custom servlet.

If you are using a database or servlet, it should be capable of handling the frequency of updates from the user-base. Furthermore, if you are working in a clustered environment, the database or servlet must be located on the Master Presentation Server machine.

Using a File Store approach, the File Servlet invoked will be implicitly the one on the Master Presentation Server. Using the other two approaches, each Presentation Server is configured to access the Database or Servlet directly. There are few benefits to using a database or custom servlet when using a single Presentation Server. When using multiple Presentation Servers, such an approach will make data easier to access as is it stored in one location, and also may be faster than using file storage.

If the Application Configuration is changed then the preference store should be erased if the changes invalidate what may be already saved.

#### Preference Details

# Altio Technical Note – File Caching

In the Explorer window, select **User Preference Store**. The properties window on the right will show the modes of saving User Preferences and associated parameters.

The **Storage Type** dropdown list lets you select the type of preferences to set.

## File

To simplify integration with Applications which have no mechanism for saving or retrieving User Preference information, the Presentation Server can be configured to support a simple file store. The filename uses the user login name, application id and view name.

When the **FILE** option is selected, the following fields are displayed:

- |                    |   |
|--------------------|---|
| <b>File Path</b>   | Defines the subdirectory (under the Presentation Server root directory) that will be used to store the preferences. This directory must already exist - the Presentation Server will not create it.   |
| <b>File Prefix</b> | Each user has a single file (per Application Configuration) in which their preferences are stored. This attribute defines the prefix that is concatenated with 'user-id' and Application Configuration name to create a unique file name. E.g. the preferences for user-id JIM to access the application ROS would be stored in 'config_JIM_ROS.*' within the 'PreferenceStore' subdirectory. |
| <b>File Suffix</b> | Each user has a single file (per Application Configuration) in which their preferences are stored. This attribute defines the suffix that is concatenated with 'user-id' and Application Configuration name to create a unique file name. E.g. the preferences for user-id JIM to access the application ROS would be stored in '*_JIM_ROS.user' in the 'PreferenceStore' subdirectory.       |

In the case of a user JIM using the application ROS, user preferences would be retrieved by an HTTP request with the following parameters:

Option=GetPref&UserId=JIM&ConfigId=ROS

This will retrieve whatever data is stored and return it. If no data is stored (i.e. this is the user's first login) then an empty string is returned.

In the case of a change, preferences are overwritten by an HTTP request with the following parameters:

Option=PutPref&UserId=JIM&ConfigId=ROS&Prefs=data...

Which should have the effect of overwriting the saved preferences with whatever is supplied as 'data...'. There is no mechanism for returning an error.

Note: The file store is specific to each Presentation Server and therefore is not recommended for clustered Presentation Servers.

## Database

Preferences can be saved/retrieved from a database accessible to the Presentation Server.

When the **DATABASE** option is selected, the following fields are displayed:

- |                    |   |
|--------------------|---|
| <b>DB Driver</b>   | Defines the Database driver. This is specific to the database product used and the value to be used here will be defined in the vendor's documentation. |
| <b>DB URL</b>      | Defines how the to connect to the database. This is implementation specific and dependent on the driver that is installed.                              |
| <b>User ID</b>     | User ID required to make a connection.  |
| <b>DB Password</b> | User password required to make a connection.  |
| <b>DB Catalog</b>  | The name of the database instance   |

# Altio Technical Note – File Caching

<b>DB Table</b>	The name of the table within the database instance.
<b>DB Column Name</b>	Column name within the table that holds the User ID. This must be unique and defined as varchar.
<b>DB Column Config</b>	Column name within the table that holds the Application Config root name such that DBCOLNAME and COLCONFIG together create a unique key.
<b>DB Column Data</b>	Column name within the table that stores preference data. This must be defined as a varchar and will be populated with preference data.

## JNDI data source

Preferences can be saved/retrieved from a database accessible via JNDI. This is similar to **Database** store, but because of JNDI, the connection pool is managed by the application server.

When **JNDISTORE** is selected, the following fields are displayed:

<b>JNDI Name</b>	JNDI name for the data source
<b>DB Table</b>	The name of the table within the database instance.
<b>DB Column Name</b>	Column name within the table that holds the User ID. This must be unique and defined as varchar.
<b>DB Column App</b>	Column name within the table that holds the Application ID.
<b>DB Column Config</b>	Column name within the table that holds the Application Config root name such that DBCOLAPP, DBCOLNAME and COLCONFIG together create a unique key.
<b>DB Column Data</b>	Column name within the table that stores preference data. This must be defined as a varchar and will be populated with preference data.

## Servlet

Preferences can be saved/retrieved from a servlet by the Presentation Server.

When the **Servlet** option is selected, the following fields are displayed:

<b>URL</b>	Defines the URL that is invoked for saving/retrieving preference data - this is implementation specific.
<b>Action parameter</b>	Name of the parameter used to pass the action (save/retrieve).
<b>Name parameter</b>	Name of the parameter used to pass the User ID.
<b>Config parameter</b>	Name of the parameter used to pass the Application Config root name.
<b>Data parameter</b>	Name of the parameter used to pass the preference data.
<b>Get Action</b>	Value used to indicate a retrieval of preference store information.
<b>Put Action</b>	Value used to indicate an update of preference store information.

## 8 Appendix – Cookie Handling in Altio

Extracted from the Altio online help

### Server Side Cookie Handling

# Altio Technical Note – File Caching

---

A server, when returning an HTTP object to a client, may send a piece of state information which the client will store. Included in that state object is a description of the range of URLs for which that state is valid. Any future HTTP requests made by the client which fall in that range will include a transmittal of the current value of the state object from the client back to the server. The state object is called a **cookie**.

Server uses "Set-Cookie" or "Set-Cookie2" HTTP response header to send back the state. AltioLive HTTP service and SOAP service use HTTP protocol, therefore will handle these headers. When Presentation Server processes HTTP response and sees "Set-Cookie" or "Set-Cookie2" header, it creates a cookie object internally and stores it depending on service's cookie scope setting (**PS** | **APP** | **SESSION**).

**Session (SESSION)** scope means the cookie is stored per user session. All subsequent calls to services that have "SESSION" cookie scope from the same user session will have access to this cookie (including LINK window from different applications).

**Application (APP)** scope means the cookie is stored at the application. This cookie is shared between all services (in the same application) with APP cookie scope. This is useful if you want to maintain an application level connection to the backend (i.e datapool).

**Presentation Server (PS)** scope means the cookie is stored at Presentation Server level. This cookie is shared between all services with PS cookie scope.

You can also send client side (browser) cookies to the backend, but if a client side cookie is found in Presentation Server, the server side cookie take priority.

You shall also note that when testing a HTTP or SOAP service, the response "Set-Cookie" header will be processed as they are invoked normally.

Presentation Server uses standard rule to determine whether a cookie should be sent back to a domain. You can see the document from the following link, [http://wp.netscape.com/newsref/std/cookie\\_spec.html](http://wp.netscape.com/newsref/std/cookie_spec.html)

**End Of Document**