

Product Engineering Guide

OSM v20 WebView v12

Introduction

The WebView OSM is a Web server and is automatically loaded as part of ObServer.

WebView shares the common features of many web server applications, such as:

- Accepting HTTP requests from the network and responding to them with the requested content. For example, html, xml, image file or other document.
- Configuration of what content is available and how it is presented by the server.
- Authentication. Requesting a user name and password before allowing access to some resources.
- Logging detailed information about client requests and server responses.

The content served by WebView does not just include static files located on the disk; such as images, text, documents, etc. WebView can also generate dynamic content from pages created using the WebView Editor. The editor requires no HTML authoring knowledge and allows a library of items to be placed on a page in a graphical environment.

The WebView Server supports a library of items that may be included on a page. This library includes the typical type of items found on a Web page, such as:

- Static text
- Images
- Links to other pages
- Objects and applets – such as Flash, Java, or embedded documents from other sources, etc.

In addition, as part of ObSys, WebView can present real-time content from the many interconnected systems found within a building; including data acquisition modules, power meters, BMS controllers, lighting, and power management systems such as UPS generators.

The WebView Server library also includes the following real-time items:

- Dynamic text values
- Numerical values – e.g. temperatures, set-points, etc.
- Enumerated values – e.g. on/off values, states, etc.
- Alarm events – history of alarm messages
- Date and time – e.g. calendars, time profiles, etc.
- Historical data – logged graph data

Key Features

- Stand-alone Web server requiring no additional services, such as IIS.
- Authentication provided by North's integrated token server or your own enterprise LDAP server (e.g. Active Directory).
- No knowledge of html authoring required. Pages created using our graphical editor.
- Dynamic content pages created using standards compliant xhtml and css, enabling their viewing from any Web browser.
- Automated pages give instant access to alarms, calendar, logs, page navigation and engineering.
- Integrates with North's calendar, providing a single location to adjust time profiles for control of interconnected systems.
- Integrates with North's central alarm store, allowing alarm events from multiple stores to be viewed and action to be taken.
- Integrates with North's data manager and user database, to display historical data values.
- Integrates with ObServer, to present real-time content from the many interconnected systems.
- Flexibility to expand pages using your own scripts.
- Library of objects available.

Engineering

Step 1 – Install OSM

The WebView OSM is installed automatically with all ObSys editions and is assigned to ObServer channel 61.

Step 2 – Configure WebView Server Module

Use object engineering software, such as ObView, to configure the Web Server module. Objects are available to enable the web server, enable the display of the calendar, select which alarm stores to connect to, enable data from the user database, enable logs, and select the authentication method.

By default, WebView is enabled and opens port 80 (HTTP) on all IP addresses available on the platform, including the localhost loopback address.

Step 3 – Configure the WebView Pages

Using the WebView Editor create pages to be displayed within a browser. For More information see the [WebView Editor v11 Window User Guide](#)

Step 4 – Test the WebView Pages

Using a Web browser access the WebView Pages and check to see the correct values are being returned.

Object Types

The following object types are available from the WebView server. The appearance of all objects may be customized, and their access controlled by specifying a privilege level.

Mimic

The Mimic object is used only in the template document, and defines where the page content is placed.

PageList

The PageList object is used only in the template document. It creates an automatic list of the pages available and is used to navigate between pages.

When authentication is used, any pages that the user does not have a sufficient privilege level to access to will not be displayed.

Pages are ordered alphabetically by default, or a display order can be specified when editing the page file.

Image

The Image object positions an image file resource on to page. The file may be located on the local or a remote server. Typically the image format supported by the browser includes GIF, JPG and PNG.

Jump

The Jump object is used to place a hyperlink, or navigational jump, on the page. The jump may link to another page on the server, a resource elsewhere on a different server, or trigger a script – for example, to refresh the page or navigate backwards, etc.

StaticText

The StaticText object places some fixed text on to the page.

RemoteText

The RemoteText object is used to display and adjust text values from a connected system.

RemoteArea

The RemoteArea object is used to display a percentage value as a colour area, frame or line.

RemoteEnum

The RemoteEnum object is used to display and adjust enumerated values from a connected system. The object is engineered to display an image or descriptive label for each of the possible enumerated values.

RemoteNumber

The RemoteNumber object is used to display and adjust numerical values from a connected system. The number can be displayed differently for various numerical ranges, e.g. change colour when above a set limit.

RemoteRange

The RemoteRange object combines features from both the RemoteNumber and RemoteEnum objects to display a numerical range of values from a connected system as a single state. The object is engineered to display an image or descriptive label for each of the possible numerical ranges.

RemoteWrite

The RemoteWrite object is used to place a hyperlink, button or click-area on the page that writes a pre-configured value to a connected system.

TimeProfile

The TimeProfile object is used to display and adjust on/off time schedules from a connected system.

UserObject

The UserObject is a custom object created using the objects above. Parameters may be passed to the object. WebView is supplied with a library of user objects.

DataLog

The DataLog object can be used to display the logged data available from both User Data and Data Manager (within the local ObSys) on a page.

Note: The logs available from both the user database and Data Manager are automatically added to the page list.

Calendar

The Calendar object displays the calendar from a CalTimer module on the page.

Note: The local calendar is automatically added to the page list.

AlarmList

The AlarmList object displays the alarms from an AlarmStore module on the page. Additional filter parameters such as the priority, sort order and alarm type may also be specified.

Note: The configured alarm list modules are automatically added to the page list

UserData

The UserData object displays a user database object on the page.

Note: The local user database module and its logs are automatically added to the page list.

Frame

A Frame object is used to place an embedded resource on the page, for example: an external web page, PDF document, Java applet, Flash animation, etc.

Authentication

WebView provides user authentication and access control to pages using one of the following methods:

- North TokenMax
- LDAP

North TokenMax

The North authentication server, TokenMax, provides centralized authentication for all extensible object model (XOM) compatible applications, such as Zip door access, ObView, Data Manager, Alarm Manager, etc.

For further information on the TokenMax module refer to the [OSM v20 TokenMax v20 Engineering Guide](#).

LDAP

In an enterprise environment, WebView Server can also use an LDAP server for authentication. LDAP is available as part of Microsoft Active Directory and many other servers.

WebView connects to the LDAP server to authenticate a user's password and logon credentials, and obtains their group names. WebView Server additionally maintains a privilege level for each group name.

For further information refer to your LDAP server documentation.

File Storage

Content

WebView serves content from the following folders:

\ObSys\WebView\<<sitename>\ .. stores files and graphics for website
\ObSys\WebView\auto\ .. used for core files required by WebView

Additionally from the Object List pages, help documents may be served from the folder:

\ObSys\TypeInfo\...\ObjSpec.pdf

Log Files

Web page usage is logged in ASCII text files using the NCSA format. Logging is enabled by creating the folder:

\ObSys\WebView\logs\

Log files are created daily and have the filename:

"nc"<+<yy>+<mm>+<dd>+".log"

For example, the log file on August 1, 2008 will be "nc080801.log".

Engineering Reference

Objects

When the OSM is loaded the following objects are created within ObServer, use object software to access these objects.

Object	Label	R/W	Type
O.WS	WebView Server configuration	-	[OSM v20\WebView v12]

IP Specification

The Web server opens TCP port 80 on all available IP addresses by default. However it may be configured to use a different port number and single IP address if required.

The Web server conforms to RFC 2616 for HTTP/1.1 and supports Basic Access Authentication to RFC 2617.

Notes

Revision History

Version	Build Date	Details
1.0	09/03/07	Application released
1.0	10/09/07	New object: RemoteArea Jump, RemoteNumber, RemoteEnum and RemoteText objects: new jump style parameter RemoteEnum object: new display as parameter User objects: improved user parameters Support for object tree in editor
1.1	01/06/08	First release as OSM platform. Generate alarm on user sign-in RemoteWrite object added Object errors now displayed as **?, full error reason given in mouse over Alarm list pages improved display and added compatibility with AlarmHistory OSM Server automatically restarts if IP address changes Page order/weight can now be specified in the page's properties. Weight/order of alarms and logs specified in the PageList object. Security removed from template, it is now part of the server setup parameters ObSys 2.0 compatibility – links with build-in modules such as UserData, removed DataManager file access
1.2	25/9/08	Added support for DataManager via objects (requires DataManager v1.1). Add new Log configuration object. Added option to add User Data pages under the modules label Improved error pages Fixed problem with alarm list re-synching after multiple changes.
1.2	1/11/08	Fixed timeout problem when sending large files to server from WebView Editor.
1.2	1/12/08	Improved drawing of logs Improved core HTTP handling Improved Jump object script so it can be used to generate special navigation buttons (back, fwd, reload, etc) Added RemoteRange object
1.2	29/1/09	Fixed problem with Data Manager logs in month and year view. Improved speed of drawing logs form Data Manager Can now link to wbo pages and pass parameters in URI Added option to disable the object list (useful when no security is used)
1.2	10/2/09	Fixed problem with the user data page always showing the adjust link regardless of security. Added the page /auto/signout which forces user logout from browser Added XML service, available on /data/ URI – see the document ' ObServer v20 XML Third-Party Applications Integration ' for details.