1 Introduction

1.1 Overview

This document describes the WEStation Application Program Interface (WESAPI). WESAPI is a client/server software package that contains SHC and SPD library routines. The client is a remote computer (not connected to a WDPF® Data Highway) that uses the library routines to request data from the Data Highway. The server is a WEStation directly connected to a WDPF Data Highway as illustrated below.

Access to the WDPF Data Highway is provided via a TCP/IP network connection to a WEStation acting as the WESAPI server.

![Figure 1-1. WESAPI Overview](image)

Figure 1-1. WESAPI Overview
1.2 Using the WESAPI NT Client License Manager

When the WESAPI NT Client is initially installed, it has a license that is valid for 10 days from the installation date. During the install, you were provided with a machine code for obtaining a normal or permanent license.

You need to display the WESAPI NT Client License Manager to review your specific machine code or to upgrade your license. The license manager is named LMStandalone.exe and is located in %WESAPI_HOME%\exe. There are two ways to invoke the License Manager:

1. Manually type in from a command prompt window:
   cd % WESAPI_HOME%\exe
   LMStandalone.

OR

2. From a "convenience" Start folder type in:
   Start->Programs->WESAPI NT Client->Client->License Manager.

Your machine code is displayed in a yellow-filled box. To update your license information:

1. Select the Product "WESAPI NT Client":
2. Press the Add License button and enter your new license Code
3. Press the Dismiss button.
4. Restart the "WESAPI NT Client."

Note

The server must be restarted before the new license information takes effect. To restart the WESAPI NT Client Service, use the "convenience" Start folder:

Start->Programs-> WESAPI NT Client->Client->Restart Server
1.3 Operation Modes

The WESAPI client shared memory database of WDPF point values and statuses are maintained by periodically updating the database with data received from the WESAPI server. Once configured and started on the client and server, WESAPI creates detached or daemon processes to send updates of the current value and status of WDPF points from the server to the client.

The frequency of the updates is defined as a configuration parameter on the server and on the client (refer to the appropriate Loading and Configuring information in Section 2 for instructions).

There are two modes of operation for the updates: mapping and subscription. The mode being used is determined by a configuration parameter on the client. A description of each mode follows.

1.3.1 Mapping Mode

In mapping mode, the updates from the server contain the current value and status for all 16,383 possible WDPF point System IDs. All 16,383 points are updated to the client (based on update frequency) regardless of whether the System ID is used by a WDPF point or is an unused System ID.

The first 10 bytes of each WDPF point record are included in the update from the server to the client (a total of 163,830 bytes per update). With some overhead and network message size restrictions, the update is broken into 42 messages of 4K each for a total of 168K bytes per update.

1.3.2 Subscription Mode

In subscription mode, the updates from the server contain the current value and status for only those WDPF points that have been requested by application programs running in the client.

When application programs in the client call the WESAPI library functions to request the current value and status of a WDPF point, the point is added to the list of points “subscribed to” by the client for updates.

During each update cycle, the server sends the current value and status of only the points that are in the client's subscription list. Each point update is a separate network message of 120 bytes.

To add a point to its subscription list, the client sends a message to the server to add a subscription point. The server adds the point to the client's subscription list in the server and immediately sends the value and status of the new point. The server also starts a subscription expiration timer for the point.
If the timer expires and there has not been a subscription renewal from the client, the server removes the point from the subscription list. When subscription renewals are received from the client, the timer is restarted. The timer is set to a time equal to 60 times the update frequency.

The client also keeps a list of points being subscribed to. This list is used by the client to periodically renew point subscriptions with the server. At a frequency of four times the update frequency, the client sends a subscription renewal message to the server for each point in its subscription list. When the client adds a new point to its subscription list, a usage timer for the point is also started. The usage timer is set to a time equal to 60 times the update frequency.

Each time an application program calls a WESAPI library function to get the current value and status of the point, a usage flag is set. Every time the timer expires, the flag is examined. If the flag is set, the usage timer is reloaded and the usage flag is cleared. Otherwise, application programs no longer use the point and it is removed from the list.

The combined effect of the usage timer in the client and the subscription renewal timer in the server is that after application programs in the client stop using a point, updates will continue to be sent from the server for a time period equal to 180 times the update frequency.

### 1.3.3 Choosing the Appropriate Update Mode

There are three resources to be considered when choosing the update mode for WESAPI:

- **Server CPU usage**
- **Client CPU usage**
- **Network usage**

The effect of the mapping mode on these resources will be a constant (independent of application programs in the client). The effect of the subscription mode on these resources will vary depending on the number of points being used by application programs in the client.

The WESAPI installer should try both modes, monitor the usage of the resources, and choose the mode best suited for the resource usage goals of the installation.

In general, the mapping mode will use less server and client CPU, but more network. The subscription mode will use more server and client CPU, but less network.

### 1.3.4 Monitoring Updates at the Client

In both the mapping and subscription modes of updating, the WESAPI software in the client starts an update timer on each point in its shared memory database of point values.
and statuses. Each time an update is received from the server for a point, its update timer is restarted.

If the timer expires, the status of the point in the client database is set to timed-out (bit 15 of the status word). The timer is set to a time equal to four times the update frequency.
1.4 Redundant WESAPI Servers

Redundant WESAPI server WEStations can be configured by using redundant Computation Servers that are using the hot standby form of redundancy (refer to the section “Configuring WEStation Drop Redundancy” in the manual “System Point Directory” (U0-8205)).

The primary Computation Server is the primary WESAPI server and the backup Computation Server is the backup WESAPI server. When a failover to the backup Computation Server occurs, the backup WESAPI server takes over the WESAPI functions and notifies the WESAPI client. Future WESAPI requests from the client are sent to the new WESAPI server. The WESAPI processes and the network communication with the WESAPI client are monitored in the primary Computation Server so that any failures will cause a failover to the backup.

1.5 Additional Reference Documents

Additional reference documents that may be helpful to WESAPI users are listed below.

<table>
<thead>
<tr>
<th>Document #</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U0-0131</td>
<td>Record Types User’s Guide</td>
<td>Describes WDPF point record types.</td>
</tr>
<tr>
<td>U0-2603</td>
<td>VXI GPM Handler User’s Guide</td>
<td>Describes Event Flag Tables and Bid Tables.</td>
</tr>
<tr>
<td>U0-2604</td>
<td>VXI Support Software Routines</td>
<td>Describes Analog Average Data Tables and Program Trigger Data Tables.</td>
</tr>
<tr>
<td>U0-8110</td>
<td>Operator WEStation Configuration Manual</td>
<td>Describes the use of GUls to configure the Operator WEStation</td>
</tr>
<tr>
<td>U0-8300</td>
<td>Defining and Configuring WEStations</td>
<td>Describes the use of GUls to define and configure WEStations.</td>
</tr>
<tr>
<td>U0-8002</td>
<td>WESAPI User’s Guide</td>
<td>Describes the WEStation Application Program Interface (WESAPI). WESAPI is a client/server software package that contains SHC and SPD library routines.</td>
</tr>
</tbody>
</table>
2 Getting Started

2.1 Configuring a WEStation WESAPI Server

Use the following procedure to configure the WEStation server software, in order for the WEStation to communicate with the WESAPI client:

1. Access the WEStation Init Tool (if necessary, refer to "Defining and Configuring WEStations" (U0-8300) for instructions).

   Add the WESAPI package to an existing drop type, or add a new drop type with the WESAPI package selected.

2. Configure one or more drops as the drop type with WESAPI package.

   Access the "WEStation Admin Tool" (if necessary, refer to "Defining and Configuring WEStations" (U0-8300) for instructions). Edit two files wesapi.env (Steps 3-10) and wesapi.ctl (Steps 11-16).

   If alarms are to be transferred to the client, configure the MMI Base Alarm package appropriately (Steps 17-22).

Edit wesapi.env File

3. Select the "Maintain Project Data" function from the Function pull-down menu in the Admin Tool main window.

4. Select the "WESAPI" filter from the "Filter" scrolling list.

5. Select the "WESAPI Environment Source File" topic from the "Topic" scrolling list.

6. Select the View/Edit File button. A text editor will appear with the sample wesapi.env file loaded.

   Note

   The path for the wesapi.env source file is wdpf/rel/data/wesapi/etc/wesapi.env.

7. Select the "Edit" function from the "Mode" pull-down menu.

8. Make the changes to the following wesapi.env file parameters:

   - WESAPI_CLIENT parameter (obsolete parameter)
     If this parameter exists in the file, it can be removed or commented out. The WESAPI server is no longer restricted to communicating with only one WESAPI client at a time.
- **WESAPI_MULTICAST_IF=westation1** (new parameter)
  The multicast interface parameter defines the network interface to be used by the WESAPI server to multicast messages to all WESAPI clients. The parameter specifies the host name of the WESAPI server WEStation on the desired network interface.

- **WESAPI_SUBD_PERIOD=1**
  Change 1 to the desired update frequency (in seconds) for the dynamic point data (this must agree with the WESAPI_SUBD_PERIOD parameter in the WESAPI configuration file on the client).

- **WESAPI_SUBD_USE_UDP=Y**
  Leave the line commented to use "TCP protocol" for the mapping mode update messages from the server to the client. TCP protocol provides a guaranteed delivery of data packets sent from the WEStation containing the dynamic WDPF point data. TCP requires more network overhead and time, but is more reliable.
  Uncomment the line to use "UDP protocol". UDP protocol provides a non-guaranteed delivery of data packets sent from the WEStation containing the dynamic WDPF point data. UDP protocol provides a faster transfer of data with less overhead.

9. Select the "Drops" pull-down menu to indicate which drops will use the file. A selection of "All" means that the file is applicable to all drops that contain the WESAPI software.

10. Select the "File Utilities" menu and select “Write to Disk.” This saves the changes.

**Edit wesapi.ctl File**

11. Select the "WESAPI Control Source File" topic from the "Topic" scrolling list.

12. Select the View/Edit File button. A text editor will appear with the sample wesapi.ctl file loaded.

**Note**

The path for the wesapi.ctl source file is

```
wdpf/rel/data/wesapi/etc/wesapi.ctl
```

13. Select the "Edit" function from the "Mode" pull-down menu.

14. Edit the file to define which WESAPI processes are to be included in this configuration of WESAPI. Specify the disable option for processes that are not to be included. Refer to the comment lines in the file for instructions.

15. Select the "Drops" pull-down menu to indicate which drops will use the file. A selection of "All" means that the file is applicable to all drops that contain the WESAPI software.

16. Select the "File Utilities" menu and select “Write to Disk.” This saves the changes.
Configure MMI Base Alarm Package

17. If alarms are to be transferred to the client (this is optional), use the Admin Tool to configure the MMI Base Alarm package.

   **Note**

   Verify that the software package "Alarm Configuration Files" has been installed on the WESAPI server WEStation (refer to “Using the Initialization Tool" in U0-8300 for instructions on installing software packages).

18. Select the "Define Software Configuration" function from the "Function" pull-down menu in the Admin Tool main window.

19. Select the "Operator filter" from the "Filter" scrolling list.

20. Select the "MMI Base Alarm Configuration" topic from the "Topic" scrolling list. A configuration window will appear.

21. Select the "Alarms Historical Storage" option and perform the following:

   - Select the "Send Alarm Message to HSR" setting.
   - Enter the drop name of the WEStation that will function as the WESAPI server in the "Primary HSR Drop Name" entry field.
   - Leave the "Secondary HSR Drop Name" entry field blank.

22. Reboot the WESAPI server WEStation.
2.2 Configuring an NT WESAPI Client

Configuring the WESAPI client is a two-step process. First, the wesapi.env (located in the %WESAPI_HOME% directory) must be edited and then the WESAPI client system service must be restarted for the changes made in wesapi.env to take effect.

These steps can be performed manually:

1. cd %WESAPI_HOME$
2. notepad wesapi.env
3. Bring up the Services Applet in the Control Panel:
   Start->Settings->Control Panel
4. Stop and then start the "Wesapi Service".

These steps can be performed using "convienence" folders in the Start Folder:

(This brings up the wesapi.env file in Notepad)

Start->Programs->Wesapi NT Client->Client->Configure

(This re-starts the WESAPI Service)

Start->Programs->Wesapi NT Client->Client->Restart Client

Using the "convienence" folders is recommended, since the WESAPI service could have several dependency services (depending upon how it is being used) that in turn are required to be stopped and then restarted. The "Restart Client" folder handles all of this for you.

Below is a summary of the environment variables that need to be reviewed/ modified in the wesapi.env file:

WESAPI_USE_OPC - Indicates if WESAPI is being used (integrated) with WDPF OPC Server.

WESAPI_PDIR - This point directory is used by the SPD library on the server.

WESAPI_SUBD_PERIOD - Period in seconds for updates of the dynamic point database.

WESAPI_HIGHWAY - Current highway (network) being used.

WESAPI_MAP_NETWORK_N (n=0 - 16) - Highways which are being mapped.
2.3 Verifying Operation of an NT WESAPI Client

The overall operation of the WESAPI client/server package can be verified by using the sample programs, SHC_demo and SPD_demo, which are provided with the WESAPI Client software.

The executable files for these sample programs are located in the %WESAPI_HOME%\exe directory. There are two ways to execute these programs:

1. Manually input from a command prompt:
   
   cd %WESAPI_HOME%\exe
   
   execute the programs
   
   c:\wesapi\exe> SHC_demo
   
   c:\wesapi\exe> SPD_demo
   
   OR

2. Use the "convienence" Start folders:
   
   Start->Programs->Wesapi NT Client->Utilites->SHC_demo
   
   Start->Programs->Wesapi NT Client->SPD_demo

At the command prompt in each program, enter a "?" to display a list of valid commands.

To verify access from the client to the WDPF System Point Directory, use commands 1 and 2 of the SPD_DEMO program to retrieve SPD information for WDPF points.

To verify access from the client to WDPF point values and statuses, use commands 2, 3, 4, 5 and 6 of the SHC_demo program to retrieve the current value and status for WDPF points.

To verify access from the client to WDPF point record fields other than values and statuses, use commands 12, 13, 14, 15, 16 and 17 of the SHC_demo program to retrieve point record attributes for WDPF points.

To verify access from the client to WDPF functions not related to WDPF points, use commands 26, 37 and 39 of the SHC_demo program to retrieve the highway mode, highway time and a list of highway drops.

If there are any problems with the operations, make sure the WESAPI client configuration (in wesapi.env) is correct and invoke the Services Applet from the Control Panel and verify that the "WESAPI Service" status is "Started".

Another useful tool for checking out the client operation is discussed in Section 2-4. Monitoring Operation of WESAPI at the NT Client.
2.4 Monitoring Operation of WESAPI

2.4.1 At the Server

When the WESAPI server software is started on the server WEStation, daemon processes are created to communicate with the WESAPI client(s). There can be from five to eight daemon processes (depending upon the WESAPI server configuration in the wesapi.ctl file).

To check the current running status of the configured WESAPI server daemons, use the WESAPI program in the /usr/wdpf/wesapi/bin directory. Enter the following in a shell tool:

```
cd /usr/wdpf/wesapi/bin
wesapi
```

The WESAPI program displays a list of the configured daemons and their current running or stopped status.

If a configured daemon has stopped, the daemon’s log file in the /usr/wesapi/tmp directory can be checked for error messages. Daemon processes can be restarted by using the WESAPI program with the restart option:

```
cd /usr/wdpf/wesapi/bin
wesapi restart
```

2.4.2 Monitoring Operation of WESAPI at the NT Client

When the computer is booted, the WESAPI NT Client is executed as a system service. So, the only information available from the Service Applet on the Control Panel is whether or not the "Wesapi Service" is started or is stopped. A stopped service indicates the service had problems starting. This WESAPI NT Client service should always reflect a "started" status.

There is a log file named wesapi.log, which resides in %WESAPI_HOME%. If the "WESAPI Service" is "stopped", which is indicated by a "blank" status in the Service Applet on the Control Panel, you can review wesapi.log file to see what caused the failure. You will also note log file names wesapi_ERR.log and wesapi_WARN.log, etc, which collect only those specific status level of information. The wesapi.log file contains ALL levels of statuses for the WESAPI Client.

If it is determined that the "WESAPI Service" is "Started," then the statistics tool can be executed to see how WESAPI client is performing. The name of this tool is wesapi_show.exe and it is located in %WESAPI_HOME%\exe. There are two ways to execute this tool:
1. Manually from a command prompt:
   `cd %WESAPI_HOME%\exe`
   `wesapi_show -10`

   **Note**

   The -10 means show me 10 intervals of collection.
   You can change this to be any number, such as -3, or -50. If you provide no parameter, it is executed for the next upcoming interval only.

   **OR**

2. Using the "convenience" folder:
   **Start->Programs->Wesapi NT Client->Utilities->Statistics**
   10 intervals of information are always shown.

This tool displays the following information about highways that are broadcasting:

- The Network Highway number and whether or not it is the WESAPI_HIGHWAY defined in the wesapi.env file.
- The Mode the network is operating in: Mapping or Subscription
- The IP Address of the WESAPI Server
- The number of UDP and TCP and Server Announcement packets received.
- The number of Highway Image Packets Received (mapping mode)
- The number of Subscription Point Packets Received (Subscription Mode)
- The number of Packet Sequence Errors
- The "XFer Time", or the amount of time into the interval it took for the packets to be received. This should not approach the WESAPI_SUBD_PERIOD (defined in wesapi.env). If it does, then you may have to increase the value of WESAPI_SUBD_PERIOD to improve performance and ensure data is being received properly.

If there are no WESAPI Servers currently broadcasting, this will be reflected by this tool. You should then go to the WESAPI server and investigate why the WESAPI server is not running.
2.5 Troubleshooting

If the WESAPI Server and Client are not communicating (as determined by the failure of the SPD_DEMO and SHC_DEMO programs discussed in 2-3), the following steps can be used to troubleshoot the problem:

Verify that the WESAPI Server is Issuing the Multicast Messages

1. Become superuser:
   
   $ su

2. With the WESAPI server running, perform a snoop command on the interface connected to the WESAPI clients (for example, le0):
   
   # snoop –d le0 232.232.232.2

   Using device /dev/le (promiscuous mode)

   Drop208> 232.232.232.2 UPD D=2200 ...

3. If you see messages flowing from the interface to the default multicast address (232.232.232.2) on port 2200, then the server is behaving properly. If you do not see the messages, try looking on the other interfaces as well.

4. If the multicasts are not being sent, verify that the desired interface has multicasting enabled:
   
   # ifconfig –a

   lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST>

   le0: flags=863<UP,BROADCAST,NOTRAILERS,MULTICAST>

5. If the multicast messages are flowing out an incorrect interface, verify that the WESAPI_MULTICAST_IF parameter in the /usr/wdpf/wesapi/etc/wesapi.env file is set to the proper interface (note that the value should correspond to the host name associated with that interface)