

# Advanced Display Technologies on HP-UX Workstations

Todd M. Spencer

Paul M. Anderson

David J. Sweetser

Multiple monitors can be configured as a contiguous viewing space to provide more screen space so that users can see most, if not all, of their applications without any special window manipulations.

**I**n today's computing environment, screen space is at a premium. The entire screen can be easily consumed when primary work-specific applications are used together with browsers, schedulers, mailers, and editors. This forces the user to continuously shuffle windows, which is both distracting and unproductive.

The advanced display technologies described here allow users to increase productivity by reducing the time spent manipulating windows. Three technologies are discussed:

- Multiscreen
- Single logical screen (SLS)
- SLSclone.

Implementation details and procedures for configuring HP-UX workstations to use the SLS technology are described in references 1 and 2.

## Multiscreen

When considering the problem of limited screen space, one solution that comes to mind is to use a bigger monitor with a higher resolution.

Unfortunately, it is often impractical to add a monitor with a resolution high enough to accommodate all the data a user wants to view. Although demand has increased for monitors of higher resolution, such as 2K by 2K pixels, they are still too expensive for companies to place on every desktop. In addition, these

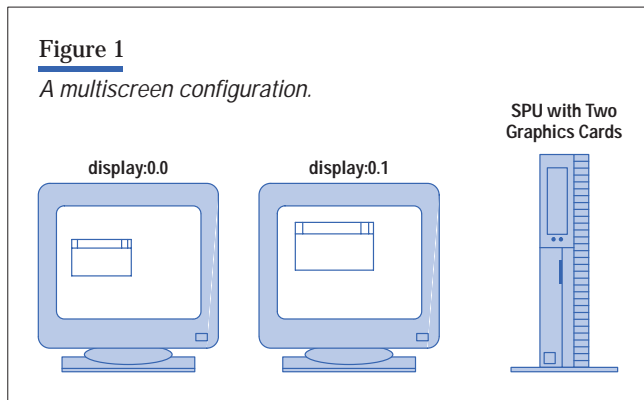
large monitors are cumbersome and heavy. There are also safety considerations: the monitor must be stable and properly supported.

A more practical, cost-effective solution is to use additional standalone monitors to increase the amount of visible screen space. The X Window System (X11) standard incorporates a feature known as multiscreen, which provides this type of environment. In multiscreen configurations, a single X server is used to control more than one graphics device and monitor simultaneously. These types of configurations are only possible on systems containing multiple graphics devices.

In these multiscreen scenarios, a single mouse and keyboard are shared between screens. This allows the pointer but not the windows to move between screens. Each application must be directed to a specific screen to display its windows. This is done by either using the `-display` command line argument or by setting the `DISPLAY` environment variable.

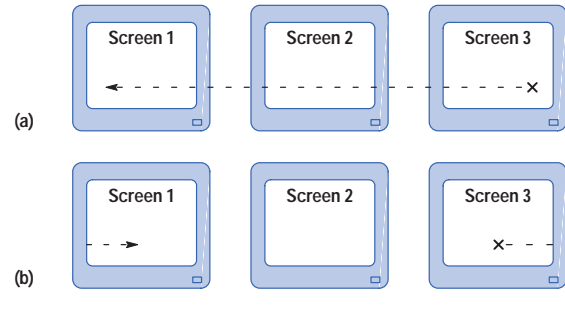
**Figure 1** shows a two-monitor multiscreen configuration. Both monitors are connected to the same workstation and are controlled by the same X server. This type of configuration effectively doubles the visible workspace. For example, users could have their alternate applications, such as web browsers, mailers, and schedulers on the left-hand monitor and their primary applications on the right-hand monitor. Since the X server controls both screens, the pointer can move between screens and be used with any application.

Multiscreen offers the advantage that it will work with any graphics device. There are no constraints that the graphics devices be identical or have the same properties.



**Figure 2**

*Cursor wraparound in a multiscreen configuration.*



For example, on an HP 9000 Model 715 workstation containing an HCRX24 display (a 24-plane device) and an internal color graphics display (an 8-plane device), the user can still create a multiscreen configuration. Of course, those applications directed to the HCRX24 will have access to 24 planes while those contained on the other are limited to 8 planes. Currently, the HP-UX X server allows a maximum of four graphics devices to be used in a multiscreen configuration.

The HP-UX X server also provides several enhancements to simplify the use of a multiscreen configuration. If a user has a 1-by-3 configuration (**Figure 2a**), there may be a need to move the pointer from screen 3 to screen 1. This requires moving the pointer from screen 3 to screen 2 to screen 1. By specifying an X server configuration option, the user can move the pointer off the right edge of screen 3, and the pointer will wrap to screen 1 (**Figure 2b**). The same screen wrapping functionality can be provided if the user has configured the screens in a column. Finally, a 2-by-2 configuration can contain both horizontal and vertical screen wrapping.

Although multiscreen is convenient, it has shortcomings. Namely, the monitors function as separate entities, rather than as a contiguous space. The different screens within a multiscreen configuration cannot communicate with one another with respect to window placement. This means that windows cannot be moved between monitors. Once a window is created, it is bound to the monitor where it is created. Although some third-party solutions are available to help alleviate this problem, they are costly, inconvenient (sometimes requiring the application to make code changes), and lack performance.

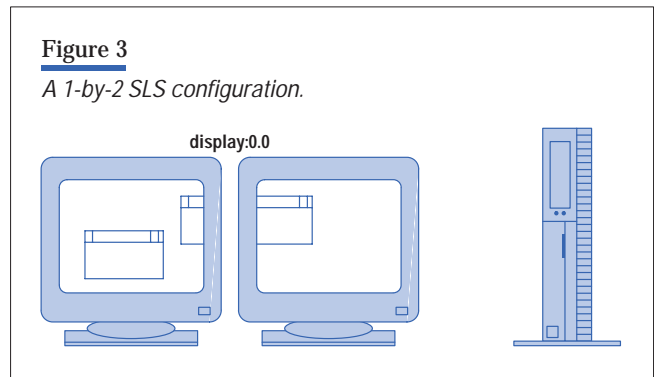
The lack of communication between screens with respect to window placement forces users to direct their applications towards a specific screen at application start time. After a screen has been selected all additional subwindows will be confined to that screen. With today's larger applications, it is possible to find that certain screens still get overcrowded, resulting in the original predicament of having to iconify and raise windows.

#### Single Logical Screen

To remedy the shortfall of the multiscreen configuration, HP developed a technology called *single logical screen* (SLS).<sup>3</sup> SLS has been incorporated into the HP standard X server product and allows multiple monitors to act as a single, larger, contiguous screen. As a result, windows can move across physical screen boundaries, and they can span more than one physical monitor. In addition, SLS functionality has been implemented in an application-transparent manner. This means that any application currently running on HP-UX workstations will run, without modification, under SLS. Therefore, SLS is not an API that application writers need to program to or that an application needs to be aware of. The application simply sees a large screen. This ease-of-use lets end users take advantage of a large workspace without requiring applications to be rewritten or recompiled.

Many of electronic design automation (EDA) and computer-aided design applications can benefit from SLS. Some of these applications, by themselves, can easily occupy an entire screen while only showing a fraction of the desired information. For example, with more screen real estate, an EDA application can simultaneously display waveforms, schematics, editors, and other data without having any of this information obscured. To do this on a workstation with only a single monitor would require displaying the waveforms, schematics, and other items in such small areas as to be unreadable.

On HP-UX Workstations, a single logical screen actually represents a collection of homogeneous graphics devices whose output has been combined into a single screen. **Figure 3**, shows an example of a 1-by-2 SLS configuration. Most HP-UX workstations are not limited to only two graphics devices. Some models support up to four devices. When using these graphics devices to create an SLS environment, any rectangular configuration is allowed.

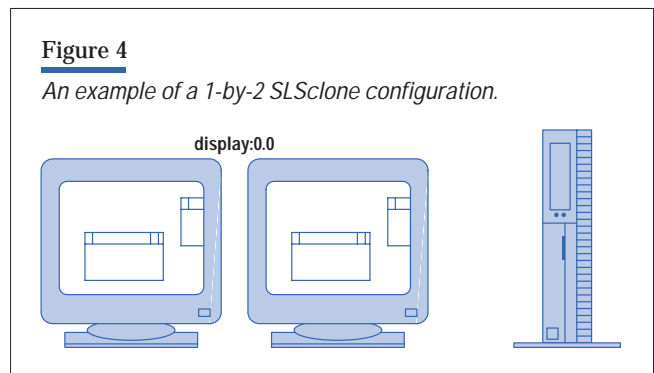


#### SLSclone

SLSclone is similar to the SLS configuration. The difference is that the contents from a selected monitor are replicated on all other monitors in the configuration (see **Figure 4**). A user can dynamically switch between SLS and SLSclone using an applet being shipped with the HP-UX 10.20 patch PHSS\_12462 or later.

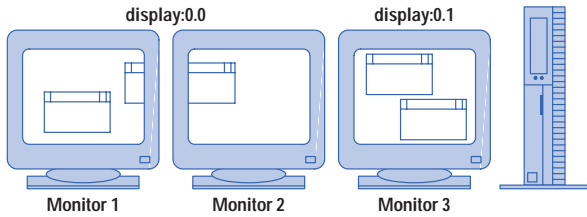
This functionality is useful in an educational or instructional environment. Instead of crowding many users around a single monitor to view its contents, SLSclone can be used to pipe these contents to neighboring monitors. As with SLS, SLSclone currently supports up to four physical monitors, depending on the workstation model.

SLSclone functionality easily lends itself to a collaborative work environment. If additional people enter a user's office to debug some software source code, for example, the user can quickly switch the SLS configuration into an SLSclone configuration, and the debugging screen will be displayed on all monitors. Also, the additional monitor can easily be adjusted to the correct height and tilt without affecting the original user's view of the display.



**Figure 5**

A hybrid configuration consisting of a 1-by-2 SLS with multi-screen.



### SLS and Multiscreen

Even with the benefits of SLS, there may be cases in which a user will want to use SLS and multiscreen at the same time. For example, a user could have a 1-by-2 SLS configuration acting as one screen, and a third monitor acting as a second screen. A depiction of this is shown in **Figure 5**.

In this type of configuration, a user can move windows between physical monitors 1 and 2 but not drag a window from monitor 2 to monitor 3. The pointer, however, can move between all monitors. This type of hybrid configuration can be useful in a software development environment. All of the necessary editors, compilers, and debuggers can be used on monitors 1 and 2, and the application can be run and tested on monitor 3.

If a workstation supports four graphics devices, another possible hybrid configuration is to use two screens, each of which consists of a two-screen SLS configuration (**Figure 6**).

In this configuration, windows can be moved between monitors 1 and 2 or between monitors 3 and 4. However, a window cannot be moved between monitors 2 and 3. As

with all multiscreen configurations, the pointer can move across all four monitors. These two screens could also be placed vertically, resulting in a 2-by-2 monitor arrangement and a 2-by-1 multiscreen configuration.

### Conclusion

Advanced display configurations can be used to increase productivity. The increase in screen space facilitates collaboration and communication of information. We have also found that these configurations are very useful for independent software vendors (ISVs) who demonstrate their applications on HP-UX workstations. They appreciate the additional screen space because they are able to display more information and rapidly describe their products without losing their customers' attention.

Finally, the configuration of an advanced display is accomplished in an easy and straightforward manner through the HP-UX System Administration Manager (SAM). Additional information on advanced display configurations and other exciting X server features are available at:

<http://www.hp.com/go/xwindow>

HP-UX Release 10.20 and later and HP-UX 11.00 and later (in both 32- and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

UNIX is a registered trademark of The Open Group.

### References

1. T. Spencer and P. Anderson, "Implementation of Advanced Display Technologies on HP-UX Workstations," *Hewlett-Packard Journal*, Vol. 49, no. 2, May 1998 (available online only).  
<http://www.hp.com/hpj/98may/ma98a7a.htm>
2. R. MacDonald, "Hewlett-Packard's Approach to Dynamic Loading within the X Server," *Hewlett-Packard Journal*, Vol. 49, no. 2, May 1998 (available online only).  
<http://www.hp.com/hpj/98may/ma98a7b.htm>
3. M. Allison, P. Anderson, and J. Walls, "Single Logical Screen," *InterWorks '97 Proceedings*, April 1997, pp. 366 - 376.

**Figure 6**

Two 1-by-2 SLS configurations combined via multiscreen.





**Todd M. Spencer**

A software engineer at the HP Workstation Systems Division, Todd

Spencer was responsible for development of the the SAM component that allows users to set up multiscreen and single logical screen configurations. He came to HP in 1989 after receiving an MS degree in computer science from the University of Utah. Todd was born in Utah, is married and has four children. His outside interests include tropical fish, camping, woodworking, piano (playing classical music), and jogging.



**Paul M. Anderson**

Paul Anderson is a software engineer at the HP Workstation Systems Di-

vision. He joined HP in 1996 after receiving a BS degree in computer science from the University of Minnesota. He is currently working on device drivers for new peripheral technologies. His professional interests include I/O drivers, operating systems, and networking. Paul was born in Edina, Minnesota. His outside interests include hiking, music, and mountain biking.



**David J. Sweetser**

With HP since 1977, David Sweetser is a project manager at the

HP Workstation Systems Division. He is responsible for the X server and some of the client-side X libraries. He received a BSEE degree and an MSEE degree from Harvey Mudd College in 1975. He was born in Woodland, California, is married and has two children. His outside interests include mountain biking, hiking, snowshoeing, cross-country skiing, and white-water rafting.

- 
- ▶ [Go to Subarticle 7a](#)
  - ▶ [Go to Subarticle 7b](#)
  - ▶ [Go to Next Article](#)
  - ▶ [Go to Journal Home Page](#)