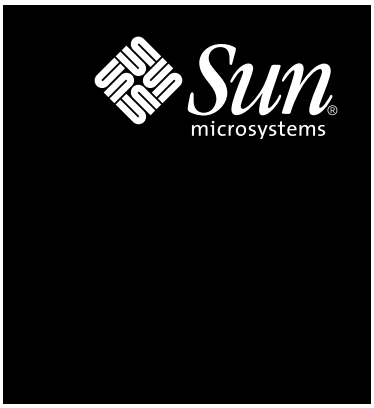
# GNOME 2.0 Desktop: Developing with the Accessibility Framework

Technical White Paper



# Table of Contents

Chapter 1

# Introduction

***Moving to GNOME 2.0***

*GNOME 2.0 provides many key features and benefits to computer users. Here are some of the most important:*

- ***Unified Desktop*** *— Runs on the Solaris Operating Environment, GNU/Linux, and many other UNIX system-based platforms, allowing users to work more productively in heterogeneous computing environments.*
- ***Powerful Yet Easy to Use*** *— Provides an intuitive, easy-to-use desktop with an attractive user interface, which can reduce learning time and costs. It also provides a Control Center and a consistent configuration management system to make it easy to personalize the system to suit individual preferences.*
- ***Built-in Accessibility*** *— People with disabilities can use the desktop productively because of integrated accessible applications and assistive technology solutions.*
- ***Flexible*** *— Runs CDE/Motif– and Java™ technology-based applications without modification, thus preserving existing software investments.*
- ***Open Standards*** *— Embraces a broad range of industry standards, facilitating interoperability, Internet communication, and seamless data interchange.*
- ***Mainstream Applications*** *— Comes with a wealth of productivity applications and utilities, helping improve desktop productivity.*

GNOME 2.0 is an advanced new user desktop that Sun Microsystems plans to provide and support for the Solaris™ Operating Environment. It is an open source, free software project that runs on most UNIX® and GNU/Linux platforms. GNOME gained prominence because of an intuitive interface that users can depend on to simplify interaction with the computer software and information they use every day.

The GNOME desktop was designed with the user in mind — it is very easy-to-use and highly personalizable, allowing individuals to tailor its behavior to suit their specific needs. This includes users with physical disabilities such as poor vision, blindness, deafness, or impaired motor skills — GNOME is built around an architecture that allows for the easy creation of accessible solutions that break down the barriers these users often face when using computer technology. In this paper we describe:

- Challenges faced by users with disabilities;
- Why it is important to provide accessibility solutions in the workspace for these users; and
- Programming interfaces GNOME provides to help developers rapidly create and deploy accessible solutions

Computers and other information technology are the catalyst of independence for many people with disabilities, provided these technologies are made accessible. Although a few progressive software and hardware developers have provided acceptable solutions, progress has been slow overall until recently. New legislation, including Section 508 of the U.S. Rehabilitation Act, which requires that information technology be accessible in order to fulfill procurement guidelines, has helped quicken the pace. The challenge for developers, who may have little or no knowledge of

accessibility issues or of the community of people with disabilities, is learning how to efficiently design and deploy accessible solutions that meet these requirements.

The problems relating to providing accessible information technology are diverse in scope. Far too many software applications and operating systems are not yet accessible, and frequent software upgrades often cause those accessible solutions that do exist to break down. In most cases, an assistive technology vendor—creating, say, a screen magnifier—must write custom solutions specifically designed for each application of interest. However, when these applications are upgraded, the assistive technology solutions usually must also be upgraded, at significant expense, in order to maintain compatibility with the applications. The bottom line is that people with disabilities are often denied convenient use of computers and information technology, while developers are under continuous and significant pressure to upgrade their products as they attempt to track moving targets. For people with disabilities, the results can be devastating—denial of employment or upward mobility, scholastic failure, and general disenfranchisement.

GNOME 2.0 has been carefully architected to avoid many of the pitfalls that have plagued developers of accessible solutions in the past, giving them an opportunity to economically develop and maintain their products. GNOME delivers several benefits to users with disabilities, assistive technology developers, and application software vendors relating to the efficient creation and delivery of accessible solutions:

- A robust and reliable platform for writing accessible applications and interfacing assistive technologies—includes a comprehensive accessibility framework and software bridges for integrating standard GNOME applications as well as Java technology-based applications.
- Accessible solutions enabled by the GNOME 2.0 desktop can result in:
    - Compliance with Section 508 of the U.S. Rehabilitation Act, thus opening up large new markets for developers
    - Lower total cost of ownership for businesses who support users with disabilities.
    - Expanding the pool of talented workers available to businesses
- Built-in accessibility solutions that provide an accessible desktop, productivity applications, utilities, and accessories.
- Integrated assistive technologies that work with desktop software, including a screen reader, screen magnifier, and on-screen keyboard for users with visual, hearing, and other physical disabilities.
- Accessibility solutions are also available across most UNIX and GNU/Linux platforms running GNOME 2.0, reducing the need for specialized or customized software and hardware, and helping keep administration costs low.

Chapter 2

# Challenges for the Community of People with Disabilities

***Over 40 million people with disabilities in the U.S.***

*According to the United States Census figures for 2000 (see http://www.census.gov/ hhes/www/ disable/census/disapick.html) approximately 40 million people in the United States have some form of permanent disability. (The definition of disability is based on the one found in the Americans with Disabilities Act (ADA) and represents any permanent condition that seriously impairs a significant life function such as seeing, hearing, moving, speaking, or learning.) This figure represents a significant fraction of the U.S. population, and the numbers for the world as a whole are obviously much higher. Clearly, the community of people with disabilities is a market of substantial size and scope. If we are to fully tap the potential of individuals with disabilities, we must leverage the benefits of accessible information technology and the Internet as well as create platforms where assistive technology can flourish.*

When one acquires a physical, sensory, learning, or other disability, more than just a physical or sensory capability is forfeited. It is necessary to begin functioning in a world where many aspects of daily life have changed radically. The barriers to independence are daunting.

Consider someone who has sustained an injury that results in the loss of functional vision. The daily newspaper, once part of her morning ritual, is now unreadable. She can no longer read the numerals on her digital alarm clock, and cannot set the time or alarm without assistance. She can't read her e-mail, faxes, or office correspondence without sighted assistance. The faces of loved ones recede into the distance, and many objects of daily life become impossible to operate independently because of poor design that assumes that all customers have equal abilities.

The barriers faced by users with disabilities hinder gainful employment, education, and independent living. If you can't use a telephone, then your activities are severely restricted because even basic communication is difficult. If you can't operate a personal computer, holding a steady job or attending university is challenging, perhaps impossible. If you want to surf the world wide Web, and you can't read the contents of a web page, you are cut off from online commerce, basic information, and even social interaction.

Inaccessibility also forces you to relinquish your privacy because you must rely on others to handle your personal matters, such as reading mail, writing checks, reading food containers, and so on. In brief, inaccessibility takes away personal independence and freedom.

The barriers faced by people with disabilities are, to a large degree, the direct result of products and services that are not designed with accessibility concerns in mind. To lower the barriers, it is necessary to add support for assistive technologies. Not only does this help a disadvantaged segment of the community, it increases the potential audience for the products and services.

***Legislation Drives Accessible Solutions***

*Accessibility support is now mandated in some markets in the United States, Canada, and other countries. Vendors who provide accessible products possess a definite advantage when bidding on contracts involving these markets.*

*For example, in the United States, Section 508 of the Rehabilitation Act requires the federal government to procure information technology that is accessible to people with disabilities. The Americans with Disabilities Act (ADA), while not directly mandating procurement of assistive technology, has a direct positive impact on the acquisition of accessible information technology and assistive technology solutions. This is because the ADA has increased awareness in the general public about the importance of accessibility.*

*In Europe, the Disability Discrimination Act in the United Kingdom is having the same positive effect on the acquisition of accessible information technology.*

Chapter 3

# Overcoming Barriers to Accessibility in the Computer World

***Accessible Products Benefit Everyone***

*Accessible products and services clearly benefit people with disabilities, and foster employment, education, and independent living for them. However, products designed for accessibility invariably have benefits that can be enjoyed by the community as a whole. Consider the familiar example of the sidewalk curbcut. Although they were initially designed for the convenience of people in wheelchairs, curbcuts are now valued by all members of the general public who use wheeled vehicles such as bicycles, baby strollers, and delivery carts.*

*In the world of computer technology there are numerous examples of "electronic curbcuts": screen readers that provide voice output, magnification software that enlarges text and graphics, and voice recognition programs that accept verbal commands. There technologies are very important to users with disabilities, however they provide significant benefits to general users as well.*

To overcome barriers to accessibility in the computer world, it is critical that you, the software developer, take the time to design applications that properly address the needs of people with disabilities. Doing so will not only allow you to reach more customers, it will also result in software with useful new features that benefit your entire user base.

For some computer platforms, developing accessible software solutions is extremely difficult and very expensive because a consistent, system-wide accessibility architecture is nowhere to be found. Creating an accessible solution often requires low-level system patching, special-casing for different applications, and other techniques that make the solution very fragile and sensitive to the slightest changes in the operating system or applications.

GNOME 2.0 is very different. It has been designed with accessibility considerations in mind and provides a robust framework that makes it much easier for accessible applications to be created. It also provides a standard interface for integrating assistive technologies such as screen readers and screen magnifiers. Notably, the fragility of accessible solutions is largely eliminated when deployed for the GNOME desktop. With GNOME's accessibility architecture, an assistive technology and an application have no direct dependencies on one another. This means, for example, that if an application is updated, the assistive technology does not necessarily have to be updated as well—and vice versa.

The GNOME 2.0 platform includes several built-in solutions designed to assist users with disabilities. In addition to a core set of accessible applications and utilities provided with the desktop, the following two assistive technologies are available:

GNOPERNICUS SCREEN READER & MAGNIFIER. Gnopernicus is designed for people who are blind or who have low vision. It combines the functions of a screen reader and a screen magnifier in a single solution. The reader can follow the focus of an accessible application and generate synthesized speech, using the FreeTTS text-to-speech engine, or send output to a refreshable Braille display or DECtalk synthesizer (available from third parties). The magnifier enlarges the area of the screen around the area of focus and automatically updates the display as the focus changes.

GNOME ON-SCREEN KEYBOARD. GOK, the GNOME on-screen keyboard, is for users who have difficulty using a standard keyboard or mouse, and is commonly referred to as an alternative input system. It supports several input devices such as switches manipulated by eye movements, head mounted pointing systems, and sip-and-puff switches, which can be used to select characters from a graphical representation of the keyboard on the computer monitor.

Accessible applications you create for the GNOME 2.0 platform will work with these assistive technologies without you having to know anything about how they work. All that is important is that they are integrated using the GNOME accessibility framework. The technologies can also be used with Java technology-based applications that use the Java Foundation Classes (Swing).

Chapter 4

# GNOME 2.0 Accessibility Architecture

GNOME 2.0 is an excellent platform on which to design and build accessible software applications and to integrate the assistive technologies users with disabilities need in order to interact with these applications.

GNOME's accessibility architecture, much of it modeled after Sun's pioneering work for the Java platform, has been carefully crafted to allow applications that use standard user interface widgets (provided by GNOME's GTK+ widget toolkit) to inherit a considerable amount of accessible behavior for free. Only a modest amount of additional engineering work, to avoid common pitfalls such as not providing accelerator keys for menu items or not providing textual descriptions of important visual elements, is needed to complete an application. Even applications using custom widgets can be made accessible using the facilities of the GNOME Accessibility Toolkit (ATK), which allows applications to expose their properties and attributes to assistive technologies in a consistent, well-defined manner.

The architecture for integrating assistive technology into the system is also compelling. GNOME 2.0 provides a standard interface, called the Assistive Technology Service Provider Interface (AT-SPI), that brokers all communication between the AT and running GNOME applications. In addition, a bridge is provided between the AT-SPI and Java technology-based applications that use Swing user interface components so that the very same AT can be used with these applications as well.

Below are brief descriptions of GTK+, ATK, and AT-SPI, and the pieces they provide for completing the accessibility framework. The definitive reference on GNOME's accessibility architecture is `located` at the Web site for the GNOME Accessibility Project. See http://developer.gnome.org/projects/gap/.

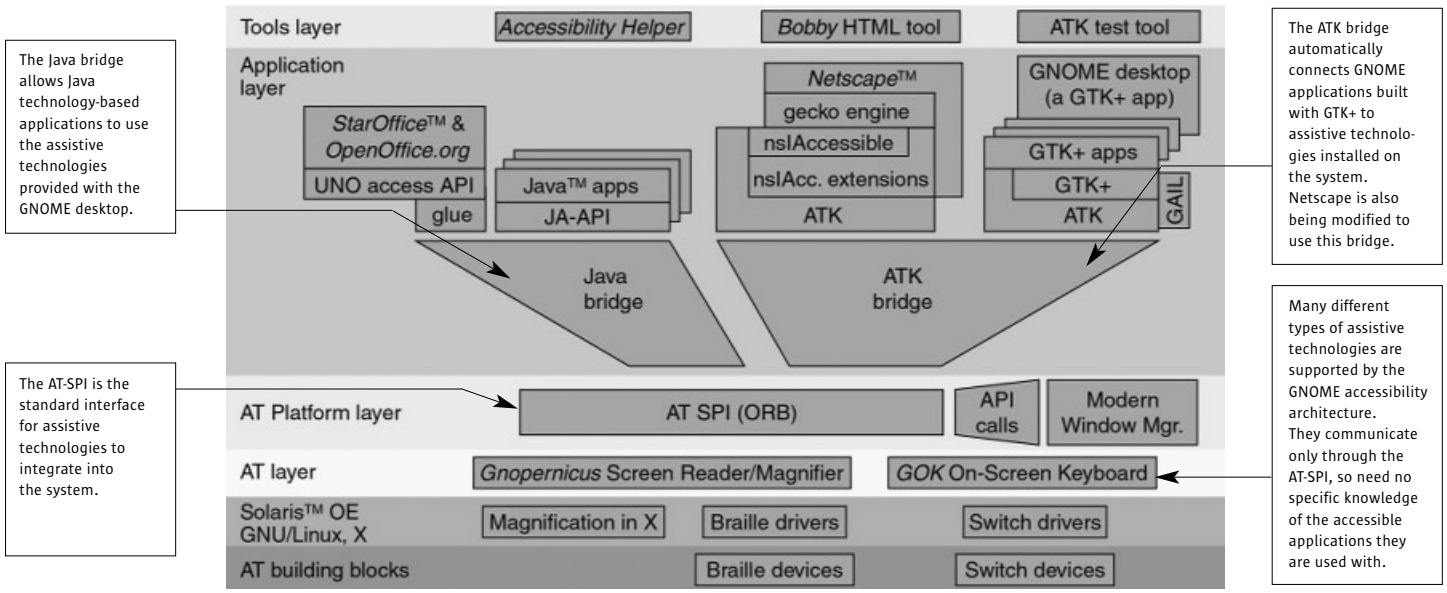



**Figure 1:** The GNOME accessibility architecture

## THE GTK+ TOOLKIT

GNOME's primary widget toolkit, GTK+, is made up of a comprehensive set of user interface widgets - check boxes, radio buttons, combo boxes, editable text fields, tables, and many more. (On the Microsoft Windows platform, objects such as these are commonly referred to as controls. Under the Java programming language, they are called components.) Most applications designed for the GNOME desktop use this standard set of widgets in the windows and dialog boxes they present to users, thus ensuring users experience a consistent look and feel not only within a single application, but also across the entire desktop.

For GNOME 2.0, the GTK+ widgets share an important characteristic—they all support GNOME's accessibility interface, an interface that makes it possible for the state of a widget to be probed and queried at run time so that information an assistive technology requires can be retrieved.

Since accessibility support has already been integrated into GTK+, if you build an application around the GTK+ widgets, no "heavy lifting" is required to provide basic accessible functionality. It is provided automatically.

Of course, there are other design guidelines that you should follow in order to ensure an application has a user interface that is well suited for use by people with disabilities. These guidelines are not particularly onerous since they embody good programming design concepts anyway. Some examples:

Applications should be fully accessible from the keyboard by having the tab key move the focus from one widget to the next in a logical order and by having accelerator keys for all the menu selections.

- Frequently used items should not be buried deep in menu trees.
- Attributes such as colors and font sizes should not be hard coded in the program.
- Information that is to be presented audibly should be presented visually at the same time.

## The Accessibility Toolkit (ATK)

It is the ATK that defines the accessibility interface that is implemented by standard GTK+ widgets so that accessibility information can be stored and retrieved in a consistent manner throughout the system. (The implementation is in a library called GAIL - the GNOME Accessibility Implementation Library.) As you would expect, this interface defines the precise list of functions that must be made available in order to support accessible behavior. The accessibility interface defined by ATK is designed to be compatibly extensible to support applications and assistive technologies as they evolve.

The accessibility functions permit assistive technology developers to extract important information from GNOME 2.0 widgets. ATK contains text-related functions, screen-coordinate information, image-related functions, selection-related functions, text-related functions, table-related functions, and value-related functions.

In simple terms, these functions provide a standard technique for assistive technologies to extract vital information from a running application, such as the name of the widget with focus, the state of the widget, and the role of the widget. Armed with this information, assistive technologies can adapt dynamically and interact with the user in a manner that makes sense for a given situation.

What if you choose not to use the standard GTK+ widgets, and to create and use custom widgets instead? This is a choice that is made very frequently, usually to give a unique look and feel to an application or to create a library of reusable objects designed for a specific market. The good news is that you can still use custom widgets when building GNOME 2.0 applications without sacrificing accessibility. Although additional effort is required, accessibility support can be integrated by implementing the interfaces defined by ATK, just as they have been implemented in the GAIL library for the standard GTK+ widgets.

Although building an application using standard GTK+ widgets is an important first step in making the application accessible, there is a little more work to do to complete the accessibility support. The ATK provides several functions you can use to make your application work as well as possible with assistive technologies. Here are some examples, which highlight many of the ATK functions at your disposal:

- For a widget that does not display a short descriptive string, specify a name using the `atk_object_set_name ()` function. You should do this for image-only buttons, panels that provide logical groupings, text areas, and so on.
- If you can't provide a tool tip for a component, use `atk_object_set_description ()` instead to provide a description that assistive technologies can give the user. For example, the following code fragment provides an accessible description for a Close button:

```
{
AtkObject *obj;
obj = gtk_widget_get_accessible (button);
atk_object_set_description (obj, _("Closes the window"));
}
```

- Use `atk_image_set_description ()` to provide a text description for all images and icons in your program.

- Whenever you have a label that describes another component, use `atk_relation_set_add_relation ()` so that assistive technologies can find the component with which the label is associated. Here's what the code looks like:

```
{
GtkWidget *widget;
GtkLabel *label;

AtkObject *atk_widget, *atk_label;
AtkRelationSet *relation_set;
AtkRelation *relation;
AtkObject *targets[1];

atk_widget = gtk_widget_get_accessible (widget);
atk_label = gtk_widget_get_accessible (GTK_WIDGET (label));

relation_set = atk_object_ref_relation_set (atk_label);
targets[0] = atk_widget;
relation = atk_relation_new (targets, ATK_RELATION_LABEL_FOR);
atk_relation_set_add (relation_set, relation);
g_object_unref (G_OBJECT (relation));
}
```

As is apparent, the amount of code required to complete accessibility support is straightforward and manageable. If you design an application using Glade, GNOME's standard user interface design tool, the work required is further reduced since names, descriptions, and relations can be specified without any coding required.

## The Assistive Technology Service Provider Interface (AT-SPI)

The Assistive Technology Service Provider Interface (AT-SPI) is intended to allow developers to easily integrate their assistive technology products into the system. It provides a technique for these products to obtain accessibility information from applications, using standard function calls to retrieve the various types of accessibility information desired.

The GNOME 2.0 implementation of AT-SPI is CORBA-based, which allows use not only by applications based on GNOME libraries, but by any technology compliant with the CORBA 2.2 standard, including applications running on remote hosts or even on other architectures. It also means that accessibility information can be provided for applications written in a variety of programming languages.

The AT-SPI is particularly interesting because it allows AT developers to deal with only one interface across all UNIX and Linux platforms that run GNOME 2.0, thus eliminating the need to support different integration techniques on different platforms. This can significantly reduce development time.

An important architectural point is that an application and an assistive technology never directly communicate with one another. All communication happens indirectly, through the AT-SPI layer. Bridges are used to connect applications to the AT-SPI layer and are presently provided for two types of applications:

- GNOME 2.0 GTK+ based applications communicate with the AT-SPI via a bridge that uses the ATK
- Java technology-based applications (based on Swing, the Java Foundation Classes) communicate with the AT-SPI via a bridge that uses the Java Accessibility API

Bridges for applications built around architectures other than GTK+ and Java Swing are also possible because the AT-SPI is a neutral service layer that can be accessed by any client.

Since assistive technology knows only about the AT-SPI layer and is independent of the applications that may bridge to the layer, it is unaffected by application upgrades or even upgrades to the desktop itself. This greatly reduces software maintenance expenses for assistive technology suppliers.

As noted above, GNOME 2.0 will include several assistive technology solutions. However, the platform is robust enough to support a wide variety of assistive technologies interfaced through the AT-SPI. Some examples of technologies that could be integrated are as follows:

- Braille displays
- Braille printers
- Optical character recognition systems
- Voice recognition and dictation systems
- Video captioning systems

Chapter 5

# How to Get Started

There are many GNOME 2.0 related resources on the Web containing a wealth of information for end users, assistive technology developers, and application developers. There are also several GNOME-related mailing lists, most involving in-depth discussion on specific topics.

Below is a list of some of the more prominent web sites, mailing lists, and newsgroups that will increase your knowledge and understanding of the GNOME 2.0 platform. Also included is information about the Java Accessibility APIs since Java technology-based applications can also take advantage of the assistive technologies provided with GNOME 2.0 through a bridge to the AT-SPI.

## GNOME 2.0 Web Sites

**http://developer.gnome.org/projects/gap/** — This is the key page to visit to learn everything about the GNOME Accessibility Project, how to create accessible software, and how to integrate assistive technology.

**http://developer.gnome.org** — This is the GNOME Foundation's Web site devoted to programming information for the GNOME 2.0 platform.

**http://www.sun.com/gnome/** — This is Sun Microsystems Web site for the GNOME project. This is where to go for Sun's marketing information, including FAQs, white papers, and product summaries, as well as more information about GNOME's support for accessibility. The GNOME software can also be downloaded from this site.

**http://www.gnome.org** — This is the official Web site of the GNOME community project.

## GNOME 2.0 Mailing Lists

Two mailing lists have been formed that focus on accessibility issues and the GNOME 2.0 project:

GNOME-accessibility-devel

GNOME-accessibility-list

Both lists are developer oriented and are valuable resources for anyone providing accessible solutions for the GNOME 2.0 platform.

For information on how to join these lists, visit http://mail.gnome.org/mailman/listinfo/.

## Web Sites on Java Accessibility

**http://java.sun.com/products/jfc/accessibility.html**—This page describes the accessibility support integrated into the Java Foundation Classes (Swing).

**http://java.sun.com/products/jfc/jaccess-1.2.2/doc/**—This page is the best place to visit to locate developer documentation on the Java Accessibility APIs.

## Web Sites on Accessibility and Assistive Technology

**http://www.cast.org/bobby**—This is the Web page for Bobby, a tool for analyzing a Web page and preparing a report on its degree of accessibility. If you are producing content for presentation in a Web browser, you should ensure that it is accessible.

**http://www.easi.cc**—The Equal Access to Software and Information web site provides general information regarding assistive technology, accessible computing environments, web accessibility. This is a very useful site with numerous links to assistive technology and disability related issues.

**http://www.abledata.com**—The AbleData Assistive Technology Online Database containing about 20,000 products and solutions to assist users with disabilities.

**http://www.access-board.gov**—The Access Board is a federal agency devoted to universal design, and provides information regarding accessibility for information technology, the world wide Web, and more. The Access Board site is relevant for developers concerned with complying with Section 508 and other regulations mandating accessibility.

**http://www.wid.org**—The World Institute on Disability is an internationally recognized | public policy center focusing on advocacy, independent living, and issues important to the disability community.

**http://www.halftheplanet.com**—This non profit foundation supports the application of technology to promote the values of the Americans with Disabilities Act. Topics include empowerment, independent living, inclusion, the arts, and employment.

**Sun Microsystems, Inc.** 901 San Antonio Road, Palo Alto, CA 94303-4900 USA **Phone** 800 786-7638 or +1 512 434-1577 **Web** sun.com


**We make the net work.**

**Sun Worldwide Sales Offices:** Africa (North, West and Central): +33-13-067-4680, Argentina: +5411-4317-5600, Australia: +61-2-9844-5000, Austria: +43-1-60563-0, Belgium: +32-2-704-8000, Brazil: +55-11-5187-2100 Canada: +905-477-6745, Chile: +56-2-3724500, Colombia: +571-629-2323, Commonwealth of Independent States: +7-502-935-8411, Czech Republic: +420-2-3300-9311, Denmark: +45 4556 5000, Egypt: +202-570-9442 Estonia: +372-6-308-900, Finland: +358-9-525-561, France: +33-134-03-00-00, Germany: +49-89-46008-0, Greece: +30-1-618-8111, Hungary: +36-1-489-8900, Iceland: +354-563-3010, India: Bangalore: +91-80-2298989 New Delhi: +91-11-6474701, Mumbai: +91-22-2075133, Indonesia: +62-21-5276789, Ireland: +353-1-8055-666, Israel: +972-9-9710500, Italy: +39-02-641511, Japan: +81-3-5717-5000, Kazakhstan: +7-3272-466774 Korea: +822-2193-5114 Latvia: +371-750-3700, Lithuania: +370-729-8468, Luxembourg: +352-49 11 33 1, Malaysia: +603-21161888, Mexico: +52-5-258-6100, The Netherlands: +00-31-33-45-15-000, New Zealand: Auckland: +64-9-976-6800 Wellington: +64-4-462-0780, Norway: +47 23 36 96 00, People's Republic of China: Beijing: +86-10-6803-5588; Chengdu: +86-28-619-9333; Guangzhou: +86-20-8755-5900; Shanghai: +86-21-6466-1228, Hong Kong: +852-2202-6688 Philippines: +632-8857867, Poland: +48-22-8747800, Portugal: +351-21-4134000, Russia: +7-502-935-8411, Singapore: +65-64381888, Slovak Republic: +421-2-4342-94-85, South Africa: +27 11 256-6300, Spain: +34-91-596-9900 Sweden: +46-8-631-10-00, Switzerland: German: 41-1-908-90-00; French: 41-22-999-0444, Taiwan: +886-2-8732-9933, Thailand: +662-3446888, Turkey: +90-212-335-22-00, United Arab Emirates: +9714-3366333 United Kingdom: +44-1-276-20444 United States: +1-800-555-9SUN OR +1-650-960-1300, Venezuela: +58-2-905-3800 Or Online at sun.com/store



5/02