Netscape System Targets

.


Recycled and Recyclable Paper

The Team:
Engineering:
Marketing:
Publications:
Quality Assurance:
Technical Support:

Version 1.0

Printed in USA
99 98 97 10 9 8 7 6 5 4 3 2 1

Contents

1

Netscape System Targets

This document summarizes the user targets recognized by the System principal when Java code or a JavaScript script passes them to `EnablePrivilege`. For detailed information about the Java Capabilities API and the use of the targets described here, see http://developer.netscape.com/library/documentation/signedobj/capabilities/index.html.

This document includes the user descriptions for each system target and lists of some of the Java methods that check for system targets:

Target Descriptions

There are three kinds of system targets:

To avoid annoying the user by displaying a separate dialog box for each primitive target your code needs to access, you should use macro targets whenever possible. If you have any ideas for new macro targets that you'd like the System principal to recognize, post your suggestions to the Security Newsgroup at snews://secnews3.netscape.com/netscape.dev.security.

In the tables that follow, four pieces of information are provided for each target:

- the string that identifies the target in your Java code or JavaScript script, such as `UniversalFileAccess`; and for macro targets, the strings that identify their primitives
- Netscape's risk classification of High, Medium, or Low
- a short description of the target, which the user always sees when first asked to grant the principal access to the target
- a long description of the target, which the user may choose to see by clicking the Details button in the Java Security dialog box

The classification of each target as High, Medium, or Low risk is based on a rough classification system that Netscape has created. These risk classifications are provided as general guidelines only. Different users are likely to have different opinions of the level of risk that a particular target entails. Netscape encourages users to read the short and long descriptions of the access requested by a particular applet or other Java code and make their own decisions.

Netscape's current risk classification system is based on these distinctions:

- **High risk.** A major security attack is possible, permitting severe damage to your system or data. Major violation of privacy is possible, such as reading any information from hard disks connected to your computer. Very significant services may be requested, such as establishing a connection over the network to a remote computer.
- **Medium risk.** Major violation of privacy is possible, such as reading any information from a hard disk connected to your computer. Some significant services may be requested, such as writing files on a hard disk or sending email on your behalf

- **Low risk.** Minor violation of privacy is possible, such as reading your user ID. Relatively minor services may be requested, such as writing a single file to a specified noncritical directory on a hard disk connected to your computer.

Macro Targets

Table 1.1 Macro targets recognized by the system principal

Name of macro target/ Primitive targets included	Risk level	Short description	Long description
`AdministratorRegistryAccess` `StandardRegistryAccess`	High	Access to any part of your computer's registry of installed software.	Most computers store information about installed software, such as version numbers, in a registry file. System administrators sometimes need to change entries in the registry for software from a variety of vendors. You should grant this form of access only if you are running software provided by your system administrator.
`StandardRegistryAccess` `PrivateRegistryAccess`	Medium	Access to shared information in the computer's registry of installed software.	Most computers store information about installed software, such as version numbers, in a registry file. This file also includes information shared by all programs installed on your computer, including information about the user or the system. Programs that have access to shared registry information can obtain information about other programs that have the same access. This allows programs that work closely together to get information about each other. You should grant this form of access only if you know that the program requesting it is designed to work with other programs on your hard disk.

Table 1.1 Macro targets recognized by the system principal

Name of macro target/ Primitive targets included	Risk level	Short description	Long description

`Account Setup`
`UniversalBrowserRead`
`UniversalBrowserWrite`
`UniversalPreferencesRead`
`UniversalPreferencesWrite`
`UniversalTopLevelWindow`
	High	Access required to setup and configure your browser.	Access to, and modification of, browser data, preferences, files, networking and modem configuration. This access is commonly granted to the main setup program for your browser.

`CanvasAccess` `UniversalBrowserWrite`	High	Displaying text or graphics anywhere on the screen.	Displaying HTML text or graphics on any part of the screen, without window borders, toolbars, or menus. Typically granted to invoke canvas mode, screen savers, and so on.
`DatabaseAccess` `UniversalFileAccess`	High	File access typically required by database programs.	Reading, modification, or deletion of any of your files, as required by database programs.
`PresentationAccess` `UniversalFileAccess`	High	File access typically required by presentation programs.	Reading, modification, or deletion of any of your files, as required by presentation programs.
`SpreadsheetAccess` `UniversalFileAccess`	High	File access typically required by spreadsheet programs.	Reading, modification, or deletion of any of your files, as required by spreadsheet programs.
`WordProcessorAccess` `UniversalFileAccess`	High	File access typically required by word-processing programs.	Reading, modification, or deletion of any of your files, as required by word-processing programs.

Table 1.1 Macro targets recognized by the system principal

Name of macro target/ Primitive targets included	Risk level	Short description	Long description

`Debugger`
`UniversalExecAccess`
`UniversalPropertyWrite`
`UniversalPropertyRead`
`UniversalFileRead`
`UniversalListen`
`UniversalAccept`
`UniversalConnect`
	High	Access required by debugging software used by professional programmers.	You should grant this access only if you are a professional programmer using debugging software from a reliable vendor.

`GamesAccess` `PrivateRegistryAccess`	Low	Limited registry access typically required by games.	Reading and modification of a limited registry area as required by games to save scores.

`IIOPRuntime`
`UniversalAccept`
`UniversalConnect`
`UniversalListen`
	High	Implementing IIOP remote object software.	Internet Inter-ORB Protocol (IIOP) is an open industry standard for distributing objects. It is commonly used for programs, such as banking applications, that involve transferring information among different kinds of computer systems over a network. Granting this access permits Java code to implement IIOP on your computer and to access remote objects over a network. You should grant this access only if you are running a program, from a reliable IIOP vendor, that needs to connect with remote objects over a network.

Table 1.1 Macro targets recognized by the system principal

Name of macro target/ Primitive targets included	Risk level	Short description	Long description

`Netcaster`
`SiteArchiveTarget`
`UniversalBrowserAccess`
`UniversalConnect`
 	
`UniversalConnectWithRedirect`
`UniversalFileAccess`
`UniversalPreferencesRead`
`UniversalPreferencesWrite`
`UniversalThreadAccess`
`UniversalThreadGroupAccess`
	High	Access required by netcasting programs.	Access required by programs, such as Netscape Netcaster, that allow users to receive information over Internet channels and work with network resources off-line.

`TerminalEmulator`
`UniversalLinkAccess`
`UniversalPropertyRead`
`UniversalListen`
`UniversalAccept`
`UniversalConnect`
	High	Access required by terminal emulators and other communications programs.	Reading and writing files and establishing network connections. This form of access is required by terminal emulators such as the 3270 or VT100 emulator.

`30Capabilities`
`UniversalThreadAccess`
`UniversalThreadGroupAccess`
`UniversalLinkAccess`
`UniversalPropertyWrite`
`UniversalPropertyRead`
`UniversalListen`
`UniversalAccept`
`UniversalConnect`
`UniversalTopLevelWindow`
`UniversalPackageAccess`
`UniversalPackageDefinition`
	High	Access required by Navigator plug-ins and similar programs.	Access required by plug-ins and other programs containing native code. This form of access is allowed automatically in Navigator 3.0 for code that is downloaded to your hard disk.

Table 1.1 Macro targets recognized by the system principal

Name of macro target/ Primitive targets included	Risk level	Short description	Long description

`UniversalBrowserAccess`
`UniversalBrowserRead`
`UniversalBrowserWrite`
	High	Reading or modifying browser data.	Reading or modifying browser data that may be considered private, such as a list of web sites visited or the contents of web forms you may have filled in. Modifications may also include creating windows that look like they belong to another program or positioning windows anywhere on the screen.

`UniversalFileAccess`
`UniversalLinkAccess`
`UniversalPropertyRead`
`UniversalFileRead`
`UniversalFileWrite`
`UniversalFileDelete`
	High	Reading, modification, or deletion of any of your files.	This form of access is typically required by a program such as a word processor or a debugger that needs to create, read, modify, or delete files on hard disks or other storage media connected to your computer.

Primitive Targets

Table 1.2 lists the primitive system targets recognized by the System principal. Long descriptions that are missing will be provided with later versions of this document.

Table 1.4 lists a subset of the targets in Table 1.2 and some of the Java methods that check for those targets.

For a list of the Java methods in Table 1.4 sorted by method, see Table 1.5.

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`PrivateRegistryAccess`	Low	Access to the vendor's portion of your computer's registry of installed software.	Most computers store information about installed software, such as version numbers, in a registry file. When you install new software, the installation program sometimes needs to read or change entries in the portion of the registry that describes the software vendor's products. You should grant this form of access only if you are installing new software from a reliable vendor. The entity that signs the software can access only that entity's portion of the registry.
`SiteArchiveTarget`	High	Access to the site archive file.	Access required to add, modify, or delete site archive files and make arbitrary network connections in the process. This form of access is required only by netcasting applications such as Netscape Netcaster, which request it in combination with several other kinds of access. Applications should not normally request this access by itself, and you should not normally grant it.
`UniversalAccept`	High	Accepting connections from other computers on a network.	
`UniversalAwtEventQueueAccess`	High	Monitoring or intercepting typing or mouse movements.	

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`UniversalBrowserRead`	Medium	Reading browser data.	Access to browser data that may be considered private, such as a list of web sites visited or the contents of web page forms you may have filled in.
`UniversalBrowserWrite`	High	Modifying the browser.	Modifying the browser in a potentially dangerous way, such as creating windows that may look like they belong to another program or positioning windows anywhere on the screen.
`UniversalConnect`	High	Contacting and connecting with other computers over a network.	
`UniversalConnectWithRedirect`	High	Contacting and connecting with other computers over a network	Contacting and connecting with other computers, where the remote computer can redirect connection requests to another computer.
`UniversalDialogModality`		Displaying a dialog box that may temporarily disable the browser.	Displaying modal dialog boxes: that is, dialog boxes that require you to type or respond in some way before you can do anything else. Modal dialog boxes temporarily disable the browser and can cause problems if not correctly implemented.
`UniversalExitAccess`	High	Exiting the Communicator program.	Exiting all parts of the Communicator program that are currently running and releasing the memory they occupy.

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`UniversalFdRead`	High	Reading data from a network connection.	Reading data from a network connection via file descriptor.
`UniversalFdWrite`	High	Writing data from a network connection.	Writing data from a network connection via file descriptor.
`UniversalFileDelete`	High	Deleting files stored in your computer.	Deletion of any files stored on hard disks or other storage media connected to your computer.
`UniversalFileRead`	High	Reading files stored in your computer.	Reading any files stored on hard disks or other storage media connected to your computer.
`UniversalFileWrite`	High	Modifying files stored in your computer.	Modifying any files stored on hard disks or other storage media connected to you computer.
`UniversalLinkAccess`	High	Using native code stored in dynamically linked libraries.	Using code written specifically for the operating system of your computer. Such code must be stored in dynamically linked libraries on hard disks or other storage media connected to your computer.
`UniversalListen`	High	Accepting connections from other computers on a network.	
`UniversalMulticast`	High	Broadcasting information to multiple computers over a network.	
`UniversalPreferencesRead`	Medium	Reading preferences settings.	Access to read the current settings of your preferences

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`UniversalPreferencesWrite`	High	Modifying preferences settings.	Modifying the current settings of your preferences.
`UniversalPrintJobAccess`	Low	Printing from within Communicator.	
`UniversalPropertyRead`	Medium	Reading information stored in your computer, such as your user name.	Reading information stored in your computer that is normally kept private, such as your user name and the current directory.
`UniversalPropertyWrite`	High	Modifying sensitive information stored in your computer.	Modifying sensitive information stored in your computer that is normally kept private, such as certain security policy controls.
`UniversalSendMail`	Medium	Sending email messages on your behalf.	
`UniversalSetFactory`	High	Defining protocol handlers for network connections.	
`UniversalSystemClipboardAccess`	High	Reading and writing to the system clipboard for your computer.	
`UniversalThreadAccess`	High	Manipulating other applets (threads) running on your computer.	
`UniversalThreadGroupAccess`	High	Manipulating groups of applets (threads) running on your computer.	
`UniversalTopLevelWindow`	High	Displaying windows that don't have the unsigned applet label.	

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`LimitedInstall`	High	Installing Java software on your computer.	Installing Java class files in the "Java Download" directory. This form of access allows new files to be added to this single directory on your computer's main hard disk, potentially replacing other files that have previously been installed in the directory.

Table 1.2 Primitive targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`FullInstall`	High	Installing and running software on your computer.	Installing software on your computer's main hard disk, potentially deleting other files on the hard disk. Each time a program that has this form of access attempts to install software, it must display a dialog box that lets you choose whether to go ahead with the installation. If you go ahead, the installation program can execute any software on your computer. This potentially dangerous form of access is typically requested by an installation program after you have downloaded new software or a new version of software that you have previously installed. You should not grant this form of access unless you are installing or updating software from a reliable source.
`SilentInstall`		Installing and running software without warning you.	Installing software on your computer's main hard disk without giving you any warning, potentially deleting other files on the hard disk. Any software on the hard disk may be executed in the process. This is an extremely dangerous form of access. It should be granted by system administrators only.

Parameterized Targets

Table 1.3 Parameterized targets recognized by the System principal

Name of primitive target	Risk level	Short description	Long description
`FileRead`	High	Reading a specific file on your hard disk.	
`FileWrite`	High	Modifying a specific file on your hard disk.	

Methods That Check for System Targets

Table 1.4 lists a subset of the targets in Table 1.2 and some of the Java methods that check for those targets. For a list of the Java methods in Table 1.4 sorted by method, see Table 1.5.

Table 1.4 Selected primitive targets and some of the methods that check for them

Name of primitive target	Some of the methods that check for this target
`UniversalAccept`	`java.net.ServerSocket` `public Socket accept()`

`UniversalConnect`	`java.net.DatagramSocket`
	`public InetAddress getLocalAddress() public void`
	`public synchronized void receive(DatagramPacket p)`
	`send(DatagramPacket p)`
	
	`java.net.MulticastSocket`
	`public synchronized void send(DatagramPacket p,`
	`byte ttl)`
	
	`java.net.Socket`
	`public Socket(String host, int port)`
	`public Socket(InetAddress address, int port)`
	`public Socket(InetAddress host, int port,`
	`boolean stream)`
	`public Socket(String host, int port,`
	`InetAddress localAddr, int localPort)`
	`public Socket(InetAddress address, int port,`
	`InetAddress localAddr, int localPort)`
	`public Socket(String host, int port, boolean stream)`
	
	`netscape.net.URLConnection`
	`public void connect()`
	
	`sun.awt.macos.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`URL url)`
	
	`sun.awt.motif.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`URL url)`
	
	`sun.awt.win32.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`URL url)`
	
	`sun.awt.windows.WToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`URL url)`
	
	`sun.net.www.http.HttpClient`
	`public synchronized void openServer(String host,`
	`int port)`

Table 1.4 Selected primitive targets and some of the methods that check for them

Name of primitive target	Some of the methods that check for this target

`UniversalExecAccess`	`java.lang.Runtime`
	`public Process exec(String command, String envp[])`
	`public Process exec(String cmdarray[], String envp[])`
	
	`java.lang.System`
	`public static void setErr(PrintStream err)`
	`public static void setIn(InputStream in)`
	`public static void setOut(PrintStream out)`

`UniversalExitAccess`	`java.lang.Runtime` `public void exit(int status)`
`UniversalFileDelete`	`java.io.File` `public boolean delete()`

`UniversalFileRead`	`java.io.File`
	`public boolean canRead()`
	`public boolean exists()`
	`public String getCanonicalPath()`
	`public boolean isDirectory()`
	`public boolean isFile()`
	`public boolean isLink()`
	`public long lastAccessed()`
	`public long lastStatusChange()`
	`public long length()`
	`public String[] list()`
	
	`java.io.FileInputStream`
	`public FileInputStream(String name)`
	`public FileInputStream(FileDescriptor fdObj)`
	
	`java.io.RandomAccessFile`
	`public RandomAccessFile(String name, String mode)`
	`public RandomAccessFile(File file, String mode)`
	
	`sun.awt.macos.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`String filename)`
	
	`sun.awt.motif.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`String filename)`
	
	`sun.awt.win32.MToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`String filename)`
	
	`sun.awt.windows.WToolkit`
	`static synchronized Image getImageFromHash(Toolkit tk,`
	`String filename)`

Table 1.4 Selected primitive targets and some of the methods that check for them

Name of primitive target	Some of the methods that check for this target

`UniversalFileWrite`	`java.io.File`
	`public boolean canWrite()`
	`public boolean mkdir()`
	`public boolean renameTo(File dest)`
	
	`java.io.FileOutputStream`
	`public FileOutputStream(String name)`
	`public FileOutputStream(String name, boolean append)`
	`public FileOutputStream(FileDescriptor fdObj)`
	
	`java.io.RandomAccessFile`
	`public RandomAccessFile(String name, String mode)`
	`public RandomAccessFile(File file, String mode)`

`UniversalLinkAccess`	`java.lang.Runtime`
	`public synchronized void load(String filename)`
	`public synchronized void loadLibrary(String libname)`
	
	`java.lang.System`
	`public static void load(String filename)`
	`public static void loadLibrary(String libname)`

`UniversalListen`	`java.net.DatagramSocket`
	`public DatagramSocket()`
	`public DatagramSocket(int port, InetAddress laddr)`
	
	`java.net.MulticastSocket`
	`public MulticastSocket()`
	`public MulticastSocket(int port)`
	
	`java.net.ServerSocket`
	`public ServerSocket(int port, int backlog,`
	`InetAddress bindAddr)`

`UniversalMulticast`	`java.net.DatagramSocket`
	`public void send(DatagramPacket p)`
	
	`java.net.MulticastSocket`
	`public void joinGroup(InetAddress mcastaddr)`
	`public void leaveGroup(InetAddress mcastaddr)`
	`public synchronized void send(DatagramPacket p,`
	`byte ttl)`

`UniversalPropertyRead`	`java.lang.System`
	`public static String getProperty(String key)`
	`public static String getProperty(String key,`
	`String def)`

Table 1.4 Selected primitive targets and some of the methods that check for them

Name of primitive target	Some of the methods that check for this target

`UniversalPropertyWrite`	`java.lang.System`
	`public static Properties getProperties()`
	`public static void setProperties(Properties props)`

`UniversalSetFactory`	`java.net.ServerSocket`
	`public static synchronized void setSocketFactory (SocketImplFactory fac)`
	
	`java.net.Socket`
	`public static synchronized void setSocketImplFactory (SocketImplFactory fac)`

`UniversalThreadAccess`	`java.lang.Thread`
	`public void checkAccess()`
	`public void interrupt()`
	`public final void resume()`
	`public final void setPriority(int newPriority)`
	`public final void setName(String name)`
	`public final void setDaemon(boolean on)`
	`public final synchronized void stop(Throwable o)`
	`public final void suspend()`

`UniversalThreadGroupAccess`	`java.lang.ThreadGroup`
	`public final void checkAccess(int caller_depth)`
	`public final void destroy()`
	`public final void resume()`
	`public final void setDaemon(boolean daemon)`
	`public final void setMaxPriority(int pri)`
	`public final void stop()`
	`public final void suspend()`
	`public ThreadGroup(ThreadGroup parent, String name)`

`UniversalTopLevelWindow`	`java.awt.Window` `public Window(Frame parent)`

Table 1.5 shows the targets and Java methods listed in Table 1.4, sorted by class and method.

Table 1.5 Some methods that require access to primitive system targets

Class	Method	Primitive target
`java.awt.Window`	`Window`	`UniversalTopLevelWindow`
`java.io.File`	`canRead`	`UniversalFileRead`
	`canWrite`	`UniversalFileWrite`

Table 1.5 Some methods that require access to primitive system targets

Class	Method	Primitive target
	delete	UniversalFileDelete
	exists	UniversalFileRead
	getCanonicalPath	UniversalFileRead
	isDirectory	UniversalFileRead
	isFile	UniversalFileRead
	isLink	UniversalFileRead
	lastAccessed	UniversalFileRead
	lastStatusChange	UniversalFileRead
	length	UniversalFileRead
	list	UniversalFileRead
	mkdir	UniversalFileWrite
	renameTo	UniversalFileWrite
java.io.FileInputStream	FileInputStream	UniversalFileRead
java.io.FileOutputStream	FileOutputStream	UniversalFileWrite
java.io.RandomAccessFile	RandomAccessFile	UniversalFileRead UniversalFileWrite
java.lang.Runtime	exec	UniversalExecAccess
	exit	UniversalExitAccess
	load	UniversalLinkAccess
	loadLibrary	UniversalLinkAccess
java.lang.System	getProperties	UniversalPropertyWrite
	getProperty	UniversalPropertyRead
	load	UniversalLinkAccess
	loadLibrary	UniversalLinkAccess
	setErr	UniversalExecAccess
	setIn	UniversalExecAccess
	setOut	UniversalExecAccess
	setProperties	UniversalPropertyWrite
java.lang.Thread	checkAccess	UniversalThreadAccess

Table 1.5 Some methods that require access to primitive system targets

Class	Method	Primitive target
	interrupt	UniversalThreadAccess
	resume	UniversalThreadAccess
	setPriority	UniversalThreadAccess
	setName	UniversalThreadAccess
	setDaemon	UniversalThreadAccess
	stop	UniversalThreadAccess
	suspend	UniversalThreadAccess
java.lang.ThreadGroup	checkAccess	UniversalThreadGroupAccess
	destroy	UniversalThreadGroupAccess
	resume	UniversalThreadGroupAccess
	setDaemon	UniversalThreadGroupAccess
	setMaxPriority	UniversalThreadGroupAccess
	stop	UniversalThreadGroupAccess
	suspend	UniversalThreadGroupAccess
	ThreadGroup	UniversalThreadGroupAccess
java.net.DatagramSocket	DatagramSocket	UniversalListen
	getLocalAddress	UniversalConnect
	receive	UniversalConnect
	send	UniversalConnect UniversalMulticast
java.net.MulticastSocket	joinGroup	UniversalMulticast
	leaveGroup	UniversalMulticast
	MulticastSocket	UniversalListen
	send	UniversalConnect UniversalMulticast
java.net.ServerSocket	accept	UniversalAccept
	ServerSocket	UniversalListen
	setSocketFactory	UniversalSetFactory
java.net.Socket	MSocket	UniversalConnect
	setSocketImplFactory	UniversalSetFactory

Table 1.5 Some methods that require access to primitive system targets

Class	Method	Primitive target
`netscape.net.URLConnection`	`connect`	`UniversalConnect`
`sun.awt.macos.MTooklit`	`getImageFromHash`	`UniversalFileRead` `UniversalConnect`
`sun.awt.motif.MTooklit`	`getImageFromHash`	`UniversalFileRead` `UniversalConnect`
`sun.awt.win32.MTooklit`	`getImageFromHash`	`UniversalFileRead` `UniversalConnect`
`sun.awt.windows.MTooklit`	`getImageFromHash`	`UniversalFileRead` `UniversalConnect`