intent® JTE Support for Java^{TM} Technology

Version 1.12

1. Support for PersonalJava^{TM} Application Environment Specification Version 1.2

PJAE 1.2 is based upon the JDK 1.1.8 API, adding security as specified in JDK 1.2 . It has been designed to reflect the software needs of set-top boxes, smart phones and other networked personal consumer devices. Releases 1.1 and onwards of intent®, Java^{TM} Technology Edition fully support PJAE 1.2

For all following tables, an asterisk '*' preceding a PersonalJava^{TM} Application Environment feature indicates that this feature is optional in the PJAE 1.1.3, and a '†' indicates a feature of the PJAE 1.2 platform which is modified from its JDK 1.1.8 counterpart. Features preceded by a '‡' are specified in JDK 1.2 rather than JDK 1.1.8

Here the term 'modified' indicates that the feature is not completely supported by the PJAE. If a package is modified, then some of its classes may be optional, PJAE-specific or modified. If a class is modified, then some of its methods may be optional, PJAE-specific or modified. If a method is modified, then its semantics are changed from the JDK.

The word 'optional' is used to describe a feature that is not required to be supported by the PJAE. The choice of whether to support the feature is left to the PJAE implementer. However, if the implementer elects to support a given feature, then the implementation must support it completely and retain exactly the same API as its counterpart in the JDK. An optional feature that is left out of an implementation is called an 'unsupported optional feature.'

1.1 JDK-Based APIs

Package	**intent JTE Support**
java.applet	Yes
†java.awt	Yes Includes all optional classes Includes all optional methods except GetPrintJob
java.awt.datatransfer	Yes
java.awt.event	Yes
java.awt.image	Yes
†java.awt.peer	Yes
java.beans	Yes
†java.io	Yes Includes all optional classes
†java.lang	Yes
java.lang.reflect	Yes
*java.math	Yes
†java.net	Yes
*java.rmi	Yes
*java.rmi.dgc	Yes
*java.rmi.registry	Yes
*java.rmi.server	Yes
†java.security	Yes Includes all optional classes
*java.security.acl *(unsupported)*	No
‡java.security.cert	Yes Includes all optional classes
*‡java.security.interfaces	Yes
*‡java.security.spec	Yes
*java.sql	Yes
java.text	Yes

†java.text.resources	Yes
†java.util	Yes Includes all optional classes
java.util.jar	Yes
†java.util.zip	Yes

1.2 PJAE-Specific APIs

com.sun.awt	*interface* NoInputPreferred	Yes
	interface KeyboardInputPreferred	Yes
	interface ActionInputPreferred	Yes
	interface PositionalInputPreferred	Yes
com.sun.lang	UnsupportedOperationException	Yes
com.sun.util	Ptimer	Yes
	PTimerSpec	Yes
	PTimerWentOffEvent	Yes
	interface PTimerWentOffListener	Yes

1.3 Networking Protocols

http 1.0	Yes
*Secure Sockets Layer (SSL) 3.0	No
*gopher	No
*ftp	No
*mailto (SMTP)	No
*file	Yes

1.4 Image Formats

CompuServe GIF version 89a	Yes
JPEG (JFIF)	Yes
XBM (XBitmap)	Yes

The image formats CPM and PNG, not required by the PJAE 1.2, are also supported.

2. Changes from JDK 1.1.8

Although most of the features of the PersonalJava 1.2 Application Environment can be assumed to be identical to the equivalent features in the JDK 1.1.8, there are cases where this is not so. Such modified and optional features are described below.

2.1 java.applet

Sound support is implemented on some platforms, in which case the sound formats AU (8kHz µlaw encoding) and WAV (pcm only) are supported

2.2 java.awt

2.2.1 Dialog

There are two possible levels of support for the *Dialog* class. The minimal implementation must allow only a single modal dialog to be visible at a time. Minimal PJAE implementations do not support modeless dialogs.

It should be noted that even in a full implementation of *Dialog*, the method *setResizable* may ignore the specified value.

int**e**nt provides a full implementation of the *Dialog* class.

2.2.2 Frame

Like *Dialog*, *Frame* offers two levels of support. At minimum, an implementation must allow the *Frame* constructor to be called once to create a root for its component hierarchy. The *Frame* constructor may not be called again unless the previous instance has been destroyed.

A full implementation of *Frame* is identical to that required by the JDK 1.1.8 API. An implementation that fully implements *Frame* must also implement the optional classes *CheckboxMenuItem*, *Menu*, *MenuBar* and *MenuShortcut*.

The classes *Frame* and *Dialog* are mutually dependent; an implementation must support both at the same level. The int**e**nt implementation provides full support for both.

2.2.3 Window

There are different levels of support for *Window*. A minimum implementation can prohibit the direct creation of *Window* objects. int**e**nt provides a full implementation of *Window* as described in the JDK 1.1.8 API.

2.2.4 Scrolling Policy

Some classes in the *java.awt* package implement scrollbar display policies that are different from those defined in the JDK 1.1.8 API. The classes *List*, *Scrollbar*, *ScrollPane* and *Textarea* are affected by these changes.

2.3 java.lang.reflect

As well as the addition of classes from the JDK 1.2 Security API, the major change that has taken place in this package is that some classes that previously inherited from *java.lang.Object* now inherit instead from *java.lang.reflect.AccessibleObject*. This affects the *java.lang.reflect* classes *Constructor*, *Field* and *Method*.

2.4 java.sql

In implementations that support the *java.sql* package, the full JDK 1.1.8 API of the *java.math* package is also required.

2.5 java.text.resources

As well as requiring full support of the JDK 1.1.8 for three classes, the PJAE 1.2 requires at least two matched locale classes. int**e**nt provides all of the locale classes provided by Sun in the reference source

2.6 Optional Code Signing Mechanism

Where a PJAE 1.2 implementation supports the optional code signing mechanism, it must include all of the optional classes listed in the *java.security* and *java.security.cert* packages. The optional packages *java.security.interfaces*, *java.security.spec* and *java.math* are also required, as are the optional classes in *java.util* and *java.util.jar*. int**e**nt fully supports this mechanism.

3. Support for PersonalJava Application Environment Specification Version 1.1.3

The PJAE 1.1.3 specification is based upon the JDK 1.1.6 specification, with a small number of new APIs. Release 1.0 of int**e**nt Java Technology Edition fully supports all the required features of the PJAE 1.1.3 platform.

For all following tables, an asterisk '*' preceding a PersonalJava feature indicates that this feature is optional in the PJAE 1.1.3, and a '†' indicates a feature of the PJAE 1.1.3 platform which is modified from its JDK 1.1.6 counterpart.

3.1 JDK-Based APIs

Package	**intent JTE Support**
java.applet	Yes
†java.awt	Yes Includes all optional classes Includes all optional methods except GetPrintJob
java.awt.datatransfer	Yes
java.awt.event	Yes
java.awt.image	Yes
†java.awt.peer	Yes
java.beans	Yes
†java.io	Yes Includes all optional classes
†java.lang	Yes
java.lang.reflect	Yes
*java.math	Yes
†java.net	Yes
*java.rmi	Yes
*java.rmi.dgc	Yes
*java.rmi.registry	Yes
*java.rmi.server	Yes
*java.security	Yes
*java.security.acl *(unsupported)*	Yes
*java.security.interfaces	Yes
*java.sql	Yes
java.text	Yes
†java.text.resources	Yes
java.util	Yes
†java.util.zip	Yes Includes all optional classes except GZIPOutputStream

3.2 PJAE-Specific APIs

com.sun.awt	*interface* NoInputPreferred	Yes
	interface KeyboardInputPreferred	Yes
	interface ActionInputPreferred	Yes
	interface PositionalInputPreferred	Yes
com.sun.lang	UnsupportedOperationException	Yes
com.sun.util	Ptimer	Yes
	PTimerSpec	Yes
	PTimerWentOffEvent	Yes
	interface PTimerWentOffListener	Yes

3.3 PJAE-Specific APIs

com.sun.awt	*interface* NoInputPreferred	Yes
	interface KeyboardInputPreferred	Yes
	interface ActionInputPreferred	Yes
	interface PositionalInputPreferred	Yes
com.sun.lang	UnsupportedOperationException	Yes
com.sun.util	PTimer	Yes
	PTimerSpec	Yes
	PTimerWentOffEvent	Yes
	interface PTimerWentOffListener	Yes

3.4 Networking Protocols

http 1.0	Yes
*Secure Sockets Layer (SSL) 3.0	No
*gopher	No
*ftp	No
*mailto (SMTP)	No
*file	Yes

4. Changes from JDK 1.1.6

Although most of the features of the PersonalJava 1.1.3 platform can be assumed to be identical to the equivalent features in the JDK 1.1.6, there are cases where this is not so. Such modified and optional features are described below.

4.1 Java.awt

4.1.1 Component

The *setCursor* method in the class *Component* differs from that in the JDK 1.1.6 in that the specified cursor may be ignored. This is because some implementations of the PersonalJava platform may not support cursors, while others may have reason to limit the number of cursors displayed.

4.1.2 Frame

Like *Dialog*, there are two possible levels of implementation for *Frame*. The minimum of these allows the *Frame* constructor to be called only once to create a root for its component hierarchy.

As *Dialog* and *Frame* are mutually dependent, the int**e**nt implementation of *Frame* is also a full implementation.

4.1.3 Dialog

PersonalJava implementations can provide two levels of support for *Dialog*. In the minimum implementation, a single modal dialog must be permitted to be visible at a time. If a Java program tries to display a second, the first may be hidden for as long as the second is visible. This level of implementation does not support modeless dialogs.

The int**e**nt implementation provides all the features of the *Dialog* class as defined in the JDK 1.1.6.

Furthermore, the *setResizeable* method in the class *Dialog* allows the specified value to be ignored, which is not the case in the JDK 1.1.6.

4.1.4 Window

There are two possible levels of implementation for the *Window* class. At its minimum, the implementation can prohibit the direct creation of *Window* objects. int**e**nt provides a full implementation of *Window* as defined in the JDK 1.1.6.

4.2 java.text

The int**e**nt implementation of *java.text* provides three available encodings. These are UTF8, 8859-1 (default) and ESJIS (for Japanese characters).

4.3 java.text.resources

As well as requiring full support of the JDK 1.1.6 for three classes, the PJAE 1.1.3 requires at least two matched locale classes. int**e**nt provides all of the locale classes provided by Sun in the reference source.