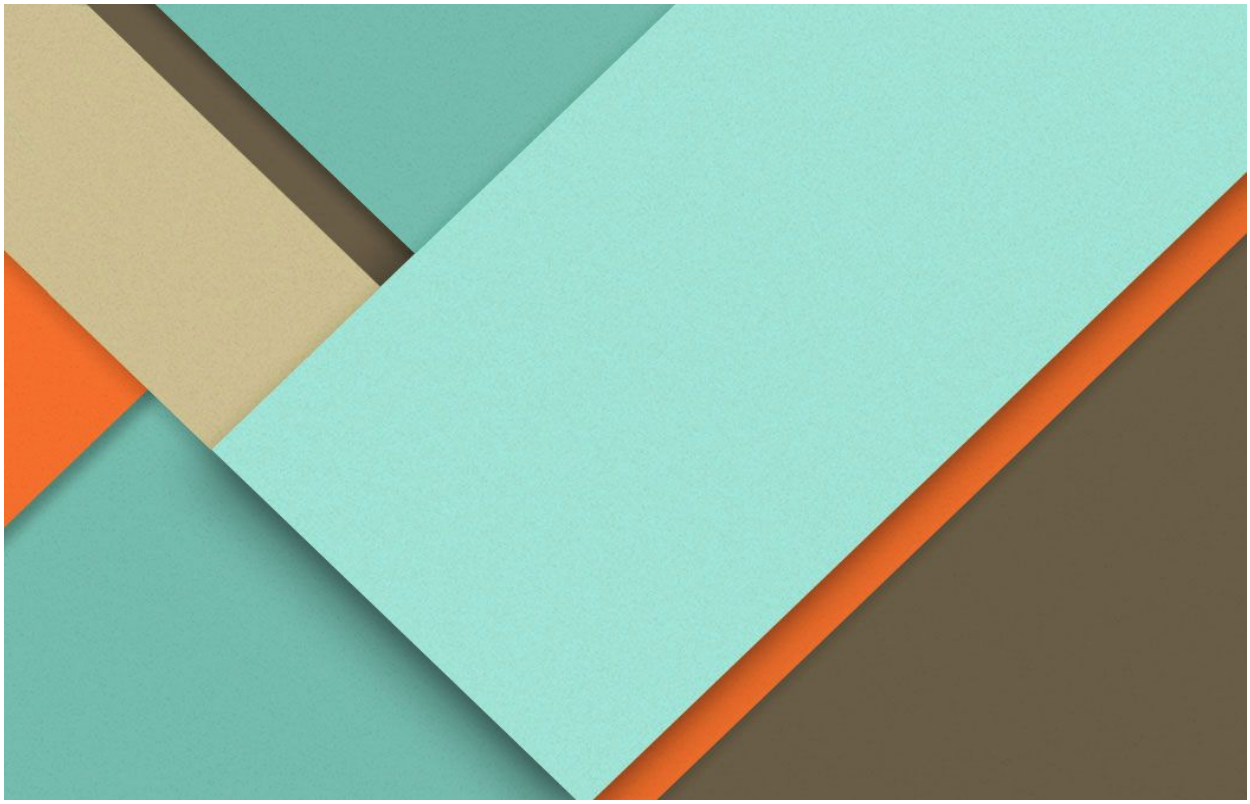


WIRELESS COMMUNICATION THROUGH SERIAL COMM PORT

PROJECT REPORT

23.11.2018

ANIMESH TYAGI

B.Tech [IT] 4th Year

Roll no - 1471913002

Axis Institute Of Technology & Management

ACKNOWLEDGEMENT

I would like to acknowledge ‘INDUSTRIAL ENGINEERING,KANPUR’ where I completed my Summer Internship in the year 2018.It was there when I got to know about this not-so-popular Serial Port present in most computers which gave me an idea to pursue this Project.

Also I am thankful of my Intern handler Mr. Santosh Kumar & Mr. Saurabh Sharma under whose guideline I was able to complete my project. I am wholeheartedly thankful to him for giving me his valueable time & attention & for providing me with a systematic way for completing my project in time.

Animesh Tyagi

CONTENTS

OVERVIEW

1.1) INTRODUCTION TO THE PROJECT

In this project we will dive into what possibilities that can be explored with Serial Port.It is highly efficient in communicating between two or more devices simultaneously.

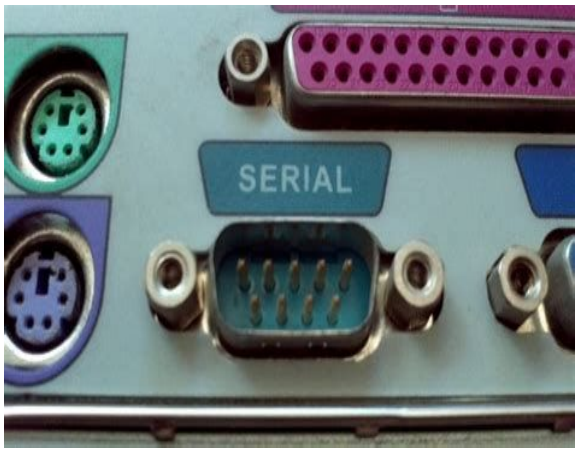
The same can be done with USB as well but programming I/O responses in USB as in sending and receiving message can be highly complicated & equally strict "tree" topology and "master-slave" protocol for addressing peripheral devices; peripheral devices cannot interact with one another except via the host, and two hosts cannot communicate over their USB ports directly.

After the abstract driver implementation the Serial Port will provide for a more reliable transport, by making writing serial data completely buffered. With the MS implementation, one can write data, but subsequently needs to check if all data is written or not. If it isn't written, then it needs to be retried.

Serial port takes a byte of data and transmits all the 8 bits of a byte, one bit at a time. So, Serial port requires only one wire to transmit the 8 bits one by one. Before transmitting each byte of a data, serial port sends a start bit, which is 0. After each byte of a data, it sends a stop bit to indicate that that the byte is complete. Serial ports depends on special controller chip, UART (Universal Asynchronous Receiver/ Transmitter). The UART takes the output parallely of the computer's system bus and transforms it into a form that can be transmitted through serial port. Most UART chips have a built-in buffer from 16 to 64 kilobytes, in order to function faster. When buffer is processing data going out to the serial port, it enables the chip to cache data coming in from system bus. Serial ports are also known as communication (COM) ports. They are bi-directional i.e., it allows each device to receive as well as transmit data. Serial devices have different pins to transmit and receive data. If we use same pins then it would limit the communication such that information could only travel in one direction at a time. Using different pins enables communication such that information can travel in both the direction at a time.

In this project we will make a program and an algorithm for it so that the two can communicate with each other with the help of Serial Port.So there are two aspects for it one will be the microcontroller which has a display and the other is our computer which can receive I/O data from the microcontroller through serial port which is already there on most computers.

1.2) INTRODUCTION TO SERIAL PORT



What is a Serial Port ?

In computing, a **serial port** is a serial communication interface through which information transfers in or out one bit at a time (in contrast to a parallel port).Throughout most of the history of personal computers, data was transferred through serial ports to devices such as modems, terminals, and various peripherals.

While such interfaces as Ethernet, FireWire, and USB all send data as a serial stream, the term "serial port" usually identifies hardware more or less compliant to the RS-232 standard, intended to interface with a modem or with a similar communication device.

Modern computers without serial ports may require USB-to-serial converters to allow compatibility with RS-232 serial devices. Serial ports are still used in applications such as industrial automation systems, scientific instruments, point of sale systems and some industrial and consumer products. Server computers may use a serial port as a control console for diagnostics. Network equipment (such as routers and switches) often use serial console for configuration. Serial ports are still used in these areas as they are simple, cheap and their console functions are highly standardized and widespread. A serial port requires very little supporting software from the host system.

Considered to be one of the most basic external connections to a computer, the serial port has been an integral part of most computers for more than 20 years. Although many of the newer systems have done away with the serial port completely in favor of USB connections, most modems still use the serial port, as do some printers, PDAs and digital cameras. Few computers have more than two serial ports.

Essentially, serial ports provide a standard connector and protocol to let you attach devices, such as modems, to your computer.

2. SYSTEM DESIGN

2.1 Hardware

Some computers, such as the IBM PC, use an integrated circuit called a UART. This IC converts characters to and from asynchronous serial form, implementing the timing and framing of data in hardware. Very low-cost systems, such as some early home computers, would instead use the CPU to send the data through an output pin, using the bit banging technique. Before large-scale integration (LSI) UART integrated circuits were common, a minicomputer or microcomputer would have a serial port made of multiple small-scale integrated circuits to implement shift registers, logic gates, counters, and all the other logic for a serial port.

Early home computers often had proprietary serial ports with pinouts and voltage levels incompatible with RS-232. Inter-operation with RS-232 devices may be impossible as the serial port cannot withstand the voltage levels produced and may have other differences that "lock in" the user to products of a particular manufacturer.

Low-cost processors now allow higher-speed, but more complex, serial communication standards such as USB and FireWire to replace RS-232. These make it possible to connect devices that would not have operated feasibly over slower serial connections, such as mass storage, sound, and video devices.

Many personal computer motherboards still have at least one serial port, even if accessible only through a pin header. Small-form-factor systems and laptops may omit RS-232 connector ports to conserve space, but the electronics are still there. RS-232 has been standard for so long that the circuits needed to control a serial port became very cheap and often exist on a single chip, sometimes also with circuitry for a parallel port.

Below is a listing of various hardware components that can be purchased and used with your serial port.

Mouse - One of the most commonly used devices for serial ports, usually used with computers with no PS/2 or USB ports and specialty mice.

Modem - Another commonly used device for serial ports. Used commonly with older computers, however, is also commonly used for its ease of use.

Network - One of the original uses of the serial port, which allowed two computers to connect together and allow large files to be transferred between the two.

Printer - Today, this not a commonly used device for serial ports. However, was frequently used with older printers and plotters.

2.2 Difference Between COM,USB,Serial Port

Serial port is a type of device that uses an UART chip, a Universal Asynchronous Receiver Transmitter. One of the two basic ways to interface a computer in the olden days, parallel ports were the other way. Serial is simple to hook up, it doesn't need a lot of wires. Parallel was useful if you wanted to go fast, typ 8 times faster than serial, but cables and connectors were expensive. Parallel I/O has completely disappeared from computer designs, caught up by tremendous advances in bus transceivers, the kind of chip that can transmit an electrical signal down a wire.

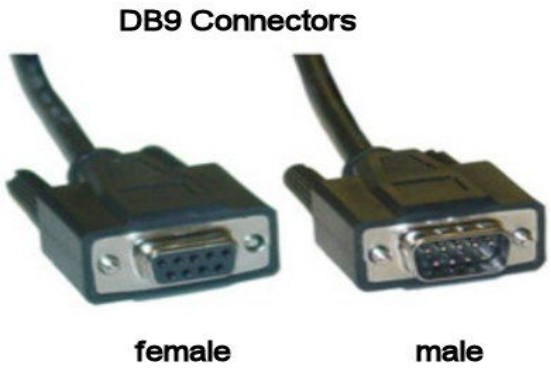
COM comes from MS-Dos, it is a *device name*. Short for "COMmunication port". Computers in the 1980's usually had two serial ports, labeled COM1 and COM2 on the back of the machine. This name was carried forward into Windows, most any driver that simulates a serial port will create a device with "COM" in its name. LPT was the device name for parallel ports, short for "Line PrinTer".

RS-232 was an electrical signaling standard for serial ports. It is the simplest one with very low demands on the device, supporting just a point-to-point connection. RS-422 and RS-485 were not uncommon, using a twisted pair for each signal, providing much higher noise immunity and allowing multiple devices connected to each other.

USB means Universal Serial Bus. Empowered by the ability to integrate a micro-processor into devices that's a few millimeters in size and costs a few dimes. It replaced legacy devices in the latter 1990s. It is Universal because it can support many different kinds of devices, from coffee-pot warmers to disk drives to wifi adapters to audio playback. It is Serial, it only requires 4 wires. And it is a Bus, you can plug a USB device into an arbitrary port. It competed with FireWire, a very similar approach and championed by Apple, but won by a land-slide.

The only reason that serial ports are still relevant in on Windows these days is because a USB device requires a custom device driver. Device manufacturers do *not* like writing and supporting drivers, they often take a shortcut in their driver that makes it emulate a legacy serial port device. So programmers can use the legacy support for serial ports built into the operating system and about any language runtime library. Rather imperfect support btw, these emulators never support plug-and-play well. Discovering the specific serial port to open is very difficult. And these drivers often misbehave in impossible to diagnose ways when you jerk a USB device while your program is using it.

2.3 Male and Female Connectors



Generally, serial port connectors are gendered, only allowing connectors to mate with a connector of the opposite gender. With D-subminiature connectors, the male connectors have protruding pins, and female connectors have corresponding round sockets. Either type of connector can be mounted on equipment or a panel; or terminate a cable.

Connectors mounted on DTE are likely to be male, and those mounted on DCE are likely to be female (with the cable connectors being the opposite). However, this is far from universal; for instance, most serial printers have a female DB25 connector, but they are DTEs

While the RS-232 standard originally specified a 25-pin D-type connector, many designers of personal computers chose to implement only a subset of the full standard: they traded off compatibility with the standard against the use of less costly and more compact connectors (in particular the DE-9 version used by the original IBM PC-AT). The desire to supply serial interface cards with two ports required that IBM reduce the size of the connector to fit onto a single card back panel. A DE-9 connector also fits onto a card with a second DB-25 connector. Starting around the time of the introduction of the IBM PC-AT, serial ports were commonly built with a 9-pin connector to save cost and space. However, presence of a 9-pin D-subminiature connector is not sufficient to indicate the connection is in fact a serial port, since this connector is also used for video, joysticks, and other purposes.

Some miniaturized electronics, particularly graphing calculators and hand-held amateur and two-way radio equipment, have serial ports using a phone connector, usually the smaller 2.5 or 3.5 mm connectors and use the most basic 3-wire interface.

Many models of Macintosh favor the related RS-422 standard, mostly using German mini-DIN connectors, except in the earliest models. The Macintosh included a standard set of two ports for connection to a printer and a modem, but some PowerBook laptops had only one combined port to save space.

Since most devices do not use all of the 20 signals that are defined by the standard, smaller connectors are often used. For example, the 9-pin DE-9 connector is used by most IBM-compatible PCs since the IBM PC AT, and has been standardized as TIA-574. More recently, modular connectors have been used. Most common are 8P8C connectors, for which the EIA/TIA-561 standard defines a pinout, while the "Yost Serial Device Wiring Standard"[4] invented by Dave Yost (and popularized by the Unix System Administration Handbook) is common on Unix computers and newer devices from Cisco Systems. 10P10C connectors can be found on some devices as well. Digital Equipment Corporation defined their own DECconnect connection system which is based on the Modified Modular Jack (MMJ) connector. This is a 6-pin modular jack where the key is offset from the center position. As with the Yost standard, DECconnect uses a symmetrical pin layout which enables the direct connection between two DTEs. Another common connector is the DH10 header connector common on motherboards and add-in cards which is usually converted via a cable to the more standard 9-pin DE-9 connector (and frequently mounted on a free slot plate or other part of the housing).

2.4 Hardware Abstraction

Operating systems usually create symbolic names for the serial ports of a computer, rather than requiring programs to refer to them by hardware address.

Unix-like operating systems usually label the serial port devices /dev/tty*. TTY is a common trademark-free abbreviation for teletype, a device commonly attached to early computers' serial ports, and * represents a string identifying the specific port; the syntax of that string depends on the operating system and the device. On Linux, 8250/16550 UART hardware serial ports are named /dev/ttyS*, USB adapters appear as /dev/ttyUSB* and various types of virtual serial ports do not necessarily have names starting with tty.

The DOS and Windows environments refer to serial ports as COM ports: COM1, COM2,..etc. Ports numbered greater than COM9 should be referred to using the \\.\COM10 syntax.

The RS-232 standard is used by many specialized and custom-built devices. This list includes some of the more common devices that are connected to the serial port on a PC. Some of these such as modems and serial mice are falling into disuse while others are readily available.

Serial ports are very common on most types of microcontroller, where they can be used to communicate with a PC or other serial devices.

Dial-up modems

Configuration and management of networking equipment such as routers, switches, firewalls, load balancers

GPS receivers (typically NMEA 0183 at 4,800 bit/s)

Bar code scanners and other point of sale devices

LED and LCD text displays

Satellite phones, low-speed satellite modems and other satellite based transceiver devices

Flat-screen (LCD and Plasma) monitors to control screen functions by external computer, other AV components or remotes

Test and measuring equipment such as digital multimeters and weighing systems

Updating firmware on various consumer devices.

CNC controllers

Uninterruptible power supply

Stenography or Stenotype machines.

Software debuggers that run on a second computer.

Industrial field buses

Printers

Computer terminal, teletype

Older digital cameras

Networking (Macintosh AppleTalk using RS-422 at 230.4 kbit/s)

Serial mouse

Older GSM mobile phones

IDE hard drive[15][16] repair[17][18]

Since the control signals for a serial port can be easily turned on and off by a switch, some applications used the control lines of a serial port to monitor external devices, without exchanging serial data.

2.5 Serial Port PIN Information

Below is a listing of each of the pins located on the DB9 connector, their purpose, and signal name. As can be seen in the above picture pin one is in the top left and pin 9 is in the bottom right.

PIN	PURPOSE SIGNAL NAME
1	Data Carrier Detect DCD
2	Received Data RxData
3	Transmitted Data TxData
4	Data Terminal Ready DTR
5	Signal Ground Gnd
6	Data Set Ready DSR
7	Request To Send RTS
8	Clear To Send CTS
9	Ring Indicator RI

Note: Most of today's computers are eliminating the serial port in favor of USB ports.

3. IMPLEMENTATION

3.1 Bi-Directional Communication

In many circumstances a transmitter might be able to send data faster than the receiver is able to process it. To cope with this, serial lines often incorporate a "handshaking" method, usually distinguished between hardware and software handshaking.

Hardware handshaking is done with extra signals, often the RS-232 RTS/CTS or DTR/DSR signal circuits. Generally, the RTS and CTS are turned off and on from alternate ends to control data flow, for instance when a buffer is almost full. DTR and DSR are usually on all the time and, per the RS-232 standard and its successors, are used to signal from each end that the other equipment is actually present and powered-up. However, manufacturers have over the years built many devices that implemented non-standard variations on the standard, for example, printers that use DTR as flow control.

Software handshaking is done for example with ASCII control characters XON/XOFF to control the flow of data. The XON and XOFF characters are sent by the receiver to the sender to control when the sender will send data, that is, these characters go in the opposite direction to the data being sent. The circuit starts in the "sending allowed" state. When the receiver's buffers approach capacity, the receiver sends the XOFF character to tell the sender to stop sending data. Later, after the receiver has emptied its buffers, it sends an XON character to tell the sender to resume transmission. It is an example of in-band signaling, where control information is sent over the same channel as its data.

The advantage of hardware handshaking is that it can be extremely fast; it doesn't impose any particular meaning such as ASCII on the transferred data; and it is stateless. Its disadvantage is that it requires more hardware and cabling, and these must be compatible at both ends.

The advantage of software handshaking is that it can be done with absent or incompatible hardware handshaking circuits and cabling. The disadvantage, common to all in-band control signaling, is that it introduces complexities in ensuring that a) control messages get through even when data messages are blocked, and b) data can never be mistaken for control signals. The former is normally dealt with by the operating system or device driver; the latter normally by ensuring that control codes are "escaped" (such as in the Kermit protocol) or omitted by design (such as in ANSI terminal control).

If no handshaking is employed, an overrun receiver might simply fail to receive data from the transmitter. Approaches for preventing this include reducing the speed of the connection so that the receiver can always keep up; increasing the size of buffers so it can keep up averaged over a longer time; using delays after time-consuming operations (e.g. in termcap) or employing a mechanism to resend data which has been corrupted (e.g. TCP).

3.2 Universal Asynchronous Receiver/Transmitter (UART)

This 40-pin Dual Inline Package (DIP) chip is a variation of the National Semiconductor NS16550D UART chip

All computer operating systems in use today support serial ports, because serial ports have been around for decades. Parallel ports are a more recent invention and are much faster than serial ports. USB ports are only a few years old, and will likely replace both serial and parallel ports completely over the next several years.

The name "serial" comes from the fact that a serial port "serializes" data. That is, it takes a byte of data and transmits the 8 bits in the byte one at a time. The advantage is that a serial port needs only one wire to transmit the 8 bits (while a parallel port needs 8). The disadvantage is that it takes 8 times longer to transmit the data than it would if there were 8 wires. Serial ports lower cable costs and make cables smaller.

Before each byte of data, a serial port sends a start bit, which is a single bit with a value of 0. After each byte of data, it sends a stop bit to signal that the byte is complete. It may also send a parity bit.

Serial ports, also called communication (COM) ports, are bi-directional. Bi-directional communication allows each device to receive data as well as transmit it. Serial devices use different pins to receive and transmit data -- using the same pins would limit communication to half-duplex, meaning that information could only travel in one direction at a time. Using different pins allows for full-duplex communication, in which information can travel in both directions at once.

Serial ports rely on a special controller chip, the Universal Asynchronous Receiver/Transmitter (UART), to function properly. The UART chip takes the parallel output of the computer's system bus and transforms it into serial form for transmission through the serial port. In order to function faster, most UART chips have a built-in buffer of anywhere from 16 to 64 kilobytes. This buffer allows the chip to cache data coming in from the system bus while it is processing data going out to the serial port. While most standard serial ports have a maximum transfer rate of 115 Kbps (kilobits per second), high speed serial ports, such as Enhanced Serial Port (ESP) and Super Enhanced Serial Port (Super ESP), can reach data transfer rates of 460 Kbps.

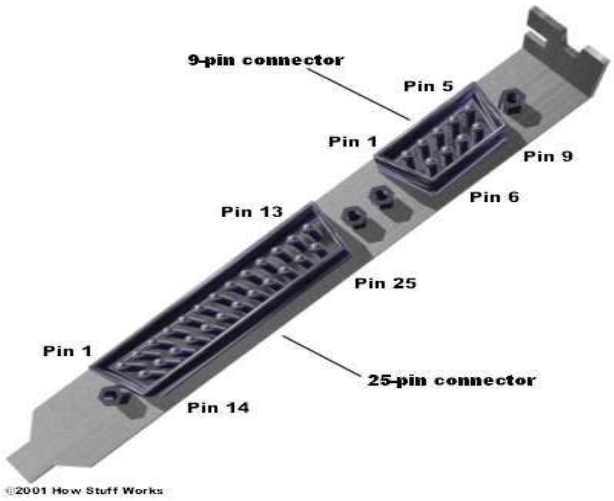
Transmitting and receiving UARTs must be set for the same bit speed, character length, parity, and stop bits for proper operation. The receiving UART may detect some mismatched settings and set a "framing error" flag bit for the host system; in exceptional cases the receiving UART will produce an erratic stream of mutilated characters and transfer them to the host system.

Typical serial ports used with personal computers connected to modems use eight data bits, no parity, and one stop bit; for this configuration the number of ASCII characters per second equals the bit rate divided by 10.

Some very low-cost home computers or embedded systems dispense with a UART and use the CPU to sample the state of an input port or directly manipulate an output port for data transmission. While very CPU-intensive (since the CPU timing is critical), the UART chip can thus be omitted, saving money and space. The technique is known as bit-banging.

3.3 The Serial Connection

The external connector for a serial port can be either 9 pins or 25 pins. Originally, the primary use of a serial port was to connect a modem to your computer. The pin assignments reflect that. Let's take a closer look at what happens at each pin when a modem is connected.



9-pin connector:

1. Carrier Detect - Determines if the modem is connected to a working phone line.
2. Receive Data - Computer receives information sent from the modem.
3. Transmit Data - Computer sends information to the modem.
4. Data Terminal Ready - Computer tells the modem that it is ready to talk.
5. Signal Ground - Pin is grounded.
6. Data Set Ready - Modem tells the computer that it is ready to talk.
7. Request To Send - Computer asks the modem if it can send information.
8. Clear To Send - Modem tells the computer that it can send information.
9. Ring Indicator - Once a call has been placed, computer acknowledges signal (sent from modem) that a ring is detected.

25-pin connector:

1. Not Used
2. Transmit Data - Computer sends information to the modem.
3. Receive Data - Computer receives information sent from the modem.
4. Request To Send - Computer asks the modem if it can send information.
5. Clear To Send - Modem tells the computer that it can send information.
6. Data Set Ready - Modem tells the computer that it is ready to talk.
7. Signal Ground - Pin is grounded.

8. Received Line Signal Detector - Determines if the modem is connected to a working phone line.
9. Not Used: Transmit Current Loop Return (+)
10. Not Used
11. Not Used: Transmit Current Loop Data (-)
12. Not Used
13. Not Used
14. Not Used
15. Not Used
16. Not Used
17. Not Used
18. Not Used: Receive Current Loop Data (+)
19. Not Used
20. Data Terminal Ready - Computer tells the modem that it is ready to talk.
21. Not Used
22. Ring Indicator - Once a call has been placed, computer acknowledges signal (sent from modem) that a ring is detected.
23. Not Used
24. Not Used
25. Not Used: Receive Current Loop Return (-)

3.4 Serial port vs Parallel Port

If you have a printer connected to your computer, there is a good chance that it uses the parallel port. While USB is becoming increasingly popular, the parallel port is still a commonly used interface for printers.

Parallel ports were originally developed by IBM as a way to connect a printer to your PC. When IBM was in the process of designing the PC, the company wanted the computer to work with printers offered by Centronics, a top printer manufacturer at the time. IBM decided not to use the same port interface on the computer that Centronics used on the printer.

Instead, IBM engineers coupled a 25-pin connector, DB-25, with a 36-pin Centronics connector to create a special cable to connect the printer to the computer. Other printer manufacturers ended up adopting the Centronics interface, making this strange hybrid cable an unlikely de facto standard.

Enhanced Parallel Port (EPP) was created by Intel, Xircom and Zenith in 1991. EPP allows for much more data, 500 kilobytes to 2 megabytes, to be transferred each second. It was targeted specifically for non-printer devices that would attach to the parallel port, particularly storage devices that needed the highest possible transfer rate.

Close on the heels of the introduction of EPP, Microsoft and Hewlett Packard jointly announced a specification called Extended Capabilities Port (ECP) in 1992. While EPP was geared toward other devices, ECP was designed to provide improved speed and functionality for printers.

In 1994, the IEEE 1284 standard was released. It included the two specifications for parallel port devices, EPP and ECP. In order for them to work, both the operating system and the device must support the required specification. This is seldom a problem today since most computers support SPP, ECP and EPP and will detect which mode needs to be used, depending on the attached device. If you need to manually select a mode, you can do so through the BIOS on most computers.

4. FEASIBILITY STUDY

4.1 Speed

Serial ports use two-level (binary) signaling, so the data rate in bits per second is equal to the symbol rate in baud. A standard series of rates is based on multiples of the rates for electromechanical teleprinters; some serial ports allow many arbitrary rates to be selected. The port speed and device speed must match. The capability to set a bit rate does not imply that a working connection will result. Not all bit rates are possible with all serial ports. Some special-purpose protocols such as MIDI for musical instrument control, use serial data rates other than the teleprinter series. Some serial port systems can automatically detect the bit rate.

The speed includes bits for framing (stop bits, parity, etc.) and so the effective data rate is lower than the bit transmission rate. For example, with 8-N-1 character framing only 80% of the bits are available for data (for every eight bits of data, two more framing bits are sent).

Bit rates commonly supported include 75, 110, 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bit/s.[20] Crystal oscillators with a frequency of 1.843200 MHz are sold specifically for this purpose. This is 16 times the fastest bit rate and the serial port circuit can easily divide this down to lower frequencies as required.

4.2 Data bits

The number of data bits in each character can be 5 (for Baudot code), 6 (rarely used), 7 (for true ASCII), 8 (for most kinds of data, as this size matches the size of a byte), or 9 (rarely used). 8 data bits are almost universally used in newer applications. 5 or 7 bits generally only make sense with older equipment such as teleprinters.

Most serial communications designs send the data bits within each byte LSB (least significant bit) first. This standard is also referred to as "little endian." Also possible, but rarely used, is "big endian" or MSB (most significant bit) first serial communications; this was used, for example, by the IBM 2741 printing terminal. (See Bit numbering for more about bit ordering.) The order of bits is not usually configurable within the serial port interface. To communicate with systems that require a different bit ordering than the local default, local software can re-order the bits within each byte just before sending and just after receiving.

5. PROJECT SPECIFICATIONS

5.1 Program Description

Program Description

Serial communication between the computer and the microcontroller follows the below steps:

Step1: Set the serial port mode.

Step2: Set the serial port baud rate.

Step3: Write and read serial port.

Step1

Inorder to configure the serial port, AT89S51/52 uses the SCON register. The SCON register consists of the following registers as in the figure: In the program for serial transmission and reception, SCON is entered with the value 0x40.

BIT	NAME	EXPLANATION
7	SM0	Serial port mode bit0
6	SM1	Serial port mode bit1
5	SM2	Multiprocessor Communication Enable
4	REN	Receiver Enable
3	TB8	Transmit Bit

2	RB8	Receiver Bit
1	TI	Transmit Flag
0	RI	Receive Flag

The bits SM0 and SM1 will determine the baud rate to be selected. Table below shows how the baud rate is selected.

SM0	SM1	BAUD RATE	EXPLANATION
0	0	8-bit shift register	Oscillator Frequency/12
0	1	9 bit UART	Set by Timer1
1	0	9 bit UART	Oscillator Frequency/64
1	1	9 bit UART	Set by Timer1

Step2

Baud Rate

Baud rate is defined as the rate at which the data is transferred in bits per second. Usually for serial communication the baud rate is set according to SM0 and SM1. If the value of SM0 and SM1 are “00” or “10”, then baud rate depends upon the oscillator frequency. In other cases, the baud rate depends upon the Timer 1 of the **microcontroller**.

The value that has to be placed in TH1 to generate the required baud rate depends upon the following equation.

TH1 = 256 - ((Crystal / 384) / Baud)

In the program, TH1 is set to the value, 0xFD such that the **baud rate** is 9600bps according to the above equation for a crystal frequency of 11.059MHz.

Step3

After step 1 and 2, the serial port can be used for transmission and reception of data. Hyper terminal is used for the reception and transmission of data through RS232.

- To open Hyper Terminal, go to Start Menu, Programs and Accessories.
- Click Communications.
- Select Hyper Terminal.
- Go to File.
- Click new connection.
- Give name to connection.
- Select Baud rate 9600bps.
- Parity as none, Data bits 8, Flow control none Stop bit 1.

In the program of serial transmission, the letter 'B' is transmitted through SBUF of the controller and is displayed on the hyper terminal of PC. While, in case of serial reception, data is entered on the hyper terminal and it is serially received through the SBUF of ***microcontroller***. Inorder to view the output on the microcontroller, in case of serial reception, **LCD** interfacing has to be done.

5.2 Program coding

For developing an algorithm for this project we will mostly rely on 'Java Version 9 Update 191' and the library we mostly use is " jSerialComm ".To access the contents of the library in my project we need to make sure to

```
import com.fazecast.jSerialComm.
```

*into our java files. You can then generate a list of all available serial ports on your system (real or virtual), by calling the following static method:

SerialPort.getCommPorts()

This will return an array of SerialPort objects through which you can iterate.

FRONT END IN HYPERTEMINAL PROGRAM

```
<Form version="1.3" maxVersion="1.9"
type="org.netbeans.modules.form.forminfo.JFrameFormInfo">
  <NonVisualComponents>
    <Component class="javax.swing.JColorChooser" name="jColorChooser1">
    </Component>
    <Container class="javax.swing.JPopupMenu" name="jPopupMenu1">


      <Layout
class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout">
        <Property name="useNullLayout" type="boolean" value="true"/>
      </Layout>
    </Container>
    <Component class="javax.swing.JMenuItem" name="jMenuItem1">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenuItem1"/>
      </Properties>
    </Component>
    <Component class="javax.swing.JMenuItem" name="jMenuItem2">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenuItem2"/>
      </Properties>
    </Component>
```

```
    <Component class="javax.swing.JMenuItem" name="jMenuItem3">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenuItem3"/>
      </Properties>
    </Component>
    <Container class="javax.swing.JMenu" name="jMenu1">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenu1"/>
      </Properties>

      <Layout
class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout">
        <Property name="useNullLayout" type="boolean" value="true"/>
      </Layout>
    </Container>
    <Container class="javax.swing.JPopupMenu" name="jPopupMenu2">

      <Layout
class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout">
        <Property name="useNullLayout" type="boolean" value="true"/>
      </Layout>
    </Container>
    <Component class="javax.swing.JMenuItem" name="jMenuItem8">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenuItem8"/>
      </Properties>
    </Component>
```

```
    <Container class="javax.swing.JMenuBar" name="jMenuBar2">

      <Layout
class="org.netbeans.modules.form.compat2.layouts.DesignAbsoluteLayout">
        <Property name="useNullLayout" type="boolean" value="true"/>
      </Layout>
      <SubComponents>
        <Menu class="javax.swing.JMenu" name="jMenu3">
          <Properties>
            <Property name="text" type="java.lang.String" value="File"/>
          </Properties>
        </Menu>
        <Menu class="javax.swing.JMenu" name="jMenu4">
          <Properties>
            <Property name="text" type="java.lang.String" value="Edit"/>
          </Properties>
        </Menu>
      </SubComponents>
    </Container>
    <Component class="javax.swing.JMenuItem" name="jMenuItem9">
      <Properties>
        <Property name="text" type="java.lang.String" value="jMenuItem9"/>
      </Properties>
    </Component>
    <Component class="javax.swing.JScrollBar" name="jScrollBar3">
    </Component>
    <Menu class="javax.swing.JMenuBar" name="jMenuBar1">
```

```
    <SubComponents>
      <Menu class="javax.swing.JMenu" name="jMenu2">
        <Properties>
          <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
            <Border info="org.netbeans.modules.form.compat2.border.EtchedBorderInfo">
              <EtchetBorder/>
            </Border>
          </Property>
          <Property name="text" type="java.lang.String" value="SETTING    "/>
          <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
            <Font name="Segoe UI" size="18" style="0"/>
          </Property>
        </Properties>
        <SubComponents>
          <MenuItem class="javax.swing.JMenuItem" name="jMenuItem6">
            <Properties>
              <Property name="text" type="java.lang.String" value="Set SSID"/>
            </Properties>
            <Events>
              <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem6ActionPerformed"/>
            </Events>
          </MenuItem>
          <MenuItem class="javax.swing.JMenuItem" name="jMenuItem7">
            <Properties>
```

```
            <Property name="text" type="java.lang.String" value="Set Password"/>
          </Properties>
          <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem7ActionPerformed"/>
          </Events>
        </MenuItem>
      </SubComponents>
    </Menu>
    <Menu class="javax.swing.JMenu" name="jMenu5">
      <Properties>
        <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
          <Border info="org.netbeans.modules.form.compat2.border.EtchedBorderInfo">
            <EtchetBorder/>
          </Border>
        </Property>
        <Property name="text" type="java.lang.String" value="INPUT TYPE"/>
        <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
          <Font name="Segoe UI" size="18" style="0"/>
        </Property>
      </Properties>
      <Events>
        <EventHandler event="menuSelected"
listener="javax.swing.event.MenuListener" parameters="javax.swing.event.MenuEvent"
handler="jMenu5MenuSelected"/>
      </Events>
```

```
        <SubComponents>
          <MenuItem class="javax.swing.JMenuItem" name="jMenuItem10">
            <Properties>
              <Property name="accelerator" type="javax.swing.KeyStroke"
editor="org.netbeans.modules.form.editors.KeyStrokeEditor">
                <KeyStroke key="Ctrl+D"/>
              </Property>
              <Property name="text" type="java.lang.String" value="DECIMAL"/>
              <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
                <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
                  <LineBorder/>
                </Border>
              </Property>
            </Properties>
            <Events>
              <EventHandler event="mouseClicked"
listener="java.awt.event.MouseListener" parameters="java.awt.event.MouseEvent"
handler="jMenuItem10MouseClicked"/>
              <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem10ActionPerformed"/>
            </Events>
          </MenuItem>
          <MenuItem class="javax.swing.JMenuItem" name="jMenuItem11">
            <Properties>
              <Property name="accelerator" type="javax.swing.KeyStroke"
editor="org.netbeans.modules.form.editors.KeyStrokeEditor">
                <KeyStroke key="Ctrl+B"/>
```

```
            </Property>
            <Property name="text" type="java.lang.String" value="BINARY"/>
            <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
              <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
                <LineBorder/>
              </Border>
            </Property>
          </Properties>
          <Events>
            <EventHandler event="mouseClicked"
listener="java.awt.event.MouseListener" parameters="java.awt.event.MouseEvent"
handler="jMenuItem11MouseClicked"/>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem11ActionPerformed"/>
            <EventHandler event="menuKeyPressed"
listener="javax.swing.event.MenuKeyListener"
parameters="javax.swing.event.MenuKeyEvent"
handler="jMenuItem11MenuKeyPressed"/>
          </Events>
        </MenuItem>
        <MenuItem class="javax.swing.JMenuItem" name="jMenuItem12">
          <Properties>
            <Property name="accelerator" type="javax.swing.KeyStroke"
editor="org.netbeans.modules.form.editors.KeyStrokeEditor">
              <KeyStroke key="Ctrl+H"/>
            </Property>
            <Property name="text" type="java.lang.String" value="HEXADECIMAL"/>
```

```
            <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
              <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
                <LineBorder/>
              </Border>
            </Property>
          </Properties>
          <Events>
            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem12ActionPerformed"/>
          </Events>
        </MenuItem>
        <MenuItem class="javax.swing.JMenuItem" name="jMenuItem13">
          <Properties>
            <Property name="accelerator" type="javax.swing.KeyStroke"
editor="org.netbeans.modules.form.editors.KeyStrokeEditor">
              <KeyStroke key="Ctrl+A"/>
            </Property>
            <Property name="text" type="java.lang.String" value="CHAR"/>
            <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
              <Border info="org.netbeans.modules.form.compat2.border.LineBorderInfo">
                <LineBorder/>
              </Border>
            </Property>
          </Properties>
          <Events>
```

```
                <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jMenuItem13ActionPerformed"/>
            </Events>
          </MenuItem>
        </SubComponents>
      </Menu>
    </SubComponents>
  </Menu>
 </NonVisualComponents>
 <Properties>
  <Property name="defaultCloseOperation" type="int" value="3"/>
 </Properties>
 <SyntheticProperties>
  <SyntheticProperty name="menuBar" type="java.lang.String" value="jMenuBar1"/>
  <SyntheticProperty name="formSizePolicy" type="int" value="1"/>
  <SyntheticProperty name="generateCenter" type="boolean" value="false"/>
 </SyntheticProperties>
 <AuxValues>
  <AuxValue name="FormSettings_autoResourcing" type="java.lang.Integer"
value="0"/>
  <AuxValue name="FormSettings_autoSetComponentName"
type="java.lang.Boolean" value="false"/>
  <AuxValue name="FormSettings_generateFQN" type="java.lang.Boolean"
value="true"/>
  <AuxValue name="FormSettings_generateMnemonicsCode"
type="java.lang.Boolean" value="false"/>
  <AuxValue name="FormSettings_i18nAutoMode" type="java.lang.Boolean"
value="false"/>
```

```
    <AuxValue name="FormSettings_layoutCodeTarget" type="java.lang.Integer"
value="1"/>
    <AuxValue name="FormSettings_listenerGenerationStyle" type="java.lang.Integer"
value="0"/>
    <AuxValue name="FormSettings_variablesLocal" type="java.lang.Boolean"
value="false"/>
    <AuxValue name="FormSettings_variablesModifier" type="java.lang.Integer"
value="2"/>
  </AuxValues>

  <Layout>
    <DimensionLayout dim="0">
      <Group type="103" groupAlignment="0" attributes="0">
          <Component id="jPanel2" alignment="0" max="32767" attributes="0"/>
          <Component id="jPanel3" alignment="0" max="32767" attributes="0"/>
          <Component id="jPanel1" alignment="0" max="32767" attributes="0"/>
      </Group>
    </DimensionLayout>
    <DimensionLayout dim="1">
      <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="0" attributes="0">
              <Component id="jPanel1" min="-2" max="-2" attributes="0"/>
              <EmptySpace min="0" pref="0" max="-2" attributes="0"/>
              <Component id="jPanel2" min="-2" max="-2" attributes="0"/>
              <EmptySpace max="-2" attributes="0"/>
              <Component id="jPanel3" max="32767" attributes="0"/>
              <EmptySpace max="32767" attributes="0"/>
          </Group>
```

```
    </Group>
  </DimensionLayout>
 </Layout>
 <SubComponents>
  <Container class="javax.swing.JPanel" name="jPanel1">

    <Layout>
      <DimensionLayout dim="0">
        <Group type="103" groupAlignment="0" attributes="0">
            <Component id="jScrollPane1" alignment="0" max="32767" attributes="0"/>
        </Group>
      </DimensionLayout>
      <DimensionLayout dim="1">
        <Group type="103" groupAlignment="0" attributes="0">
            <Group type="102" alignment="0" attributes="0">
                <Component id="jScrollPane1" min="-2" pref="362" max="-2"
attributes="0"/>
                <EmptySpace min="0" pref="0" max="32767" attributes="0"/>
            </Group>
        </Group>
      </DimensionLayout>
    </Layout>
    <SubComponents>
      <Container class="javax.swing.JScrollPane" name="jScrollPane1">
        <AuxValues>
          <AuxValue name="autoScrollPane" type="java.lang.Boolean" value="true"/>
        </AuxValues>
```

```
        <Layout
class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout
"/>

        <SubComponents>

          <Component class="javax.swing.JTextArea" name="jTextArea1">

            <Properties>

              <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">

                <Color blue="0" green="0" red="0" type="rgb"/>

              </Property>

              <Property name="columns" type="int" value="20"/>

              <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">

                <Font name="Monospaced" size="24" style="0"/>

              </Property>

              <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">

                <Color blue="0" green="99" red="0" type="rgb"/>

              </Property>

              <Property name="rows" type="int" value="5"/>

              <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">

                <Border
info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">

                  <TitledBorder title="OUTPUT">

                    <Font PropertyName="font" name="Verdana" size="18" style="1"/>

                    <Color PropertyName="color" blue="0" green="99" red="0" type="rgb"/>

                  </TitledBorder>
```

```
                </Border>
              </Property>
            </Properties>
            <Events>
              <EventHandler event="keyPressed" listener="java.awt.event.KeyListener"
parameters="java.awt.event.KeyEvent" handler="jTextArea1KeyPressed"/>
            </Events>
          </Component>
        </SubComponents>
      </Container>
    </SubComponents>
  </Container>
  <Container class="javax.swing.JPanel" name="jPanel2">

    <Layout>
      <DimensionLayout dim="0">
        <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="0" attributes="0">
            <Component id="jScrollPane2" max="32767" attributes="0"/>
            <EmptySpace min="-2" pref="0" max="-2" attributes="0"/>
          </Group>
        </Group>
      </DimensionLayout>
      <DimensionLayout dim="1">
        <Group type="103" groupAlignment="0" attributes="0">
          <Group type="102" alignment="0" attributes="0">
```

```
                <Component id="jScrollPane2" min="-2" pref="209" max="-2"
attributes="0"/>
                <EmptySpace min="0" pref="0" max="32767" attributes="0"/>
            </Group>
        </Group>
      </DimensionLayout>
    </Layout>
    <SubComponents>
      <Container class="javax.swing.JScrollPane" name="jScrollPane2">
        <AuxValues>
          <AuxValue name="autoScrollPane" type="java.lang.Boolean" value="true"/>
        </AuxValues>

        <Layout
class="org.netbeans.modules.form.compat2.layouts.support.JScrollPaneSupportLayout
"/>
        <SubComponents>
          <Component class="javax.swing.JTextArea" name="jTextArea2">
            <Properties>
              <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
                <Color blue="0" green="0" red="0" type="rgb"/>
              </Property>
              <Property name="columns" type="int" value="20"/>
              <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
                <Font name="Monospaced" size="24" style="0"/>
              </Property>
```

```
            <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
              <Color blue="0" green="99" red="0" type="rgb"/>
            </Property>
            <Property name="rows" type="int" value="5"/>
            <Property name="border" type="javax.swing.border.Border"
editor="org.netbeans.modules.form.editors2.BorderEditor">
              <Border
info="org.netbeans.modules.form.compat2.border.TitledBorderInfo">
                <TitledBorder title="INPUT">
                  <Font PropertyName="font" name="Verdana" size="18" style="1"/>
                  <Color PropertyName="color" blue="0" green="99" red="0" type="rgb"/>
                </TitledBorder>
              </Border>
            </Property>
            <Property name="cursor" type="java.awt.Cursor"
editor="org.netbeans.modules.form.editors2.CursorEditor">
              <Color id="Text Cursor"/>
            </Property>
          </Properties>
        </Component>
      </SubComponents>
    </Container>
  </SubComponents>
 </Container>
 <Container class="javax.swing.JPanel" name="jPanel3">

  <Layout>
```

```
        <DimensionLayout dim="0">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" alignment="0" attributes="0">
                  <EmptySpace min="-2" pref="47" max="-2" attributes="0"/>
                  <Group type="103" groupAlignment="0" max="-2" attributes="0">
                      <Component id="jButton7" max="32767" attributes="0"/>
                      <Component id="jButton1" max="32767" attributes="0"/>
                  </Group>
                  <EmptySpace min="-2" pref="50" max="-2" attributes="0"/>
                  <Component id="jButton2" min="-2" max="-2" attributes="0"/>
                  <EmptySpace min="-2" pref="50" max="-2" attributes="0"/>
                  <Component id="jButton8" min="-2" max="-2" attributes="0"/>
                  <EmptySpace min="-2" pref="50" max="-2" attributes="0"/>
                  <Component id="jButton9" min="-2" pref="186" max="-2" attributes="0"/>
                  <EmptySpace pref="56" max="32767" attributes="0"/>
              </Group>
          </Group>
        </DimensionLayout>
        <DimensionLayout dim="1">
          <Group type="103" groupAlignment="0" attributes="0">
              <Group type="102" attributes="0">
                  <EmptySpace min="-2" pref="2" max="-2" attributes="0"/>
                  <Group type="103" groupAlignment="3" attributes="0">
                      <Component id="jButton1" alignment="3" min="-2" max="-2"
attributes="0"/>
                      <Component id="jButton2" alignment="3" min="-2" max="-2"
attributes="0"/>
```

```
                    <Component id="jButton8" alignment="3" min="-2" max="-2"
attributes="0"/>
                    <Component id="jButton9" alignment="3" min="-2" max="-2"
attributes="0"/>
                </Group>
                <EmptySpace type="separate" max="-2" attributes="0"/>
                <Component id="jButton7" max="32767" attributes="0"/>
                <EmptySpace max="32767" attributes="0"/>
            </Group>
        </Group>
    </DimensionLayout>
</Layout>
<SubComponents>
    <Component class="javax.swing.JButton" name="jButton1">
        <Properties>
            <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
                <Color blue="33" green="33" red="33" type="rgb"/>
            </Property>
            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
                <Font name="Tahoma" size="18" style="1"/>
            </Property>
            <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
                <Color blue="ff" green="ff" red="ff" type="rgb"/>
            </Property>
            <Property name="text" type="java.lang.String" value="CONNECT"/>
```

```
        </Properties>
        <Events>
          <EventHandler event="mouseClicked" listener="java.awt.event.MouseListener"
parameters="java.awt.event.MouseEvent" handler="jButton1MouseClicked"/>
          <EventHandler event="mouseEntered" listener="java.awt.event.MouseListener"
parameters="java.awt.event.MouseEvent" handler="jButton1MouseEntered"/>
        </Events>
        <AuxValues>
          <AuxValue name="JavaCodeGenerator_VariableModifier"
type="java.lang.Integer" value="1"/>
        </AuxValues>
      </Component>
      <Component class="javax.swing.JButton" name="jButton2">
        <Properties>
          <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
            <Color blue="0" green="0" red="0" type="rgb"/>
          </Property>
          <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
            <Font name="Tahoma" size="18" style="1"/>
          </Property>
          <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
            <Color blue="ff" green="ff" red="ff" type="rgb"/>
          </Property>
          <Property name="text" type="java.lang.String" value="DISCONNECT"/>
        </Properties>
        <Events>
```

```
            <EventHandler event="mouseClicked" listener="java.awt.event.MouseListener"
parameters="java.awt.event.MouseEvent" handler="jButton2MouseClicked"/>

          </Events>

        </Component>

        <Component class="javax.swing.JButton" name="jButton7">

          <Properties>

            <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">

              <Color blue="0" green="0" red="ff" type="rgb"/>

            </Property>

            <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">

              <Font name="Tahoma" size="18" style="1"/>

            </Property>

            <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">

              <Color blue="ff" green="ff" red="ff" type="rgb"/>

            </Property>

            <Property name="text" type="java.lang.String" value="            "/>

          </Properties>

          <Events>

            <EventHandler event="actionPerformed"
listener="java.awt.event.ActionListener" parameters="java.awt.event.ActionEvent"
handler="jButton7ActionPerformed"/>

            <EventHandler event="propertyChange"
listener="java.beans.PropertyChangeListener"
parameters="java.beans.PropertyChangeEvent" handler="jButton7PropertyChange"/>

          </Events>

        </Component>

        <Component class="javax.swing.JButton" name="jButton8">
```

```
        <Properties>
          <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
            <Color blue="0" green="0" red="0" type="rgb"/>
          </Property>
          <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
            <Font name="Tahoma" size="18" style="1"/>
          </Property>
          <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
            <Color blue="ff" green="ff" red="ff" type="rgb"/>
          </Property>
          <Property name="text" type="java.lang.String" value="CLEAR OUTPUT"/>
        </Properties>
        <Events>
          <EventHandler event="mouseClicked" listener="java.awt.event.MouseListener"
parameters="java.awt.event.MouseEvent" handler="jButton8MouseClicked"/>
        </Events>
      </Component>
      <Component class="javax.swing.JButton" name="jButton9">
        <Properties>
          <Property name="background" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
            <Color blue="0" green="0" red="0" type="rgb"/>
          </Property>
          <Property name="font" type="java.awt.Font"
editor="org.netbeans.beaninfo.editors.FontEditor">
            <Font name="Tahoma" size="18" style="1"/>
```

```
            </Property>
            <Property name="foreground" type="java.awt.Color"
editor="org.netbeans.beaninfo.editors.ColorEditor">
              <Color blue="ff" green="ff" red="ff" type="rgb"/>
            </Property>
            <Property name="text" type="java.lang.String" value="CLEAR INPUT"/>
          </Properties>
          <Events>
            <EventHandler event="mouseClicked" listener="java.awt.event.MouseListener"
parameters="java.awt.event.MouseEvent" handler="jButton9MouseClicked"/>
          </Events>
        </Component>
      </SubComponents>
    </Container>
  </SubComponents>
</Form>
```

BACK END PROGRAMMING FOR HYPERTERMINAL

For the purpose of easier reading the script

is divided into following sections:

- initialization
- compilation
- jar
- execution
- debugging
- javadoc
- test compilation
- test execution
- test debugging
- applet
- cleanup

-->

```
<project xmlns:j2seproject1="http://www.netbeans.org/ns/j2se-project/1"
xmlns:j2seproject3="http://www.netbeans.org/ns/j2se-project/3"
xmlns:jaxrpc="http://www.netbeans.org/ns/j2se-project/jax-rpc" basedir=".." default="default"
name="GUI_PLAY-impl">
    <fail message="Please build using Ant 1.8.0 or higher.">
        <condition>
```

```
            <not>
                <antversion atleast="1.8.0"/>
            </not>
        </condition>
    </fail>
    <target depends="test,jar,javadoc" description="Build and test whole project."
name="default"/>
    <!--
            ======================
            INITIALIZATION SECTION
            ======================
        -->
    <target name="-pre-init">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target depends="-pre-init" name="-init-private">
        <property file="nbproject/private/config.properties"/>
        <property file="nbproject/private/configs/${config}.properties"/>
        <property file="nbproject/private/private.properties"/>
    </target>
    <target depends="-pre-init,-init-private" name="-init-user">
        <property file="${user.properties.file}"/>
        <!-- The two properties below are usually overridden -->
        <!-- by the active platform. Just a fallback. -->
        <property name="default.javac.source" value="1.4"/>
        <property name="default.javac.target" value="1.4"/>
    </target>
    <target depends="-pre-init,-init-private,-init-user" name="-init-project">
```

```
        <property file="nbproject/configs/${config}.properties"/>
        <property file="nbproject/project.properties"/>
    </target>
    <target depends="-pre-init,-init-private,-init-user,-init-project,-init-macrodef-property"
name="-do-init">
        <property name="platform.java" value="${java.home}/bin/java"/>
        <available file="${manifest.file}" property="manifest.available"/>
        <condition property="splashscreen.available">
            <and>
                <not>
                    <equals arg1="${application.splash}" arg2="" trim="true"/>
                </not>
                <available file="${application.splash}"/>
            </and>
        </condition>
        <condition property="main.class.available">
            <and>
                <isset property="main.class"/>
                <not>
                    <equals arg1="${main.class}" arg2="" trim="true"/>
                </not>
            </and>
        </condition>
        <condition property="profile.available">
            <and>
                <isset property="javac.profile"/>
                <length length="0" string="${javac.profile}" when="greater"/>
                <matches pattern="1\.[89](\..*)?" string="${javac.source}"/>
            </and>
```

```
</condition>
<condition property="do.archive">
    <or>
        <not>
            <istrue value="${jar.archive.disabled}"/>
        </not>
        <istrue value="${not.archive.disabled}"/>
    </or>
</condition>
<condition property="do.mkdist">
    <and>
        <isset property="do.archive"/>
        <isset property="libs.CopyLibs.classpath"/>
        <not>
            <istrue value="${mkdist.disabled}"/>
        </not>
    </and>
</condition>
<condition property="do.archive+manifest.available">
    <and>
        <isset property="manifest.available"/>
        <istrue value="${do.archive}"/>
    </and>
</condition>
<condition property="do.archive+main.class.available">
    <and>
        <isset property="main.class.available"/>
        <istrue value="${do.archive}"/>
    </and>
```

```
</condition>
<condition property="do.archive+splashscreen.available">
    <and>
        <isset property="splashscreen.available"/>
        <istrue value="${do.archive}"/>
    </and>
</condition>
<condition property="do.archive+profile.available">
    <and>
        <isset property="profile.available"/>
        <istrue value="${do.archive}"/>
    </and>
</condition>
<condition property="have.tests">
    <or>
        <available file="${test.src.dir}"/>
    </or>
</condition>
<condition property="have.sources">
    <or>
        <available file="${src.dir}"/>
    </or>
</condition>
<condition property="netbeans.home+have.tests">
    <and>
        <isset property="netbeans.home"/>
        <isset property="have.tests"/>
    </and>
</condition>
```

```
<condition property="no.javadoc.preview">
    <and>
        <isset property="javadoc.preview"/>
        <isfalse value="${javadoc.preview}"/>
    </and>
</condition>
<property name="run.jvmargs" value=""/>
<property name="run.jvmargs.ide" value=""/>
<property name="javac.compilerargs" value=""/>
<property name="work.dir" value="${basedir}"/>
<condition property="no.deps">
    <and>
        <istrue value="${no.dependencies}"/>
    </and>
</condition>
<property name="javac.debug" value="true"/>
<property name="javadoc.preview" value="true"/>
<property name="application.args" value=""/>
<property name="source.encoding" value="${file.encoding}"/>
<property name="runtime.encoding" value="${source.encoding}"/>
<condition property="javadoc.encoding.used" value="${javadoc.encoding}">
    <and>
        <isset property="javadoc.encoding"/>
        <not>
            <equals arg1="${javadoc.encoding}" arg2=""/>
        </not>
    </and>
</condition>
<property name="javadoc.encoding.used" value="${source.encoding}"/>
```

```
    <property name="includes" value="**"/>
    <property name="excludes" value=""/>
    <property name="do.depend" value="false"/>
    <condition property="do.depend.true">
      <istrue value="${do.depend}"/>
    </condition>
    <path id="endorsed.classpath.path" path="${endorsed.classpath}"/>
    <condition else="" property="endorsed.classpath.cmd.line.arg"
value="-Xbootclasspath/p:'${toString:endorsed.classpath.path}'">
      <and>
        <isset property="endorsed.classpath"/>
        <not>
          <equals arg1="${endorsed.classpath}" arg2="" trim="true"/>
        </not>
      </and>
    </condition>
    <condition else="" property="javac.profile.cmd.line.arg" value="-profile ${javac.profile}">
      <isset property="profile.available"/>
    </condition>
    <condition else="false" property="jdkBug6558476">
      <and>
        <matches pattern="1\.[56]" string="${java.specification.version}"/>
        <not>
          <os family="unix"/>
        </not>
      </and>
    </condition>
    <condition else="false" property="javac.fork">
      <or>
```

```
        <istrue value="${jdkBug6558476}"/>
        <istrue value="${javac.external.vm}"/>
    </or>
</condition>
<property name="jar.index" value="false"/>
<property name="jar.index.metainf" value="${jar.index}"/>
<property name="copylibs.rebase" value="true"/>
<available file="${meta.inf.dir}/persistence.xml" property="has.persistence.xml"/>
<condition property="junit.available">
    <or>
        <available classname="org.junit.Test" classpath="${run.test.classpath}"/>
        <available classname="junit.framework.Test" classpath="${run.test.classpath}"/>
    </or>
</condition>
<condition property="testng.available">
    <available classname="org.testng.annotations.Test" classpath="${run.test.classpath}"/>
</condition>
<condition property="junit+testng.available">
    <and>
        <istrue value="${junit.available}"/>
        <istrue value="${testng.available}"/>
    </and>
</condition>
<condition else="testng" property="testng.mode" value="mixed">
    <istrue value="${junit+testng.available}"/>
</condition>
<condition else="" property="testng.debug.mode" value="-mixed">
    <istrue value="${junit+testng.available}"/>
</condition>
```

```
        <property name="java.failonerror" value="true"/>
    </target>
    <target name="-post-init">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target depends="-pre-init,-init-private,-init-user,-init-project,-do-init" name="-init-check">
        <fail unless="src.dir">Must set src.dir</fail>
        <fail unless="test.src.dir">Must set test.src.dir</fail>
        <fail unless="build.dir">Must set build.dir</fail>
        <fail unless="dist.dir">Must set dist.dir</fail>
        <fail unless="build.classes.dir">Must set build.classes.dir</fail>
        <fail unless="dist.javadoc.dir">Must set dist.javadoc.dir</fail>
        <fail unless="build.test.classes.dir">Must set build.test.classes.dir</fail>
        <fail unless="build.test.results.dir">Must set build.test.results.dir</fail>
        <fail unless="build.classes.excludes">Must set build.classes.excludes</fail>
        <fail unless="dist.jar">Must set dist.jar</fail>
    </target>
    <target name="-init-macrodef-property">
        <macrodef name="property" uri="http://www.netbeans.org/ns/j2se-project/1">
            <attribute name="name"/>
            <attribute name="value"/>
            <sequential>
                <property name="@{name}" value="${@{value}}"/>
            </sequential>
        </macrodef>
    </target>
    <target depends="-init-ap-cmdline-properties" if="ap.supported.internal"
name="-init-macrodef-javac-with-processors">
```

```
        <macrodef name="javac" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${src.dir}" name="srcdir"/>
            <attribute default="${build.classes.dir}" name="destdir"/>
            <attribute default="${javac.classpath}" name="classpath"/>
            <attribute default="${javac.processorpath}" name="processorpath"/>
            <attribute default="${build.generated.sources.dir}/ap-source-output"
name="apgeneratedsrcdir"/>
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
            <attribute default="${javac.debug}" name="debug"/>
            <attribute default="${empty.dir}" name="sourcepath"/>
            <attribute default="${empty.dir}" name="gensrcdir"/>
            <element name="customize" optional="true"/>
            <sequential>
                <property location="${build.dir}/empty" name="empty.dir"/>
                <mkdir dir="${empty.dir}"/>
                <mkdir dir="@{apgeneratedsrcdir}"/>
                <javac debug="@{debug}" deprecation="${javac.deprecation}" destdir="@{destdir}"
encoding="${source.encoding}" excludes="@{excludes}" fork="${javac.fork}"
includeantruntime="false" includes="@{includes}" source="${javac.source}"
sourcepath="@{sourcepath}" srcdir="@{srcdir}" target="${javac.target}"
tempdir="${java.io.tmpdir}">
                    <src>
                        <dirset dir="@{gensrcdir}" erroronmissingdir="false">
                            <include name="*"/>
                        </dirset>
                    </src>
                    <classpath>
                        <path path="@{classpath}"/>
                    </classpath>
                    <compilerarg line="${endorsed.classpath.cmd.line.arg}"/>
```

```
                <compilerarg line="${javac.profile.cmd.line.arg}"/>
                <compilerarg line="${javac.compilerargs}"/>
                <compilerarg value="-processorpath"/>
                <compilerarg path="@{processorpath}:${empty.dir}"/>
                <compilerarg line="${ap.processors.internal}"/>
                <compilerarg line="${annotation.processing.processor.options}"/>
                <compilerarg value="-s"/>
                <compilerarg path="@{apgeneratedsrcdir}"/>
                <compilerarg line="${ap.proc.none.internal}"/>
                <customize/>
            </javac>
        </sequential>
    </macrodef>
</target>
<target depends="-init-ap-cmdline-properties"
name="-init-macrodef-javac-without-processors" unless="ap.supported.internal">
    <macrodef name="javac" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${src.dir}" name="srcdir"/>
        <attribute default="${build.classes.dir}" name="destdir"/>
        <attribute default="${javac.classpath}" name="classpath"/>
        <attribute default="${javac.processorpath}" name="processorpath"/>
        <attribute default="${build.generated.sources.dir}/ap-source-output"
name="apgeneratedsrcdir"/>
        <attribute default="${includes}" name="includes"/>
        <attribute default="${excludes}" name="excludes"/>
        <attribute default="${javac.debug}" name="debug"/>
        <attribute default="${empty.dir}" name="sourcepath"/>
        <attribute default="${empty.dir}" name="gensrcdir"/>
        <element name="customize" optional="true"/>
        <sequential>
```

```
            <property location="${build.dir}/empty" name="empty.dir"/>

            <mkdir dir="${empty.dir}"/>

            <javac debug="@{debug}" deprecation="${javac.deprecation}" destdir="@{destdir}"
encoding="${source.encoding}" excludes="@{excludes}" fork="${javac.fork}"
includeantruntime="false" includes="@{includes}" source="${javac.source}"
sourcepath="@{sourcepath}" srcdir="@{srcdir}" target="${javac.target}"
tempdir="${java.io.tmpdir}">

                <src>

                    <dirset dir="@{gensrcdir}" erroronmissingdir="false">

                        <include name="*"/>

                    </dirset>

                </src>

                <classpath>

                    <path path="@{classpath}"/>

                </classpath>

                <compilerarg line="${endorsed.classpath.cmd.line.arg}"/>

                <compilerarg line="${javac.profile.cmd.line.arg}"/>

                <compilerarg line="${javac.compilerargs}"/>

                <customize/>

            </javac>

        </sequential>

    </macrodef>

</target>

<target
depends="-init-macrodef-javac-with-processors,-init-macrodef-javac-without-processors"
name="-init-macrodef-javac">

    <macrodef name="depend" uri="http://www.netbeans.org/ns/j2se-project/3">

        <attribute default="${src.dir}" name="srcdir"/>

        <attribute default="${build.classes.dir}" name="destdir"/>

        <attribute default="${javac.classpath}" name="classpath"/>

        <sequential>
```

```
            <depend cache="${build.dir}/depcache" destdir="@{destdir}" excludes="${excludes}"
includes="${includes}" srcdir="@{srcdir}">
                <classpath>
                    <path path="@{classpath}"/>
                </classpath>
            </depend>
        </sequential>
    </macrodef>
    <macrodef name="force-recompile" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${build.classes.dir}" name="destdir"/>
        <sequential>
            <fail unless="javac.includes">Must set javac.includes</fail>
            <pathconvert pathsep="${line.separator}" property="javac.includes.binary">
                <path>
                    <filelist dir="@{destdir}" files="${javac.includes}"/>
                </path>
                <globmapper from="*.java" to="*.class"/>
            </pathconvert>
            <tempfile deleteonexit="true" property="javac.includesfile.binary"/>
            <echo file="${javac.includesfile.binary}" message="${javac.includes.binary}"/>
            <delete>
                <files includesfile="${javac.includesfile.binary}"/>
            </delete>
            <delete>
                <fileset file="${javac.includesfile.binary}"/>
            </delete>
        </sequential>
    </macrodef>
</target>
```

```
<target if="${junit.available}" name="-init-macrodef-junit-init">
    <condition else="false" property="nb.junit.batch" value="true">
        <and>
            <istrue value="${junit.available}"/>
            <not>
                <isset property="test.method"/>
            </not>
        </and>
    </condition>
    <condition else="false" property="nb.junit.single" value="true">
        <and>
            <istrue value="${junit.available}"/>
            <isset property="test.method"/>
        </and>
    </condition>
</target>
<target name="-init-test-properties">
    <property name="test.binaryincludes" value="&lt;nothing&gt;"/>
    <property name="test.binarytestincludes" value=""/>
    <property name="test.binaryexcludes" value=""/>
</target>
<target if="${nb.junit.single}" name="-init-macrodef-junit-single" unless="${nb.junit.batch}">
    <macrodef name="junit" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${includes}" name="includes"/>
        <attribute default="${excludes}" name="excludes"/>
        <attribute default="**" name="testincludes"/>
        <attribute default="" name="testmethods"/>
        <element name="customize" optional="true"/>
        <sequential>
```

```
            <property name="junit.forkmode" value="perTest"/>

            <junit dir="${work.dir}" errorproperty="tests.failed" failureproperty="tests.failed"
fork="true" forkmode="${junit.forkmode}" showoutput="true" tempdir="${build.dir}">

                <test methods="@{testmethods}" name="@{testincludes}"
todir="${build.test.results.dir}"/>

                <syspropertyset>

                    <propertyref prefix="test-sys-prop."/>

                    <mapper from="test-sys-prop.*" to="*" type="glob"/>

                </syspropertyset>

                <formatter type="brief" usefile="false"/>

                <formatter type="xml"/>

                <jvmarg value="-ea"/>

                <customize/>

            </junit>

        </sequential>

    </macrodef>

</target>

<target depends="-init-test-properties" if="${nb.junit.batch}"
name="-init-macrodef-junit-batch" unless="${nb.junit.single}">

    <macrodef name="junit" uri="http://www.netbeans.org/ns/j2se-project/3">

        <attribute default="${includes}" name="includes"/>

        <attribute default="${excludes}" name="excludes"/>

        <attribute default="**" name="testincludes"/>

        <attribute default="" name="testmethods"/>

        <element name="customize" optional="true"/>

        <sequential>

            <property name="junit.forkmode" value="perTest"/>

            <junit dir="${work.dir}" errorproperty="tests.failed" failureproperty="tests.failed"
fork="true" forkmode="${junit.forkmode}" showoutput="true" tempdir="${build.dir}">

                <batchtest todir="${build.test.results.dir}">
```

```
                    <fileset dir="${test.src.dir}" excludes="@{excludes},${excludes}"
includes="@{includes}">
                        <filename name="@{testincludes}"/>
                    </fileset>
                    <fileset dir="${build.test.classes.dir}"
excludes="@{excludes},${excludes},${test.binaryexcludes}" includes="${test.binaryincludes}">
                        <filename name="${test.binarytestincludes}"/>
                    </fileset>
                </batchtest>
                <syspropertyset>
                    <propertyref prefix="test-sys-prop."/>
                    <mapper from="test-sys-prop.*" to="*" type="glob"/>
                </syspropertyset>
                <formatter type="brief" usefile="false"/>
                <formatter type="xml"/>
                <jvmarg value="-ea"/>
                <customize/>
            </junit>
        </sequential>
    </macrodef>
</target>
<target depends="-init-macrodef-junit-init,-init-macrodef-junit-single,
-init-macrodef-junit-batch" if="${junit.available}" name="-init-macrodef-junit"/>
<target if="${testng.available}" name="-init-macrodef-testng">
    <macrodef name="testng" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${includes}" name="includes"/>
        <attribute default="${excludes}" name="excludes"/>
        <attribute default="**" name="testincludes"/>
        <attribute default="" name="testmethods"/>
        <element name="customize" optional="true"/>
```

```
            <sequential>
                <condition else="" property="testng.methods.arg"
value="@{testincludes}.@{testmethods}">
                    <isset property="test.method"/>
                </condition>
                <union id="test.set">
                    <fileset dir="${test.src.dir}" excludes="@{excludes},**/*.xml,${excludes}"
includes="@{includes}">
                        <filename name="@{testincludes}"/>
                    </fileset>
                </union>
                <taskdef classname="org.testng.TestNGAntTask" classpath="${run.test.classpath}"
name="testng"/>
                <testng classfilesetref="test.set" failureProperty="tests.failed"
listeners="org.testng.reporters.VerboseReporter" methods="${testng.methods.arg}"
mode="${testng.mode}" outputdir="${build.test.results.dir}" suitename="GUI_PLAY"
testname="TestNG tests" workingDir="${work.dir}">
                    <xmlfileset dir="${build.test.classes.dir}" includes="@{testincludes}"/>
                    <propertyset>
                        <propertyref prefix="test-sys-prop."/>
                        <mapper from="test-sys-prop.*" to="*" type="glob"/>
                    </propertyset>
                    <customize/>
                </testng>
            </sequential>
        </macrodef>
    </target>
    <target name="-init-macrodef-test-impl">
        <macrodef name="test-impl" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
```

```
            <attribute default="**" name="testincludes"/>
            <attribute default="" name="testmethods"/>
            <element implicit="true" name="customize" optional="true"/>
            <sequential>
                <echo>No tests executed.</echo>
            </sequential>
        </macrodef>
    </target>
    <target depends="-init-macrodef-junit" if="${junit.available}"
name="-init-macrodef-junit-impl">
        <macrodef name="test-impl" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
            <attribute default="**" name="testincludes"/>
            <attribute default="" name="testmethods"/>
            <element implicit="true" name="customize" optional="true"/>
            <sequential>
                <j2seproject3:junit excludes="@{excludes}" includes="@{includes}"
testincludes="@{testincludes}" testmethods="@{testmethods}">
                    <customize/>
                </j2seproject3:junit>
            </sequential>
        </macrodef>
    </target>
    <target depends="-init-macrodef-testng" if="${testng.available}"
name="-init-macrodef-testng-impl">
        <macrodef name="test-impl" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
            <attribute default="**" name="testincludes"/>
```

```
        <attribute default="" name="testmethods"/>
        <element implicit="true" name="customize" optional="true"/>
        <sequential>
            <j2seproject3:testng excludes="@{excludes}" includes="@{includes}"
testincludes="@{testincludes}" testmethods="@{testmethods}">
                <customize/>
            </j2seproject3:testng>
        </sequential>
    </macrodef>
  </target>
  <target
depends="-init-macrodef-test-impl,-init-macrodef-junit-impl,-init-macrodef-testng-impl"
name="-init-macrodef-test">
    <macrodef name="test" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${includes}" name="includes"/>
        <attribute default="${excludes}" name="excludes"/>
        <attribute default="**" name="testincludes"/>
        <attribute default="" name="testmethods"/>
        <sequential>
            <j2seproject3:test-impl excludes="@{excludes}" includes="@{includes}"
testincludes="@{testincludes}" testmethods="@{testmethods}">
                <customize>
                    <classpath>
                        <path path="${run.test.classpath}"/>
                    </classpath>
                    <jvmarg line="${endorsed.classpath.cmd.line.arg}"/>
                    <jvmarg line="${run.jvmargs}"/>
                    <jvmarg line="${run.jvmargs.ide}"/>
                </customize>
            </j2seproject3:test-impl>
        </sequential>
```

```
        </macrodef>
    </target>
    <target if="${junit.available}" name="-init-macrodef-junit-debug" unless="${nb.junit.batch}">
        <macrodef name="junit-debug" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
            <attribute default="**" name="testincludes"/>
            <attribute default="" name="testmethods"/>
            <element name="customize" optional="true"/>
            <sequential>
                <property name="junit.forkmode" value="perTest"/>
                <junit dir="${work.dir}" errorproperty="tests.failed" failureproperty="tests.failed"
fork="true" forkmode="${junit.forkmode}" showoutput="true" tempdir="${build.dir}">
                    <test methods="@{testmethods}" name="@{testincludes}"
todir="${build.test.results.dir}"/>
                    <syspropertyset>
                        <propertyref prefix="test-sys-prop."/>
                        <mapper from="test-sys-prop.*" to="*" type="glob"/>
                    </syspropertyset>
                    <formatter type="brief" usefile="false"/>
                    <formatter type="xml"/>
                    <jvmarg value="-ea"/>
                    <jvmarg line="${debug-args-line}"/>
                    <jvmarg
value="-Xrunjdwp:transport=${debug-transport},address=${jpda.address}"/>
                    <customize/>
                </junit>
            </sequential>
        </macrodef>
    </target>
```

```
    <target depends="-init-test-properties" if="${nb.junit.batch}"
name="-init-macrodef-junit-debug-batch">

        <macrodef name="junit-debug" uri="http://www.netbeans.org/ns/j2se-project/3">

            <attribute default="${includes}" name="includes"/>

            <attribute default="${excludes}" name="excludes"/>

            <attribute default="**" name="testincludes"/>

            <attribute default="" name="testmethods"/>

            <element name="customize" optional="true"/>

            <sequential>

                <property name="junit.forkmode" value="perTest"/>

                <junit dir="${work.dir}" errorproperty="tests.failed" failureproperty="tests.failed"
fork="true" forkmode="${junit.forkmode}" showoutput="true" tempdir="${build.dir}">

                    <batchtest todir="${build.test.results.dir}">

                        <fileset dir="${test.src.dir}" excludes="@{excludes},${excludes}"
includes="@{includes}">

                            <filename name="@{testincludes}"/>

                        </fileset>

                        <fileset dir="${build.test.classes.dir}"
excludes="@{excludes},${excludes},${test.binaryexcludes}" includes="${test.binaryincludes}">

                            <filename name="${test.binarytestincludes}"/>

                        </fileset>

                    </batchtest>

                    <syspropertyset>

                        <propertyref prefix="test-sys-prop."/>

                        <mapper from="test-sys-prop.*" to="*" type="glob"/>

                    </syspropertyset>

                    <formatter type="brief" usefile="false"/>

                    <formatter type="xml"/>

                    <jvmarg value="-ea"/>

                    <jvmarg line="${debug-args-line}"/>
```

```
                <jvmarg
value="-Xrunjdwp:transport=${debug-transport},address=${jpda.address}"/>
                <customize/>
            </junit>
        </sequential>
    </macrodef>
</target>
<target depends="-init-macrodef-junit-debug,-init-macrodef-junit-debug-batch"
if="${junit.available}" name="-init-macrodef-junit-debug-impl">
    <macrodef name="test-debug-impl" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${includes}" name="includes"/>
        <attribute default="${excludes}" name="excludes"/>
        <attribute default="**" name="testincludes"/>
        <attribute default="" name="testmethods"/>
        <element implicit="true" name="customize" optional="true"/>
        <sequential>
            <j2seproject3:junit-debug excludes="@{excludes}" includes="@{includes}"
testincludes="@{testincludes}" testmethods="@{testmethods}">
                <customize/>
            </j2seproject3:junit-debug>
        </sequential>
    </macrodef>
</target>
<target if="${testng.available}" name="-init-macrodef-testng-debug">
    <macrodef name="testng-debug" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${main.class}" name="testClass"/>
        <attribute default="" name="testMethod"/>
        <element name="customize2" optional="true"/>
        <sequential>
            <condition else="-testclass @{testClass}" property="test.class.or.method"
value="-methods @{testClass}.@{testMethod}">
```

```
                <isset property="test.method"/>
            </condition>
            <condition else="-suitename GUI_PLAY -testname @{testClass}
${test.class.or.method}" property="testng.cmd.args" value="@{testClass}">
                <matches pattern=".*\.xml" string="@{testClass}"/>
            </condition>
            <delete dir="${build.test.results.dir}" quiet="true"/>
            <mkdir dir="${build.test.results.dir}"/>
            <j2seproject3:debug classname="org.testng.TestNG"
classpath="${debug.test.classpath}">
                <customize>
                    <customize2/>
                    <jvmarg value="-ea"/>
                    <arg line="${testng.debug.mode}"/>
                    <arg line="-d ${build.test.results.dir}"/>
                    <arg line="-listener org.testng.reporters.VerboseReporter"/>
                    <arg line="${testng.cmd.args}"/>
                </customize>
            </j2seproject3:debug>
        </sequential>
    </macrodef>
</target>
<target depends="-init-macrodef-testng-debug" if="${testng.available}"
name="-init-macrodef-testng-debug-impl">
    <macrodef name="testng-debug-impl" uri="http://www.netbeans.org/ns/j2se-project/3">
        <attribute default="${main.class}" name="testClass"/>
        <attribute default="" name="testMethod"/>
        <element implicit="true" name="customize2" optional="true"/>
        <sequential>
            <j2seproject3:testng-debug testClass="@{testClass}" testMethod="@{testMethod}">
```

```
            <customize2/>
        </j2seproject3:testng-debug>
    </sequential>
  </macrodef>
</target>
<target depends="-init-macrodef-junit-debug-impl" if="${junit.available}"
name="-init-macrodef-test-debug-junit">
  <macrodef name="test-debug" uri="http://www.netbeans.org/ns/j2se-project/3">
    <attribute default="${includes}" name="includes"/>
    <attribute default="${excludes}" name="excludes"/>
    <attribute default="**" name="testincludes"/>
    <attribute default="" name="testmethods"/>
    <attribute default="${main.class}" name="testClass"/>
    <attribute default="" name="testMethod"/>
    <sequential>
      <j2seproject3:test-debug-impl excludes="@{excludes}" includes="@{includes}"
testincludes="@{testincludes}" testmethods="@{testmethods}">
        <customize>
          <classpath>
            <path path="${run.test.classpath}"/>
          </classpath>
          <jvmarg line="${endorsed.classpath.cmd.line.arg}"/>
          <jvmarg line="${run.jvmargs}"/>
          <jvmarg line="${run.jvmargs.ide}"/>
        </customize>
      </j2seproject3:test-debug-impl>
    </sequential>
  </macrodef>
</target>
```

```
    <target depends="-init-macrodef-testng-debug-impl" if="${testng.available}"
name="-init-macrodef-test-debug-testng">
        <macrodef name="test-debug" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${includes}" name="includes"/>
            <attribute default="${excludes}" name="excludes"/>
            <attribute default="**" name="testincludes"/>
            <attribute default="" name="testmethods"/>
            <attribute default="${main.class}" name="testClass"/>
            <attribute default="" name="testMethod"/>
            <sequential>
                <j2seproject3:testng-debug-impl testClass="@{testClass}"
testMethod="@{testMethod}">
                    <customize2>
                        <syspropertyset>
                            <propertyref prefix="test-sys-prop."/>
                            <mapper from="test-sys-prop.*" to="*" type="glob"/>
                        </syspropertyset>
                    </customize2>
                </j2seproject3:testng-debug-impl>
            </sequential>
        </macrodef>
    </target>
    <target depends="-init-macrodef-test-debug-junit,-init-macrodef-test-debug-testng"
name="-init-macrodef-test-debug"/>
    <!--
                pre NB7.2 profiling section; consider it deprecated
            -->
    <target depends="-profile-pre-init, init, -profile-post-init, -profile-init-macrodef-profile,
-profile-init-check" if="profiler.info.jvmargs.agent" name="profile-init"/>
    <target if="profiler.info.jvmargs.agent" name="-profile-pre-init">
        <!-- Empty placeholder for easier customization. -->
```

```
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target if="profiler.info.jvmargs.agent" name="-profile-post-init">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target if="profiler.info.jvmargs.agent" name="-profile-init-macrodef-profile">
        <macrodef name="resolve">
            <attribute name="name"/>
            <attribute name="value"/>
            <sequential>
                <property name="@{name}" value="${env.@{value}}"/>
            </sequential>
        </macrodef>
        <macrodef name="profile">
            <attribute default="${main.class}" name="classname"/>
            <element name="customize" optional="true"/>
            <sequential>
                <property environment="env"/>
                <resolve name="profiler.current.path" value="${profiler.info.pathvar}"/>
                <java classname="@{classname}" dir="${profiler.info.dir}"
failonerror="${java.failonerror}" fork="true" jvm="${profiler.info.jvm}">
                    <jvmarg line="${endorsed.classpath.cmd.line.arg}"/>
                    <jvmarg value="${profiler.info.jvmargs.agent}"/>
                    <jvmarg line="${profiler.info.jvmargs}"/>
                    <env key="${profiler.info.pathvar}"
path="${profiler.info.agentpath}:${profiler.current.path}"/>
                    <arg line="${application.args}"/>
                    <classpath>
                        <path path="${run.classpath}"/>
```

```
                </classpath>
                <syspropertyset>
                    <propertyref prefix="run-sys-prop."/>
                    <mapper from="run-sys-prop.*" to="*" type="glob"/>
                </syspropertyset>
                <customize/>
            </java>
        </sequential>
    </macrodef>
</target>
<target depends="-profile-pre-init, init, -profile-post-init, -profile-init-macrodef-profile"
if="profiler.info.jvmargs.agent" name="-profile-init-check">
    <fail unless="profiler.info.jvm">Must set JVM to use for profiling in profiler.info.jvm</fail>
    <fail unless="profiler.info.jvmargs.agent">Must set profiler agent JVM arguments in
profiler.info.jvmargs.agent</fail>
</target>
<!--
            end of pre NB7.2 profiling section
        -->
<target depends="-init-debug-args" name="-init-macrodef-nbjpda">
    <macrodef name="nbjpdastart" uri="http://www.netbeans.org/ns/j2se-project/1">
        <attribute default="${main.class}" name="name"/>
        <attribute default="${debug.classpath}" name="classpath"/>
        <attribute default="" name="stopclassname"/>
        <sequential>
            <nbjpdastart addressproperty="jpda.address" name="@{name}"
stopclassname="@{stopclassname}" transport="${debug-transport}">
                <classpath>
                    <path path="@{classpath}"/>
                </classpath>
```

```
            </nbjpdastart>
        </sequential>
    </macrodef>
    <macrodef name="nbjpdareload" uri="http://www.netbeans.org/ns/j2se-project/1">
        <attribute default="${build.classes.dir}" name="dir"/>
        <sequential>
            <nbjpdareload>
                <fileset dir="@{dir}" includes="${fix.classes}">
                    <include name="${fix.includes}*.class"/>
                </fileset>
            </nbjpdareload>
        </sequential>
    </macrodef>
</target>
<target name="-init-debug-args">
    <property name="version-output" value="java version &quot;${ant.java.version}"/>
    <condition property="have-jdk-older-than-1.4">
        <or>
            <contains string="${version-output}" substring="java version &quot;1.0"/>
            <contains string="${version-output}" substring="java version &quot;1.1"/>
            <contains string="${version-output}" substring="java version &quot;1.2"/>
            <contains string="${version-output}" substring="java version &quot;1.3"/>
        </or>
    </condition>
    <condition else="-Xdebug" property="debug-args-line" value="-Xdebug -Xnoagent
-Djava.compiler=none">
        <istrue value="${have-jdk-older-than-1.4}"/>
    </condition>
    <condition else="dt_socket" property="debug-transport-by-os" value="dt_shmem">
```

```
            <os family="windows"/>
        </condition>
        <condition else="${debug-transport-by-os}" property="debug-transport"
value="${debug.transport}">
            <isset property="debug.transport"/>
        </condition>
    </target>
    <target depends="-init-debug-args" name="-init-macrodef-debug">
        <macrodef name="debug" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${main.class}" name="classname"/>
            <attribute default="${debug.classpath}" name="classpath"/>
            <element name="customize" optional="true"/>
            <sequential>
                <java classname="@{classname}" dir="${work.dir}" failonerror="${java.failonerror}"
fork="true">
                    <jvmarg line="${endorsed.classpath.cmd.line.arg}"/>
                    <jvmarg line="${debug-args-line}"/>
                    <jvmarg
value="-Xrunjdwp:transport=${debug-transport},address=${jpda.address}"/>
                    <jvmarg value="-Dfile.encoding=${runtime.encoding}"/>
                    <redirector errorencoding="${runtime.encoding}"
inputencoding="${runtime.encoding}" outputencoding="${runtime.encoding}"/>
                    <jvmarg line="${run.jvmargs}"/>
                    <jvmarg line="${run.jvmargs.ide}"/>
                    <classpath>
                        <path path="@{classpath}"/>
                    </classpath>
                    <syspropertyset>
                        <propertyref prefix="run-sys-prop."/>
                        <mapper from="run-sys-prop.*" to="*" type="glob"/>
                    </syspropertyset>
```

```
            <customize/>
        </java>
    </sequential>
  </macrodef>
</target>
<target name="-init-macrodef-java">
  <macrodef name="java" uri="http://www.netbeans.org/ns/j2se-project/1">
    <attribute default="${main.class}" name="classname"/>
    <attribute default="${run.classpath}" name="classpath"/>
    <attribute default="jvm" name="jvm"/>
    <element name="customize" optional="true"/>
    <sequential>
      <java classname="@{classname}" dir="${work.dir}" failonerror="${java.failonerror}"
fork="true">
        <jvmarg line="${endorsed.classpath.cmd.line.arg}"/>
        <jvmarg value="-Dfile.encoding=${runtime.encoding}"/>
        <redirector errorencoding="${runtime.encoding}"
inputencoding="${runtime.encoding}" outputencoding="${runtime.encoding}"/>
        <jvmarg line="${run.jvmargs}"/>
        <jvmarg line="${run.jvmargs.ide}"/>
        <classpath>
          <path path="@{classpath}"/>
        </classpath>
        <syspropertyset>
          <propertyref prefix="run-sys-prop."/>
          <mapper from="run-sys-prop.*" to="*" type="glob"/>
        </syspropertyset>
        <customize/>
      </java>
    </sequential>
```

```
        </macrodef>
    </target>
    <target name="-init-macrodef-copylibs">
        <macrodef name="copylibs" uri="http://www.netbeans.org/ns/j2se-project/3">
            <attribute default="${manifest.file}" name="manifest"/>
            <element name="customize" optional="true"/>
            <sequential>
                <property location="${build.classes.dir}" name="build.classes.dir.resolved"/>
                <pathconvert property="run.classpath.without.build.classes.dir">
                    <path path="${run.classpath}"/>
                    <map from="${build.classes.dir.resolved}" to=""/>
                </pathconvert>
                <pathconvert pathsep=" " property="jar.classpath">
                    <path path="${run.classpath.without.build.classes.dir}"/>
                    <chainedmapper>
                        <flattenmapper/>
                        <filtermapper>
                            <replacestring from=" " to="%20"/>
                        </filtermapper>
                        <globmapper from="*" to="lib/*"/>
                    </chainedmapper>
                </pathconvert>
                <taskdef classname="org.netbeans.modules.java.j2seproject.copylibstask.CopyLibs"
classpath="${libs.CopyLibs.classpath}" name="copylibs"/>
                <copylibs compress="${jar.compress}" excludeFromCopy="${copylibs.excludes}"
index="${jar.index}" indexMetaInf="${jar.index.metainf}" jarfile="${dist.jar}"
manifest="@{manifest}" rebase="${copylibs.rebase}"
runtimeclasspath="${run.classpath.without.build.classes.dir}">
                    <fileset dir="${build.classes.dir}" excludes="${dist.archive.excludes}"/>
                    <manifest>
                        <attribute name="Class-Path" value="${jar.classpath}"/>
```

```
                <customize/>
            </manifest>
        </copylibs>
    </sequential>
  </macrodef>
</target>
<target name="-init-presetdef-jar">
  <presetdef name="jar" uri="http://www.netbeans.org/ns/j2se-project/1">
    <jar compress="${jar.compress}" index="${jar.index}" jarfile="${dist.jar}">
      <j2seproject1:fileset dir="${build.classes.dir}" excludes="${dist.archive.excludes}"/>
    </jar>
  </presetdef>
</target>
<target name="-init-ap-cmdline-properties">
  <property name="annotation.processing.enabled" value="true"/>
  <property name="annotation.processing.processors.list" value=""/>
  <property name="annotation.processing.processor.options" value=""/>
  <property name="annotation.processing.run.all.processors" value="true"/>
  <property name="javac.processorpath" value="${javac.classpath}"/>
  <property name="javac.test.processorpath" value="${javac.test.classpath}"/>
  <condition property="ap.supported.internal" value="true">
    <not>
      <matches pattern="1\.[0-5](\..*)?" string="${javac.source}"/>
    </not>
  </condition>
</target>
<target depends="-init-ap-cmdline-properties" if="ap.supported.internal"
name="-init-ap-cmdline-supported">
  <condition else="" property="ap.processors.internal" value="-processor
${annotation.processing.processors.list}">
```

```
            <isfalse value="${annotation.processing.run.all.processors}"/>
        </condition>
        <condition else="" property="ap.proc.none.internal" value="-proc:none">
            <isfalse value="${annotation.processing.enabled}"/>
        </condition>
    </target>
    <target depends="-init-ap-cmdline-properties,-init-ap-cmdline-supported"
name="-init-ap-cmdline">
        <property name="ap.cmd.line.internal" value=""/>
    </target>
    <target
depends="-pre-init,-init-private,-init-user,-init-project,-do-init,-post-init,-init-check,-init-macrodef-
property,-init-macrodef-javac,-init-macrodef-test,-init-macrodef-test-debug,-init-macrodef-nbjpd
a,-init-macrodef-debug,-init-macrodef-java,-init-presetdef-jar,-init-ap-cmdline" name="init"/>
    <!--
            ===================
            COMPILATION SECTION
            ===================
        -->
    <target name="-deps-jar-init" unless="built-jar.properties">
        <property location="${build.dir}/built-jar.properties" name="built-jar.properties"/>
        <delete file="${built-jar.properties}" quiet="true"/>
    </target>
    <target if="already.built.jar.${basedir}" name="-warn-already-built-jar">
        <echo level="warn" message="Cycle detected: GUI_PLAY was already built"/>
    </target>
    <target depends="init,-deps-jar-init" name="deps-jar" unless="no.deps">
        <mkdir dir="${build.dir}"/>
        <touch file="${built-jar.properties}" verbose="false"/>
        <property file="${built-jar.properties}" prefix="already.built.jar."/>
        <antcall target="-warn-already-built-jar"/>
```

```
        <propertyfile file="${built-jar.properties}">
            <entry key="${basedir}" value=""/>
        </propertyfile>
    </target>
    <target depends="init,-check-automatic-build,-clean-after-automatic-build"
name="-verify-automatic-build"/>
    <target depends="init" name="-check-automatic-build">
        <available file="${build.classes.dir}/.netbeans_automatic_build"
property="netbeans.automatic.build"/>
    </target>
    <target depends="init" if="netbeans.automatic.build" name="-clean-after-automatic-build">
        <antcall target="clean"/>
    </target>
    <target depends="init,deps-jar" name="-pre-pre-compile">
        <mkdir dir="${build.classes.dir}"/>
    </target>
    <target name="-pre-compile">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target if="do.depend.true" name="-compile-depend">
        <pathconvert property="build.generated.subdirs">
            <dirset dir="${build.generated.sources.dir}" erroronmissingdir="false">
                <include name="*"/>
            </dirset>
        </pathconvert>
        <j2seproject3:depend srcdir="${src.dir}:${build.generated.subdirs}"/>
    </target>
    <target depends="init,deps-jar,-pre-pre-compile,-pre-compile,
-copy-persistence-xml,-compile-depend" if="have.sources" name="-do-compile">
```

```
        <j2seproject3:javac gensrcdir="${build.generated.sources.dir}"/>

        <copy todir="${build.classes.dir}">

            <fileset dir="${src.dir}" excludes="${build.classes.excludes},${excludes}"
includes="${includes}"/>

        </copy>

    </target>

    <target if="has.persistence.xml" name="-copy-persistence-xml">

        <mkdir dir="${build.classes.dir}/META-INF"/>

        <copy todir="${build.classes.dir}/META-INF">

            <fileset dir="${meta.inf.dir}" includes="persistence.xml orm.xml"/>

        </copy>

    </target>

    <target name="-post-compile">

        <!-- Empty placeholder for easier customization. -->

        <!-- You can override this target in the ../build.xml file. -->

    </target>

    <target
depends="init,deps-jar,-verify-automatic-build,-pre-pre-compile,-pre-compile,-do-compile,-post-
compile" description="Compile project." name="compile"/>

    <target name="-pre-compile-single">

        <!-- Empty placeholder for easier customization. -->

        <!-- You can override this target in the ../build.xml file. -->

    </target>

    <target depends="init,deps-jar,-pre-pre-compile" name="-do-compile-single">

        <fail unless="javac.includes">Must select some files in the IDE or set javac.includes</fail>

        <j2seproject3:force-recompile/>

        <j2seproject3:javac excludes="" gensrcdir="${build.generated.sources.dir}"
includes="${javac.includes}" sourcepath="${src.dir}"/>

    </target>

    <target name="-post-compile-single">

        <!-- Empty placeholder for easier customization. -->
```

```
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target
depends="init,deps-jar,-verify-automatic-build,-pre-pre-compile,-pre-compile-single,-do-compile
-single,-post-compile-single" name="compile-single"/>
    <!--
                ====================
                JAR BUILDING SECTION
                ====================
            -->
    <target depends="init" name="-pre-pre-jar">
        <dirname file="${dist.jar}" property="dist.jar.dir"/>
        <mkdir dir="${dist.jar.dir}"/>
    </target>
    <target name="-pre-jar">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target depends="init" if="do.archive" name="-do-jar-create-manifest"
unless="manifest.available">
        <tempfile deleteonexit="true" destdir="${build.dir}" property="tmp.manifest.file"/>
        <touch file="${tmp.manifest.file}" verbose="false"/>
    </target>
    <target depends="init" if="do.archive+manifest.available" name="-do-jar-copy-manifest">
        <tempfile deleteonexit="true" destdir="${build.dir}" property="tmp.manifest.file"/>
        <copy file="${manifest.file}" tofile="${tmp.manifest.file}"/>
    </target>
    <target depends="init,-do-jar-create-manifest,-do-jar-copy-manifest"
if="do.archive+main.class.available" name="-do-jar-set-mainclass">
        <manifest file="${tmp.manifest.file}" mode="update">
            <attribute name="Main-Class" value="${main.class}"/>
```

```
        </manifest>

    </target>

    <target depends="init,-do-jar-create-manifest,-do-jar-copy-manifest"
if="do.archive+profile.available" name="-do-jar-set-profile">

        <manifest file="${tmp.manifest.file}" mode="update">

            <attribute name="Profile" value="${javac.profile}"/>

        </manifest>

    </target>

    <target depends="init,-do-jar-create-manifest,-do-jar-copy-manifest"
if="do.archive+splashscreen.available" name="-do-jar-set-splashscreen">

        <basename file="${application.splash}" property="splashscreen.basename"/>

        <mkdir dir="${build.classes.dir}/META-INF"/>

        <copy failonerror="false" file="${application.splash}"
todir="${build.classes.dir}/META-INF"/>

        <manifest file="${tmp.manifest.file}" mode="update">

            <attribute name="SplashScreen-Image" value="META-INF/${splashscreen.basename}"/>

        </manifest>

    </target>

    <target
depends="init,-init-macrodef-copylibs,compile,-pre-pre-jar,-pre-jar,-do-jar-create-manifest,-do-ja
r-copy-manifest,-do-jar-set-mainclass,-do-jar-set-profile,-do-jar-set-splashscreen" if="do.mkdist"
name="-do-jar-copylibs">

        <j2seproject3:copylibs manifest="${tmp.manifest.file}"/>

        <echo level="info">To run this application from the command line without Ant, try:</echo>

        <property location="${dist.jar}" name="dist.jar.resolved"/>

        <echo level="info">java -jar "${dist.jar.resolved}"</echo>

    </target>

    <target
depends="init,compile,-pre-pre-jar,-pre-jar,-do-jar-create-manifest,-do-jar-copy-manifest,-do-jar-
set-mainclass,-do-jar-set-profile,-do-jar-set-splashscreen" if="do.archive" name="-do-jar-jar"
unless="do.mkdist">

        <j2seproject1:jar manifest="${tmp.manifest.file}"/>

        <property location="${build.classes.dir}" name="build.classes.dir.resolved"/>
```

```
        <property location="${dist.jar}" name="dist.jar.resolved"/>
        <pathconvert property="run.classpath.with.dist.jar">
            <path path="${run.classpath}"/>
            <map from="${build.classes.dir.resolved}" to="${dist.jar.resolved}"/>
        </pathconvert>
        <condition else="" property="jar.usage.message" value="To run this application from the
command line without Ant, try:${line.separator}${platform.java} -cp
${run.classpath.with.dist.jar} ${main.class}">
            <isset property="main.class.available"/>
        </condition>
        <condition else="debug" property="jar.usage.level" value="info">
            <isset property="main.class.available"/>
        </condition>
        <echo level="${jar.usage.level}" message="${jar.usage.message}"/>
    </target>
    <target depends="-do-jar-copylibs" if="do.archive" name="-do-jar-delete-manifest">
        <delete>
            <fileset file="${tmp.manifest.file}"/>
        </delete>
    </target>
    <target
depends="init,compile,-pre-pre-jar,-pre-jar,-do-jar-create-manifest,-do-jar-copy-manifest,-do-jar-
set-mainclass,-do-jar-set-profile,-do-jar-set-splashscreen,-do-jar-jar,-do-jar-delete-manifest"
name="-do-jar-without-libraries"/>
    <target
depends="init,compile,-pre-pre-jar,-pre-jar,-do-jar-create-manifest,-do-jar-copy-manifest,-do-jar-
set-mainclass,-do-jar-set-profile,-do-jar-set-splashscreen,-do-jar-copylibs,-do-jar-delete-manifest
" name="-do-jar-with-libraries"/>
    <target name="-post-jar">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
```

```
    <target depends="init,compile,-pre-jar,-do-jar-without-libraries,-do-jar-with-libraries,-post-jar" name="-do-jar"/>
    <target depends="init,compile,-pre-jar,-do-jar,-post-jar" description="Build JAR." name="jar"/>
    <!--
            ================
            EXECUTION SECTION
            ================
        -->
    <target depends="init,compile" description="Run a main class." name="run">
        <j2seproject1:java>
            <customize>
                <arg line="${application.args}"/>
            </customize>
        </j2seproject1:java>
    </target>
    <target name="-do-not-recompile">
        <property name="javac.includes.binary" value=""/>
    </target>
    <target depends="init,compile-single" name="run-single">
        <fail unless="run.class">Must select one file in the IDE or set run.class</fail>
        <j2seproject1:java classname="${run.class}"/>
    </target>
    <target depends="init,compile-test-single" name="run-test-with-main">
        <fail unless="run.class">Must select one file in the IDE or set run.class</fail>
        <j2seproject1:java classname="${run.class}" classpath="${run.test.classpath}"/>
    </target>
    <!--
            ================
            DEBUGGING SECTION
```

```
            ================

        -->

    <target depends="init" if="netbeans.home" name="-debug-start-debugger">

        <j2seproject1:nbjpdastart name="${debug.class}"/>

    </target>

    <target depends="init" if="netbeans.home" name="-debug-start-debugger-main-test">

        <j2seproject1:nbjpdastart classpath="${debug.test.classpath}" name="${debug.class}"/>

    </target>

    <target depends="init,compile" name="-debug-start-debuggee">

        <j2seproject3:debug>

            <customize>

                <arg line="${application.args}"/>

            </customize>

        </j2seproject3:debug>

    </target>

    <target depends="init,compile,-debug-start-debugger,-debug-start-debuggee"
description="Debug project in IDE." if="netbeans.home" name="debug"/>

    <target depends="init" if="netbeans.home" name="-debug-start-debugger-stepinto">

        <j2seproject1:nbjpdastart stopclassname="${main.class}"/>

    </target>

    <target depends="init,compile,-debug-start-debugger-stepinto,-debug-start-debuggee"
if="netbeans.home" name="debug-stepinto"/>

    <target depends="init,compile-single" if="netbeans.home"
name="-debug-start-debuggee-single">

        <fail unless="debug.class">Must select one file in the IDE or set debug.class</fail>

        <j2seproject3:debug classname="${debug.class}"/>

    </target>

    <target depends="init,compile-single,-debug-start-debugger,-debug-start-debuggee-single"
if="netbeans.home" name="debug-single"/>

    <target depends="init,compile-test-single" if="netbeans.home"
name="-debug-start-debuggee-main-test">
```

```
        <fail unless="debug.class">Must select one file in the IDE or set debug.class</fail>

        <j2seproject3:debug classname="${debug.class}" classpath="${debug.test.classpath}"/>

    </target>

    <target
depends="init,compile-test-single,-debug-start-debugger-main-test,-debug-start-debuggee-mai
n-test" if="netbeans.home" name="debug-test-with-main"/>

    <target depends="init" name="-pre-debug-fix">

        <fail unless="fix.includes">Must set fix.includes</fail>

        <property name="javac.includes" value="${fix.includes}.java"/>

    </target>

    <target depends="init,-pre-debug-fix,compile-single" if="netbeans.home"
name="-do-debug-fix">

        <j2seproject1:nbjpdareload/>

    </target>

    <target depends="init,-pre-debug-fix,-do-debug-fix" if="netbeans.home" name="debug-fix"/>

    <!--

            =================

            PROFILING SECTION

            =================

        -->

    <!--

            pre NB7.2 profiler integration

        -->

    <target depends="profile-init,compile" description="Profile a project in the IDE."
if="profiler.info.jvmargs.agent" name="-profile-pre72">

        <fail unless="netbeans.home">This target only works when run from inside the NetBeans
IDE.</fail>

        <nbprofiledirect>

            <classpath>

                <path path="${run.classpath}"/>

            </classpath>
```

```
        </nbprofiledirect>
        <profile/>
    </target>
    <target depends="profile-init,compile-single" description="Profile a selected class in the IDE."
if="profiler.info.jvmargs.agent" name="-profile-single-pre72">
        <fail unless="profile.class">Must select one file in the IDE or set profile.class</fail>
        <fail unless="netbeans.home">This target only works when run from inside the NetBeans
IDE.</fail>
        <nbprofiledirect>
            <classpath>
                <path path="${run.classpath}"/>
            </classpath>
        </nbprofiledirect>
        <profile classname="${profile.class}"/>
    </target>
    <target depends="profile-init,compile-single" if="profiler.info.jvmargs.agent"
name="-profile-applet-pre72">
        <fail unless="netbeans.home">This target only works when run from inside the NetBeans
IDE.</fail>
        <nbprofiledirect>
            <classpath>
                <path path="${run.classpath}"/>
            </classpath>
        </nbprofiledirect>
        <profile classname="sun.applet.AppletViewer">
            <customize>
                <arg value="${applet.url}"/>
            </customize>
        </profile>
    </target>
```

```
    <target depends="profile-init,compile-test-single" if="profiler.info.jvmargs.agent"
name="-profile-test-single-pre72">

        <fail unless="netbeans.home">This target only works when run from inside the NetBeans
IDE.</fail>

        <nbprofiledirect>

            <classpath>

                <path path="${run.test.classpath}"/>

            </classpath>

        </nbprofiledirect>

        <junit dir="${profiler.info.dir}" errorproperty="tests.failed" failureproperty="tests.failed"
fork="true" jvm="${profiler.info.jvm}" showoutput="true">

            <env key="${profiler.info.pathvar}"
path="${profiler.info.agentpath}:${profiler.current.path}"/>

            <jvmarg value="${profiler.info.jvmargs.agent}"/>

            <jvmarg line="${profiler.info.jvmargs}"/>

            <test name="${profile.class}"/>

            <classpath>

                <path path="${run.test.classpath}"/>

            </classpath>

            <syspropertyset>

                <propertyref prefix="test-sys-prop."/>

                <mapper from="test-sys-prop.*" to="*" type="glob"/>

            </syspropertyset>

            <formatter type="brief" usefile="false"/>

            <formatter type="xml"/>

        </junit>

    </target>

    <!--

            end of pre NB72 profiling section

        -->

    <target if="netbeans.home" name="-profile-check">
```

```
        <condition property="profiler.configured">
            <or>
                <contains casesensitive="true" string="${run.jvmargs.ide}" substring="-agentpath:"/>
                <contains casesensitive="true" string="${run.jvmargs.ide}" substring="-javaagent:"/>
            </or>
        </condition>
    </target>

    <target depends="-profile-check,-profile-pre72" description="Profile a project in the IDE."
if="profiler.configured" name="profile" unless="profiler.info.jvmargs.agent">
        <startprofiler/>
        <antcall target="run"/>
    </target>

    <target depends="-profile-check,-profile-single-pre72" description="Profile a selected class in
the IDE." if="profiler.configured" name="profile-single" unless="profiler.info.jvmargs.agent">
        <fail unless="run.class">Must select one file in the IDE or set run.class</fail>
        <startprofiler/>
        <antcall target="run-single"/>
    </target>

    <target depends="-profile-test-single-pre72" description="Profile a selected test in the IDE."
name="profile-test-single"/>

    <target depends="-profile-check" description="Profile a selected test in the IDE."
if="profiler.configured" name="profile-test" unless="profiler.info.jvmargs">
        <fail unless="test.includes">Must select some files in the IDE or set test.includes</fail>
        <startprofiler/>
        <antcall target="test-single"/>
    </target>

    <target depends="-profile-check" description="Profile a selected class in the IDE."
if="profiler.configured" name="profile-test-with-main">
        <fail unless="run.class">Must select one file in the IDE or set run.class</fail>
        <startprofiler/>
        <antcal target="run-test-with-main"/>
```

```
    </target>

    <target depends="-profile-check,-profile-applet-pre72" if="profiler.configured"
name="profile-applet" unless="profiler.info.jvmargs.agent">

        <fail unless="applet.url">Must select one file in the IDE or set applet.url</fail>

        <startprofiler/>

        <antcall target="run-applet"/>

    </target>

    <!--

                ===============

                JAVADOC SECTION

                ===============

            -->

    <target depends="init" if="have.sources" name="-javadoc-build">

        <mkdir dir="${dist.javadoc.dir}"/>

        <condition else="" property="javadoc.endorsed.classpath.cmd.line.arg"
value="-J${endorsed.classpath.cmd.line.arg}">

            <and>

                <isset property="endorsed.classpath.cmd.line.arg"/>

                <not>

                    <equals arg1="${endorsed.classpath.cmd.line.arg}" arg2=""/>

                </not>

            </and>

        </condition>

        <condition else="" property="bug5101868workaround" value="*.java">

            <matches pattern="1\.[56](\..*)?" string="${java.version}"/>

        </condition>

        <javadoc additionalparam="-J-Dfile.encoding=${file.encoding} ${javadoc.additionalparam}"
author="${javadoc.author}" charset="UTF-8" destdir="${dist.javadoc.dir}" docencoding="UTF-8"
encoding="${javadoc.encoding.used}" failonerror="true" noindex="${javadoc.noindex}"
nonavbar="${javadoc.nonavbar}" notree="${javadoc.notree}" private="${javadoc.private}"
source="${javac.source}" splitindex="${javadoc.splitindex}" use="${javadoc.use}"
useexternalfile="true" version="${javadoc.version}" windowtitle="${javadoc.windowtitle}">
```

```
            <classpath>
                <path path="${javac.classpath}"/>
            </classpath>
            <fileset dir="${src.dir}" excludes="${bug5101868workaround},${excludes}"
includes="${includes}">
                <filename name="**/*.java"/>
            </fileset>
            <fileset dir="${build.generated.sources.dir}" erroronmissingdir="false">
                <include name="**/*.java"/>
                <exclude name="*.java"/>
            </fileset>
            <arg line="${javadoc.endorsed.classpath.cmd.line.arg}"/>
        </javadoc>
        <copy todir="${dist.javadoc.dir}">
            <fileset dir="${src.dir}" excludes="${excludes}" includes="${includes}">
                <filename name="**/doc-files/**"/>
            </fileset>
            <fileset dir="${build.generated.sources.dir}" erroronmissingdir="false">
                <include name="**/doc-files/**"/>
            </fileset>
        </copy>
    </target>
    <target depends="init,-javadoc-build" if="netbeans.home" name="-javadoc-browse"
unless="no.javadoc.preview">
        <nbbrowse file="${dist.javadoc.dir}/index.html"/>
    </target>
    <target depends="init,-javadoc-build,-javadoc-browse" description="Build Javadoc."
name="javadoc"/>
    <!--
                ========================
```

```
            TEST COMPILATION SECTION
            =========================
        -->
    <target depends="init,compile" if="have.tests" name="-pre-pre-compile-test">
        <mkdir dir="${build.test.classes.dir}"/>
    </target>
    <target name="-pre-compile-test">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target if="do.depend.true" name="-compile-test-depend">
        <j2seproject3:depend classpath="${javac.test.classpath}" destdir="${build.test.classes.dir}"
srcdir="${test.src.dir}"/>
    </target>
    <target
depends="init,deps-jar,compile,-pre-pre-compile-test,-pre-compile-test,-compile-test-depend"
if="have.tests" name="-do-compile-test">
        <j2seproject3:javac apgeneratedsrcdir="${build.test.classes.dir}"
classpath="${javac.test.classpath}" debug="true" destdir="${build.test.classes.dir}"
processorpath="${javac.test.processorpath}" srcdir="${test.src.dir}"/>
        <copy todir="${build.test.classes.dir}">
            <fileset dir="${test.src.dir}" excludes="${build.classes.excludes},${excludes}"
includes="${includes}"/>
        </copy>
    </target>
    <target name="-post-compile-test">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target
depends="init,compile,-pre-pre-compile-test,-pre-compile-test,-do-compile-test,-post-compile-te
st" name="compile-test"/>
```

```
    <target name="-pre-compile-test-single">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target depends="init,deps-jar,compile,-pre-pre-compile-test,-pre-compile-test-single"
if="have.tests" name="-do-compile-test-single">
        <fail unless="javac.includes">Must select some files in the IDE or set javac.includes</fail>
        <j2seproject3:force-recompile destdir="${build.test.classes.dir}"/>
        <j2seproject3:javac apgeneratedsrcdir="${build.test.classes.dir}"
classpath="${javac.test.classpath}" debug="true" destdir="${build.test.classes.dir}" excludes=""
includes="${javac.includes}" processorpath="${javac.test.processorpath}"
sourcepath="${test.src.dir}" srcdir="${test.src.dir}"/>
        <copy todir="${build.test.classes.dir}">
            <fileset dir="${test.src.dir}" excludes="${build.classes.excludes},${excludes}"
includes="${includes}"/>
        </copy>
    </target>
    <target name="-post-compile-test-single">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target
depends="init,compile,-pre-pre-compile-test,-pre-compile-test-single,-do-compile-test-single,-po
st-compile-test-single" name="compile-test-single"/>
    <!--
            =======================
            TEST EXECUTION SECTION
            =======================
          -->
    <target depends="init" if="have.tests" name="-pre-test-run">
        <mkdir dir="${build.test.results.dir}"/>
    </target>
```

```
    <target depends="init,compile-test,-pre-test-run" if="have.tests" name="-do-test-run">
        <j2seproject3:test includes="${includes}" testincludes="**/*Test.java"/>
    </target>
    <target depends="init,compile-test,-pre-test-run,-do-test-run" if="have.tests"
name="-post-test-run">
        <fail if="tests.failed" unless="ignore.failing.tests">Some tests failed; see details
above.</fail>
    </target>
    <target depends="init" if="have.tests" name="test-report"/>
    <target depends="init" if="netbeans.home+have.tests" name="-test-browse"/>
    <target
depends="init,compile-test,-pre-test-run,-do-test-run,test-report,-post-test-run,-test-browse"
description="Run unit tests." name="test"/>
    <target depends="init" if="have.tests" name="-pre-test-run-single">
        <mkdir dir="${build.test.results.dir}"/>
    </target>
    <target depends="init,compile-test-single,-pre-test-run-single" if="have.tests"
name="-do-test-run-single">
        <fail unless="test.includes">Must select some files in the IDE or set test.includes</fail>
        <j2seproject3:test excludes="" includes="${test.includes}" testincludes="${test.includes}"/>
    </target>
    <target depends="init,compile-test-single,-pre-test-run-single,-do-test-run-single"
if="have.tests" name="-post-test-run-single">
        <fail if="tests.failed" unless="ignore.failing.tests">Some tests failed; see details
above.</fail>
    </target>
    <target
depends="init,compile-test-single,-pre-test-run-single,-do-test-run-single,-post-test-run-single"
description="Run single unit test." name="test-single"/>
    <target depends="init,compile-test-single,-pre-test-run-single" if="have.tests"
name="-do-test-run-single-method">
        <fail unless="test.class">Must select some files in the IDE or set test.class</fail>
        <fail unless="test.method">Must select some method in the IDE or set test.method</fail>
```

```
        <j2seproject3:test excludes="" includes="${javac.includes}" testincludes="${test.class}"
testmethods="${test.method}"/>

    </target>

    <target depends="init,compile-test-single,-pre-test-run-single,-do-test-run-single-method"
if="have.tests" name="-post-test-run-single-method">

        <fail if="tests.failed" unless="ignore.failing.tests">Some tests failed; see details
above.</fail>

    </target>

    <target
depends="init,compile-test-single,-pre-test-run-single,-do-test-run-single-method,-post-test-run-
single-method" description="Run single unit test." name="test-single-method"/>

    <!--

            =======================

            TEST DEBUGGING SECTION

            =======================

          -->

    <target depends="init,compile-test-single,-pre-test-run-single" if="have.tests"
name="-debug-start-debuggee-test">

        <fail unless="test.class">Must select one file in the IDE or set test.class</fail>

        <j2seproject3:test-debug excludes="" includes="${javac.includes}" testClass="${test.class}"
testincludes="${javac.includes}"/>

    </target>

    <target depends="init,compile-test-single,-pre-test-run-single" if="have.tests"
name="-debug-start-debuggee-test-method">

        <fail unless="test.class">Must select one file in the IDE or set test.class</fail>

        <fail unless="test.method">Must select some method in the IDE or set test.method</fail>

        <j2seproject3:test-debug excludes="" includes="${javac.includes}" testClass="${test.class}"
testMethod="${test.method}" testincludes="${test.class}" testmethods="${test.method}"/>

    </target>

    <target depends="init,compile-test" if="netbeans.home+have.tests"
name="-debug-start-debugger-test">

        <j2seproject1:nbjpdastart classpath="${debug.test.classpath}" name="${test.class}"/>

    </target>
```

```
    <target
depends="init,compile-test-single,-debug-start-debugger-test,-debug-start-debuggee-test"
name="debug-test"/>

    <target
depends="init,compile-test-single,-debug-start-debugger-test,-debug-start-debuggee-test-meth
od" name="debug-test-method"/>

    <target depends="init,-pre-debug-fix,compile-test-single" if="netbeans.home"
name="-do-debug-fix-test">

        <j2seproject1:nbjpdareload dir="${build.test.classes.dir}"/>

    </target>

    <target depends="init,-pre-debug-fix,-do-debug-fix-test" if="netbeans.home"
name="debug-fix-test"/>

    <!--

            ========================

            APPLET EXECUTION SECTION

            ========================

        -->

    <target depends="init,compile-single" name="run-applet">

        <fail unless="applet.url">Must select one file in the IDE or set applet.url</fail>

        <j2seproject1:java classname="sun.applet.AppletViewer">

            <customize>

                <arg value="${applet.url}"/>

            </customize>

        </j2seproject1:java>

    </target>

    <!--

            ========================

            APPLET DEBUGGING  SECTION

            ========================

        -->

    <target depends="init,compile-single" if="netbeans.home"
name="-debug-start-debuggee-applet">
```

```
        <fail unless="applet.url">Must select one file in the IDE or set applet.url</fail>
        <j2seproject3:debug classname="sun.applet.AppletViewer">
            <customize>
                <arg value="${applet.url}"/>
            </customize>
        </j2seproject3:debug>
    </target>
    <target depends="init,compile-single,-debug-start-debugger,-debug-start-debuggee-applet"
if="netbeans.home" name="debug-applet"/>
    <!--
                ===============
                CLEANUP SECTION
                ===============
            -->
    <target name="-deps-clean-init" unless="built-clean.properties">
        <property location="${build.dir}/built-clean.properties" name="built-clean.properties"/>
        <delete file="${built-clean.properties}" quiet="true"/>
    </target>
    <target if="already.built.clean.${basedir}" name="-warn-already-built-clean">
        <echo level="warn" message="Cycle detected: GUI_PLAY was already built"/>
    </target>
    <target depends="init,-deps-clean-init" name="deps-clean" unless="no.deps">
        <mkdir dir="${build.dir}"/>
        <touch file="${built-clean.properties}" verbose="false"/>
        <property file="${built-clean.properties}" prefix="already.built.clean."/>
        <antcall target="-warn-already-built-clean"/>
        <propertyfile file="${built-clean.properties}">
            <entry key="${basedir}" value=""/>
        </propertyfile>
```

```
    </target>
    <target depends="init" name="-do-clean">
        <delete dir="${build.dir}"/>
        <delete dir="${dist.dir}" followsymlinks="false" includeemptydirs="true"/>
    </target>
    <target name="-post-clean">
        <!-- Empty placeholder for easier customization. -->
        <!-- You can override this target in the ../build.xml file. -->
    </target>
    <target depends="init,deps-clean,-do-clean,-post-clean" description="Clean build products."
name="clean"/>
    <target name="-check-call-dep">
        <property file="${call.built.properties}" prefix="already.built."/>
        <condition property="should.call.dep">
            <and>
                <not>
                    <isset property="already.built.${call.subproject}"/>
                </not>
                <available file="${call.script}"/>
            </and>
        </condition>
    </target>
    <target depends="-check-call-dep" if="should.call.dep" name="-maybe-call-dep">
        <ant antfile="${call.script}" inheritall="false" target="${call.target}">
            <propertyset>
                <propertyref prefix="transfer."/>
                <mapper from="transfer.*" to="*" type="glob"/>
            </propertyset>
        </ant>
```

```
    </target>
</project>
```