



---

# Macintosh<sup>®</sup> Communications Toolbox

---

# Road Map

---

**Positioning**

**Overview**

**The User Experience**

**Concepts/Details**

**Runtime Environment**

**Summary**

# Positioning

---

**Positioning**

**Overview**

**The User Experience**

**Concepts/Details**

**Runtime Environment**

**Summary**

# Macintosh® Communications Toolbox

---

Communications services for Macintosh...



Data Connections



Terminal Emulations



File Transfer Services



...integrated into basic System Software.

# Objectives

---

Increase value of the Macintosh<sup>®</sup> computer by adding pervasive communications in key environments

Leverage third party developers

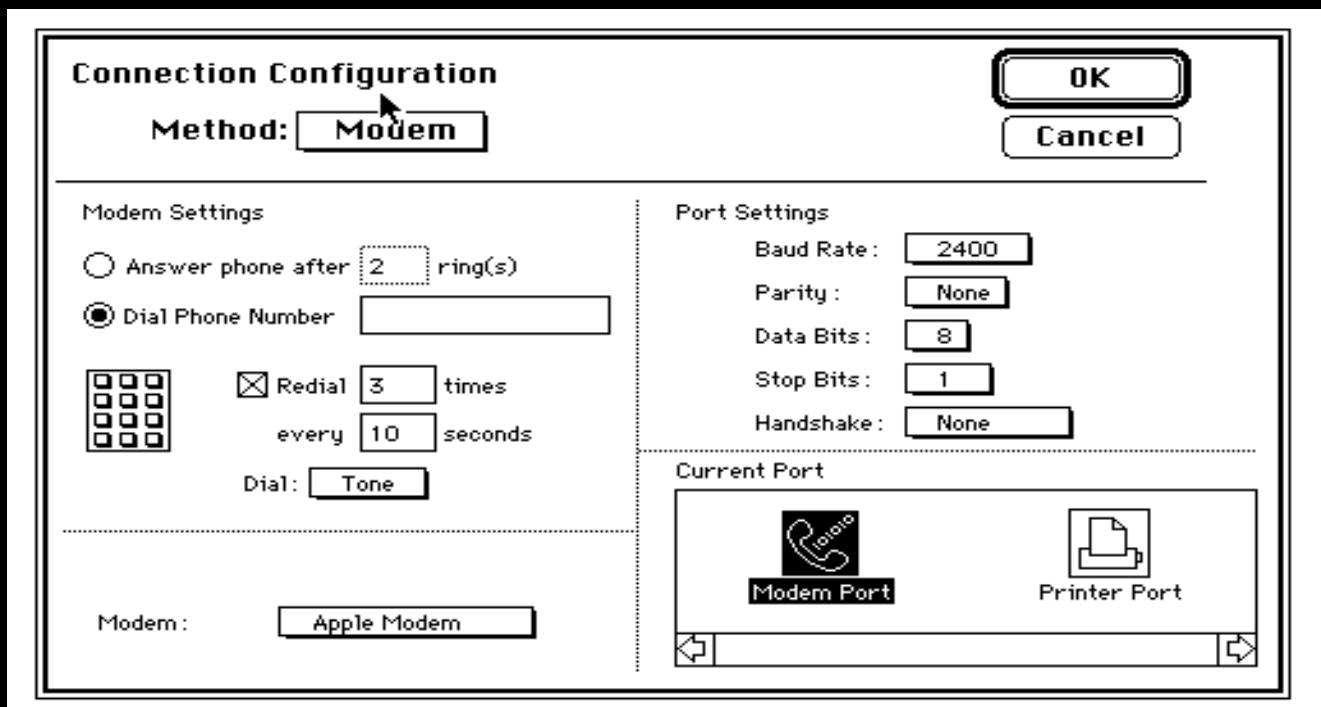
# Significance – 1

---

- Easier to create great communications applications
  - Standard API's
  - Open Architecture

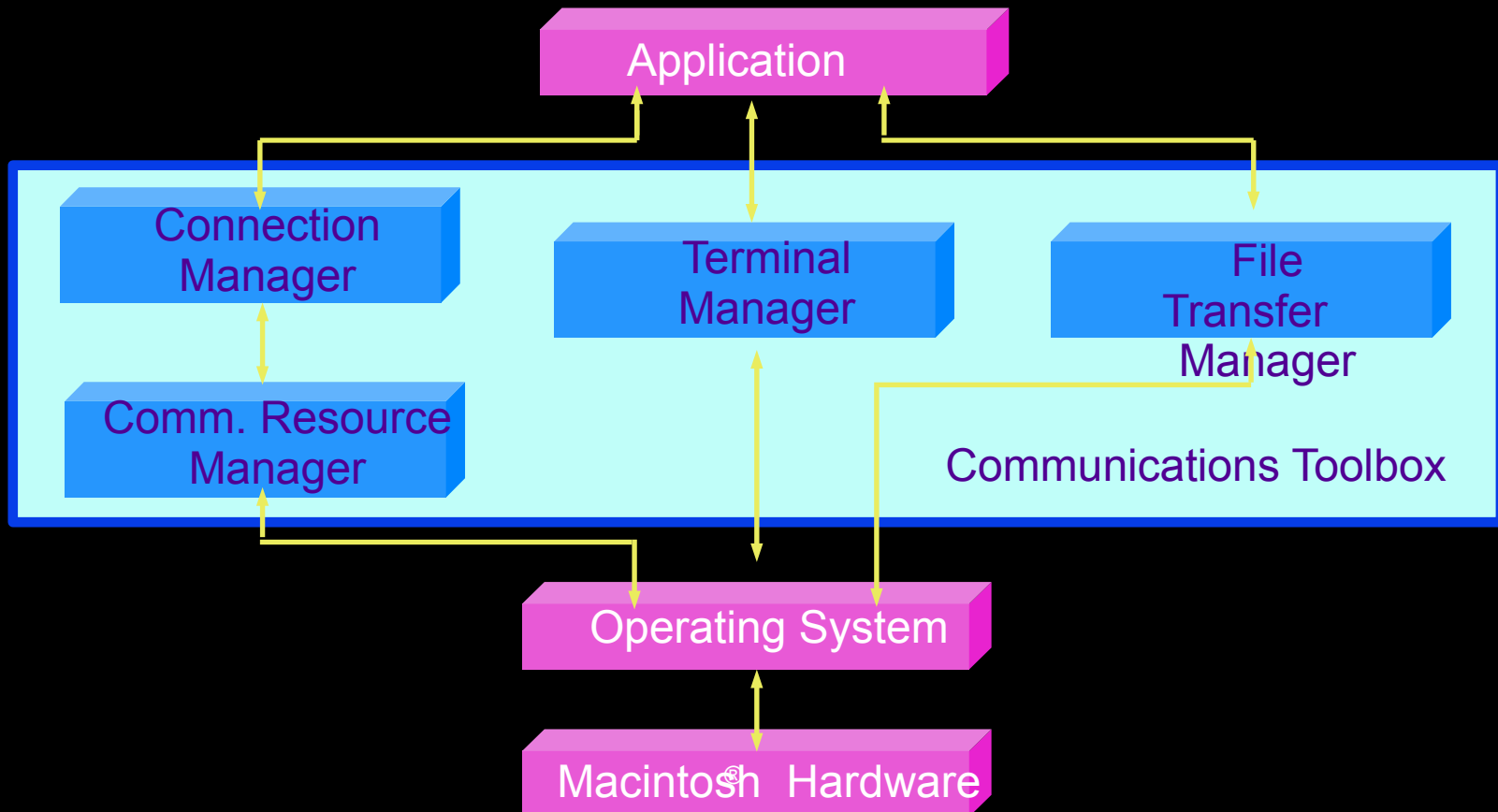
# Significance – 2

Extends the power and consistency of the Macintosh<sup>®</sup> computer to communications



# High Level Flowchart

---





# Main Components – 1

---

## Connection Manager and Tools

- Serial Connection Tool
- Modem Connection Tool
- LAT Connection Tool
- MacPAD (X.25) Connection Tool

# Main Components – 2

---

## Terminal Manager and Tools

- TTY Terminal Tool
- DEC VT102 Terminal Tool
- DEC VT320 Terminal Tool

# **Main Components – 3**

---

## **File Transfer Manager and Tools**

- ASCII Text File Transfer Tool
- XMODEM File Transfer Tool

# Main Components – 4

---

## Communications Resource Manager

- Identifies installed communication cards
- Support for multiple session environments

# Unique Advantages

---

- Consistent user interfaces across applications
- “Small” memory/disk footprint
- Open architecture

# Development Status

---

- Communications Toolbox available through APDA™
- Managers are final
- Tools are beta

# Apple Products Using the Toolbox

---

- MacX<sup>™</sup>
- MacX25<sup>™</sup>
- MacWorkStation<sup>™</sup>

# Overview

---

<b>Positioning</b>
<b>Overview</b>
<b>The User Experience</b>
<b>Concepts/Details</b>
<b>Runtime Environment</b>
<b>Summary</b>



# Design Goals

---

Standard Access to Communications Services

- Development of Better Communications Software
- Leading to Improvement in overall User Experience

# Communications Services

---



Data Connections



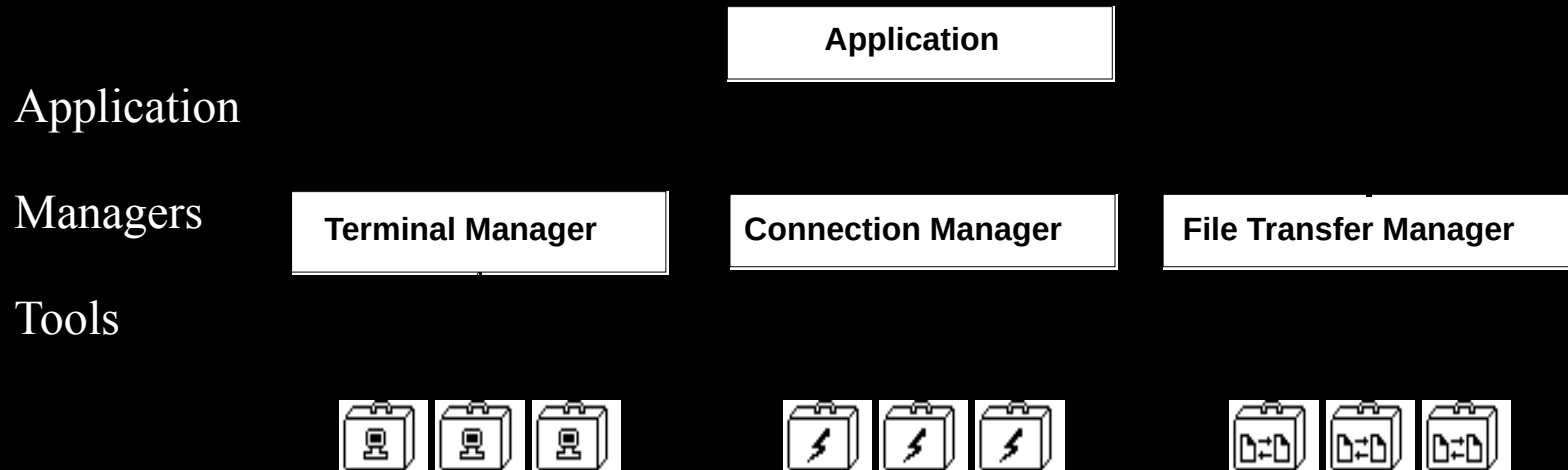
Terminal Emulations

File Transfer Services



# High Level Architecture

---



# System Requirements

---

- Bundled with System 7.0
- Compatible with System 6.0.4 and later
- Applications/Tools may have different system requirements

# Development Platforms

---

- MPW<sup>™</sup>3.x Development Platform
- Third party platforms that can convert MPW interface files and object files can easily be used for development work

# The User Experience

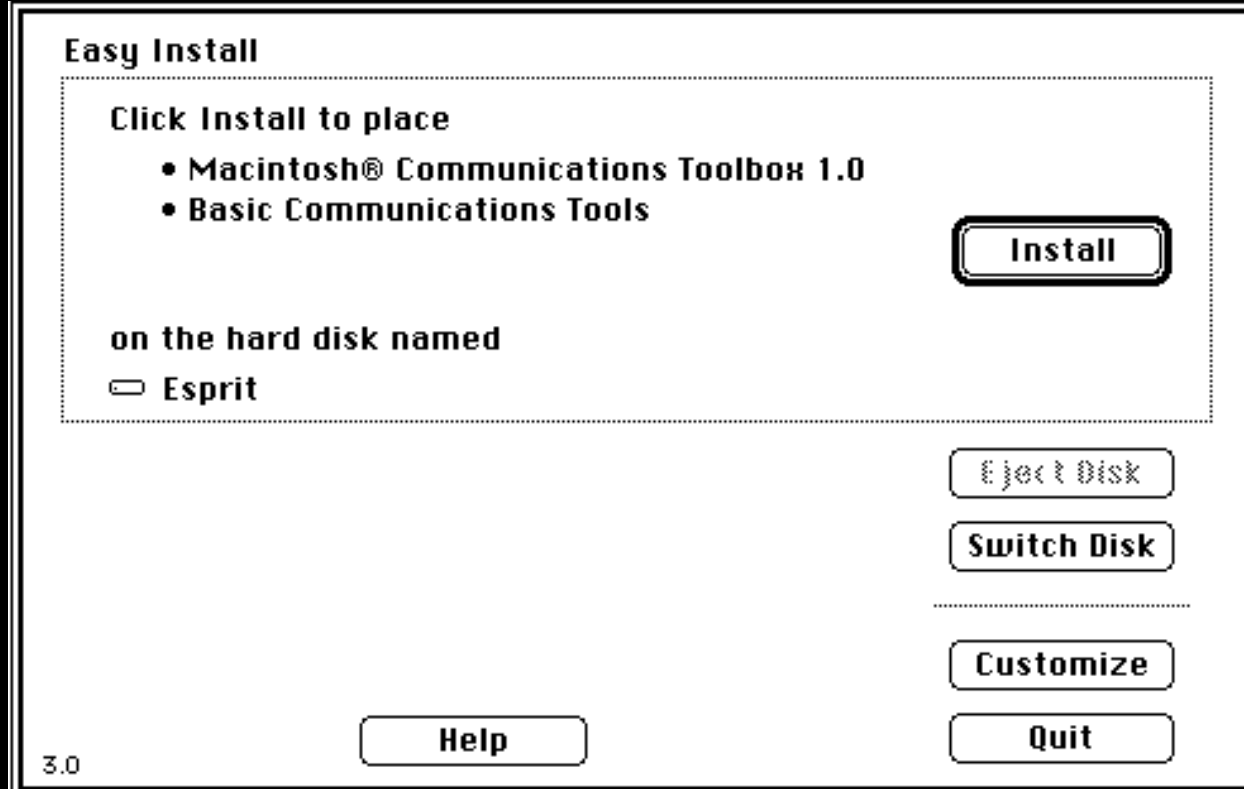
---

<b>Positioning</b>
<b>Overview</b>
<b>The User Experience</b>
<b>Concepts/Details</b>
<b>Runtime Environment</b>
<b>Summary</b>

# Installation

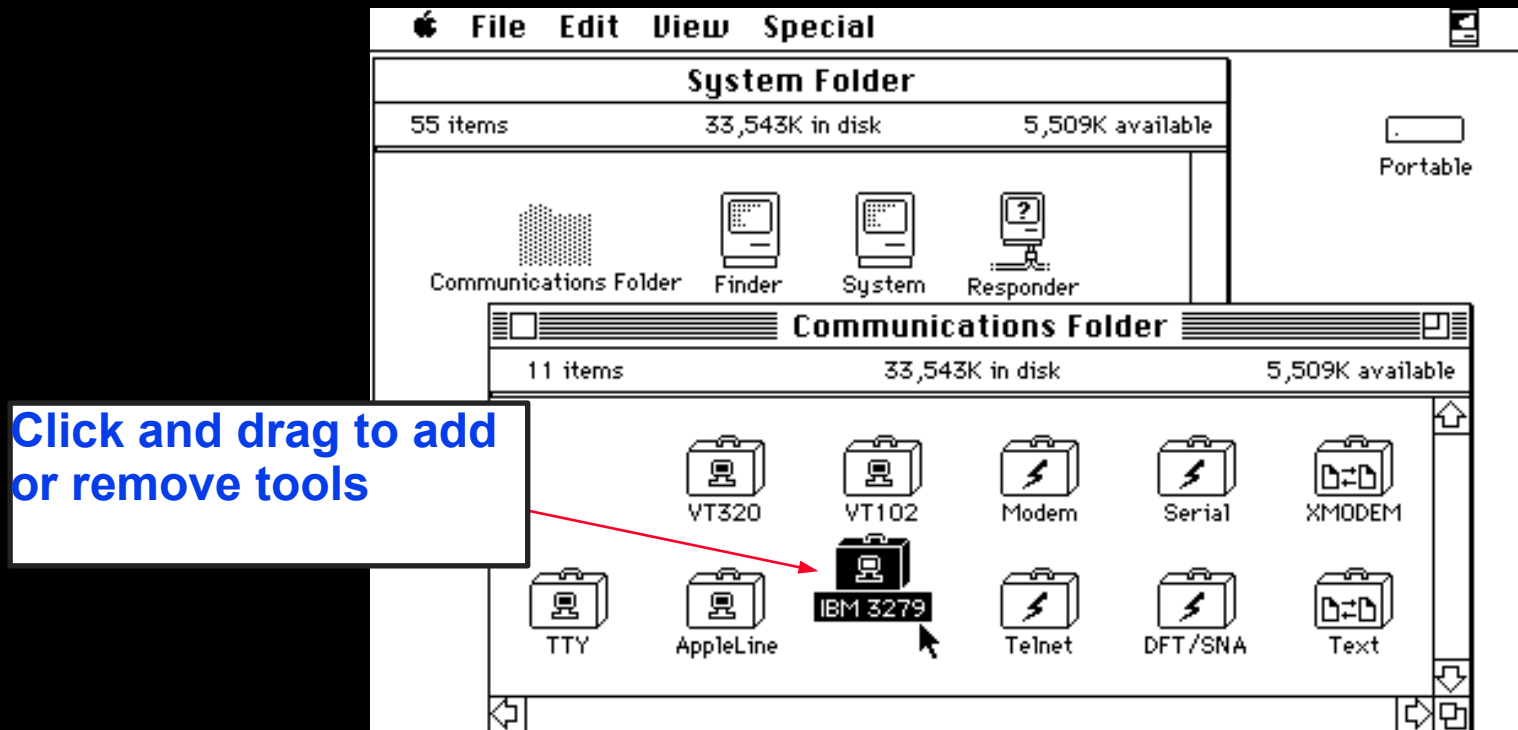
---

## Installing the Communications Toolbox



# Adding and Removing Tools

## Manipulating Communications Tools





# Applications and Tools

---

## Using Applications

- Same application independent of communications service

## Using Communications Tools

- Same user interface independent of applications

# Using Tools

---

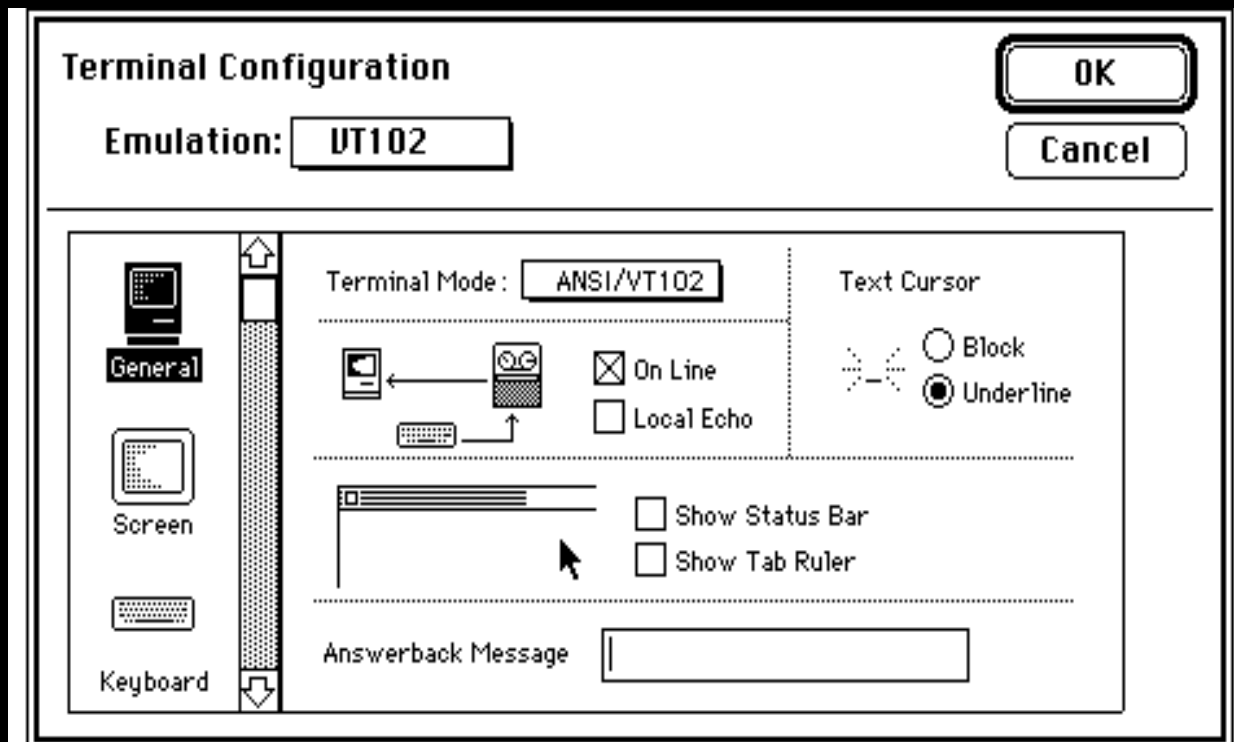
## Choosing the Right Tool

- Application has “default” settings
- Or use settings from documents

# Configuring Tools

---

Configuring is same across all applications



# Concepts and Details

---

<b>Positioning</b>
<b>Overview</b>
<b>The User Experience</b>
<b>Concepts/Details</b>
<b>Runtime Environment</b>
<b>Summary</b>

# Manager and Tools

---

## What the Managers Do

- Provide applications with access to classes of functionality

## What the Tools Do

- Perform all functionality associated with a given communications service

# Virtual Connection Space

---

Application

Connection Manager

Connection Tool

What is a connection?

- Byte stream,  
non-transactional
- Point to point  
(not broadcast)

# Core Routines

---

- Manipulate tool records (allocation, validation, disposal)
- Provide user interface for configuring tool records
- Translate tool configuration string between English and other languages

# Terminal Manager Routines

---

## Manager Routines

- Process events (keystrokes, mouse downs, updates)
- Scrolling

## Application Callbacks

- Mouse handling routines
- Transmit data to remote



# Connection Manager Routines

---

- Open or close a connection
- Read/write data
- Check connection status
- Listen for connection request
- “Break”

# File Transfer Manager Routines

---

## Manager routines

- Send/receive a file
- Stop a file transfer

## Application callbacks

- Send/receive data onto connection

# Comm. Resource Manager

---

## Two Roles

- Internal routines to support Communications Toolbox
- Registration for communications devices, *e.g.* serial port devices register input/output driver names

# Comm. Toolbox Utilities

---

## Popup Menu CDEF

- Easy inclusion of popup menus that conform to Human Interface Guidelines

## Variation Codes

- Use mctb resources
- Use AddResMenu
- Use window font/size

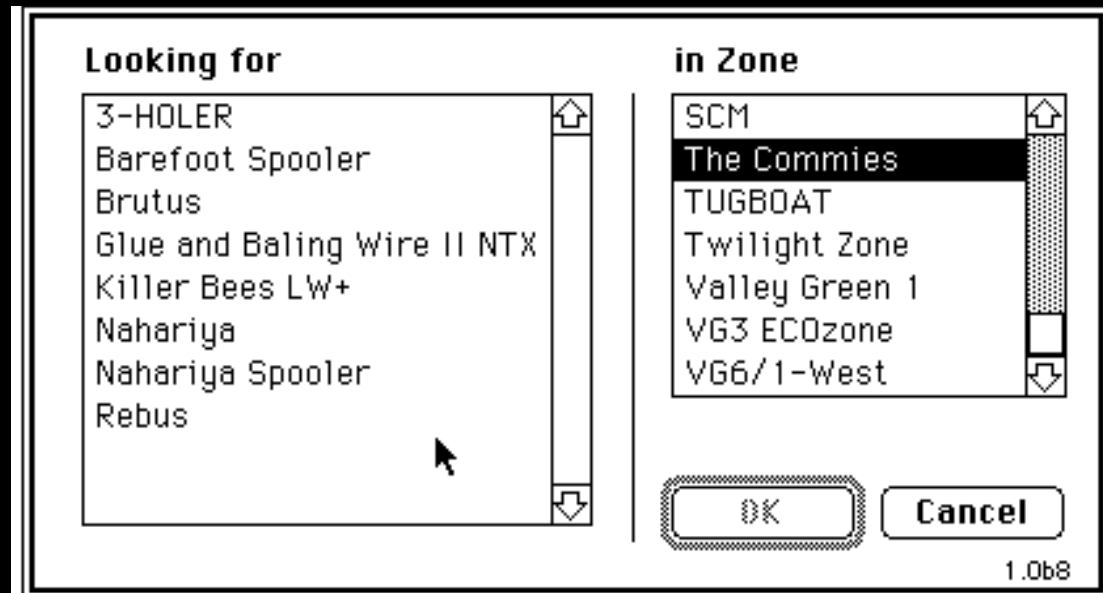
Baud Rate:	✓300
	1200
	2400
	9600
	19200
	57600

# Comm. Toolbox Utilities

---

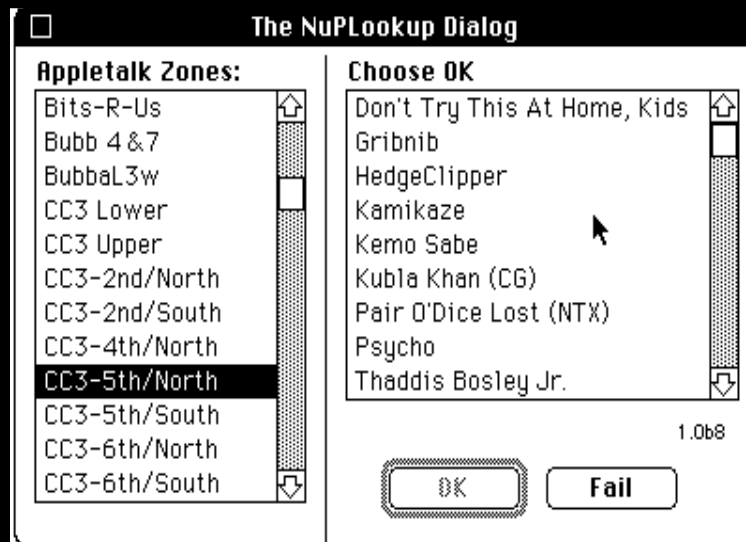
## Network Lookup Package

- Single Call to perform standard NBP Lookups



# Comm. Toolbox Utilities

---



- Fully customizable and configurable
  - Custom dialogs
  - Event filtering
  - Item hit filtering
  - Dim/remove items from zone/item lists
  - Override NBP lookup parameters

# Comm. Toolbox Utilities

---

## Dialog Manager Extensions

- Appending one item list to another
- Shortening dialog item lists

# Runtime Environment

---

<b>Positioning</b>
<b>Overview</b>
<b>The User Experience</b>
<b>Concepts/Details</b>
<b>Runtime Environment</b>
<b>Summary</b>



# Startup Time

---

Installation

- INIT 29 in System file

Access from INIT's

# Memory Models

---

Small memory model (1 MB RAM)

- Managers installed at InitXXX time

Large memory model (>1 MB RAM)

- Managers installed at startup time

# Code Sharing

---

Manager code is shared across all applications

Tools are shared within an application, but not across applications

- Tools should be designed to be reentrant and multiply instantiatable

# Components of a Tool

---

## Multiple Code Resources

- Discrete Entry points
- Message selectors

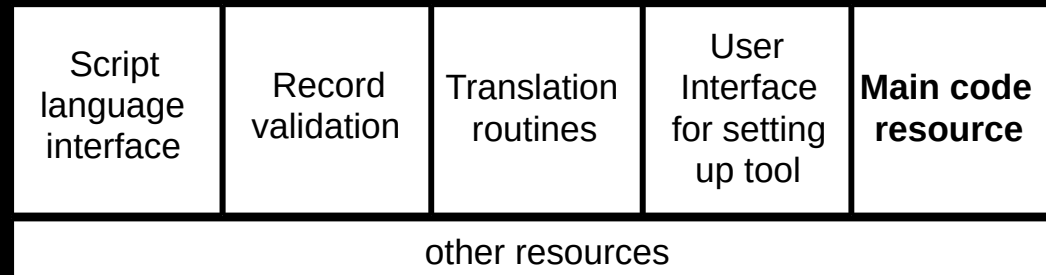
# Message Passing Architecture

---

Application calls routed to appropriate code resource

**Application**

**Manager**



# Tool Files

---

- Code resources
- Other resources
  - 'vers' resources
- File types
  - 'fbnd', 'cbnd', 'tbnd'

# Large Tools

---

- > 32K can be accommodated via MPW<sup>™</sup> Linker  
(but still offsets limited to 32K)
- Perform your own segmentation
- Development environments may provide alternative means

# Tools and Resources

---

## Resource ID's

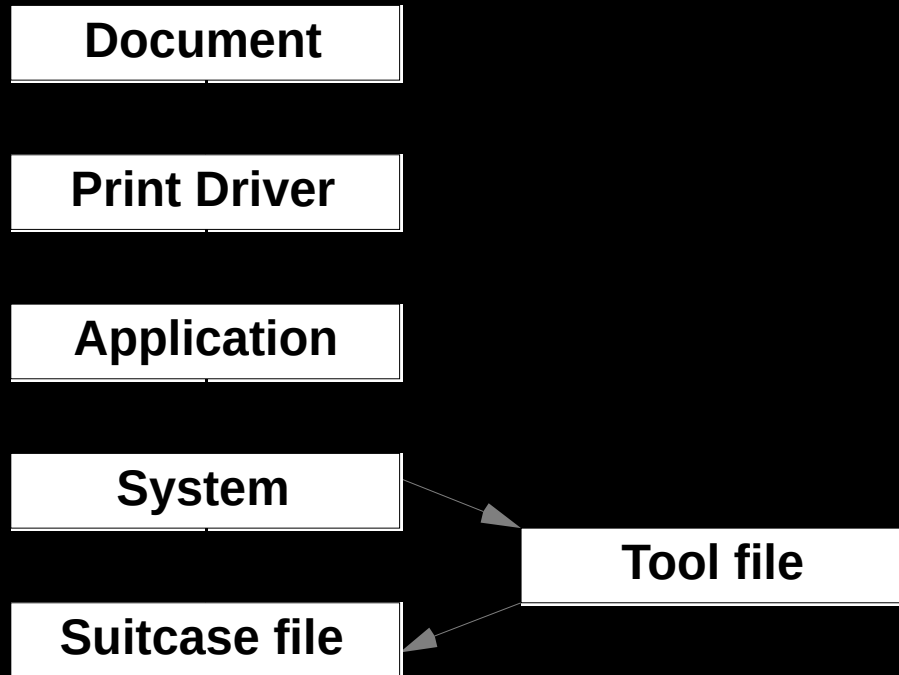
- Heuristically avoid conflicts by choosing “high” resource ID's ( $> 10000$ )
- Potential for conflicts among tools and between tools and the applications or System file still exists



# Resource Conflict Resolution

---

## Algorithm for Resolution



Tools are dynamically patched into and out of resource chain when jumping into/out of tool code resources

# Summary

---

<b>Positioning</b>
<b>Overview</b>
<b>The User Experience</b>
<b>Concepts/Details</b>
<b>Runtime Environment</b>
<b>Summary</b>

# Key Benefits

---

- Applications can be developed independently of services
- Support for services can be developed independently of applications
- Consistent user interface across applications

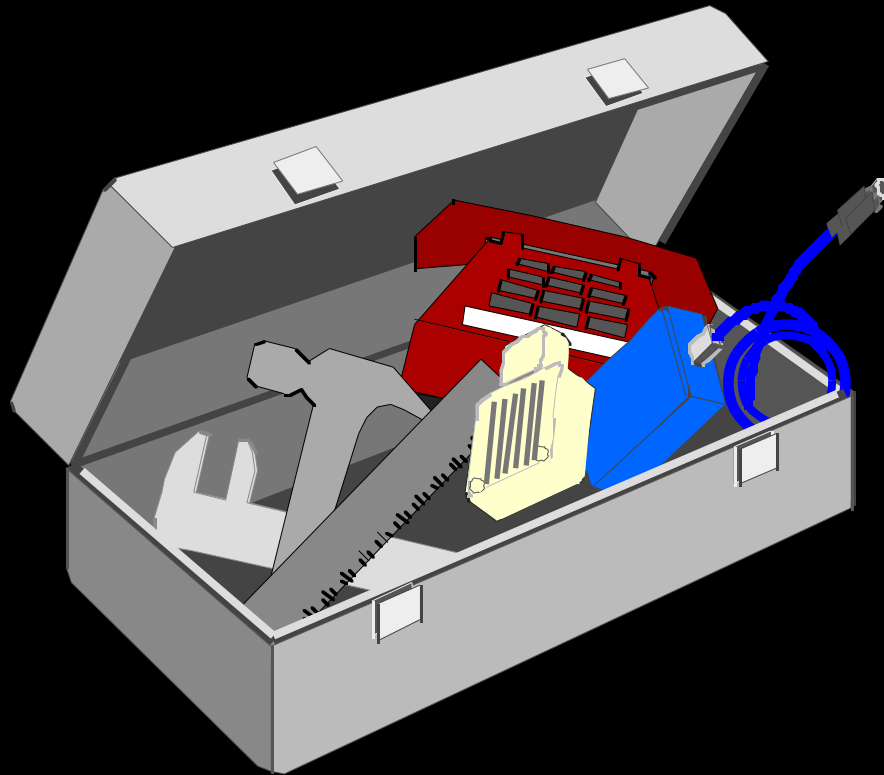
# Future Directions

---

- Libraries of Objects for use with OOP Languages (C++, Object Pascal)
- Expand the Virtual Connection Space to better support transactional environments
- Incorporation of developer and user feedback

# Questions?

---





**The power to be your best™**