Inter-Office Memorandum

To	Common Software Users	Date	Septmber 20, 1979
From	Ladner	Location	Palo Alto
Subject	Common Software Pre-release Version 3.0e - How To Use	Organization	SDD/SS

XEROX

Filed on: <AComSoft>30E>HowToUse.memo, .press

This memo describes how to use the pre-release version 3.0e of Common Software. Each package in the Common Software directory is described in its own section below. This document contains only operational details that are not contained in the document describing the package. Packages will always be referred to by their sub-directory name.

The following packages are included in this pre-release (DCS is Development Common Software, PCS is Product Common Software):

Directory	PCS	Implements a directory system for Pilot;
Envoy	PCS	Provides a remote procedure call facility;
FileStream	DCS	A byte oriented transducer for Pilot files;
GateStream	PCS	A transducer for communicating to foreign devices;
Heap	PCS	A short pointer interface to heap-type storage management;
Pup	DCS	The PUP Protocol package.
String	PCS	A string manipulation package;
Time	PCS	A time editing package;
VMS	PCS	The Virtual Memory Sharing and Atomic Transaction package.

The following packages are part of Common Software but are not available in this pre-release:

FTP	DCS	The PARC File Transfer package.

Where to get it

Unless otherwise mentioned, all files mentioned in this memo as being part of the pre-release should be obtained from the *release directory.* The D0 version is stored on <AComSoft>30E>*PackageName*>. This directory exists on both IRIS and ISIS. No Alto version was produced.

Compiling

Obtain the Pilot definitions from the Pilot release directory (see the Pilot HowToUse document). Retrieve the public definitions modules from the appropriate release directory.

Don't forget that if compiling for the D0, compile with the "-Alto/c" switch. For example:

```
Compile -Alto/c Mumble
```

The binder checks for intermixed -Alto/c and Alto/c code. (The interim compiler switch "Long/c" should *not* be used.)

Symbols

Each of the sections below discusses the disposition of the symbols for the bcds (under Implementation Modules). The general rule that was applied is that if no configuration is involved in producing the end-use bcd then the symbols are combined with the bcd. When this case obtains, no specific mention is made of symbols. If a configuration is necessary, the symbols are separted into separate files and said files are named.

The Directory Package

The Directory package implements a directory system for use with Pilot.

Sub-directory: Directory.

Documentation: PilotDirectory.memo.

Public Definitions Modules Directory.bcd, DirectoryInitialize.bcd, and CommonSoftwareFileTypes.bcd. DirectoryInitialize defines the signal AlreadyRootFile and the procedure InitializeRootAndSystemDirectories. CommonSoftwareFileTypes defines the various file types assigned for Common Software use.

Implementation Modules: DirectoryImpl.bcd.

Binding: The Directory package is used and exported by the Pilot DCS configuration. It is also used by but not bound with the VMS package.

The Envoy Package

The Envoy package implements a remote procedure call function for use with Pilot.

Sub-directory: Envoy.

Documentation: EnvoyFunSpec.press.

Public Definitions Modules: Envoy.bcd.

Implementation Modules: EnvoyImpl.bcd.

Binding: The Envoy package uses the Heap package, and the AppendString and EqualString procedures from the String package (StringsImplA).

The FileStream Package

The File Stream package implements a Pilot transducer for accessing Pilot files as a serial byte stream. There are two components of the file stream transducer. One implements the byte stream for an unadorned Pilot file, figuring the file length to the next largest page. The other utilizes a leader page to provide among other things, a file length to the exact byte.

Sub-directory: FileStream.

Documentation: FileStream.press, and LeaderPage.press.

Public Definitions Modules: FileStream.bcd, LeaderPage.bcd, FileStreamTypes.bcd. FileStream provides the interface to access any Pilot file, LeaderPage provides the leader page (= byte length files) definitions, and FileStreamTypes defines some file types for leader page files.

Implementation Modules: FileStreamInstance.bcd.

Binding: FileStreamInstance requires the Directory package.

The GateStream Package

The Gate Stream package provides a Pilot transducer for accessing foreign devices.

Sub-directory: GateStream.

Documentation: GateFunc2.0.press.

Public Definitions Modules: GateStream.bcd.

Implementation Modules: Gate.bcd. This is a configuration which contains all the necessary code. The symbols files have been generated for the sub-configurations only. These are: BSCDriver.symbols, GateFramework.symbols, and GateUtilities.symbols.

The Heap Package

The Heap package provides a short pointer storage management function.

Sub-directory: Heap.

Documentation: HeapDoc.press.

Public Definitions Modules: Heap.bcd.

Implementation Modules: HeapImpl.bcd.

Binding: HeapImpl uses the String package procedure WordsForString (StringsImplA).

The Pup Package

The Pup package implements the Pup family of communication protocols, and is included as parrt of Development Common Software on the D0 for backwards compatibilitty with existing Pup-based servers in the development environment.

Sub-directory: Pup.

Documentation: Consult "Mesa Pup Package Functional Specification", Version 5.0, April 1979, released with the Alto Mesa 5.0 System. In addition, the memo HowToUsePupPackage.press describes differences from the Alto package, and points out caveats.

Public Definitions Modules: PupStream.bcd, EFTPDefs.bcd. You may also need PupTypes.bcd, BufferDefs.bcd, and PupDefs.bcd in the Pilot release directory. The purpose of each of these defs files is detailed in the documentation.

Implementation Modules: Pup.bcd. This is a configuration containing all the necessary code. The symbols are contained in the file Pup.symbols.

Binding: The Pup package must be bound with the Heap package and the Pilot DCS configuration. Optionally, the configuration CommunicationStats.bcd may be bound in as well. This configuration permits printing of statistics gathered by the Pup package and its stub in Pilot. Consult the HowToUsePupPackage memo for details.

The String Package

Sub-directory: String.

Documentation: For documentation consult the section "Alto/Mesa String Package" in the Mesa System Documentation manual, version 5.0.

Public Definitions Modules: String.bcd.

Implementation Modules: StringsImplA.bcd, StringsImplB.bcd. The implementation of the String package is contained in two modules. The following list shows which functions are defined in which module. The names are in alphabetical order by module.

StringsImplA:

AppendChar: PROCEDURE
AppendString: PROCEDURE
AppendSubString: PROCEDURE
DeleteSubString: PROCEDURE
EqualString: PROCEDURE
EqualStrings: PROCEDURE
EqualSubString: PROCEDURE
EqualSubStrings: PROCEDURE
EquivalentString: PROCEDURE
EquivalentStrings: PROCEDURE
EquivalentSubString: PROCEDURE
EquivalentSubStrings: PROCEDURE
StringBoundsFault: SIGNAL
WordsForString: PROCEDURE

StringsImplB:

AppendDecimal: PROCEDURE
AppendLongDecimal: PROCEDURE
AppendLongNumber: PROCEDURE
AppendNumber: PROCEDURE
AppendOctal: PROCEDURE
bcplStringOverflow SIGNAL
BcplToMesaString: PROCEDURE
InvalidNumber: SIGNAL
LowerCase: PROCEDURE

mesaStringOverflow: SIGNAL
MesaToBcplString: PROCEDURE
Overflow: SIGNAL
StringToDecimal: PROCEDURE
StringToLongNumber: PROCEDURE
StringToNumber: PROCEDURE
StringToOctal: PROCEDURE
UpperCase: PROCEDURE
WordsForBcplString: PROCEDURE

The Time Package

The Time package provides functions to acquire and edit times into strings.

Sub-directory: Time.

Documentation: Time.doc.

Public Definitions Modules: Time.bcd.

Implementation Modules: TimeImpl.bcd.

Binding: This package uses the AppendChar and AppendString procedures of the String package (StringsImplA).

The VMS Package

Sub-directory: VMS.

Documentation: There are several documents relating to this package. The file HowToUseVMSharing.press should be used to get started.

Public Definitions Modules: VMSharing.bcd and VMSharingClient.bcd.

Implementation Modules: VMS.bcd. This is a configuration containing all the necessary code. The symbols are contained in the file VMS.symbols.

Binding: The VM Sharing package requires special attention when binding. It interacts with PilotClient.Run and with the Directory package. ***Be sure to consult the documentation before proceeding.***