

1978

- **DPS** (Distributed **P**resentation **S**ervices)

1979

- **GDDM** (Graphical **D**ata **D**isplay **M**anager)
- **SNATAM** (Systems **N**etwork **A**rchitecture **T**erminal **A**ccess **M**ethod)
- **GDDM/PGF** (Presentation **G**raphics **F**acility)

1980s

1980

- **REXX** (Restructured **E**xtended **E**xecutor)
- **IPG** (Interactive **P**resentation **G**raphics)

1984

- **GCP** (Graphics **C**ontrol **P**rogram) pc support
- **GOCA** (Graphics **O**bject **C**ontent **A**rchitecture)
- **CICS/CMS** (**C**ustomer **I**nformation **C**ontrol **S**ystem **C**onversational **M**onitor **S**ystem)
- **GDDM/IVU** (**GDDM** **I**mage **V**iew **U**tility)

1986

- **5777-EAD + DTP**

1987

- **OS/2 Presentation Manager**
- **SAA** (Systems **A**pplication **A**rchitecture)
- **ADA DOD**, MOD, defence related software
- **LEXX** (Live **P**arsing **E**ditor)
- **IAX** (Image **P**rocessing **s**ystem)
- **Passfield** horticultural solution

1988

- **Toolsrun**
- **LPEX**
- **ORYX**
- **Finisterre**
- **Stonehenge**

1989

- **CICS/ESA V3**
- **CICS/OS/2**

1990s

1990

- **GPE** (Graphical **P**rogramming **E**nvironment)

1991

- **AconnS** (Application **C**onnection **S**ervices)

1992

- **GDDM V3.1**

1993

- **Mayflower**
- **GoServe**

1994

- **CICS/AIX**
- **CallPath**
- **DirectTalk**

1995

- **CICS** client for Windows & Mac

1996

- **Java Development toolkit** for OS/2, AIX & RS6000

1997

- **NetRexx**

1999

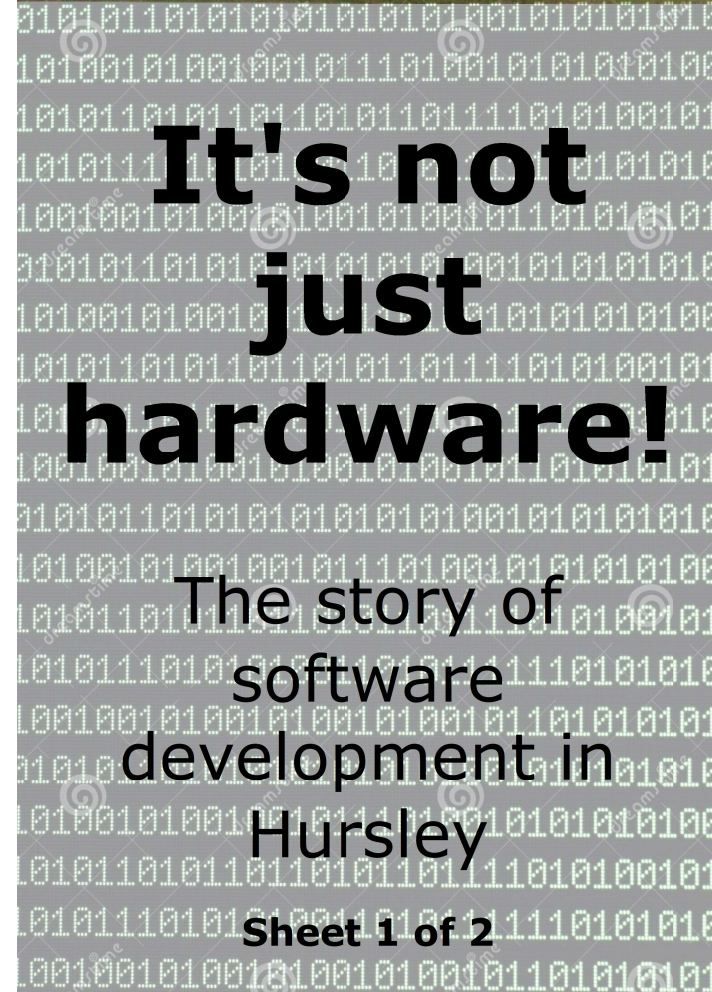
- **RMI** over **IIOP** Joint development with Sun



Find the museum online:

https://ibmhursleymuseum.info/the_museum.asp

This document was created by the Hursley Museum Curator Team. Copyright 2025.



What is software?

Software is the name given to a collection of programs, data, and instructions that tell a computer how to perform specific tasks. Unlike hardware, which refers to the physical components of a computer, software consists of the virtual elements that enable the hardware to work effectively.

There are two main types of software, System Software and Application Software:

- **System software** controls a computer's internal functioning, including managing hardware resources and providing essential services.
- **Application software** directs the computer to execute specific commands given by the user and includes programs like word processors, spreadsheets, and games.

Software Development in Hursley, early days

The **IBM British Laboratory** arrived at **Hursley** in 1958. It was still the very early days of computers, but a Hursley team, led by **John Fairclough**, started to explore magnetic storage for the control of a small computer. This project was called **SCAMP** (Scientific Computer and Modulator Processor).

SCAMP was the first computer to use the concept of microprogram control, and the start of software development in Hursley. One of the features of SCAMP was the read-only storage wheel (below), a permanent



storage device that stored the microcode (system software) instructions used by the computer system.

The experience gained on SCAMP led to Hursley's involvement with the development of the **IBM S/360**, specifically the model 40, which IBM designed from the beginning to be an architecture of compatible processors and hardware.

Before the **IBM S/360** was developed business and scientific users used different programming languages for different hardware. Business users were using languages such as **COMTRAN** and **COBOL**, whereas scientific users programmed in **Fortran**, **ALGOL**, **GEORGE**, and others.

The **IBM S/360** was designed as a common architecture for both business and scientific users, and IBM wanted a single programming language for all users. The hope was that **Fortran** (developed by **IBM** in the late 1950s) could be extended to include all the features needed for the S/360 architecture, but this was not possible, so, in 1964, a new language was designed: **PL/I** (with the Roman numeral for one, "I").

Initially developed in **IBM** locations in New York and San Jose, responsibility for the **PL/I** compilers was transferred to **Hursley** in the mid 1960s. **Hursley** then became a major site for software development within **IBM**, particularly for compilers and transaction processing software.

- A **Compiler** is a computer program that translates code written in one programming language (ie. FORTRAN or PL/I) into a low-level programming language (ie. Assembly language, object code, or machine code) to create an executable program.
- **Transaction processing** is the processing of all the information in a transaction from start to finish. For example, withdrawing money from an ATM is a transaction and the computer processes all steps in the transaction, including error handling if anything goes wrong.

The following is a list of software developed at **IBM Hursley** over the years to the best of the curators' knowledge. If you are aware of any software missing from the list, please contact the curators.

Because so many software projects were developed in Hursley, this trifold provides only a list of their names, or codewords.

For more details of the software, please refer to **Software Development Trifold 2**.

Hursley software developed by year

1960s

1964

- **PL/I**
- **PL/I F** compiler
- **SUPPAK** simulator.

1965

- **CAS** (Control Automation System)

1969

- **PL/I** Optimising Compiler

1970s

1970

- **PL/I** Checkout Compiler

1971 - 1974

- **VTAM** (Virtual Telecommunications Access Method) Development started in Hursley and was transferred to **IBM Kingston**, and released, in 1974

1973

- **Cyrillic** language support for **IBM 3741/2**
- **PFC** (Program Function Control) for **IBM 3741/2**

1974

- **INPS** (Italian social security. update software)
- **CICS** (Customer Information Control System) is transferred to Hursley from **IBM Palo Alto**.
- **PL/I** and all other compilers transferred to **IBM Santa Teresa**

1976

- **GAM** (Graphics Access Method).

1977

- **'Ironman'** – programming language for a Customer embedded system.
- **PADAS** (Product Assurance Design Automation System).