

The Resource

A Management Overview

- ☐ What It Is
- ☐ Why We Need It
- ☐ How We Are Using It

Slide 2

"Resource" - A Definition

The IEC 1131-3 term that refers to the control strategy that executes on a single control node

Slide 3

"Resource" - An Alternative Definition

Runtime
Environment
Supporting the
Operation of
User Programmed
Real-time
Control
Equipment

Slide 4

What Does The Resource Do For Us?

- ☐ A rich support environment for control applications
- ☐ A standardised firmware platform for potentially all configurable products

Slide 5

(1) A Rich Environment

- ☐ Application Framework
 - ↳ Library of useful application functionality
 - ↳ Virtual I/O
 - ↳ Services (RPC)
 - ↳ Remote Data Cache (VAR REFERENCE)
 - ↳ File Management
 - ↳ Parameterisation...
 - ↳ Set of tools, programs and services
 - ↳ Application Download / Upload
 - ↳ Control Shell
 - ↳ Communications...

Slide 6



(1) A Rich Environment (Contd)

- Functionality for free
 - ✦ Management of the user application
 - ✦ Database, Templates...
 - ✦ Network Distribution
 - ✦ Multi-Tasking
 - ✦ Inter-Task messaging
 - ✦ Data Coherency
 - ✦ Operating System / Executive Independence
 - ✦ SFC Function Block Engine
 - ✦ Runtime Communications Access
 - ✦ Application Debugging Facilities

Slide 7



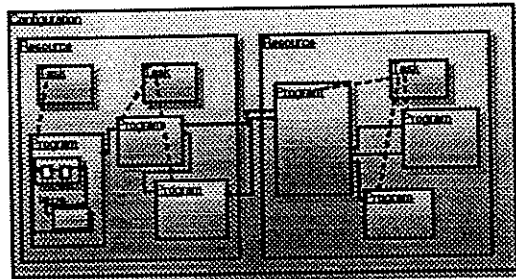
(2) A Standardised Platform

- Reusability of firmware
- Common backbone of products
- Portability of algorithms
- Common look and feel to all instruments using it

Slide 8



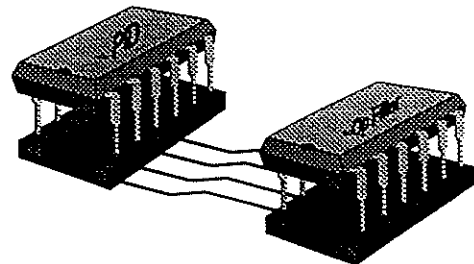
IEC 1131-3 : An Overview



Slide 9



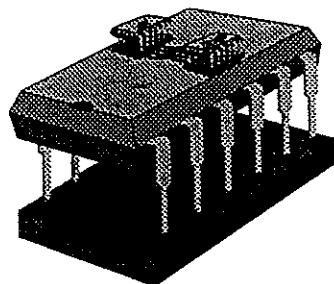
IEC 1131-3 : Function Blocks



Slide 10



IEC 1131-3 : Bigger Function Blocks

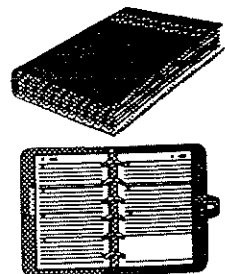


Slide 11



IEC1131-3: Customised Blocks

- Application built from :-
 - ✦ Our standard control blocks
 - ✦ Customers own application specific blocks
- Application program solution reflects the real world problem



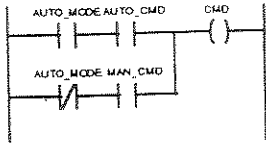
Slide 12



IEC 1131-3 : Languages (1)

- IL (Instruction List)


```

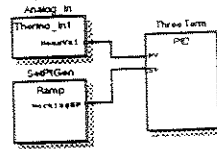
      RESET: LD 0
            ST OFLO
            JMP NEXT
      
```
- LD (Ladder Diagram)
 

Slide 13

IEC 1131-3 : Languages (2)

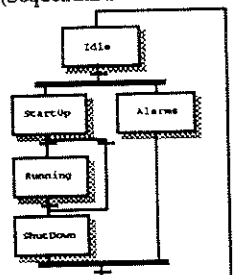
- ST (Structured Text)


```

      IF Manual THEN
        DoSomething/Quickly;;
      END_IF;
      
```
- FBD (Function Block Diagram)
 

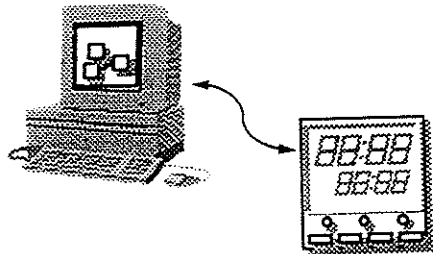
Slide 14

IEC 1131-3 : Languages (3)

- SFC (Sequential Function Chart)
 

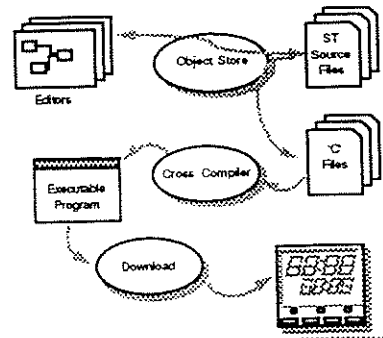
Slide 15

Where Does GCT Fit In



Slide 16

Building An Application With GCT



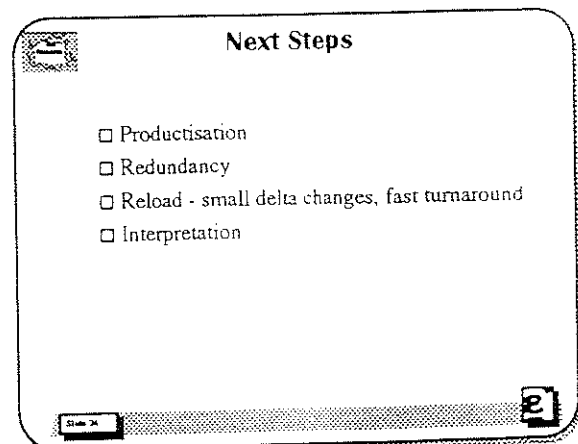
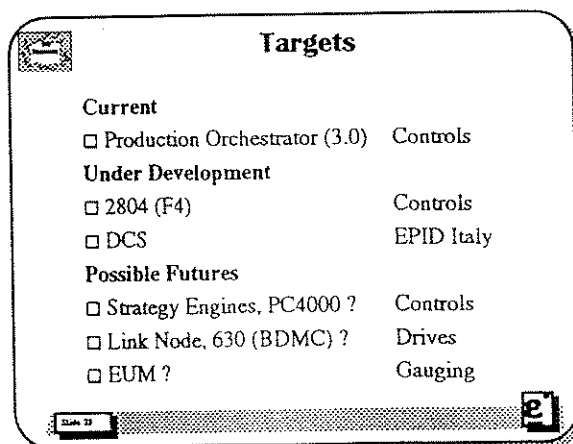
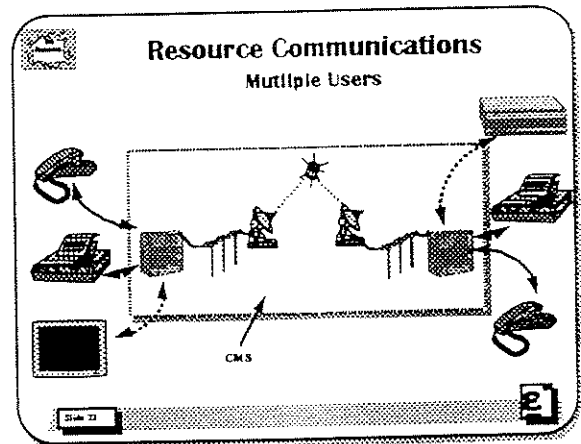
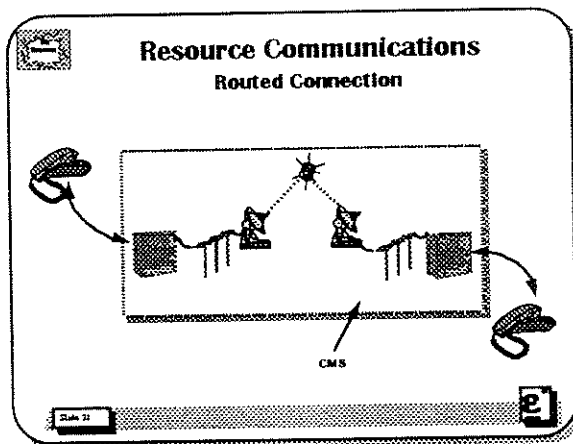
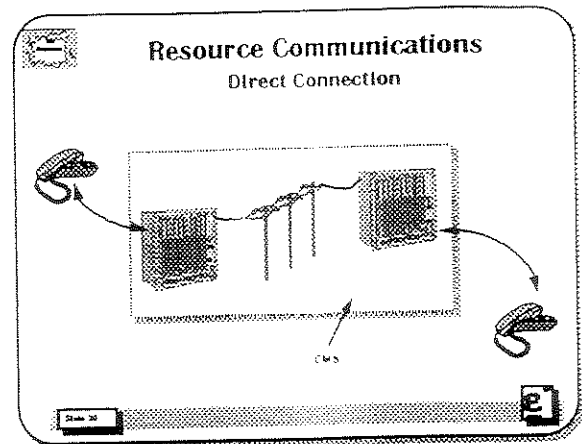
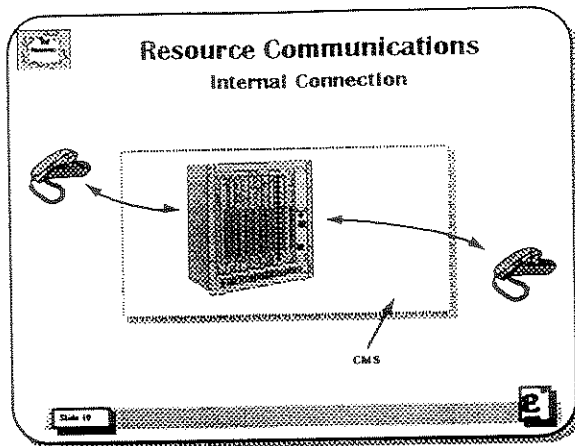
Slide 17

Resource And GCT

The Resource provides GCT with

- Application control
 - ↳ Download, upload, start, stop
- Application monitoring
 - ↳ Parameter read / write
- Application debugging
 - ↳ Breakpoints, single step, single shot block / task
- A simulation environment under Windows

Slide 18





Summary

The Resource is

- ☐ A run-time environment
- ☐ Aimed at configurable products
- ☐ Configured by GCT
- ☐ Portable, scalable, distributable, ...
- ☐ Mature (ish)
- ☐ An exciting opportunity for the Eurotherm Group to provide inter-operable products!

Slide 12



The Resource

A Technical Overview

Slide 1

The Resource Project



Slide 2

The Resource Project - What is it ?

It was originally conceived as an implementation of an IEC1131-3 database from the RESOURCE level down - hence the name.

Slide 3

The Resource Project - What Happened ?

It became apparent that the IEC1131-3 standard did not go "far enough" and there was a need to address other issues, especially communication.

Slide 4

The Resource Project - What Happened (2) ?

The project evolved to provide communications support for access to the database. This in turn evolved into a general communication system.

Slide 5

The Resource Project - What Happened (3) ?

From this point it grew to encompass many of the "services" that might be required of a real-time control application.

Slide 6



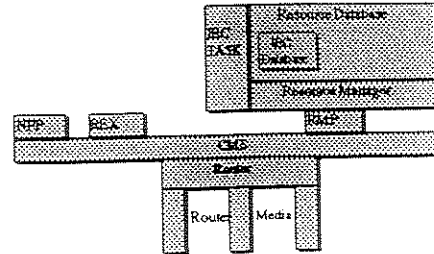
The Resource Project - What does it contain now ?

- ☐ IEC1131-3 database
- ☐ IEC1131-3 Execution Engines
- ☐ IEC1131-3 Function Library
- ☐ Extensions to the IEC1131-3 including communications
- ☐ Communications Messaging Services
- ☐ File System
- ☐ Remote Execution

Slide 7



The Resource Project - Architecture



Slide 8



The Resource Project - What does it do for us ?

- ☐ Provide for portable applications
- ☐ Common development framework
- ☐ Permit greater inter-operability between different products.
- ☐ Reduce time-to-market for developments

Slide 9



The Resource Project - What type of products ?

- ☐ GCT configured
- ☐ Flexible
- ☐ Distributed
- ☐ Closely coupled
- ☐ Not those with "fixed" functionality (e.g. Discrete 2000 B, M, L)

Slide 10



IEC1131-3



Slide 11



IEC1131-3 - What is it ?

It is an international standard for programming languages for industrial control equipment.

Slide 12



IEC1131-3 - What's different ?

- ☐ CONFIGURATIONs not supported
- ☐ VAR_ACCESS definition level
- ☐ Arbitrary FUNCTION_BLOCK TASK associations

Slide 12

IEC1131-3 - Extensions

- ☐ Extra types (e.g. QTIME, EDGE)
- ☐ Resource Level Wiring
- ☐ VAR REFERENCE
- ☐ SERVICE

Slide 13

IEC1131-3 - A Standard ?

- ☐ Is a pick list of features
- ☐ Some "types" are subject to interpretation
- ☐ Some functions are not fully defined

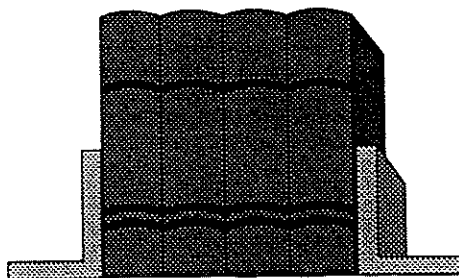
Slide 14

IEC1131-3 - Why use it ?

- ☐ Provides a portable framework for custom applications
- ☐ Provides a portable framework for generation and expression of custom function blocks
- ☐ Solutions can be expressed in customers terms not ours

Slide 15

The Resource Database



Slide 16

The Resource Database - What is it ?

A database of the structure of the data of an IEC1131-3 application.

Slide 17

The Resource Database - What is in it ?

- ☐ Contains descriptions (templates) of all function blocks
- ☐ Contains function block relationships
- ☐ Contains top level block execution ordering
- ☐ Instance data

Slide 19



The Resource Database - Templates

- ☐ A template is a description of an item containing :-
 - ☐ Mode (INPUT, OUTPUT ...)
 - ☐ Type (DINT, REAL, function block type ...)
 - ☐ Name

Slide 20



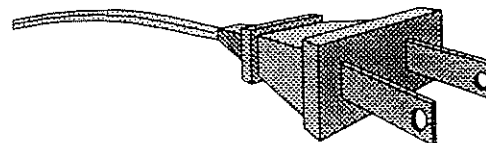
The Resource Database - Why have it ?

- ☐ Makes application structure accessible at run-time
- ☐ Provides a means to locate data without prior knowledge of the application

Slide 21



Resource Level Wiring



Slide 22



Resource Level Wiring - What is it ?

Defines the connections between the top level blocks at run-time. The IEC1131-3 only allows initial values.

Slide 23




Resource Level Wiring - Why use it ?

- ☐ Provides some simple inter RLO communication.
- ☐ Provides run-time flexibility - less is hard coded. This means that a RESOURCE can be partially reconfigured on-line without having to return to GCT.

Slide 24



Var References



Slide 25

Var References - What are they ?

A Var Reference is a mechanism which can be used to reference any CDL object that is not declared within the normal scoping rules.

Slide 26

Var References - What can be referenced ?

- ☐ Any individual data item e.g. :-
 - ✧ A DINT
 - ✧ An element of an ARRAY
- ☐ An ARRAY of items e.g. :-
- ☐ A collection of items
 - ✧ Function block
 - ✧ Structure

Slide 27

Var References - As if local

- ☐ Add keyword "REFERENCE".
- ☐ Set up reference string to point to remote data.
- ☐ Set up a scan rate (for data to be read).

```

VAR_INPUT
SP: REAL;
END_VAR

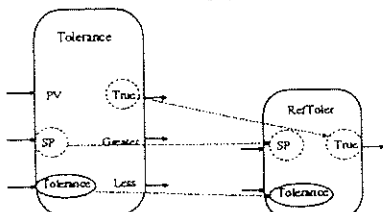
PID( SP := SP );

VAR REFERENCE
SP: REAL;
ref := 'MASTERLoop.SP';
scan := T#500ms;
END_VAR

PID( SP := SP );
  
```

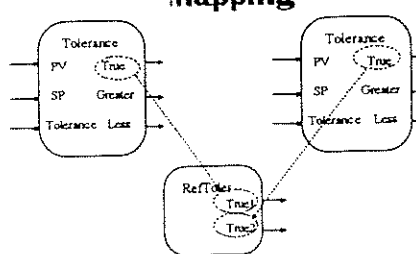
Slide 28

Var References - Subset Mapping



Slide 29

Var References - Arbitrary Mapping



Slide 30

Var References - Dynamic Run Time Control

- ☐ Remap (using ^ref property) to other "equivalent" remote data.
- ☐ Alter scan rate (using ^scan property) of remote data.

Slide 31



Var References - Dynamic Run Time Address Resolution

- ☐ "Node" is identified by RESOURCE name and not by number.
- ☐ Data is identified by name and not register number. This means no "gate files". This allows both producer and consumer to be updated without "re-mapping".

Slide 32



Var References - Template Matching

- ☐ All data items of the same type (e.g. DINT)
- ☐ ARRAY dimensions are equivalent
- ☐ Modes (i.e. INPUT/OUTPUT) are the same
- ☐ Write Protection

Slide 33



Var References - Status

- ☐ ^status - status of last/current operation
- ☐ ^readStatus - status of last/current read
- ☐ ^timeStamp - date and time of data
- ☐ ^newRead - new data sample available

Slide 34



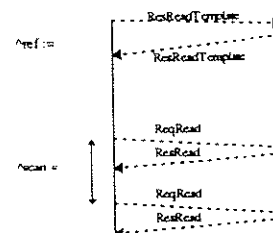
Var References - "Equivalent ARRAYS"

- ☐ ARRAYS have same number of elements (e.g. ARRAY [1..4,1..2] OF DINT is "equivalent" to ARRAY[1..8] OF DINT.
- ☐ This is termed the "shape" of an ARRAY
- ☐ STRINGS are equivalent even if of different lengths (e.g. STRING(20) is equivalent to STRING(40)).

Slide 35



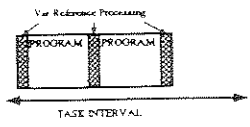
Var References - RMP



Slide 36



Var References - Coherence



- Access once per TASK cycle
- Access in-between RLB (PROGRAM) executions


Slide 37

Var References - Why use them ?

- Access to remote data as if local
- Own "view" of remote data
- Dynamic run-time control
- Dynamic run-time address resolution
- Template matching
- TASK coherence

Slide 40

Services



Slide 41

Services - What are they ?

- SERVICES are a Eurotherm extension to the IEC1131-3 standard.
- They are extra methods on a function block.

Slide 42

Services - What do they provide ?

- Remote procedure calls
- Rendezvous
- Additional methods on blocks

Slide 43

Services - Example


- EMPTY replaces setting a variable(s)
- The ACCEPT provides queuing of request until ready.

```

FUNCTION_BLOCK TASK
  FILLING: BOOL;
  TARGET_LEVEL: REAL;
  SERVICE EMPTY
    TARGET_LEVEL := 0.0;
  END_SERVICE

  WHEN NOT FILLING ACCEPT EMPTY;
END_FUNCTION_BLOCK
  
```

Slide 44




Services - Users

- ☐ The block itself
- ☐ The instantiator of the block
- ☐ Var References

Slide 43

2




Services - Why use them ?

- ☐ Easier to write distributed applications
- ☐ More modular code
- ☐ Easier to express real-world functionality
- ☐ Security of use

Slide 44

2



Services - Var Referenced

- ☐ Templates must match exactly
- ☐ All INPUTs are written for every invocation
- ☐ All OUTPUTs are read on completion

Slide 47

2




The Resource Manager



Slide 48

2




The Resource Manager - What is it ?

A library of methods (functions) and tools for accessing the database.

Slide 49

2



The Resource Manager - Database Loader

Translates the GCT output into the run-time database. i.e., instances the RESOURCE.

Slide 50

2

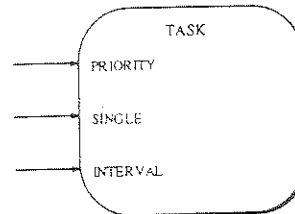
The Resource Manager - Task Engine

- ☐ Executes the function blocks in the required order.
- ☐ Implements the TASK function block (i.e. TASK INTERVAL)
- ☐ Schedules the CMS message handling
- ☐ Implements the TASK state machine

Slide 31



The Resource Manager - The IEC TASK block



Slide 32



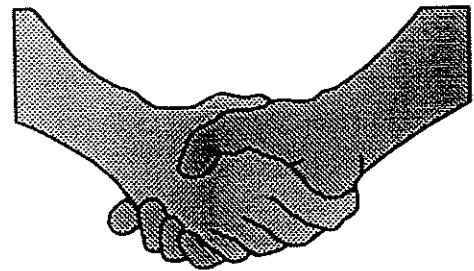
The Resource Manager - Tool Set

- ☐ Report the database structure
- ☐ Modify instance data
- ☐ Change Resource Level wiring
- ☐ Obtain Var Reference Diagnostics
- ☐ Modification and inspection of Var Reference properties
- ☐ A Resource Debugger

Slide 33



RMP



Slide 34



RMP - What is it ?

The Resource Messaging Protocol (a.k.a. RDP) is a CMS protocol designed to provide access to the resource database

Slide 35



RMP - Services

- ☐ Read Template
- ☐ Simple Read
- ☐ Simple Write
- ☐ Complex Write then Read
- ☐ Service
- ☐ Read Description
- ☐ Debug Command and Output

Slide 36



RMP - Debugging

The RMP provides a protocol for the setting of break and trace points within ST.

Slide 37

Virtual I/O

Slide 38

Virtual I/O - What is it ?

Virtual I/O is a mechanism to decouple the application from the I/O hardware.

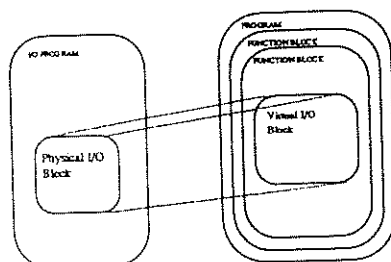
Slide 39

Virtual I/O - How ?

- ☐ Algorithms are written making references only to "virtual I/O" blocks. These are common across all targets and have no hardware dependency.
- ☐ All the hardware "physical I/O" blocks are referenced from the virtual I/O.
- ☐ All address resolution is through the resource database at done at "cold start" time.

Slide 40

Virtual I/O - Illustrative

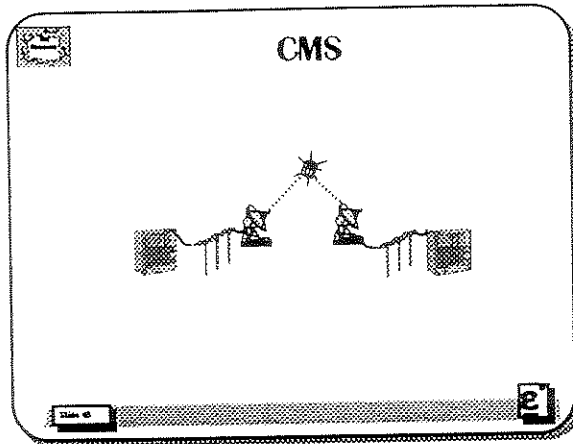


Slide 41

Virtual I/O - Why use it ?

- ☐ Makes application portable to different Resource Manger targets.
- ☐ Enables simulation off-line (e.g. from GCT)

Slide 42



CMS - What is it ?

A set of communications messaging services capable of delivering messages of different protocols across different media.

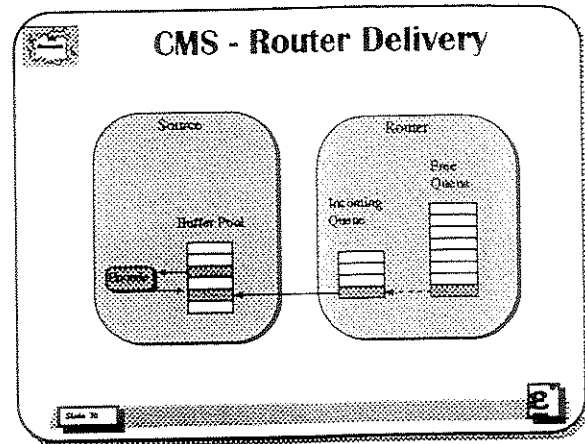
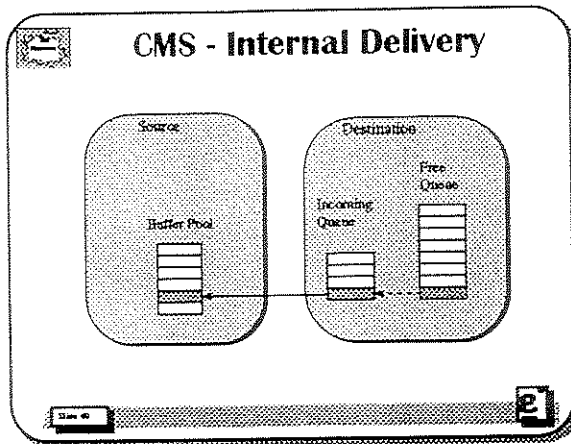
Slide 41

- ## CMS - Objects
- ☐ Node
 - ☐ Process
 - ☐ Access Point (c.f. SAP in MMS)
 - ↳ Single protocol
 - ↳ Optional application entity name
- Slide 42

- ## CMS - Addressing
- ☐ Logical - By application entity name only
 - ☐ Physical - By node plus :-
 - ↳ Application entity name
 - ↳ Access point number
 - ↳ Protocol
- Slide 43

- ## CMS - The Router
- ☐ One router per CMS node
 - ☐ Each router may have any number of media
- Slide 47

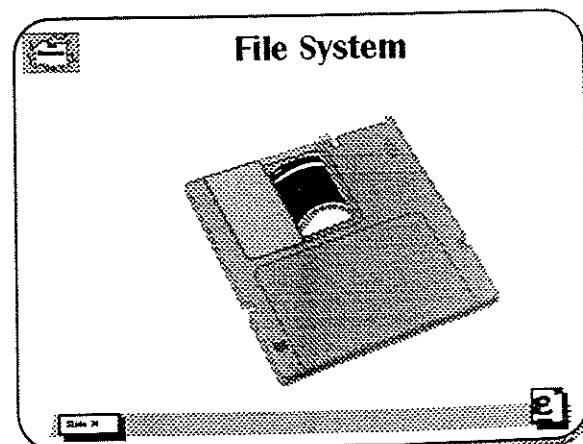
- ## CMS - Delivery
- ☐ Internal - To another CMS process within the same node
 - ☐ Direct - To another CMS process on another node via a direct link between them
 - ☐ Routed - To another CMS process on another node via other CMS nodes
- Slide 48



- ### CMS - Protocols
- ☐ RMP - Resource Messaging Protocol
 - ☐ NFP - Network File Protocol
 - ☐ REX - Remote EXecution Protocol
 - ☐ ROU - ROuter Protocol
- Slide 71

- ### CMS - Media
- ☐ UDP (Unconfirmed Datagram Protocol)
 - ☐ Serial point-to-point (using CSS)
 - ☐ TCP/IP (No longer in use)
 - ☐ EIC/HIN (Used internally)
 - ☐ EILIN (Obsolete)
- Slide 72

- ### CMS - Why use it ?
- ☐ Is medium independent
 - ☐ Highly portable
 - ☐ Efficient local communications
 - ☐ Other Eurotherm companies are using it
 - ☐ Supports indirect connections
 - ☐ Supports many protocols
- Slide 73





File System - What is it ?

- ☐ A CMS protocol for file transfers
- ☐ Common set of programming interfaces (similar to the Posix model)

Slide 75



File System - Interfaces

- ☐ Local File System (LFS)
- ☐ Remote File System (RFS)
- ☐ General File System (GFS)

Slide 76



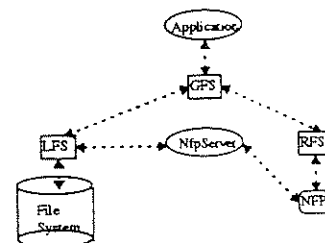
File System - NFP Services

- ☐ Open
- ☐ Close
- ☐ Read
- ☐ Write
- ☐ Append
- ☐ Delete

Slide 77



File System - Architecture



Slide 78



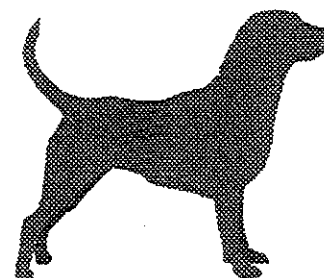
File System - Why use it ?

- ☐ Portable
- ☐ CMS compliant

Slide 79



REX



Slide 80



REX - What is it ?

The REX protocol is a CMS protocol based around the "telnet" protocol.

Slide 11

REX - Uses

- ☐ Downloading RESOURCE
- ☐ Downloading delta changes
- ☐ Control of TASK state machines
- ☐ Instance data capture
- ☐ Development

Slide 12

REX - Services

- ☐ Connect
- ☐ Disconnect
- ☐ Data

Slide 13

REX - Why use it ?

- ☐ Portable
- ☐ CMS compliant
- ☐ Easy to add "special" network functionality

Slide 14


General I/O

Slide 15

General I/O - What is it ?


A general class for handling I/O without media dependencies


Slide 16



General I/O - Media


- ☐ "Standard I/O"
- ☐ REX
- ☐ LFS


Slide 47 



General I/O - Why use it ?


- ☐ Applications can be written to handle I/O without knowing where it came from it goes to (c.f. < and > in unix/DOS).
- ☐ Portable
- ☐ Extensible


Slide 48 



Platforms


- ☐ XEC/XEM (Eurotherm real-time executive)
- ☐ Windows
- ☐ Unix (Interactive/Ultrix/???)
- ☐ VMS


Slide 49 



Processors


- ☐ Motorola 68k family
- ☐ Intel 80x86 family
- ☐ Mips (RISC)


Slide 50 



What is Happening Now ?


- ☐ Enhancement of Resource Level Wiring
- ☐ Adding enumerations
- ☐ Add RETAINED data
- ☐ Interpreted Resource Level

Slide 51 



What Next ?

- ☐ Reloadable RLOs
- ☐ Delta changes
- ☐ GCT ST debug support over RMP
- ☐ CMS redundancy

Slide 52 



Who will be working on it ?

- ☐ Controls UK
- ☐ Controls US
- ☐ Eurotherm Italy
- ☐ EPA UK
- ☐ Gauging US ?
- ☐ Drives UK/US ?
- ☐ Anyone else ?

Slide 10



Summary

- ☐ A common run-time environment between products
- ☐ A common application framework
- ☐ Portable environment
- ☐ Inter-operability
- ☐ Flexible tool set

Slide 11



The End !



Slide 12

