

CONTROLLED DISTRIBUTION ONLY IF COLOUR STAMPED

**CONTROLLED  
DISTRIBUTION COPY**

Resource Manager

RETAINED Variables

CONTROLLED DISTRIBUTION LIST

Project File



APPROVAL FOR DOCUMENT REVISION	Author <i>MF</i>	Checked <i>MF</i>	ELECTRONICALLY STORED DOCUMENT DIRECTORY PATH	
JOB TITLE	Signature	Date	osprey:/osprey2/shared/resource/docs/specs/HPC326/retain.tex	
Programme Manager	<i>Q</i>	30/1/95	ORIGINATING DEPT: ENGINEERING	
			CONTROLLING DEPT: ENGINEERING	
			<b>CONTROL SHEET</b>	NO. of SHEETS
				6
AUTHOR: M Fox			DOC. TYPE: Software Design Specification	
CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Resource Manager		
		RETAINED Variables		
EUROTHERM CONTROLS <small>© Copyright 1995 Eurotherm Controls Ltd</small>			DOCUMENT NUMBER HP024105C326	SHT. 1

## DOCUMENT REVISION HISTORY

Doc. Revision	Date	Changes
1	January 12, 1995	First formal issue

### 1 Scope

This document specifies the implementation of the IEC 1131-3 **RETAIN** qualifier in data declarations in the ST compiler and the Resource Manager.

### 2 Related Documents


- [1] IEC1131 Programmable Controllers Part 3: Programming Languages First Edition 1993-03
- [2] HP024105C325 Resource Manager - VAR\_GLOBAL and VAR\_EXTERNAL

### 3 Introduction

The current implementation of the ST Compiler provides an implementation of **RETAIN** ( on the switch **-SplitRetain** ). The compiler splits out data specified with the **RETAIN** qualifier from other data and so generates 2 structures at each level of the hierarchy. This method shall continue to be used. This implementation does however have a number of significant defects.

- It is coupled with an option which inhibits generation of templates ( and other things required for the **RESOURCE** database ).
- Is is coupled with an option which requires access to **RETAINED** through access functions. It is not in general necessary to use access functions to access data that may be marked as non-volatile. Use of such access functions ought to be a separate option.
- It does not generate GAD tables even if the **-GI** is supplied.
- Non-volatile structures are generated of the same name as the volatile data except preceded by **nvol\_**. This is "unsafe" as it does not distinguish between this case and a template name beginning **nvol\_** ( perfectly legal ).
- It does not support the qualifier on **FUNCTION\_BLOCKS**.
- It does not support the provision of 'C' data ( using the **source.dht** construct ) for **RETAINED** data.
- No provision is made at **RESOURCE** level for **RETAIN**.
- The **-DisableNon1131Vars** permits the **RETAIN** qualifier to be added to declarations other than **VAR**, **VAR\_OUTPUT** and **VAR\_GLOBAL**.

All of the above need to be rectified as part of the provision of general **RETAIN** support.

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Resource Manager	
		RETAINED Variables	
EUROTHERM CONTROLS © Copyright 1995 Eurotherm Controls Ltd		DOCUMENT NUMBER	SHT.
		HP024105C326	3

### 4.3 instance.c

These are unchanged except that :-

- An additional offset ( MyRetainOffset <sup>1</sup> ) is required for RETAINED data.

```
typedef struct
{
    VtableP ObjectType;
    int MyOffset;
    int MyRetainOffset;
    TypeTemplateC* MyTemplate;
} GADTableEntryC;
```

### 4.4 resource.c

All RLO template types ( RLObjC, RLBObjC, RLGlobObjC and RLTaskObjC ) require an additional member ( RetainInstanceData <sup>2</sup> to define the size of the RETAINED data. eg

```
typedef struct RLObjStruct
{
    VtableP          ObjectType;
    ChildTemplateC*  Template;
    char*            Connections;
    int               InstanceData;
    int               RetainInstanceData;
    struct RLObjStruct* NextRLB;
} RLObjC;
```

### 4.5 Compatibility

All blocks in a RESOURCE must either be compiled with split instance data or without it, it will not be possible to "mix and match".

<sup>1</sup>The offset is currently defined to be an int. It ought to be an unsigned quantity. This imposes a limit of 64k (32k for an int ) on Windows GCT.

<sup>2</sup>The size is currently defined to be an int. It ought to be an unsigned quantity. This imposes a limit of 64k ( 32k for an int ) on Windows GCT.

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Resource Manager

RETAINED Variables

EUROTHERM CONTROLS



DOCUMENT NUMBER  
HP024105C326

SHT.

5

CONTROLLED DISTRIBUTION ONLY IF COLOUR STAMPED

**CONTROLLED  
DISTRIBUTION COPY**

## Resource Manager

### VAR\_GLOBAL and VAR\_EXTERNAL

#### CONTROLLED DISTRIBUTION LIST

Project File

APPROVAL FOR DOCUMENT REVISION	Author <i>M7</i>	Checked <i>M7</i>	ELECTRONICALLY STORED DOCUMENT	
JOB TITLE	Signature	Date	DIRECTORY PATH osprey:/osprey2/shared/resource/docs/specs/HPC325/glob.tex	
Programme Manager	<i>CS</i>	30/1/95	ORIGINATING DEPT: ENGINEERING	
			CONTROLLING DEPT: ENGINEERING	
			<b>CONTROL SHEET</b>	NO. of SHEETS 5
AUTHOR: M Fox			DOC. TYPE: Software Design Specification	
CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION 1	Resource Manager		
		VAR_GLOBAL and VAR_EXTERNAL		
EUROTHERM CONTROLS		DOCUMENT NUMBER HP024105C325		SHT. 1
© Copyright 1995 Eurotherm Controls Ltd				

## DOCUMENT REVISION HISTORY

Doc. Revision	Date	Changes
1	January 12, 1995	First formal issue

## 1 Scope

This document specifies the provision of IEC 1131-3 VAR\_GLOBAL and VAR\_EXTERNAL by the ST compiler and the RESOURCE database.

## 2 Related Documents

- [1] HP024105C309 Resource Manager - Enhanced Resource Level Functionality
- [2] IEC1131 Programmable Controllers Part 3: Programming Languages First Edition 1993-03
- [3] Proposed Technical Corrigendum to IEC1131-3 65B/WG7/TF3(MILANO) (1994-10-14)

## 3 Introduction

VAR\_GLOBALs are to be provided at RESOURCE level. These may then be referenced by VAR\_EXTERNALs placed anywhere within the same RESOURCE.

## 4 Compiler


This section details the output of the ST compiler required to implement VAR\_GLOBAL and VAR\_EXTERNAL in a RESOURCE.

### 4.1 VAR\_GLOBAL

A VAR\_GLOBAL is a new type of Resource Level Object ( RLO ) and introduces the concept of a simple RLO ( up to now the only RLO types have been FUNCTION\_BLOCK, PROGRAM and TASK, all of which are complex types. ).

Each VAR\_GLOBAL results in a new type of RLO of the form :-

```
typedef struct
{
    VtableP      ObjectType;      /* RLGLOBAL */
    ChildTemplateC* Template;
    int          InstanceData;    /* size of instance data required */
    GADTableEntryC* GADTbl;
    uint16       NGADs;
} RLGlobObjC ;
```

CONTROLLED DISTRIBUTION COPY ONLY IF COLOUR STAMPED ON CONTROL SHEET.	DOCUMENT REVISION  1	Resource Manager  VAR_GLOBAL and VAR_EXTERNAL	
		DOCUMENT NUMBER HP024105C325	SHT.  3

For any complex types that require access the lock and free pair of access functions should be used.

```
void VarGlobalLock () ;  
void VarGlobalFree () ;
```

The ST compiler should plant the appropriate calls in all ST that accesses data through VAR\_EXTERNALS.

## 5 Resource Database

Access to data either through VAR\_GLOBALS or VAR\_EXTERNALS requires locking in the same way as the ST compiler. The same functions will be used.

### 5.1 VAR\_GLOBAL

VAR\_GLOBALS only require the concept of simple RLOs to be introduced.

### 5.2 VAR\_EXTERNAL

VAR\_EXTERNALS are accessed in the same may as VAR\_IN\_OUTs, ie by de-referenced pointer. The only difference is that on loading the RESOURCE loader resolves the instance data pointers to VAR\_GLOBALS in the RESOURCE. If any VAR\_EXTERNALS in a TASK cannot be resolved then the TASK will not be allowed to execute.

→ ○ ←

CONTROLLED DISTRIBUTION COPY  
ONLY IF COLOUR STAMPED ON  
CONTROL SHEET.

DOCUMENT  
REVISION

1

Resource Manager

VAR\_GLOBAL and VAR\_EXTERNAL

EUROTHERM CONTROLS



DOCUMENT NUMBER

HP024105C325

SHT.

5

© Copyright 1995 Eurotherm Controls Ltd