

THE RESOURCE MANAGER

A Guide to the Resource Task

User Guide

© COPYRIGHT MCMXCIV EUROTHERM LIMITED

All rights strictly reserved. No part of this document may be stored in a retrieval system, or transmitted, in any form or by any means without prior written permission from Eurotherm Ltd

HA024105C008 1 M Fox



Contents

1	Scope	4
2	Related Documents	4
3	Introduction	4
4	Execution Model	4
4.1	Running	5
4.2	Checksum	5
5	Resource Messaging	5
5.1	CMS	5
5.2	Acting As Proxy	5
5.3	Message Queue	6
5.4	VAR REFERENCES	6
5.5	Outstanding Operation Table	6
5.6	Pending Service table	6
6	Task Control	7
7	The Proxy Task	7
A	Summary of Task Options	7
B	Summary of resctl Options	8
C	Summary of resproxy Options	8

VERSION HISTORY

Version	Date	Changes
1	December 20, 1994	Initial Issue

1 Scope

This document describes the Resource task (Version 2.2).

2 Related Documents

- [1] HA024105C001 A Guide to Var References
- [2] HA024105C002 A Guide to the Resource Debugger
- [3] HA024105C003 A Guide to Tuning the Resource
- [4] HA024105C005 A Guide to Setting Up CMS Networks
- [5] HA024105C007 A Guide to the Resource Loader

3 Introduction

The resource task is the actual program that executes a **RESOURCE TASK**. It is a generic program that attaches to a **RESOURCE** and runs the required **TASK** and its associated **FUNCTION_BLOCKS**.

Once the **RESOURCE** has been loaded by the resource loader [5] each of the **TASKS** specified in the **RESOURCE** may be run with the resource task.

4 Execution Model

The execution model of a **TASK** is modelled by 3 state machines as listed below :-

State Machine	States	Description
Load	Unloaded	TASK is not yet loaded or run
	Loaded	TASK is loaded
Run	Running Suspended	Blocks will execute if TASK is loaded Blocks will not execute if TASK is loaded but the TASK will service incoming RMP requests
	Stopped	Blocks will not execute nor will RMP requests be serviced.
Data	Uninitialised	Data is all zeros.
	Reset	Block cold starts have been executed.
	Modified	Data has been modified from the cold start values by the TASK , ie blocks have executed.

4.1 Running

When a **TASK** is loaded and run it goes through a 3 stage process :-

Cold Start - apply the **FUNCTION_BLOCK** cold starts.

Cold Start Expressions - Not currently implemented

Execution - execute the blocks

The process may be broken up into its individual stages by invoking the task 3 times with the options **-coldstart** **-expcoldstart** and **execute** in turn. The cold start options will cause the task to terminate on completion.

4.2 Checksum

The loader generates a checksum for the whole of the invariant parts of the **RESOURCE** database. If a **TASK** is run with the **-checksum** this **TASK** checks this checksum periodically and if it fails will terminate all the **TASKS** in the **RESOURCE**.

5 Resource Messaging

A **RESOURCE TASK** supports messaging using the Resource Messaging Protocol (**RMP**) across **CMS** as either a client using **VAR REFERENCES** [1] and/or as a server servicing **VAR REFERENCES** from other **TASKS** or other utilities such as the resource debugger [2]. It also a requirement that a task provides the services of the proxy for the **RESOURCE**, this is essentially servicing template requests [1].

5.1 CMS

Each **TASK** is a **CMS** process with an application entity name [3] of the form **<ResourceName:TaskName>**. If a **TASK** is designated to be the proxy (§5.2) for a **RESOURCE** then it will also have an additional application entity name **<ResourceName>:**. Each **CMS** process is required to have a number of **CMS** buffers for messaging. The number of buffers available to each **TASK** may be individually set using the **-i** option. The default is "4:256 4:1024 4:4096".

If a **TASK** has no **VAR REFERENCES** and is not to act as the proxy then it have be run without a **CMS** by specifying the **-nocms** option. If this is the case then it will not be possible to access its data across **RMP** except via the proxy task if loaded (§7) .

5.2 Acting As Proxy

Any one of the **TASKS** in a **RESOURCE** may be designated to be the proxy task. This may have been specified by the loader [5] before the **TASKS** are run. If this has not happened then the default is that the first **TASK** that is ran is the proxy. This may be controlled by any **TASK** using the **-noproxy** option to prevent itself becoming the proxy.

In order to act as a proxy the **TASK** needs some resources for template parsing. These resources are specified as the number of individual data values (an **ARRAY** counts a one value) that are referenced in an individual template request. By default this value is 200, but may be changed by the **-proxy** option. This does of course have no effect if the proxy task has already been allocated.

5.3 Message Queue

All incoming messages to a **TASK** are queued up and then serviced at the end of the **TASK** execution and between individual **Resource Level Block (RLB)** executions. The size of the queue determines the maximum number of messages that may be serviced at these points. The default value for this is 32 but may be altered with the **-r** option.

This figure may already have been set by the loader in which case the **-r** option has no effect.

5.4 VAR REFERENCES

If a **TASK** has **VAR REFERENCES** then it may be desired to alter the timeouts. These are :-

Option	Default	Description
-tot	60000 (T#60s)	Template request timeout (in milliseconds)
-lor	5000 (T#5s)	Read request timeout (in milliseconds)
-tow	5000 (T#5s)	Write request timeout (in milliseconds)

These are very long in us.

5.5 Outstanding Operation Table

If a **TASK** has some **VAR REFERENCES** then it needs an **outstanding operation table (OOT)** [1]. By default this table is set to be large enough to allow outstanding operations on all **VAR REFERENCES**. If it is required to modify this because of :-

- Dynamic var references are to be used as well
- The number of outstanding operations is known to never reach maximum.
- There is a requirement to **"throttle" the number of outstanding requests.**

then the size of the OOT can be modified with the **-oot** option which takes 3 forms :-

-oot <Size> - Set the size of the OOT to <Size>.

-oot +<Size> - Set the size of the OOT to the default + <Size>.

-oot -<Size> - Set the size of the OOT to the default - <Size>.

This figure may already have been set by the loader in which case the **-oot** option has no effect.

5.6 Pending Service table

If a **TASK** has **SERVICES** which may be accessed using **VAR REFERENCES** then it requires a **Pending Service table (PST)**. The size of this table determines how many outstanding service requests are premitted on the **RESOURCE**. By default this size is the number of **SERVICES** on the **RESOURCE**. This value may be altered using the **-pst** option. This option takes 3 forms. These are :-

-pst <Size> - Set the size of the PST to <Size>.

-pst +<Size> - Set the size of the PST to the default + <Size>.

-pst -<Size> - Set the size of the PST to the default - <Size>.

6 Task Control

Control may be exercised over the execution of the **TASK** through its task model. This control may be exercised by the **resctl** tool.

resctl may operate on a single named **TASK** (invoked as **resctl task <Name>**) or else on the whole of the **RESOURCE** (invoked as **resctl resource**).

The following commands may be added to the command line of **resctl** to the following effects :-

Command	Effect
suspend	List out the 3 state machines. Mark the TASK as suspended. If the TASK is already stopped then it is marked as stopped and suspended.
resume	Un-suspend the TASK . If a TASK is stopped and suspended then it becomes stopped.
stop	Mark the TASK as stopped. If the TASK is already suspended then it is marked as stopped and suspended.
start	Un-stop the TASK . If a TASK is stopped and suspended then it becomes suspended.
unload	Request the TASK to unload itself and terminate.
reset	Request the TASK to execute its cold start and wait for a continue command.
continue	Execute the cold start expressions and resume its previous running state.

resctl also accepts an additional qualifier to a command **-confirm** which accepts an optional argument of a number of seconds (default is 10) which causes **resctl** to wait until the command has been accepted by the **TASK** and if no prints out an error message.

7 The Proxy Task

An external task **resproxy** can be used to act as the proxy for a **RESOURCE** to reduce the burden of this functionality on the **TASKS**. There can only be one proxy task acting for any given **RESOURCE**.

If a **TASK** has no **VAR REFERENCES** then it has no need for CMS except to act as a server provide for RMP messages. If a **TASK** is executed with **-cms** then **resproxy** can provide the read and write services (but not **SERVICES**) for the **TASK**. When **resproxy** fulfills this functionality it cannot provide **TASK** coherence and is therefore not generally suitable.

A Summary of Task Options

The task is invoked as either **task** or **loadtask task**. The applicable options are :-

<TaskName> - Specifies the name of the **RESOURCE TASK** that this executable is to act as.

-checksum - Checksum the database. (§4.2)

-coldstart - Execute the block cold starts. (§4.1)

-expcoldstart - Execute the block cold start expressions. (§4.1)

- execute** - Execute the blocks. (§4.1)
- i** <CmsInitData> - Specifies the buffer distribution to be used. (§5.1)
- nocms** - Specifies that the **TASK** is not to use CMS. (§5.1)
- noproxy** - Inhibits the **TASK** from becoming the proxy. (§5.2)
- oot** <Num> - Specifies the size of the OOT. (§5.5)
- proxy** <Size> - Specifies the complexity of template that may be serviced by this **TASK** if it is the proxy. (§5.2).
- pst** <Size> - Specifies the size of the PST. (§5.6)
- quiet** - Inhibits informational messages on start up.
- r** <Size> - Specifies the size of message queue. (§5.3)
- tot** <Milliseconds> - Specifies the **VAR REFERENCE** read template request timeout. (§5.4)
- tor** <Milliseconds> - Specifies the **VAR REFERENCE** read request timeout. (§5.4)
- tow** <Milliseconds> - Specifies the **VAR REFERENCE** write request timeout. (§5.4)
- y** <Key> - The Resource IPC key.

B Summary of resctl Options

The applicable options are :-

- resource** - Operate on the whole **RESOURCE**.
- task** <Name> Operate only on **TASK** <Name>.
- <Command> - Specifies the command to be executed.
- confirm** [<Timeout>] - Confirm command completes.
- quiet** - Inhibits messages.
- y** <Key> - The Resource IPC key.

C Summary of resproxy Options

The applicable options are :-

- act** - Attempt to not only service templates but also service read and write requests.
- checksum** - Checksum the database. (§4.2)
- i** <CmsInitData> - Specifies the buffer distribution to be used.
- proxy** <Size> - Specifies the complexity of template that may be serviced by this **TASK** if it is the proxy. (§5.2).
- quiet** - Inhibits informational messages on start up.
- r** <Size> - Specifies the size of message queue. (§5.3)
- y** <Key> - The Resource IPC key.

