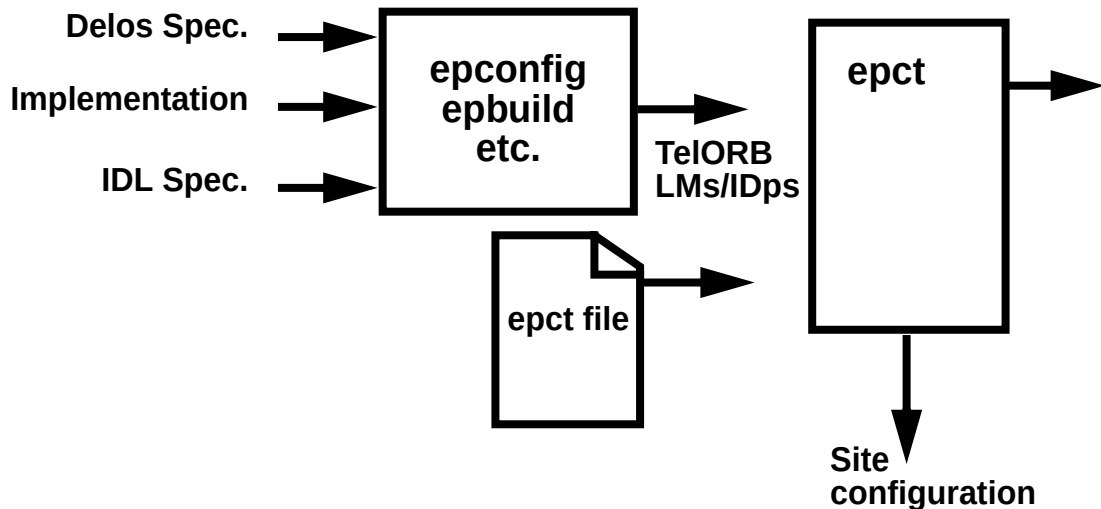

Contents

TelORB Configuration Parameters.....	1
Introduction	1
Minimum Configuration File	2
Site Name and Zone	5
Required and Optional IDPs.....	7
Required IDPs	7
Optional IDPs	9
Initial Data	11
Alphabetic Index	11
File System	13
Database	19
CORBA/IIOP	23
Dumping.....	27
Memory Management	30
Java	31
Loadmodule Size	33
Network Redundancy	35
Resource Control and Load Regulation	37
PXCS (Pilot) Debugger	40
Table Sizes.....	41
Date and Time	42
Other Parameters	46
Environmental Data.....	51
Alphabetic Index	52
AXmp Environment Variables.....	52
Ethernet Switch Environment Variables	55
Network Redundancy Environment Variables.....	56
GEM Cabinet Hardware Supervision.....	61
RMS / HW Alarms Environment Variables.....	64
Other Environment Variables	66
Processors.....	67
Keywords.....	67
Example.....	68
Pool Configuration for the Optional NetRed_IDP	70

TelORB Configuration Parameters

Introduction

A site configuration is built by using the epct command. One or several epct files are input to this command. These epct files describe the hardware and software configuration of the site. The output is a file/directory structure containing all required code, bootloadtables, and so on.



Before using the epct command you must prepare the epct file. This is an ordinary text document, you can see an example skeleton for a minimum configuration file under [Minimum Configuration File](#).

This reference covers the configuration data you can specify in the file, giving explanations, default values, value ranges and comments.

The EPCT files specify the following:

- 1 The name of the site See [Site Name and Zone](#).
- 2 Software used at the site. See [Required and Optional IDPs](#).

- 3 Initial data settings. See [Initial Data](#)
- 4 “Environment Variables”. See [Environmental Data](#)
- 5 Configuration data for each processor. See [Processors](#)

Minimum Configuration File

Below is the skeleton to an example configuration file that will start up TelORB in a minimum configuration with no running application. Click the links to read the appropriate reference information you need for setting the value. You would also need to customize the Ethernet addresses of the TelORB processors, and so on, for your site.

```
// This version completed 20 April 2001.
```

```
//Site name and Zone name – for details see Site Name and Zone
```

```
NAME Example_Initial_epct_file;
```

```
//
```

```
ZONE 1;
```

```
//
```

```
// Software IDPs – for details see Required IDPs
```

```
//
```

```
SOFTWARE
```

```
    IDP/MD14,          // Dicos
```

```
    IDP/DUO2,          // Net_IDP, to show load on processors
```

```
    IDP/MD16,          // DQN_SoftwareSupport_IDPEthernet
```

```
//When using the new Dicos file system:
```

```
    IDP/D16,           // File System Components IDP when using the
new Dicos file system
```

```
//When using Interzone IPC:
```

```
    IDP/MD11,
```

```
//
```

```
// Optional IDPs – for details see Optional IDPs
```

```
//
```

```
//Make sure that the last IDP entry in the list under the SOFTWARE keyword
```

ends with a semicolon (;)

INITIAL DATA

All comments for initial data must start with

Initial data needed for a customer site

#

#For details on a parameter click the link:

[DdbBackupLoadmodule SectionSize](#) =

[DorblIOPHostName](#) =

#

[DUMP MODE](#) =

[DUMP DRIVER](#) =

#

A number of parameters to set up the file system, see:

[FileSystemCommand <n>](#)

ENDDATA

ENVIRONMENT DATA

[ETHSW ADDRESSES](#) =

ENDDATA

//

//Processor Information

//

// See [Processors](#) for information on how to write this section.

//

Site Name and Zone

NAME

The keyword used to specify the name of the site configuration.

Value Range

A sequence of alpha-numeric characters and underlines

Comments

The name could, for instance, contain location, type of application and revision as in the example below:

Example:

NAME Stockholm_GSM_UMTS_HLR_R1A;

ZONE

The keyword used to identify the If the application uses network redundancy.

Explanation

A zone name is included so that network redundancy knows what network the system belongs to.

Default Value

ZONE 1 is default if no keyword is typed.

Value Range

ZONE 1 specifies the master zone.

ZONE 2 specifies the standby zone

Comments

- If there is no network redundancy the keyword ZONE is omitted.

Example:

ZONE 1;

Required and Optional IDPs

All required software must be listed in terms of IDPs under the SOFTWARE keyword in the .epct file. That includes both TelORB IDPs and application IDPs.

Required IDPs

IDP/MD14	The Dicos IDP is used in all systems.
IDP/DUO2	The Net IDP is used for extracting processor load information from the system.
IDP/MD16	The DQN_SoftwareSupport_IDPEthernet IDP is always used in a real system.
IDP/MD17	When using a simulated environment (Vega), IDP/MD16 is replaced by IDP/MD17 (UDP) Note that IDP/MD17 must never be used in a real system.
IDP/D11	Include IDP/D11 if you specify one or more processors of type Vega under the PROCESSOR keyword (See Processors). This is the management interface for the processor type. Note that IDP/D11 must never be used in a real system.
IDP/D13	Include IDP/D13 if you specify one or more processors of type IntelPc under the PROCESSOR keyword (See Processors). This is the management interface for the processor type.
IDP/D16	Include the IDP/D16 File System Components IDP when using the new Dicos file system in a real system.
IDP/D15	Include the IDP/D15 VegaFileSystem IDP when using the new Dicos file system with a Vega simulator. Note that IDP/D15 must never be used in a real system.
Comments	
Make sure that the last IDP entry in the list under the SOFTWARE keyword ends with a semicolon (;).	

IDP/MD11

Include the IDP/MD11-CommLinkUdpAdaptation IDP when using Interzone IPC. It enables UDP-based IPC communication,

Example

SOFTWARE

IDP/MD14,
IDP/DU02,
IDP/MD16,
IDP/D16,
IDP/D15,
IDP/D13;

Optional IDPs

IDP/MD18

If your application is implemented in Java you must include the Java_IDP, which contains the TelORB specific java classes.

IDP/D2

Include the NetRed_IDP to provide network redundancy functionality

IDP/D3

Include the KnuerrRms_IDP when using the Hardware Alarms Collector (also known as the Rack Monitoring System).

Note that a TelORB system should be configured with either:

- GemScbSupervision_IDP (IDP/D22)

or, in the case of an older cabinet:

- KnuerrRms_IDP (IDP/D3)

but never with both at the same time.

IDP/D22

Include GemScbSupervision_IDP to provide supervision for GEM cabinet hardware.

Note that a TelORB system should be configured with either:

- GemScbSupervision_IDP (IDP/D22)

or, in the case of an older cabinet:

- KnuerrRms_IDP (IDP/D3)

but never with both at the same time.

IDP/D8

Include the AXmp_IDP when using the AXmp support processor

Comments

- Make sure that the last IDP entry in the list under the SOFTWARE keyword ends with a semicolon (;).

Example

```
SOFTWARE
    IDP/MD14,
    IDP/DU02,
    IDP/MD16,
    IDP/D13,
    IDP/MD18;
```


Initial Data

In the INITIAL DATA section of your .epct file you can set a number of configuration parameters that are used by the lower levels of TelORB software when their default values are not suitable. Unlike environmental data (See [Environmental Data](#)) Initial Data parameters cannot be changed at run-time by operators.

Initial data parameters can be divided into a number of areas:

- ◆ [File System](#)
- ◆ [Database](#)
- ◆ [CORBA/IIOP](#)
- ◆ [Dumping](#)
- ◆ [Memory Management](#)
- ◆ [Java](#)
- ◆ [Loadmodule Size](#)
- ◆ [Network Redundancy](#)
- ◆ [Resource Control and Load Regulation](#)
- ◆ [PXCS \(Pilot\) Debugger](#)
- ◆ [Table Sizes](#)
- ◆ [Date and Time](#)
- ◆ [Other Parameters](#)

Alphabetic Index

† = deprecated as of TelORB R14)

BOOT buffer size	CosNamingIor ResponderPort
CosNaming Verbose	DUMP DRIVER
DUMP HOST ID	DUMP HOST ID2
DUMP HOST PORT	DUMP MODE
DdbBackupLoadmodule SectionSize	DdbDbuMinutes
DdbExtSynch NoOfProcesses	DdbLazyPurgeInterval
DdbMasterMemoryShare	DdbMasterRecordTab Share †

<u>DdbMdpTabSize</u>	<u>DdbRecordTab MasterSize †</u>
<u>DdbRecordTabLazySize †</u>	<u>DdbnrProcBacklogBytes</u>
<u>DorblIOPBootstrapPrt</u>	<u>DorblIOPGwSlavePort</u>
<u>DorblIOPHostName</u>	<u>DorblIOPPort</u>
<u>DorbSsIOPGwSlavePort</u>	<u>DorbSsIOPPort</u>
<u>FileSystemCommand <n></u>	<u>HEAP_DEBUG</u>
<u>IIOp_GatewayVerbose</u>	<u>IpcPortTableSize</u>
<u>JAVA_CAPSULE_HEAP</u>	<u>JAVA_HEAP</u>
<u>JavaDebug</u>	<u>JavaPXCSDebug</u>
<u>JavaPXCSsuspend</u>	<u>JavaPXCStrace</u>
<u>JavaRunHProf</u>	<u>KernelUtilityDaemonPort</u>
<u>LMd_buffer_size</u>	<u>LMR_buffer_size</u>
<u>LoadRegulation BasicInterval</u>	<u>LoadRegulationMax</u>
<u>MonitorTableSize</u>	<u>NTP_SERVER_IP</u>
<u>NTP_SERVER_PORT</u>	<u>NumericAddress <n></u>
<u>PXCServicesPort</u>	<u>ProcessResourceLimit</u>
<u>MonitorTableSize</u>	<u>SYSLOG_LOG_LEVEL</u>
<u>SYSLOG_PORTBASE</u>	<u>SemaphoreTableSize</u>
<u>SignallerTableSize</u>	<u>StopInMonitor</u>
<u>SymbolicAddress <n></u>	<u>TIME_DST_DATES</u>
<u>TIME_TZ_OFFSET</u>	<u>TimeoutTableSize</u>

File System

Configuring the TelORB R14 file system is done by setting up a sequence of commands to prepare and mount the file structures.

You set up the commands as string values to

FileSystemCommand_<n> parameters in the initial data section of the epct-file. The commands required will differ slightly depending on file system type, DAS (Direct Access file System) or NFS (Network File System). DAS file systems are for use on Vega simulators.

Setting up a TelORB R14 file system as described here requires that the appropriate file system IDP has been specified under the SOFTWARE keyword in the .epct file. (See [Required and Optional IDPs](#).)

Note that, with this file system, the DUCK-server and FileServerInterpreter are no longer needed.

FileSystemCommand_<n> Specify TelORB file system commands

Explanation

To configure a TelORB file system you specify a series of commands to create the directories and the special block device files required to mount the file system.

Value Range

File system commands are entered as *string values* to **FileSystemCommand_<n>** parameters.

n is an ordinal number (0, 1, 2, and so on) to identify the separate commands and order them in a sequence.

Comments

- Each **FileSystemCommand_<n>** parameter and value string must be typed unbroken by any carriage returns.
 - We recommend, for the sake of readability, that you limit each **FileSystemCommand** value string to a single command. If you should wish to concatenate command statements in a parameter string they must be separated by semicolons (;).
-

Setting up the File System

Here is a suggested flow for entering the commands:

- 1 Make the required directories and special block device file for each of the required loadingGroups using the [mkdir](#) and [mknod](#) commands. You can remove directories with the [rmdir](#) command
- 2 Mount each of the loadingGroups with the **mount** command. The mount command differs slightly depending on whether you are creating a [DAS](#) or [NFS](#) file system
- 3 Make the required directories and special block device file for upgradePackages.
- 4 Mount the upgradePackages.

The mkdir Command

You use the mkdir command on the root file system to create the directory structure required for the block devices, loadingGroups and upgradePackages of the file system.

```
mkdir -m <mode> <directory name>
```

-m, <mode>	Set the -m flag to assign a mode to the created directory. <mode> is a unix mode such as rwxrwxrwx OR r-xr-xr-x.
<directory name>	The name of the created directory

Examples :

```
mkdir /testMount
mkdir -m rwxrwxrwx /testMount
```

The mknod command

Use this command to make a block device

```
mknod -m <mode> -b <device> major minor
```

-m, <mode>	Set the -m flag to assign a mode to the created device. <mode> is a unix mode such as rwxrwxrwx OR r-xr-xr-x.
-------------------------	--

```
mknod -m <mode> -b <device> major minor
```

```
-b <device>    Specifies the device name.
```

```
major          The major device number (2 for NFS/RPC, 3 for DAS/UNIX)
```

```
minor          The minor device number. Unique for each block device.
```

The rmdir command

Use this command to remove directories

```
rmdir <directory name>
```

```
<directory name>    The name of the directory to be removed
```

Example :

```
rmdir /testMount
```

The mount command for DAS file systems

This command mounts "Direct Access" mount points:

```
mount -t das <device> <directory> <remote dir>
```

```
<device>          The block device to be mounted on.
```

```
<directory>       The directory to be mounted on.
```

```
<remote dir>      The directory to be mounted.
```

Example:

```
mount -t das /dev/unix4 /testMount/ /tmp/testMount
```

The mount command for NFS file systems

This command mounts NFS mount points

```
mount -t nfs [-o options] <device> <directory> <remote directory>
```

The TelORB file system NFS mount command has the following options and parameters:

[-o option]	Meaning	Possible Values	Default Value
proto	File transport protocol, User Datagram Protocol OR Transmission Control Protocol	udp tcp	udp
rsiz	Read buffer size, in bytes	4096 8192	8192
wsiz	Write buffer size, in bytes	4096 8192	8192
nmbofreadbuf	Maximum number of concurrent read requests (the number of buffers to be sent on read).		2
nmbofwritebuf	Maximum number of concurrent write requests (the number of buffers to be sent on write).		6

[-o option]	Mandatory options	Default Value
uid	Userid for TelORB user on the IO-machine	none
gid	Group id for the TelORB user on the IO-machine. If TelORB is a member of more than one group, you should specify a gid value for each group	none
hostname	Hostname for the IO - machine acting as remote file server	none
ipaddr	The ip address of an IO - machine acting as NFS file server. You should specify an ipaddr values for each internal TelORB network (there are often two).	none

Parameters

<device>	The block device to be mounted on.
<directory>	The directory to be mounted on.
<remote dir>	The directory to be mounted.

Examples:

```
mount -t nfs -o uid=101 -o gid=501 -o ipaddr=134.138.184.176 /dev/rpc2  
/testMount/ /tmp/testMount
```

```
mount -t nfs -o proto=udp -o rsize=8192 -o wsize=8192  
-o nmbofreadbuf=2 -o nmbofwritebuf=6 -o uid=101 -o gid=501  
-o ipaddr=134.138.184.176 /dev/rpc2 /loadingGroup01_2/  
/tmp/loadingGroup01_2_2
```

Examples

File system commands for configuring a DAS (Direct Access file System) for a system running on Vega with the loading groups placed under /tmp locally:

INITIAL DATA

...

```
FileSystemCommand_1 = "mkdir /dev;  
FileSystemCommand_2 = "mkdir /loadingGroup01;"  
FileSystemCommand_3 = "mknod -b /dev/unix1 3 1;"  
FileSystemCommand_4 = "mount -t das /dev/unix1 /loadingGroup01  
/tmp/axe/loadingGroup01;"  
FileSystemCommand_2 = "mkdir /upgradePackages;"  
FileSystemCommand_3 = "mknod -b /dev/unix2 3 2;"  
FileSystemCommand_4 = "mount -t das /dev/unix2 /upgradePackages  
/tmp/axe/upgradePackages;"
```

...

ENDDATA

File system commands for configuring an NFS (Network File System) for a system running on real hardware. The loading groups are exported and placed under /opt/telorb/axe/ on the IO processors.

The userid and groupids are set to those of telorb on the IO machines. Note also that two IP addresses have been given for each mount command - the two IP addresses of the relevant IO machine.

INITIAL DATA

...

FileSystemCommand_1 = "mkdir /dev"

FileSystemCommand_2 = "mkdir /loadingGroup01_1"

FileSystemCommand_3 = "mknod -b /dev/rpc1 2 1"

FileSystemCommand_4 = "mount -t nfs -o uid=101 -o gid=501
-o ipaddr=134.138.63.25
-o ipaddr=134.138.64.25 /dev/rpc1 /loadingGroup01_1/
/opt/telorb/axe/loadingGroup01_1"

FileSystemCommand_5 = "mkdir /loadingGroup01_2"

FileSystemCommand_6 = "mknod -b /dev/rpc2 2 2"

FileSystemCommand_7 = "mount -t nfs -o uid=101 -o gid=501
-o ipaddr=134.138.63.26 -o ipaddr=134.138.64.26 /dev/rpc2
/loadingGroup01_2/ /opt/telorb/axe/loadingGroup01_2"

FileSystemCommand_8 = "mkdir /upgradePackages"

FileSystemCommand_9 = "mknod -b /dev/rpc3 2 3"

FileSystemCommand_10 = "mount -t nfs -o uid=101 -o gid=501
-o ipaddr=134.138.63.25 -o ipaddr=134.138.64.25 /dev/rpc3
/upgradePackages/ /opt/telorb/axe/upgradePackages"

...

ENDDATA

Database

DdbLazyPurgeInterval

The period in milliseconds for the lazy-purger.

Explanation

If a database object is accessed on a processor to which the object is not replicated, a cached copy is installed. These cached copies are called “lazy copies” for historical reasons. The lazy-purger is a garbage collector of such cached copies.

If the period for the lazy copy purger is too high then the processor may, in a worst-case scenario, run-out of memory.

Before running out of memory, the processor would have problems with repeatedly having to expand the table holding the lazy-copies. If this happens you will see a lot of warning messages on the processor’s console.

Default Value

= 60000

Value Range

’0’, or the number of milliseconds for the period

Comments

- A value of ’0’ will cause the lazy-purger to be idle (that is, never run).
 - Avoid values below 15,000 (purge intervals shorter than 15 seconds)
 - The default value of 60,000 (once a minute) works well for most systems. The exception would be a system that generates a lot of lazy copies, for example during a bulk loading phase.
-

DdbMdpTabSize	The number of Master Data Partition (MDP) objects (DUs with POTs) that can be in the table, rounded upwards with 25% Explanation Each key-introducing POT will have a number of MDPs equal in number to the DU-limit for the DUT of that POT. If more than one key-introducing POT is associated with a DUT, then each has to be counted. The DdbMdpTabSize should be set to be 25% larger than the sum of all MDPs. Dicos itself has approximately 150 MDPs. For any given system, you should take into account the applications that will run on that system, or be likely to be installed on the system. The table can expand dynamically but this can cause performance problems and memory fragmentation. Default Value = 20000 Value Range 500 - 50000. Comments • The default value of 20,000 was designed for the Jambala HLR application: a system designed to scale up to be very large, thus having large distribution limits. • The largest allowed distribution limit is currently 1024
DdbRecordTabMasterSize †	In earlier releases of TelORB, the maximum number of objects per processor (including replication). Deprecated as of TelORB R14. In R14, you no longer have to do dimensional calculations of how many database object that are expected to reside on each processor. Any setting of this parameter in an existing system will have no effect on the system after upgrade to R14.

DdbRecordTabLazySize †	In earlier releases of TelORB, the maximum number of lazy copies expected per processor. Deprecated as of TelORB R14. Any setting of this parameter in an existing system will have no effect on the system after upgrade to R14.
-------------------------------	--

DdbBackupLoadmodule SectionSize	The maximum load module size (in bytes and excluding headers) for data backup.
--	--

Explanation

A DB backup load module consists of three parts:

- The LoadModuleHeader
- The DBbackupHeader
- A data sector for the contained data.

DdbBackupLoadmoduleSectionSize sets the maximum allowed size for the data sector within a DB backup load module before compression.

Default Value

= 0xF4240

Value Range

- The default, 1 MByte (0xF4240), is intended for simulated environments (Vega).
- For target systems we recommend 4 MByte (0x3D0900).

Comments

- By increasing the maximum allowed section size you can limit the number of created DB backup files/load modules per MDP (and Distribution Unit).
-

DdbMasterMemoryShare	Provides resource control of memory usage. Explanation The parameter specifies how many percent of the total physical memory on each processor that may be used by the database to store persistent data. If this limit should be reached, transactions that attempt to create objects will be aborted. Default Value = 80 Value Range 0 - 100 Comments • Setting the value to 100 disables the database memory resource control.
DdbMasterRecordTab Share †	In earlier releases of TelORB, this parameter provided resource control of the MasterRecord Table. Deprecated as of TelORB R14. Any setting of this parameter in an existing system will have no effect on the system after upgrade to R14.
DdbDbuMinutes	Specifies how long a DBU migration is allowed to take before being aborted and rolled back. Default Value = 120 Value Range The default value should be sufficient for all but exceptionally large DBU migration cases.

CORBA/IIOP

DorbIIOPBootstrapPrt

The port number for bootstrapping the standard CORBA Naming Service

Explanation

This parameter sets the TCP/IP port number through which the standard CORBA Naming Service implementation can be bootstrapped using the new standard CORBA bootstrapping mechanism.

Value Range

Integer <TCP/IP port number>

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.

DorbIIOPHostName

The IP addresses of IIOP gateway agents for the system.

Explanation

DorbIIOPHostName identifies the IP addresses to which external CORBA clients should address initial CORBA-IIOP requests.

Value Range

We recommend you set this to:

"<IP address of TelORB Proc 1 on Net A> : <IP address of TelORB Proc 2 on Net B>"

Comments

- Use different processors for each net.
 - The processors must belong to the DicosPlgLoaderPool.
 - The IP addresses must be local to the site.
-

DorbIIOPPort

The TCP/IP port number for the IIOP gateway agent for TelORB.

Default Value

= 30330

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

DorbSsIIOPPort

The TCP/IP port number that should be used for CORBA-SSL communication to the IIOP gateway agents on processors 1 and 2.

Default Value

= 30330

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

DorbIIOPGwSlavePort

The TCP/IP port number for the IIOP gateway slaves for TelORB.

Default Value

= 30530

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

DorbSsIIOPGwSlavePort

The TCP/IP port number that should be used for CORBA-IIOP communication to the IIOP gateway slaves on all processors.

Default Value

= 30730

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

IIOP_GatewayVerbose

Enables verbose printouts from the IIOP gateways. Do not set this parameter in a production system, it generates a lot of printouts.

Default Value

= 0

Value Range

0 = verbose printout off
1 = verbose printout enabled

Comments

- Can be set to 1 when debugging IIOP communication.
-

**CosNamingIor
ResponderPort**

The TCP/IP port number at which the old proprietary bootstrapping daemon for the Corba Naming Service runs.

Explanation

When a TCP/IP connection is established to this port, TelORB writes the CORBA IOR of the root Naming Context onto the connection.

Default Value

= 7111

Value Range

Integer <TCP/IP Port number>

Comments

- Does not normally need to be set, unless several users are sharing a workstation to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

CosNaming_Verbose

Enables debug output from the Corba Naming Service

Explanation

Set to 1 to get output from the Corba Naming Service.

Default Value

= 0

Value Range

0 = no output 1 = output enabled

Comments

- Only set this parameter when debugging applications that use the Corba Naming Service
-

Dumping

TelORB can generate several types of dumps when error situations occur. The parameters below control switching on and off these dump functions and where the dumps shall be sent.

DUMP_MODE

Switches dumping on or off

Explanation

The parameter value corresponds to a three-bit binary code, where:

- bit 0 - Console Dump
- bit 1 - Remote Capsule Dump
- bit 2 - Remote Processor Dump

Set a bit to “0” to switch dumping off.
Set a bit to “1” to switch dumping on.

Default Value

= 1

Value Range

0-7

Comments

- By default, Console Dumps are switched on.
- To set all three dump types on, set
DUMP_MODE = 7

DUMP_HOST_ID

The IP-address for the dump tool daemon.

Explanation

The IP-address of the host where the dump tool daemon ‘dumpd’ is running.
Preferably the IO computer of the system

Default Value

= 0 (Null)

Value Range

Integer <IP-address>

Please note that the syntax for setting this parameter is unusual: you must use commas instead of periods (dots) as separators.

Example

For a dump tool demon on a host with the IP address 134.138.186.21 type:

DUMP_HOST_ID = (134,138,186,21)

Comments

- This parameter is normally only used on Vega simulators, real systems get the value via DHCP.
-

DUMP_HOST_ID2

The IP-address for the dump tool daemon on the second IO processor.

Explanation

The IP-address of the host where the dump tool daemon 'dumpd' is running.

Preferably the IO computer of the system

Default Value

= 0 (Null)

Value Range

Integer <IP-address>

Please note that the syntax for setting this parameter is unusual: you must use commas instead of periods (dots) as separators.

Example

For a dump tool demon on a host (second IO processor) with the IP address 134.138.186.21

type:

DUMP_HOST_ID2 = (134,138,186,21)

Comments

- This parameter is normally only used on Vega simulators, real systems get the value via DHCP.
-

DUMP_HOST_PORT

The UDP-port that ‘dumpd’ listens on.

Default Value

= 5556

Comment

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

DUMP_DRIVER

The name of the driver for remote dumps.

Default Value

No default value

Comment

- The only currently available driver is the TFTP-dump driver.
-

Memory Management

HEAP_DEBUG

Enables console printout of extended information from the TelORB memory management when a process crashes due to violation of memory management rules. Only to be used in test systems because of the memory consumption.

Explanation

Setting this parameter can be useful when a process is writing into the next segment of the memory area. The console printout will include a part telling from where in the code the nearest memory segment was allocated. The address can be translated into a certain line in the source code (with the [epelfaddr](#) command) and you will then almost certainly be able to pin-point the code responsible for the overwriting.

Deletion of an object two times is another example of when you would get extra information.

Default Value

= 0

Value Range

0 or 1

Comments

- This parameter shall only be set in test systems because of the high memory consumption.
 - The extra information is not shown in the error dump, only in the console printout.
-

Java

JavaRunHProf

Configures the Java Profiler available in TelORB

Explanation

A Java profiler is available for simulated environment processors (Vega, on Sparc). This parameter is used to set the initial options for the profiler.

Default Value

= (none)

Value Range

"-Xrunhprof | :<option=value>, ..."

For more information, see [TelORB Java Profiler](#)

JAVA_HEAP

Sets the size available for Java-code and Java-data.

Explanation

The size must be big enough to hold all your un-packed application classes, and all data objects.

Default Value

= 4

Value Range

maximum 240 MB

Comments

- With the present HotSpot JVM, the JAVA_HEAP is not a part of the JAVA_CAPSULE_HEAP
-

JAVA_CAPSULE_HEAP Sets the size of the Java Capsule Heap

Explanation

The size must be big enough to hold the JVM-code and internal data, and all JavaJar-LMs that you load initially (The JavaJar-LMs are temporary stored on the Capsule-heap until the JVM is completely started).

Default Value

= javaHeap+12

Value Range

The C/C++/Capsule Heap may be extended to 223 MB.

Comments

- Always set a value for this parameter.
 - With the present HotSpot JVM, the JAVA_HEAP is not a part of the JAVA_CAPSULE_HEAP
-

JAVA_PERM_HEAP Sets the size of the Java Permanent Heap.

Default Value

= 4

Value Range

Recommended value 16 MB

JavaDebug Configures the TelORB JVM to allow the use of any JDWP (Java Debugger Wire Protocol) debugger.

Explanation

Specifies a TCP/IP port number that is used to connect a JDWP (Java Debugger Wire Protocol) debugger and also tells the JVM to support debugging.

Default Value

= 8989

Loadmodule Size

LMD_buffer_size

Buffer size for the LMDistributor process

Explanation

LMReciever and LMDistributor are processes having memory areas reserved on each processor as [LMR_buffer_size](#) and LMD_buffer_size. These buffers should be big enough to contain the largest filesize of any of the LMs used in the system.

Default Value

= 3*1024*1024

Value Range

Recommended value for the HotSpot JVM: 8 MB
(At present the largest known filesize is 6MB).

LMR_buffer_size

Buffer size for the LMReciever process

Explanation

LMReciever and LMDistributor are processes having memory areas reserved on each processor as LMR_buffer_size and [LMD_buffer_size](#). These buffers should be big enough to contain the largest filesize of any of the LMs used in the system.

Default Value

= 3*1024*1024

Value Range

Recommended value for the HotSpot JVM: 8 MB
(At present the largest known filesize is 6MB)

BOOT_buffer_size

Buffer size for BOOT

Explanation

During loading, memory areas are reserved in the kernel on each processor for BOOT as BOOT_buffer_size. This buffer has to be as big as the both the LMReciever and LMDistributor process buffers together
See [LMR buffer size](#), [LMD buffer size](#)

Default Value

= 5000000

Value RangeRecommended value for the HotSpot JVM: 16Mb

Network Redundancy

DdbExtSynch NoOfProcesses

The maximum number of DdbDynamicDbSynchC processes on each processor that is engaged in synchronizing the database from an external backup.

Default Value

= 5

Value Range

The default value is recommended for most installations. Each process reads one backup file at a time. If the DdbExtSynchNoOfProcesses is set too low, the external synch takes longer time than necessary.

Comments

- With the new Dicos file system, calculating the number of required file descriptors as described for earlier releases of TelORB is no longer relevant.
-

DdbnrProcBacklogBytes

The maximum number of bytes used for backlog in mirroring processes of each processor.

Explanation

The DdbnrProcBacklogBytes parameter specifies the maximum number of bytes used for backlog in all mirroring processes of each processor. When the limit is reached, the mirroring processes will start discarding the oldest transactions.

This limit should be set high enough to contain all mirrored transactions of a reconciliation cycle, with a margin, because the standby cannot add any transactions after zone-reload until the first external sync is ready. If it is set too low, a reconciliation will miss out on some transactions and a new reconciliation will be needed. (That reconciliation will, however, probably need less backlog, as most of the new transactions can be accepted after the first synchronization.)

If it is set too high, a big backlog (for example if resync fails) could consume all the memory of the processor and result in a processor reload. Note also that memory once used for the backlog may only be used for processes (application heaps, Java) but not for database data or load modules. Thus, if the limit is too high, a large backlog may prohibit database growth or smooth system upgrade until the next processor reload.

Example

If the NetRed mirrored transaction load is 100kB/s evenly distributed on 4 processors and the time to do backup, transfer and external synch is 1800 sec (30 minutes), the limit should be:

$$100000/4*1800 = 45000000 + \text{a margin}$$

Let's say 60000000.

Default Value

= 90000000 (90 MB)

Comments

- The size of the backlog (if non-zero) is displayed on the console

Resource Control and Load Regulation

ProcessResourceLimit

Sets a limit to the number of system resources available to an individual process instance.

Explanation

In order to further increase the system's robustness to software faults there is a limit to number of system resources available to an individual process instance. System resources are resources in the operating system, such as timers, ports (on which dialogues are built), transactions, sockets, and so on.

The purpose of the limit is to detect and abort processes that erroneously allocate too many resources before the total number of resources in the system is exhausted. The default limit has been chosen so that most processes should not need to care at all. Processes that do need to allocate many resources (hundreds or thousands) as part of their function may change the limit through TelORB APIs.

If a process reaches the limit, it is aborted and a dump is generated. For C++ processes this is done immediately, while for Java processes, a `java.lang.SecurityException` (that should not be caught) is thrown.

Default Value

= 400

Value Range

Note that no exact calculation need to be made, since the limit is only there to detect faults. If you think the process uses about 100 resources, set the limit to 1000. A leak or an uncontrolled loop allocating resources will be detected anyway.

**LoadRegulation
BasicInterval**

The period of time load is sampled and regulation parameters updated

Explanation

LoadRegulationBasicInterval specifies (in milliseconds) the length of the period of time load is sampled and regulation parameters updated.

If you have few calls per second per processor (often the case with Java) you may want to change the value from the default. With few calls per second and too long a sampling time the variations between samples may become so large that load regulation sometimes misunderstands the situation and rejects calls.

Default Value

= 1000

Value Range

1000 to 6000 milliseconds

LoadRegulationMax

The initial value for the Load Regulation level.

Explanation

The initial value for the Load Regulation level.

Load Regulation is typically called by an application on entry, to determine whether there's sufficient capacity available to perform a task. If the cpu load rises above the specified load regulation level, applications which use Load Regulation will start to reject calls. The Load Regulation should never be set above the 75% default, but you can set it lower to provide lower latency times, or spare capacity for tasks such as backups or upgrades.

Default Value

= 750 (75%)

Comments

- This variable probably does not need to be set since there is an operator interface to change it.
-

PXCS (Pilot) Debugger

Parameters related to Java debugging with the Pilot debugger.

JavaPXCSDebug	The port number to connect to when debugging Java with the Pilot debugger Explanation JavaPXCSDebug should be set to the TCP port number that the Pilot debugger should connect to when debugging Java. When set, the JVM is started with debugging enabled, which makes it run much slower. Default Value = Null Value Range Integer <TCP/IP port number> Comments • Only one port number per TelORB processor can be in use.
JavaPXCStrace	Set this parameter to 1, if you want to make a trace for debugging.
JavaPXCsuspend	Set this parameter to 1 if you want to suspend all threads at startup.

PXCServicesPort	The port number for the PXCServices port on the TelORB processor. Explanation This value of this parameter is entered in Connect dialog on the debugger. Default Value = 8888 Comments • This value of this parameter is entered by the user Connect dialog on the debugger.
------------------------	---

Table Sizes

IpcPortTableSize	The maximum number of IPC ports on each processor. Default Value = 32767
-------------------------	---

MonitorTableSize	The maximum number of simultaneous monitors on a processor.
-------------------------	--

Value Range
Recommended value 16383

Default Value
= 2047

Comments

- Monitors are mainly used by the JVM.

ProcessTableSize	The maximum number of processes on a processor Default Value = 8191
-------------------------	--

SemaphoreTableSize	The maximum number of semaphores per processor. Default Value = 2047
SignallerTableSize	The maximum number of signallers per processor. Explanation Signallers are used for instance by drivers to wake-up a process. Default Value = 4095
TimeoutTableSize	The maximum number of DicosTimers and DicosPeriodicTimers (sum total of both types) per processor. Default Value 32767

Date and Time

NTP_SERVER_IP	The IP-address of the Network Time Protocol (NTP) server. Explanation The system will try to synchronize its time with this server on initial start. Default Value = 0 (Null)
----------------------	---

Value Range

Integer <IP-address>

Please note that the syntax for setting this parameter is unusual: you must use commas instead of periods (dots) as separators.

Example

For an NTP server with the IP address 134.138.186.21 type:

NTP_SERVER_IP = (134,138,186,21)

Comments

- This parameter is normally only used on Vega simulators, real systems get the value via DHCP.

NTP_SERVER_PORT

The port for Network Time Protocol (NTP) requests.

Default Value

= 123

Comments

- The default is 123, and it should only be altered for special tests.
-

TIME_DST_DATES

This parameter is used to pre-load the Daylight Savings Time (DST) data-base at initial start.

Explanation

You may wish to set this parameter when initial start is used often, that is, for Vega simulators and in a test lab environment.

Value Range

This array consists of date and offset pairs:

YYYYMMDDHH,offset

- The date is the non-DST adjusted local (not UTC) time when the local time is altered.
- The offset is in hours.

Default Value

none

Example

On the northern hemisphere you might set the offset to 1 in March and restore it to 0 in October:

TIME_DST_DATES = (2000032602,1, 2000102902,0)

TIME_TZ_OFFSET

Use this parameter to set the local time offset

Explanation

Set this parameter to set the local time offset in relation to UTC time.

- Because this parameter requires a positive integer value, the value that you type must be adjusted according to the following conditional expression:
value = (offset > 12) ? offset - 24 : offset;

Value Range

Positive integer value

Default Value

= 0 (Greenwich or UTC time)

Comments

- This parameter is normally only used on Vega simulators, real systems get the value via DHCP.
-

Other Parameters

SymbolicAddress_<n>

Used to configure, in the TelORB DNS system, both the IP addresses of a symbolic hostnames and the IP address of remote DNS servers.

Value Range - Parameter Name

The parameter name must be of the form NumericAddress_<n> with <n> replaced with an ordinal number (0, 1, 2, and so on). For each value specified for <n>, there must be a corresponding [NumericAddress_<n>](#) specification.

Value Range - Parameter Value

The parameter value must be a quoted string representing the name(s) of a host for which IP addresses will be specified. If two or more hostnames shall resolve to the same IP addresses, they may be specified in a colon (:) delimited string. As a special case, the host name DNS_SERVERS represents the host(s) which the DNS resolver code in TelORB shall contact to resolve host names that are not specifically specified with a SymbolicAddress_<n> parameter.

Examples

SymbolicAddress_1="telorb1"

SymbolicAddress_2="telorb2a:telorb2b"

SymbolicAddress_3="DNS_SERVERS"

Comments

- Values configured with SymbolicAddress_<n> may be changed after the system has been initialized through the **Config** tab of the TelorbManager application.
-

NumericAddress_<n>

Used to configure the TelORB DNS to contain a symbolic hostname and its related IP address

Value Range - Parameter Name

The parameter name must be of the form NumericAddress_<n> with <n> replaced with an ordinal number (0, 1, 2, and so on).

For each value specified for <n>, there must be a corresponding [SymbolicAddress<n>](#) specification

Value Range - Parameter Name

The parameter value must be a quoted string containing the IP addresses of the hostname which was defined in a corresponding [SymbolicAddress <n>](#) configuration value. If a hostname shall resolve to two or more IP addresses, the IP addresses may be specified in a colon (:) delimited string.

Examples

```
NumericAddress_1="138.10.10.1"           // telorb1
```

```
NumericAddress_2="138.10.10.2:138.10.10.3"  
// telorb2a:telorb2b
```

```
NumericAddress_3="138.85.1.10:130.100.128.25"  
// DNS_SERVERS
```

Comments:

- Values configured with NumericAddress_<n> may be changed after the system has been initialized through the **Config** tab of the TelorbManager application
-

KernelUtilityDaemonPort IP port number for the KernelUtilityDaemon (Q-Util).

Explanation

Each processor will try to publish the given value for “KernelUtilityDaemonPort”. In addition, the port number “KernelUtilityDaemonPort” + “processor number” will be published.

This means:

- In a “real” target system “KernelUtilityDaemonPort” can be used when connecting to any of the processors.
- In a “Vega” environment where all processors are running on the same host it’s safest to use “KernelUtilityDaemonPort” + “processor number”.

Default Value

= 8100

Comments

- Does not normally need to be set, unless several users are sharing a host machine to run their Vega simulators, or if using a firewall makes the default port number unacceptable.
-

SYSLOG_LOG_LEVEL

Sets the printout level for DicosSyslog

Explanation

You can set this parameter to set the level of detail in the syslog printout.

Default Value

= 5 (normal but significant conditions are shown)

Value Range

Integer from 0 to 7, signifying the following log levels:

- 0 LOG_EMERG, system is unusable
 - 1 LOG_ALERT, action must be taken immediately
 - 2 LOG_CRIT, critical conditions
 - 3 LOG_ERR, error conditions
 - 4 LOG_WARNING, warning conditions
 - 5 LOG_NOTICE, normal but significant condition (default)
 - 6 LOG_INFO, informational
 - 7 LOG_DEBUG debug-level messages
-

SYSLOG_PORTBASE

The (UDP) port number for DicosSyslog

Explanation

The default should not be altered on a real system, but may be changed when running multiple Vegas simulators on the same host. 5900.

Default Value

= 5900

Value Range

Integer <UDP port number>

Comments

- This parameter value must match the port number given in the "-t" option to the **DicosSyslogClient** program
-

StopInMonitor

Enables the “Stop in Monitor” function for advanced debugging.

Explanation

When a processor crashes and this parameter is set to 1:

- The processor will not be restarted
- The TelORB proprietary PROM monitor is started

Default Value

= 0

Comments

- This parameter should only be set for advanced debugging and never on a real system.
-

Environmental Data

Under the ENVIRONMENT DATA keyword of the .epct file you can define the applications own data.

One of the main advantage of using environment data is that they can be changed at run-time by operators, whereas initial data cannot. Environment data can be accessed by the application at run-time via the DicosEnvironment interface declared in DicosEnvironment.hh for C++ applications. For Java applications this data is accessed through the `se.ericsson.telorb.util.Environment` class.

The data you write here will be contained in the loadingGroup01/DicosEnvironment file of the site configuration.

You can change environment data through telnet to port 8000 and going to the `env` directory, then use the `setenv` command to set an environment variable.

TelORB environment data in the .epct file can be divided into a number of areas:

- ◆ [AXmp Environment Variables](#)
- ◆ [Ethernet Switch Environment Variables](#)
- ◆ [Network Redundancy Environment Variables](#)
- ◆ [RMS / HW Alarms Environment Variables](#)
- ◆ [Other Environment Variables](#)

Alphabetic Index

[AXMP ADDRESSES](#)[AXMP POLL INTERVAL](#)[DnrcAutoGiveBack](#)[DnrcMirrorTailDelay](#)[ETHSW ADDRESSES](#)[ETHSW POLL INTERVAL](#)[NetRedDisableTraffic Decision](#)[NetRedEnableZone ReloadDetection](#)[RMS IP ADDRESS](#)[RMS NO CABINET B](#)[RSLimit](#)[AXMP HIGH TEMP](#)[AXMP VOLTAGE INTERVAL](#)[DnrcInterZoneTimeDiff](#)[DnrcTransferFinished Timeout](#)[ETHSW CFG INTERVAL](#)[NetRedDisableMonitors](#)[NetRedEnableScheduler](#)[RMS INSANE TEMP](#)[RMS MAX TEMP](#)[RMS POLL INTERVAL](#)[ShellHttpAccess](#)

AXmp Environment Variables

If your system uses SS7 signalling, then it will probably have an AXmp (a multi CPU Ultra Sparc computer in a standard rack-mountable packaging) to terminate the SS7. The AXmp is an

external (non TelORB) support processor. The following table lists environment variables for the supervision provided through the optional AXmp_IDP, [IDP/D8](#).

AXMP_ADDRESSES	List of symbolic names for AXmp processors Explanation This string value is a list of symbolic names for the AXmp processors to supervise. A UDP port can be specified, and should be separated from the name with a colon. The default port is 161, which is the standard port for SNMP. Example AXMP_ADDRESSES = "AXMP0:9977,AXMP1:9977" Comments • This parameter should be set when the optional AXMp_IDP is used.
AXMP_HIGH_TEMP	AXmp temperature alarm limit Explanation This string value specifies temperature in degrees Celsius at which the temperature alarm is generated. The default is 50 degrees. Default Value = "50"
AXMP_POLL_INTERVAL	AXmp poll interval Explanation The AXmps are polled rather than being checked with SNMP traps. This string value specifies the poll interval in milliseconds. The default is 20,000, that is, every 20 seconds. Default Value = "20000"

**AXMP_VOLTAGE_
INTERVAL**

Accepted CPU core voltage range

Explanation

This string value specifies the accepted range for CPU core voltage specified in 1/10 Volts (25 thus means 2.5 Volts). Default value is 24-31 (that is, 2.4 to 3.1 Volts). If the voltage drops below, or peaks above, this range, the appropriate alarm is generated.

Default Value

= "24-31"

Ethernet Switch Environment Variables

ETHSW_ADDRESSES

List of IP-addresses for supervised ethernet switches.

Explanation

This string value is a comma separated list of IP-addresses of the ethernet switches to supervise. The first will be instance “A” in the alarms, the second “B” and so on.

Example

ETHSW_ADDRESSES = "192.168.1.50,192.168.1.51,192.168.2.50,192.168.2.51"

ETHSW_CFG_INTERVAL

Specified how often TelORB checks for environment data changes

Explanation

This string value specifies (in milliseconds) how often TelORB checks the environment data for changes. TelORB is not notified automatically when the operator changes the environment, so this has to be checked regularly. Checking the environment data is a slow be done frequently. The default is every 30 minutes., (1,800,000 ms).

Default Value

= "1800000"

Value Range

We suggest that you keep the default value. The minimum value is 10,000.

ETHSW_POLL_INTERVAL	Specifies how often TelORB checks that the switches are available. Explanation This string value specifies (in milliseconds) the interval between polls to check whether or not the switches can be reached. The default value is every 15 seconds (15 000 ms) Default Value = "15000"
----------------------------	---

Network Redundancy Environment Variables

Environment variables for the optional NetRed_IDP, [IDP/D2](#)

DnrcMirrorTailDelay	Network redundancy variable used at Smooth Takeover and Give Back. Explanation This variable sets the maximum time a Smooth Takeover or a Give Back is delayed to await the tail of mirrored transactions before comparing the transactions counters. Default Value = "500" Value Range In a system with high latency between zones the value might have to be increased to double the default value or more to ensure successful Smooth Takeover or Give Back completion.
----------------------------	---

RSLimit

Network redundancy variable used to set the maximum allowed number of objects in a quick sync.

Explanation

The value will be the maximum number of objects in the compensating transaction.

Default Value

= 30000

DnrcInterZoneTimeDiff

Configurable time difference between two Network Redundancy zones.

Explanation

The value of this parameter should be increased if the time difference between the zones exceeds the default value. This may be the case for mated pairs of zones that are not synchronized from a common NTP server.

Default Value

5 minutes

Value Range

If the DnrcInterZoneTimeDiff is set "too high" the only consequence is that it may take unnecessarily long time for a standby to get back in sync after being out of sync and having performed an external sync.

A lower value than default is not recommended. If you set a lower value, you must be certain that the time difference in reality is less than the default value.

NetRedEnableScheduler	Network redundancy variable used to turn on the use of Jambala Scheduler Explanation This variable toggles between using the Jambala Scheduler and using the new Network Redundancy features that do not make use of the Jambala Scheduler Default Value = 0 Value Range 0 = Use the new Network Redundancy features 1 = Do not use the new features, use the Jambala Scheduler instead.
NetRedDisableTraffic Decision	Network Redundancy variable used to disable the Traffic Decision logic. Explanation This variable allows to disable the Traffic Decision logic within the NetRedController. Should this logic be disabled, no traffic affecting actions (such as Takeovers) will be taken due to Monitor state changes. Normal takeovers will not be affected, however. Default Value = 0 Value Range 0 = Use the Traffic Decision logic 1 = Disable the Traffic Decision logic

**DnrcTransferFinished
Timeout**

Network Redundancy variable used to configure the timeout value of a backup transfer from one zone to another as performed by the TransferManager.

Explanation

The NetRed controller will timeout after the amount of time set by this variable when waiting for a backup to be transferred to the other zone.

Default Value

= 120 (2 hours)

Value Range

The value is set in minutes

DnrcAutoGiveBack

Network Redundancy variable used to enable automatic giveback.

Explanation

Used to enable automatic giveback every time a manual reconciliation is performed.

Note that an automatic giveback will *always* be performed after a manual reconciliation when on the preferred Primary of a HotStandby NetRed system.

Default Value

= 0

Value Range

0 = No auto giveback.

1 = Enable auto giveback in any NetRed system.

**NetRedEnableZone
ReloadDetection**

Network Redundancy variable used to enable the detection of a zone reload of the remote zone.

Explanation

This variable is used to turn on the ZoneReload detection feature. This feature is a periodic verification of the state of the update channel; if this fails, the local zone perform a takeover since the other zone is not reachable, which can indicate a zone reload.

Default Value

= 0

Value Range

0 = Do not use the zone reload dection feature
1 = Enable the zone reload dection feature

NetRedDisableMonitors

Network Redundancy variable used to disable any communication between the StateMonitors and the NetRed Controller.

Explanation

This variable is used to disable the communication between the StateMonitors and the NetRed Controller. When set, it will ignore any command issued to the NRC by a StateMonitor process.

Default Value

= 0

Value Range

0 = use the StateMonitors function
1 = Disable the StateMonitor function

GEM Cabinet Hardware Supervision

Environmental variables for the GemScbSupervision_IDP (IDP/D22) This TelORB package (IDP) reads the HW status from the SCB boards and issues HW alarms.

A TelORB system should be configured with either the

- ◆ GemScbSupervision_IDP ([IDP/D22](#))

or, in the case of an older cabinet:

- ◆ KnuerrRms_IDP ([IDP/D3](#))

but never with both at the same time.

GemScbSupervision is configured with the DicosEnvironment. The environment variables are:

GEM_SCB_ADDRESSES	List of IP-addresses for supervised SCBs.
--------------------------	---

Explanation

This is a comma separated list of SCBs to supervise. The SCBs should be defined with their symbolic IP names, for example "SCB0". This symbolic name is used as "instance" id for hardware alarms from the SCB.

An optional port number can be added to the name. The port number should be separated with a colon ":".

Example: "SCB0:9978". If no port is given the standard SNMP port is used (161).

The symbolic names should be entered in the "DNS" in the TelORB system (see paragraph 3.1).

Value Range

Example

Comments

-

GEM_SCB_POLLINTERVAL

Explanation
The SCBs are polled by the TelORB system (SNMP traps are not used). The GEM_SCB_POLLINTERVAL specifies the poll interval in milliseconds. The default is 15000.

Value Range

Default

Example

Comments

-

GEM_SCB_MAXFANS

Value Range

Example

Comments

-

GEM_SCB_ADDRESSES -

This is a comma separated list of SCBs to supervise. The SCBs should be defined with their symbolic IP names, e.g. "SCB0". This symbolic name is used as "instance" id for HW-alarms from the SCB.

An optional port number can be added to the name. The port number should be separated with a colon ":". Example: "SCB0:9978". If no port is given the standard SNMP port is used (161).

The symbolic names should be entered in the "DNS" in the TelORB system (see paragraph 3.1).

GEM_SCB_POLLINTERVAL -

The SCBs are polled by the TelORB system (SNMP traps are not used). The GEM_SCB_POLLINTERVAL specifies the poll interval in milliseconds. The default is 15000.

GEM_SCB_MAXFANS -

An SCB can have up to 8 fan units connected, but usually they have only one (or none). You can set the number of checked fans with this parameter. The default is 1.

NOTE: This parameter only affects performance. The supervision will work perfectly with this value set to 8, even if only one fan unit is connected. In that case the system will however try to supervise all 8 fans units, which is unnecessary.

If the GemScbSupervision_IDP is loaded but no value is specified for GEM_SCB_ADDRESSES, you will get an alarm saying: "No GEM-SCB addresses given, HW-supervision disabled"

Example:

A typical GEM-SCB configuration in the epct file:

```
ENVIRONMENT DATA
GEM_SCB_ADDRESSES = "SCB0:161, SCB1:161, SCB2:161, SCB4:161"
GEM_SCB_POLLINTERVAL = 15000
GEM_SCB_MAXFANS = 1
ENDDATA
```

RMS / HW Alarms Environment Variables

Environmental variables for the optional Hardware Alarms Collector (KnuerrRms_IDP).
A TelORB system should be configured with either the GemScbSupervision_IDP (IDP/D22) or the KnuerrRms_IDP (IDP/D3), in case of an older cabinet, but not both at the same time.

RMS_IP_ADDRESS	The IP address for the Knuerr Hardware Alarms Collector Value Range A string with the IP address of your Hardware Alarms Collector. We recommend that you create a symbolic hostname “RMS” for the IP-address in TelORB DNS. Example RMS_IP_ADDRESS = "RMS:5555" (An optional port can be passed after the :) Comments • This parameter should be set when the optional Hardware Alarms Collector KnuerrRms_IDP is used.
RMS_NO_CABINET_B	Specifies the number of cabinets in the system. Used by the Knuerr Hardware Alarms Collector Value Range RMS_NO_CABINET_B = 0 the system consists of two cabinets RMS_NO_CABINET_B = 1 the system consists of one cabinet Default Value = 0 (Two cabinets)

RMS_MAX_TEMP

Temperature level for Knuerr RMS maximum temperature alarms

Explanation

The temperature (in degrees Celsius) at which the RMS will start to report temperature alarms.

Default Value

= 55

Value Range

0 to 99

RMS_POLL_INTERVAL

Poll interval for the Knuerr Hardware Alarms Collector.

Explanation

The interval (in milliseconds) between polls of the Knuerr RMS by TelORB.

Default Value

= 15000 (15 seconds)

RMS_INSANE_TEMP

Used to suppress false alarms.

Explanation

When the temperature sensor is Not Connected (NC) it should not trip alarms.

Default Value

= 5

Comments

- To disable all temperature alarms, set this value lower than [RMS_MAX_TEMP](#)
-

Other Environment Variables

ShellHttpAccess	Enables the http interface of the TelORB Shell. Explanation This variable controls access to the http interface of the TelORB Shell. Default Value = "ON" Value Range Default is "ON", any other settings turns off access to the http interface.
-----------------	--

Processors

Keywords

Each processor specification begins with the PROCESSOR keyword and end with a semicolon (;)

For each processor the following information is listed:

PROCESSOR		The name of the processor. Must be unique within the TelOrb system.
INITIAL		Used with the LOADER keyword to specify that the processor acts as initial loader.
TARGET	Sparc i486	Target (Sparc or Intel Pentium)
TYPE	IntelPC Vega	The processor type
LOADER		The processor acts as loader. A minimum of two processors should be defined as loaders. If your TelORB cabinet contains GEM Hardware, you should ensure that the processors you define as loaders do not all reside in the same magazine.
ADDRESS		The ethernet addresses of the processor, two for each processor
IN		The processor pools that the processor participates in
ALSO IN		Use PROCESSOR with this keyword to add a processor specified elsewhere to a processor pool.

Example

At a real site you would also find some application defined pools. The pools found in the example below all belong to the TelORB operating system.

PROCESSOR Processor1

INITIAL

TARGET i486

TYPE IntelPc

LOADER

ADDRESS "00:80:50:00:94:81 00:80:50:00:91:42"

IN DicosCommonPool, DicosPlgLoaderPool, DicosDbReplicaPool,
DicosQnPool;

PROCESSOR Processor2

TARGET i486

TYPE IntelPc

LOADER

ADDRESS "00:80:50:00:cb:ed 00:80:50:00:8f:52"

IN DicosCommonPool, DicosPlgLoaderPool, DicosDbReplicaPool,
DicosQnPool;

PROCESSOR Processor3

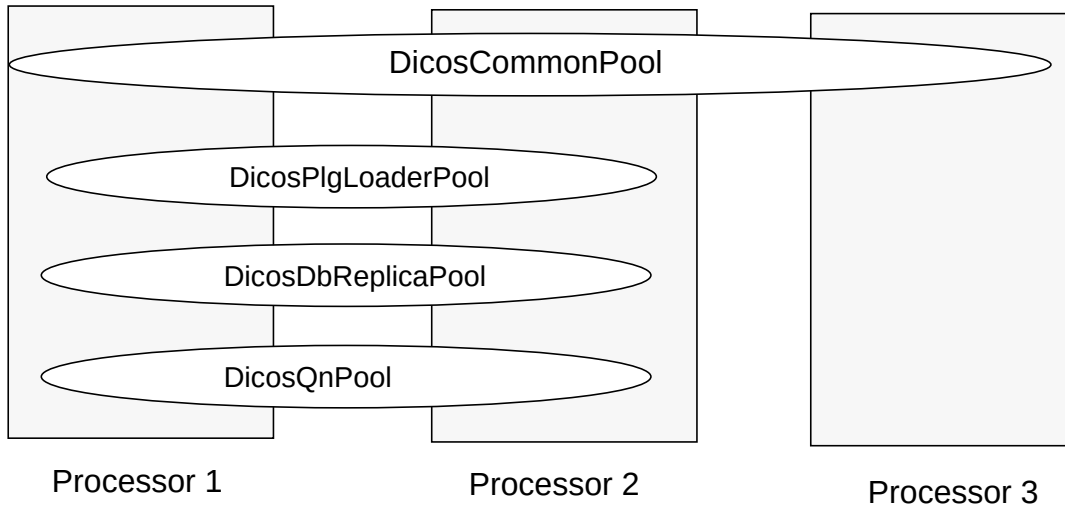
TARGET i486

TYPE IntelPc

ADDRESS "00:80:50:00:94:8d 00:80:50:00:91:75"

IN DicosCommonPool;

The pool configuration above corresponds to the figure below. Every pool introduced by the application must also be defined, that is done in a file called config.epct stored under the IDP folder.



Pool Configuration for the Optional NetRed_IDP

To specify separately that processors should be included in to the DicosNetRedPool and DicosNetRedConnectPool when using the optional NetRed_IDP, add PROCESSOR entries with the ALSO IN keyword to your .epct file as in the following example:

```
PROCESSOR Processor1
    ALSO IN DicosNetRedPool, DicosNetRedConnectPool;
PROCESSOR Processor2
    ALSO IN DicosNetRedPool, DicosNetRedConnectPool;
PROCESSOR Processor3 ALSO IN DicosNetRedPool;
```

