

SOFTWARE CENTER OPERATOR MANUAL

for

CARDAMOM

CARDAMOM

CONTRACT N° 02-067

Prepared by:

THALES

45, rue de Villiers
92526 Neuilly-Sur-Seine CEDEX
FRANCE

SELEX Sistemi Integrati S.p.A.

Via Tiburtina, Km 12.400
00131 ROMA
ITALY

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	1 of 70

CARDAMOM_General_Template_REV08

THIS DOCUMENT IS PART OF CARDAMOM ® WHICH IS JOINTLY DEVELOPED BY THALES AND SELEX SISTEMI INTEGRATI S.P.A.. ALL RIGHTS RESERVED.
COPYRIGHT © THALES AND SELEX SISTEMI INTEGRATI S.P.A. 2004-2008. ALL RIGHTS RESERVED.

CARDAMOM		
<i>DOCUMENT TITLE:</i> SOFTWARE CENTER OPERATOR MANUAL		
<i>DOCUMENT CONTENT & PURPOSE:</i> The purpose of this document is to provide installation procedures for CARDAMOM and associated tools.		
INTERNAL APPROVALS		
<i>REVISION:</i> 11	<i>Signature</i>	<i>Date</i>
<i>WRITTEN BY:</i> F. MARCHIGIANI and J. ENOCH		23/06/2008
<i>CHECKED BY:</i> Quality Assurance: C. WYLOMANSKI		
<i>APPROVED BY:</i> Technical Architects: F. MORCIANO and H. SOUAMI		
EXTERNAL APPROVALS		

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	2 of 70

CHANGES

Rev.	Date	Description	Person in Charge of	Change Request Reference
01	06/02/2004	First version	J.ENOCH	-
02	29/06/2004	Updated tools installation.	M. RIZZUTO	PCR-0040
03	06/07/2004	Updated package names.	T.HUYNH	ECR-0025
04	02/08/2004	Merged CDO note ref. CDO-040729-TEC-2-040 with Rev03. Added constraint on the programmer's environment. Updated versions of tools.	T.HUYNH	ECR-0014
05	18/01/05	Updated with the applicable modifications of PCR-0073 (issued on SCOM rev. 03). Updated with the corrections requested by CDO-050114-PRS-1-137.	M. RIZZUTO	PCR-0073 CDO-050114-PRS-1-137.
06	03/02/2005	Add Open Fusion JacORB 2.1.3 Installation procedure Add Open Fusion TAO 1.4.1 Installation procedure Remove OpenORB Time and Event services installation procedures	R.PANELLA S.MCQUEEN A.ROUSSEL	ECR-0083 ECR-0068 PCR-0216
	08/02/2005	Added R 1.9.0 Installation procedure (statistical tool)	P.PAPA	ECR-0076
	17/02/2005	Added cppunit 1.10.2 Installation procedure.	F.MORCIANO	PCR-0219
	21/02/2005	Added doxygen 1.4.1 Installation procedure. Fixed cppunit installation procedure. Fixed R installation procedure.	F.MARCHIGIANI F.MORCIANO	PCR-0241 PCR-0219 ECR-0076
	22/02/2005	Removed Jacorb 2.0p2 Installation procedure. Update "needed-tools" output	R.PANELLA A.ROUSSEL	ECR-0083 PCR-0216
07	18/04/2005	Update the installation procedure for Xerces c++	F. MOTTE	PCR-0339

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	3 of 70

Rev.	Date	Description	Person in Charge of	Change Request Reference
	29/04/2005	Updated the Version of TAO to use Clarify the way to install OpenFusion JacOrb 2.1.3 Re-number the list of tools to install according to their installation order. The ordered list is given before the description of the tools and not at the end. Remove the requirement to perform "startCdmw.sh configure" before "startCdmw.sh needed-tools"	M.SAHEB	PCR-0316 PCR-0326 PCR-0355
	15/06/2005	Change of company name	H.P. HENQUEZ	ECR-0140
	28/06/2005	Update SCOM according to PCR-0390 Update appendix C	J. ENOCH	PCR-0390 PCR-0421
	04/07/2005	Updated with the corrections requested by CDO-050701-PRS-1-231.	J. ENOCH	CDO-050701-PRS-1-231
08	25/07/2005	Updated appendix E according to new needed-tools output	B. HERGOTT	PCR-0420
	19/08/2005	Update SCOM according to PCR-0474	J. ENOCH	PCR-0474
	26/08/05	Update SCOM according to ECR-0180 related to Splice DDS 1.3	M.SAHEB	ECR-0180
	15/09/05	Update after peer review	B. HERGOTT	CDO-050912-PRS-1-297 PCR-0498
09	03/10/05	CLASSPATH Problem Management of different CARDAMOM version New directory into the c++ include directory Correction after the official qualification of CARDAMOM V2.1	F. MOTTE	PCR-0497 PCR-0465 PCR-0526 PCR-0576

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	4 of 70

Rev.	Date	Description	Person in Charge of	Change Request Reference
	30/10/05	Change the required minimum computer hardware reference to be coherent with SSS document	D BECU	PCR-0464
	22/12/05	Change OpenCCM version in 0.4p7	J. ENOCH	PCR-0165
	23/02/06	Perform the staticdeps during the configuration and consider all the CSCIs under the same directory	F. MOTTE	ECR-0240
	10/04/06	SCOM Remarks from ATM staff during CARDAMOM installation	L. BONENFANT	PCR-0736
	11/05/06	Modifications for V3.0 : new JDK 1.5, new Jacorb 2.1.3.3, new OpenCCM0.4p8, new TAO	B.HERGOTT	ECR-0279
	30/06/06	Update TAO release number	A.ROUSSEL	ECR-0300
	05/10/06	Add '.' to PATH during OpenCCM Installation	H. SOUAMI	PCR-0835
	10/10/2006	ant : place of /usr/bin jacob : - precision of default installation - path of build.xml OpenOrb : precision of commands javacc : - precision of script kind - remove install directory xalan : - precision of commands - installation order OpenCCM : PATH modification Xerces C : precision of commands	L. BONENFANT	PCR 0894
	02/11/2006	Added configuration for automatic starting of PlatformDaemon on host reboot.	M.RIZZUTO	ECR-0328
	10/11/2006	Update JDK version to 1.5.0_08 Update JacORB version to 2.3	F.MARCHIGIANI	PCR-0806 ECR-0326
	14/11/2006	Added TAO151_0_060524p1 installation procedure	F.MARCHIGIANI	ECR-0367
	24/11/2006	Add NTP configuration	F.ANNUNZIATA	ECR-0376
	14/12/2006	Updated SCOM with the corrections requested by the peer review	F.MARCHIGIANI	CDO-061124- PRS-1-516
	21/12/2006	Added binary installation procedure	F.MARCHIGIANI	PCR-0863

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	5 of 70

Rev.	Date	Description	Person in Charge of	Change Request Reference
10	07/03/2007	Modifications for V3.1p02: Removed TAO141_2 installation procedure.	F.MARCHIGIANI	ECR-0296
	20/03/2007	JUnit and Emma installation	A.BATCHELOR	ECR-0437
	07/06/2007	Update Emma installation	J. ENOCH	PCR-1155
	07/06/2007	Update JacORB installation (no more create jacorb.properties)	J. ENOCH	ECR-0454
	06/07/2007	Build of TAO : compilation of IFR_Service added	J. MARLINE	ECR-0402
	12/07/2007	Updated Xerces C++ version (2.7.0) and installation procedure.	F. MORCIANO	PCR-1222
	28/09/2007	Updated description of install-binary. command	F.MARCHIGIANI	PCR-1293 PCR-1296
	28/09/2007	Moved description of cdmw_config.sh generated shell script.	F.MARCHIGIANI	PCR-1294
	01/10/2007	Updated description of cdmw_config.sh generated shell script.	F.MARCHIGIANI	PCR-1295
	10/10/2007	Added description of new config files, generated during the Cardamom installation into the share directory.	F.MARCHIGIANI	ECR-0242
	17/10/2007	Added CDMW patch application procedure.	F.MARCHIGIANI	PCR-1265
	29/10/2007	Update TAO installation for DanCE compilation.	J. ENOCH	ECR-0453
	30/11/2007	Updated with the corrections requested by CDO-071102-PRS-1-661.	F.MARCHIGIANI	CDO-071102-PRS-1-661

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	6 of 70

Rev.	Date	Description	Person in Charge of	Change Request Reference
	30/11/2007	Update Demo Directory description	F.MARCHIGIANI	ECR-0475 ECR-0477 ECR-0480 ECR-0453 PCR-1068
11	14/12/2007	Update cdmw_config description	F.MARCHIGIANI	PCR-1549
	18/12/2007	Update description of automatic start of CARDAMOM services.	F. DI MATTEO	ECR-0579
	22/01/2008	Removal of Xalan and Xerces-JAVA Ant problem with nohup	F. DI MATTEO F.MARCHIGIANI	ECR-0536 PCR-1643
	01/02/2008	Alignment of CDMW compilation flag with TAO.	F. MORCIANO	PCR-1682
	13/03/2008	Update for cdmw_libs output.	F. MORCIANO	PCR-1670 PCR-1671
	14/04/2008	Update SCOM according to ECR-0089	F.MARCHIGIANI	ECR-0089
	23/06/2008	Updated with the corrections requested by CDO-080512-PRS-2-318.	F.MARCHIGIANI	CDO-080512-PRS-2-318

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	7 of 70

LIST OF ABBREVIATIONS

Refer to CARDAMOM Glossary document (ref [4]).

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	8 of 70

TABLE OF CONTENTS

1.	SCOPE	13
1.1	IDENTIFICATION	13
1.2	PRODUCT OVERVIEW	13
1.3	DOCUMENT OVERVIEW	14
2.	REFERENCED DOCUMENTS	15
2.1	CONTRACTUAL DOCUMENTS	15
2.2	CDO - CARDAMOM DOCUMENTS	15
2.3	CSCI DOCUMENTS	16
2.4	OTHER DOCUMENTS	16
3.	SOFTWARE OPERATION SUMMARY.....	17
3.1	SOFTWARE APPLICATION	17
3.2	SOFTWARE INVENTORY.....	17
3.2.1	CARDAMOM Installation Directory	17
3.2.2	Demo Directory	18
3.3	SOFTWARE ENVIRONMENT	18
3.3.1	Required Environment	18
3.3.1.1	Hardware	18
3.3.1.2	Software.....	19
3.3.2	ORB Configuration.....	19
3.3.2.1	ORBacus Configuration	19
3.3.2.2	ORBacus Java configuration	19
3.3.2.3	TAO configuration.....	19
3.3.3	Operating System Configuration.....	19
3.3.3.1	Linux Configuration.....	19
3.3.3.2	NTP configuration	20
3.4	SOFTWARE ORGANIZATION AND OVERVIEW OF OPERATION.....	24
3.4.1	Package Overview	24
3.4.2	Installation Overview.....	25
3.5	CONTINGENCIES AND ALTERNATE STATES AND MODES OF OPERATION	25
3.6	SECURITY, PRIVACY AND INTELLECTUAL PROPERTY PROTECTION	25
3.7	ASSISTANCE AND PROBLEM REPORTING.....	25

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	9 of 70

4.	INSTALLATION AND SETUP	26
4.1	CD DESCRIPTION	27
4.1.1	Source CD.....	27
4.2	INSTALLATION PROCEDURE.....	28
4.2.1	Patches	28
4.2.1.1	patching tools.....	28
4.2.1.1.1	ORBacus C++.....	29
4.2.1.1.2	ORBacus Java.....	29
4.2.1.2	patching cardamom	29
4.2.2	Configuration.....	30
4.2.3	Compilation	31
4.2.4	Testing	31
4.2.5	Installation.....	32
4.2.5.1	Installation of binary delivery	32
4.2.5.2	environment variables setting.....	35
4.3	CLEANUP	43
4.4	MANAGEMENT OF DIFFERENT VERSION OF CARDAMOM	44
4.4.1	Configuration listing	44
4.4.2	Re-use of old configuration.....	45
5.	DESCRIPTION OF RUNS	46
6.	APPENDIX A: TOOLS INSTALLATION ARCHITECTURE	47
7.	APPENDIX B: INSTALLATION OF THE TOOLS ON LINUX PC	48
7.1	GCC 3.4.4 INSTALLATION.....	49
7.2	JDK INSTALLATION.....	49
7.3	ANT INSTALLATION	49
7.4	OPENFUSION TAO INSTALLATION	50
7.5	OPENFUSION JACORB INSTALLATION.....	51
7.6	OPENORB INSTALLATION.....	52
7.7	JAVACC INSTALLATION	52
7.8	OPENCCM INSTALLATION	53
7.9	XERCES C++ INSTALLATION	54
7.10	SAXON INSTALLATION	54

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	10 of 70

7.11	R INSTALLATION	54
7.12	SPLICE-DDS V1.3	55
7.13	CPPUNIT INSTALLATION	55
7.14	JUNIT INSTALLATION	56
7.15	DOXYGEN INSTALLATION.....	56
7.16	JAVA ORBACUS 4.1.0 INSTALLATION.....	56
7.17	ORBACUS 4.1.0 INSTALLATION.....	57
7.18	VALGRIND INSTALLATION	58
7.19	EMMA INSTALLATION	58
8.	APPENDIX C: CONFIGURE EXAMPLE	60
9.	APPENDIX D: TYPICAL REPORT OF THE RUNTEST.....	64
10.	APPENDIX E: NEEDED-TOOLS TARGET EXAMPLE.....	66
11.	APPENDIX F: CHECK-HOST-RESOURCES TARGET EXAMPLE.....	69
12.	APPENDIX G: GETTING STARTED.....	70

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	11 of 70

LIST OF FIGURES

Figure 4-1 cdmw_config.sh options.....	35
Figure 4-2 cdmw_config.sh usage example.....	38
Figure 4-3 generated include files example.....	43

LIST OF TABLES

Table 7-1 tools installation order	48
--	----

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	12 of 70

1. SCOPE

1.1 IDENTIFICATION

• Document Name	: SOFTWARE CENTER OPERATOR MANUAL
• Document Short Name	: SCOM
• CSCI Short Name	: CDM
• CDRL Number	: N/A
• THALES Number	: 61 486 532 AA
• SELEX Sistemi Integrati Number	: R216-03-1033SCOM
• CTD	: 593
• Revision	: 11
• Revision Date	: 23/06/2008
• File Name	: CARDAMOM SCOM Rev11.doc

This Document is applicable to the **CARDAMOM** Product.

CARDAMOM is free software; you can redistribute it and/or modify it under the terms of the GNU Library General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. CARDAMOM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Library General Public License for more details. You should have received a copy of the GNU Library General Public License along with CARDAMOM; see the file COPYING. If not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

1.2 PRODUCT OVERVIEW

CARDAMOM is a middleware platform enabling component based software architecture for the implementation of safety and mission critical systems, such as those for air traffic control and combat management systems.

CARDAMOM provides a framework for the integration of both **Business Components** of the **functional** architecture and **Technical Components** of **non-functional** architecture.

Components are either proprietary components or COTS (Commercial Off The Shelf) components.

To provide a standard Plug and Play of a large set of components, **CARDAMOM** is based on the main interoperability **standards**, defined by the **Object Management Group** (OMG) organisation:

- At business level **CARDAMOM** uses **UML** (OMG standard) and **XML** (W3C standard),
- As separation layer **CARDAMOM** uses **CCM** OMG standard in order to isolate the business logic from the technical services,
- At technical level **CARDAMOM** uses **CORBA** OMG standard.

For a more complete overview of **CARDAMOM** refer to the Product Overview Document (ref. [5]).

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	13 of 70

1.3 DOCUMENT OVERVIEW

This document contains the description of the CARDAMOM, and the related tools, installation procedure.

- [Chapter 1](#) provides an overview of the CARDAMOM product and the purpose of this document.
- [Chapter 2](#) lists all the documents referenced in this document.
- [Chapter 3](#) defines CARDAMOM concepts and describes services available in CARDAMOM, its installation directory and its required environment.
- [Chapter 4](#) describes in details the installation and set up procedure.
- [Chapter 5](#) describes how to develop an application using CARDAMOM.
- [Chapter 6](#) propose a specific architecture for the tools installation
- [Chapter 7](#) describes the installation procedures for tools on Linux
- [Chapter 8](#) and [Chapter 9](#) present typical traces for the CARDAMOM configuration and test.
- [Chapter 10](#) and [Chapter 11](#) present typical trace for some CARDAMOM Build system utility commands.
- [Chapter 12](#) is a "Getting Started with CARDAMOM"

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	14 of 70

2. REFERENCED DOCUMENTS

2.1 CONTRACTUAL DOCUMENTS

Name of the document	Acronym	Reference number
[1] CARDAMOM Collaboration Agreement	N.A.	Contract 02-067, 19/05/2003
[2] CARDAMOM Statement Of Work for a THALES/AMS Common Software Platform Project	SOW	26/05/2003

2.2 CDO - CARDAMOM DOCUMENTS

Name of the document	Acronym	Reference number
[3] CARDAMOM Product Item and Documentation Index	PIDI	PIDI-CI: 61 486 532 AA / E295-07-00014PDC-001 PIDI-DOC: 61 486 532 AA / E295-07-00015PDD - 021 PIDI-Tools: 61 486 532 AA / E295-07-00016PDT - 190
[4] CARDAMOM Glossary	GLO	61 486 532 AA -E295-06-0003GLO - 447
[5] CARDAMOM Product Overview	PROV	61 486 532 AA - E295-06-00005PROV - 502
[6] CARDAMOM Version Description Document	VDD	61 486 532 AA - R216-03-1034SVDD - 498
[7] CARDAMOM Software User Manual	SUM	61 486 532 AA - R216-03-1031SUM - 108
[8] CARDAMOM Software Configuration Management Plan	SCMP	61 486 532 AA - R216-02-0916SCMP - 396
[9] CARDAMOM Build Software User Manual	BMG SUM	61 486 533 AAA - E216BMG0-01SUM - 108
[10] CARDAMOM System Segment Specification	SSS	61 486 532 AA – E295-02-00006SSS - 305
[11] CARDAMOM Qualification Test Plan / Description / Report	QTPDR	61 486 532 AA - E295-02-00013QTPDR - 206

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	15 of 70

The exact issue of the documentation is given in the CARDAMOM PIDI (ref [3]).

2.3 CSCI DOCUMENTS

Not Relevant.

2.4 OTHER DOCUMENTS

Name of the document	Acronym	Reference number
[12] ORBacus for C++ and Java, version 4.0.4		http://www.omg.org/technology/corba/corbadownloads.htm

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	16 of 70

3. SOFTWARE OPERATION SUMMARY

3.1 SOFTWARE APPLICATION

For an overview of CARDAMOM refer to the Product Overview Document (ref. [5]) and to the chapter 3 of the Software User Manual (ref [7]) for a more complete overview.

3.2 SOFTWARE INVENTORY

3.2.1 CARDAMOM INSTALLATION DIRECTORY

```
CDMW-INSTALLATION
|-- bin
|-- demos
|   |-- demo
|   |   |-- c++
|   |   |   |-- data
|   |   |   |-- idl
|   |   |   |-- include
|   |   |   |-- run
|   |   |   |-- src
|   |   |   |-- xml
|   |   |-- java
|   |       |-- data
|   |       |-- idl
|   |       |-- run
|   |       |-- src
|   |       |-- xml
|   |-- demo_ccm
|   |-- demo_ccm_ft
|
|   .....
|   |-- demo_philosophers
|   |-- philosophers_idls
|-- idl
|-- include
|   |-- c++
|   |   |-- CCMContainer
|   |   |   |-- ccmCIF
|   |   |   |-- ccmcommon
|   |   |   |-- idllib
|   |   |       |-- TA00OpenFusion-1.5.1
|   |   |-- CCMDeployment
|   |   |-- ConfAndPlug
|   .....
|   |-- SystemMngt
|   |-- TA00OpenFusion
|   |-- Time
|   |-- TraceAndPerf
|   |-- XMLServer
|-- lib
|   |-- c++
|   |-- java
|-- share
```

On the left, there is an example of snapshot of the CARDAMOM installation directory contents:

with:

- *bin*: executables, scripts and Jar files
- *demos*: demonstration C++, and Java applications
- *include*: include with a subdirectory:
- *c++* for C++ .hpp files. In this directory are all the CSCI directory plus a specific directory named with the name of the ORB used. This directory contains all the stub and skel generated by the CARDAMOM idl files.
- *idl*: IDL files
- *lib*: libraries with two subdirectories:
- *c++* for C++ libraries
- *java* for Java libraries
- *share*: directory where to place not included in the directories mentioned above (such as XML DTD files, Java and C++ templates used by the code generator, ORB configuration files)

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	17 of 70

3.2.2 DEMO DIRECTORY

Several demonstration applications are provided with the CARDAMOM CD:

- a C++ demo
- a Java demo
- a Fault Tolerant (FT) demo
- demo_crash, a demo to test different cases of crash (C++)
- demo_perf, that provides an example of performance measurement in a basic Cklient Server CARDAMOM application
- federated_controlled_executor and federated_controlled_executor_observer, that provides different examples of the use of federated clocks
- demo_state_transfer, that provides an example of use of the state transfer library for Fault Tolerance (C++)
- demo_state_transfer_no_memory_leak, that provides an example of memory leak problem in the backup, following state transfer between the primary server and its backups (C++)
- several CCM demos:
 - o demo_ccm, that provides a basic example of the use of the CARDAMOM CCM (for C++ and Java)
 - o demo_philosophers, that provides a CCM solution to the famous "Philosophers' problem" (for C++, Java and both C++ and Java)
 - o demo_ccm_ft, a CCM demo coupled with Fault Tolerance in C++
 - o philosophers_idls, a CCM demo that provides an example of CARDAMOM IDL 3 to IDL 2 generation (for C++, Java and both C++ and Java)
 - o demos of the CCM Deployment with DAnCE:
 - demo_dance1, that provides a basic example (for C++, Java and both C++ and Java)
 - demo_dance2, that provides an example with several nodes (C++)
 - demo_dance_ft, that provides an example with Fault Tolerance (C++)
 - demo_dance_lb, that provides an example with Load Balancing (C++)

Installed demonstrations depend on the CARDAMOM package(s) selected during "configure" step explained in section 4.2.2.

To run these demos, go in the appropriate directory and follow the instructions contained in the README file.

3.3 SOFTWARE ENVIRONMENT

3.3.1 REQUIRED ENVIRONMENT

3.3.1.1 HARDWARE

The recommended hardware platform is the one described in the SSS ([10]) in the requirement REQ-SSS-CNS-0100 (§3.10 Computer Resource Requirements).

The hardware platform used to compile, install and qualify the CARDAMOM product is the one described in the QTPDR document ([11] - §3.1 Description of test sites).

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	18 of 70

3.3.1.2 SOFTWARE

Refer to PIDI ([3]).

Note that while CARDAMOM provides a lot of services with many languages and ORB, the user is not intended to install all tools identified in the PIDI.

To know exactly what software shall be installed, please use the command:

```
startCdmw.sh needed-tools
```

as explained in section 4.2.2.

3.3.2 ORB CONFIGURATION

3.3.2.1 ORBACUS CONFIGURATION

No specific configuration is needed since ORBacus is configured by CARDAMOM Run-time through XML Deployment File. Note that, for a proper operation, a patch has to be performed. Refer to section 4.2.1 for a description of this patch.

For detailed information about ORBacus configuration, refer to [12].

Note that, for a proper operation, a patch has to be performed. Refer to section 4.2.1 for a description of this patch.

3.3.2.2 ORBACUS JAVA CONFIGURATION

No specific configuration is needed since ORBacus Java is configured by CARDAMOM Run-time through XML Deployment File. Note that, for a proper operation, a patch has to be performed. Refer to section 4.2.1 for a description of this patch.

For detailed information about ORBacus Java configuration, refer to [12].

WARNING: ORBacus Java has to be used with boot classpath (option `-Xbootclasspath/p:`). See ORBacus Java install note for reasons. Thus, the Java parts of CARDAMOM are build with this option, and must be started with this option (this is already done into CARDAMOM launching scripts). The user Java applications must also use this option for compilation and run. But, when a user application uses a ClassLoader, some problems may occur at runtime. In this case, the user Jar should not be set into boot classpath, but into normal classpath.

Note that, for a proper operation, a patch has to be performed. Refer to section 4.2.1 for a description of this patch.

3.3.2.3 TAO CONFIGURATION

Note that, for a proper operation, a patch has to be performed. Refer to section 4.2.1 for a description of this patch.

3.3.3 OPERATING SYSTEM CONFIGURATION

3.3.3.1 LINUX CONFIGURATION

Please refer to PIDI document [3] for the exact Linux release required.

Check the `/etc/hosts` file; it must not contain the name of the host for the 127.0.0.1 address (*localhost* is ok).

In order to enable multicast on hosts, it might be needed to issue the following command:

```
/sbin/ip route add 224.0.0.0/4 dev eth0
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	19 of 70

(It is strongly advised to put it into the file /etc/rc.d/rc.local)

No other specific configuration operation has to be performed.

WARNING: The firewall of the OS must allow the multicast.

In order to automatically start the CARDAMOM services, their starting scripts can be put in file /etc/rc.d/rc.local as well. For PlatformDaemon, ftmanager and supervision, the following entries can be added (this allows to fully exploit the host reboot commands):

```
. <set_env_tools_script>
CDMW_INSTALL=<CDMW-INSTALLATION-PATH>
export LD_LIBRARY_PATH=$CDMW_INSTALL/lib/c++:$LD_LIBRARY_PATH
export PATH=$CDMW_INSTALL/bin:$PATH
$CDMW_INSTALL/bin/cdmw_platform_kill_all_pids.sh /tmp/CdmwDaemon_<port>
cd $CDMW_INSTALL/share
cdmw_host_start.sh
```

The Process cdmw_host_start.sh start the CARDAMOM services

Moreover, the following entries can be added in /root/.bashrc:

```
. <set_env_tools_script>
CDMW_INSTALL=<CDMW-INSTALLATION-PATH>
export LD_LIBRARY_PATH=$CDMW_INSTALL/lib/c++:$LD_LIBRARY_PATH
export PATH=$CDMW_INSTALL/bin:$PATH
```

where:

<set_env_tools_script> is a custom script which sets the environment for the needed tools

<CDMW-INSTALLATION-PATH> is the installation directory of CARDAMOM

3.3.3.2 NTP CONFIGURATION

For each host :

Check that the file /etc/ntp.conf contains correct information related to NTP configuration.

In order to synchronize the system at boot time:

- Server side: edit the file /etc/ntp.conf as in the following example (assuming an ntp server address 10.0.0.138):

o ntp.conf

```
# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
```

```
# -- CLIENT NETWORK -----
```

```
# Permit systems on this network to synchronize with this
# time service. Do not permit those systems to modify the
# configuration of this service. Also, do not use those
# systems as peers for synchronization.
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	20 of 70

```
# restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap

# --- OUR TIMESERVERS -----
# or remove the default restrict line
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.

# restrict mytrustedtimeserverip mask 255.255.255.255 nomodify notrap
# noquery
# server mytrustedtimeserverip

restrict 10.0.0.138 mask 255.255.0.0 nomodify notrap noquery

# --- NTP MULTICASTCLIENT ---
# multicastclient # listen on default 224.0.1.1
# restrict 224.0.1.1 mask 255.255.255.255 notrust nomodify notrap
# restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap

# --- GENERAL CONFIGURATION ---
#
# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available. The
# default stratum is usually 3, but in this case we elect to use stratum
# 0. Since the server line does not have the prefer keyword, this driver
# is never used for synchronization, unless no other other
# synchronization source is available. In case the local host is
# controlled by some external source, such as an external oscillator or
# another protocol, the prefer keyword would cause the local host to
# disregard all other synchronization sources, unless the kernel
# modifications are in use and declare an unsynchronized condition.
#
server 127.127.1.0 # local clock
fudge 127.127.1.0 stratum 10

#
# Drift file. Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
#
driftfile /var/lib/ntp/drift
broadcastdelay 0.008

#
# Authentication delay. If you use, or plan to use someday, the
# authentication facility you should make the programs in the auth_stuff
# directory and figure out what this number should be on your machine.
#
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	21 of 70

authenticate yes

```
#
# Keys file. If you want to diddle your server at run time, make a
# keys file (mode 600 for sure) and define the key number to be
# used for making requests.
#
# PLEASE DO NOT USE THE DEFAULT VALUES HERE. Pick your own, or remote
# systems might be able to reset your clock at will. Note also that
# ntpd is started with a -A flag, disabling authentication, that
# will have to be removed as well.
#
keys /etc/ntp/keys
```

- Client side: edit the file /etc/ntp/step-tickers adding a list of your time server(s) hostname(s), (time_server_name_IP_address or hostname) (according with information contained in ntp.conf) as in the following example:

```
o /etc/ntp.conf
# Prohibit general access to this service.
restrict default ignore
restrict 10.0.0.138 mask 255.255.255.255 nomodify notrap noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1

# -- CLIENT NETWORK -----
# Permit systems on this network to synchronize with this
# time service. Do not permit those systems to modify the
# configuration of this service. Also, do not use those
# systems as peers for synchronization.
# restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap

# --- OUR TIMESERVERS -----
# or remove the default restrict line
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.

# restrict mytrustedtimeserverip mask 255.255.255.255 nomodify
# notrap noquery
# server mytrustedtimeserverip
server 10.0.0.138

# --- NTP MULTICASTCLIENT ---
#multicastclient # listen on default 224.0.1.1
# restrict 224.0.1.1 mask 255.255.255.255 notrust nomodify notrap
# restrict 192.168.1.0 mask 255.255.255.0 notrust nomodify notrap
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	22 of 70

```
# --- GENERAL CONFIGURATION ---
#
# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available. The
# default stratum is usually 3, but in this case we elect to use stratum
# 0. Since the server line does not have the prefer keyword, this driver
# is never used for synchronization, unless no other other
# synchronization source is available. In case the local host is
# controlled by some external source, such as an external oscillator or
# another protocol, the prefer keyword would cause the local host to
# disregard all other synchronization sources, unless the kernel
# modifications are in use and declare an unsynchronized condition.
#
#server 127.127.1.0      # local clock
#fudge 127.127.1.0 stratum 10

#
# Drift file. Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
#
driftfile /var/lib/ntp/drift
broadcastdelay 0.008

#
# Authentication delay. If you use, or plan to use someday, the
# authentication facility you should make the programs in the auth_stuff
# directory and figure out what this number should be on your machine.
#
authenticate yes

#
# Keys file. If you want to diddle your server at run time, make a
# keys file (mode 600 for sure) and define the key number to be
# used for making requests.
#
# PLEASE DO NOT USE THE DEFAULT VALUES HERE. Pick your own, or remote
# systems might be able to reset your clock at will. Note also that
# ntpd is started with a -A flag, disabling authentication, that
# will have to be removed as well.
#
keys                /etc/ntp/keys
```

```
o /etc/ntp/step-tickers
```

10.0.0.138

In order to automatically start the ntp service at boot time, the following commands can be issued (as root user):

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	23 of 70

```
chkconfig --level 2345 ntpd on
chkconfig --list ntpd
```

The command **ntpq -p** can be used to verify that the ntp daemon works correctly on the local machine:

```
/usr/sbin/ntpq -p
```

The command should print a table with one status line for each reference time source which has been configured for the NTP daemon on the specified host:

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*10.0.0.138	LOCAL(0)	11	u	739	1024	377	1.632	-1.681	0.276

An asterisk * in the first column marks the reference time source which is currently preferred by the NTP daemon

The column **remote** displays the IP address or the host name of the reference time source.

The column **reach** shows if a reference time source could be reached at the last polling intervals, i.e. the **reference time source was synchronized**. The maximum value **377** means that the eight last queries were completed successfully.

3.4 SOFTWARE ORGANIZATION AND OVERVIEW OF OPERATION

3.4.1 PACKAGE OVERVIEW

While CARDAMOM provides a full of services, it has been divided into packages allowing the user to select only its needed services. This selection is done at “configure” step explained in 4.2.2.

Packages are defined as following:

1)CORE_FT -

Foundation, SystemMngt, TraceAndPerf, Repository, LifeCycle, ConfAndPlug, CodeGen, Event, FaultTolerance

2)DEFAULT -

Foundation, SystemMngt, TraceAndPerf, Repository, XMLServer, LifeCycle, ConfAndPlug, CodeGen, Event, CCMContainer, CCMDeployment, Time, LoadBalancing

3)DEFAULT_FT -

Foundation, SystemMngt, TraceAndPerf, Repository, XMLServer, LifeCycle, ConfAndPlug, CodeGen, Event, CCMContainer, CCMDeployment, Time, LoadBalancing, FaultTolerance

Note that packages are not exclusive: the user is able to select several of them.

At “configure” step, only available package on the current platform and with the current release of CARDAMOM are put forward to the user selection.

For each package, several interfaces may be available: C++, Java. It depends on the platform where CARDAMOM is installed and the release of CARDAMOM. This interface language defines the language in which the user application will be developed. Note that all CARDAMOM executables are coded in C++.

Each interface may be available on several ORBs and compiler.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	24 of 70

All these choices are done at “configure” step (ref. to section 4.2.2). For more information on what CARDAMOM functionality is supported by a specific platform, a specific compiler and a specific ORB, refer to VDD [6].

3.4.2 INSTALLATION OVERVIEW

All CARDAMOM source files are delivered with CARDAMOM. The user shall then “configure” CARDAMOM that is select one or several needed packages and needed interface languages (C++, Java): it determines what libraries and executables are to be compiled and installed. For a package, if the C++ and the Java interface are available, they are compiled.

All tools installation paths are also specified at this step. Then the user shall compile CARDAMOM, check that all is OK in running automatic unit test and finally install CARDAMOM in an installation directory specified at “configure” step.

Installation and configure step are well described in section 4.

3.5 CONTINGENCIES AND ALTERNATE STATES AND MODES OF OPERATION

Not Applicable.

3.6 SECURITY, PRIVACY AND INTELLECTUAL PROPERTY PROTECTION

Refer to Collaboration Agreement (ref.[1]).

3.7 ASSISTANCE AND PROBLEM REPORTING

Refer to Software Configuration Management Plan ([8]).

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	25 of 70

4. INSTALLATION AND SETUP

This section describes the procedure that has to be run for downloading, building and installing CARDAMOM Run-time.

In “APPENDIX G: Getting started”, the CARDAMOM installation procedure is summed up with the main steps to configure, compile and install CARDAMOM.

For the first installation and for a better comprehension, please read this section.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	26 of 70

4.1 CD DESCRIPTION¹

4.1.1 SOURCE CD

```

CARDAMOM_VX_Y
|--- Readme.txt
|--- doc
|   |---- glo
|   |---- html
|   |---- pidi
|   |---- prov
|   |---- qtpdr
|   |---- sac
|   |---- scom
|   |---- srs
|   |---- ssdd
|   |---- sss
|   |---- sum
|   |---- vdd
|--- src
|--- tools
|   |---- Ant
|   |---- CppUnit
|   |---- Doxygen
|   |---- JDK
|   |---- JacORB
|   |---- JavaCC
|   |---- Lcov
|   |---- OpenCCM
|   |---- OpenORB
|   |---- R
|   |---- Saxon
|   |---- TAO
|   |---- Xerces
    
```

On the left, there is an example of snapshot of the CARDAMOM CD structure (it may evolve with CARDAMOM versions).

On the CD, there is three directories, the "doc" directory, the "src" directory and the "tools" directory.

- "doc" directory contains:
 - o **GLO** (Glossary): The CARDAMOM Acronyms document, which contains all the abbreviations, used in the CARDAMOM documentation.
 - o **HTML**: contains the html.tar.gz file which contains the html CARDAMOM documentation (class description).
 - o **PIDI** (Program Item Description Index), which contains all tools and documentation version for each CARDAMOM release.
 - o **PROV** (Product Overview): The CARDAMOM overview. This document is referenced in all other documents.
 - o **QTPDR** (Qualification Test Plan/Description/Report), which contains the Qualification Test Plan (QTP), the Qualification Test Description (QTD) and the Qualification Test Result (QTR).
 - o **SAC** (System Segment Specification at Completion), which defines the CARDAMOM product requirements at completion, including functional and non-functional requirements, external interfaces, and design and production constraints applicable to the final CARDAMOM release
 - o **SCOM** (Software Center Operator Manual): This document.
 - o **SRS** (Software Requirements Specification), which contains all the requirements allocated to CARDAMOM. There is one SRS per CSCI.
 - o **SSDD** (System/Segment Design Document): Identification of CSCIs
 - o **SSS** (System/Segment Specification): Product Requirements
 - o **SUM** (Software User Manual), which contains all information necessary to develop an application using CARDAMOM.
 - o **VDD** (Version Description Document), which contains the presentation of the current CARDAMOM version (package, platform, etc...)

- the "src" directory contains one file, named CARDAMOM_Vx_y.tar, with x.y being the version number of the CARDAMOM delivery. Users have to uncompress and untar this file so as to install the CARDAMOM development directory structure with all source files.

¹CD ROM is not delivered in case of CARDAMOM CE (Community Edition); in this case CARDAMOM sources and the needed documentation are downloadable directly from the hosting website. Tools are downloadable from their corresponding websites referenced in the PIDI [3] document.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	27 of 70

- The "tools" directory contains all the tools needed to CARDAMOM installation, compilation and run. Each subdirectory contains a tool archive or a file specifying the address where it is possible to download this archive. It may also contain a patch file if tool needs to be patched.

Tools required by CARDAMOM are listed in PIDI [3]. Note that while CARDAMOM provides a lot of services with many languages and ORB, the user is not intended to install all tools identified in the PIDI.

To know exactly what software shall be installed, please use the command

```
startCdmw.sh needed-tools
```

as explained in section 4.2.2.

In "APPENDIX A: Tools installation Architecture" is suggested a tool directory installation architecture avoiding problems when using them.

"APPENDIX B: Installation of the tools on Linux pc" describes tools installation tips for Linux PC platform.

4.2 INSTALLATION PROCEDURE

Note 1: during the installation procedure CARDAMOM Build commands are performed. They are briefly described in this chapter but **please refer to the CSCI Build SUM [9] document if you need more information.**

Note 2: any user-modification of the source and runtime file architecture provided by CARDAMOM is forbidden. Any observation at this subject from the CARDAMOM support cancels the call support and a complete installation will be necessary. These considerations are also applicable on tools delivery.

4.2.1 PATCHES

Before starting the configuration phase, could be required the application of some patches, to either CARDAMOM CSCIs source code and/or tools source code.

Patch is delivered as a single zip file and includes at least:

- Patch file to apply for each CSCI to be modified (if any)
- Patch file to apply for each tool to be modified (if any)
- ReadMe.txt file providing details on the Patch content and the installation procedure.

4.2.1.1 PATCHING TOOLS

In the following sections:

- *<platform>* is a name which denotes the target platform and is made of the processor type, the manufacturer and the operating system and its version number.

Examples:

i686-pc-linux-gnu

- *<release>* denotes the product (ORB, tools...) release number.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	28 of 70

If the following tools are required to be installed in order to install CARDAMOM (as will be detailed in 7), they have to be patched using the following procedures:

4.2.1.1.1 ORBacus C++

The patch for ORBacus C++ is named: `OB-<release>.patch` where release is the ORBacus C++ release number.

The procedure to perform this patch is the following:

- on the CD go into the ORBacus C++ directory:
`cd xxx/OB-<release>`
- perform the patch by entering the following command:
`cat OB-<release>.patch | patch -p2`

4.2.1.1.2 ORBacus Java

The patch for ORBacus Java is named: `JOB-<release>.patch` where release is the ORBacus Java release number.

The procedure to perform this patch is the following:

- on the CD go into the ORBacus Java directory:
`cd xxx/JOB-<release>`
- perform the patch by entering the following command:
`cat JOB-<release>.patch | patch -p2`

4.2.1.2 PATCHING CARDAMOM

A CSCI patch file is named the following way:

<module name>_patch_<CARDAMOM_Version>[p<Patch Level>]

Where:

- <module name>** :
 - cdmw_<CARDAMOM CSCI Name>** when patch is applicable to a CARDAMOM CSCI. In this case <CARDAMOM CSCI Name> is the 3-char CSCI name as defined in the PID1
- <CARDAMOM_version>**: Version of the product concerned by the patch.
- p<Patch Level>** : <Patch Level> is the product patch level after applying all the patch files. In some specific cases, this field can be omitted.

The procedure to apply patch is the following:

- For instance, for a patch file "cdmw_BMG_patch_V4_0_0p02", copy the patch file into the CARDAMOM root directory, and apply with the command:
`cat cdmw_BMG_patch_V4_0_0p02 | patch -E -p0 --verbose`

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	29 of 70

4.2.2 CONFIGURATION

The PATH must be set so as to reference the directories including the binaries of all compilers (gcc, java)
Then, go into the directory where CARDAMOM sources have been extracted.

To know the exhaustive list of tools to be installed before compiling CARDAMOM, start the configuration script using the following command :

```
./startCdmw.sh needed-tools
```

WARNING : The JDK and ANT² have to be previously installed in order to run the script ./startCdmw.sh (refer to 7.2 and 7.3).

A list of questions is asked like the CARDAMOM package(s) needed and the CARDAMOM interface used (CPP, JAVA,...). When these information are selected, the configure gives the needed tools and some related information (if a patch is needed, if some important compilation options are necessary, etc...).

In "APPENDIX E: needed-tools ", there is a typical result for the previous command.

The Build system provides the capability to:

- obtain a summary of the hardware configuration on the build host,
- check the availability of the hardware resources needed to complete building requests as described in the SSS ([10]) in the requirement REQ-SSS-CNS-0100 (§3.10 Computer Resource Requirements),

for both, by typing the following command:

```
./startCdmw.sh check-host-resources
```

In "APPENDIX F: check-host-resources target Example", there is an example of the possible check-host-resources output.

During the "startCdmw.sh configure" steps, the build system checks only for the availability of enough disk space to complete cardamom compilation and installation (on the partions in which the directories choosen during the "configure" reside). A warning message appear only if the space available on the selected partition is less than:

- 2250 MB for CARDAMOM compilation
- 400 MB for CARDAMOM installation

E.g.:

```
#####
warning: the recommend disk space to complete compile command is about 2250MB
and it is more than 689MB, that is the space available on the selected partition
#####
Do you want to continue? [y]
.....
```

² WARNING: starting from version 1.6.5 a bug affects Ant, while a task is executed in conjunction with nohup command (see "http://issues.apache.org/bugzilla/show_bug.cgi?id=38893" for more deatails). In order to preserve output of a compilation, for example, do the following:
nohup ./startCdmw.sh compile ... </dev/null &

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	30 of 70

When all tools are installed³ and the hardware host resources have been checked, start the configuration script:

```
./startCdmw.sh configure
```

A list of questions are asked, requesting:

- The CARDAMOM Package(s). A package identifies a set of CARDAMOM functionality, which can be built independently. Refer to [6] for the list of CARDAMOM packages available.
- The CARDAMOM interface(s) (CPP, JAVA) to build. CPP is mandatory. The CPP interface allows to build a C++ application running on CARDAMOM. In the same way, the JAVA interfaces (if available) allow to build Java applications running on CARDAMOM.
- The location of all needed tools.
- The location of the directory where to install CARDAMOM is also requested.

In “Appendix C: configure example”, there is a typical configuration output.

The “startCdmw.sh configure” step creates under the <CDMW-SOURCE-PATH>/Build directory a directory named “.CDMW” in which it stores configuration files that define properties used to compile and install CARDAMOM such as the locations of installed tools. These configuration files have the extension “#X.properties”, where “X” is a number incremented at each configuration; on subsequent configurations, the file with the last number can be used to retrieve the appropriate properties. Moreover a user can select a configuration file from which the properties are retrieved for compiling or installing CARDAMOM or eventually for another configuration by selecting the appropriate number, e.g. “startCdmw.sh compile -c 3” where “3” corresponds to the 3rd configuration.

4.2.3 COMPILATION

CARDAMOM requires specific features from the libpthread. On Red Hat Enterprise Linux, it comes in 2 flavours: with or without NPTL. Unfortunately, the libpthread with NPTL support does not implement the required features yet.

Build CARDAMOM by typing the following command:

```
./startCdmw.sh compile
```

4.2.4 TESTING

The CARDAMOM Automatic tests are built at the same time that CARDAMOM itself. To run the tests:

```
./startCdmw.sh runtest
```

At the end of tests, a report is displayed. All tests must return a successful status (OK). This report is produced as textual file under the <CDMW-SOURCE-PATH>/Build directory at each running of the “runtest” command, and it is named in the following way:

```
RunTest-Linux-i686-2.6.9-22.ELsmp#X_<time_stamp>.log
```

where “X” is the configuration number.

It is possible to run a subset of tests by running ./startCdmw.sh runtest with different parameters.

³ Refer to Chapter 7 for an exhaustive and ordered list of tools to be installed in case of full configuration of CARDAMOM:

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	31 of 70

The list of all available parameters is displayed by `./startCdmw.sh` without any parameter.

In “APPENDIX D: Typical Report of the runtest”, there is a typical report output.

WARNING : Depending on the environment some tests may fail due to the timing of test execution. Please refer to CSCI Build SUM [9] document on the use of “timescale” option in launching tests.

4.2.5 INSTALLATION

Once CARDAMOM Run-time has been built and tested, CARDAMOM can be installed in the user-defined directory specified at “configure” step.

Starting the installation phase is performed by running the following command:

```
./startCdmw.sh install
```

Installed directories and files are described in 3.2.

After the “install” step, the directory where CARDAMOM is installed has to be copied on all nodes where CARDAMOM must be running. The absolute path of this directory has to be the same on each node.

4.2.5.1 INSTALLATION OF BINARY DELIVERY

This installation procedure allow to compile CARDAMOM on a host and to install it on another host and in a directory established by the user.

The binary installation is achieved by typing the following commands:

```
./startCdmw.sh install-binary
```

This command create a compressed archive, named CDMW-<CARDAMOM version>.tar.gz (where <CARDAMOM version> may also contain the patch level), containing the binary version of CARDAMOM. To move the CDMW-<CARDAMOM version>.tar.gz in the directory in which install the binary release:

```
mv CDMW-<CARDAMOM version>.tar.gz <CDMW-INSTALLATION-PATH>
(where <CDMW-INSTALLATION-PATH> is the chosen installation directory)
```

To uncompress and untar this file:

```
tar xzf CDMW-<CARDAMOM version>.tar.gz
```

After typing this command the “binary” directory, named CDMW-<CARDAMOM version>, is created at the installation directory level. The content of this directory is the following:

- delivery: contains the CARDAMOM installation
- tools: contains the tools needed by the CARDAMOM installation
- configure.sh: command to configure the CARDAMOM installation

To configure the CDMW install, execute the extracted script configure.sh inside the CDMW-<CARDAMOM version> directory:

```
cd CDMW-<CARDAMOM version>
configure.sh
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	32 of 70

An example of the configure.sh output, is the following:

```
=====
CDMW installation tool

This tool will guide you along the installation

First part
-----
CDMW will check if all correct versions have been installed and will
prompt you in case of error

Second part
-----

CDMW will configure itself
=====

Check of tools installation in accordance with the configuration selected
-----
Check Java version 1.5.0_12
Test OK

Customization of source files in accordance with selected configuration
-----
/tmp/CDMW-4.0/delivery/bin/cdmw_package_manager.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_idl_generator.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_start_ifr_service.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_server_activator.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_code_generator.sh
/tmp/CDMW-4.0/delivery/bin/jvm.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_event_channel_manager.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_deployment_tool.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_platform_daemon.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_export.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_component_installation.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_event_admin.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_assembly_factory.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_start_ifr_feed.sh
/tmp/CDMW-4.0/delivery/bin/envi.OpenCCM.sh
/tmp/CDMW-4.0/delivery/bin/cdmw_xml_server.sh
/tmp/CDMW-4.0/delivery/demos/demo_state_transfer_no_memory_leak/c++/site.mk
/tmp/CDMW-4.0/delivery/demos/demo_perf/java/site.mk
/tmp/CDMW-4.0/delivery/demos/demo_perf/c++/site.mk
/tmp/CDMW-4.0/delivery/demos/federated_controlled_executor/c++/site.mk
/tmp/CDMW-4.0/delivery/demos/demo_crash/c++/site.mk
/tmp/CDMW-4.0/delivery/demos/demo_state_transfer/c++/site.mk
/tmp/CDMW-4.0/delivery/demos/demo_ft/c++/site.mk
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	33 of 70

```

/tmp/CDMW-4.0/delivery/demos/demo/java/site.mk
/tmp/CDMW-4.0/delivery/demos/demo/c++/site.mk
/tmp/CDMW-4.0/tools/OpenCCM-0.4p8/bin/envi.OpenCCM.csh
/tmp/CDMW-4.0/tools/OpenCCM-0.4p8/bin/jvm.sh
/tmp/CDMW-4.0/tools/OpenCCM-0.4p8/bin/envi.OpenCCM.sh
    
```

To compile and run CDMW Application, export the following variable

```

export PATH=/tmp/CDMW-4.0/delivery/bin:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/CIAO/DAnCE/Starter:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/orbsvcs/Naming_Service:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/TAO_IDL:
/tmp/CDMW-4.0/tools/cppunit-1.10.2/bin:
/tmp/CDMW-4.0/tools/jdk1.5.0_12/bin:
/tmp/CDMW-4.0/tools/OpenCCM-0.4p8/bin:
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/tools/exec/source
nav/bin:./tools/exec/R-2.3.1/bin:/tools/exec/cppunit-
1.10.2/bin:/home11/fmarchigiani/bin
export LD_LIBRARY_PATH=/tmp/CDMW-4.0/delivery/lib:
/tmp/CDMW-4.0/tools/jdk1.5.0_12/lib:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/TAO_IDL:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/tao:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/orbsvcs/orbsvcs:
/tmp/CDMW-4.0/tools/TA0151_1-FT/src/lib/./ace:
/tmp/CDMW-4.0/tools/cppunit-1.10.2/lib/bin:
/tmp/CDMW-4.0/tools/xerces-c_2_7_0/lib:/tools/exec/OB-4.1.0/lib:
export CLASSPATH=/tmp/CDMW-4.0/delivery/lib/java/libcdmweventinterface.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcoseventidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwosupport.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwxmlserverlib.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwtestutils.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmweventidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwpullmonitorableidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmdeploymentutil.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwplatforminterface.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwftinit.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmcomponentserver.jar:
.....
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmcif.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmassembly.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmserveractivator.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmcommon.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwxmllibrary.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcommon.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcosnotificationidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwperformancelibrary.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcomponentidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwsystemmngtidl.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/wrapper-3.1.0.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/avalon-framework-4.1.5.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/idl.jar:
    
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	34 of 70

```
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/jacorb.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/logkit-1.2.jar:
```

The CDMW install is now configured
You can built demos

4.2.5.2 ENVIRONMENT VARIABLES SETTING

The installation process will generate, under the installation path, into the "bin" directory, a shell script file "cdmw_config.sh". This script can ease developing CARDAMOM applications as it provides a complete list of compilation support options for both TAO-OpenFusion and JacORB-OpenFusion as needed by CARDAMOM, All the displayed information are extracted from the Build System during the CARDAMOM configuration process. The same properties are stored into the file "Site-xxx.properties", generated under the <CDMW-SOURCE-PATH>/Build directory into the ".CDMW" directory.

Figure 4-1 shows the options provided by the "cdmw_config.sh" script:

```
Usage: cdmw_config.sh [OPTION] ...

Generic options:
--version      output Cardamom version information.
--help         display this help and exit.

Compilation support options:
--home/--prefix      print CDMW installation folder
--cpp-orb             print the cpp orb
--java-orb            print the java orb
--cflags              print C preprocessor and compiler flags
--libs               print library linking information
    --cdmw-libs       print Cardamom specific library
    --tao-libs        print ACE/TAO specific library
    --libs-only       print library to pass to linker
--java-libs           print java class libraries information
    --cdmw-java-libs  print Cardamom specific java class libraries information
    --orb-java-libs   print ORB specific java class libraries information
--cpp-idlflags        print CPP idl compiler flags
--java-idlflags       print JAVA idl compiler flags
```

Figure 4-1 cdmw_config.sh options.

Figure 4-2 shows an example of the output provided by the "cdmw_config.sh" script (this list is not exhaustive)

```
cdmw_config.sh --cpp-orb
TA00openFusion1.5.1

cdmw_config.sh --java-orb
JacORB2.3
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	35 of 70

```
./cdmw_config.sh --cflags
-I/tmp/CDMW-4.0/delivery/include/c++/CCMContainer
-I/tmp/CDMW-4.0/delivery/include/c++/CCMDeployment
-I/tmp/CDMW-4.0/delivery/include/c++/CodeGen
-I/tmp/CDMW-4.0/delivery/include/c++/ConfAndPlug
-I/tmp/CDMW-4.0/delivery/include/c++/Event
-I/tmp/CDMW-4.0/delivery/include/c++/FaultTolerance
-I/tmp/CDMW-4.0/delivery/include/c++/Foundation
-I/tmp/CDMW-4.0/delivery/include/c++/LifeCycle
-I/tmp/CDMW-4.0/delivery/include/c++/LoadBalancing
-I/tmp/CDMW-4.0/delivery/include/c++/ProdTests
-I/tmp/CDMW-4.0/delivery/include/c++/Repository
-I/tmp/CDMW-4.0/delivery/include/c++/SystemMngt
-I/tmp/CDMW-4.0/delivery/include/c++/Time
-I/tmp/CDMW-4.0/delivery/include/c++/TraceAndPerf
-I/tmp/CDMW-4.0/delivery/include/c++/XMLServer
-I/tmp/CDMW-4.0/delivery/include/c++/TA0OpenFusion
-I/tmp/CDMW-4.0/delivery/include/c++
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/orbsvcs/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/tao/IFR_Client
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/orbsvcs/orbsvcs/IFRService
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/tao/PI
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TA0/tao/PI_Server
-I/tmp/CDMW-4.0/tools/xerces-c_2_7_0/include
-I/tmp/CDMW-4.0/tools/cppunit-1.10.2/include
-Dtao=4203 -DCDMW_ORB_VDR=4203 -DCDMW_ORB_VER=15 -DACE_HAS_EXCEPTIONS
-D_POSIX_THREADS -D_POSIX_THREAD_SAFE_FUNCTIONS -D_REENTRANT
-DACE_HAS_AIO_CALLS -DTAO_HAS_VALUETYPE -DTAO_USE_SEQUENCE_TEMPLATES
-DTAO_HAS_EXTENDED_FT_INTERCEPTORS

cdmw_config.sh --cdmw-libs
-L/tmp/CDMW-4.0/delivery/lib/c++ -lcdmwgroupmanageridl
-lcdmwftultimatefallback -lcdmwftcommon -lcdmwrecordingidl -lcdmwjniOSSupport
-lcdmwcosnotificationidl -lcdmwtracelibrary -lcdmwcdmwinit -lcdmwlifecycle
-lcdmwccmcif -lcdmwcommonsvcsnaming -lcdmwmonitoringidl -lcdmwdaemonidl
-lcdmwplatformvaluetypes -lcdmwlbstrategy
-lcdmwNaming_and_repository_proxy_lib -lcdmwccmcomponentserver
.....
-lcdmwlbgroupcreator -lcdmwftgroupcreator -lcdmwprocessadmintools
-lcdmwfaulttoleranceidl -lcdmwftinit -lcdmwrecordingidlsrv -lcdmwftwatchdog
-lcdmwsmgtools -lcdmwlbcommon -lcdmwperformancelibrary -lcdmwlifecycleidl
-lcdmwccmdeploymentidl -lcdmwevotidl -lcdmwloadbalancingidl
-lcdmwrepositoryinterface -lcdmwcommonsvcsserviceRegistration

cdmw_config.sh --tao-libs
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	36 of 70

```
-L/tmp/CDMW-4.0/tools/TA0151_1-FT/src/lib -lACE -lTAO -lTAO_PortableGroup -
lTAO_PortableServer -lTAO_CosNaming -lTAO_CosNaming_Skel -lTAO_CosNaming_Serv
-lTAO_Messaging -lTAO_IORTable -lTAO_IFR_Client -lDAnCE_Deployment_stub
-lDAnCE_Plan_Launcher_Base -lDAnCE_ExecutionManager_stub -lTAO_CosEvent_Serv
-lTAO_IORManip -lTAO_FT_ClientORB -lTAO_PI_Server -lTAO_DynamicInterface
-lTAO_CosNotification -luuid -lTAO_CosProperty_Serv
```

```
cdmw_config.sh --libs-only
-lcdmwgroupmanageridl -lcdmwftultimatefallback -lcdmwftcommon
-lcdmwrecordingidl -lcdmwjniiossupport -lcdmwcosnotificationidl
-lcdmwtracelibrary -lcdmwcdmwinit -lcdmwlifecycle
-lcdmwccmcif -lcdmwcommonsvcsnaming -lcdmwmonitoringidl -lcdmwdaemonidl
-lcdmwplatformvaluetypes -lcdmwlbstrategy lcdmw_naming_and_repository_proxy_lib
-lcdmwccmcomponentserver -lcdmworbupport -lcdmwosthreads -lcdmweventinterface
-lcdmwftccmcif -lcdmwcommon -lcdmwpullmonitorableidl -lcdmwsupervisioncommon
.....
-lcdmwrepositoryinterface -lcdmwcommonsvcsserviceregistration -lACE -lTAO
-lTAO_PortableGroup -lTAO_PortableServer -lTAO_CosNaming -lTAO_CosNaming_Skel
-lTAO_CosNaming_Serv -lTAO_Messaging -lTAO_IORTable -lTAO_IFR_Client
-lDAnCE_Deployment_stub -lDAnCE_Plan_Launcher_Base
-lDAnCE_ExecutionManager_stub -lTAO_CosEvent_Serv -lTAO_IORManip
-lTAO_FT_ClientORB -lTAO_PI_Server -lTAO_DynamicInterface
-lTAO_CosNotification -luuid -lTAO_CosProperty_Serv
```

```
cdmw_config.sh --cdmw-java-libs
/tmp/CDMW-4.0/delivery/lib/java/libcdmweventinterface.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcoeventidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwosupport.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwxmlserverlib.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwtestutils.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmweventidl.jar:
.....
/tmp/CDMW-4.0/delivery/lib/java/libcdmwccmcommon.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwxmllibrary.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcommon.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcosnotificationidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwperformancelibrary.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwcomponentidl.jar:
/tmp/CDMW-4.0/delivery/lib/java/libcdmwsystemmngtidl.jar
```

```
cdmw_config.sh --orb-java-libs
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/wrapper-3.1.0.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/avalon-framework-4.1.5.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/idl.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/jacorb.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/logkit-1.2.jar
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	37 of 70

```
cdmw_config.sh --cpp-idlflags
-DTAO -Ge 0 -Sc -I/tmp/CDMW-4.0/delivery/idl
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/TAO_IDL/idl_specs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao/IFR_Client

cdmw_config.sh --java-idlflags
-I/tmp/Jacorb-OF-2.3_03/src/idl -I/tmp/Jacorb-OF-2.3_03/src/idl/omg
```

Figure 4-2 cdmw_config.sh usage example

The CARDAMOM installation process, will also generate a set of configuration files, that can be included in Makefiles of both C++ and Java user applications.

All these config files are delivered under the installation path, into the "share/tutorialconfigs" directory.

These generated include files are:

- site.mk: that contains the appropriate environmental variables for the tools needed by CARDAMOM
- cdmw.mk: that contains compilation flag as needed by CARDAMOM
- cpp-orb.mk: that contains compilation flag for the supported cpp orb
- java-orb.mk: that contains compilation flag for the supported java orb
- os.mk: that contains compilation flag for the supported target host

Figure 4-3 shows an example of these generated include files.

```
.....
cdmw.mk
.....
# -----
# Compilation flag for CDMW
# -----
cdmw_libs:=$(shell echo " libcdmwgroupmanageridl.so
libcdmwTestNamingInterface.so libcdmwftultimatefallback.so libcdmwftcommon.so
libcdmwrecordingidl.so libcdmwQualifCcmLbClientFT.so libcdmwjniiosupport.so
libcdmwcosnotificationidl.so libcdmwtracelibrary.so
libcdmwValidCcmFtWithTimerClient.so libcdmwcdmwinit.so libcdmwlifecycle.so
libcdmwccmcif.so libcdmwcommonsvcsnaming.so libcdmwQualifCcmLbClient.so
libcdmwmonitoringidl.so libcdmwdaemonidl.so libcdmwplatformvaluetypes.so
libcdmwlbstrategy.so libcdmw naming_and_repository_proxy_lib.so
.....
libcdmwlbgroupcreator.so libcdmwftgroupcreator.so libcdmwprocessadmintools.so
libcdmwfaulttoleranceidl.so libcdmwQualifCcmLbServer.so libcdmwftinit.so
libcdmwrecordingidlsrv.so libcdmwftwatchdog.so libcdmwsmgtools.so
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	38 of 70

```

libcdmwlbcommon.so libcdmwperformancelibrary.so libcdmwlifecycleidl.so
libcdmwccmdeploymentidl.so libcdmwevotidl.so libcdmwloadbalancingidl.so
libcdmwrepositoryinterface.so libcdmwcommonsvcsserviceregistration.so" | sed -
e 's/ lib/ -l/g' | sed -e 's/\.so//g')

#BEGIN CDMW FLAGS
CPPFLAGS_CDMW= " -I/tmp/CDMW-4.0/delivery/include/c++/CCMContainer
-I/tmp/CDMW-4.0/delivery/include/c++/CCMDeployment
-I/tmp/CDMW-4.0/delivery/include/c++/CodeGen
-I/tmp/CDMW-4.0/delivery/include/c++/ConfAndPlug
-I/tmp/CDMW-4.0/delivery/include/c++/Event
-I/tmp/CDMW-4.0/delivery/include/c++/FaultTolerance
-I/tmp/CDMW-4.0/delivery/include/c++/Foundation
-I/tmp/CDMW-4.0/delivery/include/c++/LifeCycle
-I/tmp/CDMW-4.0/delivery/include/c++/LoadBalancing
-I/tmp/CDMW-4.0/delivery/include/c++/ProdTests
-I/tmp/CDMW-4.0/delivery/include/c++/Repository
-I/tmp/CDMW-4.0/delivery/include/c++/SystemMngt
-I/tmp/CDMW-4.0/delivery/include/c++/Time
-I/tmp/CDMW-4.0/delivery/include/c++/TraceAndPerf
-I/tmp/CDMW-4.0/delivery/include/c++/XMLServer
-I/tmp/CDMW-4.0/delivery/include/c++/TA0OpenFusion
-I/tmp/CDMW-4.0/delivery/include/c++"

CXXFLAGS_CDMW=
LDFLAGS_CDMW= "-L/tmp/CDMW-4.0/delivery/lib/c++"
IDLFLAGS_CDMW= "-I/tmp/CDMW-4.0/delivery/idl"
LIBS_CDMW= $(cdmw_libs)

#END CDMW FLAGS
# -----

:~::~:
cpp-orb.mk
:~::~:
# -----
# Compilation flag for TA0OpenFusion1.5.1 as needed by CDMW
# -----
CPPFLAGS_ORB= "-I/tmp/CDMW-4.0/tools/TA0151_1-FT-070926p1/src
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/orbsvcs/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao/IFR_Client
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/orbsvcs/orbsvcs/IFRService
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao/PI
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao/PI_Server" \
"-Dtao=4203 -DCDMW_ORB_VDR=4203 -DCDMW_ORB_VER=15 -DACE_HAS_EXCEPTIONS
-D_POSIX_THREADS -D_POSIX_THREAD_SAFE_FUNCTIONS -D_REENTRANT
-DACE_HAS_AIO_CALLS -DTAO_HAS_VALUETYPE -DTAO_USE_SEQUENCE_TEMPLATES
-DTAO_HAS_EXTENDED_FT_INTERCEPTORS"
CXXFLAGS_ORB=

```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	39 of 70

```
IDLFLAGS_ORB= "-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/TAO_IDL/idl_specs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/orbsvcs
-I/tmp/CDMW-4.0/tools/TA0151_1-FT/src/TAO/tao/IFR_Client"
LDLFLAGS_ORB= "-L/tmp/CDMW-4.0/tools/TA0151_1-FT/src/lib "
LIBS_ORB= " -lACE -lTAO -lTAO_PortableGroup -lTAO_PortableServer -
lTAO_CosNaming -lTAO_CosNaming_Skel -lTAO_CosNaming_Serv -
lTAO_Messaging -lTAO_IORTable -lTAO_IFR_Client -
lDAnCE_Deployment_stub -lDAnCE_Plan_Launcher_Base -
lDAnCE_ExecutionManager_stub -lTAO_CosEvent_Serv -lTAO_IORManip -
lTAO_FT_ClientORB -lTAO_PI_Server -lTAO_DynamicInterface -
lTAO_CosNotification -luuid -lTAO_CosProperty_Serv"
IDL2CXX_ORB= -DTAO -Ge 0 -Sc
# -----

:~::~:~::~:
java-orb.mk
:~::~:~::~:
# -----
# Compilation flag for JacORB2.3 as needed by CDMW
# -----
JAVA_IDL_COMPILER= /tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/bin/idl
JIDL_CLT=$(JAVA_IDL_COMPILER)
JIDL_SVR=$(JAVA_IDL_COMPILER)

JIDL_FLAGS_CLT= -noskel
JIDL_FLAGS_SVR=
JIDL_FLAGS_TIE=

JIDL_INCLUDE_DIR_OPTION:=-I
JIDLFLAGS_ORB= -I/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/idl
-I/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/idl/omg
ORB_CLASSPATH= /tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib/build/xdoclet-web-
module-1.2.3.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/commons-collections-2.0.jar:/tmp/CDMW-
4.0/tools/Jacorb-OF-2.3_02/src/lib/build/xdoclet-ejb-module-
1.2.3.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/javacc-4.0/bin/lib/javacc.jar:/tmp/CDMW-
4.0/tools/Jacorb-OF-2.3_02/src/lib/build/xdoclet-jboss-module-
1.2.3.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/xjavadoc-1.1.jar:/tmp/CDMW-
4.0/tools/Jacorb-OF-2.3_02/src/lib/build/xdoclet-mx4j-module-
1.2.3.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/xdoclet-jmx-module-1.2.3.jar:/tmp/CDMW-
4.0/tools/Jacorb-OF-2.3_02/src/lib/build/xdoclet-1.2.3.jar:
/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/java_cup.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/build/commons-logging.jar:/tmp/CDMW-
4.0/tools/Jacorb-OF-2.3_02/src/lib/wrapper-3.1.0.jar:/tmp/CDMW-
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	40 of 70

```

4.0/tools/Jacorb-OF-2.3_02/src/lib/avalon-framework-
4.1.5.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/idl.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/jacorb.jar:/tmp/CDMW-4.0/tools/Jacorb-OF-
2.3_02/src/lib/logkit-1.2.jar
# -----

:
:
:
os.mk
:
:
# -----
# Compilation flag for each supported target
# -----
# We determine the compilation OS
HOST_TYPE:=$(shell $(CDMW_HOME)/bin/config.guess)
HOST_SUFFIX:=$(shell echo $(HOST_TYPE) | cut -f 3 -d"-" | tr "." "_")

# Solaris 2.7
# .....
CXXFLAGS_solaris2_7=-g -Wall -Werror
CPPFLAGS_solaris2_7= -I$(STLPORT_INC_PATH)/stlport
LDFLAGS_solaris2_7= -L$(STLPORT_LIB_PATH) \
    -lpthread -lrt -lsocket -lnsl -lstlport_gcc_debug
    -lstlport_gcc_stddebug
IDLFLAGS_solaris2_7=
LIBS_solaris2_7=

# Solaris 2.8
# .....
CXXFLAGS_solaris2_8=-g -Wall -Werror
CPPFLAGS_solaris2_8= -I$(STLPORT_INC_PATH)/stlport
LDFLAGS_solaris2_8= -L$(STLPORT_LIB_PATH) \
    -lpthread -lrt -lsocket -lnsl -lstlport_gcc_debug -
    lstlport_gcc_stddebug
IDLFLAGS_solaris2_8=
LIBS_solaris2_8=

# Linux
# .....
CXXFLAGS_linux= -g -Wall -Werror -D_REENTRANT -DCDMW_ASSERT_NO_THROW -
DCDMW_POSIX -Dlinux
LDFLAGS_linux=
IDLFLAGS_linux=
LIBS_linux=-lpthread -ldl -rdynamic
# -----

:
:
:
site.mk
:
:

```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	41 of 70

```
# CDMW installation folder
CDMW_HOME=/opt/cardamom/fmarchigiani/55401/installation

# Xerces C++
XERCES_INC_PATH=/tmp/CDMW-4.0/tools/xerces-c_2_7_0
XERCES_LIB_PATH=/tmp/CDMW-4.0/tools/xerces-c_2_7_0/lib

# Saxon
SAXON_JAR_PATH=/tmp/CDMW-4.0/tools/saxon-6.5.3

# OpenCCM
OPENCCM_BIN_PATH=/tmp/CDMW-4.0/tools/OpenCCM-0.4p8/bin

# C++ ORB
ifeq (TA0,TA0)
ORB_IDL_PATH=$(ACE_ROOT)/TA0/orbsvcs/orbsvcs
ORB_LIB_PATH=$(ACE_ROOT)/ace
ORB_CPP=tao14-OF
ORB_CPP_NAME=tao
endif
ifeq (TA0openFusion,TA0)
ORB_IDL_PATH=$(ACE_ROOT)/TA0/orbsvcs/orbsvcs
ORB_LIB_PATH=$(ACE_ROOT)/ace
ORB=tao13
endif
ifeq (TA0openFusion,TA0openFusion)
ifeq (1.5.1,1.4.1)
ORB_IDL_PATH=$(ACE_ROOT)/TA0/orbsvcs/orbsvcs
ORB_LIB_PATH=$(ACE_ROOT)/ace
ORB=tao14-OF
endif
ifeq (1.5.1,1.5.1)
ORB_IDL_PATH=$(ACE_ROOT)/TA0/orbsvcs/orbsvcs
ORB_LIB_PATH=$(ACE_ROOT)/ace
ORB=tao15-OF
endif
endif
ifeq (TA0openFusion,ORBacusCPP)
ORBACUS_IDL_PATH=@config.ORBacusCPP.idl.dir@
ORBACUS_BIN_PATH=@config.ORBacusCPP.bin.dir@
ORBACUS_LIB_PATH=@config.ORBacusCPP.lib.dir@
ORBACUS_INC_PATH=@config.ORBacusCPP.include.dir@
ORB=ORBacus4
endif

# JAVA ORB
ifeq (JacORB,ORBacusJAVA)
ORBACUS_JAVA_IDL_PATH=@config.ORBacusJAVA.idl.dir@
ORBACUS_JAVA_JAR_PATH=@config.ORBacusJAVA.lib.dir@
ORBACUS_JAVA_BIN_PATH=@config.ORBacusJAVA.bin.dir@
ORB_JAVA=ORBacus410
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	42 of 70

```
endif
ifeq (JacORB, JacORB)
  ifeq (2.3, 2.0)
    JACORB_IDL_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/idl
    JACORB_JAR_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib
    JACORB_BIN_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/bin
    TAO_IDL_PATH=$(ACE_ROOT)/TAO/orbsvcs/orbsvcs
    ORB_JAVA=JacORB20
  endif
  ifeq (2.3, 2.1)
    JACORB_IDL_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/idl
    JACORB_JAR_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib
    JACORB_BIN_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/bin
    TAO_IDL_PATH=$(ACE_ROOT)/TAO/orbsvcs/orbsvcs
    ORB_JAVA=JacORB21
  endif
  ifeq (2.3, 2.3)
    JACORB_IDL_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/idl
    JACORB_JAR_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/lib
    JACORB_BIN_PATH=/tmp/CDMW-4.0/tools/Jacorb-OF-2.3_02/src/bin
    TAO_IDL_PATH=$(ACE_ROOT)/TAO/orbsvcs/orbsvcs
    ORB_JAVA=JacORB23
  endif
endif
endif
```

Figure 4-3 generated include files example

4.3 CLEANUP

The cleanup is achieved by typing the following command:

```
./startCdmw.sh distclean
```

This command performs the destruction of all files produced by the compile operation. This leads to have the directories in the state that follows CARDAMOM software unzipping. If you need more details on this part (for example if you want to know about the “clean” target compared to “distclean” target), please refer to CSCI Build SUM [9] document.

NOTE 1: if you want to recompile CARDAMOM with a new configuration simply go again to step 4.2.2

NOTE 2: if you want to recompile CARDAMOM with your previous configuration you can also go to step 4.2.2 but a quicker way, that avoid a new configure + creation of a new configuration, is the following :

```
./startCdmw.sh reconfigure
rm <CDMW-SOURCE-PATH>/Build/.ccache/cc#x.properties (where x is the
configuration number corresponding to the configuration you want to recompile)
```

and then go to step 4.2.3

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	43 of 70

4.4 MANAGEMENT OF DIFFERENT VERSION OF CARDAMOM

In the case, if the user want to use different version of CARDAMOM in the same time, the Build provides some options to easiness the management of its.

These options are the following:

-f		-cfgfile CFG_FILE	use a specific configuration file
-c		-cfgnum CFG_NUM	use a specific configuration number

In this paragraph, the -c option can be replace by one of the previous options.

When the user performs a “configure”, the Build give a number at the end of this “configure”. This number is unique and it is the next number since the last configuration.

4.4.1 CONFIGURATION LISTING

In order to find all the configuration already performed by the Build, typing the following command:

```
./startCdmw.sh configsum
```

The following list appears

```
.....
[configuration #50]
date      = 29-June-2006 15:46:00
host      = c3po / Linux-i686-2.4.21-15.ELsmp
packages  = CORE_FT DEFAULT DEFAULT_FT
interfaces = C++ JAVA
compilers  = GCC J2SDK
orbs      = JacORB TA00openFusion
version   = 2.1
Installation directories
            /<CDMW-SOURCE-PATH>/CCMContainer
            /<CDMW-SOURCE-PATH>/CCMDeployment
            /<CDMW-SOURCE-PATH>/CodeGen
            /<CDMW-SOURCE-PATH>/ConfAndPlug
            /<CDMW-SOURCE-PATH>/Event
            /<CDMW-SOURCE-PATH>/FaultTolerance
            /<CDMW-SOURCE-PATH>/Foundation
            /<CDMW-SOURCE-PATH>/LifeCycle
            /<CDMW-SOURCE-PATH>/LoadBalancing
            /<CDMW-SOURCE-PATH>/ProdTests
            /<CDMW-SOURCE-PATH>/Repository
            /<CDMW-SOURCE-PATH>/SystemMngt
            /<CDMW-SOURCE-PATH>/Time
            /<CDMW-SOURCE-PATH>/TraceAndPerf
            /<CDMW-SOURCE-PATH>/XMLServer
.....
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	44 of 70

The previous list contains

- The configuration number
- The date of the configure
- The host used (name and OS)
- The packages selected
- The interfaces selected
- The compilers selected
- The orbs selected
- The version of CARDAMOM
- The installation directory of each CSCI used for this configuration.

4.4.2 RE-USE OF OLD CONFIGURATION

All the ./startCdmw.sh commands supports the -c option. This option allows using a specific option if the wanted configuration is not the last.

- “-c” option with “configure” :
When the “configure” is performed with the “-c” option, the default values proposed are the values content into the configuration file of this configuration. The “configure” create a new configuration file and the number of this configuration is a new one (the last configuration number presents into the configuration directory plus one).
- “-c” option with “compile” :
When the “compile” is performed with the “-c” option, the compilation used the configuration file of this configuration.
- “-c” option with “runtest” :
Same like “compile”.
- “-c” option with “install” :
Same like “compile”.

To use an old configuration, use the “-c” option with the good configuration number. To find the good configuration number, see the paragraph 4.4.1.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	45 of 70

5. DESCRIPTION OF RUNS

All information about developing application using CARDAMOM and running them is explained in SUM (ref.[7]).

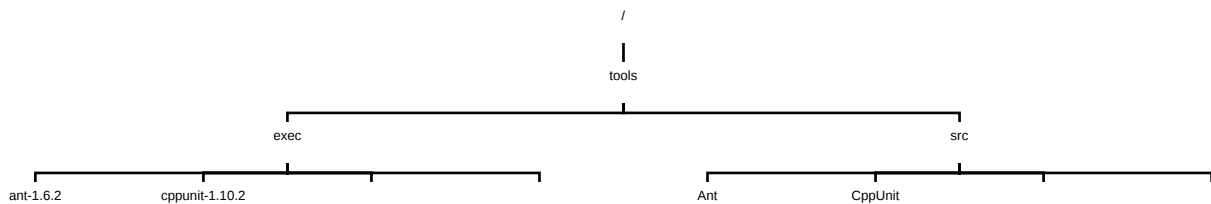
CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	46 of 70

6. APPENDIX A: TOOLS INSTALLATION ARCHITECTURE

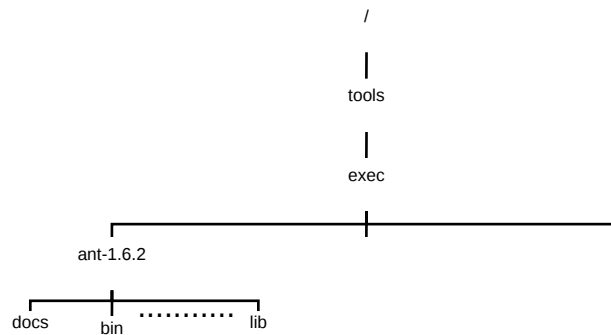
To avoid unfortunate experience, we suggest you an architecture to install all tools needed by CARDAMOM.

The installation of the tools appear like:

- A first directory named “tools”
- In this directory, a directory “src” and a directory “exec”.
 - “src”: contains all the sources of the tools
 - “exec”: contains all the binary files generated by the compilation of the tools sources.



- In directory “exec” and “src” each tool is stored in its own directory.
- In directory “exec” tool’ directory contain a target host dependant directory.



WARNING : CARDAMOM has been validated against the specified versions of tools (refer to document PIDI [3]); nevertheless different versions can be used but in this case the behaviour of CARDAMOM cannot be assured.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	47 of 70

7. APPENDIX B: INSTALLATION OF THE TOOLS ON LINUX PC

Hereafter is an exhaustive list of tools to be installed in case of full configuration of CARDAMOM.

WARNING: the installation procedures showed hereafter for each tool, can contain tool versions that differ from the required with the current CARDAMOM version. To know the appropriate version to install, refer to document PIDI [3].

Note 1: Some of the tools, the installation procedure of which, is presented below, are not needed by CARDAMOM. Some of them are maintained for compliance with old CARDAMOM releases (e.g. ORBacusCPP or SpliceDDS) or are presented for the sole purpose of showing which tools can be used to extract useful information about CARDAMOM (e.g. CARDAMOM includes support for Valgrind to detect memory error).

To determine which are the mandatory tools that are required to be installed, according to the CARDAMOM package chosen (section 3.4.1), refer to the list, returned by “startCdmw.sh needed-tools”. The command “startCdmw.sh needed-tools” is available after the installation of ANT and JDK.

Note 2: Some tools assume that some others are already installed, therefore the tool installation order must be respected as described in the following table.. For the same order level the tools may be installed in parallel. The TAO installation taking a lot of time, we suggest installing it at sooner⁴.

Order					Tools
1	2	3	4	5	
					GCC (already installed), refer to 7.1
X					JDK, refer to 7.2
	X				ANT, refer to 7.3
		X			Launch “startCdmw.sh needed-tools”
				X	OpenFusion TAO 1.5.1.*, refer to 7.4
			X		OpenFusion JACORB, refer to 7.5
			X		OPEN ORB, refer to 7.6
			X		JAVACC, refer to 7.7
				X	OPENCCM, refer to 7.8
			X		XERCES C++, refer to 7.9
			X		SAXON, refer to 7.10
			X		Cppunit, refer to 7.13
			X		Junit, refer to 7.14

Table 7-1 tools installation order

After tools installations are performed, the user-defined directories specified at “./startCdmw configure” step must be duplicated on all nodes where CARDAMOM will be run.

⁴ Tools not listed in the table, can be installed independently of each other.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	48 of 70

7.1 GCC 3.4.4 INSTALLATION

The compilation of GCC is no longer required. Install it from the Red Hat Enterprise Linux installation CDs.

To manually check you have the correct version (CARDAMOM configure does it automatically) type :

```
/usr/bin/gcc -v
```

7.2 JDK INSTALLATION

The steps to install JDK are the followings:

Copy the delivered self-extracting binary file (jdk-1_5_0_<version>-linux-i586.bin) into the location where you would like the JDK files to be installed:

```
cp jdk-1_5_0_<version>-linux-i586.bin /tools/exec/
```

Change directory to the choosen location:

```
cd /tools/exec/
```

Make sure that execute permissions are set on the self-extracting binary. Run this command:

```
chmod +x jdk-1_5_0_<version>-linux-i586.bin
```

The next step installs the JDK into the current directory. Run the self-extracting binary.

Execute the self-extracting binary file, prepended by the path to it. For example, if the file is in the current directory, prepend it with "." (necessary if "." is not in the PATH environment variable):

```
./jdk-1_5_0_<version>-linux-i586.bin
```

The binary code license is displayed, and you are prompted to agree to its terms.

The JDK files are installed in the:

```
/tools/exec/jdk-1_5_0_<version>  
directory.
```

After having installed JDK, add the following environment variable:

```
export JAVA_HOME=/tools/exec/jdk1.5.0_<version>  
export PATH=$JAVA_HOME/bin:/usr/bin:$PATH  
export LD_LIBRARY_PATH=$JAVA_HOME/lib:$LD_LIBRARY_PATH
```

7.3 ANT INSTALLATION

Pre-condition: JDK is installed.

The steps to install Ant are the followings:

```
tar -C /tools/exec -jxf apache-ant-<version>-bin.tar.bz2  
tar -C /tools/exec/apache-ant-<version> -jxf ant-contrib-<version>-bin.tar.bz2  
mv /tools/exec/apache-ant-<version> /tools/exec/ant-<version>
```

To complete installation set the following environment variables:

```
export ANT_HOME=/tools/exec/ant-<version>
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	49 of 70

```
export PATH=$ANT_HOME/bin:/bin:$PATH
for i in `find $ANT_HOME -name "*.jar"`; do export CLASSPATH=$CLASSPATH:$i; done
```

Post-installation procedure:

From the command line, run:

```
ant -version
```

You should read a message like the following:

Apache Ant version 1.6.2 compiled on July 16 2004.

If a different message (from an older or from a different version of ANT) is printed on your console, check your environment:

If any of the files /etc/ant.conf, \$HOME/.ant/ant.conf or \$HOME/.antrc exist then rename it with a .BAK suffix.

7.4 OPENFUSION TAO INSTALLATION

Pre-condition: To install only if returned by "startCdmw.sh needed-tools" as a tool to install

Each distribution is contained in a tar file. E.g. TA0151_1-RedHat-ES4-gcc344-inline-FT-071011.tar (Refer to PID1 (ref [3]) for the exact TAO reference).

Uncompress the archive to a temporary directory and run the installation script providing the absolute name of the directory you wish to install TAO into.

```
tar -xf TA0151_1-RedHat-ES4-gcc344-inline-FT-071011.tar
./install.sh /tools/exec/TA0151_1_071011
```

Unpack the ACE / TAO source code:

```
cd /tools/exec/TA0151_1_071011/src
tar xzf src.tar.gz
```

Verify in the file:

```
/tools/exec/TA0151_1_071011/src/include/makeinclude/platform_macros.GNU
```

The following options are like shown:

```
optimize=1
debug=0
inline=1
```

If you need to change it PLEASE TO CONTACT CARDAMOM SUPPORT.

Set the following environment variables:

```
export ACE_ROOT=/tools/exec/TA0151_1_071011/src
export TAO_ROOT=$ACE_ROOT/TAO
export CIAO_ROOT=$TAO_ROOT/CIAO
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	50 of 70

```
export PATH=$TAO_ROOT/TAO_IDL:$TAO_ROOT/orbsvcs/Naming_Service:$PATH
export
LD_LIBRARY_PATH=$ACE_ROOT/ace:$TAO_ROOT/orbsvcs/orbsvcs:$TAO_ROOT/tao:$TAO_ROOT/TAO_IDL:$LD_LIBRARY_PATH
export TAO_DISABLE_CODESET_ENCODING=1
```

Some part of TAO (DanCE part) requires Xerces-C++ 2.7.0 to be installed, and the following environment variable to be defined :

```
export XERCESCR00T=<Xerces-2.7.0-install-dir>
```

Build TAO:

```
cd $ACE_ROOT
make -C ace
make -C ACEXML
make -C apps/gperf
make -C TAO/tao
make -C TAO/TAO_IDL
make -C TAO/orbsvcs/orbsvcs/MMPersistence
make -C TAO/orbsvcs/orbsvcs/MMPersistenceConfiguration
make -C TAO/orbsvcs/orbsvcs -f GNUmakefile
make -C TAO/orbsvcs/CosEvent_Service
make -C TAO/orbsvcs/Naming_Service
make -C TAO/orbsvcs/ImplRepo_Service
make -C TAO/orbsvcs/IFR_Service
make -C TAO/CIAO/ciao -f GNUmakefile.CIAO_Client
make -C TAO/CIAO/DAnCE
make -C TAO/CIAO/tools/XML
make -C TAO/CIAO/tools/Config_Handlers
make -C TAO/CIAO/DAnCE/Plan_Launcher
```

7.5 OPENFUSION JACORB INSTALLATION

Pre-condition: To install only if returned by “startCdmw.sh needed-tools” as a tool to install

To install JacORB launch Setup.class running this command:

```
java -cp . Setup
```

⇒ Follow the GUI instructions (use the Default Installation and specify the installation directory. e.g.: /tools/exec/Jacorb-OpenFusion)

Mind the Jacorb Home installation directory, the correct path is /tools/exec/Jacorb-OpenFusion/src/

Do these steps:

```
cd /tools/exec/Jacorb-OpenFusion/src
tar xzf src.tar.gz
cd /tools/exec/Jacorb-OpenFusion/src/bin
chmod a+x *
cp idl idljaco
```

To complete Installation set the following variables:

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	51 of 70

```
export JACORBHOME=/tools/exec/Jacorb-OpenFusion/src
for i in `ls $JACORBHOME/lib/*.jar`; do export CLASSPATH=$i:$CLASSPATH; done
```

NOTE 1: if you want to rebuild JacORB from the sources (e.g. if a patch has to be applied), do these steps:

```
cd /tools/exec/Jacorb-OpenFusion/src
vi build.xml
```

replace the sentence `"${basedir}/prism/src/org/jacorb/orb/naming/build.xml"` with this one:
`"${basedir}/src/org/jacorb/naming/build.xml"`

```
ant
```

7.6 OPENORB INSTALLATION

Pre-condition: To install only if returned by “startCdmw.sh needed-tools” as a tool to install

The steps to install OpenORB are the followings:

Uncompress the archive and move the directory OpenORB-<version> in the directory /tools/exec

```
tar -C /tools/exec -xzf OpenORB-<version>.tgz
```

Because of a compatibility problem between JacORB and OpenORB (problem referenced as PCR-0183), the LANG environment variable has to be set to en_US.UTF-8:

```
export LANG=en_US.UTF-8
```

To complete installation set the following environment variables:

```
export OPENORBHOME=/tools/exec/OpenORB-<version>
for i in `find $OPENORBHOME -name "*.jar"`; do export CLASSPATH=$CLASSPATH:$i; done
```

7.7 JAVACC INSTALLATION

Pre-condition: the JDK is installed.

The steps to install JavaCC are the followings:

```
unzip -d JavaCC<version> JavaCC<version>.zip
cd JavaCC<version>
java -cp . JavaCC<version>
```

⇒ follow the GUI instructions (specify the installation directory. e.g.: /tools/exec/javacc<version> and select “UNIX Shell Scripts” for the “Select Script Kind” window)

Remove the JavaCC<version> directory

```
cd ..
rm -rf JavaCC<version>
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	52 of 70

To complete installation set the following environment variable:

```
export PATH=/tools/exec/javacc<version>/bin:$PATH
```

7.8 OPENCCM INSTALLATION

Pre-conditions:

- OpenORB, JavaCC and Ant are installed.
- To install only if returned by "startCdmw.sh needed-tools" as a tool to install

The steps to install OpenCCM 0.4p8 are the followings (Refer to PID1 (ref [3]) for the exact OpenCCM reference):

Uncompress the archive, then:

```
tar xzf OpenCCM-0.4p8.tar.gz
cd OpenCCM-0.4p8
./build.sh
```

Note that this step is needed in order to generate the build.properties file (to be manually edited) and it is normal that it ends with an error message.

- Edit the file build.properties:
- Set the proper ORB.name variable:
ORB.name=OpenORB-1.3.0
- Set the install.dir variable, for instance:
install.dir=/tools/exec/OpenCCM-0.4p8
- Set the ORB.home.dir to the JOB install path, for instance:
ORB.home.dir=/tools/exec/OpenORB-1.3.0
- Set the javacc.dir variable, for instance:
javacc.dir=/tools/exec/javacc2.1/bin/lib
- Then:

```
export PATH=.:$PATH
./build.sh
```

Note that with this step OpenCCM is build.

```
./build.sh install
```

To complete installation set the following environment variable:

```
export PATH=/tools/exec/OpenCCM-0.4p8/bin:$PATH
export OPENCCMHOME=/tools/exec/OpenCCM-0.4p8
for i in `find $OPENCCMHOME -name "*.jar"`; do export
CLASSPATH=$CLASSPATH:$i; done
```

Remove the OpenCCM-0.4p8 directory

```
cd ..
rm -rf OpenCCM-0.4p8
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	53 of 70

7.9 XERCES C++ INSTALLATION

Pre-conditions:

- Gcc3.4.4 is installed.
- To install only if returned by “startCdmw.sh needed-tools” as a tool to install

The steps to compile XERCES are the followings (Refer to PIDI (ref [3]) for the exact Xerces C++ reference):

- Uncompress the archive

```
tar -xzf xerces-c-current.tar.gz
```

- Set the environment variable

```
export XERCECROOT=/tools/src/xerces-c-src_2_7_05
```

```
cd $XERCECROOT/src/xercesc
./runConfigure -p linux -c gcc -x g++ -d -P /tools/exec/xerces-c_2_7_0
```

```
make
su
insert the root password
```

```
make install
```

To complete installation set the following environment variable:

```
export LD_LIBRARY_PATH=/tools/exec/xerces-c_2_7_0/lib:$LD_LIBRARY_PATH
rm -rf /tools/src/xerces-c-src_2_7_0
```

7.10 SAXON INSTALLATION

Uncompress the archive into directory /tools/exec/saxon-<version>.

```
unzip -d /tools/exec/saxon-<version> saxon<version>.zip
```

To complete installation set the following environment variable:

```
export SAXONHOME=/tools/exec/saxon-<version>
export CLASSPATH=$CLASSPATH:$SAXONHOME/saxon.jar:$SAXONHOME/saxon-jdom.jar
```

7.11 R INSTALLATION

R is a statistical tool needed for analysis of data produced during benchmark tests.

The steps to install R are the following:

from the directory which contains the compress archive:

```
su
insert the root password
tar -C /tools/src -zxvf R-<version>.tgz
cd /tools/src/R-<version>
```

⁵Avoid the usage of XERCECROOT in the shell environment to set LD_LIBRARY_PATH, XERCECROOT is required just for compilation.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	54 of 70

```
./configure --prefix=/tools/exec/R-<version>
make
make check
make info
make install
make install-info
exit
```

To use R it is also required to set the following environment variables:

```
export PATH=$PATH:/tools/exec/R-<version>/bin
```

7.12 SPLICE-DDS V1.3

The installation of this tool is optional to run CARDAMOM.

The steps to install Splice-dds are the following:

- Navigate into the Splice directory (/SPLICE_HDE_V1.3_x86.linux2.6).

Check that package.ins is executable. If not, perform the following command:

```
chmod +x package.ins
```

- Run the following command:

```
./package.ins /tools/exec/Splice-dds
```

- where "tools/exec/Splice-dds" represents the path where you wish to install Splice DDS.
- To complete installation set the following environment variable:

```
source /tools/exec/Splice-dds/V1.3/x86.linux2.6/release.com
export LD_LIBRARY_PATH=/tools/exec/Splice-
dds/V1.3/x86.linux2.6/lib:$LD_LIBRARY_PATH
```

A version of SPLICE DDS 1.3 can be requested to PrismTechnologies (www.primstech.com).

7.13 CPPUNIT INSTALLATION

cppunit is a framework to manage implementation and running of unit tests.

The steps to install cppunit are the following:

from the directory which contains the compress archive:

```
su
insert the root password
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	55 of 70

```
tar -C /tools/src -zxf cppunit-<version>.tar.gz
cd /tools/src/cppunit-<version>
./configure --prefix=/tools/exec/cppunit-<version> --quiet
make
make -k install
exit
```

To use cppunit it is also required to set the following environment variables:

```
export PATH=$PATH:/tools/exec/cppunit-<version>/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/tools/exec/cppunit-<version>/lib
```

7.14 JUNIT INSTALLATION

junit is a framework to manage implementation and running of unit tests.

The steps to install junit are the following:

from the directory which contains the compress archive:

```
su
insert the root password

cd /tools/exec
tar -jxvf /tools/src/junit-<version>.tar.bz2
To use junit it is also required to for it to be on the classpath
```

7.15 DOXYGEN INSTALLATION

Doxygen is a program to generate documentation from code.

The steps to install doxygen are the following:

from the directory which contains the compress archive,

```
su
insert the root password

tar -C /tools/src -zxvf doxygen-<version>.src.tar.gz
cd /tools/src/doxygen-<version>
./configure --prefix /tools/exec/doxygen-<version>
make
make -k install
exit
```

To use doxygen it is also required to set the following environment variables:

```
export PATH=/tools/exec/doxygen-<version>/bin:$PATH
```

7.16 JAVA ORBACUS 4.1.0 INSTALLATION

Pre-condition: ORBACUS is installed.

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	56 of 70

The Steps to patch Java ORBacus:

- locate in the Java ORBacus sources directory:
`cd xxx/JOB-4.1.0`
- use the file JOB-4.1.0.patch with this command:
`cat JOB-4.1.0.patch | patch -p2`

The steps to compile ORBacus are the followings:

- Edit the file JOB-4.1.0/config/Make.rules
- Set "prefix=/tools/exec/JOB-4.1.0"
- Set "JIDL=/tools/exec/OB-4.1.0/bin/jidl"
- If JDK is 1.4.x uncomment the line "#BOOTCLASSPATH=..."
- Then:

```
make
mkdir /tools/exec/JOB-4.1.0
make install
```

- Copy /tools/exec/OB-4.1.0/bin into /tools/exec/JOB-4.1.0/bin

7.17 ORBACUS 4.1.0 INSTALLATION

Pre-condition: GCC3.4.4 is installed.

Steps to patch ORBacus:

- locate in the ORBacus sources directory:
`cd xxx/OB-4.1.0`
- use the file ob.patch with this command:
`cat OB-4.1.0.patch | patch -p2`

The steps to compile ORBacus are the followings:

```
./runconfig
  b  bourne shell
gcc2.95.3
yes shared library
no code optimization
yes debug info
  extra preprocessor flags
  extra compiler flags
  extra linker flags
  extra archiver flags
/tools/exec/OB-4.1.0 installation directory
. ./go
make
mkdir /tools/exec/OB-4.1.0
make install
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	57 of 70

WARNING: Some test was removed because OB was not yet configured for GCC 3.4.4.
The pthread-test on gcc-check file was removed.
Modification in some ch_* directory (the number of argument is different for a fstream function)
(see for example the ch_21/icp.cc file, line 317)

To complete installation set the following environment variable:

```
export PATH=/tools/exec/OB-4.1.0/bin:$PATH
export LD_LIBRARY_PATH=/tools/exec/OB-4.1.0/lib:$LD_LIBRARY_PATH
```

7.18 VALGRIND INSTALLATION

Pre-condition: GCC3.4.4 is installed.

Valgrind is a suite of tools for debugging and profiling Linux programs.

The steps to install valgrind are the following:

from the directory which contains the compress archive,

```
su
insert the root password
```

Note: an old version of valgrind is shipped with the RHES4U2 and could be already installed. Performs the followings commands to remove it:

```
rpm -e valgrind-callgrind
rpm -e valgrind
```

```
tar -C /tools/src -jxvf valgrind-<version>.tar.bz2
cd /tools/src/valgrind-<version>
./configure --prefix /tools/exec/valgrind-<version>
make
make -k install
exit
```

To use valgrind it is also required to set the following environment variables:

```
export PATH=/tools/exec/valgrind-<version>/bin:$PATH
```

7.19 EMMA INSTALLATION

Emma is a code coverage tool for Java™ programs.

The steps to install EMMA are the following (Refer to PIDI (ref [3]) for the exact EMMA reference)::

uncompress the archive into directory /tools/exec.

```
unzip -d /tools/exec emma-2.0.531p01.zip
```

To complete installation set the following environment variable:

```
export CLASSPATH=/tools/exec/emma-2.0.531p01/lib/ema.jar:$CLASSPATH
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	58 of 70

(WARNING: emma.jar is also present in JacORB directory. This patched emma.jar has to be present in CLASSPATH before the JacORB one's).

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	59 of 70

8. APPENDIX C: CONFIGURE EXAMPLE

In the following section we show an example of a configuration session, the part in the brackets < > means optional or user defined, the example is not exhaustive nor accurate it is only to show a typical configuration session.

```
$ ./startCdmw.sh configure
Initializing...
warning: your Build CSCI version is 8.0p03, this CSCI was previously patched
=====
Which CSCIs are available on your system ?
< Your previous selection was [CCMContainer, CCMDeployment, CodeGen, ConfAndPlug,
Event, FaultTolerance, Foundation, LifeCycle, LoadBalancing, ProdTests, Repository,
SystemMngt, Time, TraceAndPerf, XMLServer]. >
Do you want to discard the list and make a new selection ? (y,n) <[n]>
y
List of CSCI(s):
  1) CCMContainer
  2) CCMDeployment
  3) CodeGen
  4) ConfAndPlug
  5) Event
  6) FaultTolerance
  7) Foundation
  8) LifeCycle
  9) LoadBalancing
 10) ProdTests
 11) Repository
 12) SystemMngt
 13) Time
 14) TraceAndPerf
 15) XMLServer
Enter the number(s) of the CSCI and/or the group of CSCIs separated by semi-colons
(ex: 1;3;5-12;4)
1-15
Final selection: [CCMContainer, CCMDeployment, CodeGen, ConfAndPlug, Event,
FaultTolerance, Foundation, LifeCycle, LoadBalancing, ProdTests, Repository,
SystemMngt, Time, TraceAndPerf, XMLServer]
Which packages do you want to build ?
<Your previous selection was [DEFAULT] with the [c++, java] interfaces.>
Do you want to discard the list and make a new selection ? (y,n) [n]
y
List of package(s):
  1) CORE_FT -
Foundation, SystemMngt, TraceAndPerf, Repository, LifeCycle, ConfAndPlug, CodeGen, Event, F
aultTolerance
  2) DEFAULT -
Foundation, SystemMngt, TraceAndPerf, Repository, XMLServer, LifeCycle, ConfAndPlug, CodeG
en, Event, CCMContainer, CCMDeployment, Time, LoadBalancing
  3) DEFAULT_FT -
Foundation, SystemMngt, TraceAndPerf, Repository, XMLServer, LifeCycle, ConfAndPlug, CodeG
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	60 of 70

```

en,Event,CCMContainer,CCMDeployment,Time,LoadBalancing,FaultTolerance
Enter the number of the package and/or the group of packages separated by semi-
colons (ex: 1;3;5-12;4)
2
Selected package(s): [DEFAULT]
Do you want to add more packages ? (y,n)
n

Final selection: [DEFAULT]/[c++, java]
Do you want to compile the C++ interface in DEBUG mode ? (y,n) <[y]>

Do you want to compile with the optimisation flags? (y,n) <[y]>

Enter the trace level you want for your CARDAMOM System"
    0) No trace
    1) Fatal traces only
    2) Error traces + selection 1)
    3) Warning traces + selection 2)
    4) Info traces + selection 3)
    5) Debug traces + selection 4)
    6) All : Customer debug + selection 5)
Enter your choice:(0,1,2,3,4,5,6) [6]

Should the Call-Traces messages from the CARDAMOM services be activated ?(y,n) [n]

Where should this product be installed ? <[/<CDMW-INSTALLATION-PATH>]>

Enter the path of a common directory <[/<CDMW-SOURCE-PATH>]>

Checking all CSCIs properties...

Generated dependancies
Starting the tool configuration...
The configurator cannot determine which of these tools {ORBacusCPP=ORBacus C++,
TAO=The ACE ORB, TAOOpenFusion=The TAO OpenFusion} to use for the CCMContainer
CSCI, however you have previously selected [TAOOpenFusion].
Do you want to discard [TAOOpenFusion] and make a new selection ? (y,n) [n]
It is strongly advised to compile all the CSCIs against the same ORB.
Would you like to enable this constraint ? (y,n) <[y]>

The configurator cannot determine which of these tools {JacORB=JacORB,
ORBacusJAVA=ORBacus JAVA} to use for the CCMContainer CSCI, however you have
previously selected [JacORB].
Do you want to discard [JacORB] and make a new selection ? (y,n) <[n]>

Enter the JacORB installation directory [/tools/exec/Jacorb-OpenFusion-2.3/src]

Enter the JacORB bin directory [/tools/exec/Jacorb-OpenFusion-2.3/src/bin]

Enter the JacORB lib directory [/tools/exec/Jacorb-OpenFusion-2.3/src/lib]
    
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	61 of 70

```

Enter the JacORB idl directory [/tools/exec/Jacorb-OpenFusion-2.3/src/idl]
Enter the cppunit installation directory [/tools/exec/cppunit-1.10.2]
Enter the cppunit include directory [/tools/exec/cppunit-1.10.2/include]
Enter the cppunit lib directory [/tools/exec/cppunit-1.10.2/lib]
Enter the JUnit installation directory [/tools/exec/junit4.1]
Enter the JUnit lib directory [/tools/exec/junit4.1]
Enter the OpenCCM installation directory [/tools/exec/OpenCCM-0.4p8]
Enter the OpenCCM bin directory [/tools/exec/OpenCCM-0.4p8/bin]
Enter the OpenCCM lib directory [/tools/exec/OpenCCM-0.4p8/lib]
Enter the OpenCCM idl directory [/tools/exec/OpenCCM-0.4p8/idl]
Enter the OpenORB installation directory [/tools/exec/OpenORB-1.3.0]
Enter the OpenORB lib directory [/tools/exec/OpenORB-1.3.0/lib]
Enter the Saxon installation directory [/tools/exec/saxon-6.5.3]
Enter the Saxon lib directory [/tools/exec/saxon-6.5.3]
Enter the Apache Xerces C++ installation directory [/tools/exec/xerces-c_2_7_0]
Enter the Apache Xerces C++ include directory [/tools/exec/xerces-c_2_7_0/include]
Enter the Apache Xerces C++ lib directory [/tools/exec/xerces-c-2_7_0/lib]

Checking if g++ can compile a basic C++ program
Checking the g++ version
Checking if g++ has multithread support
Checking the SUN Java 2 SDK version
Checking if javac is found in the PATH
Checking the JacORB version
Checking the ACEOpenFusion version
Checking the ACEOpenFusion libraries
Checking the TA0OpenFusion version
Checking the TA0OpenFusion libraries
Checking the TAO IDL compiler version
Checking if tao_idl can compile a basic IDL file
Checking the cppunit include directory
Checking the cppunit lib directory
Checking the OpenCCM bin directory
Checking the OpenCCM lib directory
  
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	62 of 70

```

Checking the OpenCCM idl directory
Checking the OpenORB lib directory
Checking the Saxon lib directory
Checking the Apache Xerces C++ include directory
Checking the Apache Xerces C++ lib directory
Do you want to use Splice ? (y,n) [n]

Dispatching build files...
The new settings are saved into /<CDMW-SOURCE-PATH>/Build/.CDMW/Site-Linux-i686-
2.4.21-15.ELsmp#6.properties

BUILD SUCCESSFUL
Total time: X minutes Y seconds
    
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	63 of 70

9. APPENDIX D: TYPICAL REPORT OF THE RUNTEST

When "startCdmw.sh runtest" is launched, the following message appeared:

```
Buildfile: MasterBuild.xml
```

```
+-----+
| CARDAMOM SELF-TESTING |
+-----+
```

```
CARDAMOM is about to run the test applications to ensure that the
executables and libraries are built and that they function correctly.
Hence, it assumes that you did run the compilation of the product at
least once.
```

```
The tests are verbose on purpose. You can most probably ignore most of
the logs. A summary will be shown when all the tests are run.
```

```
+-----+
```

Then, for each CSC test, a report appeared with the number of test failed and the number of test succeeded.

For example:

```
[CSC] /home11/cardamom/Time/clockservice/test/testpolicies/c++
multicast address for federated clocks corbaloc:miop:1.0@1.0-TopicUpdateManager-
1/225.1.1.225:15555
ControllerEnforceIntervalPolicyTest::check_policy : OK
ControllerEnforceIntervalPolicyTest::PCR_0450_on_set_rate : OK
ControllerEnforceIntervalPolicyTest::PCR_0450_on_set : OK
ControllerEnforceIntervalPolicyTest::PCR_0450_on_pause_resume : OK
ControllerEnforceIntervalPolicyTest::on_set_time_and_rate_while_in_pause : OK
ControllerEnforceIntervalPolicyTest::more_controllers_with_different_policies_at_th
e_same_time : OK
ControllerEnforceIntervalPolicyTest::PCR_0450_on_terminate : OK
FederatedControllerEnforceIntervalPolicyTest::check_policy : OK
FederatedControllerEnforceIntervalPolicyTest::PCR_0450_on_set_rate : OK
FederatedControllerEnforceIntervalPolicyTest::PCR_0450_on_set : OK
FederatedControllerEnforceIntervalPolicyTest::PCR_0450_on_pause_resume : OK
FederatedControllerEnforceIntervalPolicyTest::on_set_time_and_rate_while_in_pause :
=====
Ending test testenforceintervalpolicy
=====
Summary:
Failed      : 0
Succeeded   : 14
Expected successfull tests (CPP) :14
>>>> Test result is set as TESTABLE_OK <<<<
```

If the message ">>>> Test result is set as TESTABLE_OK <<<<" appeared, the CSC test is OK

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	64 of 70

```
[CSC] /home/cardamom/Time/clockservice/test/testpolicies/c++
multicast address for federated clocks corbaloc:miop:1.0@1.0-TopicUpdateManager-
1/225.1.1.225:15555
ControllerCancelAllPolicyTest::on_set : OK
ControllerCancelAllPolicyTest::on_set_rate : OK
ControllerCancelAllPolicyTest::on_pause_resume : OK
ControllerCancelAllPolicyTest::on_terminate : OK
FederatedControllerCancelAllPolicyTest::on_set : OK
FederatedControllerCancelAllPolicyTest::on_set_rate : OK
FederatedControllerCancelAllPolicyTest::on_pause_resume : error
FederatedControllerCancelAllPolicyTest::on_terminate : OK
=====
Ending test testcancelallpolicy
=====
Summary:
  Failed      : 1
  Succeeded   : 7
  Expected successful tests (CPP) : 8
  >>>> Test result is NOT OK <<<<
```

If the CSC test is not OK the following message appeared : ">>>> Test result is NOT OK <<<<". In the summary, the numbers of failed and succeeded the are indicated.

```
+-----+
| SELF-TESTING REPORT |
+-----+

[TEST] /home/cardamom/FaultTolerance/ftwatchdog/test/testtimeoutrequest/c++
ID : U-FLT-0028 : "Test of detection of request timeout on a non FT process"
+
|   ### TEST FAILED ###
|   Testable objects to check           : 1
|   Testable objects successfully checked : 0
+-----+

[TEST] /home/cardamom/FaultTolerance/ftwatchdog/test/testtimeoutrequest/java
ID : U-FLT-1028 : "Test of detection of request timeout on a non FT process"
+
|   TEST OK
+-----+

[TEST] /home/cardamom/Foundation/common/test/teststring/c++
ID : U-FDS-0006 : no description
+
|   TEST OK
+-----+
```

At the end of the runtest, a report is edited. For each CSC, the status is displayed. All the status should be "TEST OK".

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	65 of 70

10. APPENDIX E: NEEDED-TOOLS TARGET EXAMPLE

After the `./startCdmw.sh` needed-tools and reply to all questions, the following message appeared:

```
-----
List of needed tools
-----
It is HIGHLY ADVISABLE to install each of these tools into separate locations
(e.g. /tools/tool1, /tools/tool2, etc.) so as to ease software upgrades and
to avoid unmanageable conflicts.

[GCC] GNU C++ compiler version 3.4.4

        License: GPL 2
        URL: http://gcc.gnu.org

        use the Enterprise Linux GCC rpm package

[Binutils] GNU Binutils version 2.15.92.0.2-15

        License: GPL 2
        URL: http://www.gnu.org

        use the Enterprise Linux Binutils rpm package

[J2SDK] SUN Java 2 SDK version 1.5.0_08

        The SUN Java 2 SDK is not bundled with CARDAMOM.
        License: Sun Community Source License
        URL: http://java.sun.com

[JacORB] JacORB version 2.3

        CARDAMOM JacORB PrismTech version.
        License: LGPL 2 (June 1991)
        URL: ftp://user:passwd@ftp-eu.prismtechnologies.com

        Refer to the SCOM for the installation

[TA0OpenFusion] The TA0 OpenFusion version 1.4.1

        License: ACE+TA0 License
        URL: ftp://user:passwd@ftp-eu.prismtechnologies.com

        Refer to the SCOM for the installation
```

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-103SCOM		Rev	11
	UNCLASSIFIED		Page	66 of 70

[CPPUnit] Cppunit version 1.10.2

CARDAMOM CDRom provides the CPPUnit tool, a framework for unit test.

License: LGPL 2.1 (Feb 1999)
 URL: cppunit.sourceforge.net

Refer to the SCOM for the installation

[OpenCCM] OpenCCM version 0.4p8

The CARDAMOM CDRom provides a patched version of OpenCCM. It requires JavaCC in order to compile.

License: LGPL 2.1
 URL: <http://openccm.objectweb.org>

- 1) tar zxf <OpenCCM-version.tar.gz> -C <tmp.dir>
- 2) cd <tmp.dir>
- 3) ./build.sh
- 4) edit <tmp.dir>/build.properties
- 5) ./build.sh
- 6) ./build.sh install

[JavaCC] Java Compiler Compiler version 2.1

JavaCC is a JAVA compiler compiler. It is not required by CARDAMOM but is necessary for OpenCCM to compile.

License: Java Compiler Compiler License
 URL: <http://javacc.dev.java.net>

- 1) unzip JavaCC2_1.zip
- 2) java -cp . JavaCC2_1

[OpenORB] OpenORB version 1.3.0

Required by OpenCCM.

License: Exolab License (BSD-like)
 URL: <http://openorb.sourceforge.net>

- 1) Uncompress OpenORB archive file.
- 2) Copy the directory OpenORB-1.3.0 in the directory /tools/exec.

[Saxon] Saxon XSLT Processor version 6.5.3

License: Mozilla PL

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	67 of 70

URL: <http://saxon.sourceforge.net>

- 1) Uncompress the archive file.
- 2) `mkdir -p /tools/exec/saxon-6.5.3`
- 3) Copy all the jar files into `/tools/exec/saxon-6.5.3`

[XercesCPP] Apache Xerces C++ version 2.7.0

License: Apache Software License v2.0

URL: <http://xml.apache.org/xerces-c>

- 0) Uncompress the archive file into `/tools/src/xerces-c-src2_7_0`
- 1) `export XERCECROOT=/tools/src/xerces-c-src2_7_0`
- 2) `cd $XERCECROOT/src`
- 3) `chmod +x ./configure`
- 4) `./runConfigure -plinux -gcc -xg++ -d -P/tools/exec/xerces-c-`
- 5) `make`
- 6) `make install`
- 7) `export LD_LIBRARY_PATH=/tools/exec/xerces-c-`

2_7_0

2_7_0/lib:\$LD_LIBRARY_PATH

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	68 of 70

11. APPENDIX F: CHECK-HOST-RESOURCES TARGET EXAMPLE

In case you see a warning message like the following one, when running "startCdmw.sh check-host-resources",:

```
Number of Processors: 4
CPU Speed: 3591.578 MHz
MEM Size: 4051 MB
SWAP Size: 0 MB
#####
warning: the total amount of swap memory in the system should be at least equal to
1024 MB
#####
Available disk space in current partition: 16678MB

BUILD SUCCESSFUL
Total time: 1 second
```

This means that you are trying to build CARDAMOM in a platform that have less than the recommended swap memory size.

The same warning message will be displayed in the following cases:

- o the number of processors is less than 2;
- o the total amount of fisical memory is less than 512 MB;

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	69 of 70

12. APPENDIX G: GETTING STARTED

This annex resumes the procedure to install CARDAMOM in the UNIX account of your choice

1. Copy the CARDAMOM archive and uncompress it
2. Configure the OS and patch it if necessary (refer to section 3.3.3 for more details)

Into the *Build* directory, run `./startCdmw.sh needed-tools` to know the needed tools necessary to use CARDAMOM following the package(s) selected (refer to the 4.2.2 and the 3 for more details)

WARNING: JDK and ANT have to be previously installed in order to run the script `./startCdmw.sh` (refer to 7.2, and 7.3).

3. Install all the tools needed and patch if necessary (refer to section 7 for Linux tools)
4. Configure the ORB (see section 3.3.2 for more details)
5. Run `./startCdmw.sh configure` and reply to the question (see section 4.2.2 for more details)
6. Run `./startCdmw.sh compile` to compile CARDAMOM (see section 4.2.3 for more details)
7. Run `./startCdmw.sh runtest` to run all the CARDAMOM Automatic tests. At the end, the report must be OK for all the CSCI (see section 4.2.4 for more details)
8. Run `./startCdmw.sh install` to install CARDAMOM your development environment (see section 4.2.5 for more details)

CARDAMOM CDM	61 486 532 AA	CTD 593	Date	23/06/2008
	R216-03-1033SCOM		Rev	11
	UNCLASSIFIED		Page	70 of 70