
COMPRO Common Component Architecture User's Guide

Release 2.0

Convergent Manufacturing Technologies Inc.

September 17, 2008

CONTENTS

1	Introduction	3
2	System Requirements	5
3	ABAQUS/COMPRO User's Guide	7
3.1	Introduction	7
3.2	Basic Concepts	7
3.3	Setting up the analysis	8
3.3.1	Considerations for the composite part	8
3.3.2	Considerations for the tool	13
3.3.3	Considerations for the tool-part interface	13
3.3.4	Load and boundary condition considerations	13
3.3.5	Tool removal considerations	15
3.4	Running the analysis	15
3.5	Analyzing the results	16
4	MSC.Marc/COMPRO User's Guide	19
4.1	Basic Concepts	19
4.2	Setting up the analysis	19
4.3	Running the analysis	21
5	The cca-configuration.xml File	23
5.1	A Brief Note About XML Files	23
5.1.1	Elements	23
5.1.2	Data	24
5.1.3	Comments	24
5.2	Required Elements and Attributes	24
5.2.1	<configuration> ... </configuration>	24
5.2.2	<directory type= "data" path= "mydata" />	25
5.2.3	<directory type= "output" path= "" />	25
5.2.4	<model_units system= "SIImperialUSCS" />	25
5.2.5	<analysis_code engine= "ABAQUS" />	25
5.2.6	<debug screen= "false" file= "false" filename= "debug.csv" debug_element= "1" />	25
5.2.7	<user_material id= "some_id">...</user_material>	25
	<composite_file name= "AS4-3501-6.xml" />	25
	<initial_doc DOC= "0.001" />	25
	<smearing A= "1.0" B= "0.0" />	26
6	The CCA Data File Formats	27
6.1	Material Type="Composite"	28

6.1.1	<constituent volume_fraction= "[0-1]" type= "fibrelmatrix" > name </constituent>	28
6.1.2	<micromechanics model= "default bogetti_and_gillespie modified_rosen_and_hashin chamis" />	28
6.1.3	<lamina_properties>	28
6.2	Material Type= "Fibre"	30
6.3	Material Type= "Matrix"	31
6.4	Creating a Simple Material Datafile	32
6.4.1	Material Data	32
6.4.2	Destination Format	32
6.4.3	File Naming Convention	32
6.4.4	Matrix Data File	33
	XML Declaration	34
	File Header	34
	Material Element	34
	Conductivity	35
	CTE	35
	Cure Kinetics	35
	Density	35
	Modulus Development	36
	Poisson's Ratio	36
	Specific Heat	36
	Cure Shrinkage	36
	Viscosity	37
	Complete Data File	37
6.4.5	Fibre Data File	39
6.4.6	Composite Data File	39

Note: This guide is intended for qualified users who will exercise sound engineering judgement. The COMPRO Common Component Architecture is inherently complex, and the examples and procedures outlined in this guide are not intended to be exhaustive or applicable to any particular situation. Users are cautioned to satisfy themselves as to the accuracy and suitability of the results of their analyses.

Convergent Manufacturing Technologies, Inc. (CMT) will not be held responsible for the accuracy or usefulness of any analysis performed using the COMPRO software or the procedures, examples, or explanations in this manual. CMT shall not be responsible for the consequences of any errors or omissions that may appear in this manual.

CMT makes no representations or warranties, either express or implied, with respect to the guide and specifically disclaims any implied warranty of non-infringement or merchantability or fitness for a particular purpose. CMT shall in no event be liable for any loss of profits, be they direct, consequential, incidental, or special or other similar or like damages arising from any defect, error or failure to perform with respect to the guide, even if CMT has been advised of the possibility of such damages. The Recipient hereby acknowledges that it has been advised by CMT to undertake its own due diligence with respect to the guide.

In no event shall CMT be liable for special, indirect or consequential damages, including those damages arising from lost profits in the use or misuse of the guide, even if CMT has been advised of the possibility of such damages.

This guide is intended to assist with the process design and is not a substitute for independent testing of a product's stress, safety and utility. CMT does not warrant the accuracy of the results obtained through use of the guide.

The COMPRO Common Component Architecture (CCA) software described in this guide is available only under license from Convergent Manufacturing Technologies, Inc., and may be used or reproduced only in accordance with the terms of such license.

This guide and the software described in the guide are subject to change without notice.

No part of this manual may be reproduced or distributed in any form without prior written permission of Convergent Manufacturing Technologies, Inc.

©Convergent Manufacturing Technologies, Inc., 2005. All rights reserved.

TRADEMARKS:

The trademarks and service marks (trademarks) in this guide are the property of Convergent Manufacturing Technologies, Inc. or third parties. You are not permitted to use these trademarks without the prior written consent of Convergent Manufacturing Technologies, Inc.

The following are trademarks of Convergent Manufacturing Technologies, Inc.: COMPRO, COMPRO Editor, COMPRO Common Component Architecture, COMPRO 2D, COMPRO 1D, COMPRO 0D.

Other company, product, and service names may be trademarks or service marks of their respective owners.

Convergent Manufacturing Technologies, Inc.

403-6190 Agronomy Road

Vancouver BC V6T 1Z3

CANADA

Tel: +1-604-822-9682

Fax: +1-604-822-9659

Email: support@convergent.ca

WWW: <http://www.convergent.ca>

Introduction

The COMPRO Common Component Architecture (CCA) is a set of tools that allows interoperability between various pre-processors, post-processors, finite element engines and the COMPRO process modelling engine (see Figure 1). In practice, this means that process modelling of composite materials can be performed with commercial finite element codes such as ABAQUS, ANSYS, or NASTRAN¹. As of this writing, only ABAQUS and MSC.Marc are supported, but other codes will be supported based on customer demand.

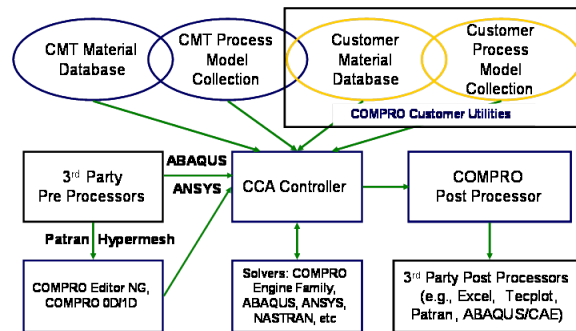


Figure 1.1: COMPRO Common Component Architecture overview.

¹ABAQUS is a registered trademark of Dassault Systemes. ANSYS is a registered trademark of ANSYS, Inc. NASTRAN is a registered trademark of the National Aeronautics Space Administration (NASA).

System Requirements

Running 3D simulations of composite materials with sufficient detail to capture processing-related effects requires considerable computational resources. A modern CPU (or CPUs) such as a fast Pentium 4 or AMD Athlon is recommended. A minimum of 1 GB of memory is also recommended. The results of the simulations can be quite large, and a large amount of free hard drive space, ~100 GB, is recommended before starting new simulations.

The CCA has been tested on x86 compatible hardware architectures. The CCA has been tested on Microsoft Windows, and various flavours of Linux, but should run on any system that has a compatible FORTRAN 90 and C++ compiler. Note that a compiler is not required for end-user use.

In addition to these requirements, each component has its own requirements:

ABAQUS/COMPRO: A valid ABAQUS/Standard v6.4 - v6.8 license

MSC.Marc/COMPRO: A valid MSC.Marc 2005-r2 or 2005-r3 license

ABAQUS/COMPRO User's Guide

3.1 Introduction

This guide outlines in general terms the steps required to set-up and run a processing model in ABAQUS/COMPRO. Some basic familiarity with ABAQUS/Standard is assumed. The guide makes many references to ABAQUS command cards (keywords) and only provides minimal detail on the commands. The ABAQUS User's Manual and ABAQUS Keywords Reference Manual should be consulted for complete details of each keyword.

3.2 Basic Concepts

Generally speaking, there are four steps associated with setting up and running an ABAQUS/COMPRO analysis:

1. Generate a mesh, with appropriate loads, boundaries, and materials in a pre-processor.
2. Edit the ABAQUS input file (.inp) to configure it to use the CCA routines.
3. Edit or verify the CCA configuration file.
4. Run the analysis.

Note that due to how ABAQUS/Standard is structured, a full process modelling simulation requires two or three analyses. First, the thermal problem must be solved, where the heat transfer, heat generation due to reaction, etc., is used to determine the temperature history of each integration point in the model. Next, the flow problem may be solved, using the temperature history generated in the thermal analysis. This determines the changes in fibre volume fraction before the resin gels. Finally, the stress problem is solved, using the temperature history, and optionally the volume fraction history, to determine the displacements and stress due to the cure cycle. A different input file is required for each of these analyses. Note that as of this writing, the stress analysis cannot import the volume fraction from the flow analysis, but this will change with the next release of COMPRO CCA.

Warning: If a flow analysis is going to be run, the .inp file must be prefaced with `flow` as the first four letters. For example, `flow-testjob.inp` is an admissible file name for an ABAQUS/COMPRO flow analysis, whereas `testjob-flow.inp` is not.

This two-step (or three-step) analysis is achieved by writing the temperature history to a `.fil` or `.odb` file during the thermal analysis.

Note: You can only write to `.odb` files in ABAQUS 6.5 or higher.

The stress analysis then reads the temperature data from this file. The `.fil` files, however, can quickly become quite large, often many GB in size. The `.fil` file is generated by telling ABAQUS to write the nodal temperature to file for the node set containing all nodes:

```
*NODE FILE, NSET=ALL  
NT
```

.odb files are generated in a similar manner (ABAQUS 6.5+ only):

```
*NODE OUTPUT, NSET=ALL  
NT
```

A similar procedure will be required to transfer the volume fraction from the flow analysis to the stress/deformation analysis. In this case, field variables must be written to the .fil file. ABAQUS v6.6 does not support writing field variables to .odb files. Additionally, field variables can only be written for elements, and not nodes, so the output must be:

```
*EL FILE, ELSET=ALL  
FV
```

This causes a problem, however. ABAQUS reads field variables from a file using the *FIELD input card. The *FIELD card will only read nodal data. An extra step (to be implemented in the next CCA release) will be required to convert the element-based field variables to nodal values that can be read by the *FIELD card.

Simplified analyses can be performed, in which just the thermal analysis is run (if the analysis is being run to determine exotherms, for example), or in which just the stress analysis is run (in this case ignoring any thermodynamic effects and assuming that the part is the same temperature as the air temperature, everywhere). Similarly, the flow analysis can be run independently, or the thermal analysis can be followed by only a flow analysis, without a subsequent stress analysis.

3.3 Setting up the analysis

Create the geometry and element mesh in ABAQUS/CAE or your favourite pre-processor. In the process modelling of composites, especially in 3D analyses, quite a bit of care and planning must be put into the pre-processing.

Warning: Note that currently, the mesh must be in either metres or inches, depending on your unit system (SI or USCS). If your mesh is in not in these units, the results will not be meaningful. Particularly, in USCS (Imperial), be certain that all units (including derived units) are in inches, pounds, Fahrenheit / Rankine, seconds, BTU, etc.

Most analyses will include a tool, some sort of interaction between the tool and the part, and the part. The following sections outline some things that must be considered when creating the model.

3.3.1 Considerations for the composite part

The most complex problem is meshing the composite part. There are several issues that must be considered:

- Element selection

In ABAQUS, there is a large number of 3D elements to choose from. The elements that you use depend on the situation:

- For simple stress analyses, we recommend using either C3D8 (8 noded linear continuum hexagons) or C3D20R elements (20 noded linear continuum hexagons).

Warning: Avoid using C3D20 elements if the aspect ratio is large – you may have shear locking in the elements.

- For stress analyses involving contact (to simulate tool-part interaction), specialized variable node elements, C3D27R must be used. These elements are 20 noded elements with additional nodes added to ensure that contact functions properly. One node is automatically added to the centroid of the element, and then you need to specify an additional node per face of the element that is in contact. So, for elements that are in contact on one face, 22 nodes are required. For contact on two faces, 23 nodes are required. See p. 14.1.4-21 of the ABAQUS User's Manual for details on these element types.
- For simple thermal analyses, we recommend using either DC3D8 (8 noded diffusive heat transfer linear elements) or DC3D20 (20 noded diffusive heat transfer quadratic elements).
- For thermal analyses that will be followed by flow or stress analyses, you should use elements that have the same number of nodes in each step. So, DC3D8 thermal elements can be followed by C3D8 stress elements. DC3D20 elements should be used in conjunction with C3D20 or C3D20R stress elements. If you use elements with fewer nodes in the thermal analysis than the stress analysis, then when the *TEMPERATURE card is specified in the stress analysis, the MIDSIDE parameter must be specified.
- For flow analyses, we recommend using either C3D8PH (8 noded pore pressure linear elements) or C3D20RPH (20 noded pore pressure quadratic elements).

• Section definition

Typically, you will assign several or many plies to a single element, and for stress analyses, ABAQUS/Standard provides composite solid sections to do this. Note that thermal analyses and flow analyses cannot use composite solid sections, and certain material smearing factors must be supplied to the input deck to approximate layups by element average behaviours (discussed later in this section).

Composite solid sections are not supported in ABAQUS/CAE, so the .inp file must be edited with a text editor to change the section definition. See p. 18.22-1 of the ABAQUS v6.5 Keywords Reference Manual for details on the *SOLID SECTION card.

To use composite solid sections, add the COMPOSITE option to the section definition, and specify the STACK DIRECTION. The stack direction is based on the face numbers of the element, which are in turn determined from the local element numbering. See p. 14.1.1-15 of the ABAQUS v6.5 User's Manual. Figure 2 illustrates the element numbering, the resulting face numbering, and the stack direction numbering for an 8 noded element.

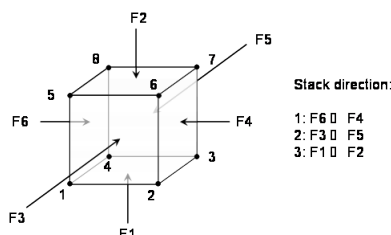


Figure 3.1: Illustration of face numbering and stack direction for 8 noded elements.

After specifying the section in this manner, each ply must be defined. The plies are defined by specifying the relative or absolute thickness, number of integration points, material, and orientation of each layer. An example *SOLID SECTION definition is:

```
*SOLID SECTION, COMPOSITE, ELSET=EX1, STACK DIRECTION=3
1, 1, AS4-3501-6, ORI0
1, 1, AS4-3501-6, ORI45
1, 1, AS4-3501-6, ORI-45
1, 1, AS4-3501-6, ORI90
```

In this case, the ORI0, ORI45, etc., are named *ORIENTATION specifications (discussed below).

There is a problem, however, in thermal analyses. ABAQUS/Standard does not have composite solid sections for diffusive heat transfer elements. As such, you cannot use *SOLID SECTION, COMPOSITE in thermal

analyses. The only material parameter of interest in thermal analyses that is directional, though, is the conductivity. Conductivity is therefore smeared through the element, and in-plane weighting factors are applied to the element local 1 and 2 directions. See the commentary on *USER MATERIAL below.

- Directionality must be maintained as the part moves through 3D space

Due to the orthotropic nature of composites, keeping track of the laminate 0° direction presents some challenges. A seemingly simple curve in space, like a transition from a flange to a web, requires that the elements have local orientations, so that the laminate 0° orientation is pointing in the correct direction over the whole curve.

ABAQUS does not let you define a reference curve for directionality (as in COMPRO 2D), but it does allow you to define local orientations within elements using a method called `OFFSET TO NODES`. The *ORIENTATION card in the ABAQUS input file defines a local axis system for a material or element property definition, and among the available options is `DEFINITION=OFFSET TO NODES`. This orientation definition specifies the local orientation within elements relative to the node ordering in the element definition. That is, the 0° direction can be set to be parallel to the line segment joining nodes 1 and 2 (in element local numbering) of the element. Or, the 0° direction can be set at some specified angle from a line segment joining two nodes.

To take advantage of the `OFFSET TO NODES` orientation, quadrilateral or hexagonal elements should be used, and one edge of each element should be oriented in the same direction as the rest of the elements. Further, care must be taken to ensure that all elements are created using the same internal node numbering system. For shell elements, this is as simple as ensuring that the element normals are pointing in the same direction. For brick elements, it can be more challenging, but starting with shell elements that have properly oriented normals is a good step.

As an example, consider the hexagonal element in Figure 3. To define a local coordinate system such that the 0° direction is parallel to the line formed by the line segment from node 1 to node 2, and the 90° direction is parallel to the line segment from node 1 to node 4, the following orientation definition could be used:

```
*ORIENTATION, NAME=DEMO, DEFINITION=OFFSET TO NODES
2, 4, 1
1, 0
```

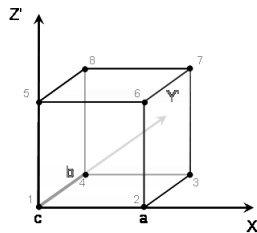


Figure 3.2: Illustration of setting a local coordinate system using `OFFSET TO NODES`.

Sometimes, the geometry or the meshing does not lend itself to using `OFFSET TO NODES` to achieve the proper orientation. In these circumstances, it may be necessary to define orientations relative to the global coordinate system for small groups or even individual elements. Careful planning of the meshing can minimize the amount of local orientation definitions.

- The parts usually considered in 3D tend to be large, compared to the thickness of the composites, leading to a large number of elements, and large aspect ratios. Aspect ratios not exceeding 10:1 are recommended for linear elements (8 noded hexagonal bricks), and aspect ratios not exceeding 100:1 for second-order elements (20 noded bricks).
- When specifying the composite material(s) in ABAQUS/CAE, use a reasonable material identifier, but just specify generic material properties (i.e., isotropic). The material definition must be edited later in a text editor in order to use the ABAQUS/COMPRO subroutines.

Several input cards must be changed or added to the `.inp` file in order to have a working ABAQUS/COMPRO analysis.

Beneath the appropriate `*MATERIAL` card,

- `*MATERIAL` must be modified

The `NAME` parameter must be set to be equal to one of the unique names listed in the `cca-configuration.xml` file. If an ineligible material name is specified, an error is generated and the run will not proceed.

In addition to changing the `*MATERIAL` card, this will probably mean changing the `*SECTION` card to ensure that the materials specified match the name of the materials listed in the `*MATERIAL` cards. An example `*MATERIAL` card is:

```
*MATERIAL, NAME=as4-3501-6-2005-04-06
```

- `*USER MATERIAL` must be added

This card indicates to ABAQUS that a user subroutine is used. An example `*USER MATERIAL` card is:

```
*USER MATERIAL, CONSTANTS=0
```

In the past, two constants were defined to account for smearing in thermal analyses, but the smearing constants have been moved to the `cca-configuration.xml` file. Please see Section 4.2 for details.

- `*DEPVAR` must be added

This card tells ABAQUS how many internal state variables are used in the user subroutines. In the COMPRO Components, this number is 8. An example `*DEPVAR` card is:

```
*DEPVAR
8
```

Although the `*DEPVAR` card is intended for use within the `*USER MATERIAL` cards, there may be some circumstances to use `*DEPVAR` within a built-in `*MATERIAL` card. In cases where hybrid elements consist of composite and non-composite materials, you will want to initialize the composite material (see `*INITIAL CONDITIONS`, below). However, `*INITIAL CONDITIONS` operates on elements and not layers or plies. In order to use `*INITIAL CONDITIONS` on a hybrid element, all materials in the hybrid element must have `*DEPVAR` specifications, even if the built-in material models do not use the `*DEPVAR`s.

- For flow analyses, user defined field relationships are used, and as such, the material must include a `*USER DEFINED FIELD` card. This card has no parameters:

```
*USER DEFINED FIELD
```

- Due to how permeability is specified in ABAQUS, flow analyses must pre-process the permeability. A utility has been supplied to do this task, `cca_flowpre.exe`. This utility is a command-line utility, and requires the name (unique name) of the CCA material as an input. The utility generates a file that is named prefixed with `K_` followed by the name of the material and `.inp` as the file extension. This file is simply a table that defines permeability as a function of the field variables. This file is unique to a particular material and only needs to be regenerated if the material properties change.

For example, using the material `as4-3501-6-2005-04-06`:

```
C:\Analysis> cca_flowpre.exe as4-3501-6-2005-04-06
                [...]
                *Output file name: K_AS4-3501-6-2005-04-06.inp
                *Wrote      964 lines.
C:\Analysis>
```

Now, this file must be referenced in the material definition of in the main ABAQUS input file:

```
*INCLUDE, INPUT= K_AS4-3501-6-2005-04-06.inp
```

- The `*EXPANSION` card must be added or changed to tell ABAQUS to use the user subroutines when determining the displacements of the elements. An example `*EXPANSION` card is:

```
*EXPANSION, TYPE=ORTHO, USER
```

- *DENSITY must be defined.

Due to how ABAQUS interacts with the user defined subroutines, the value you specify for *DENSITY is different depending on the context.

For flow analyses, you must specify the nominal density for the material:

```
*DENSITY
1535.,
```

For thermal analyses, you must specify 1.0 for the density:

```
*DENSITY
1.,
```

For stress analyses, you must specify the nominal density for the material if body forces (gravity, accelerations, etc) are involved, otherwise, it is permissible to leave it set to 1.0:

```
*DENSITY
1535.,
```

- *INITIAL CONDITIONS must be added

Finally, for each composite material, the initial values of the state variables must be defined. In older versions of COMPRO CCA, this was done using a detailed *INITIAL CONDITIONS card, but the procedure has changed for current versions of CCA.

The form of the *INITIAL CONDITIONS card is:

```
*INITIAL CONDITIONS, TYPE=SOLUTION, USER
```

The cca-configuration.xml file should be modified to account for initial degree of cure (see Section 4.2).

In flow analyses, some other initial conditions must be defined. The initial void ratio (which is the ratio V_r/V_f), initial saturation, and the field variables must be defined.

```
*INITIAL CONDITIONS, TYPE=RATIO
ALLNODES, 1.0
*INITIAL CONDITIONS, TYPE=SATURATION
ALLNODES, 1.0
```

The field variables are given in Table 1:

Table 1. Field variables used in a ABAQUS/COMPRO analysis

Number	Field Variable
1	ln(viscosity)
2	Darcy permeability (33 direction)
3	Darcy permeability (22 direction)
4	Darcy permeability (11 direction)

An example of the initiation of the field variables is:

```
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=1
ALLNODES, 2.30259
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=2
ALLNODES, 3.675E-13
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=3
ALLNODES, 3.675E-13
*INITIAL CONDITIONS, TYPE=FIELD, VARIABLE=4
ALLNODES, 3.675E-13
```

3.3.2 Considerations for the tool

The tooling typically requires much less effort to create in the pre-processor than the part. When creating the tooling, consider the following things:

- For simple analyses, the tool does not require any special attention. Built-in material models are sufficient for most cases where metallic tooling is used. In cases involving composite tooling, then an orthotropic material model must be used, and attention must be given to the orientation of the material within the elements, as in the discussion of the part, above.
- If contact surfaces are going to be used (see the next section) then the tool should be larger than the composite part, to allow the part to expand and not exceed the tooling boundaries. Additionally, the tool should have a coarser mesh density, so that the contact can be calculated properly. If nodes on the part and the tool are coincident, then the contact algorithm may not work as expected.
- If contact surfaces are used, avoid applying pressure directly to elements involved in the contact.

3.3.3 Considerations for the tool-part interface

Tool-part interaction can cause some significant processing-induced deformations in composite parts. The interaction between a tool and a part may be complete – that is, the part is completely bonded to the tool. In some cases, there may be no interaction, a perfectly frictionless interface between the tool and part. Often, the true case lies somewhere between these extremes. In COMPRO 2D, this intermediate interaction is typically modelled using a shear layer, which is a small layer of elements between the part and tool that has a very low shear stiffness. ABAQUS offers another way of handling this interaction by providing frictional contact surfaces.

- Perfect bonding is modelled by ensuring that the nodes on the tool and the part at their interface are coincident and equivalenced.
- Creating a shear layer is simply a matter of creating a layer of tool material that is thin (thickness on the order of the elements comprising the part) and has low through-thickness shear stiffnesses. Selecting the value of shear stiffness used in the shear layer is much more an art than a science. For example, a good rule of thumb for aluminium tools is to choose a shear stiffness such that the shear stresses observed in the shear layer are on the order of 12 kPa (~1.7 psi).
- Contact surfaces should be used for both frictionless and frictional tool-part interactions. Analyses involving contact surfaces will require considerably more time to complete. Pressures must be used to ensure continual contact between the tool and part. See section 29.2 of the ABAQUS User's Manual for details on implementing contact surfaces.

3.3.4 Load and boundary condition considerations

ABAQUS/COMPRO analyses typically involve two types of loads: pressure and temperature. In thermal analyses, only the temperature loads and boundaries need to be specified. In stress analyses, the pressure loads and displacement boundaries need to be specified. In flow analyses, you can specify the pore pressure.

In thermal analyses, temperature loads are typically applied in three steps. First, the initial temperature is defined:

```
*INITIAL CONDITIONS, TYPE=TEMPERATURE
ALL, 20
```

In this case, ALL is the node set ID containing all the nodes in the problem.

Next, the temperature cycle is defined. First, ensure that the current load step allows for a ramped load (in this case, temperature) magnitude:

```
*STEP, AMPLITUDE=RAMP, INC=1000
```

Next, the heat transfer engine must be configured. In this control card, the duration of the heat transfer, and the maximum temperature change allowed in an analysis increment are specified. Additionally, the initial time increment, total time period, minimum time increment, and maximum time increments must be specified:

```
*HEAT TRANSFER, END=PERIOD, DELTMX=5.0  
20, 15800, 0.01, 60.0
```

Next, the temperature cycle is defined using the *AMPLITUDE card. Note: the time is in seconds.

```
*AMPLITUDE, NAME=CureCycle  
0, 20, 3400, 180, 12400, 180, 15800, 20
```

Finally, the temperature cycle is applied to the appropriate surfaces of the part using the *FILM card:

```
*FILM, OP=NEW, AMPLITUDE=CureCycle  
SURFACE, F1, 1.0, 80.
```

The *FILM card applies a temperature (and a heat transfer coefficient) to element faces. You must specify which face of the elements to apply the temperature, according to the names (F1, F2, ..., F6) in Figure 2.

There may be some instances where you wish to fix some nodes at a reference temperature directly, without a heat transfer coefficient (effectively an infinite HTC). In this case you can either apply a temperature boundary condition or use *TEMPERATURE. Both methods directly assign the cycle temperature to the nodes:

```
*BOUNDARY, OP=NEW, AMPLITUDE=CureCycle  
AIRTC, 11, 24, 1.0
```

or

```
*TEMPERATURE, OP=NEW, AMPLITUDE=CureCycle  
AIRTC, 1.0
```

In stress analyses and flow analyses, temperature loads are typically applied using the *TEMPERATURE card. If you have previously run a thermal analysis, you can read the temperature history from a .fil or .odb file.

You can only use .odb files in ABAQUS 6.5 or higher. Further, for ABAQUS 6.5 or higher, if you are using the same order elements in the thermal and stress analyses, you should specify the INTERPOLATE parameter with the *TEMPERATURE card. If you are not, you need to use the MIDSIDE parameter, but this parameter is mutually exclusive to the INTERPOLATE parameter, and you may encounter some odd behaviour in reading the temperature history.

```
*TEMPERATURE, OP=NEW, FILE=ThermalAnalysis, INTERPOLATE
```

If you have not already run a thermal analysis, you can specify the temperatures by specifying the temperatures everywhere. In this case, no heat transfer will be considered, heat generated by the reaction will not be considered, and so on.

```
*TEMPERATURE, OP=NEW, AMPLITUDE=CureCycle  
ALL, 1.0
```

Pressure loads are typically applied in stress analyses, and in ABAQUS/Standard, these are applied using *DLOAD and an accompanying *AMPLITUDE definition. For example:

```
*AMPLITUDE, NAME=CycPress
0,0.0,1200,5.85E+5,21800.0,5.85E+5

*DLOAD, OP=NEW, AMPLITUDE=CycPress
PART_TOP_PRESS, P2, 1.0
```

As in any stress analysis, displacement boundaries must be applied to at least prevent rigid body motion. The *BOUNDARY keyword provides the constraints needed for most analyses:

```
*BOUNDARY, OP=NEW
Part_SYM, 1,1, 0.0
PART_SLIDINGZ, 2,2, 0.0
```

In flow analyses, pore pressure is degree of freedom 8. To allow flow at a boundary, you can set the pore pressure at a group of nodes to zero:

```
*BOUNDARY, AMPLITUDE=BC, OP=NEW
TOPNODES, 8
```

You can also control flow through the *FLOW keyword. Consult the ABAQUS Keyword Manual for more information.

3.3.5 Tool removal considerations

In typical COMPRO 2D analyses, the last step in the analysis is to remove the part from the tool and determine the deformations resulting from the residual stresses. This can be accomplished in ABAQUS/COMPRO by using different load steps within the analysis. Within a *STEP card, the *MODEL CHANGE keyword can be specified. *MODEL CHANGE allows the user to add or remove elements and contact pairs (master and slave surfaces). Keep in mind that boundaries, pressures, and temperatures might need to be changed as a result of removing elements.

```
*STEP, NAME=Tool_Removal
*MODEL CHANGE, REMOVE, TYPE=CONTACT PAIR
S_TOOL-PART_CONTACT, M_TOOL-PART_CONTACT
*MODEL CHANGE, REMOVE, TYPE=ELEMENT
TOOL
```

3.4 Running the analysis

The COMPRO Common Component Architecture has been implemented into ABAQUS as a set of routines called by the user subroutines UMAT, UEXPAN, UMATHT, and USDFLD. ABAQUS/COMPRO is distributed as a compiled object file that ABAQUS combines with its own engine at run time. For Windows, this file is standardU.dll and in Linux, it is libstandardU.so. This object file must be placed in an appropriate directory prior to running your analysis. On Windows, an appropriate location might be C:\Abaqus\6.7-1\lib or in Linux, /opt/abaqus/6.7-1/lib. The location is not important, but knowing where the object file is located is important.

The directory containing your ABAQUS .inp file is your job directory. The cca-configuration.xml file must be in the job directory, and an ABAQUS file, abaqus_v6.env must also be in that directory. The abaqus_v6.env file is a configuration file for ABAQUS. If you do not typically use this file in your analyses, copy the one located in the ABAQUS site directory (C:\Abaqus\6.7-1\site or /opt/abaqus/6.7-1/site) to your job directory.

After copying the `abaqus_v6.env` file to your job directory, use a text editor (such as Notepad, EMACS, vi, etc) to open the file and place one line at the bottom of the file. In Windows, the line should be of the form (the double back-slashes are important):

```
usub_lib_dir='C:\\Abaqus\\6.7-1\\lib'
```

and in Linux, it is a little different (note the direction and number of the directory separator slashes)

```
usub_lib_dir='/opt/abaqus/6.7-1/lib'
```

Make sure that you change the directory to reflect the location where the supplied object files were put.

Now, the `cca-configuration.xml` file must be edited (as of this writing, with a text editor). The details of setting the CCA options can be found in Chapter 4 of the COMPRO Common Component Architecture User's Guide.

Important: If you want to run a flow analysis, then you must have run a pre-processor for permeability. See the section regarding defining the material properties.

To run an ABAQUS/Standard v6.7-1 analysis with ABAQUS/COMPRO, invoke ABAQUS at a command line with the following options:

```
abq671 job=jobname interactive
```

Recall that for flow analyses, the jobname must have `flow` as the first four characters.

If everything is working fine, then ABAQUS should report something like:

```
NOTE: "usub_lib_dir" has been added to User Subroutine library path.
```

After linking the object file, ABAQUS will parse through the input file. If there are any ABAQUS/COMPRO errors, an error message will be printed to both the screen and the `.dat` file, and the run will terminate. If there are no errors, the run will proceed. Next, a message will be displayed on the screen indicating that the COMPRO Common Component Architecture is being used. If there are problems finding the material specified in the `.inp` file in the CCA XML data files, an error will be generated, and the run will terminate. If this happens, CCA will report the names of the CCA materials that it has found (those located in the files specified in the `cca-configuration.xml` file). Use only these names in the `*MATERIAL NAME=` card in the `.inp` file.

3.5 Analyzing the results

ABAQUS/CAE or any other compatible post-processor (such as MSC.Patran) can be used to visualize the results. The user can request that the temperatures, displacements, etc., be written to the `.odb` file using `*NODE OUTPUT` and `*ELEMENT OUTPUT` as appropriate. State variables can also be written and analyzed. Specify `SDV` in the component list of the `*ELEMENT OUTPUT` keyword (only for element sets that include curing materials):

```
*ELEMENT OUTPUT  
1, 2, 3, 4, 5, 6, 7, 8  
TEMP, SDV
```

In the COMPRO Common Component Architecture, there are eight state variables, as shown in Table 2:

Table 2. State variables used in the COMPRO Common Component Architecture

Number	State Variable
1	Degree of Cure
2	Cure rate (1/s)
3	Previous timestep cure rate (CK9 only)
4	Prev. timestep temperature rate (CK10 only)
5	Time of last State Variable update (MSC.Marc only)
6	Degree of cure at last SV update (MSC.Marc only)
7	Current Vf (for flow analyses)
8	Previous timestep temperature (for flow analyses)

MSC.Marc/COMPRO User's Guide

Please note that this section is under revision and is not complete. Please contact CMT for updated recommendations.

This guide outlines in general terms the steps required to set-up and run a processing model in MSC.Marc/COMPRO. Some basic familiarity with MSC.Marc is assumed. The guide makes many references to MSC.Marc command cards (keywords) and only provides minimal detail on the commands. The MSC.Marc documentation should be consulted for complete details of each keyword.

4.1 Basic Concepts

Generally speaking, there are five steps associated with setting up and running a MSC.Marc analysis:

1. Generate a mesh, with appropriate loads, boundaries, and materials in a pre-processor.
2. Edit the MSC.Marc input file (.dat) to configure it to use the CCA routines.
3. Edit the marc.f user-defined subroutine file.
4. Edit or verify the CCA configuration file.
5. Run the analysis.

Note that due to how MSC.Marc is structured, a full process modelling simulation requires two analyses. First, the thermal problem must be solved, where the heat transfer, heat generation due to reaction, etc., is used to determine the temperature history of each integration point in the model. Next, the stress problem is solved, using the temperature history, and optionally the volume fraction history, to determine the displacements and stress due to the cure cycle. A different input file is required for each of these analyses. Note that as of this writing, flow analyses cannot be run with MSC.Marc and COMPRO CCA.

This two-step analysis is achieved by writing the temperature history to a separate file during the thermal analysis. This file is then used as the basis of the temperature history for the stress analysis.

Simplified analyses can be performed, in which just the thermal analysis is run (if the analysis is being run to determine exotherms, for example), or in which just the stress analysis is run (in this case ignoring any thermodynamic effects and assuming that the part is the same temperature as the air temperature, everywhere).

4.2 Setting up the analysis

Create the geometry and element mesh in MSC.Patran or your favourite pre-processor. In the process modelling of composites, especially in 3D analyses, quite a bit of care and planning must be put into the pre-processing.

Warning: Currently, the mesh must be in either metres or inches, depending on your unit system (SI or USCS). If your mesh is in not in these units, the results will not be meaningful. Particularly, in USCS (Imperial), be certain that all units (including derived units) are in inches, pounds, Fahrenheit / Rankine, seconds, BTU, etc.

Very little modification of the MSC.Marc input file is required to invoke the CCA material models.

For thermal analyses, specify the material using the ANISOTROPIC card. The material ID must be a number greater than 1000 for CCA materials.

Note: The CCA library will assume that any material with an id greater than 1000 is a CCA material.

This number must correspond to the number specified in the cca-configuration.xml file for the given material. You must also ensure that the flag to call the user subroutine ANKOND is set to 1. Set the density and specific heat to 1.0, since these material properties will be set by the CCA libraries.

ANISOTROPIC

```
$ mid, ankond flag
1000,1
$ density, Cp, emissivity
1.0,1.0,0.
$ element list
Part
```

For stress analyses, the material must be specified using the HYPOELASTIC card. As in thermal analyses, the material ID must be a number greater than 1000 for CCA materials, and must correspond to the number specified in the cca-configuration.xml file. You must ensure that the flag to call the user subroutines ANEXP and HYPELA2 are set to 1 and 3 respectively. Set the density to 1.0 and the CTE to 0.0 as these will be set by the CCA libraries.

HYPOELASTIC

```
$ # of data sets, unit file #
1,
$ MID, IUANEXP, IUHYPELA2,
1,1,3
$ rho (stress), CTE, K, Cp, R, rho (HT), E
1.,0.,
$ element list
Part
```

The initial state of the elements must be specified. These statements set the initial values of the CCA state variables (see Table 2 at the end of section 3.1). Note that MSC.Marc reserves temperature as State Variable #1, so the state variable numbers in the INITIAL STATE cards are offset by +1 from the numbers shown in Table 2. Also, MSC.Marc cannot be used for flow analyses, so the largest state variable number is 6. Be certain to set the initial state for each state variable of each integration point in each element that has a CCA material, using method #4 (see the MSC.Marc documentation Vol C, Chapter 3):

INITIAL STATE

```
2,4,1,
0.001
Allelements
1 TO 8
```

...

INITIAL STATE

```
7,4,1,
0.
```

```
Allelements
1 TO 8
```

In thermal analyses, the resin curing causes internal heat generation. This is captured in the user subroutines by invoking distributed fluxes through the `DIST FLUXES` card. For all elements containing user materials, specify a distributed flux of type 3, with a magnitude of 1.0 and a flux index of 1:

```
DIST FLUXES
1
3,1.0,1
Part
```

Also in thermal analyses, the temperatures can be applied directly using the `FORCDT` card in combination with programming the cure cycle in the `marc.f` file, or by specifying heat transfer coefficients using the `FILMS` card and programming the cure cycle in the `marc.f` file.

An example of specifying HTCs on a surface:

```
FILMS
1
5,2.0383E-5,1.,1,1
Bagside
```

4.3 Running the analysis

The COMPRO Common Component Architecture has been implemented into MSC.Marc as a set of routines called by the user subroutines `film`, `forcdt`, `hypela2`, `anexp`, `flux`, `ankond`, `uspcht`, and others. MSC.Marc/COMPRO is distributed as two compiled object files that MSC.Marc combines with its own engine at run time. In Windows, these file are `libcca.lib` and `cca-MarcInterface.lib`. These object files must be placed in an appropriate directory prior to running your analysis. An appropriate location might be `C:\MSC.Software\MSC.Marc\2005r3\lib`.

Additionally, the MSC.Marc installation must be made aware of these additional libraries prior to running analyses. The file `include.bat` must be modified. Open this file in a text editor, and around line 235, there is a line that begins with:

```
SET SYSLIBS=%DYNAMICDLLS% /release /subsystem:console
```

Modify this line such that the two CCA libraries are referenced. If you placed the two LIB files in the `lib` directory, you can do this by adding `%LIBDIR%\libcca.lib %LIBDIR%\cca-MarcInterface.lib` to the end of the line. Typically this means that the line looks like (note that this is all on one line):

```
SET SYSLIBS=%DYNAMICDLLS% /release /subsystem:console /incremental:no
/nodfaultlib:libc.lib /nodfaultlib:msvcrt.lib %IMAGEHLP_LIB% libmmt.lib
libifcoremt.lib %LIBIRC_LIB% %LIBDIR%\libcca.lib %LIBDIR%\cca-MarcInterface.lib
```

The directory containing your MSC.Marc `.dat` file is your job directory. In this directory, you need the input files for the run, as well as the `cca-configuration.xml` file and the `marc.f` file (supplied in the installation package).

The `marc.f` file is a Fortran 77 file that is used to link MSC.Marc to the user subroutines. This file is also used to specify the temperature cycles through either the `film` subroutine (for specifying a time variant temperature sink on boundaries that are specified with the `FILMS` keyword) or the `forcdt` subroutine (for specifying absolute temperatures on nodes affected by the `FORCDT` keyword). This file must be edited with a text editor to specify the temperature

cycle in one (or both) of these subroutines. The other subroutines in the file are essential for linking the CCA library to MSC.Marc and should not be modified.

The `cca-configuration.xml` file establishes the link between the material models selected in the MSC.Marc input file, and the CCA data files. It also sets some global parameters related to the run. The `cca-configuration.xml` file must be edited (as of this writing, with a text editor). The details of setting the CCA options can be found in Chapter 4 of the COMPRO Common Component Architecture User's Guide.

Once these initial steps have been completed, the run can be started by typing something similar to the following at the command line:

```
run_marc.bat -j jobname -u marc.f -b n
```

If everything is working fine, then immediately after reporting the license checkout, you will see a boilerplate message:

```
*****
*
* COMPRO Common Component Architecture
*
* Revision 631 (2007-05-28 11:45:41) UID: 1181067037
*
* Convergent Manufacturing Technologies Inc.
* 6190 Agronomy Road, Suite 403
* Vancouver, British Columbia, Canada
* support@convergent.ca or http://www.convergent.ca
*
*****
```

If this message is not displayed, then the CCA libraries have not been properly integrated into MSC.Marc.

The `cca-configuration.xml` File

The CCA components are configured through the use of an XML file, `cca-configuration.xml`. This file must be present in the job directory of any CCA analysis. At the moment, this file must be edited using a text editor to set the various configuration options before any CCA analysis is performed.

5.1 A Brief Note About XML Files

XML stands for eXtensible Markup Language, and is similar to HTML, the format that most worldwide web documents use. All CCA configuration and data files are written in XML. At the moment, these files must be edited using a text editor (such as Notepad, EMACS, vi, etc.). If you are unfamiliar with XML, the following section will give you an overview of “rules” that should be observed when working with these files.

5.1.1 Elements

The core structural forms in XML files are elements. Elements are essentially containers for data. Consider composite materials: a basic prepreg consists of a fibre component and a matrix component. A material element, then, would contain information relating to that particular material, whether it was a fibre, matrix, or prepreg.

Elements are specified by tags. A tag is just a way of indicating what element you are dealing with. There are start tags and end tags, which make up the container that is an element.

Start tags always begin with a “<”, and at some point end with a “>”. The start tag for the material element could be:

```
<material>
```

End tags usually begin with “</” followed by the element name, then a “>”. The end tag for the material element could be:

```
</material>
```

Thus, everything between the start tag and end tag is information that belongs to that particular element.

Sometimes, elements are contained in a single set of “<” and “>”. Such an element might be:

```
<material ...some information... />
```

Note that this is a complete element. The end tag is replaced with just “/” and the “>” for the start tag is not used. This form is used when no data (text or other elements) appear between the start and end tags.

5.1.2 Data

Data can be passed to elements in two ways. One way is to simply include text or other elements between the start and end tags of a parent element:

```
<material>AS4/3501-6
  <matrix>3501-6</matrix>
  <fibre>AS4</fibre>
</material>
```

In the above example, the material “AS4/3501-6” has a matrix element “3501-6” and a fibre element “AS4”. The XML is formatted with the indentation to emphasize the hierarchical or object-oriented nature of the data.

Data can also be passed to elements using attributes. Attributes appear within the start tags of elements:

```
<material name="AS4/3501-6" matrix="3501-6" fibre="AS4" />
```

In this example, the same data is passed to the material element, just in a different form. The material has a name attribute equal to “AS4/3501-6”, a matrix attribute equal to “3501-6”, and a fibre attribute equal to “AS4”. Attribute names appear as-is, but the attribute data must be enclosed within quotation marks. Note that the end tag is the “/” form, since no other data is contained within the start and end tags.

These forms can be combined:

```
<material name="AS4/3501-6" >
  <matrix name="3501-6" />
  <fibre name="AS4" />
</material>
```

Again, the same information is passed to the material element, just in a different structure.

It is this last syntax that CCA uses. Attributes are used to describe elements, but often elements are contained within other elements (as in the matrix and fibre elements being contained within the material element in the above example).

5.1.3 Comments

Comments in XML files are contained within special start and end tags. Comments are always preceded by the “<!--” tag, and are always terminated by the “-->” tag. Everything in a comment container is ignored by CCA. Note that due to limitations in the XML parser (the code that reads the XML files), comments cannot exceed 8192 characters in length. For longer comments, use multiple “<!-- -->” pairs.

```
<!-- This is a comment -->
```

5.2 Required Elements and Attributes

The file `cca-configuration.xml` requires the following elements and attributes:

5.2.1 <configuration> ... </configuration>

This is the “wrapper” container for the `cca-configuration.xml` file. All the configuration data must be within this container or an error will be written to the screen and the analysis will end.

5.2.2 <directory type= "data" path= "mydata" />

This element specifies the path of the CCA data files, relative to the current (i.e., job) directory. If the path attribute is left as "", then the current directory is assumed.

5.2.3 <directory type= "output" path= "" />

This element specifies the path of the output files, relative to the current (i.e., job) directory. If the path attribute is left as "", then the current directory is assumed.

5.2.4 <model_units system= "SI|Imperial|USCS" />

This element specifies the unit system used in the FE data files (not the CCA data files). Specify only one of the three unit systems. The SI units are m, kg, s, and the Imperial and USCS (US Customary System) are in (not feet), lbs, sec.

5.2.5 <analysis_code engine= "ABAQUS" />

This element specifies the analysis engine. Currently only ABAQUS and MARC are supported.

5.2.6 <debug screen= "false" file= "false" filename= "debug.csv" debug_element= "1" />

The CCA components are still in development. These options allow some debug information to be printed to screen or file. Most users will not use any of these functions. Note that for ABAQUS, the debug element can be specified as the global element number, or as an instance number of the form `PartName.LocalElementID` (e.g., `Part.12`)

5.2.7 <user_material id= "some_id">...</user_material>

Each user material must be specified within a `user_material` container. The id is either a text-based identifier (for ABAQUS) or a numerical identifier (for MSC.Marc). Up to 20 different user materials can be specified, and at a minimum there must be one.

The following elements should be specified within the `user_material` container:

<composite_file name= "AS4-3501-6.xml" />

For each CCA data file that is a composite material, but not for individual matrices or fibres, an element must contain the filename of that material. These files must be located in the directory specified in the `<directory type= "data" path= "" />` element. Without these entries in the `cca-configuration.xml` file, the CCA component will not be able to locate the material data files.

<initial_doc DOC= "0.001" />

For each user material, you must specify the initial degree of cure. For autocatalytic reactions, this is typically non-zero – usually 0.001 or higher if the material has been characterized at CMT.

`<smearing A= "1.0" B= "0.0" />`

For each user material, you must specify the smearing parameters. These parameters are used in determining the effective in-plane properties of composites for thermal and flow analyses. Neither MSC.Marc nor ABAQUS support *composite* sections (or element configurations) so smearing factors must be used for directional material parameters such as conductivity and permeability.

The “A” and “B” parameters account for material properties according to the following relationship (conductivity is used as an example):

$$K_{11} = AK_L + BK_T$$
$$K_{22} = AK_T + BK_L$$

Where K_L is the longitudinal conductivity, and K_T is the transverse conductivity for a single ply. For a unidirectional lay-up, which is the default if nothing is specified, the parameters are A=1.0, B=0. For a cross-ply or quasi-orthotropic lay-up the values are A=0.5, B=0.5.

`<fbc_override Ex="1.0E9" Ey="1.0E9" Ez="1.0E4"/>`

In flow analyses, any layup other than unidirectional will result in complex in-plane fibre bed properties. As previously discussed, the analysis engines do not support “composite” sections in flow analyses, so the fibre bed stiffness properties must be given to the engine separately. At the moment (subject to change), the tensile properties of the fibre bed are input in this manner.

The CCA Data File Formats

The COMPRO Common Component Architecture stores the material data in XML files. These files can be edited by any text or XML editor, but for the most part, users should not be required to change anything in these files. However, in case some tweaking of the data is required, this section describes the XML format of the data files.

The CCA data files describe materials. As such, the file must start with the material element:

```
<material unique_name="unique_name" general_name="general_name"
display_name="display_name" type="fibre|matrix|composite" last_update="date" ...
last_update_by="person" comments="some comments"> ... </material>
```

Attribute	Required?	Description
unique_name	Required	The unique name should be the same as the filename, and for traceability, unique. For AS4 fibre, this might be “AS4-2005-04-07”.
general_name	Optional	This is a generic name for the material. For AS4 fibre, this might be “AS4 fibre”.
display_name	Optional	In the future, CCA components will be built to handle the data files and avoid manual editing of the XML. In these future components, the material will be displayed by its display_name.
type	Required	This attribute tells the CCA components what type of material the file describes. The type must be one of “fibre”, “matrix”, or “composite”.
last_update	Optional	For traceability, the last date the file was updated.
last_update_by	Optional	For traceability, the name of the person last to update the file.
comments	Optional	Short comments related to the material.

After specifying the material element and its attributes, the remaining elements and attributes depend on the type of material being specified. For fibre and matrix material types, the elements consist of particular material properties, and parameters used in the models that describe the material properties. These elements are of the general form (note that the material_property label is generic; actual material property elements are density, cure_shrinkage, etc.):

```
<material_property model="default" number_of_parameters= "8">
<parameter name="Nominal_density" parameter_number="1" units="kg/m3" value="1000." />
<parameter name="Factor1" parameter_number="2" units="none" value="0.3" />
...
<parameter name="Factor7" parameter_number="8" units="none" value="0.123" />
</material_property>
```

Material_property element

Attribute	Required?	Description
model	Required	Either the name or the number of the material property model. May also be "none", depending on the material property. See the COMPRO Common Component Architecture Model Reference for details.
number_of_parameters	Optional	The total number of parameters associated with this material model. Note that this must be specified if the model name is something other than "none".

Blank line required after table.

parameter element

Attribute	Required?	Description
name	Required	The name of the parameter, for reference. This should match the same name given in the COMPRO Common Component Architecture Model Reference.
parameter_number	Required	The number associated with the parameter name.
units	Required	The units for the value specified. See the COMPRO Common Component Architecture Model Reference for unit specification examples.
value	Required	The value of the parameter

Some material properties are at the composite level rather than the constituent level. The "composite" material element may have a number of `material_property` and `parameter` elements.

6.1 Material Type="Composite"

Composite material types contain all the information for creating a composite from its constituents. This means specifying the constituents, the micromechanics model, and some lamina properties. The elements are described below:

6.1.1 <constituent volume_fraction= "[0-1]" type= "fibre|matrix" > name </constituent>

At the moment, two constituents should be specified, both a fibre and a matrix, otherwise the micromechanics models will behave unpredictably. The volume fraction of each should be specified, and the total volume fraction should sum to 1.0. The name of the constituent must match the unique name of the desired constituent.

6.1.2 <micromechanics model= "default | bogetti_and_gillespie | modified_rosen_and_hashin | chamis" />

The micromechanics model must be specified. The default model is the same as the Bogetti and Gillespie model.

6.1.3 <lamina_properties>

This element is a generic container for several lamina-level properties. It has no data and no attributes, but must be present. Within the `lamina_properties` element, the following material properties must be specified (with models, or "none" as the model name) according to the `material_property` template described above:

```

<LumpedDensity ... >
<LumpedCp ... >
<LumpedK ... >
<LumpedCTE ... >
<permeability ... >
<fibre_bed_compaction ... >

```

An example type="composite" data file follows:

```

<material unique_name="as4-3501-6-2005-04-06" general_name="as4/3501-6"
  display_name="AS4/3501-6 Prepreg" type="composite" last_update="2005-04-06"
  last_update_by="AMF" comments="Initial assembly of AS4/3501-6 in the CCA XML
  data structure.">
  <constituent volume_fraction="0.558" type="fibre">as4-2005-04-05 </constituent>
  <constituent volume_fraction="0.442" type="matrix">3501-6-2005-04-05 </constituent>
  <micromechanics model="default" />
  <lamina_properties>
    <LumpedDensity model="none" />
    <LumpedCp model="none" />
    <LumpedK model="none" />
    <LumpedCTE model="default" number_of_parameters="8">
      <parameter name="NomCTE11" parameter_number="1" units="m/(m K)" value="0.000" />
      <parameter name="CTE11TF" parameter_number="2" units="m/(m K2)" value="2.000E-09" />
      <parameter name="CTE11AF" parameter_number="3" units="1/K" value="0.00" />
      <parameter name="NomCTE22" parameter_number="4" units="m/(m K)" value="2.250E-05" />
      <parameter name="CTE22TF" parameter_number="5" units="m/(m K2)" value="2.500E-08" />
      <parameter name="CTE22AF" parameter_number="6" units="1/K" value="-3.500E-06" />
      <parameter name="CTET0" parameter_number="7" units="C" value="20.00" />
      <parameter name="CTEAL0" parameter_number="8" units="none" value="0.000" />
    </LumpedCTE>
    <permeability model="4" number_of_parameters="4">
      <parameter name="Rf" parameter_number="1" units="m" value="4.0000E-06" />
      <parameter name="Va" parameter_number="2" units="none" value="8.100E-01" />
      <parameter name="kx" parameter_number="3" units="none" value="0.7" />
      <parameter name="kz" parameter_number="4" units="none" value="0.2" />
    </permeability>
    <fibre_bed_compaction model="2" number_of_parameters="4">
      <parameter name="E1" parameter_number="1" units="Pa" value="2.00E+11" />
      <parameter name="G13" parameter_number="2" units="Pa" value="5.000E+07" />
      <parameter name="E3T" parameter_number="3" units="Pa" value="1.000E+06" />
      <parameter name="Theta" parameter_number="4" units="rad" value="0.0" />
      <table coll="EPS3" units1="none" col2="SIG3" units2="Pa">
        <datarow coll="0.000E+00" col2="0.000E+00" />
        <datarow coll="-3.500E-02" col2="-7.295E+04" />
        <datarow coll="-5.240E-02" col2="-1.930E+05" />
        <datarow coll="-7.930E-02" col2="-4.560E+05" />
        <datarow coll="-1.006E-01" col2="-9.130E+05" />
        <datarow coll="-1.181E-01" col2="-2.000E+06" />
        <datarow coll="-2.000E-01" col2="-7.000E+06" />
        <datarow coll="-5.000E-01" col2="-25.70E+06" />
      </table>
    </fibre_bed_compaction>
  </lamina_properties>
</material>

```

6.2 Material Type= "Fibre"

The "fibre" material elements contain only `material_property` elements. These elements follow the `material_property` template discussed above. The material models for the fibre type are:

```
<density ... >
<specific_heat ... >
<conductivity ... >
<cte ... >
<elastic_properties ... >
```

An example `type="fibre"` data file follows:

```
<material unique_name="as4-2004-12-03" general_name="as4" display_name="AS4 Fibre" type="fibre">
  <density model="default" number_of_parameters="3">
    <parameter name="NominalDensity" parameter_number="1" units="kg/m3" value="1.790E+03" />
    <parameter name="T0" parameter_number="2" units="C" value="20." />
    <parameter name="Tf" parameter_number="3" units="kg/(m3 K)" value="0." />
  </density>
  <specific_heat model="default" number_of_parameters="3">
    <parameter name="NominalCp" parameter_number="1" units="J/(kg K)" value="750." />
    <parameter name="T0" parameter_number="2" units="C" value="0." />
    <parameter name="Tf" parameter_number="3" units="J/(kg K2)" value="2.05" />
  </specific_heat>
  <conductivity model="default" number_of_parameters="5">
    <parameter name="NominalKt" parameter_number="1" units="W/(m K)" value="7.69" />
    <parameter name="NominalKl" parameter_number="2" units="W/(m K)" value="2.4" />
    <parameter name="T0" parameter_number="3" units="C" value="0." />
    <parameter name="Tft" parameter_number="4" units="W/(m K2)" value="1.560E-02" />
    <parameter name="Tfl" parameter_number="5" units="W/(m K2)" value="5.070E-02" />
  </conductivity>
  <cte model="ctel" number_of_parameters="5">
    <parameter name="NominalCTEt" parameter_number="1" units="m/(m K)" value="-3.6E-07" />
    <parameter name="NominalCTEl" parameter_number="2" units="m/(m K)" value="4.83E-06" />
    <parameter name="T0" parameter_number="3" units="C" value="20." />
    <parameter name="Tft" parameter_number="4" units="m/(m K2)" value="0." />
    <parameter name="Tfl" parameter_number="5" units="m/(m K2)" value="0." />
  </cte>
  <elastic_properties model="default" number_of_parameters="15">
    <parameter name="E11" parameter_number="1" units="Pa" value="2.100E+11" />
    <parameter name="E11_refT" parameter_number="2" units="C" value="0" />
    <parameter name="E11_Tf" parameter_number="3" units="Pa/C" value="0" />
    <parameter name="E33" parameter_number="4" units="Pa" value="1.724E+10" />
    <parameter name="E33_refT" parameter_number="5" units="C" value="0" />
    <parameter name="E33_Tf" parameter_number="6" units="Pa/C" value="0" />
    <parameter name="G13" parameter_number="7" units="Pa" value="2.760E+10" />
    <parameter name="G13_refT" parameter_number="8" units="C" value="0" />
    <parameter name="G13_Tf" parameter_number="9" units="Pa/C" value="0" />
    <parameter name="nu13" parameter_number="10" units="Pa" value="0.2" />
    <parameter name="nu13_refT" parameter_number="11" units="C" value="0" />
    <parameter name="nu13_Tf" parameter_number="12" units="Pa/C" value="0" />
    <parameter name="nu23" parameter_number="13" units="Pa" value="0.25" />
    <parameter name="nu23_refT" parameter_number="14" units="C" value="0" />
    <parameter name="nu23_Tf" parameter_number="15" units="Pa/C" value="0" />
  </elastic_properties>
</material>
```

6.3 Material Type= "Matrix"

The "matrix" material elements contain only `material_property` elements. These elements follow the `material_property` template discussed above. The material models for the matrix type are:

```
<cure_kinetics ... >
<modulus_development ... >
<poissons_ratio ... >
<cure_shrinkage ... >
<density ... >
<viscosity ... >
<specific_heat ... >
<conductivity ... >
```

An example `type="matrix"` data file follows:

```
<material unique_name="3501-6-2005-04-05" general_name="3501-6" display_name="3501-6 Resin"
type="matrix">
  <cure_kinetics model="ck1" number_of_parameters="9">
    <parameter name="HeatOfReaction" parameter_number="1" units="J/kg" value="4.740E+05" />
    <parameter name="A1" parameter_number="2" units="1/s" value="3.503E+07" />
    <parameter name="DELE1" parameter_number="3" units="J/mol" value="8.070E+04" />
    <parameter name="A2" parameter_number="4" units="1/s" value="-3.357E+07" />
    <parameter name="DELE2" parameter_number="5" units="J/mol" value="7.780E+04" />
    <parameter name="A3" parameter_number="6" units="1/s" value="3.267E+03" />
    <parameter name="DELE3" parameter_number="7" units="J/mol" value="5.660E+04" />
    <parameter name="BB" parameter_number="8" units="none" value="0.4700" />
    <parameter name="AlphaC" parameter_number="9" units="none" value="0.3000" />
  </cure_kinetics>
  <modulus_development model="rml" number_of_parameters="7">
    <parameter name="Er0" parameter_number="1" units="Pa" value="3.500E+06" />
    <parameter name="Erinf" parameter_number="2" units="Pa" value="3.500E+09" />
    <parameter name="Gamma" parameter_number="3" units="none" value="0.0" />
    <parameter name="Alpha1" parameter_number="4" units="none" value="0.3500" />
    <parameter name="Alpha2" parameter_number="5" units="none" value="1.0000" />
    <parameter name="TZero" parameter_number="6" units="C" value="0.0" />
    <parameter name="TFact" parameter_number="7" units="1/C" value="0.0" />
  </modulus_development>
  <poissons_ratio model="pr1" number_of_parameters="5">
    <parameter name="NuR0" parameter_number="1" units="none" value="0.37" />
    <parameter name="ANuR" parameter_number="2" units="none" value="0.00" />
    <parameter name="NuRT0" parameter_number="3" units="C" value="0.00" />
    <parameter name="BNuR" parameter_number="4" units="none" value="0.00" />
    <parameter name="NuRA1" parameter_number="5" units="none" value="0.00" />
  </poissons_ratio>
  <cure_shrinkage model="cs1" number_of_parameters="4">
    <parameter name="VSHI" parameter_number="1" units="none" value="0.03" />
    <parameter name="ALC1" parameter_number="2" units="none" value="0.0000" />
    <parameter name="ALC2" parameter_number="3" units="none" value="0.5000" />
    <parameter name="ASH" parameter_number="4" units="none" value="0.03" />
  </cure_shrinkage>
  <density model="default" number_of_parameters="5">
    <parameter name="NominalDensity" parameter_number="1" units="kg/m3" value="1.300E+03" />
    <parameter name="T0" parameter_number="2" units="C" value="20." />
    <parameter name="Tf" parameter_number="3" units="kg/(m3 K)" value="0." />
    <parameter name="a0" parameter_number="4" units="none" value="0." />
    <parameter name="af" parameter_number="5" units="kg/m3" value="0." />
  </density>
</material>
```

```
</density>
<viscosity model="1" number_of_parameters="4">
  <parameter name="GelDegreeOfCure" parameter_number="1" units="none" value="0.25" />
  <parameter name="MuInf" parameter_number="2" units="Pa s" value="4.600E-17" />
  <parameter name="U" parameter_number="3" units="J/mol" value="1.145E+05" />
  <parameter name="K" parameter_number="4" units="none" value="1.4580E+01" />
</viscosity>
<specific_heat model="default" number_of_parameters="5">
  <parameter name="NominalCp" parameter_number="1" units="J/(kg K)" value="1260." />
  <parameter name="T0" parameter_number="2" units="C" value="20." />
  <parameter name="Tf" parameter_number="3" units="J/(kg K2)" value="0.0" />
  <parameter name="a0" parameter_number="4" units="none" value="0." />
  <parameter name="af" parameter_number="5" units="J/(kg K)" value="0.0" />
</specific_heat>
<conductivity model="default" number_of_parameters="5">
  <parameter name="NominalK" parameter_number="1" units="W/(m K)" value="0.167" />
  <parameter name="T0" parameter_number="2" units="C" value="20." />
  <parameter name="Tf" parameter_number="3" units="W/m K2)" value="0.0" />
  <parameter name="a0" parameter_number="4" units="none" value="0.0" />
  <parameter name="af" parameter_number="5" units="W/(m K)" value="0.0" />
</conductivity>
</material>
```

6.4 Creating a Simple Material Datafile

In order to use a material with CCA, a set of data files must be created for that material. This example demonstrates how to generate the data files for a simple tooling material (7075-T6 Aluminum). This material has no cure behaviour, which simplifies the data files required. Note that all references to “CCA” in this document include RAVEN, since RAVEN uses CCA for its material property calculations.

6.4.1 Material Data

Before creating the data files, the properties of the material must be obtained. This data can often be found in material datasheets, but the data for this example was taken from a textbook¹. In the case that the data is unavailable, experiments may be necessary, or approximate values can be used from similar materials. Be aware that any approximate values used can impact the resulting analyses to varying degrees.

6.4.2 Destination Format

All CCA material data is stored in XML files. Chapter 4 provides an overview of XML files, and Chapter 5 describes the format used for composite, fibre, and matrix material files for CCA. Although simple materials such as breathers aren't composites made up of a fibre and matrix, it is necessary to include all three data files for each material. For simple materials such as this, the matrix is used to represent the material, while the fibre is a placeholder, and the composite is given a fibre volume fraction of 0.

6.4.3 File Naming Convention

Every material has two components: resin and fibres. Material filenames for composite and each of its component should follow these conventions:

¹Callister, W. D., 2000, Materials Science and Engineering: An Introduction, John Wiley & Sons, Inc.

- Resin: {Resin Name}-v{version number}.xml (e.g. TORAY2510-v1.xml)
- Fibre: {Fibre Name}-v{version number}.xml (e.g. AS4-v1.xml)
- Composite: {ResinName}-{FibreName}-{type}-v{version number}.xml (e.g. TORAY2510-AS4-Fabric-v1.xml)

Where types are:

- Tape
- Fabric
- PW (for plain weave fabric)
- other standard weave abbreviations
- NeatResin

For example:

- 3501-6-v1.xml
- AS4-v4.xml
- 3501-6-AS4-Fabric-v1.xml

Also:

- Resin and fibre names should be written in upper case letters. (e.g. TORAY2510.xml)
- The descriptors should be written as one word in title case format. (e.g. NeatResin)
- NullFibre-v1.xml file is a dummy fibre file used when the fibre volume fraction is zero (i.e. for neat resins). The fibre name appears as “Null” in the material file name: e.g. TORAY2510-Null-NeatResin.xml

6.4.4 Matrix Data File

Although simple materials don't exhibit cure advancement, and the property variation associated with that, cure-related `material_property` elements are still necessary in the material data file, but will be defined to provide constant values. The models required in this case are:

- conductivity
- cte
- cure_kinetics
- density
- modulus_development
- poissons_ratio
- specific_heat
- cure_shrinkage
- viscosity

XML Declaration

In keeping with XML standards, an XML declaration is required in the material file:

```
<?xml version="1.0" encoding="UTF-8"?>
```

File Header

Each material file should have a header. The header includes general Convergent contact information, information about the models included, the developer that produced the data, and the most recent date that each model has been modified. The following template should be used for the header of each material file:

```
<!-- ===== -->
<!-- Convergent Common Component Architecture (CCA) for COMPRO/Raven -->
<!-- -->
<!-- Convergent Manufacturing Technologies Inc. -->
<!-- 6190 Agronomy Road, Suite 403 -->
<!-- Vancouver, British Columbia, Canada -->
<!-- support@convergent.ca or www.convergent.ca -->
<!-- -->
<!-- (c)2008 Convergent Manufacturing Technologies, Inc -->
<!-- All rights reserved. -->
<!-- ===== -->
<!-- -->
<!--      Models      Developer      Date      -->
<!--      ...          ...          ...      -->
<!-- -->
<!-- -->
<!-- Comments: -->
<!-- -->
<!-- -->
<!-- -->
<!-- ===== -->
<!-- RELEASE/PERMISSIONS -->
<!--      Release:  Open -->
<!--      Internal: Open -->
<!--      Customer: N/A -->
<!-- -->
<!-- ===== -->
```

Material Element

The basis of the data is a root XML `material` element, which defines some general attributes of the material, such as material type and the various material names:

```
<material type="matrix" unique_name="Aluminum-7075-T6-v1"
  display_name="Aluminum-7075-T6" general_name="Aluminum-7075-T6-v1">
  ...
</material>
```

Next, an element is added for each model used by the material (as listed above).

Conductivity

The conductivity of this material is a constant 130 W/(m K). There is no constant value model for conductivity, but the default model can be provided with values which prevent any variation with temperature or degree of cure:

```
<conductivity model="default" number_of_parameters="5">
  <parameter units="W/(m K)" parameter_number="1" name="NominalK" value="130"/>
  <parameter units="C" parameter_number="2" name="T0" value="0"/>
  <parameter units="W/(m K)" parameter_number="3" name="Tf" value="0"/>
  <parameter units="none" parameter_number="4" name="a0" value="0"/>
  <parameter units="none" parameter_number="5" name="af" value="0"/>
</conductivity>
```

CTE

The CTE of this material is a constant 2.34E-05 1/K:

```
<cte model="ctel" number_of_parameters="5">
  <parameter units="1/K" parameter_number="1" name="NominalCTE" value="2.340E-05"/>
  <parameter units="1/K2" parameter_number="2" name="aCTEr" value="0"/>
  <parameter units="C" parameter_number="3" name="T0" value="0"/>
  <parameter units="1/K" parameter_number="4" name="bCTEr" value="0"/>
  <parameter units="none" parameter_number="5" name="a0" value="0"/>
</cte>
```

Cure Kinetics

Although this material doesn't cure, a cure kinetics model with a minimum degree of cure of 1 is included:

```
<cure_kinetics model="ck1" number_of_parameters="9">
  <parameter units="J/kg" parameter_number="1" name="HeatOfReaction" value="1"/>
  <parameter units="1/s" parameter_number="2" name="A1" value="0"/>
  <parameter units="J/mol" parameter_number="3" name="DELE1" value="0"/>
  <parameter units="1/s" parameter_number="4" name="A2" value="0"/>
  <parameter units="J/mol" parameter_number="5" name="DELE2" value="0"/>
  <parameter units="1/s" parameter_number="6" name="A3" value="0"/>
  <parameter units="J/mol" parameter_number="7" name="DELE3" value="0"/>
  <parameter units="none" parameter_number="8" name="BB" value="0"/>
  <parameter units="none" parameter_number="9" name="AlphaC" value="0"/>
  <constraint variable="doc" value="1.0" isMinimum="True" severity_model="1"
    initial_severity="100.0" severity_parameters="0.0;100.0;" />
</cure_kinetics>
```

Density

Finally, a constant density of 2.80E+03 kg/m³ can be defined in a similar way:

```
<density model="default" number_of_parameters="5">
  <parameter units="kg/m3" parameter_number="1" name="NominalDensity" value="2.80E+03"/>
  <parameter units="C" parameter_number="2" name="T0" value="0"/>
  <parameter units="kg/(m3 K)" parameter_number="3" name="Tf" value="0"/>
  <parameter units="none" parameter_number="4" name="a0" value="0"/>
</density>
```

```
<parameter units="none" parameter_number="5" name="af" value="0"/>
</density>
```

Modulus Development

The modulus of 71GPa is also set:

```
<modulus_development model="rml" number_of_parameters="7">
  <parameter units="Pa" parameter_number="1" name="Er0" value="7.100E+10"/>
  <parameter units="Pa" parameter_number="2" name="Erinf" value="7.100E+10"/>
  <parameter units="none" parameter_number="3" name="Gamma" value="0"/>
  <parameter units="none" parameter_number="4" name="Alpha1" value="0"/>
  <parameter units="none" parameter_number="5" name="Alpha2" value="1"/>
  <parameter units="C" parameter_number="6" name="TZero" value="0"/>
  <parameter units="1/K" parameter_number="7" name="TFact" value="0"/>
</modulus_development>
```

Poisson's Ratio

Poisson's Ratio Model 2 provides a single-value model using a constant bulk modulus:

```
<poissons_ratio model="pr2" number_of_parameters="1">
  <parameter units="none" parameter_number="1" name="NuInf" value="0.33"/>
</poissons_ratio>
```

Specific Heat

As with conductivity, a constant value of specific heat (9.60E+02 J/(kg K)) can be defined using the default model with appropriate constants:

```
<specific_heat model="default" number_of_parameters="5">
  <parameter units="J/(kg K)" parameter_number="1" name="NominalCp" value="9.60E+02"/>
  <parameter units="C" parameter_number="2" name="T0" value="0"/>
  <parameter units="J/(kg K)" parameter_number="3" name="Tf" value="0"/>
  <parameter units="none" parameter_number="4" name="a0" value="0"/>
  <parameter units="none" parameter_number="5" name="af" value="0"/>
</specific_heat>
```

Cure Shrinkage

Since no cure occurs, no cure shrinkage constants are needed:

```
<cure_shrinkage model="cs1" number_of_parameters="4">
  <parameter units="none" parameter_number="1" name="VSHI" value="0"/>
  <parameter units="none" parameter_number="2" name="ALC1" value="0"/>
  <parameter units="none" parameter_number="3" name="ALC2" value="1"/>
  <parameter units="none" parameter_number="4" name="ASH" value="0"/>
</cure_shrinkage>
```

Viscosity

Once again, setting some of the model constants allows a varying model to be used for a constant viscosity. As this material is a solid, a large viscosity value of 1E+13 is used:

```
<viscosity model="default" number_of_parameters="4">
  <parameter units="none" parameter_number="1" name="GelDegreeOfCure" value="0.5"/>
  <parameter units="unknown" parameter_number="2" name="MuInf" value="1E+13"/>
  <parameter units="unknown" parameter_number="3" name="U" value="0"/>
  <parameter units="unknown" parameter_number="4" name="K" value="0"/>
</viscosity>
```

Complete Data File

The entire matrix data file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ===== -->
<!-- Convergent Common Component Architecture (CCA) for COMPRO/Raven -->
<!-- ----- -->
<!-- Convergent Manufacturing Technologies Inc. -->
<!-- 6190 Agronomy Road, Suite 403 -->
<!-- Vancouver, British Columbia, Canada -->
<!-- support@convergent.ca or www.convergent.ca -->
<!-- ----- -->
<!-- (c)2008 Convergent Manufacturing Technologies, Inc -->
<!-- All rights reserved. -->
<!-- ===== -->
<!-- ----- -->
<!--          Models          Developer          Date          -->
<!--          ...              ...              ...              -->
<!-- ----- -->
<!-- Comments: Aluminum 7075-T6 using properties taken from -->
<!--             Callister, W.D., 2000, Materials Science and -->
<!--             Engineering: An Introduction, John Wiley & Sons -->
<!--             Inc. -->
<!-- ----- -->
<!-- RELEASE/PERMISSIONS -->
<!--   Release:   Open -->
<!--   Internal:  Open -->
<!--   Customer:  N/A -->
<!-- ----- -->
<material type="matrix" unique_name="Aluminum-7075-T6-v1"
  display_name="Aluminum-7075-T6" general_name="Aluminum-7075-T6-v1">
  <conductivity model="default" number_of_parameters="5">
    <parameter units="W/(m K)" parameter_number="1" name="NominalK" value="130"/>
    <parameter units="C" parameter_number="2" name="T0" value="0"/>
    <parameter units="W/(m K)" parameter_number="3" name="Tf" value="0"/>
    <parameter units="none" parameter_number="4" name="a0" value="0"/>
    <parameter units="none" parameter_number="5" name="af" value="0"/>
  </conductivity>
  <cte model="ctel" number_of_parameters="5">
    <parameter units="1/K" parameter_number="1" name="NominalCTE" value="2.340E-05"/>
```

```

    <parameter units="1/K2" parameter_number="2" name="aCTEr" value="0"/>
    <parameter units="C" parameter_number="3" name="T0" value="0"/>
    <parameter units="1/K" parameter_number="4" name="bCTEr" value="0"/>
    <parameter units="none" parameter_number="5" name="a0" value="0"/>
</cte>
<cure_kinetics model="ck1" number_of_parameters="9">
    <parameter units="J/kg" parameter_number="1" name="HeatOfReaction" value="1"/>
    <parameter units="1/s" parameter_number="2" name="A1" value="0"/>
    <parameter units="J/mol" parameter_number="3" name="DELE1" value="0"/>
    <parameter units="1/s" parameter_number="4" name="A2" value="0"/>
    <parameter units="J/mol" parameter_number="5" name="DELE2" value="0"/>
    <parameter units="1/s" parameter_number="6" name="A3" value="0"/>
    <parameter units="J/mol" parameter_number="7" name="DELE3" value="0"/>
    <parameter units="none" parameter_number="8" name="BB" value="0"/>
    <parameter units="none" parameter_number="9" name="AlphaC" value="0"/>
    <constraint variable="doc" value="1.0" isMinimum="True" severity_model="1"
        initial_severity="100.0" severity_parameters="0.0;100.0;" />
</cure_kinetics>
<density model="default" number_of_parameters="5">
    <parameter units="kg/m3" parameter_number="1" name="NominalDensity" value="2.80E+03"/>
    <parameter units="C" parameter_number="2" name="T0" value="0"/>
    <parameter units="kg/(m3 K)" parameter_number="3" name="Tf" value="0"/>
    <parameter units="none" parameter_number="4" name="a0" value="0"/>
    <parameter units="none" parameter_number="5" name="af" value="0"/>
</density>
<modulus_development model="rml" number_of_parameters="7">
    <parameter units="Pa" parameter_number="1" name="Er0" value="7.100E+10"/>
    <parameter units="Pa" parameter_number="2" name="Erinf" value="7.100E+10"/>
    <parameter units="none" parameter_number="3" name="Gamma" value="0"/>
    <parameter units="none" parameter_number="4" name="Alpha1" value="0"/>
    <parameter units="none" parameter_number="5" name="Alpha2" value="1"/>
    <parameter units="C" parameter_number="6" name="TZero" value="0"/>
    <parameter units="1/K" parameter_number="7" name="TFact" value="0"/>
</modulus_development>
<poissons_ratio model="pr2" number_of_parameters="1">
    <parameter units="none" parameter_number="1" name="NuInf" value="0.33"/>
</poissons_ratio>
<specific_heat model="default" number_of_parameters="5">
    <parameter units="J/(kg K)" parameter_number="1" name="NominalCp" value="9.60E+02"/>
    <parameter units="C" parameter_number="2" name="T0" value="0"/>
    <parameter units="J/(kg K)" parameter_number="3" name="Tf" value="0"/>
    <parameter units="none" parameter_number="4" name="a0" value="0"/>
    <parameter units="none" parameter_number="5" name="af" value="0"/>
</specific_heat>
<cure_shrinkage model="cs1" number_of_parameters="4">
    <parameter units="none" parameter_number="1" name="VSHI" value="0"/>
    <parameter units="none" parameter_number="2" name="ALC1" value="0"/>
    <parameter units="none" parameter_number="3" name="ALC2" value="1"/>
    <parameter units="none" parameter_number="4" name="ASH" value="0"/>
</cure_shrinkage>
<viscosity model="default" number_of_parameters="4">
    <parameter units="none" parameter_number="1" name="GelDegreeOfCure" value="0.5"/>
    <parameter units="unknown" parameter_number="2" name="MuInf" value="1E+13"/>
    <parameter units="unknown" parameter_number="3" name="U" value="0"/>
    <parameter units="unknown" parameter_number="4" name="K" value="0"/>
</viscosity>
</material>

```

6.4.5 Fibre Data File

Although the fibre constituent will be given a volume fraction of 0, it is still necessary to define a material. A general-purpose null fibre material is included with CCA, called NullFibre-v1.xml, which will be used for this example.

6.4.6 Composite Data File

Now that the matrix and placeholder fibre materials are defined, a composite data file is required to represent the combined constituents. As described earlier, a fibre volume fraction of 0 will be used. As with the other data files, an XML declaration and file header (not shown), as well as a material root XML element are required:

```
<material type="composite" unique_name="TOOLING-Aluminum-7075-T6-Null-v1"
  display_name="Tooling: Aluminum-7075-T6" general_name="TOOLING-Aluminum-7075-T6-Null-v1">
  ...
</material>
```

Each constituent is then described using a `constituent` element, with attributes for the volume fraction and type. The content of this element is the unique_name of the constituent material data file:

```
<constituent volume_fraction="0" type="fibre">NullFibre-v1</constituent>
<constituent volume_fraction="1" type="matrix">Aluminum-7075-T6-v1</constituent>
```

Finally, a `property_mask` element is included to tell CCA which properties this material can output. This mask is used both for data generation, as well as by RAVEN to selectively display/hide certain materials which are unsuitable for certain analyses. The mask is a 32-bit number with different properties enabled by their corresponding bits. The processing mask bits are:

Bit	Property
0	Degree of Cure
1	Cure Rate
2	Tg
3	Resin Modulus
4	Resin Poisson's Ratio
5	Resin Instant CS Factor
6	Resin Instant CTE
7	Resin Viscosity
8	Resin Density
9	Resin Heat Capacity
10	Ply Modulus (L)
11	Ply Modulus (T)
12	Ply Poisson's Ratio (L-T)
13	Ply Shear Modulus (L-T)
14	Ply Effective CS Factor (L)
15	Ply Effective CS Factor (T)
16	Ply Effective CTE (L)
17	Ply Effective CTE (T)
18	Ply Density
19	Ply Heat Capacity
20	Ply Conductivity (L)
21	Ply Conductivity (T)
22	Ply Permeability (L)
23	Ply Permeability (T)
24	Ply Modulus (TT)
25	Ply Poisson's Ratio (L-TT)
26	Ply Poisson's Ratio (T-TT)
27	Ply Shear Modulus (L-TT)
28	Ply Shear Modulus (T-TT)
29	Ply Effective CS Factor (TT)
30	Ply Effective CTE (TT)
31	Ply Conductivity (TT)

For example, if a mask including Degree of Cure, Cure Rate and Resin Viscosity was required, the value would be $2^0+2^1+2^7=131$. A suitable mask for Aluminum 7075-T6, for use in all CCA analyses, would be:

```
<property_mask mask="16777208"/>
```

The completed composite data file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ===== -->
<!-- Convergent Common Component Architecture (CCA) for COMPRO/Raven -->
<!-- -->
<!-- Convergent Manufacturing Technologies Inc. -->
<!-- 6190 Agronomy Road, Suite 403 -->
<!-- Vancouver, British Columbia, Canada -->
<!-- support@convergent.ca or www.convergent.ca -->
<!-- -->
<!-- (c)2008 Convergent Manufacturing Technologies, Inc -->
<!-- All rights reserved. -->
<!-- ===== -->
<!-- -->
<!-- Models Developer Date -->
```

```

<!--      ...      ...      ...      -->
<!--      -->
<!--      -->
<!-- Comments: Aluminum 7075-T6 using properties taken from      -->
<!--           Callister, W.D., 2000, Materials Science and      -->
<!--           Engineering: An Introduction, John Wiley & Sons      -->
<!--           Inc.      -->
<!--      -->
<!-- =====      -->
<!-- RELEASE/PERMISSIONS      -->
<!--   Release:   Open      -->
<!--   Internal:  Open      -->
<!--   Customer:  N/A      -->
<!--      -->
<!-- =====      -->
<material type="composite" unique_name="TOOLING-Aluminum-7075-T6-Null-v1"
  display_name="Tooling: Aluminum-7075-T6" general_name="TOOLING-Aluminum-7075-T6-Null-v1">
  <comments/>
  <constituent volume_fraction="0" type="fibre">NullFibre-v1</constituent>
  <constituent volume_fraction="1" type="matrix">Aluminum-7075-T6-v1</constituent>
  <property_mask mask="16777208"/>
</material>

```