

**CAPE-OPEN**

Expanding Process Modeling Capability  
through  
Software Interoperability Standards

---

## Errata and clarifications

### Thermodynamic and Physical Properties v1.0

---



[www.colan.org](http://www.colan.org)

---

## ARCHIVAL INFORMATION

---

Filename	Outstanding issues 1.0.doc
Authors	CO-LaN consortium
Status	Public
Date	March 2008
Version	version 1.0.8
Number of pages	16
Versioning	version 1.0.1 edited by Michel Pons (CO-LaN) and Werner Drewitz (BASF AG)
	Version 1.0.2 commented by Jasper van Baten
	Version 1.0.3 edited by Werner Drewitz
	Version 1.0.4 edited by Michel Pons
	Version 1.0.5 edited by Werner Drewitz
	Version 1.0.6 edited by Werner Drewitz: added paragraph 4.3 and new flashtypes with HF and SF.
	Version 1.0.7 edited by Jasper van Baten: added paragraph 3.8
	Version 1.0.8 edited by Michel Pons (proof reading)
Additional material	
Web location	<a href="http://www.colan.org/Spec%2010/Outstanding_issues_1.0.pdf">http://www.colan.org/Spec%2010/Outstanding_issues_1.0.pdf</a>
Implementation specifications version	1.0
Comments	

---

## IMPORTANT NOTICES

---

### **Disclaimer of Warranty**

CO-LaN documents and publications include software in the form of *sample code*. Any such software described or provided by CO-LaN --- in whatever form --- is provided "as-is" without warranty of any kind. CO-LaN and its partners and suppliers disclaim any warranties including without limitation an implied warrant or fitness for a particular purpose. The entire risk arising out of the use or performance of any sample code --- or any other software described by the CAPE-OPEN Laboratories Network --- remains with you.

Copyright © 2001-2008 CO-LaN. All rights are reserved unless specifically stated otherwise.

CO-LaN is a not for profit organization established under French law of 1901.

### **Trademark Usage**

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in CO-LaN publications, and the authors are aware of a trademark claim, the designations have been printed in caps or initial caps.

Microsoft, Microsoft Word, Visual Basic, Visual Basic for Applications, Internet Explorer, Windows and Windows NT are registered trademarks and ActiveX is a trademark of Microsoft Corporation.

Netscape Navigator is a registered trademark of Netscape Corporation.

Adobe Acrobat is a registered trademark of Adobe Corporation.

---

## SUMMARY

---

This document describes clarifications and corrects errors in the CAPE-OPEN Thermodynamic and Physical Properties interface specification version 1.0.

---

## ACKNOWLEDGEMENTS

---

Many individuals and their organizations have contributed to this document. Richard Szczepanski from Infochem Computer Services Ltd., Werner Drewitz from BASF AG, Jasper van Baten from AmsterCHEM and Michel Pons from CO-LaN were among the main contributors.

---

# CONTENTS

---

<b>1.</b>	<b>INTRODUCTION.....</b>	<b>7</b>
<b>2.</b>	<b>AUDIENCE .....</b>	<b>7</b>
<b>3.</b>	<b>CLARIFICATIONS.....</b>	<b>7</b>
3.1	ICapeThermoMaterialObject::PhaseIds (section 4.9.2.2) .....	7
3.2	List of Flashtypes (sections 4.9.2.8 and 4.9.4.5).....	9
3.3	What is “Cape Variant (String Array)”? .....	12
3.4	Lower Bound of an Array - 0 or 1?.....	12
3.5	Different Return Types for GetComponentConstant and GetProp? .....	12
3.6	How to handle multiple Liquid Phases? .....	12
3.7	Is heat of formation included in entropy and enthalpy? .....	12
3.8	How to enumerate property packages from the registry? .....	13
<b>4.</b>	<b>ERRATA .....</b>	<b>14</b>
4.1	“Vapor” .....	14
4.2	Section 4.12.1 Constant Properties Identifiers, page 75.....	14
4.3	GetComponentConstant returns array of Variants .....	15
<b>5.</b>	<b>BIBLIOGRAPHY.....</b>	<b>16</b>

## 1. Introduction

This document lists errata and clarifications that have been raised and added after the formal approval process for the CAPE-OPEN Thermodynamic and Physical Properties interface specification version 1.0.

The intent is to clarify the interface specification where needed and to pinpoint what is missing in the specification document.

## 2. Audience

This document is intended primarily for software developers who want to build CAPE-OPEN Thermo System, Property Package, Equilibrium Server and Calculation Routine components. It is also intended for the developers of other software components, such as Unit Operations and Reaction Packages, which make use of these Thermodynamic and Physical Properties software components. Finally, it is intended for developers of CAPE-OPEN-compliant Process Modeling Environments because it relates to how the interfaces are to be used to implement communication between the environment and the external software components.

This document is not intended for end-users of CAPE-OPEN software components or process simulation software.

## 3. Clarifications

### 3.1 ICapeThermoMaterialObject::PhaseIds (section 4.9.2.2)

Issue 1: May PhaseIds method return “Overall” as an existing phase?

Answer 1: It is explicitly stated in the textual interface specification document (section 4.9.2.2):  
“PhaseIds ... returns the phases existing in the MO at that moment. The Overall phase and multiphase identifiers cannot be returned by this method”.

On the other hand in section 4.9.4.9 on GetPhaseList method:  
”GetPhaseList... provides the list of the supported phases. When supported, the Overall phase and multiphase identifiers must be returned by this method.”

See also section 6.8.2 “Existence of a phase”.

This point is very important because otherwise the communication between PME and PropertyPackage may fail when “Overall” is given back and then this phaseId is used for e.g. calcProp.

Issue 2: It is not clear in the definition of phaseIds() (section 4.9.2.2) what the valid phaseIds are.

**Proposal:** A reference should be added: For a list of valid phaseIds see 4.13.1 .

**Adopted:** Section 4.9.2.2 has been modified as follows

#### 4.9.2.2.PhaseIds

**Interface Name**      **ICapeThermoMaterialObject**

**Method Name**        PhaseIds

**Returns**              CapeError

---

#### **Description**

It returns the phases existing in the Material Object at that moment.

---

#### **Arguments**

<b>Name</b>	<b>Type</b>	<b>Description</b>
[out, retval] *phaseIds	CapeVariant (String Array)	List of phases

---

#### **Notes**

For a list of valid phaseIds see – besides Overall phase:

The Overall phase and multiphase (i.e. VaporLiquid) identifiers cannot be returned by this method. This point is very important because otherwise the communication between PME and PropertyPackage will fail when “Overall” is sent back and this phaseId is used for e.g. a CalcProp call. See notes on [Existence of a phase](#) for more information.

### 3.2 List of Flashtypes (sections 4.9.2.8 and 4.9.4.5)

Issue: The textual interface specification refers to a list of Flashtypes (e.g. p.68: "...according to list of flash types in specification." but there is no list...

Adopted: The correct flash type list is the following:

<b>Flash Type</b>	<b>Descriptions</b>
TP	Temperature-Pressure
PH	Pressure-Enthalpy
TH	Temperature-Enthalpy
TVF	Temperature-Vapor Fraction
PVF	Pressure-Vapor Fraction

Sections 4.9.2.8 and 4.9.4.5 read now the following

#### 4.9.2.8 CalcEquilibrium

**Interface Name**      **ICapeThermoMaterialObject**

**Method Name**        CalcEquilibrium

**Returns**              CapeError

---

#### Description

This method is responsible for delegating flash calculations to the associated Property Package or Equilibrium Server. It must set the amounts, compositions, temperature and pressure for all phases present at equilibrium, as well as the temperature and pressure for the overall mixture, if not set as part of the calculation specifications. See [CalcProp and CalcEquilibrium](#) for more information.

---

#### Arguments

Name	Type	Description
[in] flashType	CapeString	Flash calculation type.
[in] props	CapeVariant (String Array)	Properties to be calculated at equilibrium. emptyVariant for no properties. If a list, then the property values should be set for each phase present at equilibrium.

---

#### Notes

The following flash calculation types are allowed:

Flash Type	Descriptions
TP	Temperature-Pressure
PH	Pressure-Enthalpy
TH	Temperature-Enthalpy
TVF	Temperature-Vapor Fraction
PVF	Pressure-Vapor Fraction

Some Thermo Systems / Property Package providers extend the standard and also support flash types like PT, PS, TS, HS, UV, SV, PV and TV and also using the identifiers “enthalpyF” or “entropyF” instead of “enthalpy” and “entropy” giving more flash types: PHF, THF, HFVF, HFS, PSF, TSF, HFSF and SFV. Since these flash types are not part of the standard, one should not assume they are supported.

#### 4.9.4.5 CalcEquilibrium

**Interface Name**      **ICapeThermoPropertyPackage**

**Method Name**        CalcEquilibrium

**Returns**              CapeError

---

#### Description

Method is responsible for calculating/delegating flash calculation requests. It must set the amounts, compositions, temperature and pressure for all phases present at equilibrium, as well as the temperature and pressure for the overall mixture, if not set as part of the calculation specifications. See [CalcProp and CalcEquilibrium](#) for more information.

---

#### Arguments

Name	Type	Description
[in] *materialObject	CapeInterface	The MaterialObject
[in] flashType	CapeString	Flash calculation type.
[in] Props	CapeVariant (String Array)	Properties to be calculated at equilibrium. emptyVariant for no properties. If a list, then the property values should be set for each phase present at equilibrium.

---

#### Notes

The following flash calculation types are allowed:

Flash Type	Descriptions
TP	Temperature-Pressure
PH	Pressure-Enthalpy
TH	Temperature-Enthalpy
TVF	Temperature-Vapor Fraction
PVF	Pressure-Vapor Fraction

Some Thermo Systems / Property Package providers extend the standard and also support flash types like PT, PS, TS, HS, UV, SV, PV and TV and also using the identifiers “enthalpyF” or “entropyF” instead of “enthalpy” and “entropy” giving more flash types: PHF, THF, HFVF, HFS, PSF, TSF, HFSF and SFV. Since these flash types are not part of the standard, one should not assume they are supported.

### 3.3 What is “Cape Variant (String Array)”?

Issue: In the text often appears the type “Cape Variant (String Array)” e.g. in 4.9.2.1 for argument CompIds. Its signification has been questioned.

Answer: It corresponds to the idle term “CapeArrayString“ and “CapeArrayT” in chapter 5.1 “Elementary types” of the document “Methods&Tools Integrated Guidelines v1.0”. It can be understood as a **safearray of BSTR**. Safearrays of Variants are only used in few cases like in GetComponentConstant where in the same array double constants as well as string constants (e.g. casRegistryNumber) can also be returned.

### 3.4 Lower Bound of an Array - 0 or 1?

Issue: Should the lower bound of arrays be 0 or 1?

Answer: It is not fixed in the textual interface specification so both is possible and therefore it is necessary to check the lower bound. But it is recommended to use 0 as lower bound because otherwise VBA applications will have problems.

### 3.5 Different Return Types for GetComponentConstant and GetProp?

Issue: Why does GetComponentConstant return a “CapeVariant (Variant Array)” while GetProp returns a “CapeVariant (Double Array)”?

Answer: GetComponentConstant can also return string values (i.e. CAS Registry number) while GetProp returns only numerical values i.e. doubles. Hence the Variant Array.

### 3.6 How to handle multiple Liquid Phases?

Issue: There is only one identifier “Liquid” for liquid phases in the list 4.13.1 Phase Details. How can multiple liquid phases be handled?

Answer: To do that it was decided that all identifiers that start with “Liquid” are allowed. So valid identifiers are “Liquid”, “Liquid2”, “LiquidWater” or anything else.

Adopted: list 4.13.1 reads now: see 4.1 this document!

### 3.7 Is heat of formation included in entropy and enthalpy?

Issue: It is not stated in the spec whether entropy and enthalpy includes the heat of formation, so it is up to decision of the implementer of the Property Package. It may or may not be included!

Answer: It was decided to introduce new identifiers “entropyF” and “enthalpyF” for entropy and enthalpy including heat of formation. The reference state is defined by the Property Package and has to be consistent for all compounds.

For appropriate flash calculations (PH, PS, HS) only the (overall) identifiers “enthalpy” and “entropy” are to be used for the analogue properties as before. A corresponding note is added to 4.9.2.8 and 4.9.4.5, see 3.1 this doc.

Adopted: list 4.12.3 Non-constant Properties (or Model Dependent Properties) includes now:

Density	Density **	mol/m <sup>3</sup>
Enthalpy	Enthalpy*	J/mol
enthalpyF	Enthalpy*, including heat of formation	J/mol
Entropy	Entropy*	J/(mol K)
entropyF	Entropy*, including heat of formation	J/(mol K)
Energy	Internal energy*	J/mol

### 3.8 How to enumerate property packages from the registry?

**Issue:** The CAT IDs defined for stand-alone Property Packages and Thermo Systems allow one to find implementations from the Windows registry. To do this, the “Classes” sub-keys in the registry need to be enumerated. There are 3 version of such keys: HKEY\_CLASSES\_ROOT, HKEY\_CURRENT\_USER\Software\Classes and HKEY\_LOCAL\_MACHINE\Software\Classes. Which of those keys is to be used?

**Answer:** The HKEY\_CURRENT\_USER\Software\Classes key stores the COM classes that have been installed for the current user. The HKEY\_LOCAL\_MACHINE\Software\Classes key stores system wide COM classes. To modify the HKEY\_LOCAL\_MACHINE\Software\Classes, one may require administration rights. Generally one does not require administration rights to modify HKEY\_CURRENT\_USER\Software\Classes. It is therefore up to the installer of the software component to choose whether it is installed for the user only, or for all users. Property packages should be listed from both the HKEY\_CURRENT\_USER\Software\Classes and HKEY\_LOCAL\_MACHINE\Software\Classes keys, with the keys in the former taking precedence over the same keys in the latter.

The COM classes provided in HKEY\_CLASSES\_ROOT provide exactly that union. It is therefore generally sufficient for a CAPE-OPEN application to list components that appear in HKEY\_CLASSES\_ROOT. For more information, see <http://msdn2.microsoft.com/en-us/library/ms724475.aspx>

## 4. Errata

### 4.1 “Vapor”

Issue: In some places in the document vapour is used although mostly vapor appears.

Answer: The correct form is “Vapor”. The form “Vapour” is wrong.

Adopted: e.g. list 4.13.1 reads now:

#### **4.13.1 Phase Details**

Permitted phases have been restricted to the following:

Phase	Description
Vapor	Vapor phase
Liquid	Liquid phase
LiquidX	Liquid phase X
Solid	Solid phase
SolidX	Solid phase X
Overall	All phases

X is a placeholder for anything: nothing, “2”, “3”, “Water” or anything else. So valid identifiers are "Liquid", "Liquid2", "Liquid3", "LiquidWater" or anything else.

### 4.2 Section 4.12.1 Constant Properties Identifiers, page 75

The units for **vanderwaalsArea** and **vanderwaalsVolume** should be **m<sup>2</sup>/mol** and **m<sup>3</sup>/mol** respectively instead of m<sup>2</sup> and m<sup>3</sup>.

So the list of pure compound constant properties should end with:

Identifiers	Meaning	SI Units
.....		
vanderwaalsVolume	Van der Waals Volume	<b>m<sup>3</sup>/mol</b>
vanderwaalsArea	Van der Waals Area	<b>m<sup>2</sup>/mol</b>

### 4.3 GetComponentConstant returns array of Variants

In the code example on page 15 in the specs the method GetComponentConstant returns CapeArrayDouble:

```
CapeError GetComponentConstant(in CapeArrayString propList, out CapeArrayDouble propVals);
```

**This is wrong!**

GetComponentConstant returns CapeVariant (Variant Array), see also paragraph 4.9.43 p. 44 in the specs. So the code snippet should be:

```
CapeError GetComponentConstant(in CapeArrayString propList, out CapeVariant propVals);
```

## **5. Bibliography**

1. “Thermodynamic and Physical Properties interface specification” v1.0, CO-LaN, 2002.
2. “Methods&Tools Integrated Guidelines” v1.0, CO-LaN, 2002.