



---

# OpenClovis Software Development Kit (SDK) Service Description and API Reference for Rule Based Engine (RBE) Service

For OpenClovis SDK Release 2.3 V0.4  
Document Revision Date: March 08, 2007

---

**Copyright © 2007 OpenClovis Inc.**

**All rights reserved**

This document contains proprietary and confidential information of OpenClovis Inc., and may not be used, modified, copied, reproduced, disclosed or distributed in whole or in part except as authorized by OpenClovis Inc. This document is intended for informational use and planning purposes only. All planned features, specifications, and content are subject to change without notice.

**Third-Party Trademarks**

Sun, Sun Microsystems, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. UNIX is a registered trademark of The Open Group. Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries. CLEI is a trademark of Telcordia Technologies, Inc. Adobe, Acrobat, and Acrobat Reader are registered trademarks of Adobe Systems, Inc. All other trademarks, service marks, product names, or brand names mentioned in this document are the property of their respective owners.

**Government Use**

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR 12.212 (Commercial Computer Software-Restricted Rights) and DFAR 227.7202 (Rights in Technical Data and Computer Software), as applicable.

**Note:** This document is not subject of the GPL license, even if you have obtained this document as a part of the GPL-ed version of OpenClovis SDK.

# Contents

<b>1</b>	<b>Functional Overview</b>	<b>1</b>
1.1	Interaction with other components . . . . .	1
<b>2</b>	<b>Service Model</b>	<b>3</b>
<b>3</b>	<b>Service APIs</b>	<b>5</b>
3.1	Type Definitions . . . . .	5
3.1.1	clRuleExprT . . . . .	5
3.1.2	clRuleExprFlagsT . . . . .	5
3.2	Functional APIs . . . . .	7
3.2.1	clRuleExprAllocate . . . . .	7
3.2.2	clRuleExprDeallocate . . . . .	8
3.2.3	clRuleExprAppend . . . . .	9
3.2.4	clRuleExprDuplicate . . . . .	10
3.2.5	clRuleExprEvaluate . . . . .	11
3.2.6	clRuleDoubleExprEvaluate . . . . .	12
3.2.7	clRuleExprLocalConvert . . . . .	13
3.2.8	clRuleExprConvert . . . . .	14
3.2.9	clRuleExprFlagsSet . . . . .	15
3.2.10	clRuleExprOffsetSet . . . . .	16
3.2.11	clRuleExprMaskSet . . . . .	17
3.2.12	clRuleExprValueSet . . . . .	18
3.2.13	clRuleExprFlagsGet . . . . .	19
3.2.14	clRuleExprOffsetGet . . . . .	20
3.2.15	clRuleExprMaskGet . . . . .	21
3.2.16	clRuleExprValueGet . . . . .	22
3.2.17	clRuleExprMemLenGet . . . . .	23
3.2.18	clRuleExprPack . . . . .	24
3.2.19	clRuleExprUnpack . . . . .	25

## CONTENTS

---

3.2.20 clRuleExprPrint . . . . .	26
<b>4 Service Management Information Model</b>	<b>27</b>
<b>5 Service Notifications</b>	<b>29</b>
<b>6 Debug CLIs</b>	<b>31</b>

# Chapter 1

## Functional Overview

The OpenClovis Rule Based Engine (RBE) provides a mechanism to create rules to be applied to the system instance data, based on simple expressions.

An expression consists of a mask and a value. These expressions are evaluated on user data and a boolean value is generated for the decision process.

For instance, RBE is used by the Event Service to support filter-based subscriptions. The event is published with a pattern that is matched against the filter provided by the subscribers. Only those subscribers that match successfully are notified. The RBE library provides simple bit-based matching based on the flags specified.

### 1.1 Interaction with other components

The RBE engine forms a general purpose library that can be used by any other software component. The RBE library does not depend on any component.



## **Chapter 2**

# **Service Model**

TBD





# Chapter 3

## Service APIs

### 3.1 Type Definitions

#### 3.1.1 ClRuleExprT

```
typedef struct RuleExpr{
    CIUInt8T flags;
    CIUInt8T len;
    CIUInt16T offset;
    struct RuleExpr *next;
    CIUInt8T Byte [1]
    CIUInt32T Int [1]
    BI_u
} ClRuleExprT;
```

The structure, `ClRuleExprT`, contains the rule to filter data. It provides the expression definition. The attributes of the structure are:

- *flags* - Architecture and other flags.
- *len* - Expression length in multiples of four bytes.
- *\*next* - Multiple RBEs can be chained to build complex expression.
- *offset* - Offset with the data where expression is applied.
- *Byte, Int* - Expression mask and value array.

#### 3.1.2 ClRuleExprFlagsT

```
typedef enum{
    CL_RULE_LITTLE_END = 0x1,
    CL_RULE_BIG_END = 0x2,
    CL_RULE_NON_ZERO_MATCH = 0x4,
    CL_RULE_MATCH_EXACT = 0x8,
    CL_RULE_EXPR_CHAIN_AND = 0x10,
} ClRuleExprFlagsT;
```

The values of the `ClRuleExprFlagsT` enumeration, contain the expression qualification. It filters the data based on the rule. The attributes of the enumeration are:

- *CL\_RULE\_LITTLE\_END* - Little endian
- *CL\_RULE\_BIG\_END* - Big endian.
- *CL\_RULE\_NON\_ZERO\_MATCH* - Non-zero match.
- *CL\_RULE\_MATCH\_EXACT* - Exact match.
- *CL\_RULE\_EXPR\_CHAIN\_AND* - Appending two RBE expressions with AND relation.

## 3.2 Functional APIs

### 3.2.1 clRuleExprAllocate

#### clRuleExprAllocate

**Synopsis:**

Allocates RBE expression.

**Header File:**

clRuleApi.h

**Syntax:**

```
ClRcT clRuleExprAllocate(  
    CL_IN ClUInt8T len,  
    CL_OUT ClRuleExprT** ppExpr);
```

**Parameters:**

**len:** (in) Length of the mask or value in multiples of four bytes.

**ppExpr:** (out) Allocated expression is returned in this parameter.

**Return values:**

**CL\_OK:** The function executed successfully.

**CL\_RULE\_RC(CL\_ERR\_NULL\_POINTER):** ppExpr contains a NULL pointer.

**CL\_RULE\_RC(CL\_ERR\_NO\_MEMORY):** An error has occurred.

**Description:**

This function allocates an RBE expression and initializes it appropriately.

**Library File:**

libCIUtils

**Related Function(s):**

[clRuleExprDeallocate](#)

### 3.2.2 `clRuleExprDeallocate`

#### `clRuleExprDeallocate`

**Synopsis:**

Frees memory or structures used by an RBE expression.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprDeallocate(  
    CL_IN ClRuleExprT* pExpr);
```

**Parameters:**

***Expr***: (in) RBE expression to be freed.

**Return Values:**

***CL\_OK***: The function executed successfully.

***CL\_RULE\_RC(CL\_ERR\_NULL\_POINTER)***: `pExpr` contains a NULL pointer.

***CL\_RULE\_RC(CL\_ERR\_NO\_MEMORY)***: An error has occurred.

**Description:**

This function frees the memory or structures used by RBE expression.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprAllocate](#)

## 3.2 Functional APIs

---

### 3.2.3 clRuleExprAppend

#### clRuleExprAppend

##### Synopsis:

Appends a RBE expression.

##### Header File:

clRuleApi.h

##### Syntax:

```
ClRcT clRuleExprAppend (  
    CL_IN ClRuleExprT* pFirstExpr,  
    CL_IN ClRuleExprT* pNextExpr);
```

##### Parameters:

**pFirstExpr:** (in) RBE expression on which pNextExpr needs to be appended.

**pNextExpr:** (in) RBE expression that is appended to pFirstExpr.

##### Return values:

**CL\_OK:** The function executed successfully.

**CL\_RULE\_RC(CL\_ERR\_NULL\_POINTER):** pFirstExpr or pNextExpr contains a NULL pointer.

##### Description:

This function appends an expression to another. It is used to create complex expressions by combining existing expressions.

##### Library File:

libCIUtils

##### Related Function(s):

[clRuleExprAllocate](#)

### 3.2.4 `clRuleExprDuplicate`

#### `clRuleExprDuplicate`

**Synopsis:**

Duplicates a RBE expression.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
CL_RCT clRuleExprDuplicate(  
    CL_IN ClRuleExprT* pSrcExpr,  
    CL_OUT ClRuleExprT** ppDstExpr);
```

**Parameters:**

***pSrcExpr***: (in) Source RBE expression to Copy.

***ppDstExpr***: (out) Pointer to a new copy of the expression.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RC\_ERROR***: An error has occurred.

**Description:**

This function makes a copy of the given RBE expression. The function allocates the required memory that must be freed by the caller using `clRuleExprDeallocate()`.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprAllocate](#), [clRuleExprDeallocate](#)

## 3.2 Functional APIs

---

### 3.2.5 clRuleExprEvaluate

#### clRuleExprEvaluate

##### Synopsis:

Evaluates a complex RBE expression.

##### Header File:

clRuleApi.h

##### Syntax:

```
ClRuleResultT  clRuleExprEvaluate(  
                                CL_IN ClRuleExprT* pExpr,  
                                CL_IN ClUInt32T *pData,  
                                CL_IN int dataLen);
```

##### Parameters:

**pExpr:** (in) RBE expression to be evaluated.

**pData:** (in) Data pointer against which the RBE needs to be compared.

**dataLen:** (in) Length of the data in multiples of 4 bytes.

##### Return values:

**CL\_RULE\_TRUE:** The RBE expression evaluates to `TRUE`.

**CL\_RULE\_FALSE:** The RBE expression evaluates to `FALSE`.

##### Description:

This function evaluates an RBE expression. RBE expression could be a complex expression, that is, multiple expressions can be chained together against a flat buffer `pData`, of length `dataLen`.

##### Library File:

libClUtils

##### Related Function(s):

[clRuleDoubleExprEvaluate](#)

### 3.2.6 `clRuleDoubleExprEvaluate`

#### `clRuleDoubleExprEvaluate`

**Synopsis:**

Evaluates Double RBE expressions.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRuleResultT  clRuleDoubleExprEvaluate
                CL_IN ClRuleExprT* pExpr1,
                CL_IN ClRuleExprT* pExpr2);
```

**Parameters:**

***pExpr1:*** (in) First RBE expression to be evaluated.

***pExpr2:*** (in) Second RBE expression to be evaluated.

***dataLen:*** (in) Length of the data in multiples of 4 bytes.

**Return values:**

***CL\_RULE\_TRUE:*** The RBE expression evaluates to TRUE.

***CL\_RULE\_FALSE:*** The RBE expression evaluates to FALSE.

**Description:**

This function evaluates an RBE expression against another RBE expression. Both expressions are assumed to be of simple type.

**Library File:**

`libCIUtils`

**Related Function(s):**

[`clRuleExprEvaluate`](#)



## 3.2 Functional APIs

---

### 3.2.7 clRuleExprLocalConvert

#### clRuleExprLocalConvert

**Synopsis:**

Converts an RBE expression to local endianness.

**Header File:**

clRuleApi.h

**Syntax:**

```
ClRcT clRuleExprLocalConvert (
                                CL_IN ClRuleExprT* pExpr);
```

**Parameters:**

**pExpr:** (in) RBE expression to be converted.

**Return values:**

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

**Description:**

This function converts an RBE expression to match local endianness.

**Library File:**

libCIUtils

**Related Function(s):**

[clRuleExprConvert](#)

### 3.2.8 clRuleExprConvert

#### clRuleExprConvert

**Synopsis:**

Endian converts a complex RBE expression.

**Header File:**

clRuleApi.h

**Syntax:**

```
ClRcT clRuleExprConvert (
                                CL_IN ClRuleExprT* pExpr);
```

**Parameters:**

**pExpr:** (in) RBE expression to be converted.

**Return values:**

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

**Description:**

This function converts endianness of a complex RBE expression.

**Library File:**

libCIUtils

**Related Function(s):**

[clRuleExprLocalConvert](#)

## 3.2 Functional APIs

---

### 3.2.9 clRuleExprFlagsSet

#### clRuleExprFlagsSet

##### Synopsis:

Sets the flags of an RBE expression.

##### Header File:

clRuleApi.h

##### Syntax:

```
CL_RcT clRuleExprFlagsSet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_IN ClRuleExprFlagsT flags);
```

##### Parameters:

**pExpr:** (in) RBE expression for which flags to be set.

**flags:** (in) Flags to be set.

##### Return values:

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

##### Description:

This function sets the RBE expression flags as specified by the parameter flags.

##### Library File:

libCIUtils

##### Related Function(s):

[clRuleExprFlagsGet](#)

### 3.2.10 clRuleExprOffsetSet

#### clRuleExprOffsetSet

**Synopsis:**

Sets offset of a RBE expression.

**Header File:**

clRuleApi.h

**Syntax:**

```
ClRcT clRuleExprOffsetSet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_IN ClUInt16T offset);
```

**Parameters:**

**pExpr:** (in) RBE expression for which flags to be set.

**offset:** (in) Offset value (multiples of 4 bytes).

**Return values:**

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

**Description:**

This function sets the offset field of an RBE expression as specified by the offset parameter.

**Library File:**

libCIUtils

**Related Function(s):**

[clRuleExprOffsetGet](#)

## 3.2 Functional APIs

---

### 3.2.11 clRuleExprMaskSet

#### clRuleExprMaskSet

##### Synopsis:

Set mask of an RBE expression.

##### Header File:

clRuleApi.h

##### Syntax:

```
clRuleExprMaskSet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_IN ClUInt16T offset,
                                CL_IN ClUInt32T mask);
```

##### Parameters:

**pExpr:** (in) RBE expression for which flags to be set.

**offset:** (in) Offset at which the mask to be set.

**mask:** (in) Mask value to be set.

##### Return values:

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

##### Description:

This function sets the mask field of an RBE expression as specified by the parameter `mask`.

##### Library File:

libCIUtils

##### Note:

Offset is a multiple of 4 bytes and refers to the overall offset. `mask` is set to `expr->mask[offset - expr->offset]`.

##### Related Function(s):

[clRuleExprMaskGet](#)

### 3.2.12 `clRuleExprValueSet`

#### `clRuleExprValueSet`

**Synopsis:**

Sets value of an RBE expression.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprValueSet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_IN ClUInt16T offset,
                                CL_IN ClUInt32T value);
```

**Parameters:**

***pExpr***: (in) RBE expression for which flags are to be set.

***offset***: (in) Offset at which the value is to be set.

***mask***: (in) Mask value to be set.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RC\_ERROR***: An error has occurred.

**Description:**

This function sets the value field of an RBE expression to the value specified by the parameter `value`.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprValueGet](#)

## 3.2 Functional APIs

---

### 3.2.13 `clRuleExprFlagsGet`

#### `clRuleExprFlagsGet`

**Synopsis:**

Retrieves RBE expression flags.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprFlagsGet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_OUT ClRuleExprFlagsT *pFlags);
```

**Parameters:**

***pExpr***: (in) RBE expression for which flags to be set.

***pFlags***: (out) Flags to be returned.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RC\_ERROR***: An error has occurred.

**Description:**

This function retrieves the flags of an RBE expression.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprFlagsGet](#)

### 3.2.14 `clRuleExprOffsetGet`

#### `clRuleExprOffsetGet`

**Synopsis:**

Retrieves RBE expression Offset value.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprOffsetGet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_OUT ClUInt16T *pOffset);
```

**Parameters:**

***pExpr***: (in) RBE expression for which flags to be set.

***pOffset***: (out) Offset to be returned.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RC\_ERROR***: An error has occurred.

**Description:**

This function retrieves the offset value of an RBE expression that is set using `clRuleExprOffsetSet()`.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprOffsetSet](#)



## 3.2 Functional APIs

---

### 3.2.15 clRuleExprMaskGet

#### clRuleExprMaskGet

##### Synopsis:

Retrieves RBE expression mask value.

##### Header File:

clRuleApi.h

##### Syntax:

```
CL_RCT clRuleExprMaskGet (
    CL_IN ClRuleExprT* pExpr,
    CL_IN ClUInt16T offset,
    CL_OUT ClUInt32T *pMask);
```

##### Parameters:

**pExpr:** (in) RBE expression for which flags to be set.

**offset:** (in) Get Mask from this offset.

**pMask:** (out) Mask to be returned.

##### Return values:

**CL\_OK:** The function executed successfully.

**CL\_RC\_ERROR:** An error has occurred.

##### Description:

This function retrieves the mask of an RBE expression that is set using `clRuleExprMaskSet()`.

##### Library File:

libCIUtils

##### Related Function(s):

[clRuleExprMaskSet](#)

### 3.2.16 `clRuleExprValueGet`

#### `clRuleExprValueGet`

**Synopsis:**

Retrieves RBE expression value.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprValueGet (
                                CL_IN ClRuleExprT* pExpr,
                                CL_IN ClUInt16T offset,
                                CL_OUT ClUInt32T *pValue);
```

**Parameters:**

***pExpr*** :(in) RBE expression for which flags to be set.

***offset*** :(in) Get mask from this offset.

***pValue*** :(out) Value to be returned.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RC\_ERROR***: An error has occurred.

**Description:**

This function retrieves the value of an RBE expression that is set using `clRuleExprValueSet()`.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprValueSet](#)

## 3.2 Functional APIs

---

### 3.2.17 `clRuleExprMemLenGet`

#### `clRuleExprMemLenGet`

**Synopsis:**

Retrieves the total memory used by the expression.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClUInt32T clRuleExprMemLenGet (
                                     CL_IN ClRuleExprT* pExpr);
```

**Parameters:**

***pExpr*** (in) RBE expression.

**Return Value:**

Length in bytes needed to pack an expression.

**Description:**

This function returns the memory needed to pack an expression into a contiguous memory location.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprPack](#)

### 3.2.18 `clRuleExprPack`

#### `clRuleExprPack`

**Synopsis:**

Packs an RBE expression into the given memory area.

**Header File:**

`clRuleApi.h`

**Syntax:**

```
ClRcT clRuleExprPack (
                                CL_IN ClRuleExprT* pSrcExpr,
                                CL_IN char *pBuf);
```

**Parameters:**

***pSrcExpr***: (in) Source RBE expression to pack.

***pBuf***: (in) Pointer to the memory location where to pack.

**Return values:**

***CL\_OK***: The function executed successfully.

***CL\_RULE\_RC(CL\_ERR\_NULL\_POINTER)***: *pSrcExpr* or *pBuf* contains a NULL pointer.

**Description:**

This function packs the given RBE expression. The expression is packed at the given memory pointed by *pBuf*. It is assumed that the *pBuf* contains enough space to accommodate *srcExpr*. The caller can call `clRuleExprMemLenGet()` to obtain the memory needed to pack *srcExpr*.

**Library File:**

`libCIUtils`

**Related Function(s):**

[clRuleExprMemLenGet](#), [clRuleExprUnpack](#)

## 3.2 Functional APIs

---

### 3.2.19 clRuleExprUnpack

#### clRuleExprUnpack

##### Synopsis:

Unpacks an RBE expression.

##### Header File:

clRuleApi.h

##### Syntax:

```
ClRcT clRuleExprUnpack(  
                                CL_IN ClRuleExprT* pSrcExpr,  
                                CL_OUT ClRuleExprT** ppDstExpr);
```

##### Parameters:

**pSrcExpr:** (in) Source RBE expression to unpack.

**pDstExpr:** (out) Pointer to a new copy of the expression.

##### Return values:

**CL\_OK:** The function executed successfully.

**CL\_RULE\_RC(CL\_ERR\_NULL\_POINTER):** pSrcExpr or ppDstExpr contains a NULL pointer.

##### Description:

This function unpacks the given RBE expression.

##### Library File:

libCIUtils

##### Related Function(s):

[clRuleExprPack](#)

### 3.2.20 clRuleExprPrint

#### clRuleExprPrint

**Synopsis:**

Prints a complex RBE expression.

**Header File:**

clRuleApi.h

**Syntax:**

```
ClRcT clRuleExprPrint (
                                CL_IN ClRuleExprT* pExpr);
```

**Parameters:**

**pExpr:** (in) RBE expression to be printed.

**Return Value:**

None.

**Description:**

This function prints the given RBE expression.

**Library File:**

libCIUtils

**Related Function(s):**

None.

## **Chapter 4**

# **Service Management Information Model**

TBD





## **Chapter 5**

# **Service Notifications**

TBD



## **Chapter 6**

# **Debug CLIs**

TBD

# Index

clRuleDoubleExprEvaluate, [12](#)  
clRuleExprAllocate, [7](#)  
clRuleExprAppend, [9](#)  
clRuleExprConvert, [14](#)  
clRuleExprDeallocate, [8](#)  
clRuleExprDuplicate, [10](#)  
clRuleExprEvaluate, [11](#)  
clRuleExprFlagsGet, [19](#)  
clRuleExprFlagsSet, [15](#)  
CIRuleExprFlagsT, [5](#)  
clRuleExprLocalConvert, [13](#)  
clRuleExprMaskGet, [21](#)  
clRuleExprMaskSet, [17](#)  
clRuleExprMemLenGet, [23](#)  
clRuleExprOffsetGet, [20](#)  
clRuleExprOffsetSet, [16](#)  
clRuleExprPack, [24](#)  
clRuleExprPrint, [26](#)  
CIRuleExprT, [5](#)  
clRuleExprUnpack, [25](#)  
clRuleExprValueGet, [22](#)  
clRuleExprValueSet, [18](#)