**Specification**

USPD: US.ECO.00012-01 90

Бэкуса — Наура Форма

UGUID:

Status: Draft

Date: November 8, 2021

Version: 1.0

|  |  |
| --- | --- |
| **Authors** | **Company** |
| Vladimir Bashev | PEERF |
|  |  |

Content

[**1.** **Overview** 4](#_Toc182070587)

[**1.1.** **Introduction** 4](#_Toc182070588)

[**1.2.** **Note** 4](#_Toc182070589)

[**1.3.** **Links** 4](#_Toc182070590)

[**2.** **Eco.Toolchain.BNF1 Component** 5](#_Toc182070591)

[**3.** **IEcoToolchainBNF1 Interface** 6](#_Toc182070592)

[**3.1.** **IEcoToolchainBNF1 IDL** 6](#_Toc182070593)

[**3.1.1.** **AddRule function** 6](#_Toc182070594)

[**3.1.2.** **AddComment function** 6](#_Toc182070595)

[**3.1.3.** ***get\_RuleList* function** 7](#_Toc182070596)

[**3.1.4.** ***get\_Alphabet* function** 7](#_Toc182070597)

[**3.1.5.** ***get\_TerminalAlphabet* function** 7](#_Toc182070598)

[**3.1.6.** ***get\_NonTerminalAlphabet* function** 7](#_Toc182070599)

[**3.1.7.** ***Clear* function** 7](#_Toc182070600)

[**4.** **IEcoToolchainBNF1Rule Interface** 8](#_Toc182070601)

[**4.1.** **IEcoToolchainBNF1Rule IDL** 8](#_Toc182070602)

[**4.1.1.** **get\_Name function** 9](#_Toc182070603)

[**4.1.2.** **get\_Id function** 9](#_Toc182070604)

[**4.1.3.** **AddConcatenation function** 9](#_Toc182070605)

[**4.1.4.** **AddAlternative function** 9](#_Toc182070606)

[**4.1.5.** **AddSequenceGroup function** 9](#_Toc182070607)

[**4.1.6.** **AddOptionalSequence function** 9](#_Toc182070608)

[**4.1.7.** **AddSpecialSequence function** 9](#_Toc182070609)

[**4.1.8.** **AddValueRangeAlternatives function** 10](#_Toc182070610)

[**4.1.9.** **AddIncrementalAlternatives function** 10](#_Toc182070611)

[**4.1.10.** **get\_RuleSet function** 10](#_Toc182070612)

[**5.** **IEcoToolchainBNF1Element Interface** 11](#_Toc182070613)

[**5.1.** **IEcoToolchainBNF1Element IDL** 11](#_Toc182070614)

[**5.1.1.** **get\_Name function** 11](#_Toc182070615)

[**5.1.2.** **get\_Id function** 12](#_Toc182070616)

[**5.1.3.** **IsOptional function** 12](#_Toc182070617)

[**5.1.4.** **IsTerminal function** 12](#_Toc182070618)

[**5.1.5.** **IsGroup function** 12](#_Toc182070619)

[**5.1.6.** **IsEpsilon function** 12](#_Toc182070620)

[**5.1.7.** **IsExcep function** 12](#_Toc182070621)

[**5.1.8.** **IsSpecial function** 12](#_Toc182070622)

[**5.1.9.** **IsRepeated function** 12](#_Toc182070623)

[**5.1.10.** **set\_Repetition function** 12](#_Toc182070624)

[**5.1.11.** **get\_LeastOccurrences function** 13](#_Toc182070625)

[**5.1.12.** **get\_MostOccurrences function** 13](#_Toc182070626)

[**5.1.13.** **get\_SpecialType function** 13](#_Toc182070627)

[**5.1.14.** **get\_SequenceSet function** 13](#_Toc182070628)

[**6.** **In development …** 14](#_Toc182070629)

[**Application А** 14](#_Toc182070630)

1. **Overview**

This document describes the requirements for the implementation of the Eco.Toolchain.BNF1 component (Бэкуса — Наура форма).

* 1. **Introduction**

Description.

* 1. **Note**
* Keywords
  1. **Links**

This paragraph contains links to information to help you understand this document:

[] – name of the link

Available by: http://address

1. **Eco.Toolchain.BNF1 Component**

The component has the following description:



1. **IEcoToolchainBNF1 Interface**
   1. **IEcoToolchainBNF1 IDL**

|  |
| --- |
| **ECO IDL** |
| import "IEcoBase1.idl" | | | |
| [  object,  uguid(93102958-9BDA-4637-A249-5674399D4F11),  ] | | | |
| interface IEcoToolchainBNF1 : IEcoUnknown { | | | |
|  | | | |
| IEcoToolchainBNF1Rule\* | | ***AddRule*** | ([in] char\_t\* Name); |
|  | | |  |
| IEcoToolchainBNF1Rule\* | | ***AddComment*** | ([in] char\_t\* Note); |
|  | | |  |
| IEcoList1\* | | ***get\_RuleList*** | ([in] void); |
|  | |  |  |
| IEcoList1\* | | ***get\_Alphabet*** | ([in] void); |
|  | |  |  |
| IEcoList1\* | | ***get\_TerminalAlphabet*** | ([in] void); |
|  | |  |  |
| IEcoList1\* | | ***get\_NonTerminalAlphabet*** | ([in] void); |
|  | |  |  |
| int16\_t | | ***Clear*** | ([in] void); |
|  | | | |
| } | |  |  |

* + 1. **AddRule function**

The function adds a rule to a BNF instance with the given ***Name*** and returns a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface. The returned rule is of type ECO\_BNF\_1\_RT\_RULENAME.

* + 1. **AddComment function**

The function adds a comment to the BNF instance specified by the ***Note*** parameter and returns a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface. The returned rule is of type ECO\_BNF\_1\_RT\_COMMENT.

* + 1. ***get\_RuleList* function**

The function returns a list of all rules as a pointer to the ***IEcoList1***\* list interface. The list element is a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface.

* + 1. ***get\_Alphabet* function**

The function returns the alphabet of all terminal and non-terminal symbols as a pointer to the interface for working with the list ***IEcoList1***\*. A list element is a pointer to a string of ***char\_t***\* type.

* + 1. ***get\_TerminalAlphabet* function**

The function returns the alphabet of all terminal characters as a pointer to the interface for working with the list ***IEcoList1***\*. A list element is a pointer to a string of ***char\_t***\* type.

* + 1. ***get\_NonTerminalAlphabet* function**

The function returns the alphabet of all non-terminal characters as a pointer to the interface for working with the list ***IEcoList1***\*. A list element is a pointer to a string of ***char\_t***\* type.

* + 1. ***Clear* function**

The function clears the BNF instance by removing all rules.

1. **IEcoToolchainBNF1Rule Interface**
   1. **IEcoToolchainBNF1Rule IDL**

|  |
| --- |
| **ECO IDL** |
| import "IEcoBase1.idl" | | | |
|  | | | |
| [  object,  uguid(E67AC8DC-0D75-43B7-8FB1-D5D825DE67A8),  ] | | | |
| interface IEcoToolchainBNF1Rule : IEcoUnknown { | | | |
|  | | | |
| char\_t\* | | ***get\_Name*** | ([in] void); |
|  | | |  |
| int16\_t | | ***get\_Id*** | ([in] void); |
|  | | | |
| ECO\_BNF\_1\_RULE\_TYPE\_e | | ***get\_Type*** | ([in] void); |
|  | | | |
| IEcoToolchainBNF1Element\* | | ***AddConcatenation*** | ([in] int32\_t indexSet, |
|  | |  | [in] char\_t\* name, |
|  | |  | [in] bool\_t except); |
|  | | | |
| IEcoToolchainBNF1Element\* | | ***AddAlternative*** | ([in] char\_t\* name, |
|  | |  | [in] bool\_t except, |
|  | |  | [out] int32\_t\* indexSet); |
|  | |  |  |
| IEcoToolchainBNF1Rule\* | | ***AddSequenceGroup*** | ([in] bool\_t alternation, |
|  | |  | [in] bool\_t except, |
|  | |  | [in] int16\_t least, |
|  | |  | [in] int16\_t most, |
|  | |  | [out] int32\_t\* indexSet); |
|  | |  |  |
| IEcoToolchainBNF1Rule\* | | ***AddOptionalSequence*** | ([in] bool\_t alternation); |
|  | |  | [in] bool\_t except, |
|  | |  | [in] int16\_t least, |
|  | |  | [in] int16\_t most, |
|  | |  | [out] int32\_t\* indexSet); |
|  | |  |  |
| IEcoToolchainBNF1Rule\* | | ***AddSpecialSequence*** | ([in] int32\_t type, |
|  | |  | [in] bool\_t alternation, |
|  | |  | [in] bool\_t except, |
|  | |  | [in] int16\_t least, |
|  | |  | [in] int16\_t most, |
|  | |  | [out] int32\_t\* indexSet); |
|  | |  |  |
| IEcoList1\* | | ***AddValueRangeAlternatives*** | ([in] uchar\_t start, |
|  | |  | [in] uchar\_t end, |
|  | |  | [out] int32\_t\* indexSet); |
|  | | | |
| IEcoList1\* | | ***AddIncrementalAlternatives*** | ([in] IEcoToolchainBNF1Rule\* Rule); |
|  | |  |  |
| IEcoList1\* | | ***get\_RuleSet*** | ([in] void); |
|  | |  |  |
|  | | | |
| } | |  |  |

* + 1. **get\_Name function**

The function returns the name string of the rule.

* + 1. **get\_Id function**

The function returns the identifier of the rule.

* + 1. **AddConcatenation function**

The function adds the rule name specified by the ***Name*** parameter to the ordered sequence of rule names by the ***indexSet*** index of the rule set.

* + 1. **AddAlternative function**

The function adds the rule name specified by the ***Name*** parameter to a new rule set defined by the returned index ***indexSet***.

* + 1. **AddSequenceGroup function**

The function adds a group for elements enclosed in parentheses, returning a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface (interpreted as a single element with strict ordering). The returned rule is of type ECO\_BNF\_1\_RT\_SEQUENCE.

* + 1. **AddOptionalSequence function**

The function adds an optional sequence for elements enclosed in square brackets, returning a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface (treated as a single element). The returned rule is of type ECO\_BNF\_1\_RT\_SEQUENCE.

* + 1. **AddSpecialSequence function**

The function adds a specialized sequence for elements enclosed in special characters of a user-defined type, returning a pointer to the ***IEcoToolchainBNF1Rule***\* grammar rule interface (treated as a single element). The returned rule is of type ECO\_BNF\_1\_RT\_SEQUENCE.

* + 1. **AddValueRangeAlternatives function**

The function adds ranges of variants of digital values by parameters ***start*** start of range and ***end*** end of range.

* + 1. **AddIncrementalAlternatives function**

The function adds additional rule set options to the initial rule.

* + 1. **get\_RuleSet function**

The function returns a set of rules.

1. **IEcoToolchainBNF1Element Interface**
   1. **IEcoToolchainBNF1Element IDL**

|  |
| --- |
| **ECO IDL** |
| import "IEcoBase1.idl" | | | |
|  | | | |
| [  object,  uguid(716BE46A-AEB2-4C65-8A0F-B914B81E9F18),  ] | | | |
| interface IEcoToolchainBNF1Element : IEcoUnknown { | | | |
|  | | | |
| char\_t\* | | ***get\_Name*** | ([in] void); |
|  | | | |
| int16\_t | | ***get\_Id*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsOptional*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsTerminal*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsGroup*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsEpsilon*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsExcept*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsSpecial*** | ([in] void); |
|  | |  |  |
| bool\_t | | ***IsRepeated*** | ([in] void); |
|  | |  |  |
| void | | ***set\_Repetition*** | ([in] int16\_t least, |
|  | |  | [in] int16\_t most); |
|  | |  |  |
| int16\_t | | ***get\_LeastOccurrences*** | ([in] void); |
|  | |  |  |
| int16\_t | | ***get\_MostOccurrences*** | ([in] void); |
|  | |  |  |
| int32\_t | | ***get\_SpecialType*** | ([in] void); |
|  | |  |  |
| IEcoList1\* | | ***get\_SequenceSet*** | ([in] void); |
|  | | | |
| } | |  |  |

* + 1. **get\_Name function**

The function returns the name string of the rule.

* + 1. **get\_Id function**

The function returns the identifier of the rule.

* + 1. **IsOptional function**

The function checks if the element is optional.

* + 1. **IsTerminal function**

The function checks if the element is a terminal.

* + 1. **IsGroup function**

The function checks whether an element is a group of ordered elements.

* + 1. **IsEpsilon function**

The function checks whether an element is an empty set.

* + 1. **IsExcep function**

The function checks whether an element is “except”.

* + 1. **IsSpecial function**

The function checks if the element is a special element***.***

* + 1. **IsRepeated function**

The function checks if the element is repeatable.

* + 1. **set\_Repetition function**

The function sets a variable number of repetitions of the element***.***

* + 1. **get\_LeastOccurrences function**

The function returns the minimum number of repetitions.

* + 1. **get\_MostOccurrences function**

The function returns the maximum number of repetitions***.***

* + 1. **get\_SpecialType function**

The function returns a custom type***.***

* + 1. **get\_SequenceSet function**

The function returns a set of sequences of elements.

1. **In development …**

# **Application А**