

Standard

ECMAScript Internationalization API Specification

ECMA AkiXXX Draft

Ecma/TC39/2012/NN

2 Conformance 3 Normative References. 4 Overview	Page i: [2] Deleted	Norbert Lindenberg	6/28/12 13:20	
3 Normative References	1 Scope			
4 Overview 4.1 Internationalization, Localization, and Globalization 4.2 API Overview 4.3 Implementation Dependencies 5 Notational Conventions 6 Identification of Locales, Time Zones, and Currencies 6.1 Case Sensitivity and Case Mapping 6.2 Language Tags 6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(locale) 6.2.3 CanonicalizeLanguageTag(locale) 6.2.4 DefaultLocale () 6.2.3 CanonicalizeLanguageTag(locale) 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object so a LocaleList 9.1.2 The Intl.LocaleList Constructor Used in a new Expression 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.1.1 Intl.LocaleList.Prototype 9.3 Properties of the Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of Intl.LocaleList Instances 9.4.1 length 10.1 Locale Aparameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)	2 Conformance			
4 Overview 4.1 Internationalization, Localization, and Globalization 4.2 API Overview 4.3 Implementation Dependencies 5 Notational Conventions 6 Identification of Locales, Time Zones, and Currencies 6.1 Case Sensitivity and Case Mapping 6.2 Language Tags 6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(locale) 6.2.3 CanonicalizeLanguageTag(locale) 6.2.4 DefaultLocale () 6.2.3 CanonicalizeLanguageTag(locale) 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object so a LocaleList 9.1.2 The Intl.LocaleList Constructor Used in a new Expression 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.1.1 Intl.LocaleList.Prototype 9.3 Properties of the Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of Intl.LocaleList Instances 9.4.1 length 10.1 Locale Aparameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)	3 Normative References			
4.1 Internationalization, Localization, and Globalization 4.2 API Overview 4.3 Implementation Dependencies 5 Notational Conventions 6 Identification of Locales, Time Zones, and Currencies 6.1 Case Sensitivity and Case Mapping 6.2 Language Tags 6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(locale) 6.2.3 CanonicalizeLanguageTag (locale) 6.2.4 DefaultLocale () 6.3 Currency Codes 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8 The Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Used in a new Expression 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.1.1 Intl.LocaleList.Prototype 9.3 Properties of the Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of the Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of Mith Array Index Names 10.1 Internal Properties of Service Constructors 10.2.1 Intl.LocaleList Constructor 10.2.1 Intl.SocaleList (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)				
4.2 API Overview				
5 Notational Conventions 6 Identification of Locales, Time Zones, and Currencies 6.1 Case Sensitivity and Case Mapping 6.2 Language Tags 6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(locale) 6.2.3 CanonicalizeLanguageTag (locale) 6.2.4 DefaultLocale () 6.3 Currency Codes 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.3.1 Intl.LocaleList, prototype 9.3.1 Intl.LocaleList, prototype 9.3.2 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList, prototype. 9.3.1 Intl.LocaleList, prototype. 9.3.2 Properties of Intl.LocaleList Instances 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of Service Constructor 9.4.1 Internal Properties of Service Constructor 9.4.2 Properties of Intl.LocaleList Instances 9.4.1 Intl. LocaleList Constructor 9.4.2 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties of Intl.LocaleList Instances 9.4.1 Internal Properties of Service Constructors 10.2.2 Abstract Operations 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales) 10.2.5 BestFitMatcher (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)	4.2 API Overview			
6 Identification of Locales, Time Zones, and Currencies. 6.1 Case Sensitivity and Case Mapping	4.3 Implementation Depen	dencies		
6.1 Case Sensitivity and Case Mapping 6.2 Language Tags 6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(Iocale) 6.2.3 CanonicalizeLanguageTag (Iocale) 6.2.4 DefaultLocale () 6.3 Currency Codes 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl Object 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype 9.3 Properties of Intl.LocaleList Instances 9.4.1 length 9.4 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, requestedLocales) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)	5 Notational Conventions.			
6.2.1 Language Tags				
6.2.1 Unicode Locale Extension Sequences 6.2.2 IsStructurallyValidLanguageTag(locale)		•		
6.2.2 IsStructurallyValidLanguageTag(locale)				
6.2.4 DefaultLocale () 6.3 Currency Codes 6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.3.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale). 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales). 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales).				
6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Used in a new Expression 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.1.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, Iocale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
6.3.1 IsWellFormedCurrencyCode (currency) 6.4 Time Zone Names 7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype. 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
7 Requirements for Standard Built-in ECMAScript Objects 8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale). 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData). 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales). 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales).	6.3.1 IsWellFormedCurren	cyCode (currency)		
8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale). 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData). 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
8 The Intl Object 8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale). 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData). 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)	7 Requirements for Standa	ard Built-in ECMAScript Objects		
8.1 Constructor Properties of the Intl Object 9 LocaleList Objects 9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)	-			
9.1 The Intl.LocaleList Constructor 9.1.1 Initializing an Object as a LocaleList 9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
9.1.1 Initializing an Object as a LocaleList	9 LocaleList Objects			
9.1.2 The Intl.LocaleList Constructor Called as a Function 9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
9.1.3 The Intl.LocaleList Constructor Used in a new Expression 9.2 Properties of the Intl.LocaleList Constructor 9.2.1 Intl.LocaleList.prototype 9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
9.2 Properties of the Intl.LocaleList Constructor				
9.3 Properties of the Intl.LocaleList Prototype Object 9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length	9.2 Properties of the Intl.Le	ocaleList Constructor		
9.3.1 Intl.LocaleList.prototype.constructor 9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
9.4 Properties of Intl.LocaleList Instances 9.4.1 length 9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
9.4.2 Properties With Array Index Names 10 Locale and Parameter Negotiation				
10 Locale and Parameter Negotiation 10.1 Internal Properties of Service Constructors 10.2 Abstract Operations 10.2.1 IndexOfMatch (availableLocales, locale) 10.2.2 LookupMatcher (availableLocales, requestedLocales) 10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData) 10.2.5 LookupSupportedLocales (availableLocales, requestedLocales) 10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
10.1 Internal Properties of Service Constructors	•			
10.2 Abstract Operations				
10.2.1 IndexOfMatch (availableLocales, locale)				
10.2.3 BestFitMatcher (availableLocales, requestedLocales) 10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData)	10.2.1 IndexOfMatch (availa	ableLocales, locale)		
10.2.4 ResolveLocale (availableLocales, requestedLocales, options, relevantExtensionKeys, localeData)				
relevantExtensionKeys, localeData)10.2.5 LookupSupportedLocales (availableLocales, requestedLocales)10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)				
10.2.6 BestFitSupportedLocales (availableLocales, requestedLocales)	relevantExtensionKe	ys, localeData)		

10.2.8 10.2.9	GetGetOption (options)	
11 C	ollator Objects	
11.1	The Intl.Collator Constructor	
11.1.1	Initializing an Object as a Collator	
11.1.2	The Intl.Collator Constructor Called as a Function	14
11.1.3	The Intl.Collator Constructor Used in a new Expression	
11.2	Properties of the Intl.Collator Constructor	15
11.2.1	Intl.Collator.prototype	
11.2.2	Intl.Collator.supportedLocalesOf (requestedLocales [, options])	
11.2.3	Internal Properties	15
11.3	Properties of the Intl.Collator Prototype Object	16
11.3.1	Intl.Collator.prototype.constructor	
11.3.2	Compare (collator, x, y)	16
11.3.3	Intl.Collator.prototype.compare	
11.3.4	Intl.Collator.prototype.resolvedOptions	
11.4	Properties of Intl.Collator Instances	17
12 N	umberFormat Objects	18
	The Intl.NumberFormat Constructor	
12.1.1	Initializing an Object as a NumberFormat	
12.1.2	The Intl.NumberFormat Constructor Called as a Function	
12.1.3	The Intl.NumberFormat Constructor Used in a new Expression	
12.2	Properties of the Intl.NumberFormat Constructor	
12.2.1	Intl.NumberFormat.prototype	20
12.2.2	Intl.NumberFormat.supportedLocalesOf (requestedLocales [, options])	20
12.2.3	Internal Properties	20
12.3	Properties of the Intl.NumberFormat Prototype Object	
12.3.1	Intl.NumberFormat.prototype.constructor	
12.3.2	Intl.NumberFormat.prototype.format (value)	
12.3.3	Intl.NumberFormat.prototype.resolvedOptions	
12.4	Properties of Intl.NumberFormat Instances	23
13 Da	ateTimeFormat Objects	24
	The Intl.DateTimeFormat Constructor	
13.1.1	Initializing an Object as a DateTimeFormat	
13.1.2	The Intl.DateTimeFormat Constructor Called as a Function	
13.1.3	The Intl.DateTimeFormat Constructor Used in a new Expression	27
13.2	Properties of the Intl.DateTimeFormat Constructor	27
13.2.1	Intl.DateTimeFormat.prototype	27
13.2.2	Intl.DateTimeFormat.supportedLocalesOf (requestedLocales [, options])	
13.2.3	Internal Properties	
	Properties of the Intl.DateTimeFormat Prototype Object	
13.3.1	Intl.DateTimeFormat.prototype.constructor	
13.3.2	Intl.DateTimeFormat.prototype.format ([date])	
13.3.3	Intl.DateTimeFormat.prototype.resolvedOptions	
13.4	Properties of Intl.DateTimeFormat Instances	30
14 Lo	ocale Sensitive Functions of the ECMAScript Language Specification	30
	Properties of the String Prototype Object	
14.1.1	String.prototype.localeCompare (that [, localeList [, options]])	
	Properties of the Number Prototype Object	
14.2.1	Number.prototype.toLocaleString ([localeList [, options]])	
	Properties of the Date Prototype Object	
14.3.1	Date.prototype.toLocaleString ([localeList [, options]])	
14.3.2	Date.prototype.toLocaleDateString ([localeList [, options]])	31
14.3.3	Date.prototype.toLocaleTimeString ([localeList [, options]])	

	Page Break———	
"DISCLAIMER		
Page iv: [3] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [4] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [4] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)	•	1, 2,
Page iv: [5] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)	No. bert Ellineering	0/10/11 10:10
Page iv: [5] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		0/10/11 10:10
Page iv: [6] Deleted	Nouhout Lindouhous	6/29/12 12:20
	Norbert Lindenberg se explain it or assist in its implementa	6/28/12 13:20
mat comment on or otherwi	se explain it of assist in its implementa	tion
Page iv: [7] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [8] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [9] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [10] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [11] Formatted	Norbert Lindenberg	6/28/12 13:20

Page iv: [12] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [13] Deleted	Norbert Lindenberg	6/28/12 13:20
needed for the purpose of c	leveloping any document or delive	rable produced by Ecma
International		
Page iv: [14] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)	itorbeit Einaenbeig	0/10/11 10:10
3 - (- /		
Dago ive [15] Deleted	Nouhout Lindonhous	6/20/12 12:20
Page iv: [15] Deleted	Norbert Lindenberg	6/28/12 13:20
on the standard are reserved	rior to final version of this documen by Ecma International	t. After approval all rights
Page iv: [16] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [17] Formatted	Norbert Lindenberg	6/28/12 13:20
M4, No widow/orphan control		
Page iv: [18] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		0/ 10/ 11 13/10
3 ()		
Page in [10] Formatted	Nouhout Lindonhous	6/29/12 12:20
Page iv: [19] Formatted English (UK)	Norbert Lindenberg	6/28/12 13:20
English (OK)		
Page iv: [20] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page iv: [21] Formatted	Norbert Lindenberg	6/28/12 13:20
Font:Not Italic, English (UK)		
Page iv: [22] Deleted	Norbert Lindenberg	6/28/12 13:20
	Page Break———	
Page 1: [23] Deleted	Norbert Lindenberg	6/28/12 13:20
E 1 adition or augocoper		

ISO/IEC 10646:2003: Information Technology – Universal Multiple-Octet Coded Character Set (UCS) plus Amendment 1:2005 and Amendment 2:2006, plus additional amendments and corrigenda, or successor

ISO 4217:2008, Codes for the representation of currencies and funds, or successor

The Unicode Standard, Version 5.0, or successor

<u>Unicode Technical Standard 35, Unicode Locale Data Markup Language</u>, version 2.0.1 or successor

Page 1: [24] Deleted Norbert Lindenberg 6/28/12 13:20

RFC 5646, Tags for Identifying Languages, or successor RFC 4647, Matching of Language Tags, or successor

IETF RFC 6067, BCP 47 Extension U, or successor

Page 6: [25] Deleted Norbert Lindenberg 6/28/12 13:20

LocaleList Objects

LocaleList objects represent lists of language tags identifying locales. They can be used in two ways:

To represent a language priority list, as described in RFC 4647, section 2.3, or successor. Algorithms interpreting a LocaleList object in this sense treat the list as ordered in descending order of priority.

To represent a set of locales, such as those supported by an application or by the implementation of an object described in this specification. Algorithms interpreting a LocaleList object in this sense treat the list as unordered.

LocaleList objects have the properties of a generic array-like object: A length property and other properties whose names are array indices, as defined in ES5, 15.4. The value of the length property is numerically greater than the name of every property inserted during construction whose name is an array index. As LocaleList objects are extensible, this invariant is not guaranteed to be maintained after construction.

The LocaleList constructor is a property of the Intl object.

The Intl.LocaleList Constructor

Initializing an Object as a LocaleList

The abstract operation InitializeLocaleList accepts the argument *localeList*, which must be an object, and the optional argument *locales*. It initializes *localeList* as a LocaleList object by taking the following steps:

If *locales* is not provided or is **undefined**, then

Let *seen* be a new List containing the String returned by the DefaultLocale abstract operation. Else

Page 6: [26] Deleted Norbert Lindenberg 6/28/12 13:20

If the result of calling the abstract operation IsStructurallyValidLanguageTag, passing *tag* as the argument, is **false**, then throw a **RangeError** exception. Let *tag* be the result of calling the abstract operation CanonicalizeLanguageTag, passing *tag* as the argument.

Page 6: [27] Deleted

Norbert Lindenberg

6/28/12 13:20

Let *index* be 0.

Repeat for each element tag of seen, in list order

Call the [[DefineOwnProperty]] internal method of *localeList* with arguments ToString(*index*), Property Descriptor {[[Value]]: tag, [[Writable]]: false, [[Enumerable]]: true, [[Configurable]]: false}, and true.

Increase *index* by 1.

Call the [[DefineOwnProperty]] internal method of *localeList* with arguments "length", Property Descriptor {[[Value]]: *index*, [[Writable]]: **false**, [[Enumerable]]: **false**, [[Configurable]]: **false**}, and **true**.

Set the [[initializedLocaleList]] internal property of *localeList* to **true**.

NOTE Non-normative summary: The function interprets the *locales* argument as an array and copies its elements into *localeList*, validating the elements as structurally valid language tags and canonicalizing them, and omitting duplicates. It tags this object as an initialized locale list.

NOTE If an object has been previously initialized using the Intl.LocaleList constructor, and the canonicalization of the previously used *locales* differs from the canonicalization of the *locales* used now, then either step 4.a or step 5 will fail.

The Intl.LocaleList Constructor Called as a Function

When Intl.LocaleList is called as a function rather than as a constructor, it accepts the optional argument *locales* and takes the following steps:

If **this** is the Intl object or **undefined**, then

Return the result of creating a new object as if by the expression new Intl.LocaleList(locales), where Intl.LocaleList is the standard built-in constructor defined in 9.1.3.

Page 6: [28] Deleted

Norbert Lindenberg

6/28/12 13:20

Call the InitializeLocaleList abstract operation with arguments *obj* and *locales*. Return *obj*.

The Intl.LocaleList Constructor Used in a new Expression

When Intl.LocaleList is called as part of a new expression, it is a constructor. It accepts the optional argument *locales*, and initializes the properties of the newly constructed object by calling the InitializeLocaleList abstract operation (9.1.1), passing the newly constructed object and *locales* as arguments.

The [[Prototype]] internal property of the newly constructed object is set to the original Intl.LocaleList prototype object, the one that is the value of Intl.LocaleList.prototype (9.2.1).

Page 6: [29] Deleted

Norbert Lindenberg

6/28/12 13:20

Properties of the Intl.LocaleList Constructor

Besides the internal properties and the length property (whose value is 1), the Intl.LocaleList constructor has the following properties:

Intl.LocaleList.prototype

The value of Intl.LocaleList.prototype is the built-in Intl.LocaleList prototype object (9.3).

Page 6: [30] Deleted Norbert Lindenberg 6/28/12 13:20

Properties of the Intl.LocaleList Prototype Object

The Intl.LocaleList prototype object is itself an Intl.LocaleList instance as specified in 9.4, whose properties are set as if it had been constructed by the expression new Intl.LocaleList([]).

Intl.LocaleList.prototype.constructor

The initial value of Intl.LocaleList.prototype.constructor is the built-in Intl.LocaleList constructor.

Properties of Intl.LocaleList Instances

Intl.LocaleList instances inherit properties from the Intl.LocaleList prototype object.

Intl.LocaleList instances and other objects that have been successfully initialized as a LocaleList have an [[initializedLocaleList]] internal property whose value is **true**.

Intl.LocaleList instances and other objects that have been successfully initialized as a LocaleList also have the following properties.

length

The length property of an Intl.LocaleList object is a data property whose value is the number of array index properties added to the object by the InitializeLocaleList abstract operation.

The length property has the attributes { [[Writable]]: false, [[Enumerable]]: false, [[Configurable]]: false }.

Properties With Array Index Names

A LocaleList object has properties whose names are array indices from 0 to (length - 1). The value of each of these properties is a String value representing a structurally valid language tag. The values are unique within a LocaleList object.

These properties initially have the attributes { [[Writable]]: false, [[Enumerable]]: true, [[Configurable]]: false }.

Page 7: [31] Deleted Norbert Lindenberg 6/28/12 13:20

If $index \neq -1$

Page 8: [32] Deleted Norbert Lindenberg 6/28/12 13:20

availableIndex \neq -1, then

Let

Page 9: [33] Deleted Norbert Lindenberg 6/28/12 13:20

If requestedLocales is undefined, then

Let requestedLocales be the result of creating a new LocaleList object as if by the expression new Intl.LocaleList() where Intl.LocaleList is the standard built-in constructor defined in 9.1.3.

Else

Let requestedLocales be ToObject(requestedLocales).

If *requestedLocales* does not have an [[initializedLocaleList]] internal property with value **true**, then

Let requestedLocales be the result of creating a new LocaleList object as if by the expression new Intl.LocaleList(requestedLocales), where Intl.LocaleList is the standard built-in constructor defined in 9.1.3.

Page 10: [34] Deleted

Norbert Lindenberg

6/28/12 13:20

Let requestedLocales be ToObject(requestedLocales).

If requestedLocales does not have an [[initializedLocaleList]] internal property with value true, then Let requestedLocales be a new LocaleList object created as if by the expression new Intl.LocaleList (requestedLocales), where Intl.LocaleList is the standard built-in constructor defined in 9.1.3.

Page 11: [35] Formatted	Norbert Lindenberg	6/28/12 13:20
Alexander and alexander Francisco (LDZ)		

Algorithm variable, English (UK)

Page 11: [36] Deleted	Norbert Lindenberg	6/28/12 13:20
-----------------------	--------------------	---------------

GetGetOption

Page 11: [36] Deleted Norbert Lindenberg 6/28/12 13:20

GetGetOption

Page 11: [36] Deleted	Norbert Lindenberg	6/28/12 13:20
-----------------------	--------------------	---------------

GetGetOption

Algorithm variable

Page 11: [37] Formatted	Norbert Lindenberg	6/28/12 13:20
-------------------------	--------------------	---------------

Algorithm variable

raye 11. [30] Fullilatieu Rui Dei t Elliueli Dei q U/20/12 13.2	Page 11: [38] Formatted	Norbert Lindenberg	6/28/12 13:20
-----------------------------------------------------------------	-------------------------	--------------------	---------------

Algorithm variable

raye 11. [30] Fullilatieu Rui Dei t Elliueli Dei q U/20/12 13.2	Page 11: [38] Formatted	Norbert Lindenberg	6/28/12 13:20
-----------------------------------------------------------------	-------------------------	--------------------	---------------

Algorithm variable

Page 11: [39] Deleted Norbert Lindenberg 6/28/12 13:20

When the GetGetOption abstract operation is called with argument *options*, the following steps are taken.

Let *getOption* be a function which, when called with arguments *property*, *type*, *values*, and *fallback*, takes the following steps:

Page 11: [40] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		
Page 11: [41] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted	•	., .,
Page 11: [42] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		0,10,11 13.13
Page 11: [42] Deleted	Norhout Lindonhous	6/20/12 12:20
Page 11: [43] Deleted the result of calling the indexOf method of	Norbert Lindenberg	6/28/12 13:20
i.		
Page 11: [43] Deleted	Norbert Lindenberg	6/28/12 13:20
the result of calling the indexOf method of		
Page 11: [43] Deleted	Norbert Lindenberg	6/28/12 13:20
the result of calling the indexOf method of		
Page 11: [44] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		
Page 11: [45] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		
Page 11: [46] Deleted	Norbert Lindenberg	6/28/12 13:20
Return getOption.	Norbert Emdemberg	0, 20, 12 13:20
rectain geropiion.		
GetGetNumberOption		
1.1.1		
Page 11: [47] Deleted	Norbert Lindenberg	6/28/12 13:20
GetGetNumberOption		
-		
Page 11: [47] Deleted	Norbert Lindenberg	6/28/12 13:20
GetGetNumberOption		
Page 11: [48] Deleted	Norbert Lindenberg	6/28/12 13:20
When the GetGetNumberOption abstra	act operation is called with a	gument options, the following

When the GetGetNumberOption abstract operation is called with argument *options*, the following steps are taken.

Let *getNumberOption* be a function which, when called with arguments *property*, *minimum*, *maximum*, and *fallback*, takes the following steps:

Page 11: [49] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		
Page 11: [50] Formatted	Norbert Lindenberg	6/28/12 13:20
Formatted		0/10/11 10:10
Dave 11, [F1] Fermatted	Naubauk Lindaubaun	6/20/12 12:20
Page 11: [51] Formatted Formatted	Norbert Lindenberg	6/28/12 13:20
Tomatted		
Page 11: [52] Moved from page 1:	1 (Move #9)Norbert Lindenberg	6/28/12 13:20
1.2 The Intl.Collator Cons	structor	
Page 11: [53] Moved to page 11 (Move #9) Norbert Lindenberg	6/28/12 13:20
1.1 The Intl.Collator Cons		· · · · ·
Page 11: [54] Deleted	Norbert Lindenberg	6/28/12 13:20
argument		
_		
Page 11: [54] Deleted	Norbert Lindenberg	6/28/12 13:20
argument	Norbeit Eindenbeig	0/20/12 13:20
a.gae		
		4/20/42 42 22
Page 11: [54] Deleted	Norbert Lindenberg	6/28/12 13:20
argument		
Page 11: [54] Deleted	Norbert Lindenberg	6/28/12 13:20
argument		
Page 12: [55] Deleted	Norbert Lindenberg	6/28/12 13:20
Let getOption be the result of	calling the GetGetOption abstract operation	with argument options.
Page 12: [55] Deleted	Norbert Lindenberg	6/28/12 13:20
	calling the GetGetOption abstract operation	
	-	-
Page 12: [55] Deleted	Norbert Lindenberg	6/28/12 13:20
Let <i>getOption</i> be the result of	calling the GetGetOption abstract operation	with argument options.
Page 12: [55] Deleted	Norbert Lindenberg	6/28/12 13:20
Let getOption be the result of	calling the GetGetOption abstract operation	with argument options.
Page 12: [56] Deleted	Norbert Lindenberg	6/28/12 13:20
30 [00] Deleted	1101 DOTE BINGOIDOLY	0/ 20/ 12 13:20

Page 12: [56] Deleted	Norbert Lindenberg	6/28/12 13:20
If u		
Page 12: [57] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [57] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [57] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [57] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [58] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [58] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 12: [58] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
	Norbert Lindenberg	6/28/12 13:20
Page 12: [58] Deleted	Norbert Lindeliberg	0/20/12 15:20
getOption	Norbert Eindenberg	0/20/12 13.20
getOption Page 13: [59] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption Page 13: [59] Deleted properties of the newly cons		6/28/12 13:20 Collator abstract operation
getOption Page 13: [59] Deleted properties of the newly cons	Norbert Lindenberg tructed object by calling the InitializeOnstructed object, localeList, and options as	6/28/12 13:20 Collator abstract operation
page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor	Norbert Lindenberg tructed object by calling the InitializeOnstructed object, localeList, and options as	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20
page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor	Norbert Lindenberg tructed object by calling the Initialize of the initialize of the structed object, localeList, and options as the structed object.	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly con Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop	Norbert Lindenberg tructed object by calling the Initialize of th	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true.
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop	Norbert Lindenberg tructed object by calling the Initialize of th	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true.
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop Page 16: [61] Deleted is	Norbert Lindenberg tructed object by calling the Initialize on tructed object, localeList, and options as Move #10)Norbert Lindenberg erty of the newly constructed object is se	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true. 6/28/12 13:20
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop Page 16: [61] Deleted is	Norbert Lindenberg tructed object by calling the Initialize on tructed object, localeList, and options as Move #10)Norbert Lindenberg erty of the newly constructed object is se	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true. 6/28/12 13:20
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop Page 16: [61] Deleted is Page 16: [61] Deleted is	Norbert Lindenberg tructed object by calling the Initialize of th	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true. 6/28/12 13:20
Page 13: [59] Deleted properties of the newly cons (11.1.1), passing the newly cor Page 13: [60] Moved to page 19 (The [[Extensible]] internal prop Page 16: [61] Deleted is Page 16: [61] Deleted is	Norbert Lindenberg tructed object by calling the Initialize of th	6/28/12 13:20 Collator abstract operation s arguments 6/28/12 13:20 et to true. 6/28/12 13:20

Page 16: [63] Deleted Norbert Lindenberg 6/28/12 13:20

If this Collator does not have an [[initializedCollator]] internal property with value **true**, then throw a **TypeError** exception.

If the [[boundCompare]] internal property of this Collator is **undefined**, then: Let *that* be **this**. Let bc be a function that takes the arguments x and y and performs the following steps: Return the result of calling the Compare abstract operation with arguments that, x, and y.

Set the [[boundCompare]] internal property of this Collator to bc. Return the value of the [[boundCompare]] internal property of this Collator.

The [[Set]] attribute is undefined.

Page 16: [64] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 16: [64] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a	Horbert Emachberg	0/20/12 15:20
Page 16: [64] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 16: [64] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		01-201-201-20
to the first of the second		
Page 16: [65] Deleted	Norbert Lindenberg	6/28/12 13:20
an		
Page 16: [65] Deleted	Norbert Lindenberg	6/28/12 13:20
an		
Page 16: [65] Poloted	Naukaut Lindaukaun	C/20/12 12:20
Page 16: [65] Deleted	Norbert Lindenberg	6/28/12 13:20
an		
Page 16: [66] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [66] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)	Norbert Emdenberg	0/20/12 13:20
rugenum code, English (ent)		
Page 16: [67] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [67] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		, .,
, J = (= /		

		4/20/42 42 22
Page 16: [67] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [67] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [68] Deleted	Norbert Lindenberg	6/28/12 13:20
"		
Page 16: [68] Deleted	Norbert Lindenberg	6/20/12 12:20
"	Norbert Lindenberg	6/28/12 13:20
Page 16: [69] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [69] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)		
Page 16: [69] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm code, English (UK)	norbeit Emacinbeig	0/20/12 15:20
,ge code,g (e)		
		6/20/42 42 22
Page 17: [70] Deleted	Norbert Lindenberg	6/28/12 13:20
The Intl.NumberFormat Cor	istructor	
Dago 17: [71] Dolotod	Noubout Lindonbove	6/20/12 12:20
Page 17: [71] Deleted argument	Norbert Lindenberg	6/28/12 13:20
argument		
Page 17: [71] Deleted	Norbert Lindenberg	6/28/12 13:20
argument		
Page 17: [72] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page 17: [73] Deleted	Norbert Lindenberg	6/28/12 13:20
the optional arguments localeList and		
Page 17: [73] Deleted	Norbert Lindenberg	6/28/12 13:20
the optional arguments <i>localeList</i> and		.,,
Dago 17, [74] Doloke J	Novbout Lindonbo	6/20/42 42:20
Page 17: [74] Deleted	Norbert Lindenberg	6/28/12 13:20

Let *getOption* be the result of calling the GetGetOption abstract operation with argument *options*.

Let getNumberOption be the result of calling the GetGetNumberOption abstract operation with argument options.

Page 17: [75] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 17: [75] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 17: [75] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 17: [76] Formatted	Norbert Lindenberg	6/28/12 13:20
Default Paragraph Font		
Page 17: [77] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)		
Page 17: [78] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)		0/20/12 10:20
3		
Page 17: [79] Deleted	Norbert Lindenberg	6/28/12 13:20
the <i>localeList</i> argument	Norbert Lindenberg	0/20/12 13.20
Page 17: [79] Deleted	Norbert Lindenberg	6/28/12 13:20
the localeList argument	Norbert Lindenberg	0/20/12 13.20
Page 17: [80] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)	Norbert Endemberg	0/20/12 13:20
rugenum vanasie, Englien (ens)		
D 47- [04] D-1-4-4	North and Unidentifying	C /20 /42 42-20
Page 17: [81] Deleted getOption	Norbert Lindenberg	6/28/12 13:20
6.	North and Day down and	C /20 /42 42-20
Page 17: [81] Deleted getOption	Norbert Lindenberg	6/28/12 13:20
7.	North and Day down	C/20/42 42-20
Page 17: [82] Formatted Default Paragraph Font	Norbert Lindenberg	6/28/12 13:20
Deladit Faragraph Font		
Page 17: [82] Formatted	Norbert Lindenberg	6/28/12 13:20
Default Paragraph Font		
Page 17: [83] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 17: [83] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
		6/20/12 12:20
Page 17: [83] Deleted getOption	Norbert Lindenberg	6/28/12 13:20

Page 17: [84] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm keyword, English (UK)		
Page 17: [85] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
n. Page 17: [85] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 17: [86] Formatted	Norbert Lindenberg	6/28/12 13:20
Default Paragraph Font		
Page 17: [86] Formatted	Norbert Lindenberg	6/28/12 13:20
Default Paragraph Font		
Page 17: [87] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page 17: [88] Deleted	Norbert Lindenberg	6/28/12 13:20
equal to		
Page 17: [88] Deleted	Norbert Lindenberg	6/28/12 13:20
equal to		
Page 18: [89] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [89] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [90] Deleted	Norbert Lindenberg	6/28/12 13:20
equal to		
Page 18: [90] Deleted	Norbert Lindenberg	6/28/12 13:20
equal to		
Page 18: [90] Deleted	Norbert Lindenberg	6/28/12 13:20
equal to		
Page 18: [91] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [91] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [92] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [92] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 18: [93] Deleted	Norbert Lindenberg	6/28/12 13:20

Page 18: [93] Deleted	Norbert Lindenberg	6/28/12 13:20
getNumberOption		
Page 21: [94] Deleted	Norbert Lindenberg	6/28/12 13:20

If this NumberFormat object does not have an [[initializedNumberFormat]] internal property with value **true**, then throw a **TypeError** exception.

Let *x* be ToNumber(*value*).

Page 22: [95] Deleted	Norbert Lindenberg	6/28/12 13:20
	nethod of m with argument ".".	6/28/12 13:20
If $period > 0$	nethod of m with argument ".".	
1.		
Page 25: [96] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [96] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [96] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [96] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [97] Deleted	Norbert Lindenberg	6/28/12 13:20
the <i>localeList</i> argument		
Page 25: [97] Deleted	Norbert Lindenberg	6/28/12 13:20
the localeList argument		
Page 25: [98] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [98] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [98] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [99] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [99] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [100] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [100] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 25: [100] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption 12.		
Page 25: [100] Deleted	Norbert Lindenberg	6/28/12 13:20
getOption		
Page 29: [101] Deleted	Norbert Lindenberg	6/28/12 13:20

Page 29: [101] Deleted	Norbert Lindenberg	6/28/12 13:20
new		., .,
Page 29: [102] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)	Norbert Emdenberg	0/20/11 13:10
3 - (- /		
Dago 20: [102] Formattod	Mouhout I indonhous	6/20/12 12:20
Page 29: [102] Formatted English (UK)	Norbert Lindenberg	6/28/12 13:20
English (Sit)		
D 20- [402] D-l-tJ	Nach and Cardan barre	C/20/12 12:20
Page 29: [103] Deleted	Norbert Lindenberg	6/28/12 13:20
1)		
Returns a String value represen	ting the result of calling ToNumber(date)	
Page 29: [104] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)		
Page 29: [104] Formatted	Norbert Lindenberg	6/28/12 13:20
English (UK)	Norbert Emdenberg	0/20/12 15:20
3 ()		
Page 29: [105] Formatted	Norbert Lindenberg	6/28/12 13:20
English (US)	Norbert Emdemberg	0/20/12 15:20
J = (= =)		
Page 29: [105] Formatted	Norbert Lindenberg	6/28/12 13:20
English (US)	Norbert Emidemberg	0/20/12 13.20
English (55)		
Page 29: [105] Formatted English (US)	Norbert Lindenberg	6/28/12 13:20
English (03)		
Page 29: [106] Deleted	Norbert Lindenberg	6/28/12 13:20
1.	imeFormat]] internal property with value true	
Page 29: [106] Deleted	Norbert Lindenberg	6/28/12 13:20
2.	imeFormat]] internal property with value true	
Page 29: [107] Deleted	Norbert Lindenberg	6/28/12 13:20
omitted		
Page 29: [107] Deleted	Norbert Lindenberg	6/28/12 13:20
omitted		
Page 29: [107] Deleted	Norbert Lindenberg	6/28/12 13:20
omitted		

Norbert Lindenberg	6/28/12 13:20
Norbert Lindenberg	6/28/12 13:20
Norbert Lindenberg	6/28/12 13:20
Norbert Lindenberg	6/28/12 13:20
Norbert Lindenberg	6/28/12 13:20
Norbert Lindenberg	6/28/12 13:20
	Norbert Lindenberg Norbert Lindenberg Norbert Lindenberg Norbert Lindenberg

Let *format* be the standard built-in function object defined in 12.3.2.

Page 30: [110] Deleted	Norbert Lindenberg	6/28/12 13:20

Let *format* be the standard built-in function object defined in 12.3.2.

Page 30: [111] Deleted Norbert Lindenberg 6/28/12 [[Call]] internal method of format Page 30: [111] Deleted Norbert Lindenberg 6/28/12 [[Call]] internal method of format Page 30: [111] Deleted Norbert Lindenberg 6/28/12 [[Call]] internal method of format	
Page 30: [111] Deleted Norbert Lindenberg 6/28/12 [[Call]] internal method of format Page 30: [111] Deleted Norbert Lindenberg 6/28/12	13:20
[[Call]] internal method of <i>format</i> Page 30: [111] Deleted Norbert Lindenberg 6/28/12	13:20
Page 30: [111] Deleted Norbert Lindenberg 6/28/12	
[[Call]] internal method of <i>format</i>	13:20
3	
Page 30: [112] Deleted Norbert Lindenberg 6/28/12	13:20
[[Call]] internal method of format	
Page 30: [112] Deleted Norbert Lindenberg 6/28/12	13:20
[[Call]] internal method of format	
Page 30: [112] Deleted Norbert Lindenberg 6/28/12	13:20
[[Call]] internal method of format	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20
oguels	
equals	
Page 30: [113] Deleted Norbert Lindenberg 6/28/12	13:20

viii.		
Page 30: [114] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)		
Page 30: [114] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)		
Page 30: [114] Formatted	Norbert Lindenberg	6/28/12 13:20
Algorithm variable, English (UK)		
Page 30: [115] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 30: [115] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 30: [115] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 30: [115] Deleted	Norbert Lindenberg	6/28/12 13:20
value of the [[Get]] attribute is a		
Page 33: [116] Deleted	Norbert Lindenberg	6/28/12 13:20

an argument list containing the single item this.

References

ECMA-262, ECMAScript Language Specification

http://www.ecma-international.org/publications/standards/Ecma-262.htm

ISO 4217:2008, Codes for the representation of currencies and funds

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=46121

ISO/IEC 10646:2003: Information Technology - Universal Multiple-Octet Coded Character Set

(UCS) plus Amendment 1:2005 and Amendment 2:2006 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=39921

http://www.iso.org/iso/iso catalogue/catalogue tc/catalogue detail.htm?csnumber=40755

http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41419

The Unicode Standard

http://www.unicode.org/versions/latest

Unicode Technical Standard 10, Unicode Collation Algorithm

http://unicode.org/reports/tr10/

<u>Unicode Technical Standard 35, Unicode Locale Data Markup Language</u> http://unicode.org/reports/tr35/

<u>Unicode Common Locale Data Repository</u> http://cldr.unicode.org/

IETF BCP 47:

RFC 5646, Tags for Identifying Languages http://tools.ietf.org/html/rfc5646 RFC 4647, Matching of Language Tags http://tools.ietf.org/html/rfc4647

<u>IETF RFC 6067, BCP 47 Extension U</u> http://tools.ietf.org/html/rfc6067

IANA Time Zone Database http://www.iana.org/time-zones