Microsoft

2018年12月6日（木）
文字情報技術促進協議会 セミナー
『新元号対応の最新動向』

新元号対応について

日本マイクロソフト株式会社
業務執行役員 NTO

田丸 健三郎

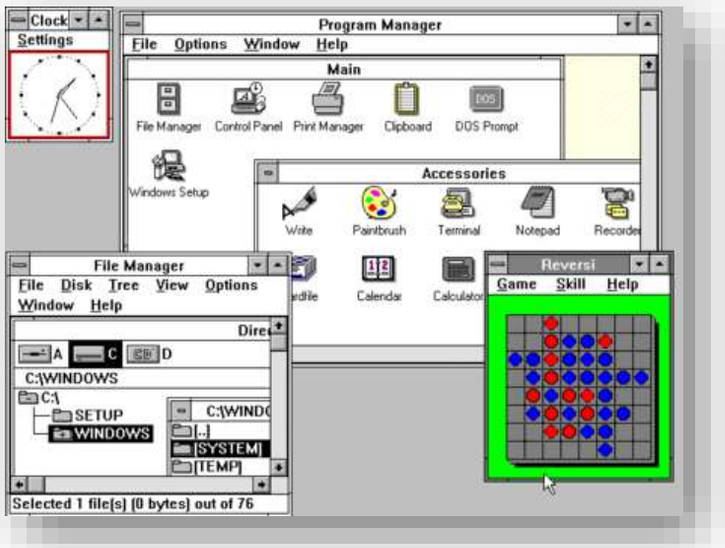
アジェンダ

- 改元とデータ交換
- 改元と文字と標準
- マイクロソフトにおける主な対応
- .NET Framework
- 更新対象製品
- 『新元号検証ラボ』について

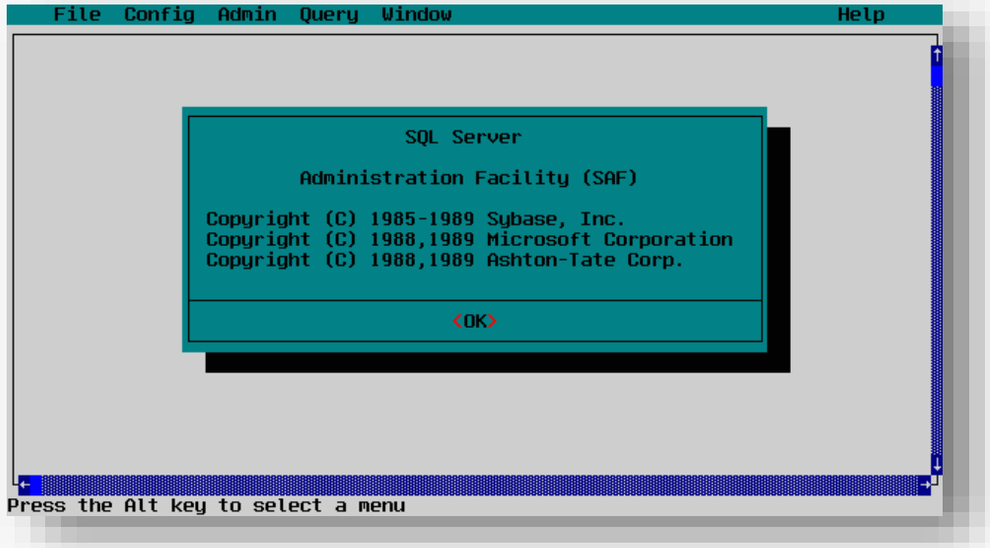
最後に元号が変わったのは約30年前

日本語版のマイクロソフト製品はほとんど存在していなかった

Intel486 DX-33



Windows 3.0



SQL Server 1.0

主な日程

2018年 秋	10月12日	第１回天皇陛下の御退位及び皇太子殿下の御即位に伴う式典委員会を開催		
2019年	2月24日	天皇陛下ご在位三十年記念式典		
	4月30日	退位の礼	**退位礼正殿の儀**	即位を広く明らかにし、陛下が退位前に国民の代表に会われる
	5月1日	即位の礼	**剣璽等継承の儀**	即位に伴い皇室に由緒ある品を引き継がれる
	10月22日		**即位後朝見の儀**	即位後初めて国民の代表に会われる
			即位礼正殿の儀	即位を公に宣言し、国内外の代表がことほぐ
	10月22日以降		**祝賀御礼の儀**	即位を広く国民に披露し、祝福を受けられる
	10月23日		**饗宴の儀**	即位を披露され、祝福を受けられる
	11月14～15日	首相夫妻主催晩さん会		新天皇が国家安寧や五穀豊穣を祈られる
2020年	4月19日	**立皇嗣宣明の儀**		秋篠宮さまが皇嗣となられたことを広く国民に明らかにする

主な作業

2019年	2月24日	天皇陛下ご在位三十年記念式典	
	4月30日	退位の礼	**退位礼正殿の儀**
	5月1日	即位の礼	**剣璽等継承の儀**
	10月22日		**即位後朝見の儀**
			即位礼正殿の儀
	10月22日以降		**祝賀御礼の儀**
	10月23日		**饗宴の儀**
	11月14～15日	首相夫妻主催晩さん会	

オペレーティングシステム、
アプリケーションの更新作業

改元とデータ交換

日付の有効性確認

- .NET Frameworkにおける振る舞い

DateTime.Parse(“昭和７０年１月１日”)

System.FormatException

現在のよくある実装

無効な日付でエラー（エクセプション）が発生することを利用し、入力された日付を確認。（ユーザーが入力した日付の有効性確認）

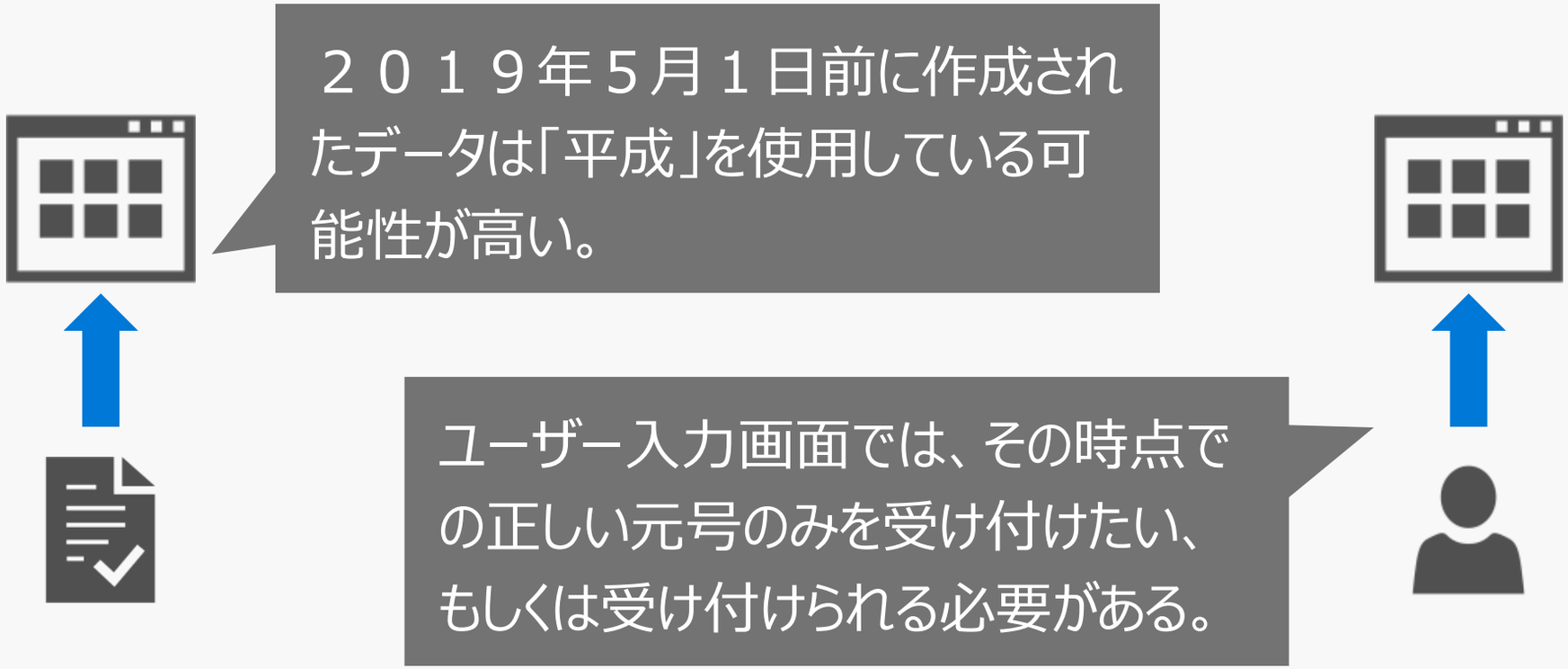
昭和と平成の違い

- 「昭和70年」を西暦「1995年」と変換するニーズは存在しない。
- 異なる「平成31年」の扱い

	ユーザーインターフェース（入力確認）		データ（ファイル読込/通信）	
	昭和70年	平成31年	昭和70年	平成31年
2018年11月28日時点	エラー処理	2019年に変換	エラー処理	2019年に変換
新元号発表後	エラー処理	エラー処理?	エラー処理	2019年に変換?

実装変更が必要になる場合

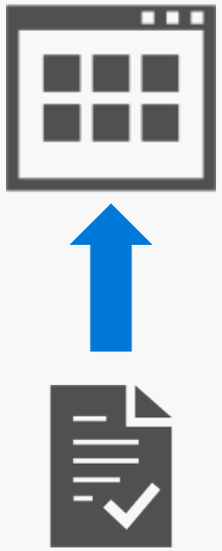
- 「平成３１年１０月１日」



実装変更が必要になる場合

- 「平成３１年１０月１日」時点…

２０１９年５月１日前に作成されたデータは
「平成」を使用している可能性が高い。



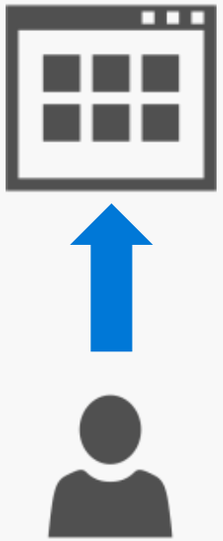
更新の有無に関わらず扱える事が望まれる。しかし、
システムAPIは、必ずどちらかのパターンでエラーとなる。

■ 環境によりエラーとなるパターンが異なる

■「平成３１年１０月１日」

または

■「〇〇元年年１０月１日」

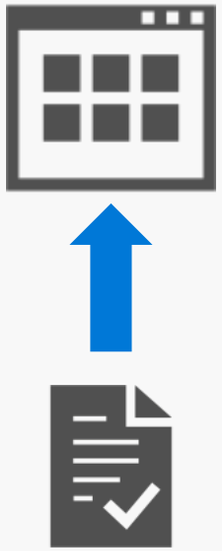


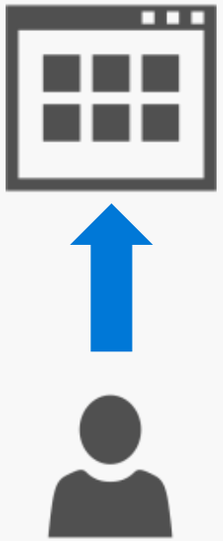
実装変更が必要になる場合

- 「平成３１年１０月１日」時点…

ユーザー入力画面では、その時点での正しい元号のみを受け付けたい、もしくは受け付けられる必要がある。

更新されているシステムは「平成」がエラーに。更新されていないシステムでは、新元号がエラーに。





CSV、JSON、XML

- 拡張子から見えないデータ構造
  - 異なるベンダーソリューション間のデータ交換には、可読性の高いデータフォーマットが用いられている場合が多い。
  - 多くのシステムがデータ交換、保存など様々な用途にCSV、JSON、XMLを使用している。
- データ交換の原則による柔軟な読み込み


```
  <?xml version="1.0" encoding="ISO-8859-1"?>
- <sst uniqueCount="1" count="1"
  xmlns="http://schemas.openxmlformats.org/spreadsheetml/2006/main">
    - <si>
        <t>平成31年5月1日</t>
        <rPh eb="2" sb="0">
            <t>ヘイセイ</t>
        </rPh>
      - <rPh eb="5" sb="4">
            <t>ネン</t>
        </rPh>
      - <rPh eb="7" sb="6">
            <t>ガツ</t>
        </rPh>
      - <rPh eb="9" sb="8">
            <t>ニチ</t>
        </rPh>
        <phoneticPr fontId="1"/>
      </si>
  </sst>
```

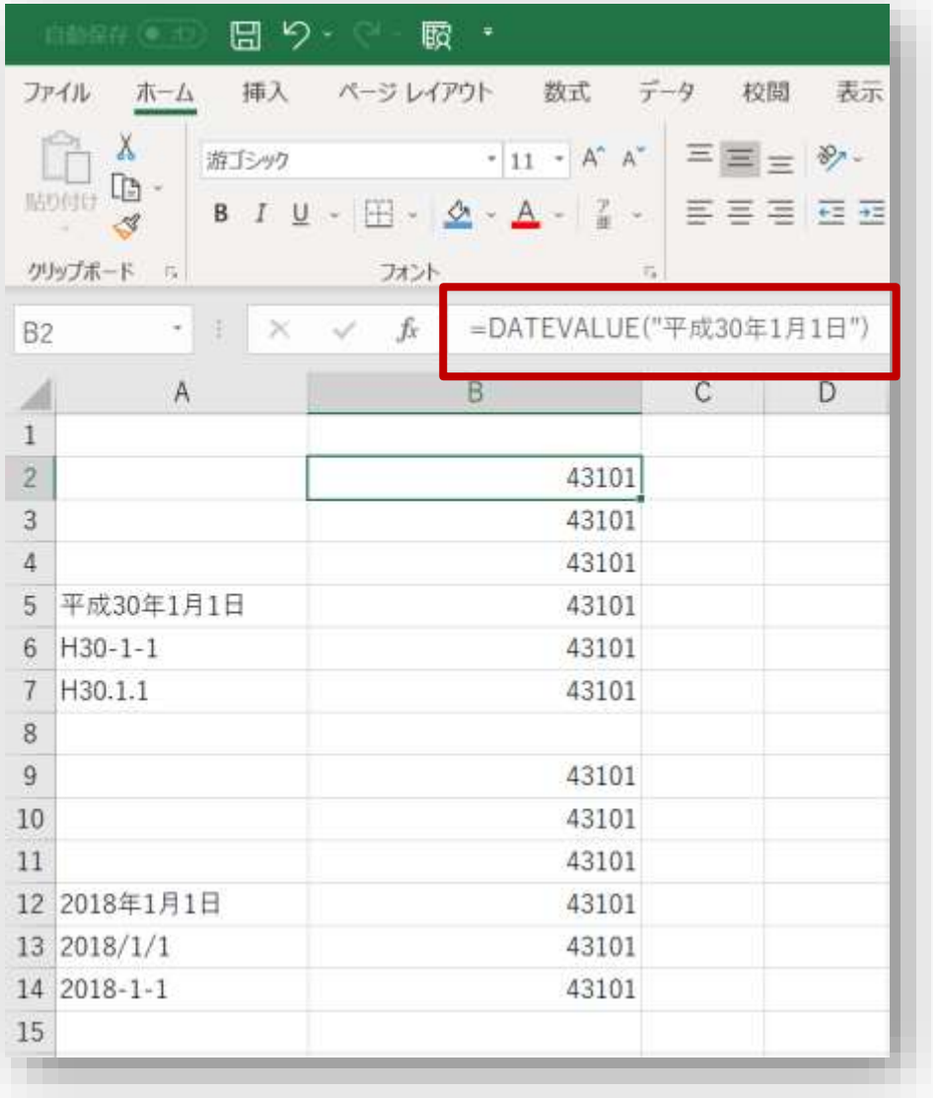
データ交換における原則

送信は**厳格**に。受信は**寛容**に。

データにバインドされたスクリプト

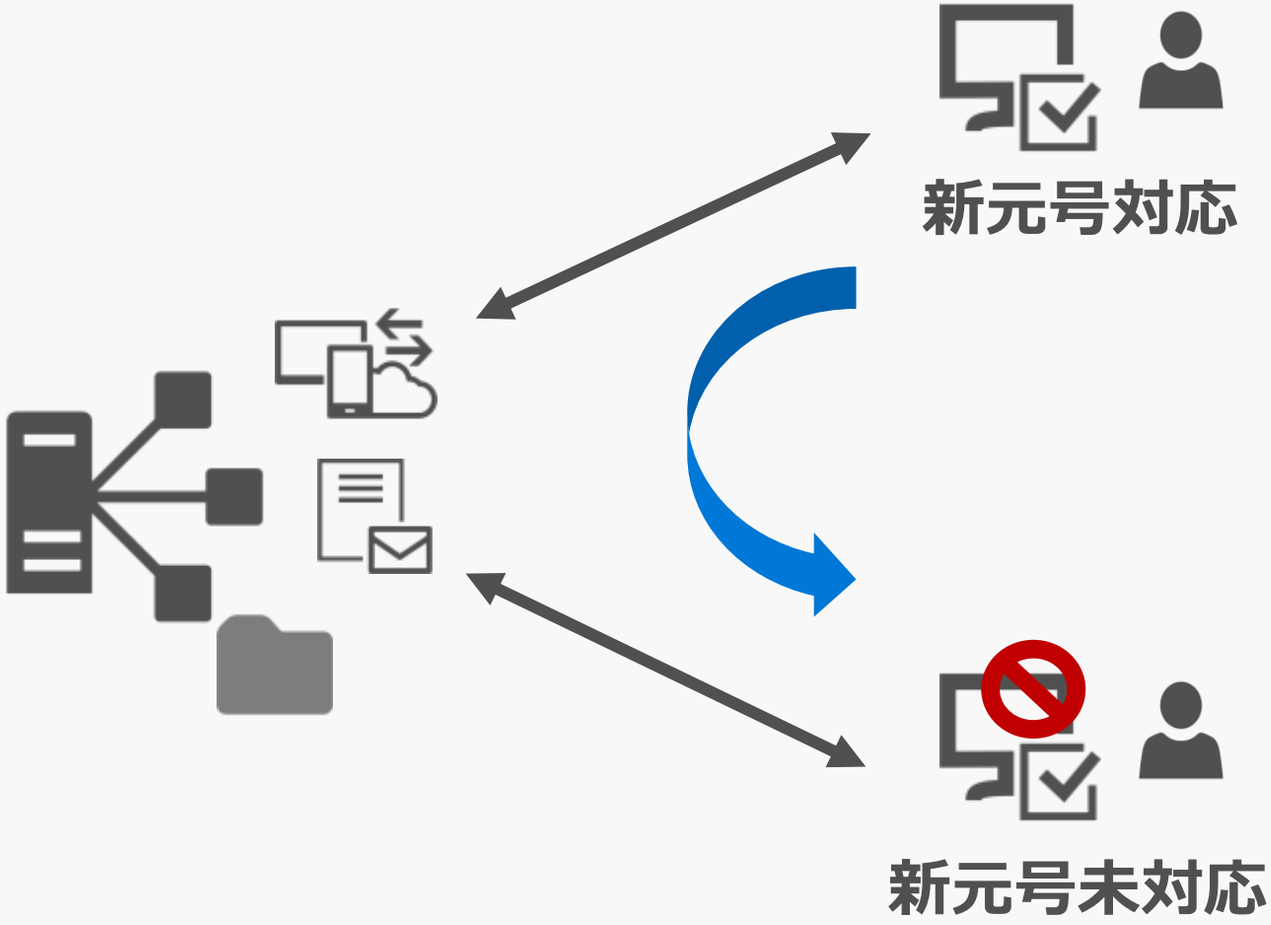
- データは必ずしも単独で存在しない。
  - Word、Excelはじめとする多くのマイクロソフト製品では関数、マクロをサポートする。

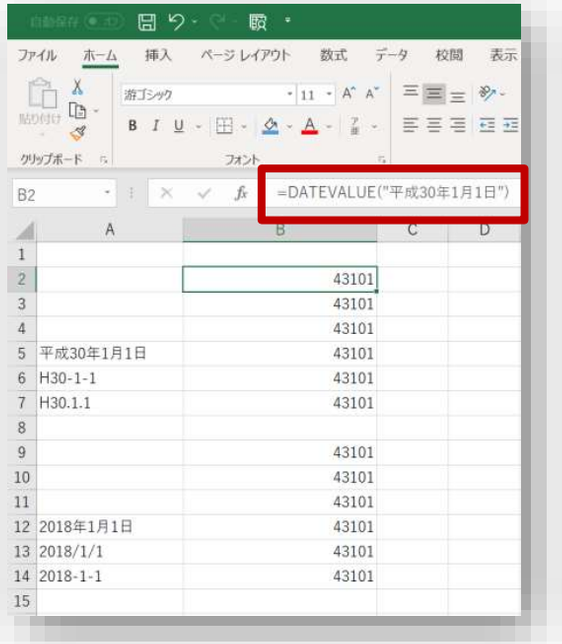
例：Excelマクロ、Outlookフォームなど

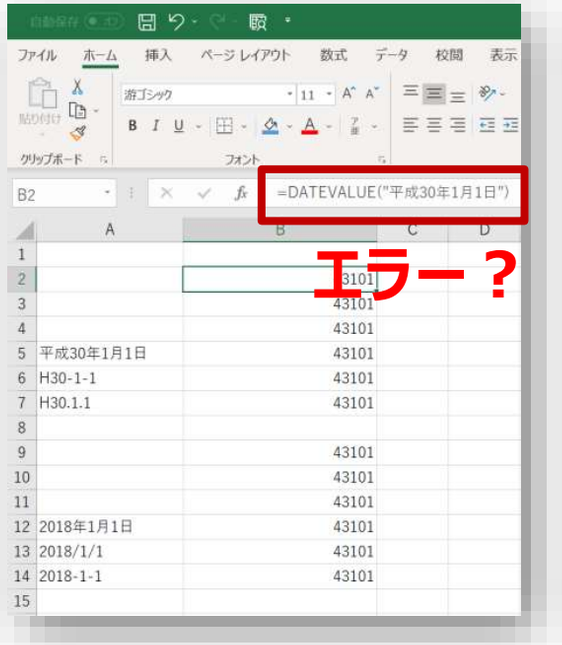


データにバインドされたスクリプト

- 同一組織内でも対応状況が異なる可能性も

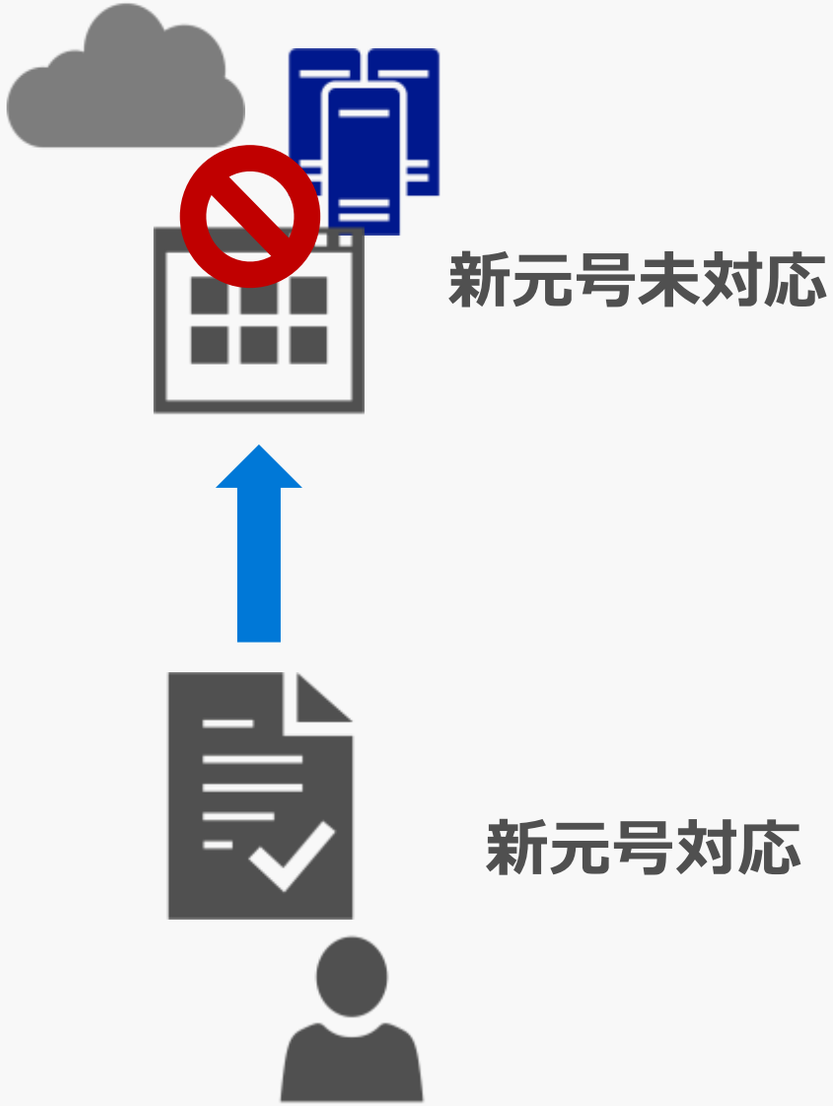






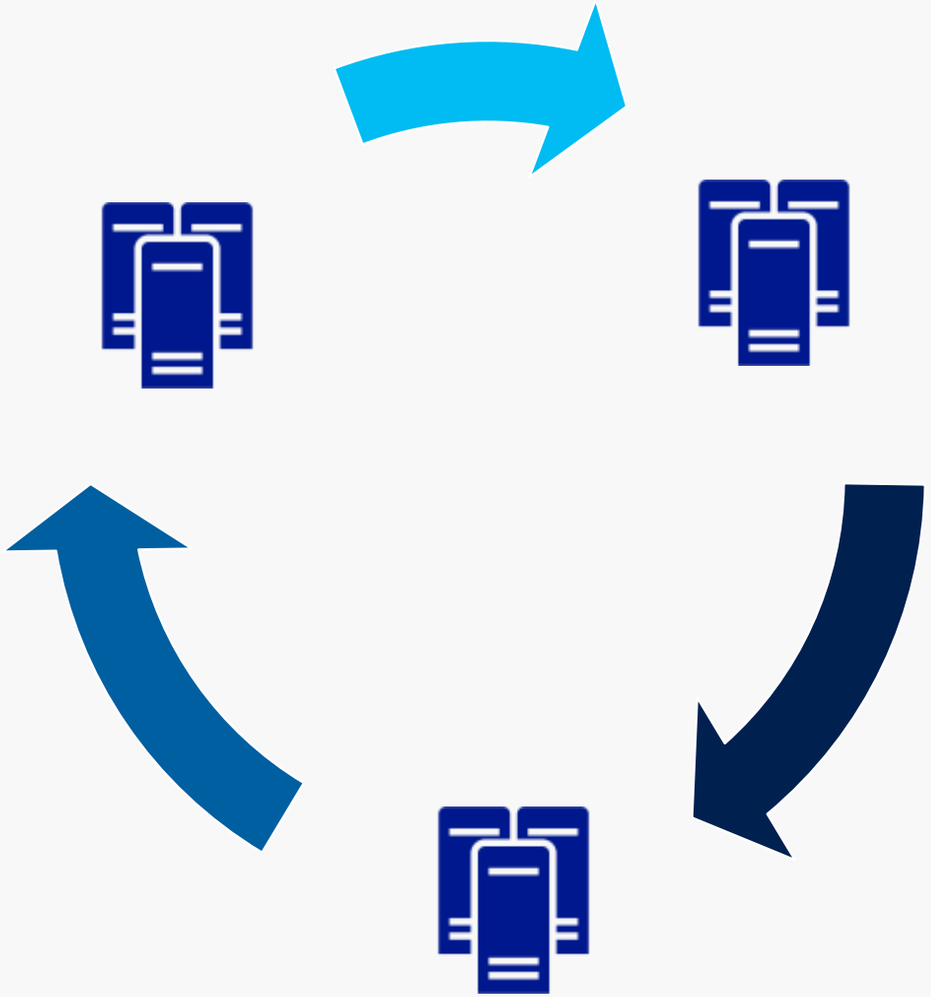
システム間相互運用　～Webサービスとクライアント～

- 様々なユーザー、システムと連携するサービスは、相互運用性の問題を慎重に評価する必要がある。
- ユーザー環境のシステムが新元号に対応済み、サービスが未対応の場合は、不通となる可能性が考えられる。



全ての依存関係を明らかにすることは難しい

全て**同時更新**が必要な場合も



循環した依存関係

新元号を使い始めるのはいつが良いのか

- 組織における新元号使用開始タイミング
  - システム更新順序

インストーラーと更新モジュール

- 新元号対応により更新されるモジュールは多岐に渡る
- MSIに再配布モジュールを含めている場合は注意が必要
  - .NET Framework
  - KBs

- ツールを確認
  - WiX
  - InstallShield
  - など

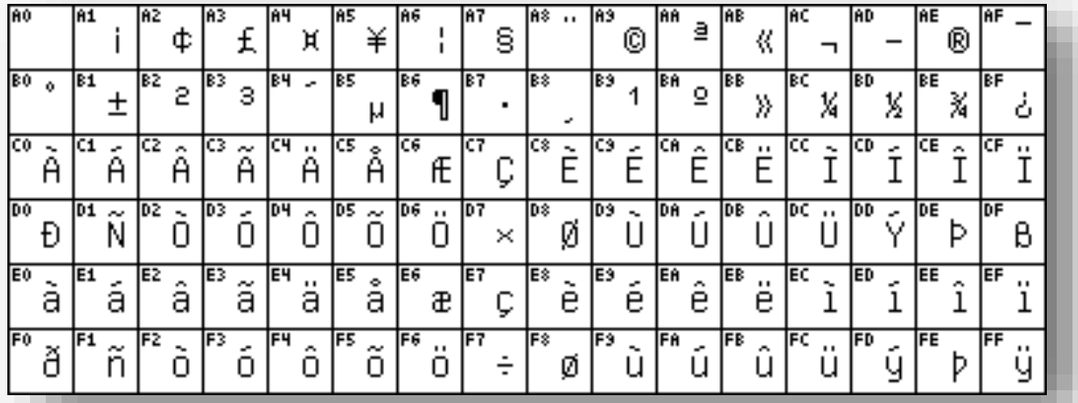
■ アプリケーション
■ 関連モジュール

■ パッチ
■ .NET Framework

改元と文字と標準

元号には1文字表記がある → 合字 (Ligature)

- 2つ以上のグリフから１つの文字を構成したもの
- ISO 8859 v.s. ASCII （主にdiacritical characters）



A0	A1 ¡	A2 ¢	A3 £	A4 ¤	A5 ¥	A6 ¦	A7 §	A8 ¨	A9 ©	AA ª	AB «	AC ¬	AD -	AE ®	AF ¯
B0 °	B1 ±	B2 ²	B3 ³	B4 ´	B5 µ	B6 ¶	B7 ·	B8 ¸	B9 ¹	BA º	BB »	BC ¼	BD ½	BE ¾	BF ¿
C0 À	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Ç	C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
D0 Ð	D1 Ñ	D2 Ò	D3 Ó	D4 Ô	D5 Õ	D6 Ö	D7 ×	D8 Ø	D9 Ù	DA Ú	DB Û	DC Ü	DD Ý	DE Þ	DF ß
E0 à	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 ç	E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
F0 ð	F1 ñ	F2 ò	F3 ó	F4 ô	F5 õ	F6 ö	F7 ÷	F8 ø	F9 ù	FA ú	FB û	FC ü	FD ý	FE þ	FF ÿ

AE → Æ		ij → ij	
ae → æ		st → ﬆ	
OE → Œ		ft → ﬁ	
oe → œ		et → &	
ff → ﬀ		ſs → ß	
fi → ﬁ		ffi → ﬃ	

明 治 ⟷ 明治

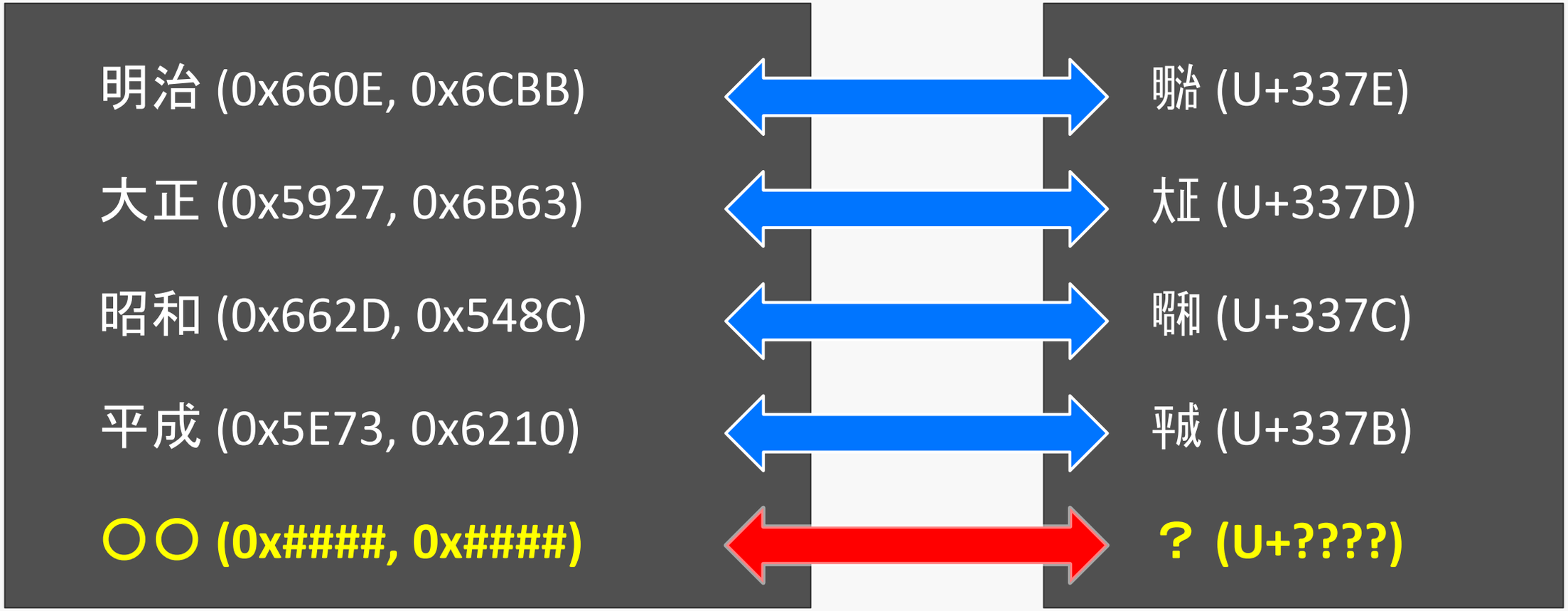
大 正 ⟷ 大正

昭 和 ⟷ 昭和

平 成 ⟷ 平成

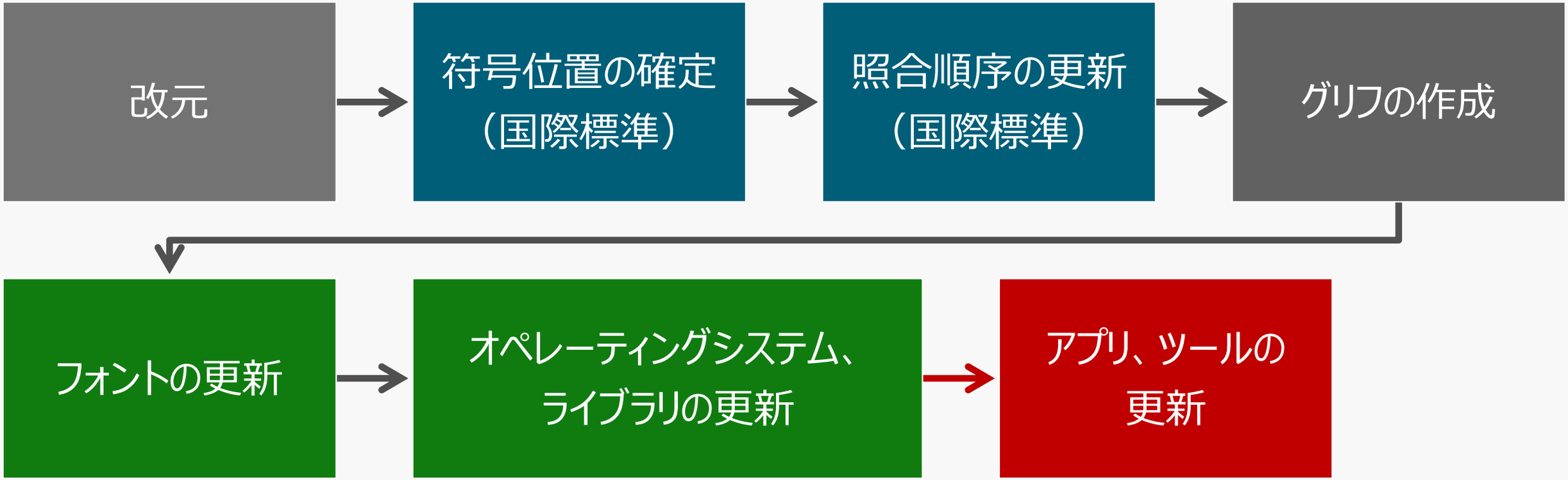
合字と符号位置

- ISO 10646 (ISO/IEC JTC1 SC2)



改元と文字

元号が変わることにより、新たに文字を追加しなくてはならない



考慮が必要となる主な項目

日付フォーマット変換

西暦 ↔ 和暦のフォーマット変換ルール

符号位置

Unicode（文字コードの国際標準）のどの場所に文字を追加するのか

照合順序

日付（和暦）による並べ替えの更新（明治→大正→昭和→平成→〇〇）

正規化

文字列検索、比較等の要素としての正規化

フォント

製品搭載フォントの更新（Windows、Office製品など）

共通実装

Unicodeのベンダー間共通実装であるICU (International Component for Unicode)の更新

日付フォーマット

- 目的、用途に応じて非常に多くのバリエーションが存在する日付フォーマット
- データ交換の原則に基づいた実装が必要

- 2019/11/28
- 2019-11-28
- 2018年11月28日
- 19年11月28日
- 平成30年11月28日
- 平成三〇年十一月二十八日
- H30.11.28

符号位置

- 文字の住所に相当する符号位置
  - 元号が変わる、即ち新たな文字（合字）の追加登録が必要
  - 現在登録されている合字の前後に空いているスペースはない

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+3360	8点	9点	10点	11点	12点	13点	14点	15点	16点	17点	18点	19点	20点	21点	22点	23点
U+3370	24点	hPa	da	AU	bar	oV	pc	dm	dm²	dm³	IU	平成	昭和	大正	明治	株式会社
U+3380	pA	nA	μA	mA	kA	KB	MB	GB	cal	kcal	pF	nF	μF	μg	mg	kg
U+3390	Hz	kHz	MHz	GHz	THz	μℓ	mℓ	dℓ	kℓ	fm	nm	μm	mm	cm	km	mm²
U+33A0	cm²	m²	km²	mm³	cm³	m³	km³	m/s	m/s²	Pa	kPa	MPa	GPa	rad	rad/s	rad/s²

新元号の合字はどこへ

- 当初、基本多言語面（BMP）以外の面（Plane）への追加を検討。即ち、サロゲートペアによる符号化（UTF-16）

Request to reserve one code point for Japanese new era name in BMP

Tetsuji Orita / JNB chair

Dear ISO/IEC JTC1/SC2 chair and Unicode Technical Committee chair,

Recently, Japanese government has decided the schedule of the abdication of current emperor and the enthronement of the new emperor. The date of abdication is scheduled on 31th April 2019, and the date of enthronement is on 1st May 2019.

With the final decision of the schedule, JNB requests kind consideration to reserve a new code point for the new era name in any position of BMP.

As you know, current ISO/IEC 10646 and Unicode Standard already standardized Japanese era names, 明治(U+337E),大正(U+337D),昭和(U+337C),平成(U+337B). These code points for era names are widely used in Japanese IT systems, especially relatively old fashioned, huge enterprise systems and embedded systems with small display device. Legacy system often stores the era information in a single character format (16-bit in UTF-16) into database as same date can be used for both of display and actual data. This has been very beneficial for data size sensitive solution, and an option often chosen.

Microsoft Japan investigated and heard its client enterprises the necessity to standardize the new era name in UCS/Unicode. There are not many but very strong requirements for standardization, with the strong demand to be encoded in BMP because of reasons noted above. Also many of legacy solutions do only support BMP.

JNB knows that the rest area of BMP is very small, however, it is very difficult and will waste huge unnecessary cost and time to replace current systems.

From the industrial and market view point, for the smooth and rapid transition of Japanese current enterprise IT systems, to have a new code point for the new era name in BMP is inevitable.

The new era name will be decided and announced before the enforcement with some lead time for preparation, but the schedule is not decided yet.

JNB will formally submit proposal document as soon as possible after the new era name will be decided. However, the lead time is expected to be very short, so, JNB requests kind consideration to reserve a new code position and publicly announce the code position for internal preparation of

符号位置の予約
~ Unicode ~

明治 (0x660E, 0x6CBB)	⟷	明治 (U+337E)
大正 (0x5927, 0x6B63)	⟷	大正 (U+337D)
昭和 (0x662D, 0x548C)	⟷	昭和 (U+337C)
平成 (0x5E73, 0x6210)	⟷	平成 (U+337B)
○○ (0x####, 0x####)	⟷	**？ (U+32FF)**

照合順序（Collation）

- 実装の更新が必要となる和暦の並べ替え
- 検索エンジン、データベースを始めとするデータの同期、マージに照合順序を使用するシステムは注意が必要。（これまでJIS X 0213などを除き、照合順序への影響は限定的）

正規化

UTR #15: Unicode Normalization

名称（英語）	日本語名称	説明
NFD Normalization Form Canonical Decomposition	正規化形式D	文字は正準等価性によって分解される。
NFC Normalization Form Canonical Composition	正規化形式C	文字は正準等価性によって分解され、再度合成される。結果として文字の並びが変換前と変わることもありうる。
NFKD Normalization Form Compatibility Decomposition	正規化形式KD	文字は互換等価性によって分解される。
NFKC Normalization Form Compatibility Composition	正規化形式KC	文字は互換等価性によって分解され、正準等価性によって再度合成される。

見た目が同じ

- 文字の見た目、意味共に全く同じ
  - しかしコード上は異なる
- 文字列を比較した時の期待される結果は？
- どのようなAPIを使用するべき？

が U+304C

が = か U+304B + ゛ U+3099

ë U+0451

ë = e U+0435 + ¨ U+0308

正規等価

意味がほぼ同じ

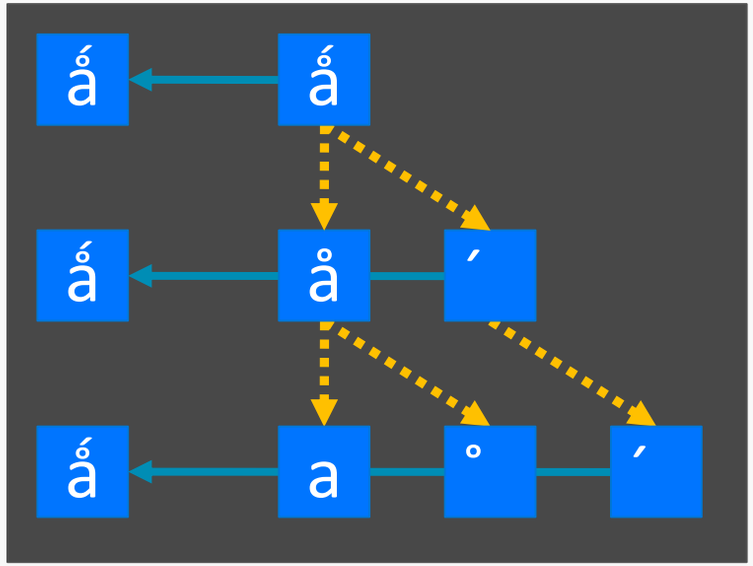
- 見た目は若干異なる
- 意味的にはほぼ同じ

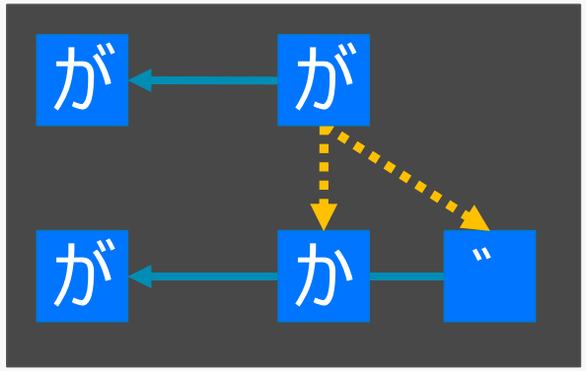
①	U+2460	1	U+0031		
カ	U+30AB	ｶ	U+FF76		
㎜	U+339C	m	U+006D	m	U+006D
㎜³	U+33A3	m m ³	U+00B3		
7日	U+33E6	7	U+0037	日	U+65E5
アンペア	U+3302	ア ン ペ ア			

互換等価

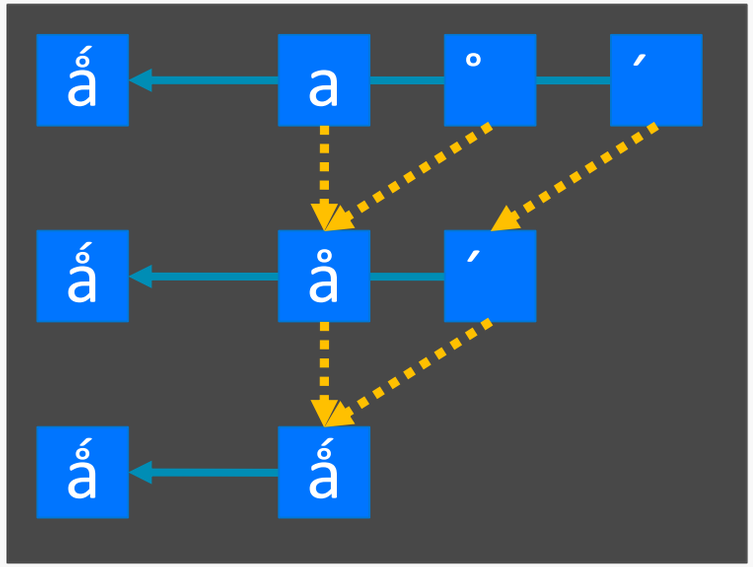
合成、分解

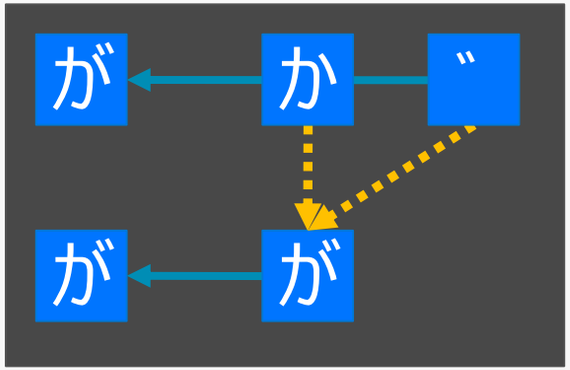
分解





合成





正規化

正規化と合字

名称（英語）	元	正規化後
NFD Normalization Form Canonical Decomposition	𠦪	𠦪
NFC Normalization Form Canonical Composition		𠦪
NFKD Normalization Form Compatibility Decomposition		平成
NFKC Normalization Form Compatibility Composition		平成

フォント – Unicode Standard Annex #50

~ Unicode Vertical Text Layout ~

- 縦書きグリフ（Rotated Glyph）の検討

U+337B SQUARE ERA NAME HEISEI	平成	平成
U+337C SQUARE ERA NAME SYOUWA	昭和	昭和
U+337D SQUARE ERA NAME TAISYOU	大正	大正
U+337E SQUARE ERA NAME MEIZI	明治	明治

ICU - International Components for Unicode

- CLDRを参照したUnicode、文字ハンドリングの為の共通実装

ICU is a mature, widely used set of C/C++ and Java libraries providing Unicode and Globalization support for software applications. ICU is widely portable and gives applications the same results on all platforms and between C/C++ and Java software.

http://site.icu-project.org/

ABAS Software, **Adobe**, **Amazon** (Kindle), Amdocs, Apache, Appian, Apple, Argonne National Laboratory, Avaya, BAE Systems Geospatial eXploitation Products, BEA, BluePhoenix Solutions, BMC Software, Boost, BroadJump, Business Objects, caris, CERN, CouchDB, **Debian Linux**, Dell, Eclipse, **eBay**, EMC Corporation, ESRI, **Facebook** (HHVM), Firebird RDBMS, **FreeBSD**, Gentoo Linux, **Google**, GroundWork Open Source, GTK+, Harman/Becker Automotive Systems GmbH, HP, Hyperion, **IBM**, Inktomi, Innodata Isogen, Informatica, Intel, Interlogics, IONA, IXOS, Jikes, Library of Congress, LibreOffice, Mathworks, **Microsoft**, Mozilla, Netezza, Node.js, Oracle (Solaris, Java), Lawson Software, Leica Geosystems GIS & Mapping LLC, Mandrake Linux, OCLC, Progress Software, **Python**, QNX, Rogue Wave, **SAP**, SIL, SPSS, Software AG, SuSE, Sybase, Symantec, Teradata (NCR), ToolAware, Trend Micro, Virage, webMethods, Wine, WMS Gaming, XyEnterprise, **Yahoo!**, Vuo, 他

CLDR - Unicode Common Locale Data Repository

Unicode CLDR は、様々な言語をサポートするソフトウェアのための主要なビルディングブロックをロケールデータの最大かつ最も広範なロケールデータの標準リポジトリともに提供しています。このデータは、ソフトウェアの国際化とローカリゼーションのために多くの企業によって使用されており、ソフトウェアをさまざまな言語、慣習に合わせて、一般的なソフトウェア開発に適応できます。

key building blocks for
rld's languages, with the
ndard repository of locale
ed by a wide spectrum of
companies for their software internationalization and localization, adapting software to the conventions of different languages for such common software tasks.

http://cldr.unicode.org/

CLDR

- **Locale-specific patterns for formatting and parsing:** dates, times, timezones, numbers and currency values
- **Translations of names:** languages, scripts, countries and regions, currencies, eras, months, weekdays, day periods, timezones, cities, and time units
- **Language & script information:** characters used; plural cases; gender of lists; capitalization; rules for sorting & searching; writing direction; transliteration rules; rules for spelling out numbers; rules for segmenting text into graphemes, words, and sentences
- **Country information:** language usage, currency information, calendar preference and week conventions, and telephone codes
- **Other:** ISO & BCP 47 code support (cross mappings, etc.), keyboard layouts

アプリケーション実装で必要なデータを提供

アプリケーション、オペレーティングシステム

Internationalization / Globalization ライブラリ (ICU)

Unicode ロケールデータ

Unicode エンコーディング

Unicode ロケール

- 日付、時間フォーマット
- 数値、通貨フォーマット
- 測定単位
- 照合順序: 並べ替え、検索、照合
- 言語名、地域、スクリプト、タイムゾーン、通貨の名前
- 言語、ロケールの一致
- …

元年（CLDR）

平成元年当時は、異なるシステム間の情報交換を目的とする「平成元年」、「平成１年」を含むデータはほぼ存在しない。

JIS漢字拡張など文字の相互運用も困難な時代

平成１年 OR 平成一年 OR 平成元年

マイクロソフトにおける主な対応

オペレーティングシステムと文字

MS-DOS
日本語版

1983年

第一世代 Shift JIS
(JIS78 or JIS83) + メーカー拡張

明治 大正 昭和

ITベンダー固有の拡張漢字が使用される。
これらの文字はJIS規格には含まれない。

平成

1989年1月8日

平成

「㍾」、「㍿」、「㍼」同様に初期のJIS規格には含まれない。（JIS X 0208, JIS X 0212)

Windows 3.0

1990年5月22日

互換性、相互運用性が担保されていない。

オペレーティングシステムと文字

Windows 3.0　1990年5月22日

Windows 3.1　1992年4月6日

日本語版　1993年5月18日

互換性、相互運用性が担保されていない。

相互運用可能に

第二世代 マイクロソフト 標準キャラクタ セット

JIS90, 10646

文字コード対応

- ベースとなる標準に準拠
  - マイクロソフトにより独自対応は行わない
  - Code Page 932 / 拡張文字を含むシフトJISにおける新元号対応は行わない
  - ISO-10646 / Unicode標準の対応に準じた更新を予定

	明治	大正	昭和	平成
JIS X 0208				
JIS X 0213	1面13区77点	1面13区78点	1面13区79点	1面13区63点
ISO-2022-JP (0201+0208)				
Windows-31J Code Page 932	878D	878E	878F	877E
Unicode	U+337E	U+337D	U+337C	U+337B

マイクロソフト標準キャラクタ セット

１年と元年

日付のデフォルトフォーマットが変わります

１年 → 元年

※ オプトアウト可能

フォントの更新

- 新元号対応予定フォント

フォント名
MS ゴシック
MS 明朝
MS Pゴシック
MS P明朝
Meiryo UI
Yu Gothic UI
他

MSGothic.ttc
MSMincho.ttc
Meiryo.ttc
Meiryob.ttc
YuGothR.ttc
YuGothM.ttc
YuGothL.ttc
YuGothB.ttc
yumin.ttf
yumindb.ttf
yuminl.ttf
UDDigiKyokashoN-R.ttc
UDDigiKyokashoN-B.ttc
BIZ-UDGothicR.ttc
BIZ-UDGothicB.ttc
BIZ-UDMinchoM.ttc

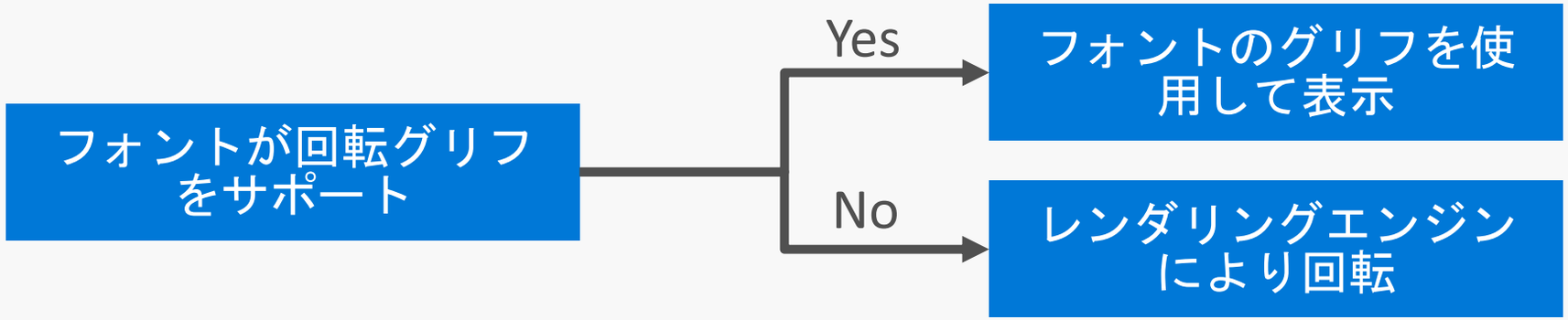
明治大正昭和平成□: MS 明朝
明治大正昭和平成□: MS ゴシック
明治大正昭和平成□MS UI Gothic
明治大正昭和平成□MS P 明朝
明治大正昭和平成□MS P ゴシック
明治大正昭和平成□ : Meiryo UI
明治大正昭和平成□: メイリオ
明治大正昭和平成☒ : Yu Gothic UI
明治大正昭和平成☒ : Yu Gothic UI Light
明治大正昭和平成☒ : Yu Gothic UI Semibold
明治大正昭和平成☒ : Yu Gothic UI Semilight
明治大正昭和平成☒: 遊ゴシック
明治大正昭和平成☒: 遊ゴシック Light
明治大正昭和平成☒: 遊ゴシック Medium
明治大正昭和平成☒: 遊明朝
明治大正昭和平成☒: 遊明朝 Demibold
明治大正昭和平成☒: 遊明朝 Light
明治大正昭和平成□: UD デジタル教科書体 N-B
明治大正昭和平成□UD デジタル教科書体 NK-B
明治大正昭和平成□UD デジタル教科書体 NK-R
明治大正昭和平成□: UD デジタル教科書体 NP-B
明治大正昭和平成□: UD デジタル教科書体 NP-R
明治大正昭和平成□: UD デジタル教科書体 N-R
明治大正昭和平成☒BIZ UDP ゴシック
明治大正昭和平成☒BIZ UDP 明朝 Medium
明治大正昭和平成☒: BIZ UD ゴシック
明治大正昭和平成☒: BIZ UD 明朝 Medium

明治大正昭和平成□: MS 明朝
明治大正昭和平成□: MS ゴシック
明治大正昭和平成□MS UI Gothic
明治大正昭和平成□MS P 明朝
明治大正昭和平成□MS P ゴシック
明治大正昭和平成□: Meiryo UI
明治大正昭和平成□: メイリオ
明治大正昭和平成☒ : Yu Gothic UI
明治大正昭和平成☒ : Yu Gothic UI Light
明治大正昭和平成☒ : Yu Gothic UI Semibold
明治大正昭和平成☒ : Yu Gothic UI Semilight
明治大正昭和平成☒: 遊ゴシック
明治大正昭和平成☒: 遊ゴシック Light
明治大正昭和平成☒: 遊ゴシック Medium
明治大正昭和平成☒: 遊明朝
明治大正昭和平成☒: 遊明朝 Demibold
明治大正昭和平成☒: 遊明朝 Light
明治大正昭和平成□: UD デジタル教科書体 N-B
明治大正昭和平成□UD デジタル教科書体 NK-B
明治大正昭和平成□UD デジタル教科書体 NK-R
明治大正昭和平成□: UD デジタル教科書体 NP-B
明治大正昭和平成□: UD デジタル教科書体 NP-R
明治大正昭和平成□: UD デジタル教科書体 N-R
明治大正昭和平成☒BIZ UDP ゴシック
明治大正昭和平成☒BIZ UDP 明朝 Medium
明治大正昭和平成☒: BIZ UD ゴシック
明治大正昭和平成☒: BIZ UD 明朝 Medium

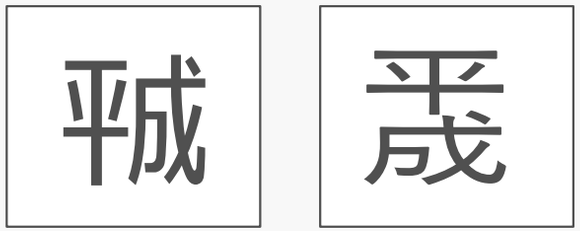
システム標準フォントは少なくない

縦書きと回転

- 縦書きにおいて、フォントが90度回転したグリフ（Rotated Glyph）を持っている場合と、そうでない場合の描画方法が異なる



- フォントによって、90度回転したグリフのデザインが異なる



各Windowsバージョンとフォントの関係

OSバージョンによりフォントバージョンも異なる

	Vista SP2 / Server 2008 SP2	Win7 SP1 / Server 2008 R2	Win8 / Server 2012	Win8.1 / Server 2012 R2	TH1 (1507)	RS1 (1607)/ Server 2016	RS2 (1703)	RS3 (1709)	RS4 (1803)	RS5 (1809)
MSGothic.ttc	5	5.01	5.1	5.1	5.1	5.1	5.1	5.11	5.11	5.11
MSMincho.ttc	5	5.01	5.1	5.1	5.1	5.1	5.1	5.11	5.11	5.11
Meiryo.ttc	5	6.02	6.12	6.2	6.2	6.26	6.26	6.27	6.27	6.28
Meiryob.ttc	5	6.02	6.12	6.2	6.2	6.26	6.26	6.27	6.27	6.28
YuGothR.ttc	NA	NA	NA	NA	1.71	1.72	1.8	1.81	1.82	1.83
YuGothM.ttc	NA	NA	NA	NA	1.71	1.73	1.8	1.82	1.82	1.83
YuGothL.ttc	NA	NA	NA	NA	1.71	1.72	1.8	1.81	1.82	1.83
YuGothB.ttc	NA	NA	NA	NA	1.71	1.73	1.8	1.8	1.82	1.83
yumin.ttf	NA	NA	NA	NA	1.7	1.7	1.7	1.71	1.71	1.71
yumindb.ttf	NA	NA	NA	NA	1.7	1.7	1.7	1.71	1.71	1.71
yuminl.ttf	NA	NA	NA	NA	1.61	1.61	1.61	1.71	1.71	1.71
UDDigiKyokashoN-R.ttc	NA	NA	NA	NA	NA	NA	NA	1	1	1
UDDigiKyokashoN-B.ttc	NA	NA	NA	NA	NA	NA	NA	1	1	1
BIZ-UDGothicR.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.03
BIZ-UDGothicB.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.03
BIZ-UDMinchoM.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	1.03

更新後のフォントバージョン

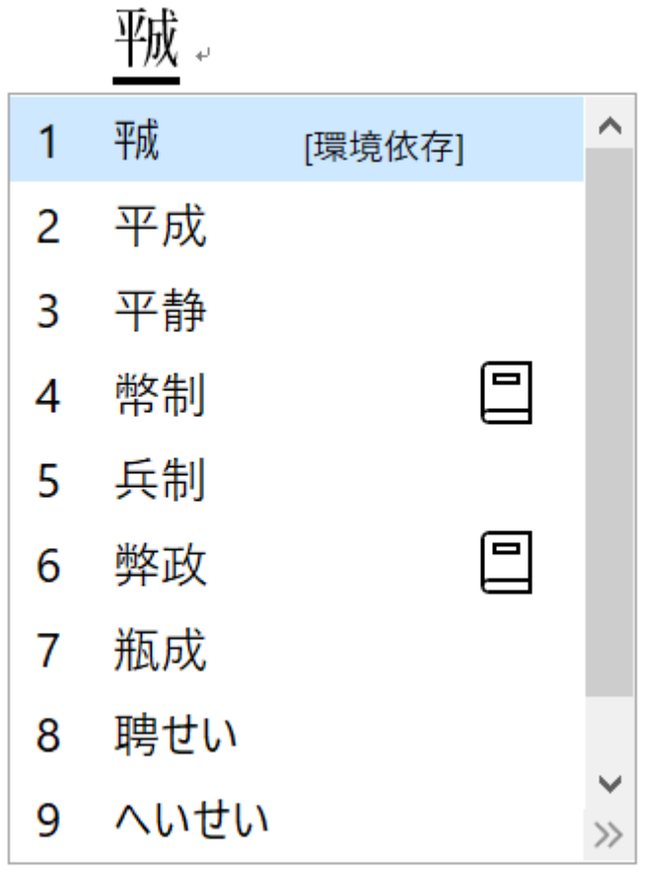
全てのフォントバージョンを更新

	Vista SP2 / Server 2008 SP2	Win7 SP1 / Server 2008 R2	Win8 / Server 2012	Win8.1 / Server 2012 R2	TH1 (1507)	RS1 (1607)/ Server 2016	RS2 (1703)	RS3 (1709)	RS4 (1803)	RS5 (1809)
MSGothic.ttc	5.02	5.06	5.15	5.16	6.00	6.00	6.00	6.00	6.00	6.00
MSMincho.ttc	5.02	5.06	5.15	5.16	6.00	6.00	6.00	6.00	6.00	6.00
Meiryo.ttc	5.02	6.06	6.16	6.26	7.00	7.00	7.00	7.00	7.00	7.00
Meiryob.ttc	5.02	6.06	6.16	6.26	7.00	7.00	7.00	7.00	7.00	7.00
YuGothR.ttc	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
YuGothM.ttc	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
YuGothL.ttc	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
YuGothB.ttc	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
yumin.ttf	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
yumindb.ttf	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
yuminl.ttf	NA	NA	NA	NA	2.00	2.00	2.00	2.00	2.00	2.00
UDDigiKyokashoN-R.ttc	NA	NA	NA	NA	NA	NA	NA	2.00	2.00	2.00
UDDigiKyokashoN-B.ttc	NA	NA	NA	NA	NA	NA	NA	2.00	2.00	2.00
BIZ-UDGothicR.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.00
BIZ-UDGothicB.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.00
BIZ-UDMinchoM.ttc	NA	NA	NA	NA	NA	NA	NA	NA	NA	2.00

※ バージョン番号は予定

入力変換（辞書の更新）

フォントを含む全ての更新作業後の対応



OSバージョンにより辞書フォーマットが異なる

実装、展開上のポイント

- 西暦、和暦の変換

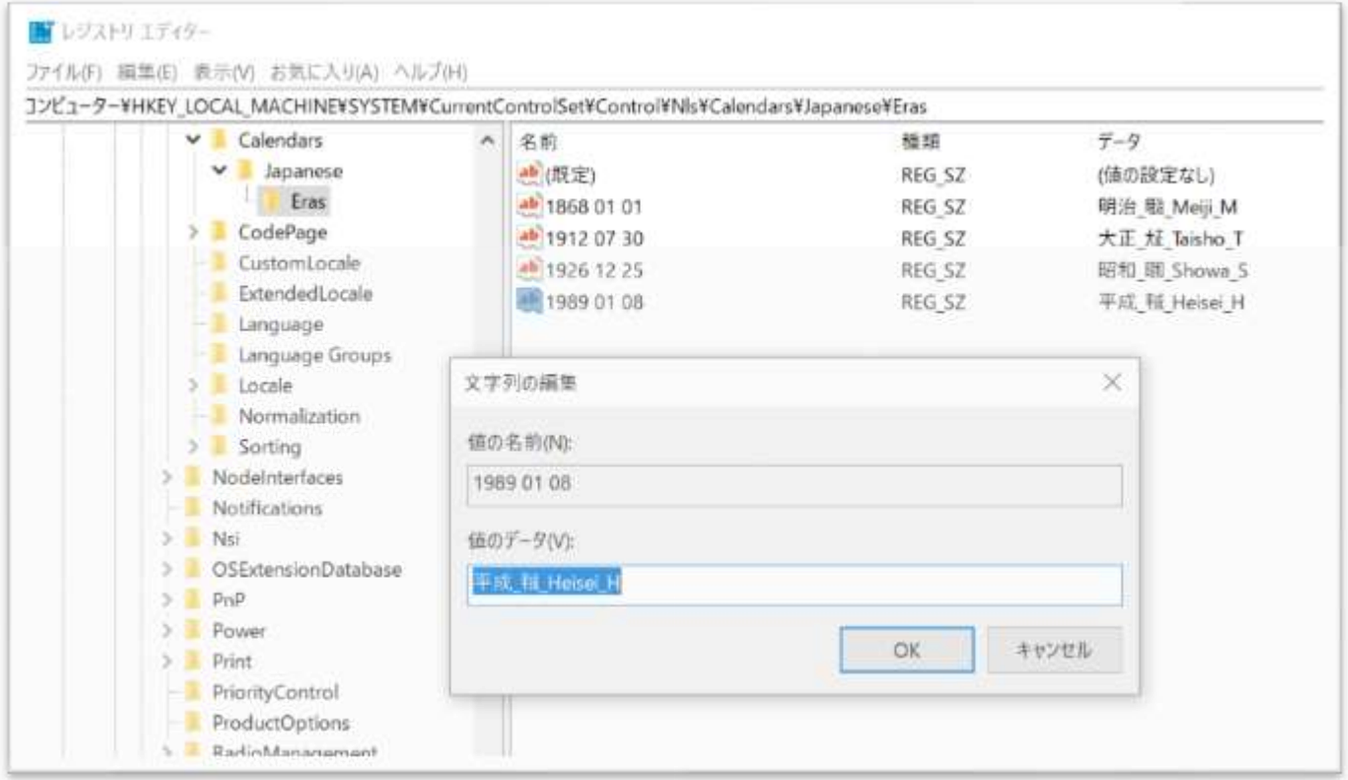
```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥Nls¥
Calendars¥Japanese¥Eras]

"1868 01 01"="明治_㬎_Meiji_M“
"1912 07 30"="大正_㐧_Taisho_T“
"1926 12 25"="昭和_㬎_Showa_S“
"1989 01 08"="平成_㦸_Heisei_H“
"2019 05 01"="？？_？_??????_?"
```

https://msdn.microsoft.com/en-us/library/windows/desktop/ee923790.aspx

.NET Framework

.NET Frameworkにおける振る舞いの変更

- 和暦の検証、フォーマットにおける振る舞いに変更があります。
- オプトアウトする手段を提供しています。
- 事前検証が可能です。



.NET Frameworkにおける振る舞いの変更

文字列から日付を読み取る場合

- 無効な和暦で例外が発生するDateTime.Parse()
  - 有効な和暦かの判定にしばしば使用される
  - 【注意】デフォルトで例外が発生しない実装へ変更
  - オプトアウト可能

```
var ciJapanese = new CultureInfo("ja-JP")
{
   DateTimeFormat = { Calendar = new JapaneseCalendar() }
};

// 読み取り対象の和暦による日付
// 日付は無効な年を設定
string formattedString = "昭和 70年2月1日 0:00:00";

// 文字列を読み込み DateTime オブジェクトに変換
DateTime dt = DateTime.Parse(formattedString, ciJapanese);
```

.NET Framework 4.5.2以前

- レジストリキーの設定によりオプトアウト可能
- デフォルトでキーおよび値の設定なし
- グローバル設定
  - アプリケーション毎の設定はできない

Key	HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥.NETFramework¥AppContext
Name	Switch.System.Globalization.EnforceJapaneseEraYearRanges
Type	REG_SZ
Value	1

.NET Framework 4.6以降のバージョン

- .NET Framework 4.6以降ではレジストリキーに依存することなく、DateTime.Parse()の振る舞いをコントロール可能
  - アプリケーションインスタンス毎に設定可能

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

<configuration>
  <runtime>

    <AppContextSwitchOverrides
        value="Switch.System.Globalization.EnforceJapaneseEraYearRanges=true" />

  </runtime>
</configuration>
```

.NET Core

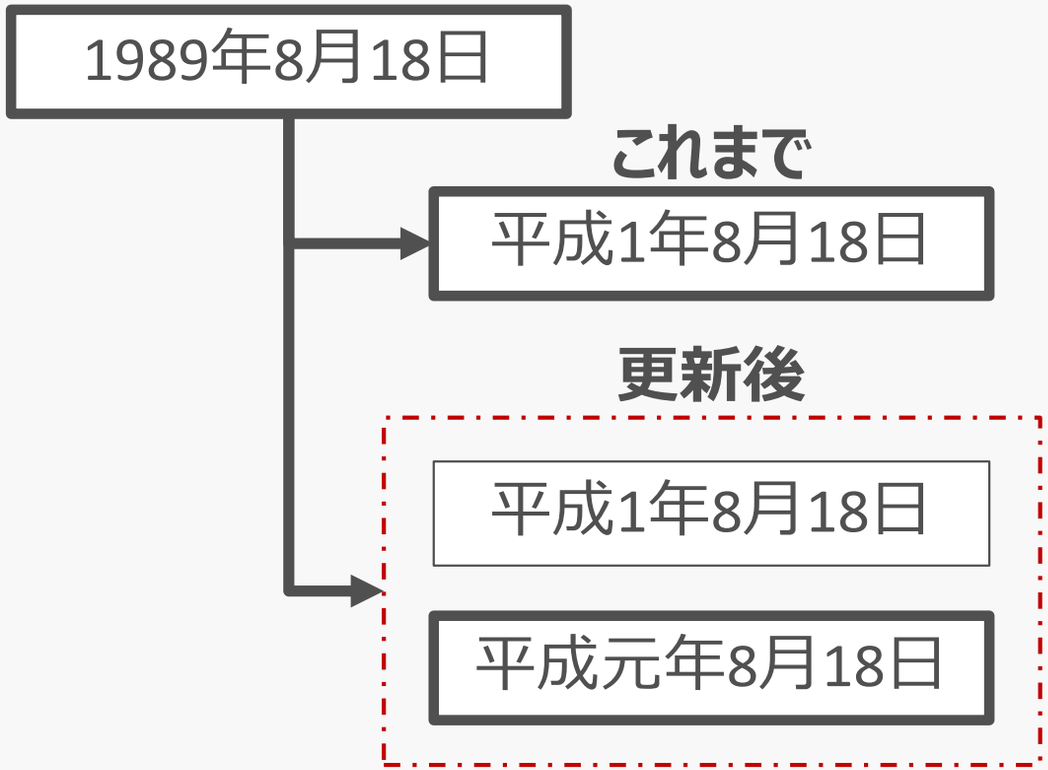
- ランタイムオプションによりデフォルトの振る舞いを変更（オプトアウト）可能

```
"runtimeOptions": {
   "configProperties": {

     "Switch.System.Globalization.EnforceJapaneseEraYearRanges": true

   }
}
```

.NET Frameworkにおける振る舞いの変更

日付データを文字列にフォーマットする場合

- 大きな変更の１つ
- デフォルトフォーマットの変更




```
using System;
using System.Globalization;

public class Example

{
    public static void Main()

    {
        var dat = new DateTime(1989, 8, 18);
        var cal = new JapaneseCalendar();
        var jaJP = new CultureInfo("ja-JP");

        jaJP.DateTimeFormat.Calendar = cal;

        CultureInfo.CurrentCulture = jaJP;

        Console.WriteLine($"{dat:ggy'年'M'月'd'日'}");
    }
}
```

.NET Framework 4.5.2以前

- レジストリキーの設定によりオプトアウト可能
- デフォルトでキーおよび値の設定なし
- グローバル設定
  - アプリケーション毎の設定はできない

Key	HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥.NETFramework¥AppContext
Name	Switch.System.Globalization.FormatJapaneseFirstYearAsNumber
Type	REG_SZ
Value	1

.NET Framework 4.6以降のバージョン

- .NET Framework 4.6以降ではレジストリキーに依存することなく、和暦変換時の「元年」、および「１年」の振る舞いをアプリケーションインスタンス毎にコントロール可能
  - アプリケーションインスタンス毎に設定可能

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

<configuration>
    <runtime>

        <AppContextSwitchOverrides
        value="Switch.System.Globalization.FormatJapaneseFirstYearAsANumber=true" />

    </runtime>
</configuration>
```

.NET Core

- ランタイムオプションによりデフォルトの振る舞いを変更（オプトアウト）可能

```
"runtimeOptions": {
   "configProperties": {

      "Switch.System.Globalization.FormatJapaneseFirstYearAsNumber": true

   }
}
```

.NET Frameworkにおける振る舞いの変更

元年 を含む文字列の読み取り時の例外処理

- 従来の実装では「元年」を 1 年と認識できず、例外が発生
  - 「元年」をフィルタリングするために例外を使用
  - 「元年」をマニュアルで返還後にParse
- .NET Framework更新後は「元年」をデフォルトで「1年」と同様に認識

これまで

○ 平成1年8月18日

✕ 平成元年8月18日

更新後

○ 平成1年8月18日

○ 平成元年8月18日

.NET Framework 4.5.2以前

- レジストリキーの設定によりオプトアウト可能
- デフォルトでキーおよび値の設定なし
- グローバル設定
  - アプリケーション毎の設定はできない

Key	HKEY_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥.NETFramework¥AppContext
Name	Switch.System.Globalization.EnforceLegacyJapaneseDateParsing
Type	REG_SZ
Value	1

.NET Framework 4.6以降のバージョン

- .NET Framework 4.6以降ではレジストリキーに依存することなく、「元年」を「1年」と正しく認識させるか、例外を発生させるかをアプリケーションインスタンス毎にコントロール可能
  - アプリケーションインスタンス毎に設定可能

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

<configuration>
    <runtime>

        <AppContextSwitchOverrides
        value="Switch.System.Globalization.EnforceLegacyJapaneseDateParsing=true" />

    </runtime>
</configuration>
```

.NET Core

- ランタイムオプションによりデフォルトの振る舞いを変更（オプトアウト）可能

```
"runtimeOptions": {
   "configProperties": {

        "Switch.System.Globalization.EnforceLegacyJapaneseDateParsing": true

   }
}
```

更新対象製品

元号対応における基本方針

- 2019年5月1日時点で延長サポートを終了していない製品
- 現時点で和暦に対応している

Windows

- Windows Updateによる対応を予定
- 累積パッチ

Windows Client

Windows 10, version 1809
Windows 10 Enterprise LTSC 2019
Windows 10, version 1803
Windows 10, version 1709
Windows 10, version 1703
Windows 10, version 1607
Windows 10 Enterprise LTSB 2016
Windows 10 Enterprise LTSB 2015
Windows 8.1
Windows 7 SP1

Windows Server

Windows Server 2019
Windows Server, SAC version 1809
Windows Server, SAC version 1803
Windows Server, SAC version 1709
Windows Server 2016
Windows Server 2012 R2
Windows Server 2012
Windows Server 2008 R2 SP1
Windows Server 2008 SP2

Windows Embedded & IoT

Windows 10 IoT Core, SAC version 1809
Windows 10 IoT Core, LTSC version 1809
Windows 10 IoT Enterprise, SAC version 1809
Windows 10 IoT Enterprise LTSC 2019
Windows 10 IoT Core, SAC version 1803
Windows 10 IoT Enterprise, SAC version 1803
Windows 10 IoT Core, SAC version 1709
Windows 10 IoT Enterprise, SAC version 1709
Windows 10 IoT Core, SAC version 1703
Windows 10 IoT Enterprise, SAC version 1703
Windows 10 IoT Core, SAC version 1607
Windows 10 IoT Enterprise, SAC version 1607
Windows 10 IoT Enterprise LTSB 2016
Windows 10 IoT Enterprise LTSB 2015
Windows Embedded 8.1 Industry Enterprise
Windows Embedded 8.1 Industry Pro
Windows Embedded 8.1 Pro
Windows Server 2012 R2 for Embedded Systems
Windows Embedded 8 Standard
Windows Server 2012 for Embedded Systems
Windows Embedded POSReady 7
Windows Embedded Standard 7 SP 1
Windows Embedded Compact 2013
Windows Embedded Compact 7

主な作業

2019年	2月24日	天皇陛下ご在位三十年記念式典	
	4月30日	退位の礼	**退位礼正殿の儀**
	5月1日	即位の礼	**剣璽等継承の儀**
	10月22日		**即位後朝見の儀**
			即位礼正殿の儀
	10月22日以降		**祝賀御礼の儀**
	10月23日		**饗宴の儀**
	11月14～15日	首相夫妻主催晩さん会	

オペレーティングシステム、
アプリケーションの更新作業

- ソリューションベンダーによる差分確認
- 実環境への展開

まとめ

重要なポイント

- 組織の中で使用されているソリューション、サービスの依存関係
- 更新の順番
- 累積パッチを適用することによる既存システムへの影響
- シフトJIS環境は新元号の合字（Square Glyph）をサポートしない
- 最初の年の扱いが変わる（1年→元年）

まとめ

- Windowsを含む製品、コンポーネントの更新が必要となります。
- .NET Frameworkなど再配布モジュールをインストーラーに組み込んでいる場合は、それらの更新が必要となります。
- OSとフォントの更新タイミングが異なります。

モジュール	ステータス
レジストリ	新元号発表後の早期に更新予定
フォント	新元号発表後に作業開始となるため、更新までに一定の時間を要する。
.NET Frameworkなどのライブラリ、API	作業可能な更新は既に開始している。

新元号検証ラボ

「新元号検証ラボ」

新元号発表前の検証を支援するプログラム

- 仮の元号を設定したWindows環境のご提供
  - 和暦・西暦変換
  - フォント（縦書きを含む合字対応）
  - IME辞書（拡張辞書による対応）、IME変換候補フォント変更ツール
  - Win32 API、.NET Framework等の更新された検証環境
  - サポート対象の全てのWindows環境（要事前指定）

開始場所：　日本マイクロソフト 品川本社
条件：　検証条件への合意、覚書への署名
検証環境：　Windows 7, 8および10
ブース、個室（セキュリティ要件により）
検証期間：　各社最大５営業日

参考情報

- Japan New Era Name Support Blog
  https://blogs.technet.microsoft.com/jperablog/

- The Japanese Calendar’s Y2K Moment
  https://blogs.msdn.microsoft.com/shawnste/2018/04/12/the-japanese-calendars-y2k-moment/

- Handling a new era in the Japanese calendar in .NET
  https://blogs.msdn.microsoft.com/dotnet/2018/11/14/handling-a-new-era-in-the-japanese-calendar-in-net/

Microsoft