Release Management Group Report to UTC #185

October 22, 2025
Peter Constable & UTC Release Management Group

The Release Management Group (“RMG”) is composed of UTC working group leaders and other volunteers working on the process for development of the Unicode 17.0 release and for evolving processes for future Unicode Standard releases.

1. Unicode 17.0 release

Unicode 17.0 was released on September 9, according to plan. French charts typically lag behind the main release, and were published in mid-October.

There were some minor glitches on release day, mainly related to the reorganization of published data files as approved by UTC #182 (see 182-C2): there were details related to the Public/zipped folder for which a plan had not been thoroughly worked out in advance. This was addressed and will not be an issue for future releases.

Also, shortly after the release it was found that some references within the core spec to the current version were incorrectly referring to Unicode 16: there were a handful of places in which the version number was hard-coded to “16.0.0” rather than using a parametric version component. This was quickly fixed and will not be an issue for future releases. See the top of https://www.unicode.org/versions/Unicode17.0.0/erratafixed.html.

2. Process development

After each new Unicode version release (including certain post-release-date tasks), it has been RMG’s practice to conduct a retrospective assessment of the just-completed release cycle—what has worked well, what could be improved, and which potential improvements should be prioritized for the next release cycle. In this section, we’ll summarize some of the outcomes from that exercise.

During this retrospective, RMG revisited Ken Whistler’s “White Paper on Unicode Release Management Issues” which he presented to UTC back in 2021. In that presentation, he identified release-related responsibilities for which UTC would need to plan to transition to new owners. Status in relation to that longer-term transition was a key focus of this year’s retrospective, and will be discussed in more detail in sections 2.1 – 2.4.

In relation to long-term transition and sustainability, two important areas that were *not* discussed in Ken's 2021 presentation were (i) processes for editing and publication of the Core Spec, and (ii) processes for preparation of the code charts. There has been a lot of improvement in both of these areas since 2021, most of which were in place by the Unicode 16.0 release and covered in RMG's report to UTC #181 (L2/24-258R). That will not be repeated here.

2.1 Ken's five areas (fall 2021)

Ken had mentioned five key areas of responsibility:

1. **Overall project management** for the release: identifying each essential task for the release, tracking status and driving progress to completion.
2. **Core spec managing editor**: driving triage and being a final decision maker on content changes to include in the core spec for a given release.
3. **Core data file generation:** drafting values for all UCD properties for new characters added to a release—something that requires comprehensive understanding of all of the properties involved.
4. **Data deployment and QA management:** validating completeness and consistency of all the data files and ensuring all of the correct files are published at the right times.
5. **Technical report document management:** shepherding all of the UTRs, UAXes and UTSes through preparation of updates and through each preview and release publication process.

Since 2021, good progress has been made in three of these areas (1, 2, 4); some details will be provided in section 2.2. In RMG's assessment, however, the other two areas require more improvement; these will be discussed in sections 2.3 and 2.4. Finally, section 2.5 will summarize process improvements that RMG recommends be prioritized for the Unicode 18.0 cycle.

2.2 Areas with good progress

RMG feels the following areas are in good shape: overall project management, editorial management of the core spec, and property data QA and deployment.

2.2.1 Overall project management

Whereas in the past overall project management had been handled by Ken, RMG has been collectively handling project management for three full cycles now. While there is a long list of tasks involved in each release, the collective understanding of the task list has matured such that the group is confident that there is no longer a critical dependency on Ken as a single point of failure.

2.2.2 Editorial management of the core spec

In the past, this was handled primarily by Julie Allen with assistance from Ken. There has been significant progress over the past few release cycles in two respects:

- As covered in RMG's report to UTC #181, the tooling and production processes for the core spec were completely modernized during the Unicode 16.0 cycle such that there is no longer a critical dependency on one person working in Adobe FrameMaker, as was the case in the past. That has enabled more collaborative ownership of content changes within the Editorial Working Group.

- By the end of the Unicode 17.0 cycle, the Editorial Working Group has matured with multiple people having senior editor roles, with Liang Hai as lead, working effectively together to manage work on the core spec.

2.2.3 Data QA and deployment

There has been a lot of progress in this area. Over the Unicode 16 and 17 release cycles, the deployment of data files was streamlined significantly. Data files used to be individually updated, published, and tracked. Now a small number of sets of related files are developed, generated, and tested together, deployed in whole sets, and, for final publication, placed onto a staging server well ahead of the release day. This has significantly improved consistency, reduced tracking overhead, and simplified release day deployment.

Also, in the past, a published snapshot of data files could include updates to some files without corresponding changes in other files (e.g., derived properties). This made it difficult for reviewers or downstream consumers to test with those data files except at certain points in time when the set was declared complete. Now, as a result of changes to how data files are generated and deployed, previews of updated data files can be made available sooner, updated more frequently, and always with assurance that the set of data files is complete and self-consistent.

2.3 Core data file generation: some progress, but further improvement needed

When UTC takes decisions to encode new characters, data for those characters needs to be generated for many different files. In the past, this was largely handled by a few people, according to the files involved:

- Unihan: John Jenkins (who is no longer contributing), plus Michel Suignard for CJK sources.
- UCDXML: Eric Muller and Laurențiu Iancu (who are no longer contributing)
- DUCET: Ken Whistler
- NamesList: Ken Whistler and Michel Suignard
- Select data files (e.g., security, confusables, emoji…): Mark Davis
- Remainder: Ken Whistler

Some of these contributors are no longer available, and Mark and Ken have been winding down their contributions in recent years. Some improvements have been imposed by necessity in regard to processes and ownership of the work. The following are noteworthy improvements that have been made:

- Unihan: Ken Lunde has carefully documented processes for much of the Unihan data.
- UCDXML: Whereas the data, and UAX #42, were previously generated by Eric or Laurențiu using Eric's private tooling, John Wilcock has created new tooling in the unicodetools repo with automated builds.
- A set of PAG members have collectively taken over preparation of most of what Ken and Mark had previously prepared. Further, all tools that were used by Ken Whistler to produce these files in the past are now source-controlled in a repository under Unicode control. (Mark's tools have become the unicodetools, which are actively used in development of the bulk of UCD data and data associated with synchronized UTSes.) Roozbeh Pournader used to produce some files using his own tooling too (Indic properties and ScriptExtensions.txt); these are now handled normally by the unicodetools.
- The drafting of UCD data for new proposals was documented but had only been done by Robin Leroy in the past year; however, John Wilcock has recently started contributing to that task.

However, there is still need to further improve on expertise concentrated in select individuals. These are key areas that deserve attention:

- Ken Whistler and Michel are still on the critical path for the generation of NamesList.txt.
- The code for confusables data is complex; while there is still some dependency on Mark for knowledge of the code base, maintenance has recently been handled by Roozbeh and Josh Hadley.
- Unihan: While Ken Lunde has documented processes for Unihan, he is still the only one that has *run* those processes. Someone else should run the processes to validate the documentation.
- Emoji: The tooling was written by Mark, and its usage is documented; however, only Ned Holbrook has actually performed the documented steps.
- DUCET: Markus Scherer has run the tooling, but we still have critical dependency on Ken Whistler for preparation of data when adding new characters.

As there is opportunity, some improvements in these areas may be considered during the Unicode 18.0 cycle. In the case of **NamesList.txt**, RMG recommends that the Charts group be designated as owners for this file, but does *not* recommend planning for tooling or process changes related to generation of this file during the Unicode 18.0 cycle.

2.4 Needing improvement: document management for technical reports

There are two aspects of the technical reports that would benefit from improvement.

- These documents are maintained as plain HTML files, without the benefit of componentization or other modern Web technologies that are now used for the core spec.
- The content is maintained in a GitHub repo, providing version control *in a low-level sense*. However, management of the document versions / revisions and draft numbers is entirely manual. Also, processes for deployment are also manual.

There has been some preliminary exploration into the former, exploring the possibility of migrating the technical reports into the same processes and framework used for the core spec.

However, RMG recommends prioritization in the latter aspect. This represents a significant portion of the concerns for this area that Ken called out back in fall 2021. Other impacted parties, including the Editorial Working Group and Unicode support staff, also consider this a priority.

2.5 Priorities for process improvement during the Unicode 18.0 cycle.

Having reviewed what did or didn't go well during the Unicode 17.0 cycle, the status on longer-term challenges, and other longer-term opportunities, RMG has identified the following as process improvement goals for the 18.0 cycle:

- Technical report document management: Set a goal to automate document version management and deployment for UAXes, UTSes and UTRs. This would need to satisfy certain requirements (see below) and result in significant improvement and efficiency for all stakeholders collectively.
    - Requirements include:
        - Ease of use and benefit for document authors and owning working groups, not just RMG and supporting Unicode staff.
        - Must work for final releases and also for preview releases, including integration with PRI processes.
        - Need to continue to have stable snapshots of updates used as reference for UTC decisions.
    - If that goal can't be achieved, then a fallback goal will be to have better documentation for existing processes.
- Continue to improve staging processes that simplify release-day tasks and allow for earlier validation.

- Explore possibilities for new generation / deployment mechanisms for auxiliary charts. (E.g., leverage representative glyph collection now used for core spec.)

In the course of retrospective discussions, RMG also considered current maintenance of the Pipeline page. Updates to this page are driven directly from UTC decisions. For that reason, Robin Leroy, as UTC recording secretary, proposed that he take over maintenance of that page from Ken Whistler.

RMG also discussed other requirements related to the status of code points. Between the Pipeline on the one hand, and UCD data and charts on the other, the status of code points and characters that UTC has taken action on are captured—prior to a new release and after release, respectively. Yet there is some need to track status of *all* code points. The Roadmap pages provide that to some extent, but at a coarse level of granularity. SEW, in particular, has a regular need to recommend available code points for proposed new characters, and to check for name conflicts.[1] Jan Kučera has indicated interest in investigating possible tooling to support this need.

3. Unicode 18.0 scope and timeline

Unicode 18.0 should be planned as a “full” release including new characters and scripts…

A timeline for the development and release of Unicode 18.0 is proposed with the following key dates:

- 2025-10-29, UTC #185: Confirm scope and timeline for Unicode 18.0
- 2025-12-29: snapshot of PRI feedback
- 2026-01-23, UTC #186: Finalize alpha content
    - New characters / emoji, charts
    - Property data
    - Advance in-progress UTSes, UAXes
- 2026-02-10: Start of alpha
- 2026-03-31: End of alpha, snapshot of feedback
- 2026-04-23, UTC #187: Finalize beta content
    - Charts
    - Property data
    - Draft UTSes, UAXes
    - Draft of complete core spec content
- 2026-05-26: start of beta review
- 2026-07-07: end of beta public review, snapshot of feedback
- 2026-07-30, UTC #188: Finalize 18.0 content
- 2026-09-15: Unicode 18.0 release

[1] CJK & Unihan WG has a similar need, but the challenge for them is much less complicated than for SEW.

RMG proposes that UTC #185 take the following action:

> [184-C] Consensus: UTC approves the proposed timeline and plan for Unicode 18.0 outlined in document L2/25-234.

4. Recommendation of new repertoire for 18.0

In RMG's report to UTC #183 (L2/25-093R), we gave a preliminary recommendation for new repertoire to be encoded in Unicode 18.0. Subsequently, UTC #184 took the following actions accepting characters for version 18.0:

> **[184-C2] Consensus:** Of characters approved for encoding in Unicode 17.0 (ref. 181-C60, 181-C61), the following 43 characters are removed from Unicode 17.0 and are targeted instead for Unicode 18.0:
>
> 1. 09FF BENGALI LETTER SANSKRIT BA
> 2. 0B53 ORIYA SIGN DOT ABOVE and 0B54 ORIYA SIGN DOUBLE DOT ABOVE
> 3. 40 Chisoi script characters at 16D80..16DA9 and the Chisoi block at 16D80..16DAF.
>
> **[184-C17] Consensus:** UTC accepts U+20C3 UAE DIRHAM SIGN for encoding in the Currency Symbols block based on L2/25-159, for Unicode Version 18.0. [Ref. 2.1 in L2/25-187]
>
> **[184-C19] Consensus:** Accept 16 code points U+1F7DB and U+1F7F1..U+1F7FF in the Geometric Shapes Extended block as described in WG 2 N5330, for Unicode Version 18.0. [Ref. 2.3 in L2/25-187]
>
> **[184-C26] Consensus:** U+1FADD APPLE CORE, approved for encoding in Unicode Version 17.0 (ref. Consensus 181-C5), is removed from Unicode Version 17.0 and is targeted instead for Unicode Version 18.0.

Thus, 61 characters are already accepted for encoding in Unicode 18.0, but none of the characters previously given provisionally assigned code points are as yet accepted for Unicode 18.0. RMG recommends that UTC approve 1,597 characters for encoding in Unicode 18.0 for which code points have been provisionally assigned, as described in the proposed actions shown below.

RMG proposes that UTC #185 take the following actions:

[185-C] Consensus: UTC accepts for encoding in Unicode 18.0 the following 1,276 characters for new scripts for which code points have previously been assigned:

- 311 characters for Archaic Cuneiform Numerals at 12550..12686 (ref. 181-C31)
- 914 characters for Jurchen at 18E00..19191 (ref. 180-C28)
- 51 characters for Jurchen radicals at 191A0..191D2 (ref. 180-C29)

[185-C] Consensus: UTC accepts for encoding in Unicode 18.0 the following 321 Arabic, Armenian, Bengali, Cuneiform, Devanagari, Hebrew, Kana, Khitan, Latin, Mongolian, Phonetic and other symbol characters for which code points have previously been assigned:

- Arabic (39 characters—ref. 180-C22, 180-C26): 10EC9..10ECF, 10ED9..10EEE, 10EF0..10EF9
- Armenian (3 characters—ref. 179-C46): 0558, 058B..058C
- Bengali (1 character—ref. 180-C30): 0984
- Cuneiform numerals (12 characters—ref. 182-C3): 1246F, 12475..1247F
- Devanagari (1 character—ref. 182-C5): 11B0A
- Hebrew (1 character—ref. 182-C4): 05C8
- Kana (7 characters—ref. 180-C6, 182-C31, 183-C54, 184-C38): 1B123..1B125, 1B126, 1B127..1B128, 1B168
- Khitan (5 characters—ref. 184-C5): 18CD6..18CDA
- Latin (54 characters—ref. 181-C8, 181-C10, 182-C6, 182-C7, 182-C8, 182-C9, 183-C8):
  2E60..2E63, A7DD, A7E2, AB6C..AB6D, 1DF57..1DF59, 1DF5A..1DF66, 1DF67, 1DF68..1DF81, 1DFCD..1DFCF
- Mongolian (1 character—ref. 178-C30): 1879
- Phonetic (114 characters—ref. 179-C55, 179-C59, 179-C60, 180-C32, 180-C33, 180-C34, 180-C35, 180-C36, 180-C37, 181-C33, 181-C34, 181-C35, 181-C36, 181-C45, 183-C10):
  1ADE..1ADF, 1AEC..1AF0, 208F, 209D..209F, 107BB..107BF, 1DF1F..1DF24, 1DF2D..1DF3A, 1DF3B..1DF3D, 1DF3E..1DF3F, 1DF40..1DF56, 1DFD0, 1DFD1, 1DFD2..1DFD7, 1DFD8..1DFE8, 1DF39..1DFF2, 1DFF3..1DFF4, 1DFF5..1DFF9, 1DFFA..1DFFF
- Symbols (81 characters—ref. 178-C31, 178-C36, 178-C37, 180-C38, 180-C39, 180-C40, 181-C38, 181-C39, 181-C40, 182-C10, 182-C11, 183-C12, 183-C13, 184-C18):
  20C2, 1CEF1..1CEF5, 1D127..1D128, 1D1EB..1D1F6, 1D1F7..1D1FE, 1D1FF, 1D250..1D255, 1D256..1D25A, 1D25B..1D25F, 1D260, 1D261, 1D262..1D27F, 1D280..1D281, 1F1AE, 1F7DA
- Tangut (2 characters—ref. 183-C7, 184-C4: 18D1F..18D20

[185-A] Action Item for Robin Leroy, RMG: Update the pipeline to reflect the assignment of provisionally assigned code points for encoding in Unicode 18.0. [Ref. consensus 185-C??, 185-C??]

[185-A] Action Item for Markus Scherer, PAG: Update the relevant UCD data files in preparation for Unicode version 18.0 to include the additional characters approved for encoding. [Ref. consensus 185-C??, 185-C??]

Note 1: RMG anticipates that, during UTC #185, the Script Encoding Working Group could independently propose additional characters be accepted for encoding in Unicode 18.0. One script RMG anticipates being recommended by SEW is Seal, with a very large repertoire but relatively simple properties, and charts already drafted.

Note 2: ISO/IEC JTC1/SC2 has had a new edition of 10646 in development. The characters mentioned in 184-C2 above that had previously been approved for Unicode 17.0 were deferred to Unicode 18.0 due to national body comments on drafts of the new edition of 10646 either requesting more time for review or, in the case of 09FF, requesting that the character be removed from the draft. Until we get confirmation that these characters are no longer a NB concern, we must assume there is risk of getting approval for the characters on the SC2 side in time for the 18.0 release. Status of these characters should be reviewed by UTC #187, prior to the Unicode 18.0 beta.