confschedule

Conference Schedule and Proceedings Management

2025/05/16 v1.0

Abstract

confschedule manages the full pipeline for conference proceedings: submission files are ingested by a LuaTEX module (`process-submissions.lua`) that populates a database of sessions, talks, and participants; the package then typesets schedule grids, session and abstract listings, and a participant index from that database. All session types, colours, and formatting hooks are user-configurable.

Contents

1 Usage Example

The following minimal example shows the complete workflow.

1.1 Session submission files

Each session corresponds to a subdirectory whose name is the session id. The directory contains any number of `.tex` files, each holding exactly one environment (`session` or `talk`). Files are processed in sorted (alphabetical) order. Files whose name begins with digits are assigned a talk index; gaps in the numeric sequence cause `\missingTalk` to be emitted automatically. Files without a leading digit (e.g. `session.tex`) are processed but do not consume a talk index slot. Any content outside the environments in a file is silently ignored.

A typical session directory looks like this:

```
sessions/
  contrib-1/
    session.tex     <- session metadata and description
    01.tex          <- first talk
    02.tex          <- second talk
    04.tex          <- fourth talk (03.tex absent: \missingTalk{3} emitted)
```

Session file (`session.tex`) Contains one `session` environment. The body is the session abstract or description; it is re-typeset verbatim by `\renderSessions`.

```
\begin{session}
  {Numerical Optimisation}     % session title
  {Jane Smith}                 % chair (first organiser)
  {MIT}                        % organiser affiliation
  {j.smith@mit.edu}            % organiser email
  {}{}{}                       % second organiser (leave blank to omit)
This session covers recent advances in first-order methods for
large-scale convex and non-convex problems.
\end{session}
```

Talk file (`01.tex`) Contains one `talk` environment. The body is the abstract; it is re-typeset verbatim by `\renderTalks`.

```
\begin{talk}
  {A Fast Gradient Method}     % title
  {Alice Brown}                % speaker
  {Stanford University}        % affiliation
  {a.brown@stanford.edu}       % email
  {C.~White}                   % coauthors (empty if none)
```

```
  {}                            % special note (empty if none)
We present a gradient method achieving optimal complexity for
smooth strongly-convex objectives.
\end{talk}
```

Cancelled talk (`02.tex`) Use `\cancelTalk` in the optional hook argument.

```
\begin{talk}
  {Convergence Rates}
  {Bob Green}{Cambridge}{b.green@cam.ac.uk}{}{}
  [\cancelTalk]
This abstract will not be rendered and the talk will appear
with strikethrough in the schedule.
\end{talk}
```

1.2 Main document

```
\documentclass{book}
\usepackage{confschedule}

% Register session types with display label and background colour.
\defineSessionType{contrib}{Contributed Talk}{blue!15}
\defineSessionType{invited}{Invited Talk}{green!20}
\defineSessionType{plenary}{Plenary}{orange!30}

% Redefine schedule colours if desired (before or after \usepackage).
\colorlet{clrScheduleOdd}{blue!5}

\begin{document}

% Ingest all session directories under sessions/.
% Must run under LuaLaTeX; use \directlua or a Lua file loaded at startup.
\setDefaultSessionType{contrib}
\directlua{
  local proc = require("process-submissions")
  proc.process_root("sessions")
}

% Assign locations, and override colour/label for individual sessions.
% All three keys are optional; values with commas need braces.
\setSessionsData{plenary-1}{location={Auditorium}}
\setSessionsData{contrib-1, contrib-2}{
  location = {Room~A},
  color    = {yellow!25},
  label    = {Special~Session}
}

% A single standalone talk (e.g.\ a plenary with its own .tex file):
\addSessionTalk[plenary]{plenary-1}{Prof.\ Keynote}{plenary.tex}
```

```
\chapter{Programme}

\begin{schedule}{2}{Monday Morning}
  \scheduleEvent{gray!20}{09:00}{Registration and Coffee}
  \scheduleSessionTalk{10:00}{plenary-1}        % full-width row, talk 1
  \scheduleEvent{gray!20}{11:00}{Coffee Break}
  \scheduleSessions{11:30}                      % concurrent sessions
    {Talk 1, Talk 2, Talk 3}                    % per-row time labels
    {contrib-1, contrib-2}                      % session ids (columns)
\end{schedule}

\chapter{Session Descriptions}
\renderSessions{contrib,invited}

\chapter{Abstracts}
\renderTalks*{contrib,invited}   % starred: include chair line per talk

\chapter{Participants}
\printParticipants

\end{document}
```

2 User Interface

2.1 Session types and configuration

\defineSessionType `\defineSessionType {⟨key⟩} {⟨label⟩} {⟨colour⟩}`

Registers a session type. ⟨*key*⟩ is the internal identifier used in all other commands. ⟨*label*⟩ is the display string shown in schedule cells. ⟨*colour*⟩ is any xcolor colour expression used as the cell background. The colour and label are copied into each session's own property record when the session is opened (via the Lua processor or `\openSession`), so `\defineSessionType` must be called before sessions are loaded. Individual sessions may override these values afterwards with `\setSessionsData`.

\setDefaultSessionType `\setDefaultSessionType {⟨key⟩}`

Sets the type stamped onto sessions opened subsequently. The initial default is `UNKNOWN`. Must be called before `\openSession`; changing the type while a session is open has no effect on that session.

`\setSessionsData` `\setSessionsData {⟨id, id, ...⟩} {⟨key=value, ...⟩}`

Sets one or more properties on each session in the comma-list. All three keys are optional; omit any that should remain unchanged.

Key	Aliases
`location`	`loc`
`color`	`colour`, `clr`
`label`	`lbl`

Values containing commas must be wrapped in braces, e.g. `location={Room A, Level 2}`. Issues a warning for each unknown session id. Must be called after session files have been loaded. Unknown keys are an error.

`\sessionLocation` `\sessionLocation {⟨id⟩}`
`\sessionSlot` `\sessionSlot {⟨id⟩}`
`\talkSlot` `\talkSlot {⟨id@n⟩}`

Query commands that expand to the stored location, schedule slot label, or talk slot label respectively. Error if the id is unknown or has not been scheduled.

`\sessionCount` `\sessionCount {⟨type, type, ...⟩}`
`\talkCount` `\talkCount {⟨type, type, ...⟩}`
`\participantCount` `\participantCount`

Typeset the count of scheduled sessions or non-cancelled scheduled talks whose type appears in the comma-list, or the number of participants in the database who have not been removed via `\removeParticipant`. All three are valid anywhere after the submission files have been processed.

`\setInputFilepath` `\setInputFilepath {⟨path⟩}`

Records the path of the file about to be `\input`. Called automatically by the Lua processor before each file; not normally needed in document files.

2.2 Session and talk lifecycle

`\openSession` `\openSession {⟨id⟩} {⟨path⟩}`
`\closeSession` `\closeSession`

Open and close a session accumulator. Called automatically by the Lua processor; use directly only if driving the pipeline from TEX rather than Lua.

`session` (*env.*)

```
\begin{session}
{⟨title⟩} {⟨chair⟩} {⟨org1-name⟩} {⟨org1-affil⟩} {⟨org1-email⟩}
{⟨org2-name⟩} {⟨org2-affil⟩} {⟨org2-email⟩}
[⟨extra-setup⟩]
⟨body⟩
\end{session}
```

Declares the session metadata. Omit the second organiser by passing three empty braces. ⟨*extra-setup*⟩ (optional argument) is executed in the session context, e.g. `\addSessionOrganizer`. The ⟨*body*⟩ is the session abstract or description. It is retypeset verbatim by `\renderSessions` and may contain arbitrary prose. Leave it empty if no description is provided.

talk (*env.*)

```
\begin{talk}
{⟨title⟩} {⟨speaker⟩} {⟨affil⟩} {⟨email⟩}
{⟨coauthors⟩} {⟨special-note⟩}
[⟨hook⟩]
⟨abstract⟩
\end{talk}
```

Declares one talk and its abstract. ⟨*hook*⟩ is executed after all fields are stored but before the talk is committed; use it for `\cancelTalk`. The ⟨*abstract*⟩ is the talk abstract. It is re-typeset verbatim by `\renderTalks`.

`extrainfo` (*env.*) Content inside this environment is silently discarded. Useful in submission files for author-side notes or metadata that should not appear in the proceedings.

`\addSessionOrganizer` `\addSessionOrganizer {⟨name⟩} {⟨affil⟩} {⟨email⟩}`

Adds an organiser to the current session (beyond the two in the `session` environment).

`\setSessionChair` `\setSessionChair {⟨name⟩}`

Overrides the chair of the current session.

`\addSessionTalk` `\addSessionTalk [⟨type⟩] {⟨id⟩} {⟨chair⟩} {⟨filepath⟩}`

Convenience wrapper that opens a one-talk session of the given ⟨*type*⟩ (default `plenary`), inputs ⟨*filepath*⟩, then closes the session.

`\cancelTalk` Marks the enclosing talk as cancelled. Call inside the optional ⟨*hook*⟩ argument of the `talk` environment. Cancelled talks appear with `\cancelledTalkFormat` in the schedule and session listing, and are omitted from `\renderTalks`. The speaker remains a participant; use `\removeParticipant` to suppress them from the printed participant list.

`\cancelledTalkFormat` `\cancelledTalkFormat {⟨text⟩}`

Applied to cancelled speaker/title text in the schedule and session listing. Default: `\sout{⟨text⟩}` (strikethrough via ulem). Override with `\renewcommand`.

`\missingTalk` `\missingTalk {⟨index⟩}`

Inserts a blank talk placeholder for ⟨*index*⟩. Emitted automatically by the Lua processor when a numbered file is absent from a session directory.

2.3 Participant management

`\addParticipant` `\addParticipant {⟨name⟩} {⟨email⟩} {⟨affil⟩}`

Registers a participant who is not a speaker or organiser. Appears in `\printParticipants` with empty session/talk lists.

`\removeParticipant` `\removeParticipant {⟨email⟩}`

Marks a participant as removed, identified by their ⟨*email*⟩ address. Removed participants are silently skipped by `\printParticipants` but remain in the database, so their names still appear correctly in session and talk listings. Issues a warning if the address is not found in the database. No-op for a blank address.

2.4 Schedule construction

All schedule commands are valid only inside the `schedule` environment.

`schedule` (*env.*)

```
\begin{schedule} {⟨num-cols⟩} {⟨heading⟩}
⟨body⟩
\end{schedule}
```

Typesets a tabularx schedule grid with ⟨*num-cols*⟩ session columns and a header row showing ⟨*heading*⟩. The ⟨*body*⟩ contains `\scheduleSessionTalk`, `\scheduleSessions`, and `\scheduleEvent` calls.

`\scheduleSessionTalk` `\scheduleSessionTalk [⟨n⟩] {⟨slot⟩} {⟨id⟩}`

Adds a full-width row for the ⟨*n*⟩-th talk (default 1) of session ⟨*id*⟩ at time ⟨*slot*⟩. Shows location, type label, speaker, title (with page reference), and chair. Cancelled talks appear with strikethrough and no page reference.

`\scheduleSessions`

```
\scheduleSessions {⟨slot-title⟩}
{⟨row-label, row-label, ...⟩}
{⟨id, id, ...⟩}
```

Adds a concurrent-session block. ⟨*row-labels*⟩ (comma-list) are used as time labels for each talk row; ⟨*ids*⟩ (comma-list) name the session columns. The header row uses per-column type colours; talk rows alternate `clrScheduleOdd`/`clrScheduleEven`. First placement of any session or talk wins; duplicates are silently ignored.

`\scheduleEvent` `\scheduleEvent {⟨colour⟩} {⟨slot⟩} {⟨content⟩}`

Adds a full-width coloured row for a non-talk event such as a coffee break, meal, or ceremony.

2.5 Output

`\renderSessions` `\renderSessions {⟨type, type, ...⟩}`

Outputs in schedule order all sessions whose type is in the comma-list. Each session entry contains: a coloured slot/location bar, a `\subsection*` title with a `\label`, the organiser list with affiliations and email addresses, the session description body (re-input from the submission file), and a talk list with fill-to-right page references.

`\renderTalks`

```
\renderTalks  {⟨type, type, ...⟩}
\renderTalks* {⟨type, type, ...⟩}
```

Renders talk abstracts in schedule order, filtered by type. Each entry contains: a coloured slot/location bar, an optional session title back-reference, an optional chair line (starred form only), a `\subsection*` title with `\label`, speaker name, affiliation, email, coauthors, and the abstract body. Cancelled talks are silently skipped.

\printParticipants `\printParticipants [⟨min⟩]`
`\printParticipants* [⟨min⟩]`

Typesets an alphabetically sorted two-column list of participants. Each entry shows name, affiliation, email, and page references to their sessions and talks. The optional ⟨*min*⟩ argument (default 0) suppresses participants with fewer than ⟨*min*⟩ combined session and talk engagements. The starred form additionally prints each participant's scheduled slots below the page references, one per line. Each slot line shows a role label ("Organizer:" or "Speaker:"), the session title in quotes, an inline page reference, and the slot string; for example:

> *Speaker:* "Numerical Optimisation", p. 3, Monday Morning, 10:00

Issues a warning for any slot that cannot be resolved. Removed participants are always omitted.

2.6 Page references

\blockPageref `\blockPageref {⟨label⟩}`
\fillPageref `\fillPageref {⟨label⟩}`

Both expand to "p. ⟨*N*⟩". `\blockPageref` is inline; `\fillPageref` adds a `\hfill` so the reference floats to the right margin.

2.7 Colour customisation

The following colours are defined by the package with `\@ifundefined` guards, so they may be pre-defined in the document class or redefined with `\definecolor`/`\colorlet` after `\usepackage`:

Name	Default	Used in
`clrScheduleEmpty`	gray 0.85	time cell in `\scheduleSessions` header
`clrScheduleOdd`	gray 0.93	odd talk rows in `\scheduleSessions`
`clrScheduleEven`	white	even talk rows in `\scheduleSessions`

Session type colours are set per type via `\defineSessionType`; each cell uses `\cellcolor` for the type colour and `\rowcolor` for the alternating row colour.

3 Lua Submission Processor

`process-submissions.lua` is a LuaLaTeX module that scans a directory tree and drives the TEX pipeline. Each session occupies a subdirectory; each file in that directory contains exactly one `session` or `talk` environment. Content outside an environment in a file is ignored. Load the module with `\directlua` or from a Lua startup file:

```
local proc = require("process-submissions")

-- Process all session subdirectories under "sessions/":
proc.process_root("sessions")

-- Or process a single session directory directly:
proc.process("sessions/contrib-1")
```

process_root `proc.process_root(⟨dir⟩)`

Iterates over every immediate subdirectory of ⟨*dir*⟩ (skipping hidden entries). For each subdirectory it emits `\openSession{`⟨*name*⟩`}{`⟨*path*⟩`}`, calls `process`, then emits `\closeSession`. The subdirectory name becomes the session id.

process `proc.process(⟨dir⟩)`

Collects all `.tex` files in ⟨*dir*⟩, sorts them alphabetically, then inputs each in turn via `\setInputFilepath` and `\input`. Each file is expected to contain exactly one `session` or `talk` environment; content outside any environment is ignored. Files whose name begins with digits have the leading number extracted as the talk index (`\currentTalkIndex`); gaps in the sequence trigger automatic `\missingTalk` calls. Files without a leading digit are input normally but do not advance the talk index counter.

4 Implementation

4.1 Package file (`confschedule.sty`)

4.1.1 Package infrastructure

Required packages and default colour definitions. xcolor must receive the `table` option before being loaded to avoid option-clash warnings; `\PassOptionsToPackage` ensures this even if xcolor was loaded earlier. The three `clrSchedule*` colours use `\@ifundefined` guards so that pre-defined values are respected.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{confschedule}
  [2025/05/16 v1.0 Conference schedule and proceedings management]

\PassOptionsToPackage{table}{xcolor}
\RequirePackage{expl3}
\RequirePackage{xparse}
\RequirePackage{xcolor}
\RequirePackage{tabularx}
\RequirePackage{multicol}
\RequirePackage[normalem]{ulem}

\@ifundefined{color@clrScheduleEmpty}
  { \definecolor{clrScheduleEmpty}{gray}{0.85} }{}
\@ifundefined{color@clrScheduleOdd}
  { \definecolor{clrScheduleOdd}{gray}{0.93} }{}
\@ifundefined{color@clrScheduleEven}
  { \colorlet{clrScheduleEven}{white} }{}
```

4.1.2 Declarations

All expl3 code runs inside `\ExplSyntaxOn`. The three `\def`s give safe defaults for the values that the Lua processor overwrites before each `\input`. The `\nolinkurl` fallback covers documents that load neither url nor hyperref.

```
\ExplSyntaxOn

\def\currentSessionType{UNKNOWN}
```

```
\def\currentFilepath{}
\def\currentTalkIndex{}

\cs_if_exist:NF \nolinkurl
  { \cs_new:Npn \nolinkurl #1 { \texttt{#1} } }

% --- Core session/talk storage ---
\prop_new:N \g_sessions_prop
\prop_new:N \g_session_prop
\prop_new:N \l_sched_talk_prop
\prop_new:N \l_sched_session_prop
\seq_new:N  \g_session_organizers_seq
\seq_new:N  \g_session_talks_seq
\seq_new:N  \g_orphan_talks_seq
\tl_new:N   \g_current_session_id_tl

% --- Schedule state ---
\seq_new:N  \g_schedule_sessions_seq
\seq_new:N  \g_schedule_talk_order_seq
\prop_new:N \g_talk_titles_prop
\prop_new:N \g_sched_session_slots_prop
\prop_new:N \g_sched_talk_slots_prop
\prop_new:N \g_session_type_colors_prop
\prop_new:N \g_session_type_labels_prop

% --- Schedule table row-building state ---
\tl_new:N   \l_sched_header_tl
\tl_new:N   \l_sched_numcols_tl
\seq_new:N  \l_sched_rows_out_seq
\tl_new:N   \l_sched_row_out_tl
\bool_new:N \l_sched_row_odd_bool
\int_new:N  \l_sched_talk_idx_int
\clist_new:N \l_sched_ids_clist
\clist_new:N \l_sched_labels_clist
\prop_new:N \l_sched_record_prop

% --- Named temporaries for resolved schedule values ---
\tl_new:N   \l_sched_color_tl
\tl_new:N   \l_sched_type_label_tl
\tl_new:N   \l_sched_session_id_tl
\tl_new:N   \l_sched_talks_tl
\tl_new:N   \l_sched_title_tl
\tl_new:N   \l_sched_speaker_tl
\tl_new:N   \l_sched_chair_tl
\tl_new:N   \l_sched_location_tl

% --- Participant database ---
\prop_new:N \g_participant_info_prop
\prop_new:N \g_participant_sessions_prop
\prop_new:N \g_participant_talks_prop
\prop_new:N \l_sched_participant_prop

% --- Scratch registers (single context, no nesting) ---
\tl_new:N   \l_sched_tmpa_tl
```

```
\tl_new:N   \l_sched_tmpb_tl
\tl_new:N   \l_sched_tmpc_tl
\bool_new:N \l_sched_tmpa_bool
\seq_new:N  \l_sched_tmpa_seq
\int_new:N  \l_sched_tmpa_int

% --- Cancelled-talk formatting registers ---
\tl_new:N   \l_sched_fmt_speaker_tl
\tl_new:N   \l_sched_fmt_title_tl

% --- Context flags ---
% \g_sched_in_talk_hook_bool: true only while the optional hook argument of
% the talk environment is being executed; guards \cancelTalk.
\bool_new:N \g_sched_in_talk_hook_bool

% --- Participant key and transient display-name map ---
% \g_participant_display_prop is built immediately before sorting in
% \sched_print_participants:.  It must be a flat prop because \seq_sort:Nn
% comparators must be purely expandable, and \prop_item:Nn only gives
% expandable single-level lookup.
\tl_new:N   \l_sched_participant_key_tl
\prop_new:N \g_participant_display_prop
\seq_new:N  \l_sched_sorted_participants_seq

% --- Render mode flag ---
\bool_new:N \g_sched_render_body_bool
```

4.1.3 Messages

```
\msg_new:nnn { session } { no-open-session }
  { \session_end:~called~but~no~session~is~currently~open. }

\msg_new:nnn { session } { participant-not-found }
  { Participant~key~'#1'~not~found~in~the~database. \\
    The~name~has~been~left~as~the~raw~key. }

\msg_new:nnn { schedule } { unknown-session }
  { Unknown~session~'#1'. \\
    Check~the~ID~and~that~\openSession{#1}~was~
    processed~before~\begin{document}. }

\msg_new:nnn { schedule } { render-missing-session }
  { Session~'#1'~is~in~the~schedule~but~not~found~in~the~database. }

\msg_new:nnn { schedule } { unknown-location }
  { No~location~set~for~session~'#1'. \\
    Use~\setSessionsData{#1}{location={room}}~after~loading~sessions. }

\msg_new:nnn { schedule } { session-not-scheduled }
  { Session~'#1'~has~not~been~placed~in~any~schedule. }

\msg_new:nnn { schedule } { talk-not-scheduled }
  { Talk~'#1'~not~found~in~the~schedule. }
```

```
\msg_new:nnn { schedule } { no-sessions-of-type }
  { No~sessions~of~type(s)~'#1'~found~in~the~schedule. \\
    Check~that~the~type~key(s)~match~those~passed~to~\defineSessionType. }

\msg_new:nnn { schedule } { session-not-found-for-assign }
  { Session~'#1'~not~found~in~the~database. \\
    Ensure~sessions~are~loaded~before~calling~\setSessionsData. }

\msg_new:nnn { schedule } { unknown-session-data-key }
  { Unknown~key~'#1'~passed~to~\setSessionsData. \\
    Valid~keys:~location~(loc),~color~(colour,~clr),~label~(lbl). }

\msg_new:nnn { session } { empty-participant-name }
  { A~talk,~organiser,~or~\addParticipant~entry~supplied~an~empty~name.~
    The~entry~has~been~ignored. }

\msg_new:nnn { session } { cancel-outside-hook }
  { \cancelTalk~may~only~appear~in~the~optional~hook~argument~of~
    the~talk~environment~(before~the~abstract~body). }

\msg_new:nnn { session } { organizer-outside-session }
  { \addSessionOrganizer~may~only~appear~while~a~session~is~open,~
    i.e.~inside~the~session~environment's~optional~setup~argument. }

\msg_new:nnn { session } { chair-outside-session }
  { \setSessionChair~may~only~appear~while~a~session~is~open,~
    i.e.~inside~the~session~environment's~optional~setup~argument. }

\msg_new:nnn { session } { missing-talk-outside-session }
  { \missingTalk~may~only~appear~inside~a~session~context.~
    It~is~normally~called~automatically~by~the~Lua~processor. }

\msg_new:nnn { schedule } { slot-not-found }
  { No~scheduled~slot~found~for~'#1'. \\
    Check~that~\renderSessions~or~\renderTalks~has~been~called~
    and~that~the~item~was~placed~in~a~schedule. }
```

4.1.4 Cancellation commands

`\cancelTalk` Writes the `cancelled` field into `\l_sched_talk_prop` while it is live (called from the talk hook argument, inside `\talk_store:nnnnnnnn`).

(*End of definition for* `\cancelTalk`. *This function is documented on page* 6.)

`\removeParticipant` Sets a `removed` flag on the participant record so that `\printParticipants` silently skips them. The participant remains in the database, so their page references in session and talk listings are unaffected. Issues a warning if the key is not found.

(*End of definition for* `\removeParticipant`. *This function is documented on page* 6.)

`\cancelledTalkFormat` User-overridable strikethrough hook; default uses `\sout`.

(*End of definition for* `\cancelledTalkFormat`. *This function is documented on page* 6.)

```
\NewDocumentCommand{\cancelTalk}{}
{
```

```
  \bool_if:NF \g_sched_in_talk_hook_bool
    { \msg_error:nn { session } { cancel-outside-hook } }
  \prop_put:Nnn \l_sched_talk_prop { cancelled } { true }
}

\NewDocumentCommand{\removeParticipant}{m}
{
  \tl_if_blank:nF {#1}
  {
    \tl_set:Ne \l_sched_tmpa_tl { \sched_key_normalize:n {#1} }
    \prop_get:NVN \g_participant_info_prop \l_sched_tmpa_tl \l_sched_participant_prop
    \tl_if_eq:NNT \l_sched_participant_prop \q_no_value
      { \msg_warning:nnV { session } { participant-not-found } \l_sched_tmpa_tl }
    \tl_if_eq:NNF \l_sched_participant_prop \q_no_value
    {
      \prop_put:Nnn \l_sched_participant_prop { removed } { true }
      \prop_gput:NVV \g_participant_info_prop \l_sched_tmpa_tl \l_sched_participant_prop
    }
  }
}

\NewDocumentCommand{\cancelledTalkFormat}{m}{\sout{#1}}
```

4.1.5 General helpers

`\prop_get_ne:NnN` is like `\prop_get:NnN` but stores an empty token list (not `\q_no_value`) on a miss, avoiding repetitive sentinel checks at call sites.

`\sched_session_color:NNN` resolves a session prop to its background colour and display label, falling back to `white` and the raw type key when the type has not been registered.

```
\cs_generate_variant:Nn \prop_if_in:NnTF { NV }
\cs_generate_variant:Nn \str_compare:nNnTF { eNe }
\cs_generate_variant:Nn \str_casefold:n { e }
\cs_generate_variant:Nn \str_lowercase:n { e }

\cs_new_protected:Npn \prop_get_ne:NnN #1 #2 #3
{
  \prop_get:NnN #1 {#2} #3
  \tl_if_eq:NNT #3 \q_no_value { \tl_clear:N #3 }
}
\cs_generate_variant:Nn \prop_get_ne:NnN { NVN }

% Normalize a participant key: strip leading/trailing spaces and lowercase.
% Expandable so it may be used inside e-type arguments.
\cs_new:Npn \sched_key_normalize:n #1
  { \str_lowercase:e { \tl_trim_spaces:n {#1} } }
\cs_generate_variant:Nn \sched_key_normalize:n { e, V }

% Look up the original name string for a participant key.
% Leading and trailing spaces are stripped from the key before the lookup.
% Falls back to the trimmed key itself when the participant is absent from the database.
\cs_new_protected:Npn \sched_participant_name:nN #1 #2
{
```

```
  \tl_set:Ne \l_sched_tmpa_tl { \sched_key_normalize:n {#1} }
  \prop_get_ne:NVN \g_participant_info_prop \l_sched_tmpa_tl \l_sched_participant_prop
  \tl_if_empty:NTF \l_sched_participant_prop
    {
      \msg_warning:nnn { session } { participant-not-found } {#1}
      \tl_set_eq:NN #2 \l_sched_tmpa_tl
    }
    {
      \prop_get_ne:NnN \l_sched_participant_prop { name } #2
      \tl_if_empty:NT #2 { \tl_set_eq:NN #2 \l_sched_tmpa_tl }
    }
}
\cs_generate_variant:Nn \sched_participant_name:nN { eN }

\cs_new_protected:Npn \sched_colored_subsection:nnn #1 #2 #3
{
  \subsection*{%
    {\normalfont\small
    \colorbox{#1}%
      { \parbox{\dimexpr\hsize - 2\fboxsep\relax}{\hfill#2} }}%
    \newline
    #3%
  }%
}
\cs_generate_variant:Nn \sched_colored_subsection:nnn { Vnn }

% \cs{sched\_colored\_subsection:nnn} typesets a \cs{subsection*} whose title
% argument begins with a full-width coloured slot/location bar (built from
% arguments~|#1| and~|#2|) followed by a \cs{newline} and the actual title
% text~|#3|.  Embedding both in a single \cs{subsection*} call prevents any
% page break or extra vertical space between the bar and the title.
%
% fields that were written into the session prop at creation time (by
% \cs{sched\_session\_copy\_type\_defaults:} called from \cs{openSession}
% and \cs{setDefaultSessionType}).  Falls back to \texttt{white} and the
% raw type key so that sessions with an unregistered type still render.
\cs_new_protected:Npn \sched_session_color:NNN #1 #2 #3
{
  \prop_get_ne:NnN #1 { color } #2
  \tl_if_empty:NT #2 { \tl_set:Nn #2 { white } }
  \prop_get_ne:NnN #1 { label } #3
  \tl_if_empty:NT #3
  {
    \prop_get_ne:NnN #1 { type } #3
    \tl_if_empty:NT #3 { \tl_set:Nn #3 { UNKNOWN } }
  }
}

\prg_new_protected_conditional:Npnn \sched_type_matches:nn #1#2 { T, F, TF }
{
  \clist_if_in:nnTF {#2} {#1}
    { \prg_return_true: }
    { \prg_return_false: }
}
```

```
\cs_generate_variant:Nn \sched_type_matches:nnTF { Vn }
\cs_generate_variant:Nn \sched_type_matches:nnT  { Vn  }
```

4.1.6 Schedule helpers

`\sched_get_session:nNTF` retrieves a session prop by id, issuing an error and taking the false branch when the id is unknown.

`\sched_record_talk:nnn` appends a talk record to the global schedule order sequence; first placement wins and duplicates are silently ignored.

`\sched_maybe_cancel:NnN` is the single site for the cancelled/normal formatting decision. It stores ⟨*content*⟩ in ⟨*output-tl*⟩ as-is, or wrapped in `\cancelledTalkFormat`, depending on the cancelled flag tl.

```
\prg_new_protected_conditional:Npnn \sched_get_session:nN #1 #2 { T, F, TF }
{
  \prop_get:NnN \g_sessions_prop {#1} #2
  \tl_if_eq:NNTF #2 \q_no_value
    { \msg_error:nnn { schedule } { unknown-session } {#1}
      \prg_return_false: }
    { \prg_return_true: }
}

\cs_new_protected:Npn \sched_get_nth_talk:NnN #1 #2 #3
{
  \prop_clear:N #3
  \seq_map_indexed_inline:Nn #1
  {
    \int_compare:nNnT { ##1 } = { #2 }
      { \tl_set:Nn #3 { ##2 } \seq_map_break: }
  }
}
\cs_generate_variant:Nn \sched_get_nth_talk:NnN { NVN }

\cs_new_protected:Npn \sched_record_talk:nnn #1 #2 #3
{
  \prop_if_in:NnF \g_sched_talk_slots_prop {#1@#2}
  {
    \prop_gput:Nnn \g_sched_talk_slots_prop {#1@#2} {#3}
    \prop_clear:N \l_sched_record_prop
    \prop_put:Nnn \l_sched_record_prop { session } {#1}
    \prop_put:Nnn \l_sched_record_prop { index   } {#2}
    \prop_put:Nnn \l_sched_record_prop { slot    } {#3}
    \seq_gput_right:NV \g_schedule_talk_order_seq \l_sched_record_prop
  }
}
\cs_generate_variant:Nn \sched_record_talk:nnn { nne, nee }

\cs_new_protected:Npn \sched_maybe_cancel:NnN #1 #2 #3
{
  \tl_if_empty:NTF #1
    { \tl_set:Nn #3 {#2} }
    { \tl_set:Nn #3 { \cancelledTalkFormat{#2} } }
}
\cs_generate_variant:Nn \sched_maybe_cancel:NnN { NeN }
```

4.1.7 Participant database

Participant names are normalised to a canonical key of the form "Lastname, Firstname" by `\participant_name_to_key:nN`, which accepts both "First Last" (space-separated, last word is the last name) and "Last, First" (comma-separated) input. Tildes in space-form names are treated as spaces.

`\participant_ensure:nnn` is idempotent: on a repeat encounter it back-fills blank email or affiliation but does not overwrite existing data.

```
\cs_new_protected:Npn \sched_append_to_prop_clist:Nnn #1 #2 #3
{
  \prop_get_ne:NnN #1 {#2} \l_sched_tmpa_tl
  \tl_if_empty:NTF \l_sched_tmpa_tl
    { \tl_set:Nn  \l_sched_tmpa_tl {#3} }
    { \tl_put_right:Nn \l_sched_tmpa_tl { ,#3 } }
  \prop_gput:NnV #1 {#2} \l_sched_tmpa_tl
}
\cs_generate_variant:Nn \sched_append_to_prop_clist:Nnn { NnV, NVn, NVV }

% Like \sched_append_to_prop_clist:Nnn but silently skips if the value is
% already present, preventing double-counting when render commands are called
% more than once with overlapping type lists.
\cs_new_protected:Npn \sched_append_unique_to_prop_clist:Nnn #1 #2 #3
{
  \prop_get_ne:NnN #1 {#2} \l_sched_tmpa_tl
  \tl_if_empty:NTF \l_sched_tmpa_tl
    { \prop_gput:Nnn #1 {#2} {#3} }
    {
      \clist_if_in:NnF \l_sched_tmpa_tl {#3}
      {
        \tl_put_right:Nn \l_sched_tmpa_tl { ,#3 }
        \prop_gput:NnV #1 {#2} \l_sched_tmpa_tl
      }
    }
}
\cs_generate_variant:Nn \sched_append_unique_to_prop_clist:Nnn { NnV, NVn, NVV }

\cs_new_protected:Npn \participant_split_name:nNN #1 #2 #3
{
  \tl_if_in:nnTF {#1} {,}
  {
    \seq_set_split:Nnn \l_sched_tmpa_seq {,} {#1}
    \tl_set:Ne #2 { \seq_item:Nn \l_sched_tmpa_seq {1} }
    \tl_trim_spaces:N #2
    \tl_set:Ne #3 { \seq_item:Nn \l_sched_tmpa_seq {2} }
    \tl_trim_spaces:N #3
  }
  {
    \tl_set:Nn \l_sched_tmpb_tl {#1}
    \regex_replace_all:nnN { \~ } { \x{20} } \l_sched_tmpb_tl
    \seq_set_split:NnV \l_sched_tmpa_seq { ~ } \l_sched_tmpb_tl
    \seq_remove_all:Nn \l_sched_tmpa_seq {}
    \tl_set:Ne #2 { \seq_item:Nn \l_sched_tmpa_seq { -1 } }
    \tl_clear:N #3
    \seq_map_indexed_inline:Nn \l_sched_tmpa_seq
```

```
    {
      \int_compare:nNnT {##1} < { \seq_count:N \l_sched_tmpa_seq }
      {
        \tl_if_empty:NTF #3
          { \tl_set:Nn  #3 {##2} }
          { \tl_put_right:Nn #3 { ~##2 } }
      }
    }
  }
}

\cs_new_protected:Npn \participant_name_to_key:nN #1 #2
{
  \participant_split_name:nNN {#1} \l_sched_tmpb_tl \l_sched_tmpc_tl
  \tl_if_empty:NTF \l_sched_tmpc_tl
    { \tl_set_eq:NN #2 \l_sched_tmpb_tl }
    { \tl_set:Ne #2
        { \tl_use:N \l_sched_tmpb_tl ,~ \tl_use:N \l_sched_tmpc_tl } }
}

\cs_new_protected:Npn \participant_ensure:nnn #1 #2 #3
{
  \tl_if_blank:nF {#1}
  {
    \participant_name_to_key:nN {#1} \l_sched_tmpa_tl
    \tl_if_blank:nTF {#2}
      { \tl_set_eq:NN \l_sched_participant_key_tl \l_sched_tmpa_tl }
      { \tl_set:Ne    \l_sched_participant_key_tl { \sched_key_normalize:n {#2} } }
    \prop_if_in:NVTF \g_participant_info_prop \l_sched_participant_key_tl
    {
      \prop_get:NVN \g_participant_info_prop \l_sched_participant_key_tl \l_sched_participan
      \prop_get_ne:NnN \l_sched_participant_prop { email } \l_sched_tmpb_tl
      \tl_if_empty:NT \l_sched_tmpb_tl
        { \prop_put:Nnn \l_sched_participant_prop { email } {#2} }
      \prop_get_ne:NnN \l_sched_participant_prop { affil } \l_sched_tmpb_tl
      \tl_if_empty:NT \l_sched_tmpb_tl
        { \prop_put:Nnn \l_sched_participant_prop { affil } {#3} }
      \prop_gput:NVV \g_participant_info_prop \l_sched_participant_key_tl \l_sched_participa
    }
    {
      % New participant: a single prop_gput registers the key implicitly
      \prop_clear:N \l_sched_participant_prop
      \prop_put:Nnn \l_sched_participant_prop { name    } {#1}
      \prop_put:NnV \l_sched_participant_prop { display } \l_sched_tmpa_tl
      \prop_put:Nnn \l_sched_participant_prop { email   } {#2}
      \prop_put:Nnn \l_sched_participant_prop { affil   } {#3}
      \prop_gput:NVV \g_participant_info_prop \l_sched_participant_key_tl \l_sched_participa
    }
  }
}
```

4.1.8 Talk and session storage

`\talk_store:nnnnnnnn` builds `\l_sched_talk_prop`, executes the hook argument `#8` (where `\cancelTalk` may write the `cancelled` field), registers the speaker in the participant database via `\participant_ensure:nnn` and stores their key in the `speakerkey` field of the talk prop (rather than duplicating the speaker's data), then pushes the prop to the appropriate accumulator sequence.

`\session_end:` finalises the current session accumulator and stores it in the global sessions database under its id.

```
\cs_new_protected:Npn \talk_store:nnnnnnnn #1#2#3#4#5#6#7#8
{
  \prop_clear:N \l_sched_talk_prop
  \prop_put:Nnn \l_sched_talk_prop { title     } {#1}
  \prop_put:Nnn \l_sched_talk_prop { coauthors } {#5}
  \prop_put:Nnn \l_sched_talk_prop { special   } {#6}
  \tl_set:Ne    \l_sched_tmpa_tl   { \currentFilepath }
  \prop_put:NnV \l_sched_talk_prop { filepath  } \l_sched_tmpa_tl
  \bool_gset_true:N  \g_sched_in_talk_hook_bool
  #8
  \bool_gset_false:N \g_sched_in_talk_hook_bool
  \tl_if_empty:NF \g_current_session_id_tl
  {
    \participant_ensure:nnn {#2} {#4} {#3}
    \prop_put:NnV \l_sched_talk_prop { speakerkey } \l_sched_participant_key_tl
    \tl_set:Ne \l_sched_tmpa_tl
      { \g_current_session_id_tl @ \int_eval:n { \seq_count:N \g_session_talks_seq + 1 } }
    \prop_gput:NVn \g_talk_titles_prop \l_sched_tmpa_tl {#1}
  }
  \tl_if_empty:NTF \g_current_session_id_tl
    { \seq_gput_right:NV \g_orphan_talks_seq  \l_sched_talk_prop }
    { \seq_gput_right:NV \g_session_talks_seq \l_sched_talk_prop }
}

\cs_new_protected:Npn \session_end:
{
  \tl_if_empty:NTF \g_current_session_id_tl
    { \msg_warning:nn { session } { no-open-session } }
    {
      \prop_gput:NnV \g_session_prop { talks      } \g_session_talks_seq
      \prop_gput:NnV \g_session_prop { organizers } \g_session_organizers_seq
      \prop_gput:NVV \g_sessions_prop \g_current_session_id_tl \g_session_prop
      \tl_gclear:N   \g_current_session_id_tl
    }
}

\cs_new_protected:Npn \session_add_organizer:nnn #1#2#3
{
  \participant_ensure:nnn {#1} {#3} {#2}
  % \l_sched_participant_key_tl now holds the key; append it to the accumulator.
  \seq_gput_right:NV \g_session_organizers_seq \l_sched_participant_key_tl
}
% \cs{sched\_session\_copy\_type\_defaults:} copies the colour and label
% registered for the current session's type into \cs{g\_session\_prop}.
```

```
% Called from \cs{openSession} and \cs{setDefaultSessionType} so that the
% session prop always carries its own resolved colour and label.
% If the type has not been registered the fields are left blank and
% \cs{sched\_session\_color:NNN} will fall back to safe defaults at render
% time.
\cs_new_protected:Npn \sched_session_copy_type_defaults:
{
  \prop_get_ne:NnN \g_session_prop { type } \l_sched_tmpa_tl
  \prop_get_ne:NVN \g_session_type_colors_prop \l_sched_tmpa_tl \l_sched_tmpb_tl
  \tl_if_empty:NF \l_sched_tmpb_tl
    { \prop_gput:NnV \g_session_prop { color } \l_sched_tmpb_tl }
  \prop_get_ne:NVN \g_session_type_labels_prop \l_sched_tmpa_tl \l_sched_tmpb_tl
  \tl_if_empty:NF \l_sched_tmpb_tl
    { \prop_gput:NnV \g_session_prop { label } \l_sched_tmpb_tl }
}

```

4.1.9 User-facing commands

\defineSessionType
\setDefaultSessionType
\setSessionsData
\setInputFilepath
\openSession
\closeSession
session
talk
extrainfo
\addSessionOrganizer
\setSessionChair
\addSessionTalk
\addParticipant

(*End of definition for* `\defineSessionType` *and others. These functions are documented on page* 4.)

```
\NewDocumentCommand{\defineSessionType}{mmm}
{
  \prop_gput:Nnn \g_session_type_labels_prop {#1} {#2}
  \prop_gput:Nnn \g_session_type_colors_prop {#1} {#3}
}

\NewDocumentCommand{\setDefaultSessionType}{m}
{
  \def\currentSessionType{#1}
}

\NewDocumentCommand{\setInputFilepath}{m}
  { \def\currentFilepath{#1} }

% Key family for \setSessionsData: canonical keys plus aliases.
% The handlers write into \l_sched_session_prop, which \setSessionsData
% reads back from \g_sessions_prop before calling \keys_set:nn and
% writes back afterwards.
\keys_define:nn { confschedule / session-data }
{
  location .code:n = { \prop_put:Nnn \l_sched_session_prop { location } {#1} } ,
  loc      .meta:n = { location = {#1} } ,
  color    .code:n = { \prop_put:Nnn \l_sched_session_prop { color    } {#1} } ,
  colour   .meta:n = { color = {#1} } ,
  clr      .meta:n = { color = {#1} } ,
  label    .code:n = { \prop_put:Nnn \l_sched_session_prop { label    } {#1} } ,
  lbl      .meta:n = { label = {#1} } ,
  unknown  .code:n =
    { \msg_error:nnV { schedule } { unknown-session-data-key } \l_keys_key_tl } ,
}

\NewDocumentCommand{\setSessionsData}{mm}
```

```
{
  \clist_map_inline:nn {#1}
  {
    \prop_get:NnN \g_sessions_prop {##1} \l_sched_session_prop
    \tl_if_eq:NNT \l_sched_session_prop \q_no_value
      { \msg_warning:nnn { schedule } { session-not-found-for-assign } {##1} }
    \tl_if_eq:NNF \l_sched_session_prop \q_no_value
    {
      \keys_set:nn { confschedule / session-data } {#2}
      \prop_gput:NnV \g_sessions_prop {##1} \l_sched_session_prop
    }
  }
}

\NewDocumentCommand{\openSession}{mm}
{
  \tl_if_empty:NF \g_current_session_id_tl { \session_end: }
  \tl_gset:Nn  \g_current_session_id_tl {#1}
  \seq_gclear:N \g_session_organizers_seq
  \seq_gclear:N \g_session_talks_seq
  \prop_gclear:N \g_session_prop
  \prop_gput:NnV \g_session_prop { type  } \currentSessionType
  \sched_session_copy_type_defaults:
}

\NewDocumentCommand{\closeSession}{} { \session_end: }

\NewDocumentEnvironment{talk}{mmmmmmO{}+b}
  {}
  {
    \bool_if:NTF \g_sched_render_body_bool
      { #8 }
      { \talk_store:nnnnnnnn {#1}{#2}{#3}{#4}{#5}{#6}{}{#7} }
  }

⟨*new-session⟩
\NewDocumentEnvironment{session}{m m m m O{} +b}
{
  \bool_if:NTF \g_sched_render_body_bool
  { #6 }
  {
    \prop_gput:Nnn \g_session_prop { title } {#1}
    \tl_set:Ne    \l_sched_tmpa_tl { \currentFilepath }
    \prop_gput:NnV \g_session_prop { filepath } \l_sched_tmpa_tl
    \session_add_organizer:nnn {#2}{#3}{#4}
    \prop_gput:NnV \g_session_prop { chair } \l_sched_participant_key_tl
    #5
  }
}
{}
⟨/new-session⟩
⟨*old-session⟩
\NewDocumentEnvironment{session}{m m m m m m m O{} +b}
{
```

```
  \bool_if:NTF \g_sched_render_body_bool
  { #9 }
  {
    \prop_gput:Nnn \g_session_prop { title } {#1}
    \tl_set:Ne    \l_sched_tmpa_tl { \currentFilepath }
    \prop_gput:NnV \g_session_prop { filepath } \l_sched_tmpa_tl
    \session_add_organizer:nnn {#2}{#3}{#4}
    \prop_gput:NnV \g_session_prop { chair } \l_sched_participant_key_tl
    \tl_if_blank:nF {#6} { \session_add_organizer:nnn {#5}{#6}{#7} }
    #8
  }
}
{}
⟨/old-session⟩

\NewDocumentEnvironment{extrainfo}{+b} {}{}

\NewDocumentCommand{\addSessionOrganizer}{mmm}
{
  \tl_if_empty:NTF \g_current_session_id_tl
    { \msg_error:nn { session } { organizer-outside-session } }
    { \session_add_organizer:nnn {#1}{#2}{#3} }
}

\NewDocumentCommand{\setSessionChair}{m}
{
  \tl_if_empty:NTF \g_current_session_id_tl
    { \msg_error:nn { session } { chair-outside-session } }
    { \prop_gput:Nne \g_session_prop { chair } { \sched_key_normalize:n {#1} } }
}

⟨*new-session⟩
\NewDocumentCommand{\addSessionTalk}{O{plenary}mmm}
{
  \setDefaultSessionType{#1}
  \setInputFilepath{#4}
  \openSession{#2}{}
  \begin{session}{}{}{}{}[\setSessionChair{#3}]
  \end{session}
  \input{#4}
  \closeSession
}
⟨/new-session⟩
⟨*old-session⟩
\NewDocumentCommand{\addSessionTalk}{O{plenary}mmm}
{
  \setDefaultSessionType{#1}
  \setInputFilepath{#4}
  \openSession{#2}{}
  \begin{session}{}{}{}{}{}{}{}[\setSessionChair{#3}]
  \end{session}
  \input{#4}
  \closeSession
}
```

```
⟨/old-session⟩

\NewDocumentCommand{\addParticipant}{mmm}
{
  \tl_if_blank:nTF {#1}
    { \msg_warning:nnn { session } { empty-participant-name } {} }
    { \participant_ensure:nnn {#1} {#2} {#3} }
}
```

4.1.10 Schedule table construction

`\sched_session_talk:nnn` builds one full-width table row. All token-list content is deferred into `\l_sched_rows_out_seq` via `\seq_put_right:Ne` so that variable values are captured at construction time, not at typeset time. `\sched_maybe_cancel:NeN` applies the formatting decision; the inline `\tl_if_empty:NT` eagerly resolves whether to emit a page reference during the e-expansion.

```
\cs_new_protected:Npn \sched_session_talk:nnn #1 #2 #3
{
  \sched_get_session:nNTF {#2} \l_sched_session_prop
  {
    \sched_session_color:NNN \l_sched_session_prop \l_sched_color_tl \l_sched_type_label_tl
    \prop_get_ne:NnN \l_sched_session_prop { talks } \l_sched_talks_tl
    \sched_get_nth_talk:NnN \l_sched_talks_tl { #3 } \l_sched_talk_prop
    \prop_get_ne:NnN \l_sched_talk_prop { title } \l_sched_title_tl
    \prop_if_in:NnF \g_sched_session_slots_prop {#2}
      { \prop_gput:Nne \g_sched_session_slots_prop {#2} { \l_sched_header_tl,~#1 } }
    \prop_get_ne:NnN \l_sched_session_prop { location } \l_sched_location_tl
    \tl_if_empty:NTF \l_sched_title_tl
    {
      \seq_put_right:Ne \l_sched_rows_out_seq
      {
        \exp_not:N\rowcolor{\l_sched_color_tl} #1
        & \exp_not:N\multicolumn{\l_sched_numcols_tl}{F}{} \\ \exp_not:N\hline
      }
    }
    {
      \prop_get_ne:NnN \l_sched_talk_prop    { speakerkey } \l_sched_tmpa_tl
      \sched_participant_name:eN
        { \tl_use:N \l_sched_tmpa_tl } \l_sched_speaker_tl
      \sched_participant_name:eN
        { \prop_item:Nn \l_sched_session_prop { chair } } \l_sched_chair_tl
      \sched_record_talk:nne {#2} {#3} { \l_sched_header_tl,~#1 }
      \prop_get_ne:NnN \l_sched_talk_prop { cancelled } \l_sched_tmpa_tl
      \sched_maybe_cancel:NeN \l_sched_tmpa_tl
        { \exp_not:N\textbf{\exp_not:N\textit{ \exp_not:V \l_sched_speaker_tl }} }
        \l_sched_fmt_speaker_tl
      \sched_maybe_cancel:NeN \l_sched_tmpa_tl
        { \exp_not:N\textbf{ \exp_not:V \l_sched_title_tl } }
        \l_sched_fmt_title_tl
      \seq_put_right:Ne \l_sched_rows_out_seq
      {
        \exp_not:N\rowcolor{\l_sched_color_tl} #1
        & \exp_not:N\multicolumn{\l_sched_numcols_tl}{F}
```

```
        {
          \exp_not:V \l_sched_location_tl \exp_not:N\par
          \exp_not:N\textbf{ \exp_not:V \l_sched_type_label_tl }~(#2)
          \exp_not:N\vspace{1mm}\exp_not:N\par
          \exp_not:V \l_sched_fmt_speaker_tl \qquad
          ``\exp_not:V \l_sched_fmt_title_tl''
          \tl_if_empty:NT \l_sched_tmpa_tl
          { \exp_not:N\quad \exp_not:N\blockPageref{talk:#2@#3} }
          \exp_not:N\vspace{1mm}\exp_not:N\par
          \qquad Chair:~\exp_not:N\textit{ \exp_not:V \l_sched_chair_tl }
        } \\ \exp_not:N\hline
      }
    }
  }
  {
    \seq_put_right:Ne \l_sched_rows_out_seq
    {
      \exp_not:N\rowcolor{white} #1
      & \exp_not:N\multicolumn{\l_sched_numcols_tl}{F}{} \\ \exp_not:N\hline
    }
  }
}

\NewDocumentCommand{\scheduleSessionTalk}{O{1}mm}
{
  \seq_if_in:NnF \g_schedule_sessions_seq {#3}
    { \seq_gput_right:Nn \g_schedule_sessions_seq {#3} }
  \sched_session_talk:nnn {#2} {#3} {#1}
}

\cs_new_protected:Npn \sched_conc_sessions:nnn #1 #2 #3
{
  \clist_set:Nn \l_sched_ids_clist    {#3}
  \clist_set:Nn \l_sched_labels_clist {#2}

  \tl_set:Nn \l_sched_row_out_tl { \cellcolor{clrScheduleEmpty} #1 }

  \clist_map_inline:Nn \l_sched_ids_clist
  {
    \sched_get_session:nnNTF {##1} \l_sched_session_prop
    {
      \sched_session_color:NNN \l_sched_session_prop \l_sched_color_tl \l_sched_type_label_t
      \prop_if_in:NnF \g_sched_session_slots_prop {##1}
        { \prop_gput:Nne \g_sched_session_slots_prop {##1} { \l_sched_header_tl,~#1 } }
      \prop_get_ne:NnN \l_sched_session_prop { location } \l_sched_location_tl
      \prop_get_ne:NnN \l_sched_session_prop { title } \l_sched_title_tl
      \sched_participant_name:eN
        { \prop_item:Nn \l_sched_session_prop { chair } } \l_sched_chair_tl
      \tl_put_right:Ne \l_sched_row_out_tl
      {
        & \exp_not:N\cellcolor{\l_sched_color_tl}
        \exp_not:V \l_sched_location_tl \exp_not:N\par
        \exp_not:N\textbf{ \exp_not:V \l_sched_type_label_tl }~(##1) \exp_not:N\par
        \tl_if_empty:NF \l_sched_title_tl
```

```
      {\exp_not:V \l_sched_title_tl
        \quad \exp_not:N\blockPageref{session:##1} \exp_not:N\par}
      Chair:~\exp_not:N\textit{ \exp_not:V \l_sched_chair_tl }
    }
  }
  { \tl_put_right:Nn \l_sched_row_out_tl { & } }
}
\tl_put_right:Nn \l_sched_row_out_tl { \\ \hline }
\seq_put_right:NV \l_sched_rows_out_seq \l_sched_row_out_tl

\int_zero:N \l_sched_talk_idx_int
\bool_set_true:N \l_sched_row_odd_bool

\clist_map_inline:Nn \l_sched_labels_clist
{
  \int_incr:N \l_sched_talk_idx_int
  \bool_if:NTF \l_sched_row_odd_bool
    { \tl_set:Nn \l_sched_row_out_tl { \rowcolor{clrScheduleOdd}  ##1 }
      \bool_set_false:N \l_sched_row_odd_bool }
    { \tl_set:Nn \l_sched_row_out_tl { \rowcolor{clrScheduleEven} ##1 }
      \bool_set_true:N  \l_sched_row_odd_bool }

  \clist_map_inline:Nn \l_sched_ids_clist
  {
    \sched_get_session:nNTF {####1} \l_sched_session_prop
    {
      \prop_get_ne:NnN \l_sched_session_prop { talks } \l_sched_talks_tl
      \sched_get_nth_talk:NVN \l_sched_talks_tl \l_sched_talk_idx_int \l_sched_talk_prop
      \prop_get_ne:NnN \l_sched_talk_prop { title } \l_sched_title_tl
      \tl_if_empty:NTF \l_sched_title_tl
        { \tl_put_right:Nn \l_sched_row_out_tl { & } }
        {
          \prop_get_ne:NnN \l_sched_talk_prop { speakerkey } \l_sched_tmpa_tl
          \sched_participant_name:eN
            { \tl_use:N \l_sched_tmpa_tl } \l_sched_speaker_tl
          \prop_get_ne:NnN \l_sched_talk_prop { cancelled } \l_sched_tmpa_tl
          \sched_record_talk:nee {####1}
            { \int_use:N \l_sched_talk_idx_int }
            { \l_sched_header_tl,~##1 }
          \sched_maybe_cancel:NeN \l_sched_tmpa_tl
            { \exp_not:N\textit{ \exp_not:V \l_sched_speaker_tl } }
            \l_sched_fmt_speaker_tl
          \sched_maybe_cancel:NeN \l_sched_tmpa_tl
            { \exp_not:V \l_sched_title_tl }
            \l_sched_fmt_title_tl
          \tl_put_right:Ne \l_sched_row_out_tl
          {
            &
            \exp_not:V \l_sched_fmt_speaker_tl \exp_not:N\par
            \exp_not:V \l_sched_fmt_title_tl
            \tl_if_empty:NT \l_sched_tmpa_tl
              { \exp_not:N\quad \exp_not:N\blockPageref
                {talk:####1@\int_use:N \l_sched_talk_idx_int} }
          }
```

```
          }
      }
      { \tl_put_right:Nn \l_sched_row_out_tl { & } }
    }
    \tl_put_right:Nn \l_sched_row_out_tl { \\ \hline }
    \seq_put_right:NV \l_sched_rows_out_seq \l_sched_row_out_tl
  }
}
\cs_generate_variant:Nn \sched_conc_sessions:nnn { eee }

\NewDocumentCommand{\scheduleSessions}{mmm}
{
  \clist_map_inline:nn {#3}
  {
    \seq_if_in:NnF \g_schedule_sessions_seq {##1}
      { \seq_gput_right:Nn \g_schedule_sessions_seq {##1} }
  }
  \sched_conc_sessions:eee {#1} {#2} {#3}
}

\NewDocumentCommand{\scheduleEvent}{mmm}{
  \seq_put_right:Ne \l_sched_rows_out_seq
    { \exp_not:N\rowcolor{#1} #2
      & \exp_not:N\multicolumn{\l_sched_numcols_tl}{F}{#3} \\ }
}

\NewDocumentEnvironment{schedule}{mm+b}
{
  \tl_set:Nn \l_sched_header_tl  {#2}
  \tl_set:Nn \l_sched_numcols_tl {#1}
  \seq_clear:N \l_sched_rows_out_seq
  \arrayrulecolor{white}
  #3
  \small
  \begin{tabularx}{\linewidth}{l*{\l_sched_numcols_tl}{|H}}
    & \multicolumn{#1}{F}{\cellcolor{white}\large\textbf{#2}} \\
    \seq_use:Nn \l_sched_rows_out_seq {}
  \end{tabularx}
}{}
```

4.1.11 Query commands

The location and slot queries use `\l_sched_tmpa_tl` for the result and issue an error on a miss. The three counting commands use `\l_sched_tmpa_int` as a local accumulator; `\talkCount` additionally retrieves each talk prop to check the `cancelled` field, skipping cancelled talks.

```
\NewDocumentCommand{\sessionLocation}{m}
{
  \sched_get_session:nNTF {#1} \l_sched_session_prop
  {
    \prop_get_ne:NnN \l_sched_session_prop { location } \l_sched_tmpa_tl
    \tl_if_empty:NTF \l_sched_tmpa_tl
      { \msg_error:nnn { schedule } { unknown-location } {#1} }
```

```
      { \tl_use:N \l_sched_tmpa_tl }
  }
  { }
}

\NewDocumentCommand{\sessionSlot}{m}
{
  \prop_get:NnN \g_sched_session_slots_prop {#1} \l_sched_tmpa_tl
  \tl_if_eq:NNT \l_sched_tmpa_tl \q_no_value
    { \msg_error:nnn { schedule } { session-not-scheduled } {#1} }
  \tl_use:N \l_sched_tmpa_tl
}

\NewDocumentCommand{\talkSlot}{m}
{
  \prop_get:NnN \g_sched_talk_slots_prop {#1} \l_sched_tmpa_tl
  \tl_if_eq:NNT \l_sched_tmpa_tl \q_no_value
    { \msg_error:nnn { schedule } { talk-not-scheduled } {#1} }
  \tl_use:N \l_sched_tmpa_tl
}

% \sessionCount{types}: count scheduled sessions matching type filter.
\cs_new_protected:Npn \sched_count_sessions:n #1
{
  \int_zero:N \l_sched_tmpa_int
  \seq_map_inline:Nn \g_schedule_sessions_seq
  {
    \sched_get_session:nNT {##1} \l_sched_session_prop
    {
      \prop_get_ne:NnN \l_sched_session_prop { type } \l_sched_tmpb_tl
      \sched_type_matches:VnT \l_sched_tmpb_tl {#1}
        { \int_incr:N \l_sched_tmpa_int }
    }
  }
  \int_use:N \l_sched_tmpa_int
}

% \talkCount{types}: count scheduled non-cancelled talks matching type filter.
\cs_new_protected:Npn \sched_count_talks:n #1
{
  \int_zero:N \l_sched_tmpa_int
  \seq_map_inline:Nn \g_schedule_talk_order_seq
  {
    \tl_set:Nn \l_sched_record_prop {##1}
    \prop_get_ne:NnN \l_sched_record_prop { session } \l_sched_session_id_tl
    \prop_get:NVN \g_sessions_prop \l_sched_session_id_tl \l_sched_session_prop
    \tl_if_eq:NNF \l_sched_session_prop \q_no_value
    {
      \prop_get_ne:NnN \l_sched_session_prop { type } \l_sched_tmpb_tl
      \sched_type_matches:VnT \l_sched_tmpb_tl {#1}
      {
        \prop_get_ne:NnN \l_sched_record_prop { index } \l_sched_tmpa_tl
        \prop_get_ne:NnN \l_sched_session_prop { talks } \l_sched_talks_tl
        \sched_get_nth_talk:NVN \l_sched_talks_tl \l_sched_tmpa_tl \l_sched_talk_prop
```

```
        \prop_get_ne:NnN \l_sched_talk_prop { cancelled } \l_sched_tmpc_tl
        \tl_if_empty:NT \l_sched_tmpc_tl
          { \int_incr:N \l_sched_tmpa_int }
      }
    }
  }
  \int_use:N \l_sched_tmpa_int
}

% \participantCount: count participants not marked as removed.
\cs_new_protected:Npn \sched_count_participants:
{
  \int_zero:N \l_sched_tmpa_int
  \prop_map_inline:Nn \g_participant_info_prop
  {
    \tl_set:Nn \l_sched_participant_prop {##2}
    \prop_get_ne:NnN \l_sched_participant_prop { removed } \l_sched_tmpb_tl
    \tl_if_empty:NT \l_sched_tmpb_tl
      { \int_incr:N \l_sched_tmpa_int }
  }
  \int_use:N \l_sched_tmpa_int
}

\NewDocumentCommand{\sessionCount}{m} { \sched_count_sessions:n {#1} }
\NewDocumentCommand{\talkCount}{m}    { \sched_count_talks:n {#1} }
\NewDocumentCommand{\participantCount}{} { \sched_count_participants: }
```

4.1.12 Output commands

`\sched_output_session_listing:Nn` renders one session: coloured slot bar and `\subsection*` title (combined via `\sched_colored_subsection:nnn`), organiser list, session description body (re-input with render mode active), and talk list. Cancelled talks appear with strikethrough and no page reference.

`\sched_render_talks:Nn` iterates `\g_schedule_talk_order_seq` in schedule order, filters by type, and renders each non-cancelled talk. Argument `#1` is a bool controlling whether the chair line is included (true for starred `\renderTalks`).

```
\cs_new_protected:Npn \sched_output_session_listing:Nn #1 #2
{
  \sched_session_color:NNN #1 \l_sched_color_tl \l_sched_type_label_tl

  \prop_get_ne:NnN \g_sched_session_slots_prop {#2} \l_sched_title_tl
  \prop_get_ne:NnN #1 { location } \l_sched_location_tl
  \sched_colored_subsection:Vnn \l_sched_color_tl
    {
      \tl_use:N \l_sched_title_tl
      \tl_if_empty:NF \l_sched_location_tl
        { ,\enskip \tl_use:N \l_sched_location_tl }
    }
    { \prop_item:Nn #1 { title } }
  \label{session:#2}

  \noindent\textbf{Organizers:}\par\medskip
  \prop_get_ne:NnN #1 { organizers } \l_sched_tmpa_tl
```

```
\seq_map_inline:Nn \l_sched_tmpa_tl
{
  % ##1 is a participant key; look up all display data from the participant DB.
  \prop_get:NnN \g_participant_info_prop {##1} \l_sched_participant_prop
  \tl_if_eq:NNT \l_sched_participant_prop \q_no_value
    { \msg_warning:nnn { session } { participant-not-found } {##1} }
  \tl_if_eq:NNF \l_sched_participant_prop \q_no_value
  {
    \prop_get_ne:NnN \l_sched_participant_prop { name  } \l_sched_tmpb_tl
    \noindent\textit{ \tl_use:N \l_sched_tmpb_tl }\par
    \prop_get_ne:NnN \l_sched_participant_prop { affil } \l_sched_tmpb_tl
    \tl_if_empty:NF \l_sched_tmpb_tl { \noindent \tl_use:N \l_sched_tmpb_tl \par }
    \prop_get_ne:NnN \l_sched_participant_prop { email } \l_sched_tmpb_tl
    \tl_if_empty:NF \l_sched_tmpb_tl
      { \noindent{\ttfamily \tl_use:N \l_sched_tmpb_tl}\par }
    \sched_append_unique_to_prop_clist:Nnn
      \g_participant_sessions_prop {##1} {#2}
  }
  \smallskip
}
\medskip

\prop_get_ne:NnN #1 { filepath } \l_sched_tmpa_tl
\tl_if_empty:NF \l_sched_tmpa_tl
{
  \noindent\textbf{Session~Description:}\par
  \bool_gset_true:N  \g_sched_render_body_bool
  \filename@parse{\l_sched_tmpa_tl}
  \begingroup\edef\x{\noexpand\graphicspath{{\filename@area}}}\expandafter\endgroup\x
  \file_input:V \l_sched_tmpa_tl
  \bool_gset_false:N \g_sched_render_body_bool
  \medskip
}

\prop_get_ne:NnN #1 { talks } \l_sched_tmpa_tl
\seq_map_indexed_inline:Nn \l_sched_tmpa_tl
{
  \tl_set:Nn \l_sched_talk_prop {##2}
  \prop_get_ne:NnN \l_sched_talk_prop { cancelled } \l_sched_tmpb_tl
  \prop_get_ne:NnN \l_sched_talk_prop { speakerkey } \l_sched_tmpa_tl
  \sched_participant_name:eN { \tl_use:N \l_sched_tmpa_tl } \l_sched_tmpa_tl
  \tl_if_empty:NTF \l_sched_tmpb_tl
  {
    \noindent\textit{ \tl_use:N \l_sched_tmpa_tl },~
    ~``\prop_item:Nn \l_sched_talk_prop { title }''
    \fillPageref{talk:#2@##1}\par
  }
  {
    \noindent\cancelledTalkFormat{
      \textit{ \tl_use:N \l_sched_tmpa_tl },~
      ``\prop_item:Nn \l_sched_talk_prop { title }''
    }\par
  }
  \smallskip
```

```
  }
  \bigskip
}

\cs_new_protected:Npn \sched_list_sessions_by_type:n #1
{
  \bool_set_false:N \l_sched_tmpa_bool
  \seq_map_inline:Nn \g_schedule_sessions_seq
  {
    \sched_get_session:nnNT {##1} \l_sched_session_prop
    {
      \prop_get_ne:NnN \l_sched_session_prop { type } \l_sched_tmpb_tl
      \sched_type_matches:VnT \l_sched_tmpb_tl {#1}
      {
        \bool_set_true:N \l_sched_tmpa_bool
        \sched_output_session_listing:Nn \l_sched_session_prop {##1}
      }
    }
  }
  \bool_if:NF \l_sched_tmpa_bool
  { \msg_warning:nnn { schedule } { no-sessions-of-type } {#1} }
}

\NewDocumentCommand{\renderSessions}{m}
  { \sched_list_sessions_by_type:n {#1} }

\cs_new_protected:Npn \sched_render_talks:Nn #1 #2
{
  \seq_map_inline:Nn \g_schedule_talk_order_seq
  {
    \tl_set:Nn \l_sched_record_prop {##1}
    \prop_get_ne:NnN \l_sched_record_prop { session } \l_sched_session_id_tl
    \prop_get:NVN \g_sessions_prop \l_sched_session_id_tl \l_sched_session_prop
    \tl_if_eq:NNT \l_sched_session_prop \q_no_value
      { \msg_warning:nnV { schedule } { render-missing-session } \l_sched_session_id_tl }
    \tl_if_eq:NNF \l_sched_session_prop \q_no_value
    {
      \prop_get_ne:NnN \l_sched_session_prop { type } \l_sched_tmpc_tl
      \sched_type_matches:VnT \l_sched_tmpc_tl {#2}
      {
        \prop_get_ne:NnN \l_sched_record_prop { slot    } \l_sched_tmpb_tl
        \sched_session_color:NNN \l_sched_session_prop \l_sched_color_tl \l_sched_type_label
        \prop_get_ne:NnN \l_sched_session_prop { talks } \l_sched_talks_tl

        \prop_get_ne:NnN \l_sched_record_prop { index   } \l_sched_tmpa_tl
        \sched_get_nth_talk:NVN \l_sched_talks_tl \l_sched_tmpa_tl \l_sched_talk_prop
        \prop_get_ne:NnN \l_sched_talk_prop { title } \l_sched_title_tl

        \tl_if_empty:NF \l_sched_title_tl
        {
          \prop_get_ne:NnN \l_sched_talk_prop { cancelled } \l_sched_tmpc_tl
          \tl_if_empty:NT \l_sched_tmpc_tl
          {
          \prop_get_ne:NnN \l_sched_session_prop { location } \l_sched_location_tl
```

```
\sched_colored_subsection:Vnn \l_sched_color_tl
{
  \tl_use:N \l_sched_tmpb_tl
  \tl_if_empty:NF \l_sched_location_tl
  { ,\enskip \tl_use:N \l_sched_location_tl }
}
{ \tl_use:N \l_sched_title_tl }
\label{talk:\tl_use:N \l_sched_session_id_tl @\tl_use:N \l_sched_tmpa_tl}

% Use the stored speaker key directly for participant tracking.
\prop_get_ne:NnN \l_sched_talk_prop { speakerkey } \l_sched_participant_key_tl
\tl_if_empty:NF \l_sched_participant_key_tl
{
  \tl_set:Ne \l_sched_tmpc_tl
    { \tl_use:N \l_sched_session_id_tl @ \tl_use:N \l_sched_tmpa_tl }
  \sched_append_unique_to_prop_clist:NVV
    \g_participant_talks_prop \l_sched_participant_key_tl \l_sched_tmpc_tl
}

\prop_get_ne:NnN \l_sched_session_prop { title } \l_sched_tmpc_tl
\tl_if_empty:NF \l_sched_tmpc_tl
{
  \noindent\textit{Session:~\tl_use:N \l_sched_tmpc_tl}
  \quad \fillPageref{session:\tl_use:N \l_sched_session_id_tl}
  \par
}

\bool_if:NT #1
{
  \sched_participant_name:eN
    { \prop_item:Nn \l_sched_session_prop { chair } } \l_sched_chair_tl
  \tl_if_empty:NF \l_sched_chair_tl
    { \noindent\textit{Chair:~\tl_use:N \l_sched_chair_tl}\par }
}

% Look up all speaker display data from the participant database.
\prop_get:NVN \g_participant_info_prop \l_sched_participant_key_tl
  \l_sched_participant_prop
\tl_if_eq:NNT \l_sched_participant_prop \q_no_value
  { \msg_warning:nnV { session } { participant-not-found }
      \l_sched_participant_key_tl }
\tl_if_eq:NNF \l_sched_participant_prop \q_no_value
{
\prop_get_ne:NnN \l_sched_participant_prop { name  } \l_sched_speaker_tl
\prop_get_ne:NnN \l_sched_participant_prop { affil } \l_sched_tmpa_tl
\prop_get_ne:NnN \l_sched_participant_prop { email } \l_sched_tmpb_tl
\prop_get_ne:NnN \l_sched_talk_prop { coauthors    } \l_sched_tmpc_tl
\noindent \tl_use:N \l_sched_speaker_tl \par
\tl_if_empty:NF \l_sched_tmpa_tl
{ \noindent \tl_use:N \l_sched_tmpa_tl \par }
\tl_if_empty:NF \l_sched_tmpb_tl
{ \noindent\nolinkurl{ \tl_use:N \l_sched_tmpb_tl } \par }
\tl_if_empty:NF \l_sched_tmpc_tl
{ \noindent\textit{Coauthor(s):~\tl_use:N \l_sched_tmpc_tl} \par }
```

```
          }

          \medskip
          \prop_get_ne:NnN \l_sched_talk_prop { filepath } \l_sched_title_tl
          \bool_gset_true:N  \g_sched_render_body_bool
          \filename@parse{\l_sched_title_tl}
          \begingroup\edef\x{\noexpand\graphicspath{{\filename@area}}}\expandafter\endgroup\
          \file_input:V \l_sched_title_tl
          \bool_gset_false:N \g_sched_render_body_bool
          \bigskip
          }
        }
      }
    }
  }
}

\NewDocumentCommand{\renderTalks}{s m}
{
  \IfBooleanTF {#1}
    { \sched_render_talks:Nn \c_true_bool  {#2} }
    { \sched_render_talks:Nn \c_false_bool {#2} }
}

% \cs{sched\_build\_sorted\_participant\_list:} populates
% \cs{g\_participant\_display\_prop} and
% \cs{l\_sched\_sorted\_participants\_seq} from the participant
% database, skipping removed entries and sorting case-insensitively
% by display name.  Called by both \cs{sched\_print\_participants:}
% and \cs{sched\_print\_participant\_slots:n}.
\cs_new_protected:Npn \sched_build_sorted_participant_list:
{
  \prop_gclear:N \g_participant_display_prop
  \seq_clear:N   \l_sched_sorted_participants_seq
  \prop_map_inline:Nn \g_participant_info_prop
  {
    \tl_set:Nn \l_sched_participant_prop {##2}
    \prop_get_ne:NnN \l_sched_participant_prop { removed } \l_sched_tmpb_tl
    \tl_if_empty:NT \l_sched_tmpb_tl
    {
      \prop_get_ne:NnN \l_sched_participant_prop { display } \l_sched_tmpa_tl
      \prop_gput:NnV \g_participant_display_prop {##1} \l_sched_tmpa_tl
      \seq_put_right:Nn \l_sched_sorted_participants_seq {##1}
    }
  }
  \seq_sort:Nn \l_sched_sorted_participants_seq
  {
    \str_compare:eNeTF
      { \str_casefold:e { \prop_item:Nn \g_participant_display_prop {##1} } }
      >
      { \str_casefold:e { \prop_item:Nn \g_participant_display_prop {##2} } }
      { \sort_return_swapped: }
      { \sort_return_same: }
  }
```

```
}

% \cs{sched\_print\_participants:Nn} is the single implementation behind
% \cs{printParticipants}.  Argument~|#1| is a bool: true adds slot strings
% below the page references (starred form).  Argument~|#2| is the minimum
% engagement count; participants below the threshold are silently skipped.
% In the starred form, each slot line shows the session title in quotes,
% an inline page reference, and the slot string.
% Session lines are labelled ``Organizer:'' and talk lines ``Speaker:''.
% A warning is issued for any item whose slot cannot be resolved.
\cs_new_protected:Npn \sched_print_participants:Nn #1 #2
{
  \sched_build_sorted_participant_list:
  \begin{multicols}{2}
  \seq_map_inline:Nn \l_sched_sorted_participants_seq
  {
    % Fetch engagement lists and test against the minimum threshold.
    \prop_get_ne:NnN \g_participant_sessions_prop {##1} \l_sched_tmpb_tl
    \prop_get_ne:NnN \g_participant_talks_prop    {##1} \l_sched_tmpc_tl
    \int_zero:N \l_sched_tmpa_int
    \tl_if_empty:NF \l_sched_tmpb_tl
      { \int_add:Nn \l_sched_tmpa_int { \exp_args:NV \clist_count:n \l_sched_tmpb_tl } }
    \tl_if_empty:NF \l_sched_tmpc_tl
      { \int_add:Nn \l_sched_tmpa_int { \exp_args:NV \clist_count:n \l_sched_tmpc_tl } }
    \int_compare:nT { \l_sched_tmpa_int >= #2 }
    {
      \prop_get:NnN \g_participant_info_prop {##1} \l_sched_participant_prop
      \noindent\begin{minipage}{\linewidth}
      \prop_get_ne:NnN \l_sched_participant_prop { display } \l_sched_tmpa_tl
      \noindent\textbf{ \tl_use:N \l_sched_tmpa_tl }\par
      \prop_get_ne:NnN \l_sched_participant_prop { affil } \l_sched_tmpa_tl
      \tl_if_empty:NF \l_sched_tmpa_tl
        { \noindent \tl_use:N \l_sched_tmpa_tl \par }
      \prop_get_ne:NnN \l_sched_participant_prop { email } \l_sched_tmpa_tl
      \tl_if_empty:NF \l_sched_tmpa_tl
        { \noindent\nolinkurl{ \tl_use:N \l_sched_tmpa_tl } \par }
      % Page references (always shown).
      \exp_args:NV \clist_map_inline:nn \l_sched_tmpb_tl
        { \blockPageref{session:####1}~ }
      \exp_args:NV \clist_map_inline:nn \l_sched_tmpc_tl
        { \blockPageref{talk:####1}~ }
      % Slot strings with title and inline page ref (starred form only).
      \bool_if:NT #1
      {
        \par
        % Organiser sessions: \l_sched_tmpb_tl is captured so free to reuse.
        \exp_args:NV \clist_map_inline:nn \l_sched_tmpb_tl
        {
          \prop_get_ne:NnN \g_sched_session_slots_prop {####1} \l_sched_tmpa_tl
          \tl_if_empty:NTF \l_sched_tmpa_tl
            { \msg_warning:nnn { schedule } { slot-not-found } {####1} }
            {
              \tl_clear:N \l_sched_tmpb_tl
              \sched_get_session:nNT {####1} \l_sched_session_prop
```

```
                { \prop_get_ne:NnN \l_sched_session_prop { title } \l_sched_tmpb_tl }
              \noindent\textit{Organizer:}~
              \tl_if_empty:NF \l_sched_tmpb_tl { ``\tl_use:N \l_sched_tmpb_tl'',~ }
              \blockPageref{session:####1},~\tl_use:N \l_sched_tmpa_tl\par
            }
        }
        % Speaker talks: split key on @ to get session id, look up title.
        % \l_sched_tmpc_tl is captured so free to reuse for the title.
        \exp_args:NV \clist_map_inline:nn \l_sched_tmpc_tl
        {
          \prop_get_ne:NnN \g_sched_talk_slots_prop {####1} \l_sched_tmpa_tl
          \tl_if_empty:NTF \l_sched_tmpa_tl
            { \msg_warning:nnn { schedule } { slot-not-found } {####1} }
            {
              \seq_set_split:Nnn \l_sched_tmpa_seq { @ } {####1}
              \tl_set:Ne \l_sched_tmpb_tl { \seq_item:Nn \l_sched_tmpa_seq {1} }
              \tl_clear:N \l_sched_tmpc_tl
              \prop_get:NVN \g_sessions_prop \l_sched_tmpb_tl \l_sched_session_prop
              \tl_if_eq:NNF \l_sched_session_prop \q_no_value
                { \prop_get_ne:NnN \l_sched_session_prop { title } \l_sched_tmpc_tl }
              \noindent\textit{Speaker:}~
              \tl_if_empty:NF \l_sched_tmpc_tl { ``\tl_use:N \l_sched_tmpc_tl'',~ }
              \blockPageref{talk:####1},~\tl_use:N \l_sched_tmpa_tl\par
            }
        }
      }
      \end{minipage}\par\bigskip
    }
  }
  \end{multicols}
}

\NewDocumentCommand{\printParticipants}{s O{0}}
{
  \IfBooleanTF {#1}
    { \sched_print_participants:Nn \c_true_bool  {#2} }
    { \sched_print_participants:Nn \c_false_bool {#2} }
}

\NewDocumentCommand{\missingTalk}{m}
{
  \tl_if_empty:NTF \g_current_session_id_tl
    { \msg_error:nn { session } { missing-talk-outside-session } }
    {
      \begin{talk}{}{}{}{}{}{}
        EMPTY TALK
      \end{talk}
    }
}

\NewDocumentCommand{\blockPageref}{m} {\ifcsname r@#1\endcsname
    p.\nobreakspace{}\pageref{#1}
  \fi}
\NewDocumentCommand{\fillPageref}{m}
```

```
{\ifcsname r@#1\endcsname
    ~\unskip\penalty0\hspace*{\fill}\mbox{p.\nobreakspace{}\pageref{#1}}
  \fi}
```

4.1.13 Column types

Column `F` must be defined inside `\ExplSyntaxOn` because its width formula references `\l_sched_numcols_tl`, an expl3 variable whose name uses underscores as letters.

```
\newcolumntype{Y}{>{\centering\arraybackslash}X}
\newcolumntype{H}{>{\raggedright\arraybackslash}X}
\newcolumntype{F}
  {>{\hsize=\dimexpr
      \l_sched_numcols_tl\hsize
      + \tabcolsep  * (2 * (\l_sched_numcols_tl - 1))
      + \arrayrulewidth * (\l_sched_numcols_tl - 1)
    \relax}H}

\ExplSyntaxOff
```

4.2 Lua submission processor (`process-submissions.lua`)

The module is a standard Lua table returned with `return M`. It requires `lfs` (LuaFileSystem, bundled with LuaTEX).

`M.process` iterates all sorted `.tex` files in a session directory and inputs each one. Each file is expected to contain exactly one environment; content outside an environment is silently ignored by TEX. Files with leading digits are tracked as talk slots: gaps trigger `\missingTalk`; files without leading digits (typically the session description file) are input but do not advance the slot counter.

`M.process\_root` iterates immediate subdirectories (skipping hidden entries), bracketing each with `\openSession`/`\closeSession` and delegating file processing to `M.process`.

```
local M = {}

local lfs = require("lfs")

local function list_tex(dir)
    local t = {}
    for f in lfs.dir(dir) do
        if f:match("%.tex$") then
            t[#t+1] = f
        end
    end
    table.sort(t)
    return t
end

function M.process(dir)
    local files = list_tex(dir)
    local expected = 1

    for _, f in ipairs(files) do
        local idx = f:match("^(%d+)")
        if idx then
```

```
            idx = tonumber(idx)
            while expected < idx do
                tex.print("\\missingTalk{" .. expected .. "}")
                expected = expected + 1
            end
            tex.print("\\def\\currentTalkIndex{" .. idx .. "}")
            expected = idx + 1
        else
            tex.print("\\def\\currentTalkIndex{}")
        end

        tex.print("\\setInputFilepath{" .. dir .. "/" .. f .. "}")
        tex.print("\\input{" .. dir .. "/" .. f .. "}")
    end
end

function M.process_root(dir)
    for d in lfs.dir(dir) do
        if d:sub(1,1) ~= "." then
            local full_path = dir .. "/" .. d
            local attr = lfs.attributes(full_path)

            if attr and attr.mode == "directory" then
                tex.print("\\openSession{" .. d .. "}{" .. full_path .. "}")
                M.process(full_path)
                tex.print("\\closeSession")
            end
        end
    end
end

return M
```

4.3 Submission templates

Two standalone template files are generated for submitters. Each compiles independently (no confschedule required) and renders a preview that approximates the final programme output.