

Information Engineering Technology

# Rapide Release Notes



Release 3.0.0

# Contents

<b>New Functionality (3.0)</b>	<b>5</b>
Rapide Generator No Assemble.....	5
New Column Tracking option – place field value in column header .....	5
Overlay only appear when row is selected .....	5
<b>New Functionality (2.0)</b>	<b>5</b>
Keep EAB proxy DLL Loaded .....	5
Interrupt Support.....	5
Exclude OPSLIB from Assemble .....	6
Popup Field Dialog .....	6
MessageBox support for custom buttons.....	6
NEXTTRAN_PREFIX token .....	6
Screen to GUI Transformation Enhancements .....	6
isLetterOrDigit method .....	6
Support loading Transfer Flow PSteps into Frame .....	7
New event methods postEventByName and postEventToSelf .....	7
Fire-and-forget background USE .....	7
Auto-Suggest drop-downs for Browser .....	7
Support Custom Fonts in Browser .....	7
<NONE> row to unset a permitted value drop-down.....	7
New RFL methods to dynamically change Min Width/Height of window .....	7
Option MINSIZE_DESIGNED_IF_NOT_SPECIFIED width/height separate control .....	7
Auto-width listbox columns to proportionally fit space available .....	8
Blockmode conversion selection character override .....	8
Z Order for fields can be defined .....	8
Start Protected option for controls .....	8
ScrollTop and ScrollBottom support.....	8
Enter key with no default button .....	8
Minor performance enhancement .....	8
<NONE> row to unset a permitted value drop-down – enhanced further.....	9
FileSelector Extensions parameter .....	9
Listbox auto-check option.....	9
Window/Dialog Transparency .....	9
Window/Dialog No Title Bar .....	9
Fields and Prompts Vertical Alignment.....	9

RAP MakeData support in modified block-mode conversions .....	9
Override all Date Edit Patterns .....	9
Customise colour of auto-new-row cells .....	10
New UserExit to validate NEXTTRAN permission.....	10
Clearable Date Entry Fields.....	10
Provided alternative small checkbox images for browser theme .....	10
Mechanism to intercept browser keys .....	10
Minimum size for checkbox control .....	10
Use Edit Pattern as Hint for Date/Time/Timestamp fields .....	10
RAP Make Data support to tolerate imperfect view matching.....	10
Option to give mandatory import field mandatory indicator .....	11
Mandatory fields to prevent execution .....	11
Tab into listbox entry field will auto-select whole of current value .....	11
Override default numeric edit pattern for block-mode behaviour .....	11
Designer Filter Keep Structure.....	11
Mandatory fields to prevent execution – exclude event from check.....	11
setBitmap method to support Windows/Dialogs .....	11
USE_EDPAT_AS_FIELD_HINTS for Time fields and Overlays.....	11
Designer P-Step Filter .....	11
EDPAT_DATE_OVERRIDE specify edit patterns to apply to .....	11
NEXTTRAN invalid – new popup .....	12
Block Mode Re-transformation of Edit Patterns .....	12
Client-Side Auto-Tab for Browser .....	12
NEXTTRAN with Clear Screen Input .....	12
New control type – Listboxes displayed as Dynamic Menus for browser .....	12
Field with Markup supports content from file .....	12
NEXTTRAN with Clear Screen Input .....	12
Select & Copy of listbox data .....	12
Block Mode Transformation – custom video properties .....	13
Block Mode Transformation – custom edit patterns.....	13
Provide override auto-fit column width option for individual list boxes.....	13
Auto-Height/Width for containers in Browser .....	13
Auto-Tab in overlay field of list boxes .....	14
Auto-Check in Auto-New-Row Listboxes to add new row .....	14
Permitted value drop-downs implementation as read-only fields or auto-suggest..	14
New API methods for processing child psteps and trancodes.....	14
New API methods to retrieve application assembly time .....	14
Change tracking control programmatically .....	14

New Position-Tracking option to track relative to centre of window.....	14
Disabled overlay controls able to be unprotected by MAKE .....	14
Overlay HTML properly configurable for Checkboxes .....	15

## New Functionality (3.0)

---

The following features have been added in Release 3.0.0

### Rapide Generator No Assemble

---

Some projects opt to use their own assemble process rather than the standard rapide assemble. If an environment variable RAPIDE\_NO\_ASSEMBLE is set to YES, the Assemble checkbox is not ticked automatically in the Rapide Generator.

### New Column Tracking option – place field value in column header

---

A new type of tracking has been provided to allow the value of a designed field to be used as the prompt for a listbox column header. The designed field itself will not be placed, but the value of the attribute view will be used to update the column header dynamically.

### Overlay only appear when row is selected

---

A new option for an overlay column causes the overlay controls to only appear for a selected row in a listbox. This improves performance as it reduces the number of controls visible at a time. This feature also works for dynamic overlays, but can be overridden for individual cells by passing 'A' or 'D' as the 'enabled' flag in the coded data string. 'A' means always show overlay, even if row is not selected. 'D' means always show disabled, even if row is not selected.

## New Functionality (2.0)

---

The following features were previously added in Release 2.0 as interim updates

### Keep EAB proxy DLL Loaded

---

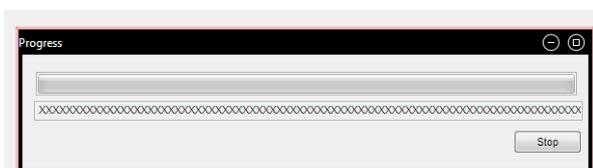
The Rapide EAB Proxy uses Java Native Access (JNA) to invoke a C function contained within the DLL. By default the DLL library will be loaded on first access and then the reference to it will be eligible for garbage collection by the JVM and the DLL unloaded. This means that any static variables defined in the C code will lose their value on a subsequent call to the DLL if it has been re-loaded.

A new Rapide generator property for a proxy EAB (Keep Loaded) can be set to ensure that a reference to the DLL is maintained by the Rapide runtime so that static variables are retained between calls to the EAB function.

### Interrupt Support

---

A new property specified for a modal dialog box allows a progress style dialog box to be interrupted with a Pause/Cancel button. When enabled, the dialog box should only have read-only controls apart from one or more push buttons that the user can press to interrupt processing. If the user presses the button, the Rapide Function Library interruptControl property is set to the name of the control pressed. Your logic determines the action taken, for example to stop processing or provide a pause/start function.



```
554 | | | ++->IF temp.rapide.object . interruptControl = "CancelGenerate"
555 | | | | CLOSE Dialog Box RAPIDE_GENERATOR_PROGRESS
556<-----ESCAPE
```

---

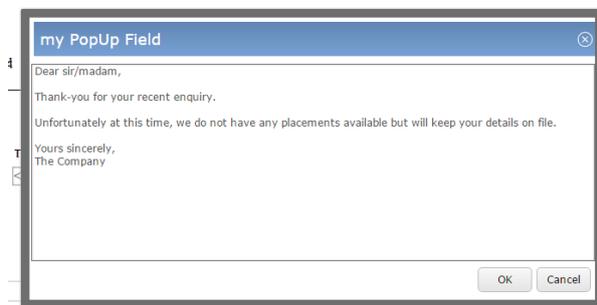
## Exclude OPSLIB from Assemble

A new property can be specified for an OPSLIB to exclude it from the assemble when defining the opslib properties in the Rapide Generator plug-in. If the exclude property is set, then the opslib .jar will not be included in the application .war and you must ensure that the .jar file is available separately in the application server context.

---

## Popup Field Dialog

An in-built API has been made available for editing/displaying multi-line text fields in a larger dialog. This can be useful where limited space on a UI results in the text value being hard to read/edit. The new API is accessed from a Rapide Function Library method `popupField`, and displays a modal dialog with a single multi-line text field which is bound to a specified export view. Buttons can be chosen, as well as various other visual aspects.



This API is only support for BROWSER targets, where visual appearance of the dialog can be customised with CSS.

---

## MessageBox support for custom buttons

The standard Gen MessageBox function implementation in Rapide now supports custom buttons by simply passing in a string of button texts separated by semi-colons e.g. "Ready;More Info;Cancel" etc. The text of the button selected is returned by the function.

---

## NEXTTRAN\_PREFIX token

Automatic prefixing and un-prefixing of NEXTTRAN and TRANCODE values can be achieved with this token – potentially useful for Block Mode conversion projects where online screens have been split into separate client and server PSteps. See Rapide Developer guide for more information.

---

## Screen to GUI Transformation Enhancements

- 1) A new property allows you to remove specified characters from the prompt, for example trailing '.' Characters.
- 2) The window control help description property can be set based on the field helpid from help descriptions placed in the RapideTransformationHelpDesc.txt file.

---

## isLetterOrDigit method

Rapide Function Library provides new method to determine if a character is a valid letter or digit (returning True), including national language characters, or returns False otherwise e.g. punctuation character etc.

## **Support loading Transfer Flow PSteps into Frame**

---

Additional support is now provided for loading PSteps that are launched via a Transfer Flow into the same parent as the PStep that initiated the flow. Works with normal Frames and windows that have the 'loadIntoParent' property set.

## **New event methods postEventByName and postEventToSelf**

---

New method postEventToSelf is provided for convenience – it is the same as postEventToPStep but with a blank PStep i.e. post to self. Also added new method postEventByName which also queues an event in the current PStep, but using the event name from the Action Diagram, and NOT the UI design's Event Type. This means you don't even need to define an event in the UI Design – you just create the event itself in the logic.

## **Fire-and-forget background USE**

---

An experimental method has been provided to allow a USE statement to be started as fire-and-forget, so it executed in the background in a separate thread. This allows for a style of asynchronous processing, although no return values can be managed. It has been created with the new AutoSuggest drop-down support primarily in mind (see next item). The new method nextUSEFireAndForget is available for this purpose in the Rapide Function Library.

## **Auto-Suggest drop-downs for Browser**

---

Many websites provide auto-suggest drop-downs, attached to entry fields, which provide lists of possible value matches for the text entered so far. This is now possible in Rapide for browser applications by creating a user event called "AutoSuggest" for an entry field. Rapide will automatically provide calls to that event whenever the value in the field is modified (as the user types). It is anticipated that this event will make use of the nextUSEFireAndForget method described above to execute logic to populate the drop-down list of possible values, calling new methods in the Rapide Function Library autoSuggestClearItems, autoSuggestAddItem and autoSuggestShow. Note that this is currently an experimental feature, as the effects of potential multiple threads of Gen logic on the same GlobData/UI may have some unexpected effects.

## **Support Custom Fonts in Browser**

---

Normally for browser applications, any fonts used must be already installed and available on the client machine. Rapide now provides a facility to allow use of external fonts, either hosted elsewhere (such as Google's WebFonts) or supplied packaged into your application. This is by use of CSS and font file resources, and a new token in rapide.properties which instructs Rapide to load a CSS file locally to the browser (LOAD\_CLIENT\_FILE). See the Rapide Developer Guide documentation for more information.

## **<NONE> row to unset a permitted value drop-down**

---

By setting new rapide.properties token PVDROPDOWN\_INCLUDE\_UNSET\_ROW=True, any Permitted Value Drop-Down control that does not already include SPACES as one of its' PVs, will have a <NONE> row provided at the top, similar to existing behaviour for Drop-Down List controls (COMBO\_LISTBOX\_INCLUDE\_UNSET\_ROW token). The text for the <NONE> row is taken from Rapide's runtime strings for the Locale, the same as the Combo functionality.

## **New RFL methods to dynamically change Min Width/Height of window**

---

New methods setMinimumWindowWidth/Height are provided to allow dynamic changing of the minimum window width/height. This can be used to affect when scrollbars start to appear during resizing of windows.

## **Option MINSIZE\_DESIGNED\_IF\_NOT\_SPECIFIED width/height separate control**

---

The existing rapide.properties token MINSIZE\_DESIGNED\_IF\_NOT\_SPECIFIED affected both width and height, but sometimes it is useful to use just the width, or just the height of the designed window as the minimum, so

2 alternative tokens are now provided MINWIDTH\_DESIGNATED\_IF\_NOT\_SPECIFIED and MINHEIGHT\_DESIGNATED\_IF\_NOT\_SPECIFIED. The original token MINSIZE\_DESIGNATED\_IF\_NOT\_SPECIFIED should be set to False or commented out in order to use these alternative tokens.

---

### **Auto-width listbox columns to proportionally fit space available**

---

A new rapide.properties token LISTBOX\_AUTO\_WIDTH can be set to True, which will automatically fill the available width of a listbox proportionally with its' columns, if the listbox is wider than the total width of the columns. This applies to all listboxes, not just those with no column headers.

---

### **Blockmode conversion selection character override**

---

By default, repeating group views translated into listboxes by the blockmode conversion utility will have their selection character set to '\*', the same as a normal Gen GUI window. A new token BLOCKMODE\_SELECT\_CHAR may be defined to override this character, as it is common to use 'S' (for example) as the selection character in blockmode design.

---

### **Z Order for fields can be defined**

---

The Z Order of a field can now be defined in the Rapide Designer to be TOP or BOTTOM, similar to existing functionality for pictures and literals.

---

### **Start Protected option for controls**

---

Any editable singleton control may not be specified as 'Start Protected' which is equivalent to using a MAKE PROTECTED statement on that attribute on initial display of the window.

For SCREENs converted to Rapide, any field defined as Protected in the screen's design will automatically now have the 'start protected' property switched on. When using the 'Clear Make' option (should be default for block-mode conversion), the 'Start Protected' property will be reapplied for every execution of the PStep's logic, which is consistent behaviour with Gen's block-mode runtime.

This enhancement has also affected the RAPSVSP source code post-processor. Source code will need to be re-compiled using the updated post-processor.

---

### **ScrollTop and ScrollBottom support**

---

Rapide now supports ScrollTop and ScrollBottom events on standard listboxes and trees on all DESKTOP and BROWSER.

---

### **Enter key with no default button**

---

When there is no Default Button on a window design, the ENTER key will normally do nothing. This behaviour can be overridden with a new rapide.properties token ENTER\_NO\_DEFAULT\_BUTTON\_ACTION which will cause the PStep logic to be executed if set, with the COMMAND set to the value defined by this token. Special value <CURRENT> will allow the COMMAND value to be taken from a COMMAND field (for block-mode converted designs), and a special value of <SPACES> will blank the COMMAND before execution of the PStep logic. This is designed primarily for applications converted from block-mode designs.

---

### **Minor performance enhancement**

---

A general change has been applied to many areas of the Rapide runtime, relating to string handling for logging, which improves performance slightly.

## **<NONE> row to unset a permitted value drop-down – enhanced further**

---

The token `PVDROPDOWN_INCLUDE_UNSET_ROW=True` in `rapide.properties` has been further enhanced to provide an option that works similarly to WebGen. By setting to 'W' (or WebGen), any permitted value drop-down control that uses an attribute that does not have a 'default' value defined in the attribute definition, will include a <NONE> row. This could of course result in having 2 rows that relate to a value of ' ', but that's how WebGen works. It is recommended to use the True value instead, which adds a <NONE> row only if there is no other included value of ' ' in the control's definition.

## **FileSelector Extensions parameter**

---

The client file selector method now comes in an alternative form with an extra argument for taking a list of file extensions for the dialog to present. This will work with DESKTOP targets only, as it requires the RAP runtime to be upgraded to a future version (3.2), which will happen with the next Rapide major release.

## **Listbox auto-check option**

---

Any listbox may now be defined as Checkable, even where the first designed column is also an overlay (was previously a restriction). In addition, where the listbox contains overlay controls, the checkbox for the row can be automatically switched on (`RAPIDE.ROW_CHECKED` view = "Y") whenever a cell value is changed or entered, including in an auto-new row. The group view must contain either a `RAPIDE.ROW_CHECKED` attribute or a `RAPIDE_ROW_CHECKED` attribute in any entity view.

## **Window/Dialog Transparency**

---

For BROWSER, a transparency can be specified for a whole window/dialog in the Designer.

## **Window/Dialog No Title Bar**

---

A window/dialog can be specified with no title bar in the Designer – ignored for full-screen windows/dialogs.

## **Fields and Prompts Vertical Alignment**

---

By default, all entry fields and their prompts are vertically aligned to the top of the control's rectangle, which is consistent with Gen's behaviour. A new Designer property can be defined for fields and their prompts independently to specify bottom or middle vertical alignment instead.

Alternatively, new `rapide.properties` tokens `VERT_ALIGN_FIELDS` and `VERT_ALIGN_PROMPTS` may be used to define the default behaviour of vertical alignment to `BOTTOM` or `CENTRE` if the individual control does not explicitly define alignment.

## **RAP MakeData support in modified block-mode conversions**

---

When a block-mode pstep is converted into client and server co-operative psteps, initially there will only be a single pstepUSE to the new co-op server, but if the client pstep logic is enhanced later to include calls to other server psteps, then Rapide MakeData and NextTran support populated from the original server will only support one server pstep – which one is unpredictable. Therefore, this MakeData and NextTran support has been enhanced to support pstepUSES to multiple co-op servers, with the restriction that if the same server is used in multiple pstepUSES, then entire export view mapping is identical in every call. A regeneration of the Rapide GUI XML for any such client psteps is required for any psteps that pstepUSE multiple servers.

## **Override all Date Edit Patterns**

---

A new `rapide.properties` token, `EDPAT_DATE_OVERRIDE`, may be set as an override to be used for all date edit patterns in listboxes and standard date entry fields. This will not affect Date/Time pickers or Calendar controls, which always take their edit pattern from the Locale. A special value of 'LOCALE' may be used for the

token which will deduce the edit pattern to use from the current Locale, meaning that all date fields/ columns will use the same format as Date pickers etc.

---

### **Customise colour of auto-new-row cells**

When using the auto-new-row option in listboxes, the cells for the new row are by default a medium grey (192,192,192 RGB). This colour can now be overridden with a new rapide.properties token NEW\_ROW\_CELL\_COLOUR.

---

### **New UserExit to validate NEXTTRAN permission**

A new exit method in UserExits is available called `determineIfNextTranAllowed`, which allows you to prevent access to specific PSteps when triggered by a NEXTTRAN request. The member name of the requested PStep is provided to the method.

---

### **Clearable Date Entry Fields**

Standard date entry fields can now be enhanced with a 'clear date' button, which appears as a small 'X' on the right side of the control. Clicking on this will cause the value in the date entry field to be set to a 'zero' date i.e. "0001-01-01". This enhancement is enabled by setting a new rapide.properties token DATE\_ENTRY\_FIELDS\_CLEARABLE=True (default is False). Setting to True will cause ALL date entry fields to get the new feature. This does not affect Date picker or Calendar controls.

---

### **Provided alternative small checkbox images for browser theme**

Rapide's default images for checkbox controls are quite large in comparison to Gen's C/GUI/WINDOWS runtime. Alternative smaller checkbox images are supplied in the CSS resources, which you may point the theme CSS to if you prefer.

---

### **Mechanism to intercept browser keys**

Some browser shortcuts may be undesirable in a Rapide application, so a new mechanism is provided to allow trapping and intercepting of specified keys, such as F5 Refresh. Various actions are possible when intercepted including ignore, process as an application accelerator key, run pstep logic with a specified command, or trigger a UI event. This feature is defined by adding BROWSER\_INTERCEPT\_KEY tokens to the rapide.properties file – see the documentation for more information.

---

### **Minimum size for checkbox control**

To support UI designs where checkboxes have been designed too small to fit the checkbox itself into, an minimum pixel width and height may be specified in rapide.properties with new tokens CHECKBOX\_MIN\_WIDTH and CHECKBOX\_MIN\_HEIGHT.

---

### **Use Edit Pattern as Hint for Date/Time/Timestamp fields**

New token USE\_EDPAT\_AS\_FIELD\_HINTS=True to have the edit pattern of date, time and timestamp fields used as the Hint.

---

### **RAP Make Data support to tolerate imperfect view matching**

MAKE DATA mapping from server back into the client is now tolerant of views not matching perfectly – it will match based on view names, so any additional attributes in the client's views will not cause the MAKE DATA to be thrown away.

## **Option to give mandatory import field mandatory indicator**

---

The Rapide Mandatory indicator can be applied automatically to any control where the import view attribute is defined as mandatory. To do this, set new rapide.properties token MAND\_IMPORT\_AUTO\_MAND=True. This gives some level of consistency with block-mode applications.

## **Mandatory fields to prevent execution**

---

A new rapide.properties token PREVENT\_LOGIC\_WHEN\_MAND\_INCOMPLETE=True will prevent any PAD logic (and certain other event types, such as flows) from executing until all fields marked as Mandatory are filled. This is consistent behaviour with block-mode applications. When execution is attempted, and there are incomplete Mandatory fields, a pop-up message box is displayed to tell the user.

## **Tab into listbox entry field will auto-select whole of current value**

---

When you tab the cursor into a listbox entry field (overlay control), previous it would place the caret at the end of the current value, which was inconsistent with tabbing into a normal entry field where the whole of the current value would be highlighted. Listbox entry field highlighting is now the same as normal entry fields. Note that this does not apply to MOBILE targets, where the cursor always goes at the end, with no highlighting.

## **Override default numeric edit pattern for block-mode behaviour**

---

By default, where a numeric field does not specify an edit pattern explicitly, Gen GUI (and Rapide) will default to ZZZZZZ9 style edit patterns. Block-mode's default is all 9s and no Zs, so to get this behaviour a new token is provided in rapide.properties EDPAT\_DEFNUM\_9\_NOT\_Z=True.

## **Designer Filter Keep Structure**

---

When using the Rapide Designer Filter in search mode, a new checkbox allows you to keep the p-step/window/control structure so that you can identify the parent for a displayed window/control.

## **Mandatory fields to prevent execution – exclude event from check**

---

The rapide.properties token PREVENT\_LOGIC\_WHEN\_MAND\_INCOMPLETE=True will prevent any PAD logic from executing until all fields marked as Mandatory are filled. When execution is attempted, and there are incomplete Mandatory fields, a pop-up message box is displayed to tell the user. Individual events may now be marked so that this check is not performed for them, so their logic will be executed regardless of mandatory fields being filled or not.

## **setBitmap method to support Windows/Dialogs**

---

Previously, Rapide's setBitmap method only supported changing the bitmap of a control. Now you can pass in simply the name of the window and this will change the background bitmap of the window/dialogbox itself.

## **USE\_EDPAT\_AS\_FIELD\_HINTS for Time fields and Overlays**

---

The rapide.properties token USE\_EDPAT\_AS\_FIELD\_HINTS=True was only previously being applied for single field Date attributes. It now applies to Time and Timestamp fields, as well as overlay fields of the same types in a listbox.

## **Designer P-Step Filter**

---

A new filter field has been added to the Designer to filter the list based on procedure step name.

## **EDPAT\_DATE\_OVERRIDE specify edit patterns to apply to**

---

EDPAT\_DATE\_OVERRIDE was previously added to cause ALL date fields to take on a specific edit pattern – usually based on the LOCALE of the running system. However, in some fields it may be preferable to continue

using the override edit pattern specified in the model instead. A new token `EDPAT_DATE_OVERRIDE_CASES` is now provided where you can list the edit patterns that will be overridden. If a date field uses an edit pattern NOT specified in this list, then it will use the edit patterns as defined in the model. The list is separated by semi-colons, an example provided in `rapide.properties` (commented by default).

---

### **NEXTTRAN invalid – new popup**

Previously, if a NEXTTRAN value specified an unknown TRANCODE, a console message would be logged, but nothing would be visible to the end user. A pop-up message box will now appear to inform the user of their mistake. Strings for the popup's title and text are provided in the standard translation properties file.

---

### **Block Mode Re-transformation of Edit Patterns**

A new option in the Rapide Designer is available to re-transform field edit patterns from the block mode screen field equivalents.

---

### **Client-Side Auto-Tab for Browser**

A new JavaScript-based auto-tab listener is now used by default on BROWSER to support auto-tab when an entry field is completed. This is much faster than Java-side (in AppServer) which caused a lag between completing the field and typing in the next field (meant that some typed characters were often lost). All previous functionality, including Changed events on the fields should still work, but this new feature is partly experimental still, so if problems are encountered, it can be deactivated by setting `rapide.properties` token `BROWSER_AUTOTBA_IN_JS=False`.

---

### **NEXTTRAN with Clear Screen Input**

Clear Screen Input is now partially supported with NEXTTRAN – passing the input into `COMMAND`.

---

### **New control type – Listboxes displayed as Dynamic Menus for browser**

Rapide listboxes can now be enhanced to render as Dynamic Menus for BROWSER targets, using the same structure and data views as for trees.

---

### **Field with Markup supports content from file**

A field that is enhanced with Markup (BROWSER only) now supports loading the HTML content from a physical file by prefixing the field value with “[FILE]” and specifying the fully-qualified location, including as resources shipped within the application

---

### **NEXTTRAN with Clear Screen Input**

Clear Screen Input now fully supports NEXTTRAN – passing the input according to the target PStep's Clear Screen Input, if defined.

---

### **Select & Copy of listbox data**

By default, listbox data is not selectable, so you cannot highlight the value in a cell and copy-to-clipboard, for example. The “Open as Report” feature is available to present the entire listbox content in HTML, but for listboxes with many rows it can be hard to then locate the specific data you want.

Therefore a new option, enabled with a `rapide.properties` token `ENABLE_CELL_SELECTION=True` is now provided which will automatically enhance every table/tree so that a 2<sup>nd</sup> click on a cell in a row that is already selected will result in a popup of the cell, preselected so you can copy-to-clipboard:

11418	Implement RP 5027 message ...	Scheduled	DPANAHY	IET	Enhancement
11417	Implement RP 5027 message	Scheduled	DPANAHY	IET	
11409	VerifIEr 8.7	In Progress	DPANAHY	IET	
11408	genIE functions for Studio Developer		DPANAHY	IET	Responsibi
11406	Rapide GuardIEEn generation 8...				
11405	Studio Developer 8.7				
11404	Split PStep include PKG Defau...				
11403	Split PStep include PKG Defau...				
11401	R87 Sync I_ with Studio				
11399	Release 8.7 Key and Dialog V...				
11397	Release 8.7 Setup	Scheduled	DPANAHY	IET	
11396	Web CR 8.6.5	Scheduled	DPANAHY	IET	

While the cell popup value is visible, nothing else can be done in the application. Clicking outside the value, or pressing Escape will remove it and return to normal operation again.

Individual listboxes may override the token with a new property in the Designer.

## Block Mode Transformation – custom video properties

A new option for Screen to GUI Transformation and re-Transformation enables you to set a GUI control's custom foreground colour based off the screen field's video property, either as a custom video property or custom theme name.

## Block Mode Transformation – custom edit patterns

If a screen field uses a default edit pattern but the field length specified in the screen design is not the same as the attribute length, i.e. the screen design truncates the attribute, on transformation to a GUI widget, a custom edit pattern is defined with the same length as the screen field length. This only applies to text and numeric fields.

## Provide override auto-fit column width option for individual list boxes

The existing rapide.properties LISTBOX\_AUTO\_WIDTH token affects all horizontally-scrolling listboxes, causing the columns to be proportionally resized in width to fit the available width of the listbox. A new property is provided on individual listbox controls to allow override of this aspect.

## Auto-Height/Width for containers in Browser

It is expected that most applications in BROWSER targets will design windows to specific sizes which will either fit full-screen into the browser, or apply minimum design widths/heights.

However, in traditional web sites, the entire page will often be scrolling (particularly vertically) adjusting automatically to the content of the page. For example, a long list of results will cause the page to be vertically much taller, and therefore the browser's native scrolling is provided.

A similar effect can now be achieved with Rapide applications, but relies on any control with a potentially varying height to auto-adjust itself according to its' content. For example, listboxes or fields with markup. For any such control which may vary in size, the runtime can be left to automatically adjust its' size either horizontally, vertically, or in both directions. The parent container of the control should then also do the same, such as a Frame, or the Window/Dialog box itself. Properties are provided on almost every type of control to allow this.

In addition, the application needs to instruct the browser that it should use native browser scrolling. This is controlled at assemble time by the Rapide assembly process scanning the rapide.properties file for a token called PAGE\_OVERFLOW, and values scrollX, scrollY, or scrollB.

## **Auto-Tab in overlay field of list boxes**

---

A new `rapide.properties` token `AUTOTAB_IN_CELLS` is provided which enables AutoTabbing out of overlay fields in listboxes when the maximum number of characters have been typed into them.

Default behaviour is to NOT enable AutoTab on overlay fields. Set to True to enable it.

## **Auto-Check in Auto-New-Row Listboxes to add new row**

---

With auto-new-row listboxes, new rows are normally only added when some actual data is typed into an overlay field and focus moves out of that field. However, with a new `rapide.properties` token `AUTO_CHECK_AUTO_ADDS_ROW=True`, new rows are also added when just the auto-check checkbox is switched on.

## **Permitted value drop-downs implementation as read-only fields or auto-suggest**

---

A Permitted Value drop-down control may now be implemented as a read-only field (with or without border), or an auto-suggest field. The displayed/entered values will be the prompts (with translation as appropriate), but the values in the view will be the actual permitted values.

## **New API methods for processing child psteps and trancodes**

---

Added new methods to the Rapide Function Library:

- `PStepInstanceList = getChildInstancesForPStep(ParentPStepInstance)`
- `Trancode = getTrancodeForInstanceCLRSCRN(PStepInstance)`
- `Trancode = getTrancodeForInstanceDLGFLOW(PStepInstance)`

## **New API methods to retrieve application assembly time**

---

A new file called `assemble_timestamp.properties` is now created by the assembly processes and placed in the root of the application. The Rapide runtime parses this file if it exists to extract the assemble timestamp, which can then be retrieved by application logic using a new Rapide Function Library property `assembleTimestamp`.

## **Change tracking control programmatically**

---

Where EDGE tracking is defined for a control which is relative to another control, you may now change which control it is relative to on-the-fly with a new API method `trackChangeRelativeControl`. This is useful if you have a number of 'stacked' controls which are not always visible.

## **New Position-Tracking option to track relative to centre of window**

---

When specifying position-tracking of controls, you can now specify that a control is tracked relative to the centre of the window either horizontally, vertically, or both. This is particularly useful for groups of controls whose overall position needs to stay centred in the window, rather than each control being exactly in the centre (though that can also be achieved with this feature). The new option is specified in the Designer by setting its' relative `Horz` or `Vert` as the window centre.

## **Disabled overlay controls able to be unprotected by MAKE**

---

Previously, if an overlay control on a listbox was designed as 'disabled', it always stayed disabled, even if a `MAKE UNPROTECTED` statement was executed on it. This is now enhanced so that certain types of overlay controls may be unprotected by a `MAKE` statement, including: fields, checkboxes, pv drop-downs, date pickers and time pickers.

## **Overlay HTML properly configurable for Checkboxes**

---

A new `rapide.properties` token, `CHECKBOX_OVERFLOW_CLIP`, controls whether the HTML Overflow property is applied to checkbox controls to enable the choice between clipping if the checkbox's design size is too small to contain the full text of its' prompt. Default is `False`, which means that the text will NOT be clipped.