



# TAP's 3270 to Web Conversion Project

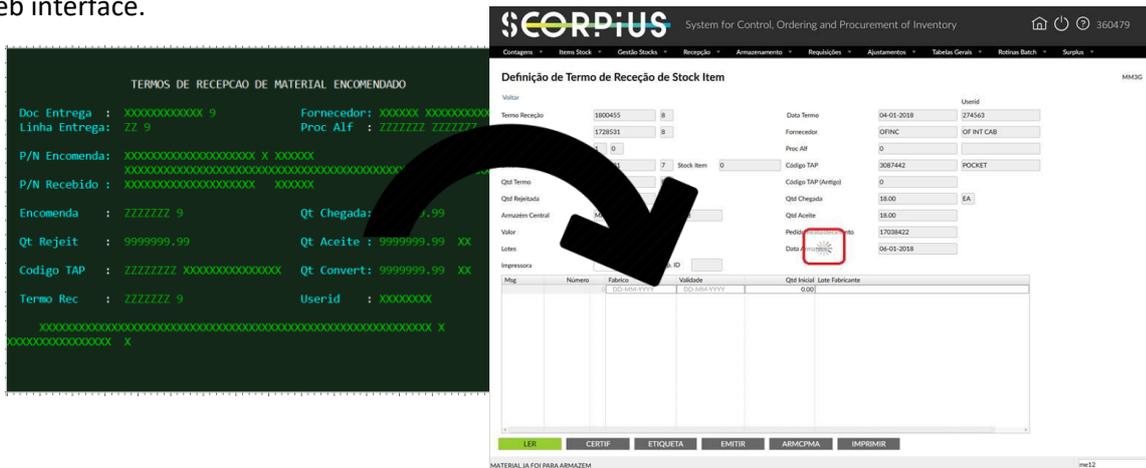
TAP successfully converted their 3270 block mode application to a web interface as part of a phased migration of the application from the mainframe to a distributed platform. Using highly automated conversion tools supplied with IET's Rapide product, TAP were able to complete the project and implement a significantly enhanced user interface.

## Introduction

TAP is Portugal's national airline, and TAP IT is TAP's internal suite of systems for managing all aspects of aircraft maintenance, developed over many years using CA Gen with the target as a 3270 Block Mode front-end on z/OS mainframe.



As part of a modernisation initiative, the TAP IT team was tasked with moving the front-end of the application away from the mainframe, onto a distributed architecture and giving it a fresh new look with a web interface.



## Objectives

TAP IT is a very large system, covering numerous aspects of TAP's business, so leveraging the vast effort put into developing it in CA Gen, rather than rewriting in a new technology was key.

Some other systems in the organisation had already been rewritten using another web development tool, which worked well, and looked slick and so the TAP IT team were tasked with producing a similar user experience for the new web application. However, the rewrites had proved costly in terms of the effort required to do the conversion, so mirroring the look-and-feel achieved in the other applications was important, but with less effort and maximum automation of the conversion process.

## The Journey

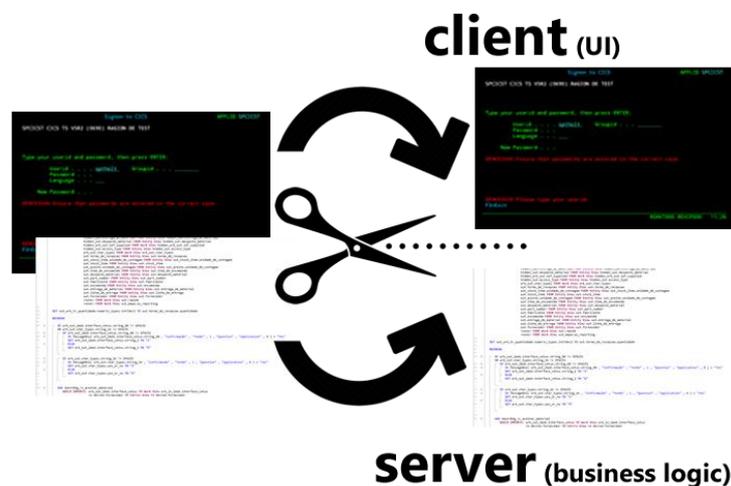
IET's Rapide product was one of the possibilities considered. Rapide is essentially a new generation option for CA Gen, but with a wide array of automation facilities to help convert Block Mode designs to web, as well as providing many modern UI options, from fresh new controls, to some powerful layouting and nesting options.



Rapide would also allow the team to retain and reuse much of the business logic in the CA Gen models, so TAP worked with IET to evaluate whether their Rapide solution would meet the objectives.

## Conversion of Flat Online Transactions to Client/Server

The first step was to convert the flat Block Mode transactions (Gen Procedure Steps, or "PSteps"), to a client/server structure, placing the business logic on the server-side, and separating the UI (still as Block Mode screens) to new client PSteps. This was important because the initial scope of the project was to retain the business logic on z/OS so that the existing DB2 database and batch suite would remain unchanged.



Part of Rapide's suite of tools is designed exactly for this purpose – you choose one or more Block Mode PSteps, provide some parameters for the conversion, and it automatically creates a client-side PStep to match each one and transfers the UI out of the existing PStep, leaving just the business logic behind – essentially performing everything required to achieve the split.

CA Gen provides a number of features that work well with a true Block Mode design, but which don't translate to client/server, so the team worked with IET to overcome these challenges...

For example, MAKE statements are used to colour the displayed attributes on a screen, but of course once the PStep has been separated into client and server PSteps, the MAKE statements will either need to be performed directly by the client PStep, or you lose the MAKE data. Rapide offers a solution to this which transfers the MAKE data as part of the client/server flow and automatically applies them on the client, even though the statements themselves were executed on the server.

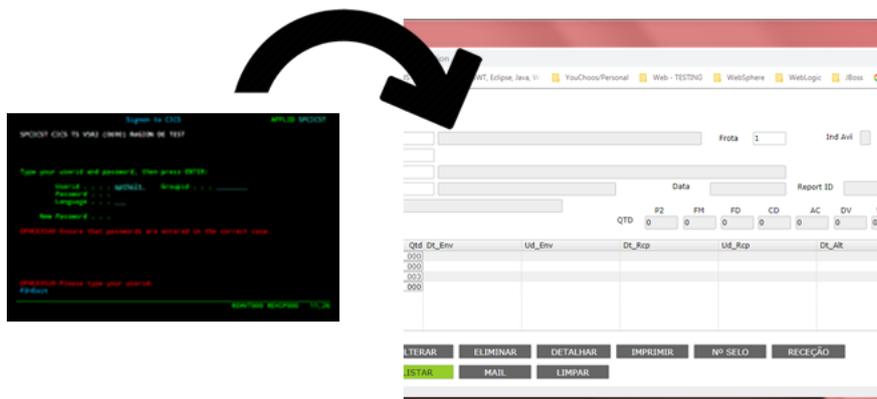
This facility is fully automated using source-code post-processing, and as long the size of the resulting export views fit within CA Gen's communications limitations, then there is no manual work to do at all.

TAP found that this ability worked well, with just a handful of PSteps needing re-work due to the view size.

Similarly, the application relied upon NEXTTRAN support in Gen, which provides the user a way to flow directly to a specific transaction, but CA Gen only provides this for Block Mode, not for client/server. Like the MAKE solution, Rapide automates the capture of a NEXTTRAN request on the server-side, transfers the request to the client, and automates processing the flow on the client-side so the application essentially operates in the same way, with no manual coding changes.

## Conversion of Block Mode to GUI

Now with the model converted to separate client and server PSteps, the next phase was to convert the Block Mode screen designs to GUI window designs.



Just like there are aspects of Block Mode logic that are not available for standard CA Gen client/server, there are many aspects of a Block Mode screen design that are not supported by a standard CA Gen GUI. For example, screen templates, command line and special fields, list selection columns, function key handling, scrolling and enterable lists.

Ideally, TAP wanted to redesign the UI to fit GUI concepts better, but that would mean a huge exercise and a lot of manual effort to change the models. Instead, they looked at Rapide's runtime abilities that give Block Mode-like features, but running with proper GUI controls. None of this would have been possible in a standard CA Gen GUI application.

Rapide offers two options for Block Mode to GUI conversion – the first keeps the designs in the model as screens, converting to GUI as a generation and runtime target, which has the benefit that the model does not change, and developers can continue to work with screens in the same way they have done for years. The second option automatically generates new GUI window designs based on the screens, which can then be enhanced and managed in the model as GUI. The project initially used the first option while prototyping, then went for the second option for the actual conversion, as they wanted to take advantage of Rapide's advanced UI capabilities and new control types.

The transition was seamless, and delivered a working application in GUI very quickly, albeit with the same structure and flow as the Block Mode design.

## Enhancing the GUI

Of course, in order to start using some of the more advanced controls and layouts, the team had to modify some of the GUI designs. The objectives here were to structure the layout to use nested frames and tab-like areas, to add corporate themes and styles, and to mimic the overall presentation that had achieved with other web applications and which had become a new corporate standard.



### Frames

One key feature was Rapide's *frame* capabilities, which allows a child window to be loaded into a designated area in a parent window. Different child windows can be loaded dependent upon user action, and this was found to be to be extremely powerful and flexible.



*Frames* were also used to apply a standard header and footer to every page, with a corporate banner/menu always visible, then different PSteps loaded into a child frame. In fact, the approach taken involved multiple nesting of frames within frames!

### Tracking & Responsive Layouts

For windows that contained list boxes, it was also decided to make use of Rapide's tracking/resizing facilities, so that the controls fill the space available in the browser. For example, a list box would resize according to the space available, and any controls below (or to the right of) the list box would move relative to the edge of the browser. This was a manual process, applying tracking to each window using Rapide's tools, but quite easy to do – a visual tool is provided to make this intuitive.



### Editable Cells

Since CA Gen's Block Mode screen designs allow editing of values in a repeating group view (table of data), but Gen's standard window design does not give cell editing, the new GUI designs also took advantage of Rapide's *cell overlay*

AC	Div#	Dept#	Department Name	Emp#
▼	1	1	ONE	1
▼	1	2	TWO	2
▼	1	3	THREE	3
▼	4	4	FOUR	5
▼	0	0		0

features, which were applied automatically as part of the Block Mode-to-GUI conversion. There was very little change needed to the designs to make this work, and IET provided various enhancements to support and improve these capabilities.

### Processes & Managing Change

The Block Mode models are managed using IET's GuardIEn change management tools, which includes automated impact analysis and implementation. Using Rapide fits in flawlessly to the process – the team simply had to configure GuardIEn to generate, build and assemble a Rapide application instead of Block Mode artefacts.



### Testing

For unit testing, the developers sometimes used workstation-based Rapide generation from the CA Gen Toolset, and Rapide's Toolset plug-ins to perform build & assemble, then run the application locally on a built-in application server provided with Rapide. This configuration is simple, since the internal application

server is light-weight and requires no configuration, allowing the developer to perform iterative builds and tests quickly without having to deploy the application to a more traditional application server.

From the CA Gen Encyclopaedia, TAP used GuardIEn to assemble the Rapide application for IBM WebSphere. This application server is effectively the client part, in TAP's case they use a variety of server types including communications through Tuxedo to CA Gen C servers on UNIX using a standard CA Gen Transaction Enabler, but the majority of the servers reside in CICS on a z/OS mainframe. Any standard CA Gen server technology would have been possible.

### Performance & Scalability

A Rapide web application is the client-side PSteps running in an application server, so the JVM of the application server must be configured with sufficient resources to support the required number of users who may be connected at any one time. Each connecting browser uses a light-weight JavaScript client which renders the UI, synchronising it with the client state on the application server through JSON messaging.



TAP employ a number of WebSphere application servers, clustered using load-balancing, and overall the application responds well, with only occasional delays while the browser and application server sync their UI, which is mostly down to network latency rather than application server performance.

### Conclusions

The success of the transition was achieved by a combination of highly automated processes to change the CA Gen models, and capabilities in the Rapide runtime to support Block Mode-style features.

Most of the time invested in the conversion related to making use of the enhanced UI features of Rapide, and the development team and end users are very pleased with the results. It is sometimes hard to believe that this modern look and feel comes from a CA Gen model!



Much of this technology was new when the project began, and IET worked closely with the team, providing fixes and enhancements throughout, often same day, which has enabled the team to keep the momentum going and gave a very positive experience.