



An Introduction to Renaissance



© CoralTree Systems Limited 2010.

CoralTree Systems Limited are not liable for any errors or inaccuracies in this manual. Any errors found should be reported to the authors for correction in future editions.

The authors and their employers disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the authors or their employers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Renaissance remains the property of CoralTree Systems Limited and can be used or copied only in accordance with the terms of the agreement. System i and iSeries are registered trademarks of International Business Machines Corporation.

Contents

Chapter 1: What is Renaissance?	4
Requirements	5
Other considerations	5
Chapter 2: Getting started	7
Navigating in Renaissance	7
Standard icons	7
Standard keystrokes.....	7
The Fastpath menu	8
Changing your password	8
Configuring Renaissance for development	8
Writing applications using the RNS framework	10
Chapter 3: Troubleshooting	11
Glossary	12
Appendix A: Useful RNS commands	13
Index	14

Chapter 1: What is Renaissance?

Renaissance is a Framework of RPG ILE programs, service programs, Javascript code and HTML designed to enable developers to quickly and easily build high-quality browser-based intranet applications for the IBM System i.

The Renaissance Framework provides:

- Full internationalisation capability.
- Full session management for maintaining state across multiple page requests.
- User security, with role-based authority to each function.
- The ability to define your own data tables and allow the end user to maintain data through functionally rich web pages without writing a single line of code.
- CSS based themes.

Applications built using the Renaissance Framework run on the System i Apache HTTP server, utilising the CGIDEV2 service procedures in order to deliver web-pages to a browser. Renaissance can optionally employ Ajax technology to provide the end user with smooth page transition and seamless server/browser interaction.

Renaissance is developed by CoralTree Systems Ltd, and freely available under the Open Source MPL Licence Version 1.1 (<http://www.mozilla.org/MPL/>).

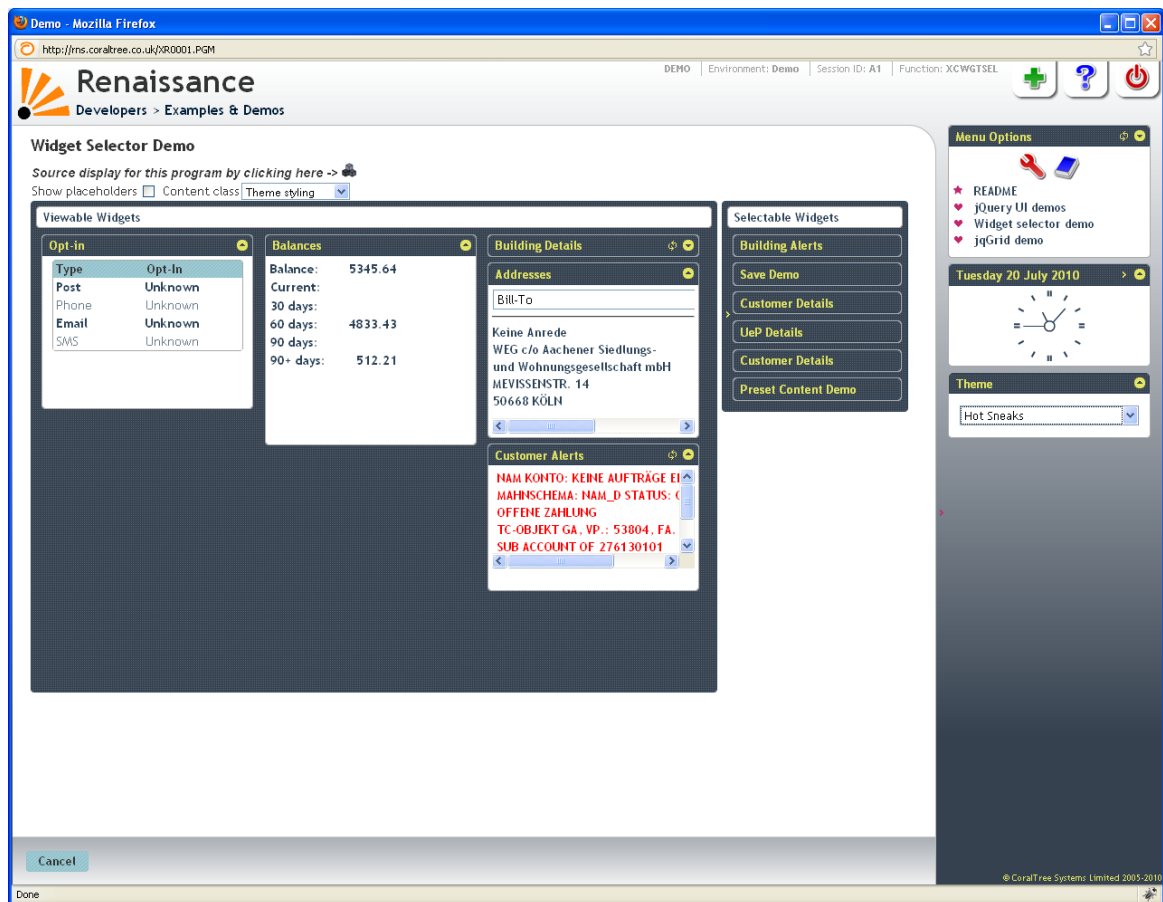
Dataqueue Name	Inbound Prefix	Outbound Prefix	Initial Nbr Of Jobs	Max Nbr Of Jobs	Job Queue	Time Out
*USER	IN	OUT	1	1	RNSDTA	*NOMAX
BATCH	IN	OUT	5	10	RNSDTA	*NOMAX
DEFAULT	IN	OUT	1	2	RNSDTA	*NOMAX
EMAIL	IN	OUT	1	1	RNSDTA	*NOMAX
INBOX	IN	OUT	1	5	RNSDTA	*NOMAX
LOGGING	IN	OUT	1	2	RNSDTA	*NOMAX
LOGIN	IN	OUT	5	10	RNSDTA	*NOMAX
PSTATS	IN	OUT	1	2	RNSDTA	*NOMAX
WSLOG	IN	OUT	1	2	RNSDTA	*NOMAX

All programs called via your web browser run on the System i and are written in ILE RPG IV using the CGI (Common Gateway Interface) standard - a cross-platform standard which establishes exactly how a program is invoked from a web browser. In this way

dynamic web pages are built and served to the browser by the Apache HTTP server hosted on the System i which is included with i5/OS. No additional hardware is required.

Renaissance makes extensive use of the CGIDEV2 development toolset, which is turn is based around a service program called CGISRVPGM2. This simplifies development by hiding the complexities of CGI Application APIs from the developer and allows HTML to be constructed dynamically from one or more pieces of template HTML on the IFS. The CGI development toolset is open-source and provided free-of-charge.

The CGI programs use data queues to send and receive requests to and from the System i server. The programs that actually retrieve and update data on the System i (commonly known as 'server programs') are invoked by data queue monitor jobs which wait for a request from a CGI program to be placed on an inbound data queue, perform the necessary database operations, and place the requested data on an outbound data queue. The CGI program waits for a response to be received on the outbound data queue, then displays the results to the user.



Requirements

To run applications using the Renaissance framework you will need:

- An System i running V5R3M0 or above.
- Apache running on your System i.

Other considerations

Before installing Renaissance on your System i, give consideration to the following:

Do you already have an Apache server listening on port 80 that you want to keep?




















If not, installing Renaissance using all the default values should present no real problem. However, if you do have an existing server listening on port 80 you will need to either specify a virtual host name during installation (having first ensured that your DNS recognises that name and points correctly to your System i), or choose a different port number.

Chapter 2: Getting started

Navigating in Renaissance

Throughout Renaissance a number of icons and keystrokes can be used to navigate and perform basic functions:

Standard icons

-  Logs you out of the application.
-  Closes the application.
-  Refreshes the current display.
-  Allows you to change your login password.
-  Prints the current display.
-  Accesses on-line help text for the page currently displayed.
-  Makes the current function the initial function run when you first log on to your application. To undo this change go to *User Preferences* and wipe out the default function name shown.
-  Hides the menu *or* Displays the previous page of data.
-  Redisplays the menu *or* Displays the next page of data.
-  Starts a new session, in a new window, without the need to log on again.
-  Allows you to sort the data currently displayed.
-  Allows you to display a subset of the data currently displayed.
-  Allows you to add a record.
-  Allows you to view a record.
-  Allows you to edit a record.
-  Allows you to copy a record.
-  Allows you to delete a record.
-  Allows you to select a date.
-  Clicking on this symbol against an error message will display that message in a pop-up window, along with any other current errors.

Standard keystrokes

(only available if running in AJAX mode)


- | | |
|---------|--|
| F5 | Refresh: refreshes the current display. |
| F6 | Create: allows you to add a record. |
| F12 | Cancel: returns you to the previous screen without updating. |
| F19 | Clears all subset options and refreshes the display. |
| Page Up | Displays the previous page of data. |

Page Down Displays the next page of data.

The Fastpath menu

Many Renaissance pages include a Fastpath menu, which is located in the top right-hand corner of the page and which allows you to switch directly to another function. The functions available will vary depending on which page you are accessing at the time but will typically include complaint logging, comment logging and customer maintenance. In each case select the function you require from the drop-down list to continue.

Changing your password

Using Renaissance, you can change the login password for the System i machine on which your application data resides. Simply click on the  icon, enter your old and new passwords, then click Change Password to update your login details.

Configuring Renaissance for development

Upon installing Renaissance you will have:

- A production database environment called RNSDATA.
- A production RNS framework library called RNS.
- A production application library called RNA.
- A single user called RNSADMIN carrying the ADMIN role.

Although you do not yet have anything in production it is likely that the first thing you will want to do is create a *development* environment. Although Renaissance has been written to interface directly with SEE/Change from Thenon for change control, there is no reason why you should use this for your own development environment. Like any System i application, Renaissance is driven by library lists stored in job descriptions, so you can configure it to use any library structure you wish. On top of this, you can also configure Renaissance to use development versions of IFS resources such as HTML templates, JavaScript, images, stylesheets, etc.

Put simply, you can set things up to suit your own environment so the following merely describes one way you might choose to work.

At CoralTree we store the production versions of our applications in the RNA library while leaving the RNS library to store the framework only. Using SEE/Change means we create change requests (which are actually just libraries for one or more developers to work in as they develop new features) with each developer configuring Renaissance to use the appropriate change request while they are working on, and testing new features. Even without having SEE/Change installed on the System i you can create your own change request libraries and configure Renaissance to use them while you are developing.

In the following example, let us assume that you require the kind of development environment described above, and that your production application is to 'live' in RNA, with your application data stored in RNADATA. We will also assume that, while testing these programs, there is to be a framework test environment called RNSTSTDTA.

1. *Create a library called RNADATA for your application data.*
2. *Create the test framework environment.*

- Ensure that your library list includes QTEMP, RNSWORK, RNS and QGPL then enter XACRTENV RNSTSTDTA at a command line.
- Within RNSTSTDTA you should find a job description called RNSJOB. Edit this job description and change the library list so that it looks as follows:

```
QTEMP
RNSWORK
RNSTSTDTA
RNADATA
RNS
RNA
QGPL
```

- Now, when returning to Renaissance and logging on as RNSADMIN you will be presented with two environments to choose from. Choose the one you have just created. Once logged in, click on Users and copy (not create) RNSADMIN to your normal System i user ID. This will allow you to log in as yourself.
 - Next, expand the *Cross-Application* menu node.
 - Next, expand the *Configuration* menu node.
 - Next, expand the *Config maintenance* menu node.
 - Next, click on the *Environments* menu option.
 - Change the name of the environment to *Development* and change the *Mode* to *Development*.
 - Change the *Job description for the server layer* to be *RNSTSTDTA/RNSJOB*. You may also wish to change the *Path for logo* to *images/development.gif* so that you can tell easily that you are running a development version of RNS rather than a production version. You may also want to change the theme, too.
 - Click *Save*.
 - Log off, then log back on again.
 - The information bar at the top of the page should change to show your server details, etc.
3. *Create a change request or development library.*
- Create a library in which to carry out development. Assume you have called it MYDEVLIB.
 - In MYDEVLIB, create all the source files you will need.
 - Next, create a job description called MYDEVLIB/CRJOB.
 - Ensure that the job description has a library list that reflects your development environment. Typically it will look something like this:
 - QTEMP
MYDEVLIB
RNSWORK
RNSTSTDTA
RNADATA
RNS
RNA
QGPL
4. *Configure Renaissance to use your development library.*

- Expand the *Cross-Application* menu node.
- Next, expand the *Configuration* menu node.
- Next, expand the *Config maintenance* menu node.
- Next, click on the *Developer Preferences* menu option.
- You should see a record representing your user ID (if not, create it by clicking the *Create* button or pressing F6).
- Make sure that your developer preferences contain a *Development library* of MYDEVLIB and that the *Reclaim activation group before each request* and *Ajax* boxes are both ticked. This will ensure that you can see the results of your changes and compiles instantly without having to restart the Apache server each time, and will also ensure that you use Ajax for your requests. End users should always use Ajax; developers can switch Ajax off if they wish to view the HTML their programs have issued to the browser (this is not possible with Ajax switched on).
- Log off and back on again. Provided you have set up the library list correctly you should now see that the information bar at the top of the page shows your change request (C/R) as *MYDEVLIB*.
- You can now develop CGI and back-end server programs in MYDEVLIB and test them without impacting on your production versions.

Writing applications using the RNS framework

When you first install Renaissance, only the layout of the screen, menus and colour schemes can be configured to suit your own preferences. However, you should find that writing CGI programs of your own, to run over the RNS framework, is fairly straightforward. Applications can be written in any language you like provided they employ the correct service procedure calls to populate the main work area.

Sample programs (XCSIMPLE and XCDEMO) are provided to further illustrate the techniques available to programmers and are a good source of sample code for anyone intending to build Renaissance applications.



Bear in mind that any new programs you write will not be visible in the menu by default. They will need to be registered in the Functions section of RNS AND added to one of the Roles that a user is likely to have associated with their user ID.

For further information, refer to the document labelled *Renaissance - A Guide For Developers* in the *rns/htdocs/documentation* folder. You can also exchange tips and ideas with other RNS developers by subscribing to our forum at <http://www.coraltree.co.uk/phpbb/>.

Chapter 3: Troubleshooting

Error: Fatal error occurred while sending a request to the back-end server.

This error will occur if the Renaissance back-end server failed to start for some reason. To fix this problem ensure subsystem QGPL/RNS is active, then manually start the back-end by entering `RNSSTART xxxxxxxxx` at a command line (where xxxxxxxxx is your data environment, normally probably RNSDATA). You should then have the following jobs running:

- An RNS environment management job (normally called RNSDATA).
- A SESSIONMGR job, which deals with expired sessions, etc.
- A LOGGING job (a server handling all logging for diagnostics and performance analysis).
- A server for your own use (normally just RNSADMIN), and one for each additional user.

If all the above jobs are running and you continue to receive this error please examine the job log and send the appropriate entries to CoralTree for further investigation.

Error: You are not enrolled in any application environments on this server

Ensure that you are logging in using the correct user ID and password. The first time you log on to Renaissance you must use the user ID and password RNSADMIN. Once you have done this you can change your password and create your own user profile (and any others you require) in the system.

Error: You already have the maximum permitted application sessions open

The number of active sessions for your user ID exceeds the 'maximum sessions' value (XUSMAXSESS) for your user ID in the XUSER file in this environment. Ensure that you have no other open sessions or increase the 'maximum sessions' value.

Error: The server is not responding to your login request – contact support

Either:

- The queue manager timed out whilst waiting for a response from the server, i.e. for an entry on the outbound data queue, possibly because there is no queue manager job running for the data queue / environment or because the queue manager job has crashed.

Or

- The call to the server program from the queue manager (XS0001) ended in error.

Error: The application environment is temporarily unavailable

The environment manager for this environment is not running, and the 'auto start' configuration parameter (ENV/GLOBAL/AUTOSTART) is not set to '1'. To check whether or not the environment manager is running, Renaissance checks the XSMANAGERA data area in your environment database library. If positions 1-10 of this data area contain '*END' this means the environment has ended so it will need to be restarted manually.

Glossary

CGI

Common Gateway Interface. A standard for cross-application programming which determines how programs running on a server are invoked from a web browser.

CGI program

In Renaissance terms, a program called from a web browser but run on the Apache HTTP server hosted on an System i. There is generally one CGI program for each service program.

Data queues

The mechanism by which incoming requests are supplied to CGI programs, and by which the results are served to the web browser. Inbound data queues handle requests sent to the System i, outbound data queues handle the results returned to the browser.

Environment

A single instance of Renaissance. Each environment has its own database library with which it shares a name.

Service procedure

A component of a service program, designed to perform a specific function. For example, the XUSYSVAL service program, which deals with system parameters, contains the RtvSysVal service procedure for retrieving a system value.

Service program

A service program is invoked by an inbound data queue monitor job in order to perform database retrieval and update and place the results on an outbound data queue. There is generally one service program for each CGI program. Each service program may comprise a number of service procedures, each performing a specific function.

Session

An instance of a user logging on to a Renaissance environment. A session can be thought of as similar to an interactive job in a green-screen application.

Appendix A: Useful RNS commands

RNS *E <server>	Ends an RNS Apache server.
RNS *S <server>	Starts an Apache server.
RNS *R <server>	Restarts an RNS Apache server.
RNS *F <server>	Forces activation group reclaim for an RNS Apache server.
RNS *E *ALL	Ends all RNS Apache servers.
RNS *S *ALL	Starts all RNS Apache servers.
RNS *R *ALL	Restarts all RNS Apache servers.
RNS *F *ALL	Forces activation group reclaim for all RNS Apache servers. This can be used to 'force' servers to use the new versions of programs without the need to stop and restart them.
RNSSTART <environment>	Starts a back-end server.
RNSEND <environment>	Ends a back-end server.
RNSRESTART <environment>	Restarts a back-end server.
RNSSTART *ALL	Starts all back-end servers.
RNSEND *ALL	Ends all back-end servers.
RNSRESTART *ALL	Restarts all back-end servers.
RNSSHUTDOWN	Shuts down all RNS Apache and back-end servers.
RNSSTARTUP	Starts all RNS Apache and back-end servers.
BLDRNS	Builds Renaissance
CFG_IMPORT	Imports configuration file from IFS
CPYAUTSTR	Copies framework authority structure
CPYCFGSTR	Copies framework configuration structure
CPYXPROMPT	Copies XPROMPT and related data
CRTCGIDEV2	Recreates service program RNS/CGISRVPGM2
INSTALL	Installs Renaissance
RNSCRTSVR	Creates the Renaissance Apache Server
RNSSAVF	Creates an RNS save file
RPDISABLE	Enables the request processor
RPENABLE	Disables the request processor
RPEND	Ends the request processor
RPRESTART	Restarts the request processor
RPSTART	Starts the request processor
TESTLOGIN	Performs a test log-in
UPGRADE	Installs a Renaissance upgrade
XACRTENV	Creates an RNS environment library
XAQUEUES	Ensures all server queues are defined and intact

Index

Applications, 9

CGI, 11

Data queues, 11

Environment, 11

Fastpath, 7

Glossary, 11

Icons, 6

Internationalisation, 4

Introduction, 4

Keystrokes, 6

Navigating, 6

Password, 7

RtvSysVal, 11

Service programs, 11

Session, 11

Session management, 4

Themes, 4

Troubleshooting, 10