# APPENDIX M – THE FASTFIX PROTOTYPE SHELL

This Appendix presents the context, construction and layout of the web application shell of the FastFIX prototype, including the User Security model and the FastFIX prototype GUI.

Some of the auxiliary features provided by the FastFIX prototype shell are also presented, including the features pertaining to the following FastFIX menu items:

- Check RuleNodes
- Check My Cases
- Check All Cases
- Write Rule Tree
- Solutions Preview
- Upload Files
- View Uploaded Files
- Test Data Integrity
- Test Rule Evaluation
- Test Rule Parsing

## *M.1 Introducing FastFIX*

I commenced (in July 2003) by designing and implementing a very general system to help trouble-shooters in any problem domain, solve any type of problem. I named the prototype system *FastFIX*.

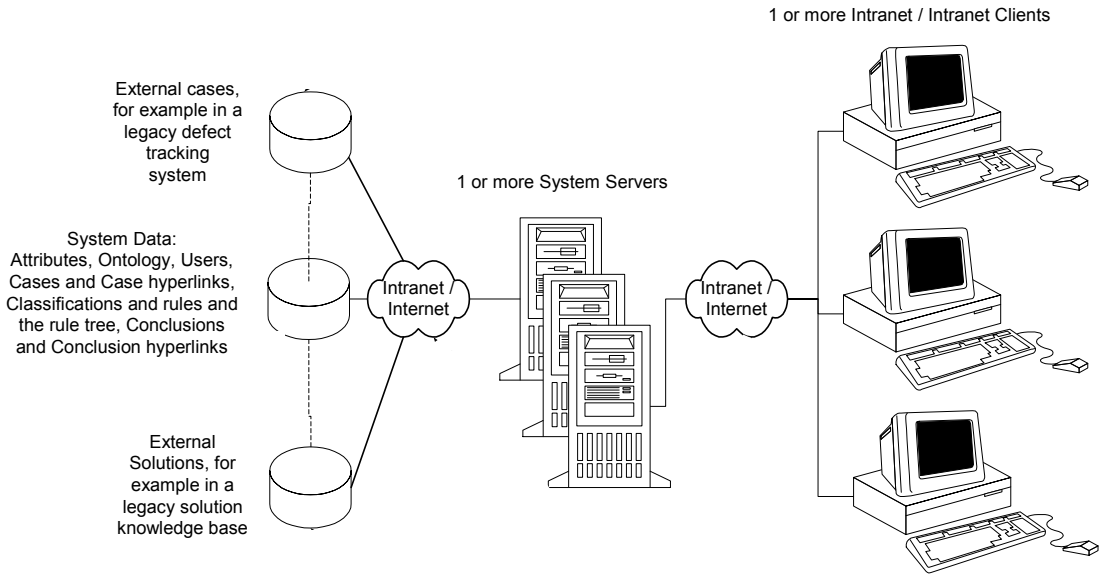
Unsure of the final target audience and the type of platforms that they would be operating on (Linux / Windows / Unix / Apple Macintosh) I built a web-based multi-user client-server expert systems shell. At that time, there was still significant unreliability in client-server java deployment through any version web browser and on any given platform, so without control over the client web-browsing environment I opted to implement the majority of my code in PHP on an Apache web server. My PHP server-side code served up client-side HTML and Javascript. One limitation of this technique is that web content is only changed at the client's request (i.e. it's a pull-model) rather than at the will of the server (i.e. push-model).

The prototype was built from scratch during the course of this PhD. With all development conceived of and performed by the researcher, time was very much of the essence, and demonstrating possible features was much more of a priority than building a robust and scalable application framework. Saying that, the prototype had no problems with robustness or scalability. It has been in live operation at a publicly accessible but password protected web-hosted Internet site for more than 6 months without any performance issues and with unbroken up-time.

Without any funding for a transaction-based commercial database, and given that the project was experimental in nature, I used opensource MySQL[236] version 4.0.14 as the data store. Transactions and rollback were not possible in this version of MySQL, however more recent versions now have this functionality. In a commercial implementation, correctly managing concurrency would be critical for a multi-user and index-intensive system such as FastFIX. Figure 102 shows the top-level architecture of my prototype FastFIX system.

[236] http://dev.mysql.com/

*Figure 102: The FastFIX Top-Level Architecture*




*Multiple intranet clients can access the FastFIX system via their web browsers. PHP[237] server side code serves up HTML and Javascript to run in the client's browsers. The condition mesh is stored in a MySQL system and provides HTTP Internet hyperlinks to Cases in a legacy defect tracking database, and Solutions in a legacy knowledge base.*

The FastFIX prototype allowed users to record, retrieve, review, refine and rate trouble-shooting knowledge in the context of specific problem classes stored as cases in CaseDB, and using existing solutions stored in SolutionDB. Each RuleNode offered a set of zero or more intra- or inter-net hyperlink references, where each reference pointed to web content that could assist with trouble-shooting the current hyperlinked CaseDB case.

## M.2 Account Management

In the FastFIX prototype, a user security model was implemented so that users would register and login once, after which their FastFIX session would persist via cookies until such time that they purposefully logged out (which could be months later). This reduced the login effort

[237] PHP is a server-side internet Hypertext Pre-Processing scripting language that is used to build web-sites on the Internet. Details are available at: http://www.php.net.

required by users, yet enabled FastFIX to effectively track user activity in the knowledge base. In the FastFIX prototype, passwords are MD5 encrypted on the server side and stored in the MySQL database.

## M.3 Login

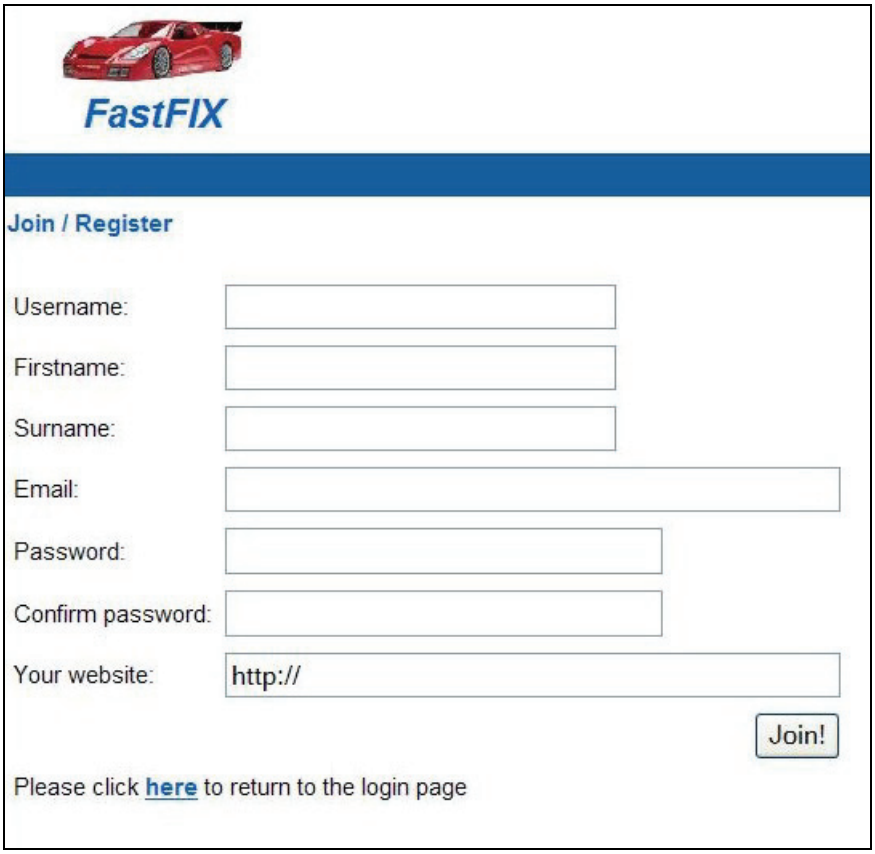
The following screenshot shows the login screen presented to all users.

*Figure 103: Login*

## M.4 Join / Register

For new users, the following screenshot shows the join / register screen.

*Figure 104: Join / Register*




## M.5 Password Reminder

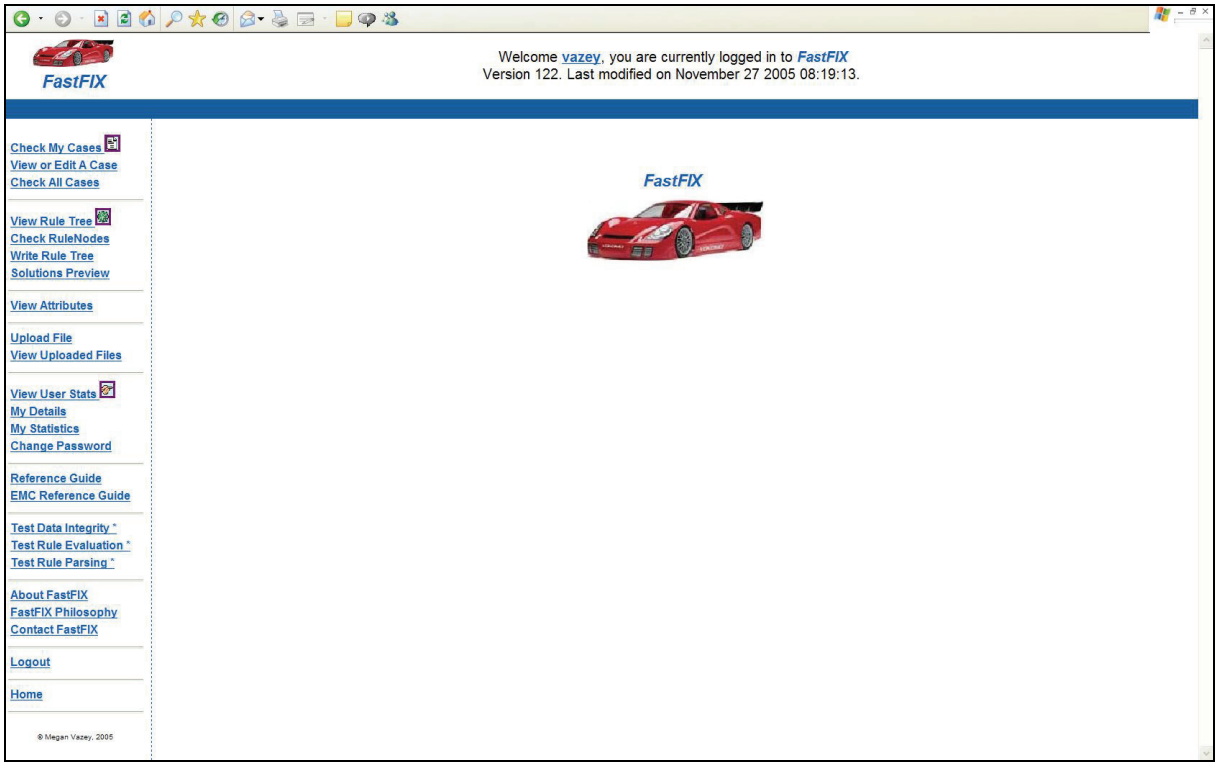
For repeat-entry users who have forgotten their password, the following screenshot shows the password reminder screen. Selecting the "Get Password" button causes the user to be emailed a new machine generated password that the user can subsequently modify.

*Figure 105: Password Reminder*


## M.6 Home Page

Once logged in, users are presented with the following screenshot that shows the layout of the prototype FastFIX GUI, including the menu options available to users at the developer level.

*Figure 106: Home page of the prototype FastFIX GUI*



Screenshots included in the remainder of this chapter and in the following chapter are located in the centre panel of the above FastFIX GUI layout, replacing the larger of the two red Ferraris.

## *M.7 Menu Items*

The following table shows the page on which a discussion of the indicated menu item can be found. No further information is provided for the *About FastFIX*, *FastFIX Philosophy*, and *Contact FastFIX* pages since the intentions of those pages are self-evident.

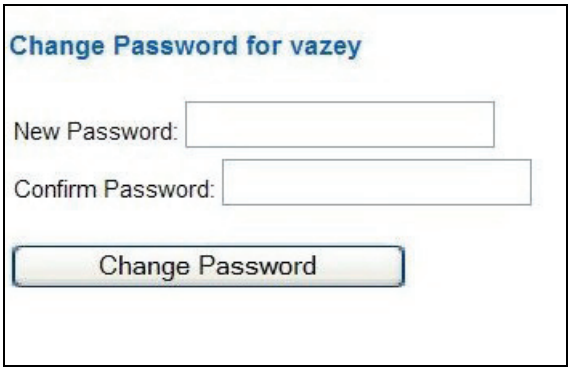
*Table 32: FastFIX Menu Items*

| | **FastFIX Menu Item** | Section | Page |
|---|---|---|---|
| **Cases** | Check My Cases | M.12 | 435 |
| | View or Edit A Case | 11.5.1 | 202 |
| | Check All Cases | M.11 | 434 |
| **RuleNodes** | View Rule Tree | 11.3.3 | 186 |
| | Check RuleNodes | M.10 | 433 |
| | Write Rule Tree | M.13 | 435 |
| | Solutions Preview | M.14 | 436 |
| **Attributes** | View Attributes | 11.4.2 | 198 |
| **Files** | Upload File | M.15 | 437 |
| | View Uploaded Files | | |
| **Users** | View User Stats | 11.2.3 | 182 |
| | My Details | 11.2.1 | 179 |
| | My Statistics | 11.2.2 | 179 |
| | Change Password | M.8 | 431 |
| **Tests** | Test Data Integrity | M.16 | 438 |
| | Test Rule Evaluation | | |
| | Test Rule Parsing | | |
| **Security** | Logout | M.9 | 432 |

## *M.8 Password Changes*

As indicated by the "Change Password" menu option, once logged in, users are able to change their password as shown in the following screenshot.

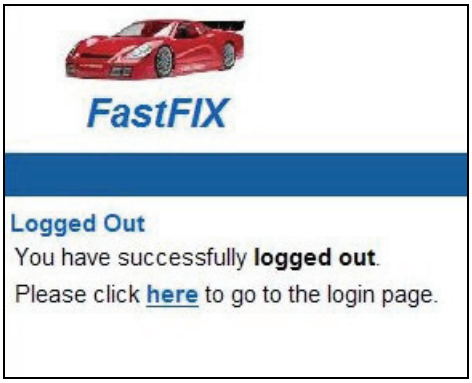
*Figure 107: Change Password*




## M.9 Logout

As indicated by the "Logout" menu option, once logged in, users may log out at any time as shown in the following screenshot.

*Figure 108: Log Out*



## M.10 "Check RuleNodes"

The following screenshot shows a view in the prototype FastFIX system that can be used to check the current status of RuleNodes in the system.

*Figure 109: The "Check RuleNodes" View*

Check RuleNodes:

Table of RuleNodes

| RuleNode | ParentNode | Registered Cases | Live Cases | Dependent Cases | Comparison of Registered vs Live Cases |
|---|---|---|---|---|---|
| 1 | none | none | 71, 103, 138, 144 | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172 | DIFFERENT - ROOT NODE |
| 2 | 1 | none | 10, 11, 12 | 10, 11, 12 | DIFFERENT |
| 3 | 1 | none | 10, 11, 12, 125, 127, 128, 129, 130, 131, 133 | 10, 11, 12, 125, 127, 128, 129, 130, 131, 133 | DIFFERENT |
| 4 | 1 | none | 3, 5, 6, 10, 11, 16, 125, 127, 128, 129, 130, 131, 133, 135 | 3, 5, 6, 10, 11, 16, 125, 127, 128, 129, 130, 131, 133, 135 | DIFFERENT |
| 5 | 1 | none | 5, 6, 10, 11, 12, 14, 16, 106, 125, 127, 128, 129, 130, 131, 133, 135 | 5, 6, 10, 11, 12, 14, 16, 106, 125, 127, 128, 129, 130, 131, 133, 135 | DIFFERENT |
| 6 | 1 | none | 5, 6, 125, 127, 128, 129, 130, 131, 133 | 5, 6, 125, 127, 128, 129, 130, 131, 133 | DIFFERENT |
| 7 | 1 | none | 1 | 1 | DIFFERENT |
| 8 | 1 | 57, 63, 74, 93, 164 | 2, 3, 48, 57, 63, 93, 107, 164, 166, 168 | 2, 3, 4, 8, 9, 12, 15, 19, 21, 22, 26, 27, 29, 31, 33, 35, 37, 39, 42, 43, 44, 46, 47, 48, 49, 50, 56, 57, 62, 63, 64, 67, 68, 73, 74, 75, 77, 81, 82, 83, 85, 86, 87, 92, 93, 94, 95, 96, 101, 105, 106, 107, 114, 136, 140, 141, 143, 145, 155, 156, 157, 159, 163, 164, 165, 166, 167, 168, 170, 171 | DIFFERENT |
| 9 | 8 | none | none | none | SAME |
| 10 | 8 | none | 167 | 167 | DIFFERENT |
| 11 | 8 | 19, 39, 42, 44, 49, 56, 95, 101, 145 | 19, 39, 44, 49, 56, 95, 101, 145, 171 | 19, 39, 42, 44, 49, 56, 83, 95, 101, 114, 145, 170, 171 | DIFFERENT |
| 12 | 8 | 8, 21, 22, 43, 50, 73, 82, 157 | 8, 12, 21, 22, 39, 43, 50, 73, 82, 155, 157 | 8, 12, 21, 22, 39, 43, 50, 73, 82, 155, 157 | DIFFERENT |
| 13 | 8 | none | none | none | SAME |
| 14 | 8 | none | none | none | SAME |
| 15 | 8 | 35, 42, 67, 77, 86, 96 | 4, 35, 42, 67, 77, 86, 96, 136, 141, 156 | 4, 35, 42, 67, 77, 86, 96, 136, 141, 156 | DIFFERENT |
| 16 | 8 | 15, 27, 29, 85, 105, 143, 159 | 15, 27, 29, 64, 68, 85, 105, 143, 159 | 15, 27, 29, 64, 68, 85, 105, 143, 159 | DIFFERENT |
| 17 | 8 | none | none | none | SAME |
| 18 | 8 | none | none | none | SAME |
| 19 | 8 | 114 | 83, 114 | 83, 114 | DIFFERENT |
| 20 | 8 | none | 9, 74, 92 | 9, 74, 92 | DIFFERENT |
| 21 | 8 | none | 4, 9, 26, 31, 47, 74, 92 | 4, 9, 26, 31, 47, 74, 92 | DIFFERENT |
| 22 | 8 | none | 31, 92 | 31, 92 | DIFFERENT |

The lists of live versus registered cases for RuleNodes in the system are compared and excluding any live and unregistered stopping RuleNodes, if there are differences in these lists then the corresponding knowledge acquisition opportunity is highlighted by the system in red font using the word "DIFFERENT". When the live and registered case lists are different, it means that either:

- an old case has been edited or a new case has been added to FastFIX and the impact of those case changes on existing RuleNodes hasn't yet been verified; or alternatively
- an old RuleNode has been edited or a new RuleNode has been added to FastFIX and the impact of those RuleNode changes on existing cases hasn't yet been verified.

Note that in an implementation that supports the shared child ConditionNode data structure, the "ParentNode" column in Figure 109 would be renamed to "ParentNodes".

## *M.11 "Check All Cases"*

The following screenshot shows a view in the prototype FastFIX system that can be used to check the current status of cases in the system.

*Figure 110: The "Check All Cases" View*

**Check All FastFIX Cases:**

**Table of FastFIX Cases**

Ordered by FastFIX Case ID, then CaseDB Case ID, then Dial Home Number. Click here to order by CaseDB Case ID. Click here to order by Username.

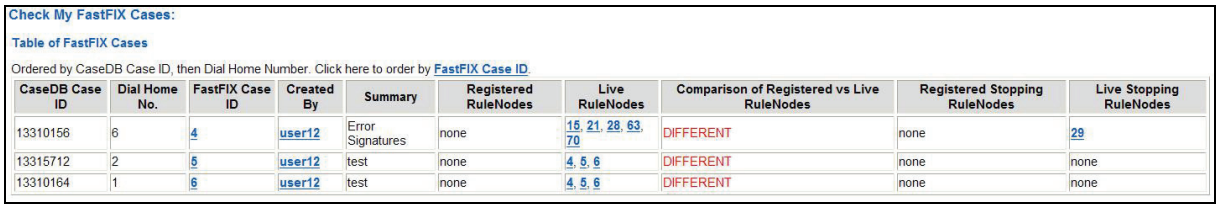
| FastFIX Case ID | CaseDB Case ID | Dial Home No. | Created By | Summary | Registered RuleNodes | Live RuleNodes | Comparison of Registered vs Live RuleNodes | Registered Stopping RuleNodes | Live Stopping RuleNodes |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 13315712 | 1 | user8 | Error Signatures | none | 7 | DIFFERENT | none | none |
| 2 | 13319936 | 1 | user4 | Error Signatures | none | 8 | DIFFERENT | none | none |
| 3 | 13760873 | 1 | user4 | Error Signatures | none | 4, 8, 53 | DIFFERENT | none | none |
| 4 | 13310156 | 6 | user12 | Error Signatures | none | 15, 21, 28, 63, 70 | DIFFERENT | none | 29 |
| 5 | 13315712 | 2 | user12 | test | none | 4, 5, 6 | DIFFERENT | none | none |
| 6 | 13310164 | 1 | user12 | test | none | 4, 5, 6 | DIFFERENT | none | none |
| 7 | 14167014 | 2 | user6 | Error Signatures | 30 | 30 | SAME | none | none |
| 8 | 14167035 | 1 | user1 | Error Signatures | 12 | 12 | SAME | none | none |
| 9 | 14167142 | 1 | user1 | Error Signatures | 28 | 20, 21, 28, 53 | DIFFERENT | 29 | 29 |
| 10 | | | user1 | | none | 2, 3, 4, 5 | DIFFERENT | none | none |
| 11 | | | user1 | | none | 2, 3, 4, 5 | DIFFERENT | none | none |
| 12 | | | user1 | | none | 2, 3, 5, 12 | DIFFERENT | none | none |
| 13 | 14167596 | 1 | user1 | Error Signatures | 28 | 28, 53 | DIFFERENT | none | none |
| 14 | 14167534 | 1 | user1 | Error Signatures | 31 | 5, 31 | DIFFERENT | none | none |
| 15 | 14168126 | 1 | user1 | Error Signatures | 16 | 16 | SAME | none | none |
| 16 | 14168342 | 1 | user1 | Error Signatures | 39 | 4, 5, 39 | DIFFERENT | none | none |
| 17 | 14051356 | 1 | user1 | Error Signatures | 47 | 47 | SAME | none | none |
| 18 | 14173184 | 1 | user1 | Error Signatures | 39 | 39, 53 | DIFFERENT | none | none |
| 19 | 14173200 | 1 | user1 | Error Signatures | 11 | 11 | SAME | none | none |
| 20 | 14173711 | 4 | user1 | Error Signatures | 28 | 28 | SAME | none | none |

The lists of live versus registered RuleNodes for cases in the system are compared, and excluding any live and not registered stopping RuleNodes, if there are differences in these lists, the corresponding knowledge acquisition opportunity is highlighted by the system in red font using the word "DIFFERENT".

## M.12 "Check My Cases"

The following screenshot shows a view in the prototype FastFIX system that can be used to check the current status of cases that the currently logged in user has created in the system. This view provides a subset of the cases shown in the previous view.

*Figure 111: The "Check My Cases" View*



Check My FastFIX Cases:

Table of FastFIX Cases

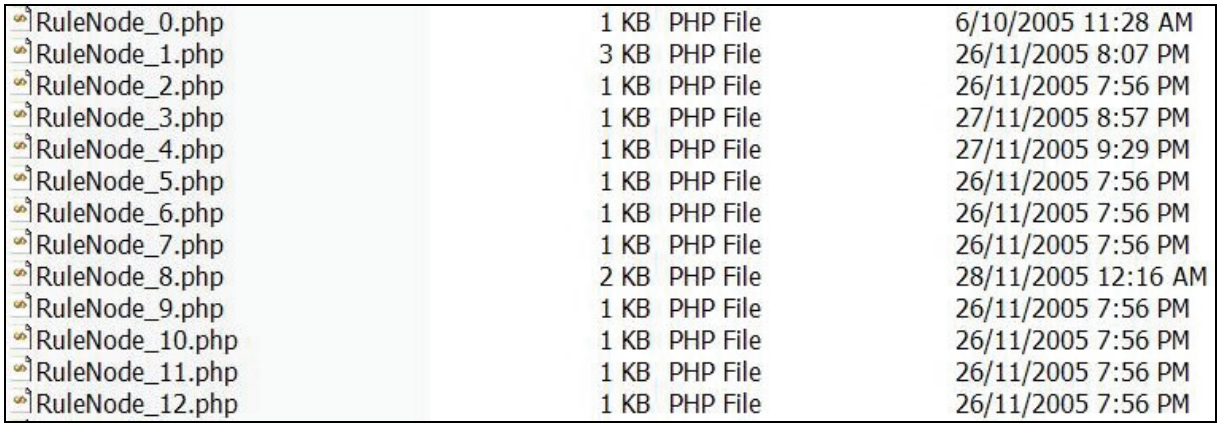
Ordered by CaseDB Case ID, then Dial Home Number. Click here to order by FastFIX Case ID.

| CaseDB Case ID | Dial Home No. | FastFIX Case ID | Created By | Summary | Registered RuleNodes | Live RuleNodes | Comparison of Registered vs Live RuleNodes | Registered Stopping RuleNodes | Live Stopping RuleNodes |
|---|---|---|---|---|---|---|---|---|---|
| 13310156 | 6 | 4 | user12 | Error Signatures | none | 15, 21, 28, 63, 70 | DIFFERENT | none | 29 |
| 13315712 | 2 | 5 | user12 | test | none | 4, 5, 6 | DIFFERENT | none | none |
| 13310164 | 1 | 6 | user12 | test | none | 4, 5, 6 | DIFFERENT | none | none |

## M.13 "Write Rule Tree"

At the click of a link, the FastFIX engine can write the entire contents of its rule tree or condition mesh out to a set of nested PHP script files, one for each RuleNode as shown in the following figure. This file set can be used independently from the FastFIX engine to evaluate either a nominated case or the parameters of some imagined case against the current rule tree or condition mesh. Whenever a RuleNode is added to or edited the current file set is automatically updated so that a complete independent back-up of the rule tree is available for execution at any time, independent from the FastFIX engine.

*Figure 112: Nested script files*



| | | | |
|---|---|---|---|
| RuleNode_0.php | 1 KB | PHP File | 6/10/2005 11:28 AM |
| RuleNode_1.php | 3 KB | PHP File | 26/11/2005 8:07 PM |
| RuleNode_2.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_3.php | 1 KB | PHP File | 27/11/2005 8:57 PM |
| RuleNode_4.php | 1 KB | PHP File | 27/11/2005 9:29 PM |
| RuleNode_5.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_6.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_7.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_8.php | 2 KB | PHP File | 28/11/2005 12:16 AM |
| RuleNode_9.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_10.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_11.php | 1 KB | PHP File | 26/11/2005 7:56 PM |
| RuleNode_12.php | 1 KB | PHP File | 26/11/2005 7:56 PM |

This feature was implemented to allow complete separation of the domain specific IP contained in the knowledge base, from the source code of the FastFIX engine itself. FastFIX can therefore be used in a consulting capacity to go into an organisation, acquire domain

specific knowledge from the experts therein, and leave behind a static version of the organisation's knowledge base. FastFIX can then be removed from the organisation without comprising the organisation's ability to use the structured knowledge that has been acquired.

As well, the static file-based rule tree can be distributed to users who require a read-only snapshot of the corporate knowledge base, rather than a full featured multi-user-update FastFIX platform. This approach may be useful for example in downloading knowledge bases to PDAs or other handheld devices. For example, Edwards suggests the use of embedded MCRDR systems in medical analytic instruments (Edwards, 1996, p228). The output language could be anything e.g. VBscript, C/C++, Java, Perl, Python and need not be PHP. It would also be possible and probably desirable to allow the class definitions and relationships resulting from the knowledge acquisition exercise to be written into more formal ontological language such as that provided by the XML-based RDF/OWL specification.

The file based rule tree could be used to implement a non-interactive batch processing facility like that used in PEIRS to process volumes of incoming cases (Kang, 1995, p 35). As with PEIRS, where required or recommended by the system an expert could separately validate the generated conclusion reports for cases, and use the write-version of the knowledge base when cases require updates to the KBS i.e. on an as-needs basis only.

## *M.14 "Solutions Preview"*

In the FastFIX prototype, a "Solutions Preview" mechanism is provided so that users can evaluate an imaginary case without having to actually add a case to the knowledge base.

The next two figures show the data entry screen and the results screen for the FastFIX "Solutions Preview" feature. The plain HTML output on the results screen and the parameterised URI interface (HTTP GET protocol[238]) to the file-based rule tree used to make these evaluations, means that a web-based HTML screen-scraper can easily run sample or real cases against the acquired knowledge base and provide a snapshot of the corresponding results in the desired format.

[238] http://www.w3.org/2001/tag/doc/whenToUseGet.html

*Figure 113: Solutions Preview – Parameter Entry*

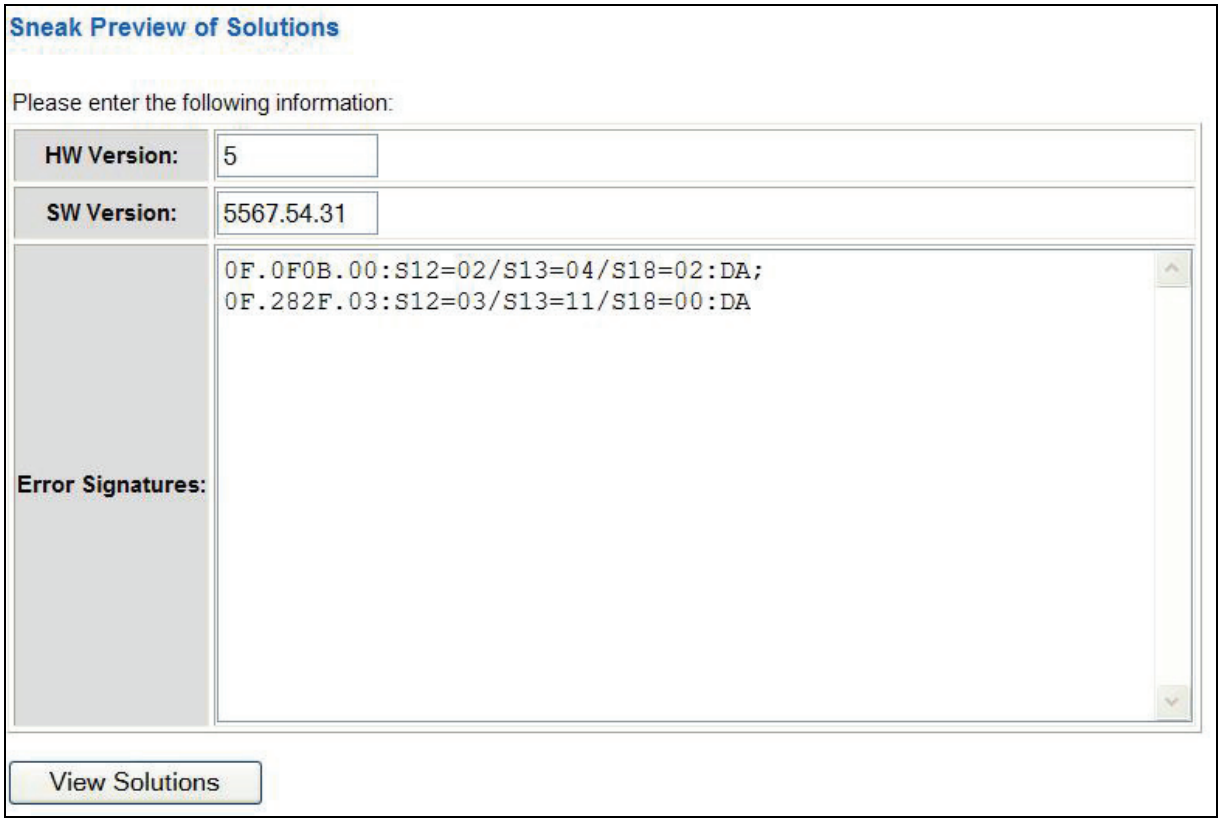
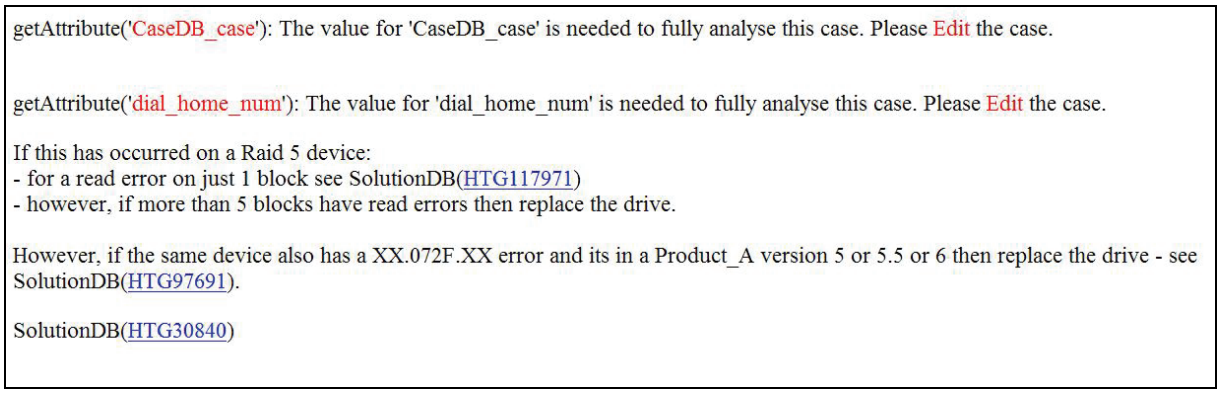



*Figure 114: Solutions Preview – Results*




This feature appears to be hinted at in (Kang, 1995, p17) where there is no stored case but instead the system recursively queries the user for data, so that the user can navigate the decision tree until a final (non-interactive) set of conclusions is reached.

## M.15 "Upload Files" and "View Uploaded Files"

The FastFIX prototype system also allows users to upload files that might otherwise be homeless. Uploaded files could be referenced by attributes or conclusions in the system by

using the format outline in Table 18 on page 198, namely: file(fileNum). The following two screenshots show the "Upload File" facility, and the "View Uploaded Files" facility.

*Figure 115: "Upload File" View*

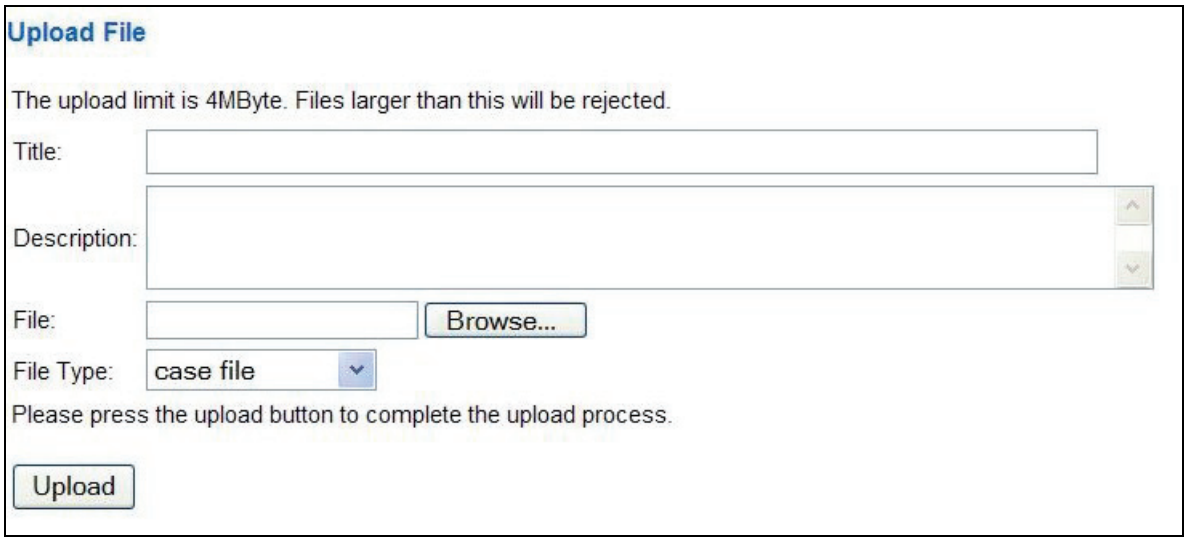
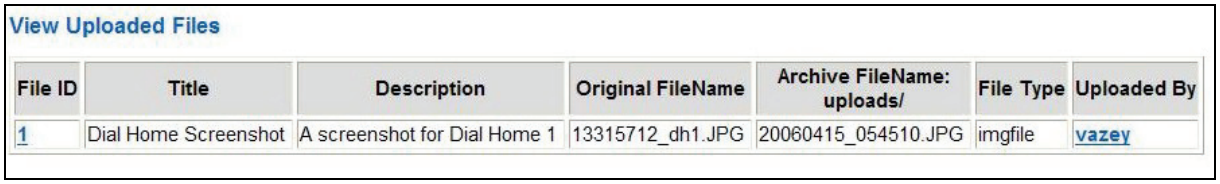



*Figure 116: "View Uploaded Files"*

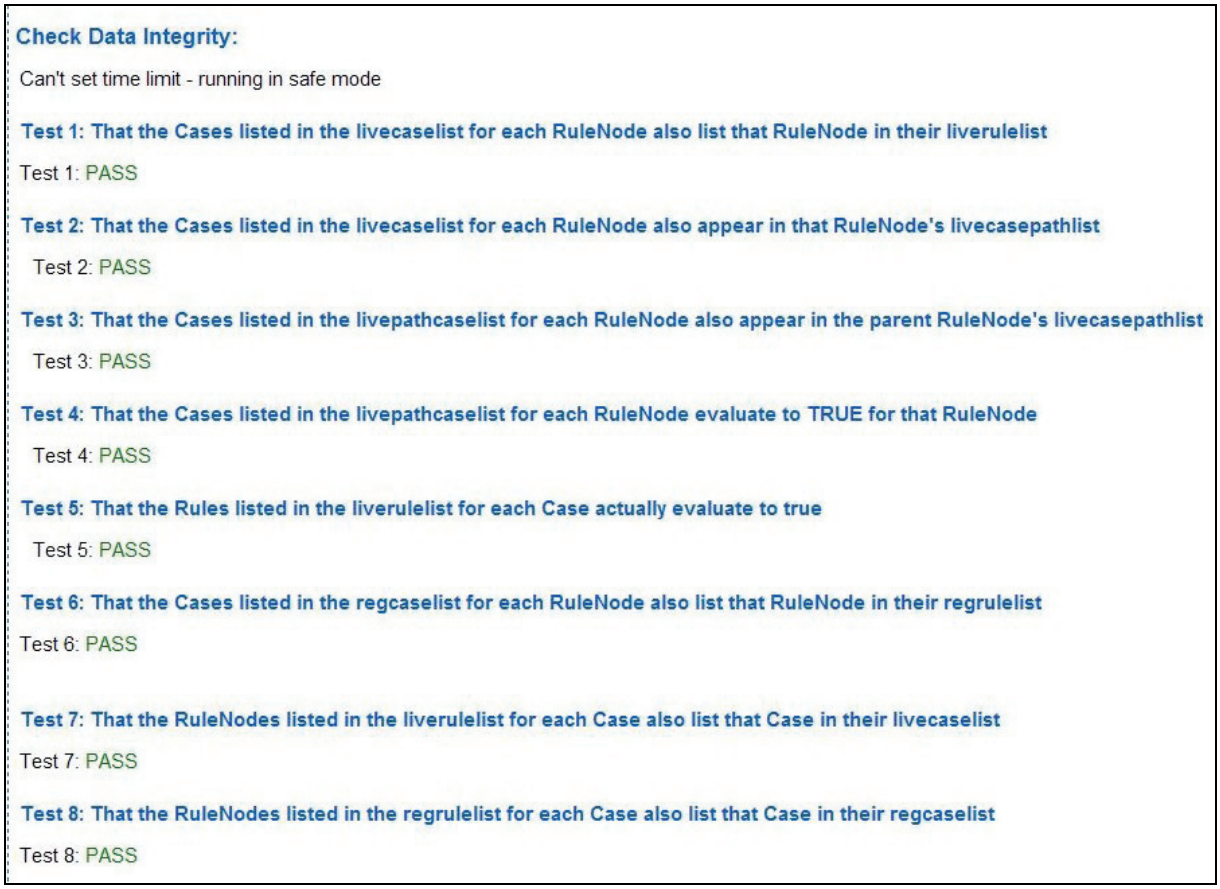



## M.16 Available Tests

As a prototype system, it was important to include some automatic tests to ensure that the integrity of the database wasn't compromised, and that the rule parsing and evaluation mechanisms were working correctly. The following screenshot lists the data integrity tests that were routinely run on the database to ensure that referential integrity was maintained.

Note that the livepathcaselist referred to in the figure below has been renamed as the *dependent case list* in the preceding text i.e. the DCL.

*Figure 117: The "Check Data Integrity" View*




## M.17 Summary

The purpose of this Appendix was to introduce the FastFIX prototype and present the layout and some of the auxiliary features of its web-based application shell. The prototype has provided an application framework in which the main features of the 7Cs system design have been developed and explored as described in Chapter 11 (commencing on page 176).