PARTNER MATRIX

*ARCHITECTURE DETAILS*

*Updated for Research Triangle Park virtual environment*

By: John M. Shoemaker

IBM Cloud

jshoe@us.ibm.com

# Introduction

The Partner Matrix is a tool created to help advise IBM teams on finding geographic based, industry and product skilled Business Partners.

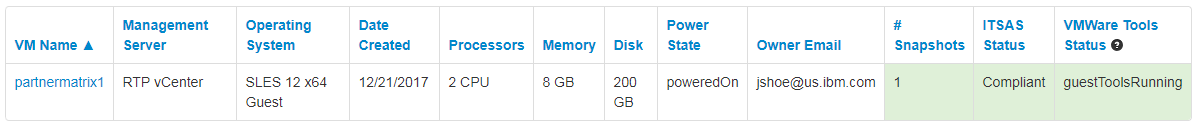
Several years ago, John Shoemaker and Michael Bottari were discussing methods to track and report about IBM business partners, without having to resort to a spreadsheet. Up to that point, we had used spreadsheets, but found them hard to communicate with others – because as soon as we sent it to someone, it became out of date. Our first efforts involved creating a Wiki site. This worked well in presentation, but maintaining the data became problematic as we would want to see different views, which required each view to be edited separately. We decided upon a new database system, which would house the data centrally yet easily allow presentations of the same data in different ways. This became the Partner Matrix.

# **Linux System**

The Partner Matrix system ran for many years in the now defunct DEMOCentral environment in a VMware instance, but was moved on March 1, 2018 to the RTP (Research Triangle Park).

The name and IP address for the current instance is **partnermatrix1.rtp.raleigh.ibm.com** (9.42.30.54)

The image can be seen here: <https://vlaunch.rtp.raleigh.ibm.com/>



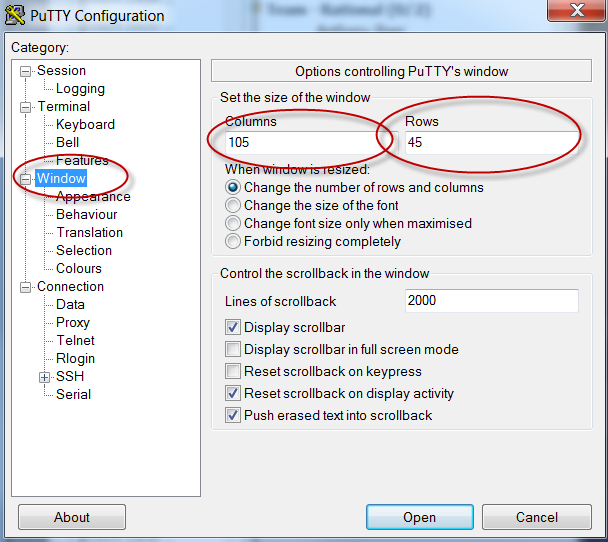
The ITSAS (Information Technology Security Automation Solution) tool for security monitoring/compliance is available at <https://itsas.raleigh.ibm.com/itsas/secure/welcome.do>

## Accessing the Linux System

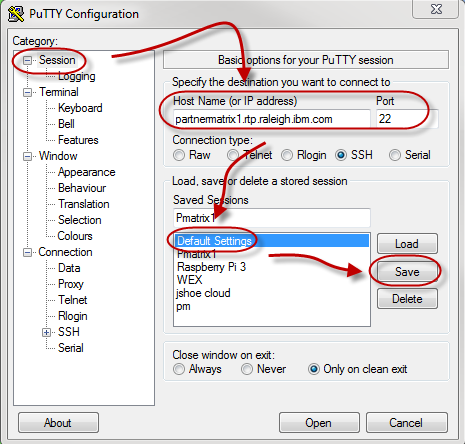
To access the running Linux environment from either a Mac or Windows machine, a terminal program (telnet client) will need to be utilized. The instructions included here are for a Windows system using the PuTTY tool as an example.

Once PuTTY is installed and configured, each new session can be started by following from step #4 below. For initial setup, start with step #1.

1. If you do not have ‘PuTTY’ on your laptop, download it from [www.putty.org](http://www.putty.org). It is only an .exe file, so you can place it wherever you want (e.g., I have a c:\bin directory for this type of thing).
2. Run it by double clicking it
3. To configure ‘PuTTY’
   1. First select ‘Window’ on the left, changing Columns and Rows as shown (don’t press Open yet). Set Columns to 105 and Rows to 45.

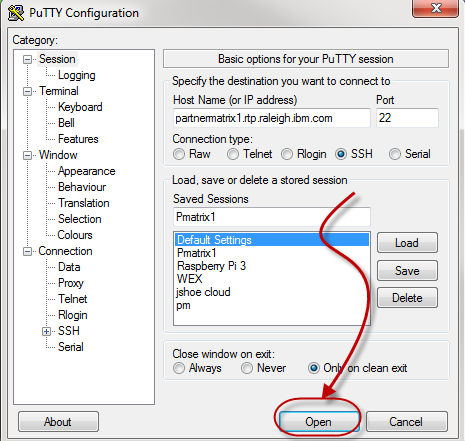


* 1. Select ‘Session’ on the left, placing the IP address and Port as shown. Then click on ‘Default Settings’ and press the ‘Save’ button. The Host Name to use: partnermatrix1.rtp.raleigh.ibm.com) on Port 22.



* 1. Your configuration is complete and should not need to be done again.

1. To start your Partner Matrix session, simply press the ‘Open’ button.



1. At the “login as:” prompt, enter your Partner Matrix ID. If you are unsure of what it is, try your internet ID (without the @us.ibm.com).
2. At the ‘Password’ prompt, enter your password.

## **Directory Structures**

The directory structure is as follows:

| ***Directory*** | ***Description*** |
| --- | --- |
| /etc/apache2 | Apache Web server directory |
| /var/log/apache2 | Apache access and error log location |
| /opt/pm | Top level dir for PM |
| /opt/pm/bin | Shell scripts for system maintenance |
| /opt/pm/cgi-bin | Shell scripts called via web pages |
| /opt/pm/code | Informix 4gl code files |
| /opt/pm/code/sql | Contains DDL (though use the nightly exported SQL to ensure it’s the most current) |
| /opt/pm/forms | Informix perform code files – these are forms for searching, inserting and updating certain tables |
| /opt/pm/query | SQL queries requested by community, manually run |
| /opt/pm/htdocs | Web pages – default http location |
| /opt/pm/NA/all/H | Rendered pages applicable to all PM implementations |
| /opt/pm/NA/I | All logo images files (jpg files) |
| /opt/pm/NA/\*\*/H | Rendered pages for specific implementation that span across all partners |
| /opt/pm/NA/\*\*/S | Rendered pages for specific implementation for specific partners – revised UI |
| /opt/pm/NA/\*\*/P | Rendered pages for specific implementation for specific partners – original UI |
| /opt/pm/NA/\*\*/R | Rendered reports for specific implementation for specific partners |
| /opt/pm/NA/\*\*/tmp | Work space for specific implementation |
| (where \*\* is NC, BU, ES, IO, SE, ST, or WU) |  |
| /opt/pm/NA/I | All logo images files (jpg files) |
| /opt/pm/raw\_files | Workspace for import files (not used much anymore – only for manually imported files) |
| /opt/pm/tmp | Work space |
| /data/exports/dbexports | Backups (dbexport) of the *partners\_na* database |
| /data/exports/backups | Location of *tar* file(s) created for sending offsite |
| /data/logs | Log files written by Partner Matrix |
| /home/ftpuser | Location used to ftp to/from the server |

## Database Server Details

The database server is the Informix Dynamic Server Version 12.10.FC7.

The server configuration file can be found here: */opt/IBM/informix/etc/onconfig.pmatrix*

The physical files holding the chunks held in the dbspaces can be found here: */data/IBM/informix/pmatrix/data*

The DBSERVERNAME is: pmatrix

The primary dbspace used for tables is: *data1dbs*

The database name used for North America is: *partners\_na*

The architecture of the Partner Matrix system is such that other IOTs could use it, each with their own database. So AP would use a database called partners\_ap, etc. At this time, only NA is using the PM system.

## Key Tables

The underlying relational database supporting the Partner Matrix is Informix. The tables shown below make up the Partner Matrix system. The production database name for North America is: **partners\_na**

| ***Table Name*** | ***Description*** | ***Status*** |
| --- | --- | --- |
| **master** | Master settings table, that allows system-wide changes to be made without making code changes. This table only contains one record, with system-wide details. See the Appendix for details on this record. | active |
| **control** | Control table that allows implementation-based changes to be made without code changes. A row must exist for each implementation (e.g., AU, SU, BU, etc). There are many flags that can be turned on/off to control behavior and what options the user will see | active |
|  |  |  |
| **coverage\_\*\*** | These tables contain the main profile information for each partner – this includes the description, CEID and name, BPR, industries, geographies, etc. There is one table for each implementation (e..g, ‘au’ for Analytics Unit) | active |
| **success\_\*\*** | These tables contain the success story information for each partner – this includes the CEID, title and URL or each story. There is one table for each implementation (e..g, ‘au’ for Analytics Unit) | active |
| **contacts\_\*\*** | These tables contain the additional contact information for each partner – this includes the CEID, name, title, email and phone number for each contact. There is one table for each implementation (e..g, ‘au’ for Analytics Unit). The primary contact information is contained in the coverage\_\*\* table | active |
| **skills\_++** | These tables contain the specific skill rating information for each partner. There is at least one table for each implementation (e..g, ‘se’ for Security Unit), but there may be up to 10 skill tables for an implementation. These skill table names are contained in the *control* table. | active |
|  |  |  |
| **tbptv** | This table is based on a feed, providing authorizations, revenue, number of people certified, etc. | deprecated |
| **certs** | This table is based on a feed, providing the details of all partner-held certifications. | active |
|  |  |  |
| **skill** | Has the name of each Skill represented for each brand | active |
| **skill\_cat** | Has the Skill Categories for each Unit and Brand, allowing Skills to be grouped by Categories | active |
|  |  |  |
| **accreditation** | Details on partner accreditation achievements – Bronze, Silver and Gold | active |
| **tbptv\_history** | Contains the quarterly history from the *tbptv* table | deprecated |
| **skills\_history** | Contacts the quarterly history from the *skills\_history* table | deprecated |
| **reps** | Has a single row for each IBM that exists in the *coverage\_\*\** tables, for BPRs and TSSs. Their name and URL to their contact details are included for each person | active |
| **reps[2-6]** | These are views that are setup to mirror the *rep* table | active |
| **vads** | Has one row for each VAD that is used to validate entries as well as provide name and URL information | active |
| **iots** | Each IOT that is using the Partner Matrix is included in this table – at this time, only the North America IOT is represented | active |
| **vap** |  |  |
|  |  |  |
| **unit\_mgrs** | Identifies the list of sales & technical managers that show up on the PM Home page of each implementation | active |
| **unit\_brand** | Identifies the relationship between Units and Brands, for example the Analytics Unit has records for BA (business analytics), EM (ecm), and IM (information management) | active |
| **region** | A unique representation of each of the NA Regions | active |
| **lastload** | Contains the timestamp for the last load of our data feeds. This table is manually updated using the *$FORMDIR/lastload* form | active |
| **brands** | Unique identifying table of brand IDs (multiple brands may exist for a Unit structure). For example, IM, BA. EM are brands that exist in the AU (Analytics) Unit | active |
| **states** | Listing of States used for validation | active |
| **access** | Historical table that lists each web access to the web server. There is a script that helps prepare the data which then replaces what’s already in the table | active |
| **tasks** | Partner Matrix change request details | active |
| **align\_brand** | Allows separation within a Unit – for example, Analytics separation to Platform and Solutions | Not in use |
| **align\_cat** | Separation within a Unit by category | Not in use |
| **align\_subbrand** | Further separation within a Unit | Not in use |
|  |  |  |
| **load\_hist** |  | deprecated |
| **bus** |  | deprecated but some references exist |
| **units** | 602-463-5976 | deprecated but some references exist |
| **Contacts** |  | deprecated |

\_ac, \_auin, \_aunew \_eucolnames, \_eucolnamesu, \_hcc, \_hci, \_ics, \_ind\_colname, \_isnew, \_ranew, \_sunew, \_t2, \_w, \_w2, \_worktab, bptv, bptv\_parts, brand\_certs, brand\_certs\_det, capability, capability\_ibm, certs\_old, cov\_ws, coverextend\_st\_del, cross, eus, eus2, fail, feedback, field, import1, imts in\_cloud, in\_cu, in\_st, in\_wat, ind\_colname, ind\_group, industry, industry\_auth, iness2, inst2, Midwest, nbptv, nbptv\_parts, ntest old\_sws, old\_vap, oldcoverage\_ra p2p, part\_details\_im, part\_details\_ws , pw\_level , reg\_eu\_state, relations,

segment\_leaders, segmentation, solutions, ssrtab, storage\_cert, svp, svp\_sol\_, task\_ord, tmp\_incom, tmp\_skill, tmp\_skill\_cat, upd\_log

## Web Server Details

### Web Server Commands

To restart the server: sudo rcapache2 restart

### URL Shortcuts

All of the URL links use the **https://ibm.biz** prefix. This site is a shortcut address for long URLs, and is managed by the IBM Snip tool: <http://snip.innovate.ibm.com/home>. The table below contains the full URL and shortcuts for each implementation.

| ***Implementation*** | ***Internal Code*** | ***Full URL*** | ***Shortcut*** |
| --- | --- | --- | --- |
| IBM Cloud | NC | <http://partnermatrix1.rtp.raleigh.ibm.com/NA/NC/H/find_partners.html> | https://ibm.biz/FindCloudPartners |
| Commerce | BU | <http://partnermatrix1.rtp.raleigh.ibm.com/NA/BU/H/find_partners.html> | https://ibm.biz/CommerceUnit |
| Systems | ST | <http://partnermatrix1.rtp.raleigh.ibm.com/NA/ST/H/find_partners.html> | https://ibm.biz/FindSystemsPartners |
| Social | ES | <http://partnermatrix1.rtp.raleigh.ibm.com/NA/ES/H/find_partners.html> | http://ibm.biz/EnterpriseSocial |
| Watson IoT | IO | <http://partnermatrix1.rtp.raleigh.ibm.com/NA/IO/H/find_partners.html> | https://ibm.biz/InternetOT |
| Security | SE | <http://bp-partnermatrix.democentral.ibm.com/NA/SE/H/homeU.html> | https://ibm.biz/SecuritySW |
|  |  |  |  |
| Watson (not production) | WU | <http://bp-partnermatrix.democentral.ibm.com/NA/WU/H/homeU.html> | http://ibm.biz/PartnerMatrix-Watson |

### DNS Resolution

On Linux systems, sometimes it takes a long time for a password prompt to show up when trying to SSH into a system on the local network. This was the case for our vmware image. It turned out there is a setting that controls whether SSHd should not only resolve remote host names but also check whether the resolved host names map back to remote IPs. The directive UseDNS controls this behavior is usually enabled, as was the case for PM. Uncommenting the line carrying the UseDNS directive and setting it to “no” disabled this feature, removing the long delay. This is documented in case the issue arises again.

### Enterprise Directory

It is desired to authenticate users that want to update the system against the corporate directory. This feature is not in place today.

# Partner Matrix

## Methodology

The Partner Matrix system is a collection of static and dynamic web pages. Most of the pages requested by a user are ‘static’, meaning they are pre-existing and only need to be served by the web server. The only activities that require an actual database connection and dynamic rendering of a web page are *Find a Partner, Elite Partner* List and *Update* mode.

Since so many pages are static, it’s important for them to be updated as content changes. When a user makes an update via Update mode, a small trigger file is created, making it known that a change for that Partner Matrix instance has been made. A *cronjob* runs for that instance, looking to see if that trigger file has been created. If the file exists, a process will kick off to rebuild the static pages for that instance.

The trigger file location variable for each instance, as well as other variables, are found in the file: *$HOME/.profile ($HOME=/opt/pm)*

## Rebuilding Static Pages

1. The *.crontab* file is found in the $CODE directory. The first line in the example below depicts a daily script that will export the entire database to files. This export requires that no other process is using the database at the same time. The other files are the scripts that will rebuild the individual Partner Matrix instances. The ‘-t’ option indicates that the rebuild will only occur if the trigger file for that instance exists. This is an example only – type the command ‘*crontab -l*’ to see what is currently configured for the system.

0 23 \* \* \* /opt/pm/bin/root\_dbexport.ksh

0 \* \* \* \* /opt/pm/bin/regenFilesU.ksh -g NA -b NC -t

12 \* \* \* \* /opt/pm /bin/regenFilesU.ksh -g NA -b BU -t

18 \* \* \* \* /opt/pm /bin/regenFilesU.ksh -g NA -b ES -t

24 \* \* \* \* /opt/pm /bin/regenFilesU.ksh -g NA -b ST -t

30 \* \* \* \* /opt/pm /regenFilesU.ksh -g NA -b IO -t

36 \* \* \* \* /opt/pm /regenFilesU.ksh -g NA -b WU -t

42 \* \* \* \* /opt/pm /regenFilesU.ksh -g NA -b SE -t

## System Backups

As shown in the crontab example in the previous example, a nightly export of the **partner\_na** database occurs. These exports are written to *dbexports* directory (see the directory locations for additional details).

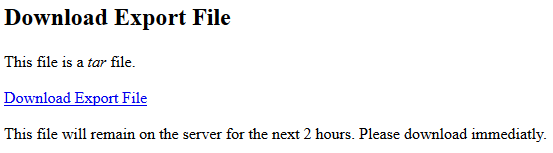
A command driven export of critical files (data and code) can be executed on demand. This currently is performed on a weekly basis (manually). The tar command that is executed is:

**tar cf $LOGFILE --exclude=\*.tar --exclude='data/IBM' pm /data**

This export is initiated by clicking in the white space between the version number of rendered date at the bottom of any Partner Matrix page. This version number is changed in the *master* table. See details on this table for more information.



This will initiate the $CGIBIN/snag.ksh script, which will create the *tar* file, and then provide a Download link. *Contrary to the message shown below, the file will not be cleaned up by the system but will need to be manually removed at some point to free up space.*



Once this tar file has been downloaded, it is manually moved to the Box folder listed below. This action should be performed on a weekly basis.

**Partner Matrix > Code Source Files > Archives**

A manual process is used to copy modified source files. This should be performed each time changes are made to the source files. These files are copied from the Linux server to their corresponding Box folders:

* 1. Partner Matrix > Code Source Files > code
  2. Partner Matrix > Code Source Files > forms
  3. Partner Matrix > Code Source Files > cgi-bin
  4. Partner Matrix > Code Source Files > bin
  5. Partner Matrix > Code Source Files > all-H
  6. Partner Matrix > Code Source Files > htdocs

The manual processed currently used to place a copy of changed files in Box is as follows:

1. On Linux system, copy file to /home/ftpuser
2. On laptop, ftp to local Downloads
   1. ftp 9.42.30.54
   2. user: ftpuser
   3. password:
3. On laptop, drag to the appropriate Box folder

## Partner Matrix Change Requests

### Steering Committee

To ensure the Partner Matrix system continues to meet the business needs of its users, a Steering Committee has been created to act as a change control board – reviewing change requests and either approving or denying the changes. Since there are many groups using the system now, it’s important that as much as possible remains common among all users. The Committee sets direction on what’s included in future editions as it expands. As requests for changes are made, they are added via the Change Management system, as described in the section below.

Some changes that are smaller in scope are added without Committee approval if they do not fundamentally change things.

### Change Management

Task Request Creation and Management is managed by running a script that opens a database form that is used to enter/update task requests for Partner Matrix changes. These changes will originate from a variety of sources, but this is where they are consolidated. By opening the form via this script, a report will be updated when the script is exited.

$FORMDIR/run\_tasks.ksh

# Coding

## Languages used

### Informix 4GL

Most of the coding is done using the Informix 4GL language (7.32.FC4). While there are two choices when compiling Informix 4GL, this system only uses the ‘p-code’ (pseudocode) compiler. This means that a true executable is not created, so a runner is used to execute the compiled code. The simple example below would not work to execute code for Partner Matrix due to the inclusion of multiple compiled files that would need to be grouped together. To facilitate this, a Makefile is in the $CODEDIR directory.

These commands are generally used to compile (fglpc) and execute (fglgo) the 4GL code.

$ fglpc filename.4gl

$ fglgo filename.4go

The 4GL code all resides in the $CODEDIR directory, with manual backups to the Box folder.

### Informix SQL - forms

There are some database forms that have been created to help maintain the system. These forms are not available to the end users. These are created using the Informix I-SQL program.

These commands are generally used to compile (sformbld) and execute (sperform) the form code.

$ sformbld filename.per

$ sperform filename.frm

The form code all resides in the $FORMDIR directory, with manual backups to the Box folder.

There is a script in the $FORMDIR that can be run that sets the correct environment variables for the forms and invokes the sperform command. This can be run as follows:

$ ./r filename

### Informix 4GL - reports

There are some database reports that have been created to produce printable downloads of data for a specific partner. These reports are available to users via the Coverage List (by BPR) listing. There is a small download message that will download the specific report for that partner. These are created using the Informix 4GL code. *Note – there is a report writer function available as part of I-SQL, but in this case, we are using the report code but embedded inside the 4GL code.*



The report code all resides in the $CODEDIR directory, with manual backups to the Box folder.

### Shell Scripting

There are several instances of Linux shell scripts in use – all of them are Korn shell (ksh). As part of these scripts, several Linux commands are utilized, including awk, sed, tr, uniq, etc. These shell scripts are generally found in $BINDIR and $CGIBIN directories.

Several other scripts in the $CGIBIN directory are not currently in use. Prior to updates being made online via web forms, updates were made by uploading spreadsheets, so there are scripts in place that were used to automatically process these files.

The scripts in the following table are the most critical.

| ***File Name*** | ***Description*** |
| --- | --- |
| **regenFilesU.ksh** | This script rebuilds all of the static pages for a specific implementation. Command line variables are as follows:  ./regenFilesU.ksh partners\_na -g NA -b AU -t -s -m  Where   * partners\_na is the database name * -g NA indicates the North America geography * -b AU indicates the Unit (Analytics Unit in this example) * -t means only run if the trigger file is present * -s instructs the script to write the ‘start’ or ‘top’ page – this is the page: <https://ibm.biz/PartnerMatrix-Top> * -m instructs the script to write the master list, which may not actually be shown, but this is the instruction to create it |
| **run\_after\_svpinfo.ksh** | After the *svp\_info* table is reloaded (manual process today), this script should be run. It does one major functions – it updates the *vad* column in each implementation if the value in the refreshed *svp\_info* table is different than the vad entry in each coverage\_\*\* table |
| **touchAll.ksh** | Following the manual updates of the svp\_info and certs tables, running this script will create the trigger files for each implementation, instructing the static pages to be rebuilt based on the current *crontab*. |
| **watch\_files.ksh** | While was used mainly to support the automated loading of new files, it is still used when the new *tbptv.csv* and *certs.csv* are manually processed. When these files are ftp’ed to the system, run this command which will do a little processing of the files and will move them to $HOME/raw\_files |
| **write\_addBPXX.ksh** | This script will from the *regenFilesU.ksh* script and will rewrite the file in $CGIBIN by the same name (actual file name written will replace XX with the implementation CODE, such as AU for Analytics). This script will use this file as the basis for the new file: *.addBPXX* |
| **write\_setupUpdXX.ksh** | This script will from the *regenFilesU.ksh* script and will rewrite the file in $CGIBIN by the same name (actual file name written will replace XX with the implementation CODE, such as AU for Analytics). This script will use this file as the basis for the new file: *setupUpdXX* |
| **write\_updRecXX.ksh** | This script will from the *regenFilesU.ksh* script and will rewrite the file in $CGIBIN by the same name (actual file name written will replace XX with the implementation CODE, such as AU for Analytics). This script will use this file as the basis for the new file: *.updRecXXXX* |
| **write\_updSkillsXX.ksh** | This script will from the *regenFilesU.ksh* script and will rewrite the file in $CGIBIN by the same name (actual file name written will replace XX with the implementation CODE, such as AU for Analytics). This script will use this file as the basis for the new file: *.updSkillsXX* |

## Compiling and Building Executables

As described above, the $CODEDIR/Makefile controls the building of the 4GL code programs. The source and *executable* files all reside in the $CODEDIR. These are not true executables, but are run using the pcode runner (see the Informix 4GL section above for these details). The general process for creating the runnable file is to concatenate the compiled p-code output from each source file into one file. The Makefile uses the Linux ‘cat’ command to perform this function.

## Coding Conventions

## Key 4GL Code Files

| ***File Name*** | ***Key Functions*** | ***Description*** |
| --- | --- | --- |
| **globals.4gl** |  | Global variables |
| **getcolor.4gl** | GetColor | Contains color functions for active/inactive menu choices |
| **headerU.4gl** | Header | Contains header functions for the Partner Matrix menu choices |
| **readcmdln.4gl** | ReadCmdLn | Contains function for extracting the command line options |
| **proccmdln.4gl** |  | Contains functions to process the command line options and to initialize the environment |
| ProcCmdLn | Process command line options and initialize environment |
| ShowVer | Displays the bottom line on every page, which includes the version number and date of rendering |
| GetProgName | Converts a numeric to a program name |
| AddText | Appends provided string to the base string provided |
| BuildSkillStr | Builds a string of selected skills that match the provided skill level |
| BuildSkillStrU | Builds a string of selected skills that match the provided skill level when multiple skill tables are used |
| LoadSkillArr | Populate an array of the skill names for the selected brand |
| LoadSkillArrU | Populate an array of the skill names for the brands used in the selected Unit |
| LoadControlFlags | Sets global variables based on the control record for the specified brand (note – moving away from these global vaiables and using the CtlRec record instead) |
| GetControlRec | Reads from the db the control record for the specified brand |
| LoadDetails | Reads from the db the time stamps of the data loads |
| GetFBRep | Read from the db the person that should be mentioned at the bottom of every page to contact to provide feedback |
| GetUnitBrandInclStr | Read from the db the list of brands associated with the specified Unit |
| **opendatabase.4gl** | OpenDatabase | Contains function to open the specified database |
| **miscfuncs.4gl** |  | A series of functions that are applicable to a wide range of programs |
| LoadActiveBrands |  |
| BuildCntStr |  |
| BuildBaseStr |  |
| BuildBaseStr2 |  |
| BuildBaseExtendStr |  |
| LoadSkills |  |
| LoadSkillsU |  |
| LoadSolutions |  |
| LoadVAP |  |
| GetVAPforBrand |  |
| IsValidVAPforBrand |  |
| LoadCrossRec |  |
| getRepLink |  |
| GetState |  |
| LoadSkillCat |  |
| GetSkillCatDesc |  |
| BuildProdStr |  |
| MatchProd |  |
| LoadContactArr |  |
| LoadContactArr2 |  |
| GetCatCnt |  |
| LoadSkillRec |  |
| UnitModeYN |  |
| GetBrandName |  |
| noSpaces |  |
| GetSSRStr |  |
| GetStorageCertStr |  |
| GetGlobalMsg |  |
| ListBoxVAD |  |
|  |  |  |

| ***File Name*** | ***Description***  ***(***\*\* is the 2-digit implementation code, such as ‘au’ for Analytics Unit) |
| --- | --- |
| **start.4gl** | Creates the $HTDOCS/index.html file |
| **homeU.4gl** | Creates the $HOME/NA/\*\*/H/homeU.html file |
|  |  |
| **findPartner\_eusU.4gl** |  |
|  |  |
| **menu\_coverageU.4gl** | Creates the $HOME/NA/\*\*/H/menu\_coverageU.html file |
| **coverage\_eusU.4gl** | Creates the $HOME/NA/\*\*/H/coverage\_eusU.html file |
| **coverage\_eus\_tssU.4gl** | Creates the $HOME/NA/\*\*/H/coverage\_eus\_tssU.html file |
| **coverage\_eus\_otherU.4gl** | Creates the $HOME/NA/\*\*/H/coverage\_eus\_otherU.html file |
|  |  |
| **authorization.4gl** | Creates the $HOME/NA/\*\*/H/authorization.html file |
|  |  |
|  |  |
| **skillsU.4gl** | Creates the $HOME/NA/\*\*/H/skillsU.html file |
|  |  |
| **industriesU.4gl** | Creates the $HOME/NA/\*\*/H/industriesU.html file |
|  |  |
| **nbuildPagesS.4gl** | Creates the individual partner files for the new UI.  Creates these files (where XXXX is a valid CEID)  $HOME/NA/\*\*/S/XXXX.html (standard partner detail pages) |
| **nbuildPagesU.4gl** | Creates these files (where XXXX is a valid CEID)  $HOME/NA/\*\*/P/XXXX.html (standard partner detail pages)  $HOME/NA/\*\*/P/XXXX-nh.html (no header files)  $HOME/NA/\*\*/P/XXXX-sec.html (secret files) |
|  |  |
| byStatus.4gl | Creates the $HOME/NA/\*\*/H/byStatus.html file |
| updRec.4gl | Executes dynamically to update Matrix profile records |
| addBP.4gl | Executes dynamically to add Matrix profile records |
| massUpdVAD | Executes dynamically to perform mass updates to VAD fields based on the tbptv table |
| adhoc1\_eusU |  |
| heatmap\_detail.4gl |  |
| heatmap.4gl |  |
| heatmap\_list.4gl |  |
| region\_heatmap\_detail.4gl |  |
| coverage\_planU.4gl | Creates the $HOME/NA/\*\*/H/coverage\_planU.html file |
|  |  |
| updHCcov.4gl | Temporary use |

# Monitoring

## Log Files

There are several log files that can be reviewed to see what has recently been requested, especially in terms of updates of adds. There is not a process available (via the web) to delete actual partner rows from the database. Contact rows and success story rows can be deleted, but not the actual partner profile (coverage\_xx) or skills rows (skills\_xx). It is recommended that these rows not be deleted – simply changing the *reseller code* to ‘N’ will remove them from the web pages.

### Location: /data/logs

|  |  |
| --- | --- |
| add\_history.log | Logged by the addBPXX.ksh scripts, showing the timestamp of when a new Partner was added to the Matrix, and who did the add |
| upd\_history.log | Logged by the setupUpdXX.ksh scripts, showing the timestamp of when an existing Partner was updated in the Matrix, and who did the update |
| updskill\_history.log | Logged by the updSkillsXX.ksh scripts, showing the timestamp of when the skills of an existing Partner was updated in the Matrix, and who did the update |
| str.log | Logged by the XXquery\_eus.ksh scripts, showing what is passed to the ADHOC1\_EUSU program |
| var1.log | Logged by the updRecXX.ksh scripts, showing what is passed to the UPDREC program |

### Location: /tmp

|  |  |
| --- | --- |
| env.txt | Logged by many of the $CGIBIN scripts, showing the Linux environment variables. This file is not appended to, so only the results from the last execution will be found here |

## Troubleshooting

The Partner Matrix system should continue to hum along without any support interference. There are a couple of things that have happened in the past, so they will be addressed here.

Problem 1 – one or more static web pages were not fully rendered. This used to occur when a rebuild kicked off while a rebuild was already in process (for the same implementation). This should no longer be possible, but if this problem does occur, the solution is to simply kick off the implementation again. Once it cycles through, the page(s) should be fine. If not – this may be a result of Problem #2.

Problem 2 – many of the pages that are built are driven by data stored in the database. When new data is entered, it’s possible that certain limits are exceeded, and a page build will be aborted. One sure way to verify this is the problem is to look in the web server log: /var/log/apache2/error\_log. This log will identify the program and line number where the error occurred. Depending on the specifics, one possible fix is to increase the size of the array (for example).

Problem 3 – none of the partner matrix pages will display. Typically, there is one of two possible things going on to cause this:

1. The redirection server is down (served by ibm.biz). If this is the case, there is not much that can be done. You can verify this is the issue by issuing the IP address on the URL line – if the system is up, the “top” page will be displayed.
2. The actual VMware image is experiencing problems. If this is the case, you need to access the management console to shut down and restart the image.

# Blog Posts

The Partner Matrix blog is located at these locations:

Full URL: <https://apps.na.collabserv.com/blogs/a2b93f22-0a53-4569-b404-c7e5cbf8ba32?lang=en_us>

Shortcut: <http://ibm.biz/PartnerMatrixBlog>

The blog is part of the Partner Matrix Community, found here: <https://apps.na.collabserv.com/communities/service/html/communitystart?communityUuid=a2b93f22-0a53-4569-b404-c7e5cbf8ba32>

Any time a big change happens to Partner Matrix, we try to notate it via a blog post. Another typical topic is an update on Partner Matrix usage.

# Feedback

Please provide feedback on this document or Partner Matrix in general to John Shoemaker ([jshoe@us.ibm.com](mailto:jshoe@us.ibm.com)).

# Appendix

## Master Record

The Master record has system-wide settings, values and wording for various parts of Partner Matrix.

The text for the help sections are defined here:

### Search Results

We want to reward those *IBM Business Partners* that are investing in themselves and in IBM technologies and are top performers that we know can manage and close opportunities.  
  
When a search is performed via the **Partner Matrix** Search Screen, the results are returned using a prioritization algorithm. While the specific sort factors may change from time to time, the general prioritization rules are:

When a product skill is in the search criteria, any Elite partner (the Cream of the Crop) will be moved to the top of the list

1. Partners with software revenue in the past 12 months will be placed higher
2. Partners that have Solutions in the Global Solutions Directory (GSD) will be placed higher
3. Partners that have product skills will be prioritized above those without
4. Finally, since it's important to have a contact at the Partner to engage them, having a contact available will be the final sort

Using this ordering technique, our best Partners should bubble up towards the top of the results.

### Resell Authorizations

Business Partners must be authorized to resell IBM software through the IBM PartnerWorld Channel Value Rewards (CVR) Program. There are different authorizations for different software product groups. The approvals listed represent those product groups this partner is authorized to resell.

Note: There are some products that are **open** and do not require authorization.

### Competency Achievement

At the beginning of 2017, PartnerWorld launched the Competency Framework, a single global capabilities initiative. Business Partner qualify based upon skills (certifications) or approved verified solution or deployments. All types of Business Partner models are recognized, including resale, influence and XaaS. Other requirements include the number of sales and technical resource and sales success. The Partner Matrix provides Competency status to further support the research into a partners capabilities.

### Skill Levels

Skill levels are the most differentiating aspect to Partner Matrix. Skill levels are given at the Product Level even if that product is part of a larger software bundle/offering. This Skill number is a combination of both Sales skills AND Technical skills for that Partner for that product. The number reflects the propensity of the Partner to close and support a passed lead on a medium/complex opportunity with no assistance from IBM. Teams interacting with a Partner can become involved in the social aspect of evaluating the partner skills. We identify skills at these different levels:

1. Beginner, high level overview presentation and basic questions
2. Intermediate, High level value proposition, demo, etc
3. Expert, solid experience, training, and advance concepts, PoC, RFP, and Implementation
4. Elite, recognized leader

### Market

Partners embrace the IBM Market designations to varying degrees. Partner Matrix tracks a partners level of involvement in a particular Market (down to the sub-market or industry level).

We identify this involvement using these values:

1) Only sell into an industry

2) Have invested in a practice

3) Have created solutions supporting this Market or Industry

Note: For the most part, this data is maintained manually by the IBM Coverage team, but some attempts are being made to update using other data we have.

### Geographical

Some partners have resources to sell and support sales opportunities throughout North America. Other partners try to maintain only regional or local coverage. Those geographical areas that a partner can support are listed here.

### Solution Details

Our Partners have a lot of experience and knowledge, which they utilize when architecting and building solutions. These solutions include IBM software and their expertise. Many of these solutions are based on specific markets and Industries and others are more of a general nature. If a solution targets specific competencies, those are also listed. The solution name is a link to details in the Global Solutions Directory (GSD).

### Reference & Successes

Our partners are very experienced and capable. When there are notable success/win stories, case studies, customer references and related content, there will be links to them here. The content might lead to another web page, or to a downloadable file.

### Certification Details

### Contacts

### Social Feedback

There are many IBMers working with our Business Partners. These Partners are critical to our business. To better understand each Partner and their capabilities, we allow users to provide feedback on them. This feedback consists of a 1-5 rating (5=best) and a descriptive comment. The numeric ratings are averaged and become the basis for the number of stars that are shown for each Partner.

The inputs are moderated and will become public once reviewed.

Each record has a status flag, which can be one of three different values:

|  |  |
| --- | --- |
| ***Status*** | ***Description*** |
| R | This is the default value for new records – meaning it’s in Review mode and the comment will not be displayed until it leaves this state. Even though the comment will not be displayed, the rating is part of the average. |
| P | Product mode – all review comments are displayed |
| S | Suspended mode – these reviews are ignored – the comments as well as the rating number |

### Details from Other Brands

There are several groups within IBM using the **Partner Matrix** tool. If there are links in this section, that means there are details from these other groups available to you. Following these links will allow you to view the details these other groups are using for this same Partner.

### Target and PV Information

The Target and Point of View (PV) information is maintained in the **Business Partner Relationship Management (BPRM)** system and is only displayed here.

The values shown are for the current quarter.

### Planning