Release 12 Security Recommendations

Chuck Kennedy
Susan Behn
Brian Bent
Solution Beacon, LLC

Introduction

When you go to your local airport, your ultimate goal is to get on an airplane, right? What is the first thing that you are asked for at the ticket counter? Form of identification, right? Then you go to the security checkpoint and what are you asked to present? Form of identification? Why is that, well in this example, the answer is simple, the airlines and the government want to make sure you are who you say you are. Maybe if you lived in the town of Mayberry, then identity verification wouldn't be that necessary, but in today's world it is and it is required before you can reach your ultimate goal of getting on your airplane. Same is true with Oracle Release 12, or any automated process, authentication of the person desiring access is critical to protecting a very valuable and important asset, your enterprise data. So let's explore some techniques and mechanisms Oracle uses to check to see if you are really who you say you are, ok?

Who Are You?

Unfortunately, we probably all know someone who has been affected by identity theft, maybe some of you have been on the receiving end of it first hand! Needless to say, it's not a pleasant experience. What is the underlying root cause of identity theft? Your identity is no longer unique to you and only you, right? Someone else has obtained specifics about your unique identity and have assumed those to pretend they are you. The same is true with Oracle Release 12 if someone else uses your unique userid or if you use a generic userid that may be used by multiple people. First question to ask is "are you unique?" – you should use a userid unique to you and only you, else it is impossible to verify who you say you are. Secondly, you need to protect your identity and for Oracle Release 12 that is primarily your password – how can you keep your password from being known to someone else? How do you know that someone else doesn't already know your password? We will look at some techniques that will help.

As part of the installation of Oracle Release 12, certain generic application userid's are created that are used by various automated processes and initial administration tasks during installation. Distribution and utilization of these userids are NOT recommended and use of these userids after initial installation should be audited. Even with auditing, you can see the dilemma generic userids present in that it is impossible to know who is actually using the generic userid. All that can be determined is that the generic userid was used, with some digging, might be able to find out what the generic userid did, but the identity of who is actually entering the keystrokes cannot be determined. Exercise due diligence with these generic userids – change their default passwords and audit their utilization after the initial installation is complete.

Let's leave the application level for a minute and descend through the technology stack to the database level. Generic userids (schemas) are installed as part of the database implementation as well as part of the Release 12 implementation. A lot of these generic userids have been around for quite some time and their default passwords that are set at implementation are widely held and known in the public domain. FNDCPASS can be executed from the OS command line and by using the "ALLORACLE" parameter will change the password for all Oracle Release 12 application schemas – but be careful with this because it causes a single point of failure in that all Oracle Release 12 application schemas will have the same password and if that becomes known, access can be jeopardized for all of the Oracle Release 12 applications not just one. It is our recommendation to NOT use the "ALLORACLE" parameter and change each Oracle Release 12 application schemas one at a time – it will take more time, but will provide multiple points of failure should any one password become jeopardized.

Just like descending down a mineshaft, let's leave the application level, leave the database level and go all the way down to the network and operating system levels. The TNSLISTENER is that mechanism that sits like a sentry or guardpost waiting for someone to ask for access to the database, therefore it is very important to the overall health of the database that all accesses to the database are from trusted sources. Any unauthorized administration of

TNSLISTENER could allow non-trusted accesses to the database which would put your enterprise data at risk. Part of installing Oracle Release 12 includes establishing the "Oracle" user at the operating system level – beginning with 10g the "Oracle" user owns the adminstration of TNSLISTENER and any other operating system user who attempts to administer TNSLISTENER is denied access. There is an option to set a password for the TNSLISTENER which would then allow for other operating system users to adminster TNSLISTENER as long as they knew the password. Our recommendation is to administer TNSLISTENER through the "Oracle" operating system userid only and not allow remote authentication to other operating system userids.

Init.ora is a configuration file that specifies startup information to the database – one parameter inside the init.ora is REMOTE_OS_AUTHENT and it is recommended that this be set to "false". By setting this parameter to false you help make sure access to the database does not come through a non-secure protocol (i.e. TCP) at the operating system level or from some other remote client/host.

"Trusted Roles or accounts" have been the bane of DBA's everywhere since Sarbanes Oxley began questioning our collective ability to monitor the update activity from these trusted folks. We all know that the very nature of a DBA's job is that they must have access to all the data within the database, hence their title "Database Administrators". Through the installation process, certain operating system level accounts are established that "own" the Oracle database and by inheritance owns the ability to update any and all data within the Oracle database. It's further complicated in that Oracle requires that this "owning" operating system level account be used for database administration tasks – enter SUDO. SUDO is a command which allows the DBA's to logon to the operating system with their own unique accounts (i.e. JSPRATT01 for DBA Jack Spratt) and then switch or change user over to the Oracle operating system level account. By doing this, it provides an audit trail of what DBA's accessed the Oracle operating system level account, when they accessed the account and how long they were logged on to the account. Unfortunately, authentication ends once the SUDO command is executed – then the DBA is cloaked by the generic account and their specific activity blends in with other DBA's accessing the Oracle account. It does provide some measure of mitigation from a SOX perspective for the DBA role.

Finally, let's talk about direct client/desktop/laptop access to the database. We've all fired up SQLPlus or TOAD or SQLDeveloper to figure out why that journal isn't in the import table or to get a quick count of invoices paid last month – but when these tools are used, a direct connection is made to the database which bypasses the application level authentication altogether. First and foremost, these accesses should be limited to read-only access and only non-generic database id's should be utilized. Further restriction is available which can permit only certain desktops to establish direct access to the database and that is done through tcp parameters in SQLNET.ora. SQLNET.ora is another configuration file that specifies certain characteristics for the SQLNET network protocol which is Oracle's prescribed network protocol. By utilizing the above parameters, you can restrict direct database access to certain I/P addresses.

Password construction for your Oracle Release 12 application userid – what does it mean – how does it help prevent identity theft and keep me unique to Oracle Release 12? Oracle Release 12 provides a series of profile options that provides for password construction and strength against guessability. These settings should mirror your company's password security policy, so if you are an Oracle Release 12 System Administrator, please make sure you understand your company's password policy before setting these password profile options. Another consideration is the impact to your existing Oracle Release 12 users – most users get ruffled when their logon parameters are changed without communication and due notice – so please consider and include your users before changing these password profile options. Password construction speaks to the length, character/numeric combinations and character repeatability restrictions. Password guessability considers how easy a password is to guess if someone performed ethical hacking against your Oracle Release 12 userid and password. Finally, password aging looks at how long of life any given password can have – the longer the life, the higher the risk of passwords being guessed, communicated, observed, etc.

Beginning with Release 12, Oracle Applications provide login assistance which is a technique that everyone has probably experienced on any number of websites. Login assistance is controlled by the "Local Login Mask" profile option where the mask value will determine whether to offer forgotten password hints, forgotten application userid hints or both. The mask settings for this profile option are discussed further but wanted to highlight this feature while we're discussing authentication.

There are separate password management policies at the database level and they only apply to database userids (schemas). These database password management policies exist only in the 10g database and apply only to database userids (schemas). For in-house developed applications that utilize an Oracle database backend, database userids may in fact be application userids and then these database password policies would be all that exist for password management – but not so with Oracle Release 12. A word of caution on the recommended settings – potential conflict can happen between Oracle Release 12 applications and 10g database on these settings because the applications communicate with the database via generic database userids and thereby expect those database userids to be active. Should these recommended database password settings cause a generic database userid to expire that is used by Oracle Release 12, then this could cause an unexpected error to the applications. Conversely, having database userids where passwords never expire are NOT recommended under any circumstance. It is better to risk an unexpected error to the applications rather than have passwords that never expire.

There are measures you can take at the desktop level to make sure your password is protected and not hijacked by someone else. Who keeps a yellow sticky note under their keyboard? You can have the strongest passwords possible, but if it's "hidden" where someone can find it – it's gone and your identity has been hijacked. Screen saver timeout options should be set in the 5 to 10 minute range to prevent someone from sitting in your chair while you're away from your desk and utilizing your Oracle session which is already logged into. I realize that there is a tradeoff between passwords that are easy to remember and passwords that are hard to guess – but try and avoid your birthdate, your spouse/children's names – calendar months/dates – family member first names, etc.

One last option on authentication and that is application level timeout parameters. These parameters work in concert with each other and when these maximums are hit, Oracle Release 12 will ask the application user to login again. They will be taken to the initial login screen and once the user has re-authenticated, then they will be taken back to the form/frame where they were on when the session timed out. That's about it on authentication – now that we know who you are, you have authenticated yourself with the airport ticket agent, you've passed through the security checkpoint and have passed the gate area and find yourself sitting on the airplane. Now let's see what you can do now that you are on that airplane – are you allowed to fly the plane? Push the beverage cart down the aisle? Play electronic devices? Just what are you authorized to do now?

What Can You Do?

Ok – you've had your identity verified at the ticket counter (Oracle Release 12 application level) and at the security checkpoint (Oracle 10g database level) and finally at the gate (network and OS level) – you're on the plane, now it's a question of what can you do while you are onboard – what behaviors will you be allowed to exhibit – what functions will you be able to perform? – what buttons will you be able to push? It's all about Authorization or What Can You Do?

There are two online access frameworks within Oracle Release 12, Oracle Forms and Oracle Application Framework with Oracle Forms being the older cousin which is a technology that has existed in ever evolving fashion since the early release of Oracle E-Business Suite. One of the fundamental foundational elements of the Oracle Forms technology is application menus. Application menus define a navigable group of functions/forms and submenus which ultimately contain functions/forms at the most discrete level. Oracle Release 12 comes packaged with seeded menus, forms/functions and submenus and they usually are designed with "great privilege" from a security perspective. By that I mean Oracle groups together every conceivable function across many Oracle applications that may be required to perform a specific job responsibility, consequently there are usually a lot of segregation of duties issues when Oracle's "great privilege" meets your enterprise's segregation of duties. Care should be taken to review and analyze all the functions grouped together in any menu before assigning it to a job responsibility to make sure duties are segregated for your particular enterprise. Once duties are segregated and menus are aligned with those segregations, consider the underlying menu tables as audit candidates to make sure unauthorized menu changes don't violate approved segregations.

Data groups is a handshake entity between Oracle Release 12 applications and the application schemas installed on Oracle 10g database. Basically a data group defines all the database schemas that can be accessed from any given responsibility and it's associated menu and request group. Oracle Release 12 comes seeded with the "Standard" data group which brings together 229 application schemas that belong to the APPS id (APPS id is the database id that owns all the Oracle Release 12 application schemas in the database) and unless you have some very hard and

specific segregation of duties, it is recommended that all your responsibilities point to the "Standard" data group that comes packaged from Oracle. Otherwise, you will be faced with some healthy function and report analysis and will be fighting the vagaries of data normalization which is rife throughout Oracle Release 12 application tables. It should also be noted that beginning in Oracle Release 12, data groups are not referenced nor used by Oracle Applications Framework which is the "other white meat" when it comes to online access – OAF will be discussed later.

Request groups are another foundational entity that relates to batch reporting – a request group brings together all the reports and batch programs that can be executed by any given responsibility. Just like menus group together online access through forms/functions, request groups bring together all the reports and batch programs that can be executed through any given responsibility. Just like menus, Oracle Release 12 installs request groups for each application and seeded responsibility. Request groups should be analyzed from a segregation of duties perspective and also from a sensitive data perspective to understand what data is being displayed on the reports in any given request group. Also, realize there are some very powerful batch update processes that are assigned to the various request groups across all the applications – you should know where these powerful batch processes (i.e. Depreciation for Oracle Fixed Assets) exist and make sure they align with your enterprise's duty segregation. Be aware also that if you choose to utilize a non-standard Data Group, where not all Oracle Release 12 application schemas are included and a report assigned to a Request Group needs access to an application table not included in the Data Group, the report will not function properly. The Data Group and Request Group specified for each responsibility need to be aligned to ensure data access is given to all the application database schemas needed by the reports.

Application responsibilities are an intersection between menus, data groups and request groups – a responsibility defines what a user can do, what data a user can see/update and what data came be reported. The design intent for responsibilities is that they would closely mirror job duties within the enterprise, albeit segregated ultimately by each Oracle application. Responsibilities can belong to one and only one Oracle Release 12 application. Please note that there is exclusion funtionality which allows certain submenus and/or forms/functions to be excluded from the menu tree – so these exclusions should be included in any segregation of duties analysis performed by your enterprise. Also note that once a responsibility is saved, it can't be removed, it can only be end-dated or inactivated. The underlying responsibility table should be considered an audit candidate to ensure segregation of duties remain in tact. The same comment on "great privilege" holds here as it did for menus – Oracle seeded responsibilities have "great privilege" and it is recommended that seeded responsibilities NOT be used and that custom responsibilities and maybe even custom menus be developed specific to your enterprise's duty segregation. Utilization of seeded responsibilities is a huge red flag for I/S auditors!

Before we dive into RBAC and UMX – let's recap, there are two online access methods, Oracle Forms and Oracle Application Framework – Oracle Forms has been around forever and OAF has it's roots in the self service applications. Some applications that were originally developed in Oracle Forms have been rewritten in Oracle Application Framework for Oracle Release 12 (i.e. Cash Management). With the advent of the newer technology and SOX asking a lot of questions about role based security – Oracle has embraced RBAC as a new standard for authentication and authorization. Responsibilities, menus, request groups and data groups all serve to create a navigable group of functions that are then assigned to an individual user. Oracle User Management - UMX (which is the application that delivers RBAC functionality) utilizes roles, data security policies and grants to define what a user can do across the applications. Roles can contain responsibilities, but can also contain data security policies that can restrict access to specific rows and columns within a certain application table. Conversely, responsibilities can only restrict down to the function level and unless personalizations are implemented, have a hard time restricting access within a table structure.

Some RBAC/UMX building blocks are data security policies and grants. An example of a data security policy would be to restrict all users associated with a particular role to only 1 or a group of specific legal entities when they are defining a bank account in Oracle Payments. By the way, beginning in Oracle Release 12, bank accounts have been elevated and integrated with the Trading Community Architecture (TCA). If the data security policy restricts to a specific legal entity, then that would be called a data instance – if the data security policy restricted to a group of legal entities, then that would be called a data instance set. Once the data security policy is defined per business requirements, then it is "granted" to specific users. Further "grants" can be used to grant users access to RBAC/UMX roles and Oracle Forms responsibilities.

Part of the Oracle Release 12 login form now can provide login hints (i.e. forgotten userid and password) – this is driven by a profile option called "Local Login Mask". To determine the mask value, determine the desired login attributes, add up the attribute values for those attributes and place that summed value into the profile option mask value.

Per MetaLink, not all product teams have shipped security wizards – as a precursor to whet our collective UMX appetites, Oracle Release 12 has shipped two security wizards, one for general role administration and one specifically for the Cash Management application.

Beginning with Oracle Release 12, a combination of profile options and a security profile, now allows you to enter transactions and report on multiple operating units without being forced to change responsibilities. Just make sure that Muiti-Org Access Control (MOAC) doesn't override any important segregation of duties from an enterprise perspective. Underlying table for security profiles (PER_SECURITY_PROFILES) should be considered an audit candidate.

Remember earlier we discussed individual desktop access through TOAD, or SQLPlus or Oracle SQLDeveloper? All these tools allow individual users to connect directly to the database and execute SQL queries (hopefully only select statements!!!) directly against the database, bypassing all application authentication and authorization. We also discussed the generic database id's that get installed as part of the Oracle Release 12 and Oracle 10g implementation and how that those generic database id's need to have their default passwords changed because they are part of the public domain and subject to hijack/unauthorized access. Now let's turn our attention to non-generic/specific database id's that may be defined for individual users. Database privileges should be examined for all database ids which are found in the DBA_USERS table – if you don't have access to this table, ask your DBA to pull a list of all database id's in this table. The next step is to understand any and all privileges assigned to these database id's and that is done by examining DBA_SYS_PRIVS, DBA_ROLE_PRIVS and DBA_TAB_PRIVS. System privileges give the database id or user the right to perform a particular action or to perform a particular action against a specific database object. System privileges such as alter, create, drop, become user, update, execute, lock and insert can all have catastrophic impact if performed against an Oracle Release 12 application table or PL/SQL package. For example, the DROP privilege, basically removes whatever object is dropped – so if you drop schema GL – you have lost all your tables for Oracle General Ledger. Execute allows any procedure, package, trigger or any executable piece of code that resides in the database to be executed – there are thousands of PLSQL packages that are part of Oracle Release 12 – execution of any piece of code outside the context of the Oracle Release 12 application can have catastrophic impact. Privileges also exist at the table level which can be seen by examining DBA_TAB_PRIVS where you should look for any Oracle Release 12 application table that might be specified here (especially if the privilege is INSERT, UPDATE, DROP or ALTER). Last thing worth mentioning are ROLES – Oracle 10g allows for roles to be defined, privileges assigned to the role and then roles assigned to database ids – so you need to examine the roles as well as database id's since role privileges can be inherited by database id's. Needless to say, updates at the database level are no laughing matter and should only be performed under the auspices of Oracle Support or you could lose your support license and cause your Oracle Release 12 to not function properly.

When you boil it all down, the Oracle 10g database which contains a majority of Oracle Release 12, is comprised of physical files. Permissions and ownership of these files are critical to avoid unauthorized access at the operating system level. Ownership should be either DBA or SYSTEM which means that these files are locked down and only available to the Oracle DBA's. Directories of interest are defined by the environment variable $APPL_TOP or beginning with Oracle Release 12 $INSTANCE_TOP. These environment variables will point to specific physical directories which contain all the physical files that comprise Oracle Release 12 – permissions on these directories, sub-directories and contents should be examined to make sure they cannot be updated. The UNIX octal permissions 3 sets of permissions for 3 different groups – let's take 751 as an example: First number 7 relates says what the owner of the file can perform, second number 5 says what the group the owner belongs to can perform and the last number 1 says what other users can do. Then if you break down each individual number itself, the attribute values are:
0 = all types of access denied, 1= execute access only, 2 = write access only, 3=write and execute access, 4=read access only, 5=read and execute access allowed, 6=read and write access allowed and 7=all access allowed – so back to our example, 7 indicates all access allowed for owner of the file, 5 indicates read and execute access allowed

to the group the owner belongs to and 1 indicates execute access only for rest of world. One note before we leave the operating system level – there is a PLSQL function called UTL_FILE_DIR which is used to create flat files to the operating system and the directories that can receive these flat files are specified in the init.ora file – check to make sure that "*" isn't specified as a receiving directory which means flat files can be created out of Oracle Release 12 to any directory on the server. That could mean sensitive data might come to rest in a directory where unauthorized eyes can view.

You may or may not have run across some of the forms that accept free form SQL statement. They are usually setup forms of some kind and are found across the entire application suite. Know that Oracle Release 12 communicates with Oracle 10g database through one database id, namely the APPS id and the APPS id has total access, insert, select, update and delete to all base tables that comprise Oracle Release 12. Since some forms accept free form SQL, care should be taken to make sure that the specified SQL isn't pulling back data that is sensitive or that the SQL isn't performing any update statements. Most of the forms do a good job of parsing and editing the free form SQL and most require a specific value to be returned, but just something to examine or audit the next time you're in the application.

What Can You See?

Ok – time for another recap – you've made it through the airport identity authentication to your airplane and we just finished covering what you are allowed or authorized to do onboard the airplane – sometimes just seeing something isn't authorized. The airplane metaphor kind of breaks down here, but maybe a person next to you has some confidential, corporate insider information on their laptop, but if you twist your head to a certain angle, you can make out what is there. Same holds true with our enterprise data, where there are ever increasing penalties and loss of goodwill facing most enterprises should their sensitive information fall into the wrong hands. Let's look at some techniques for protecting the sensitive data for your enterprise.

Obfuscation sounds like a song out of a hit Broadway show, doesn't it? To obfuscate means to confuse, bewilder or otherwise mask what's real. Each company needs to go through the exercise of identifying all data that is sensitive to them and then understand where that data comes to rest, especially across the Oracle Release 12 E-Business Suite. There are no short cuts on this exercise – talk with your legal department, tax department, HR department and key business groups to gain understand what data is sensitive for your organization and then once it's identified, make sure you understand where that data comes to rest. Let's consider your non-production environments for Oracle Release 12 – do you know how those environments are created? Usually from a clone of production – maybe there are some profile options turned off, some email addresses that are redirected – but after that, the test environment is turned over to a project team to use, sound familiar? What about all that sensitive data that just got cloned from production into that new test environment? You need to take measures to obfuscate your sensitive data in your non-production environments. There are several products on the market and these tools can assist in changing or obfuscating the sensitive data from the real value to a fake value. Solution Beacon has a PLSQL procedure on our website that will replace sensitive data with meaningless data values and it can be found at http://www.solutionbeacon.com/security.

Oracle Release 12 provides personalization mechanisms, one for forms that utilize Oracle Forms and one for forms that utilize Oracle Application Framework. Both of these personalization mechanisms will allow specific data on the form to be hidden based on predefined criteria. So rather than obfuscate some sensitive data, you could develop a forms personalization that removes the sensitive data from the form and thereby can't be seen. Word of caution though – this only mitigates unauthorized viewing at the application level – it does nothing for the database level – should users know what tables contain sensitive data and they can establish direct connections to the database through TOAD, SQLPlus or Oracle SQLDeveloper – then sensitive data can be queried through these direct connections. Understand that we are just as vulnerable if not more so from internal threats than we are from external threats.

Don't forget work tables, descriptive flex fields and flat files that are written out to the operating system directories. As mentioned earlier, if the UTL_FILE_DIR parameter in the init.ora configuration file points to a wildcard "*" – then flat files can be created out of Oracle Release 12 to any directory on the server – these flat files may contain sensitive data – do you know who can see those flat files? Would encourage you to examine your database level

roles and privileges to see who can see the tables that contain sensitive data – also check your Oracle Release 12 responsibilities and UMX roles to understand who can access the tables that contain sensitive data.

Bank account numbers, routing numbers and account holder names are sensitive for most organizations and beginning in Oracle Release 12, banks are elevated and integrated with the Trading Community Architecture (TCA) within the new Oracle Payments application. One of the features of Oracle Payments is that it provides for the encryption of bank account information – this encryption should obviously be in force and utilized. Consider the underlying table IBY_BANKACCT as an audit candidate.

Additionally, credit card numbers are sensitive to most organizations. Credit card regulatory agencies impose stiff penalties for misappropriation of their credit card numbers. Oracle Payments provides encryption for credit card numbers which should be turned on. Further, the table containing credit card numbers, IBY_CREDITCARD, should be considered a candidate for auditing.

What Did You Do?

Ok – back to our air adventure metaphor – the last thing that could be called into question during a flight is to understand what you behaviors you exhibited while onboard the airplane. Perhaps there was some type of seat cushion damage reported or smiley faces were drawn with a Sharpie on the beverage cart – whatever the issue, it may be good to have an indication of what you did during the flight from a behavior perspective and that is where auditing comes in. The best controls are proactive controls because they catch issues while they are happening; however, auditing by nature is detective in that it depends on audit trails and artifacts to recreate past behaviors. Kind of like CSI only constrained to the safe confines of Oracle Release 12!!! So let's look at some of our options for performing that detective work that everyone loves so much – auditing!

Application level auditing is controlled by two profile options and it is RECOMMENDED that the audittrail be activated and that the audit level be set to the FORM level to allow for tracking from point of initial user login. Audit reports should be reviewed at least quarterly and evidence of the review needs to be filed away as audit artifacts. More frequent reviews of the audit reports should occur for changes to sensitive data – remember your enterprise is only one unauthorized select statement away from a public relations nightmare!

Database level auditing should also be deployed as a way to mitigate risk over trusted roles, such as DBA's and operating system administrators. Activity can occur against key tables directly to the database, bypassing the application level altogether, but by having database level auditing turned on this provides a way to detect this activity.

Conclusion and Recommendations

As you can see, there are many mechanisms throughout the Release 12 technology stack that provides for the security of enterprise data. Organizations have significant investments in security to protect from external threats; however, there are any number of studies that submit that similar investments have not been made to protect against internal threats. The data contained within the E-Business Suite footprint is critical to their organizations and consideration of the mechanisms mentioned in this presentation will mitigate internal threats. Do not allow your organization to join the role call of companies that have had their sensitive data jeopardized, because that will force an investment that far exceeds the cost of implementing the best practices recommended in this study.

References

- Better Release 11i Security in 3 Days – Keeping The Bad Guys Away (Part II) – Randy Giefer, Solution Beacon – Collaborate 07
- Application Security – What Are My Options – Susan Behn, Solution Beacon – Collaborate 07
- MetaLink Doc. ID: 189367.1 – Best Practices for Securing Oracle E-Business Suite Release 12
- Solution Beacon Security Pocket Guide

- MetaLink Doc. ID: 260986.1 – Setting Listener Passwords with 10g Listener
- MetaLink Doc. ID: 385445.1 – Oracle Applications Framework Profile Options for Release 12

