Raduga

User Guide

Raduga 1.08.0001

Table of Contents

General information

Copyright

Contacts

For any questions and support regarding this product, contact Michael Dvorkin (+79185402272, support@LazyDeploy.com).

Licensing

Raduga Free software can be used for free. It is restricted to 5 environments and 50 projects. Free edition has a limited technical support.

Raduga Pro software can be used for free during the trial period of 30 days. After the end of the trial period, you must install a private license for each user to continue using the software. Raduga Pro can manage an unlimited number of environments and projects and it has full technical support.

Contact Michael Dvorkin (+79185402272, support@LazyDeploy.com) to obtain Raduga licenses.

Disclaimer

Raduga allows deleting database and file system objects. In some cases the objects are replaced during the migration of development projects. Raduga users should carefully test all development projects in a test environment before implementing them in production. We accept no liability for any damage caused by the Raduga application. Object transmission cannot be guaranteed to be secure or error-free, as migration rules can differ from one environment to other. We therefore do not accept liability for any errors or omissions in the contents of custom objects which might arise as a result of object transmission. Although we have taken reasonable precautions to ensure proper performance of Raduga software, the company cannot accept responsibility for any loss or damage arising from the use of Raduga.

About Raduga

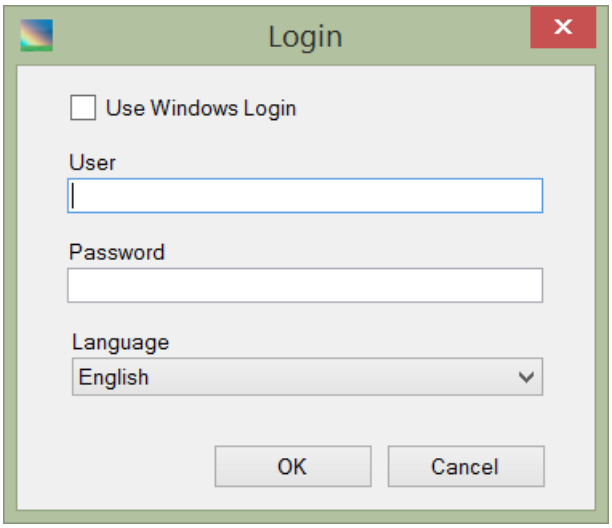
Raduga is an application that helps you manage the development and deployment process. It is designed for Oracle applications; however, it can be used in any development environment. A user-friendly interface, easy navigation between applications and projects, various migration and deployment capabilities, version control and reporting make Raduga a useful tool for programmers, team leaders and project managers.

Raduga offers to users

- Object migration between environments
- Intuitive navigation between entities
- Object comparison
- Version control and deployment history
- Monitoring environment status
- Starting/stopping environments
- Data loading capabilities
- Easy customization
- Comprehensive reporting
- File transfer capabilities
- Enhanced security

Getting Started with Raduga

After launching Raduga you see the “Login” screen:



If you don’t have a Raduga user account, ask the Raduga Administrator to create an account for you.

Supply the user name and password that you got from Raduga Administrator, choose your interface language and press OK. If your user is defined as LDAP authenticated user you should use your organizational credentials to login to Raduga.

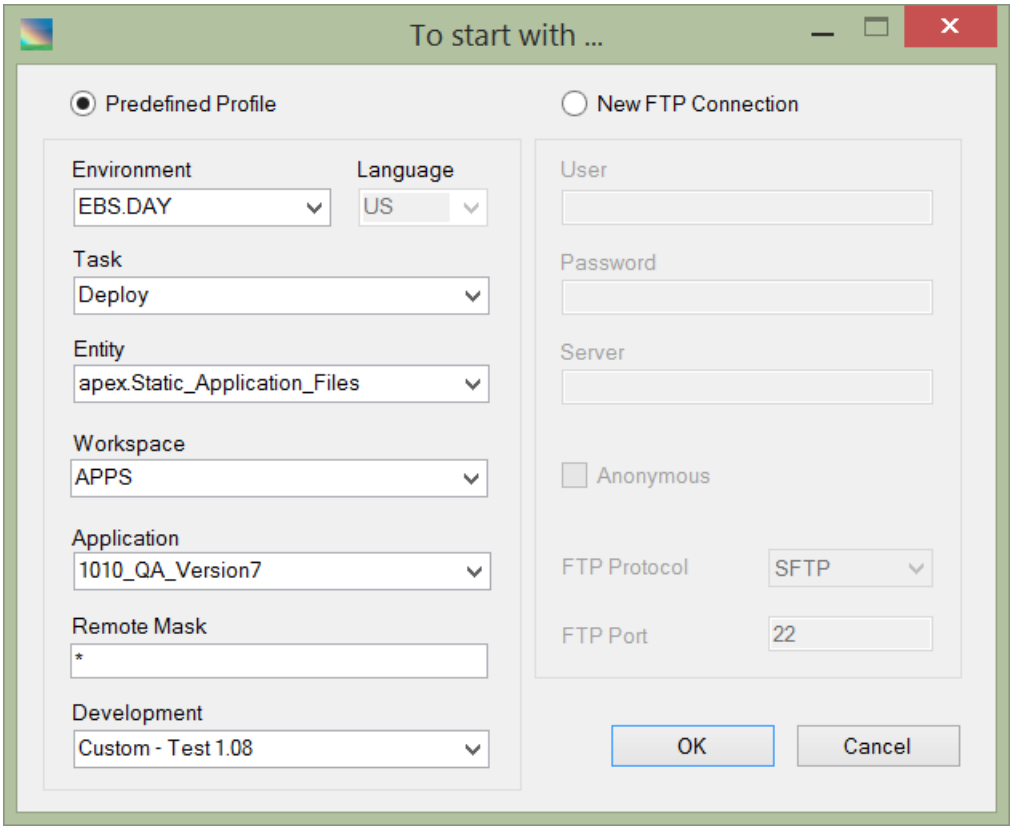
If the Raduga Administrator has enabled it, you will have the option to log in using your Windows user name. In this case you will not have to supply a password. Check the “Use Windows Login” check box in order to use this option.

After you have logged in, a screen appears where you can choose your Raduga working environment.

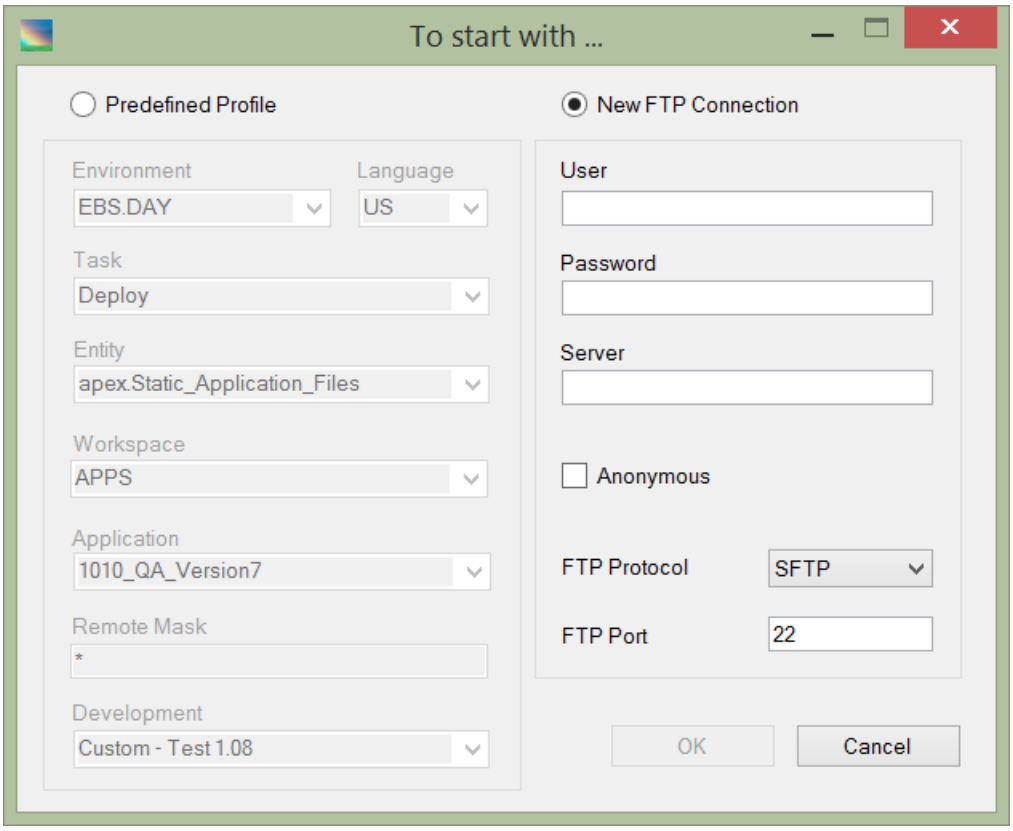
Selecting a Working Profile

Raduga lets you define a working environment before starting the session. You can choose appropriate values using the "To start with…" form.

If you select "Predefined Profile" you will be able to choose the environment from the list of environments defined by Raduga administrator to which you have access:



If you select “New FTP Connection” you have to enter the user name, password and server information for the FTP connection:



In this case Raduga will work as a pure FTP client and will be missing many of its features.

Press “Cancel” to start Raduga without connecting to any environment.

Choosing an Environment

Select “Predefined Profile” and use the “Environments” drop down list to choose an environment to work with. In the list you only see those environments that you have permissions to work with. Raduga directs you to the selected environment.

Choosing a Task

Raduga allows you to choose one of the following tasks:

- *FTP*
- *Deploy*
- *DataLoad*

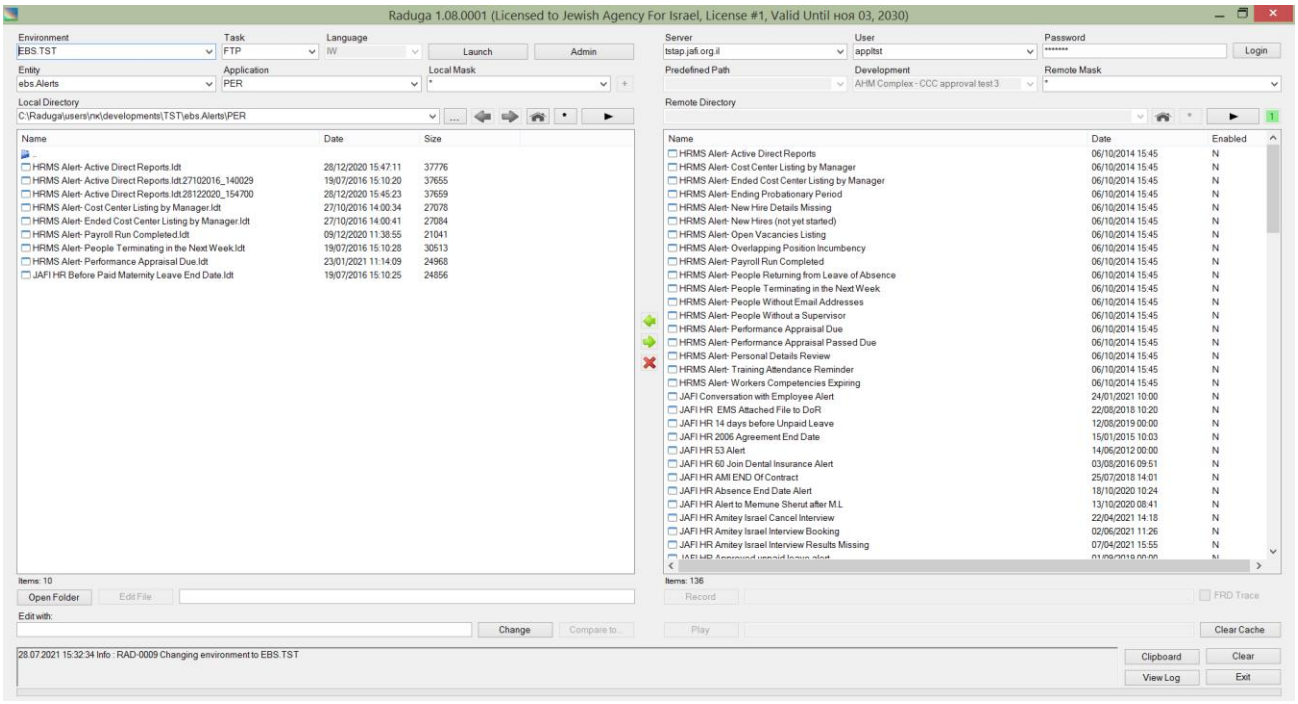
Depending on the task you select, you get access to different sets of Raduga entities.

The *FTP and Deploy* tasks let you transfer objects between local and remote computers. All objects are divided into existing and generated objects:

- Existing objects are regular files
- Generated objects need to be downloaded from the database or other system in order to become a regular file

For example, Oracle forms, reports, and scripts are existing objects. However database packages do not exist as regular files until you download them from the database using a SQL query. The same is true for different Oracle Application entities, like alerts, programs, flex fields, etc. Raduga lets you work with generated entities by transforming them into ordinary files.

As an example, you can see the list of Oracle Applications Alerts on the right (server) side of the application:



You can drag them to the left (local) side. During this operation (a “get” operation), the object is downloaded and stored as a file with extension “.ldt”.

On the left panel, you can see a list of locally stored alerts. Local objects can be transferred to the server side by dragging them into the right panel. During this operation (a “put” operation), the object is compiled in the database.

The *Deploy* task also lets you add objects to the “Development Project” panel. A Development Project can include objects from different environments, applications, entities and languages. The Development Project is used to deploy its objects in a specific environment.

There are objects that are not available for deployment, including concurrent request log, and output files. These objects are found under the FTP task and not accessible in the Deploy task.

The *DataLoad* task lets you record actions performed in Oracle Applications into a special file. The DataLoad task can work in two modes:

- Recording a Data Loader file
- Recording an FRD Trace file

In order to record an FRD Trace file, select “FRD Trace”.

Recorded data loader files can be transferred to private working area and edited using “DataLoad Classic” or “DataLoad Professional”.

The data loader files can be used for playing actions recorded in them in the Oracle Applications environment.

Selecting an Entity

Clicking on the “Entities” drop down will display a list of the entities available for the current Raduga user. The list of entities depends on the task you have chosen and on the permissions that Raduga Administrator granted to you.

An entity is a Raduga object. Examples of the entities are ebs.Profiles, ebs.Forms, db.Packages, and db.Sequences. Raduga comes with pre-installed entities that define file system, database and E-Business Suite objects. Entities can be edited (customized) using Raduga's built-in editor. A Raduga developer can also create new entities to satisfy company business needs.

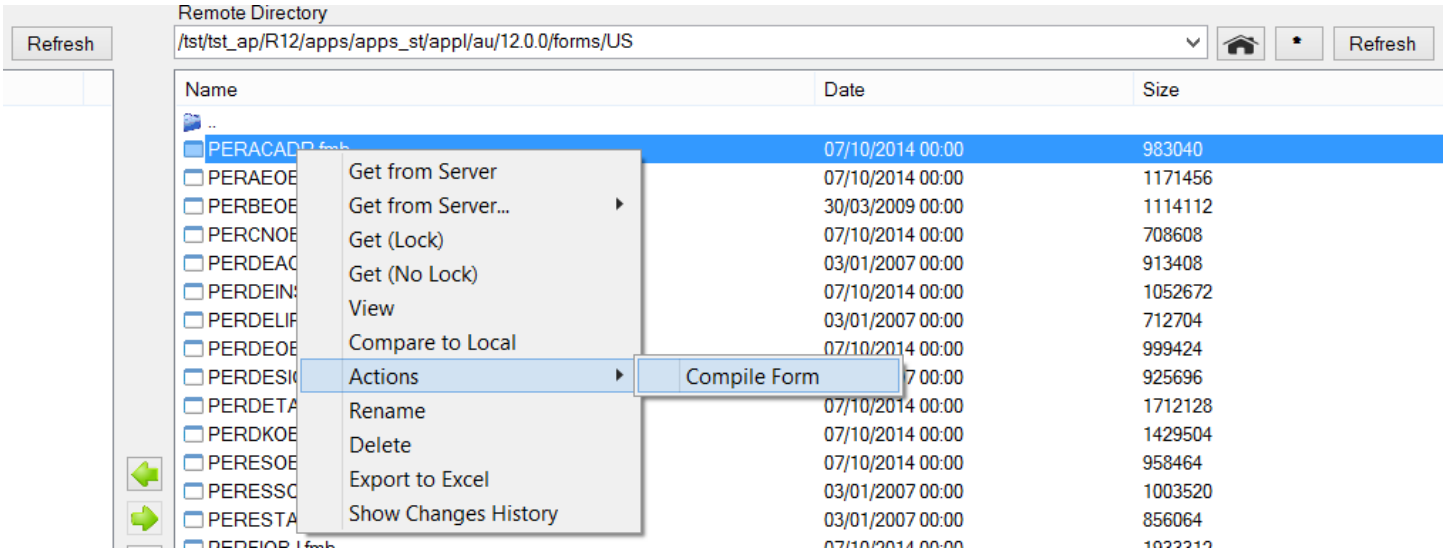
When you choose an entity, Raduga refreshes local and server paths and populates the panels with objects belonging to the entity.

For each entity you can define local and server masks. The default mask for many entities is “*” (showing all objects). However you can change the mask to shorten the object list. For example, changing the server mask to XX* will show on the right panel only objects with names starting with XX.

For some entities the “Application” and “Language” fields are available. This means that the current entity depends on the Application Module and/or Language. For example, Oracle Applications Menus do not depend on the Application, so the “Applications” field is greyed out; however they do depend on the language so the “Language” field is available. By choosing a specific application or language you will cause Raduga to show only objects belonging to that application or language.

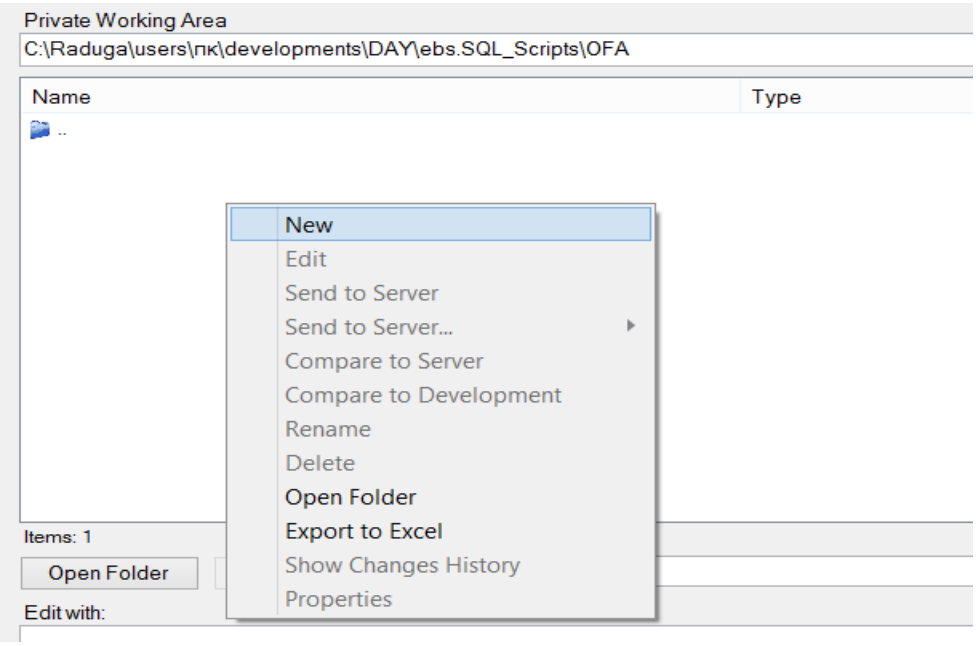
For some entities the “+” sign near the “Local Mask” textbox becomes available. Press the “+” sign to enter additional values that can influence entity behavior. For example, for Framework objects the remote path can be changed according to the value of the custom application. When you press on “+” sign, Raduga opens a window that lets you enter custom values. When you are done, the right Raduga panel refreshes to reflect the changes.

For some entities the “Actions” field in the context menu becomes available. For example, for the “ebs.Forms” entity you can use the “Compile Form” action to compile the form:

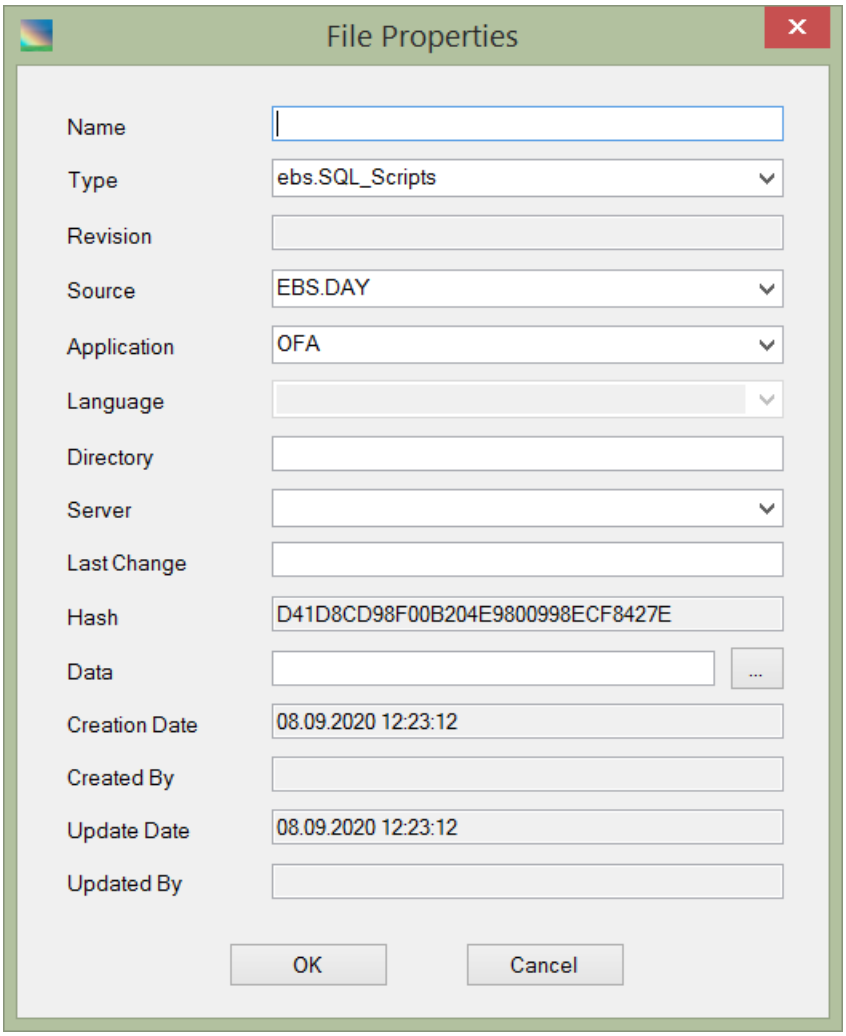


Adding new entity object

To add a new entity object, right click on the empty space of Raduga left (Private Working Area) panel and choose “New” in the list:



The “File Properties” form will open. Fill necessary values and click OK:



The empty file will be created and opened in the default editor. Edit the file with the desired content and save. This will create a new entity object that can be deployed using Raduga.

Choosing a Language

If the current entity depends on language, the “Language” drop down field is available. “Language” here refers to the Oracle Applications language (not the Raduga language that defines only text in the Raduga interface). US English is usually the default. Other languages can be installed by your Oracle Applications DBA. If an additional language is installed, most Oracle Applications objects get their translated versions. Choosing a national language in Raduga will show the translated object names wherever they are available.

Choosing an Application

Some entities are application dependent. For example Oracle Applications Alerts are defined for specific applications (PER, PO etc.). When you choose specific application in the “Applications” field Raduga will display all objects belonging to this application.

Defining a Mask

You can use * or % signs to filter the display of server objects. For generated entities (database objects) you can use SQL “like” statement escape sequences. For existing entities (files) you can use UNIX wildcards. For local objects you can use both * and % signs as well as Windows wildcards. After altering a mask, press the corresponding “Play” button to refresh

the object list. When remote object list is being refreshed the count of remote items is updated and the “Play” button changes to “Pause” button. Press “Pause” button to pause updating the remote objects list.

Using Predefined Paths

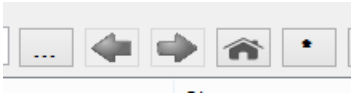
Raduga developers can define predefined paths for your convenience. Choosing a path in the “Predefined Paths” field will change the directory on the remote server to that path and refresh the object list on the server panel. Predefined paths are available only for the “FTP” task.

File System Navigation

For existing entities in “FTP” mode you can change the working directory on the server by double clicking the folder icons in the server panel. This action will make the selected folder your current directory and display in the server panel the list of objects in that directory. Click “..” to move one level up in the file system tree. On the local (left) panel the navigation is always available and does not depend on your current task and entity. The only exception is the “restricted” entity that does not allow navigation both either Raduga panel for security reasons. Press “Home” button to return to the entity’s default home directory.

Back & Forward History

You can navigate between environments and entities using Raduga history navigation buttons



Raduga will remember all previous navigation paths and you can go back and forward without need to choose environment and entity.

Transferring Objects

In both the FTP and Deploy tasks you can transfer objects to and from Remote Server and Private Working Area. Raduga defines the correct paths and actions for each object depending on its entity.

Here is a list of actions performed during a “get” operation (Private Working Area ← Remote Server):

1. **Download an object** (only for generated objects)
   Raduga downloads the object from the database or other system according to the defined rules, and creates a regular file on the remote server.

2. **Check out an object from the Revision Control System (RCS)**
   Raduga checks the object out of the RCS. Raduga adds the object to the RCS if it didn’t already exist there. If the object exists in the RCS, you are trying to lock it and it is locked by another user, Raduga displays a message about the lock and interrupts the “get” operation.

3. **Save an existing local object before overwriting it**
   If the user has checked the “Backup” check box in the user preferences window, Raduga archives existing local objects before overwriting them during the “get” operation.

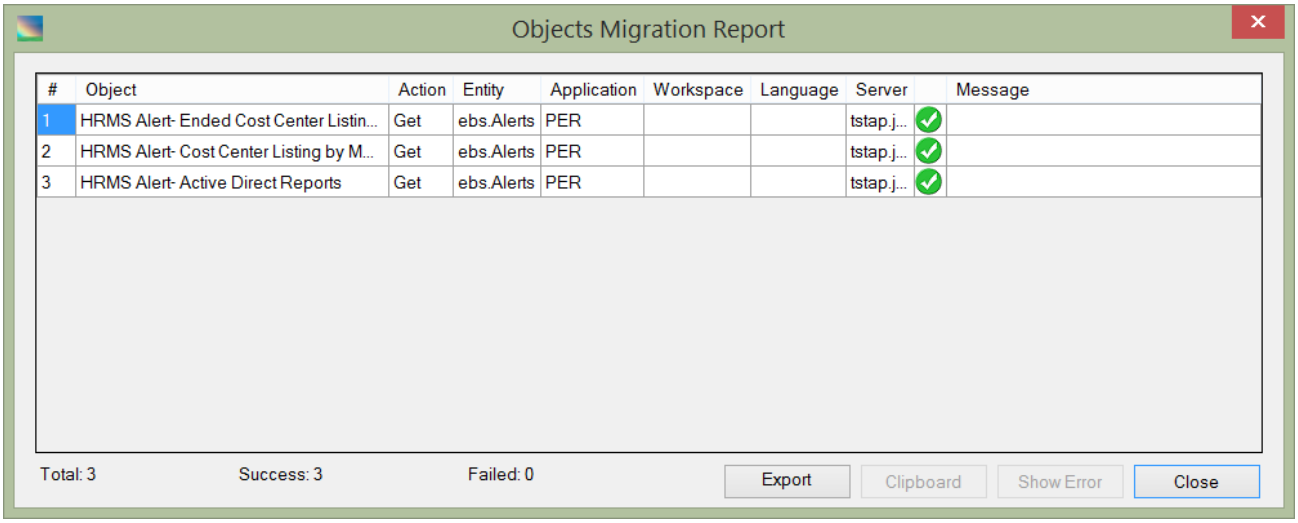
4. **Copy the object from the remote system to the local computer**
   Raduga uses the FTP or SFTP protocol for this action according to user’s definitions

5. **Install the object**

   Raduga places the object as a regular file on the local file system in location specified for it. The object's properties are recorded in a special hidden "properties" file in the same directory.

6. **Report**

   Raduga displays the "Objects Migration Report":

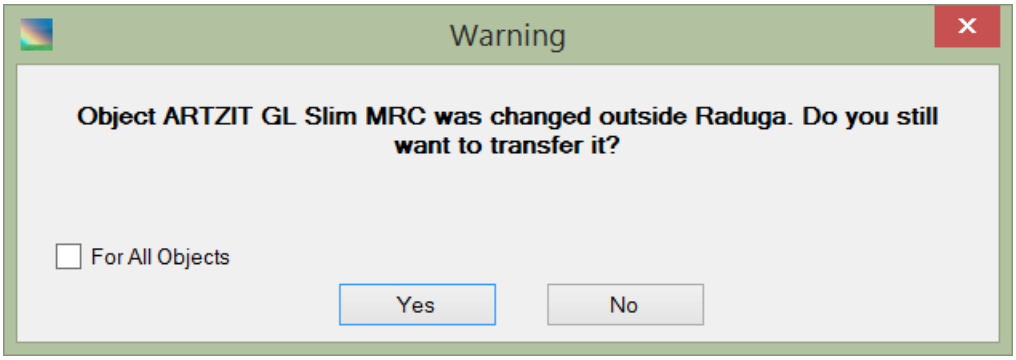


   The report shows the object migration status as well as error messages if the migration fails.

   Here is a list of actions performed during "put" operation (Private Working Area → Remote Server)

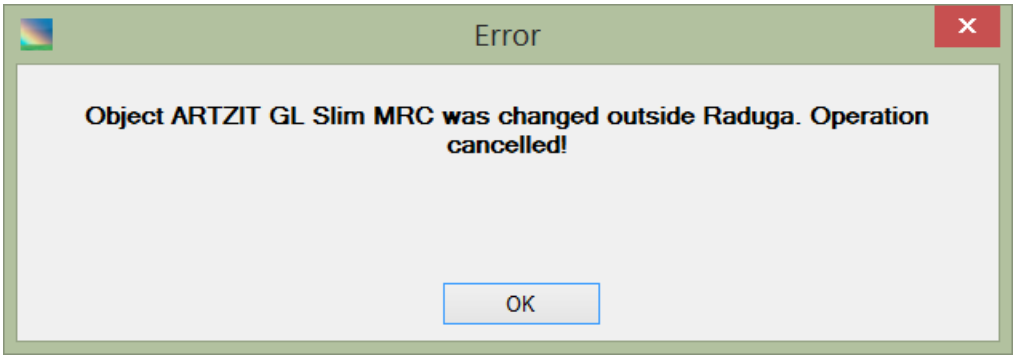
1. **Check whether the object has been changed**

   Raduga always checks whether the object that is transferred to the server has been changed. Raduga allows transferring changed files only if the current user is an Administrator or Developer or Developer Team Leader, and displays a warning message about the file:



   If the current user is not an Administrator or Developer or Developer Team Leader Raduga does not allow transferring changed files to the server. The only exception is that in the "DataLoad" task Raduga displays a "File changed" warning message and allows transferring changed files for all users. Transferring changed files during development project deployment is allowed only if the current user is an Administrator.

Here is the error message that is displayed when the object was changed outside Raduga and the deployment is not allowed:



To deploy this object you must login to Raduga as Administrator or get the object from the environment to abandon manual changes.

2. **Copy the object to the remote system**
   Raduga uses the FTP or SFTP protocol to copy the object, according to user's definitions.

3. **Check the object into the Revision Control System (RCS)**
   Raduga performs a check against the RCS, and adds the object to the RCS if it does not already exist there. If the object exists in the RCS and is locked by another user, Raduga displays a message about the lock and interrupts the "put" operation.

4. **Upload object** (only for generated objects)
   Raduga uploads the object to the database or other system according to the defined rules.

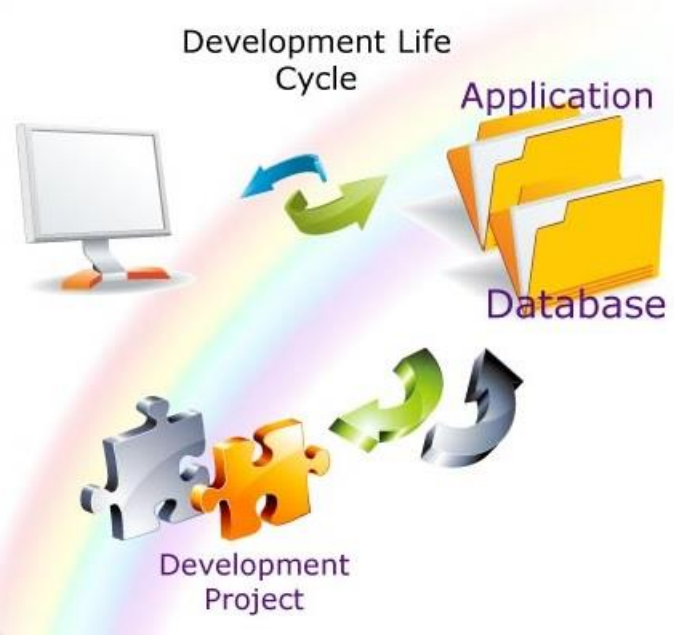
5. **Install the object** (only for existing objects)
   Raduga places the object as a regular file in the specified location on the remote file system.

6. **Report**
   Raduga displays the "Objects Migration Report", as described in the previous section. The report shows object migration status, as well as any error messages if the migration fails.

Working with Development Projects

"Deploy" mode provides a "Development" panel, where you can save objects belonging to a specific Development Project. One Development Project can contain objects of different entities brought from different environments.

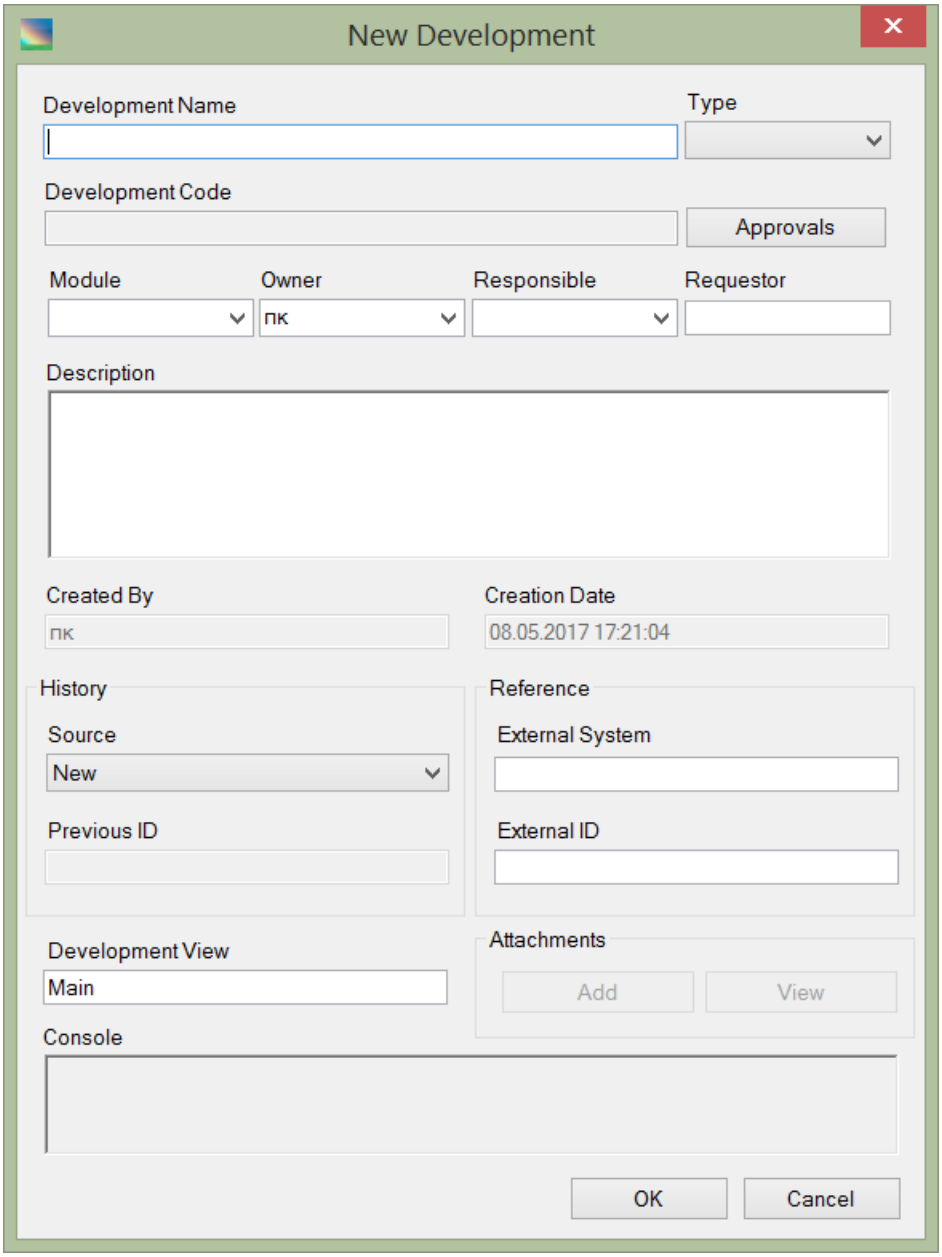
Development Projects are necessary for carrying out the Development Life Cycle:



Developers can copy objects from the Server Environment (database or application) to the local computer (Private Working Area) and update them. When the objects are ready for testing, the developers can move the objects back to the Server Environment and add them to the Development Project. Use the Development Project to move objects to and from the other environments.

Creating a New Development Project

To create a new Development Project, press “New” on the “Development” panel to display the “New Project” window:



The “New Development” form has these fields and controls:

Development Name The name of the development project

Development Code Automatically filled: <Module> <External Id> <Type> - <Development Name>

Type Development Type.

	Valid values include: Complex, Custom, Discoverer, Form, Personalization, Interface, OAF, OAF Personalization, Other, Report, Workflow. You can custom development types or edit existing ones.
Module	Application Module
Owner	Development Project owner
Responsible	Development Project responsible owner
Requestor	Development Project requestor
Description	Development Project description
Created By	Development Project creator
Creation Date	Development Project creation date
Source	Development source **New** – for new projects **Migration** – for imported projects
Previous ID	Development ID in the source system (for imported projects only)
External System	Reference to external project management system (optional)
External ID	External System ID (optional)
Development View	Development View name (see “Adding a Development View”)
Attachments	“**Add**” – attach a document to the Development Project “**View**” – view attached documents

When the form is complete, press “OK” to create the new, empty project.

A Development Project consists of the project directory with a name similar to the project code, and an XML file containing information about the project. For example, if all projects are saved in the directory \\server\Raduga\Developments and the project name is “PER Workflow - Adding Optional Approvers”, then the project directory will be \\server\Raduga\Developments\PER Workflow - Adding Optional Approvers. Raduga places a file, named in this case “PER Workflow - Adding Optional Approvers.xml”, in the \\server\Raduga\Developments\PER Workflow - Adding Optional Approvers folder.

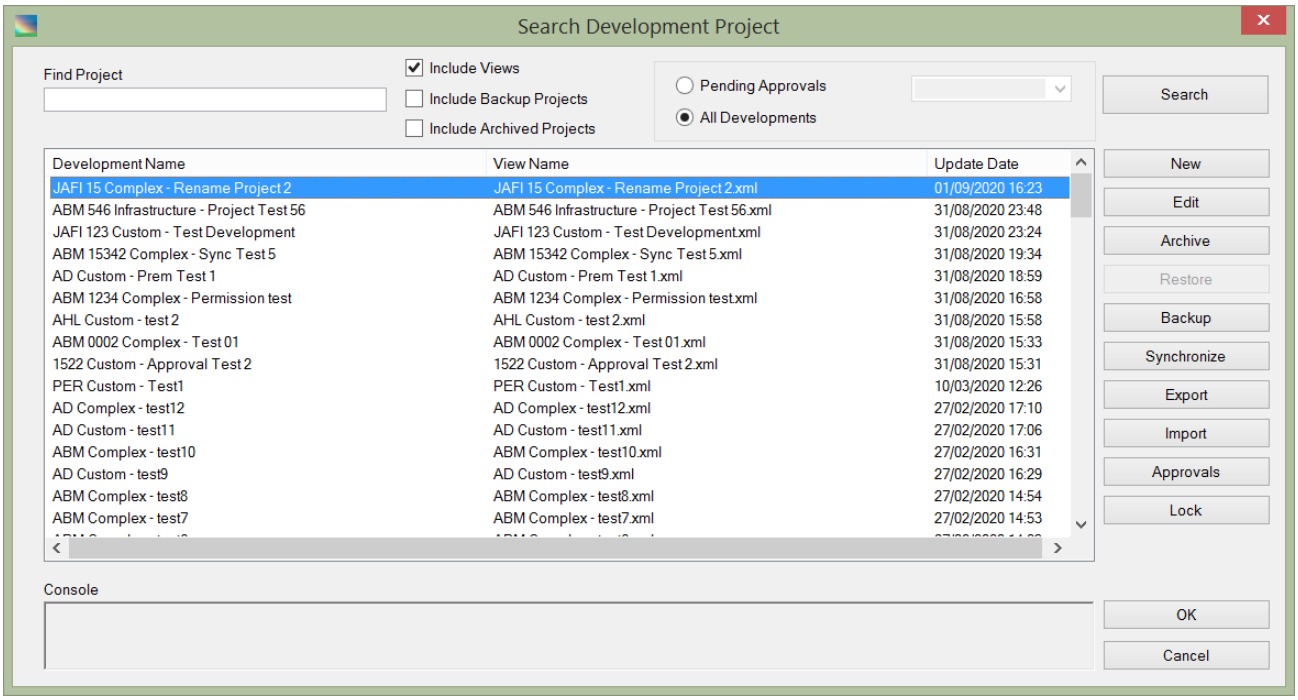
Raduga defines the default development project directory (in this case - \\server\Raduga\Developments) in the PROJECTS_DIR custom constant.

Editing a Development Project

To edit current project, press “Edit”. The “Edit Development” window open and you can update the development project's attributes.

Opening an Existing Development Project

Select “Open” to display the “Search Development Project” window. Click “Search” to display a list of all development projects. Refine your search by entering part of a project name in the “Find Project” field.



The “Search Development Project” form has these fields and controls:

Find Project Enter pattern and press “Search” to filter the displayed project’s list items

Include Views Choose to include development project views in the project’s list

Include Backup Project Choose to include development project backups in the project’s list

Include Archived Projects Choose to include archived project in the project list

Pending Approvals	Choose to show only projects that require approval by certain users. Select the approving user or “All” in the drop down box near the “Pending Approvals” check box.
All Developments	Choose to include all development projects in the project’s list
Search	Click to execute the search according to the specified criteria
New	Create a new development project
Edit	Edit the development project
Archive	Delete the development project view. Note that only project’s XML file is marked as archived. The project objects are not deleted from the disk.
Restore	Restore the selected project. This button is available only for deleted projects
Backup	Preserve the current state of objects from the current environment before overwriting them with further edits or developments.
Synchronize	Replace all project objects with objects taken from the current environment
Export/Import	Export/Import entire development project
Approvals	Open the “Deployment Approval” form for the selected project
Lock/Unlock	Lock/Unlock the project. This button is enabled if you are the user who is locking the project or you have administrative privileges

Select the project you want to open and press “OK”.

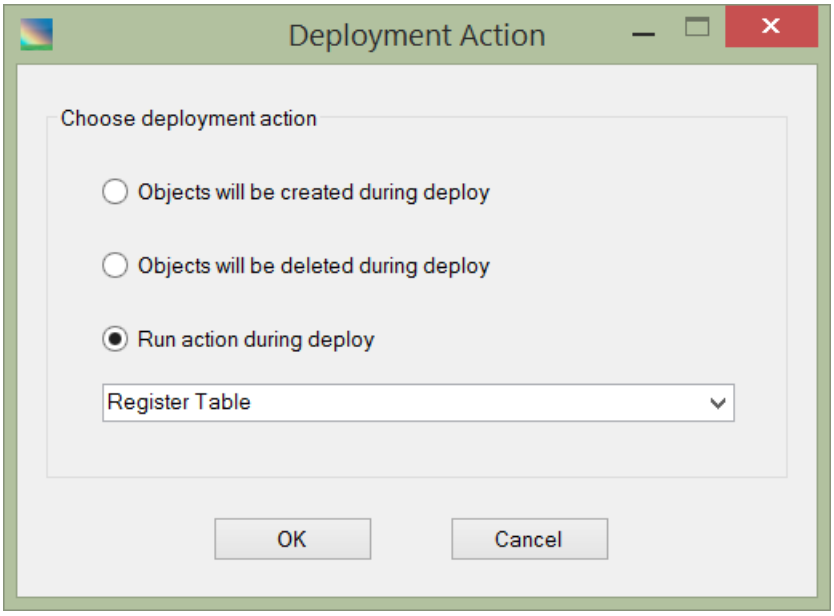
Alternatively, you can select the project you want in the “Development” drop down list at the top right of the user interface.

Adding Objects to a Development Project

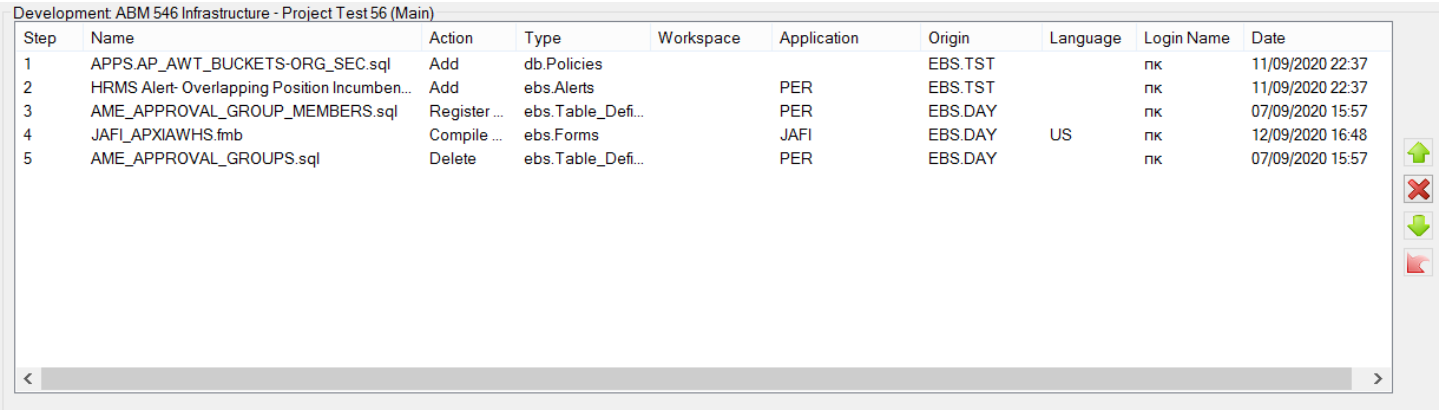
To add an object to the current Development Project, drag the object from the right (server) panel into the left (development) panel, or select the object and press “Add Objects”.

To add multiple objects at once, hold down the “Control” key and click the mouse cursor on each object you want to add, then drag the objects from the right (server) panel into the left (development) panel, or select the objects and press “Add Objects”.

If you press “Add Objects” and the object supports different deployment actions (Add, Delete, Custom) you can choose the desired deployment action before adding the objects to the development project. Dragging the object to the development panel will add it with the default “Add” action:



The objects are added to the development project with the deployment actions selected by user. In this example, the objects “APPS.AP_AWT_BUCKETS-ORG_SEC” and “HRMS Alert- Overlapping Position Incumbency“ will be created in the target environment during the project deployment, the object “AME_APPROVAL_GROUP_MEMBERS” will be registered in the target environment whereas the object “AME_APPROVAL_GROUPS” will be deleted:



After adding objects to the project you can change their order by selecting an object and pressing on the up and down arrows near the Development panel.

The Development panel shows the following information about its objects:

Step The sequential order in which the objects will be deployed in target environment.

Name Object name. The name may be slightly different from the object original name because some characters cannot be used in the file name and are removed. The original name can be seen in object properties form (right click on the object and select “properties”).

Type The object’s entity

Application The object's application

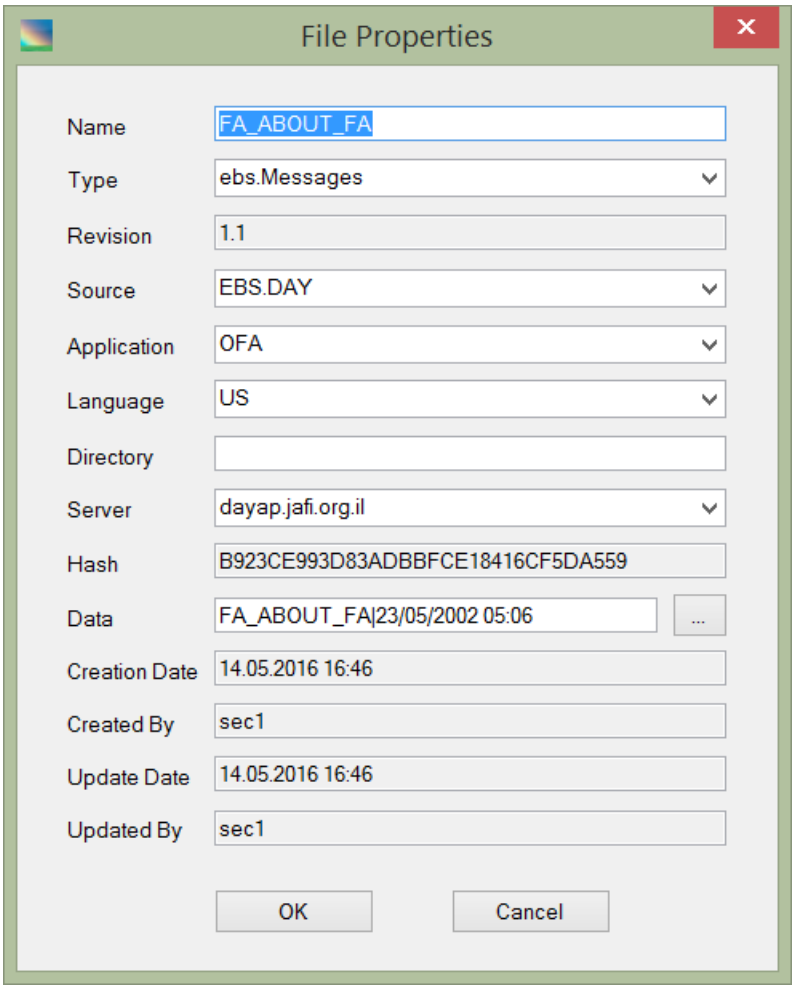
Origin The object's source environment

Language The object's language

Login Name The Raduga user who added the object to the project

Date The time when the object was added to the project

To see additional information about the object, double click it, or right-click it and select "Properties" from the context menu, to display the properties window:



The Properties form contains the following fields:

Name Object's original name

Type	Object's entity
Revision	Object's revision number in the version control system
Source	Source environment (environment the object was taken from)
Application	Object's application
Language	Object's language
Directory	Source directory (for existing objects)
Server	Source server name
Hash	Unique hash value for Raduga internal presentation
Data	Raw data about the object taken from the server panel Press on "…" button to edit object's raw data
Creation Date	The date when the object was taken from the server for the first time
Created By	The Raduga user who migrated the object from the server for the first time
Update Date	The most recent date when the object was taken from the server
Updated By	The most recent Raduga user who migrated the object from the server

You can edit an object's metadata using "Properties" form.

Deleting Objects from the Development Project

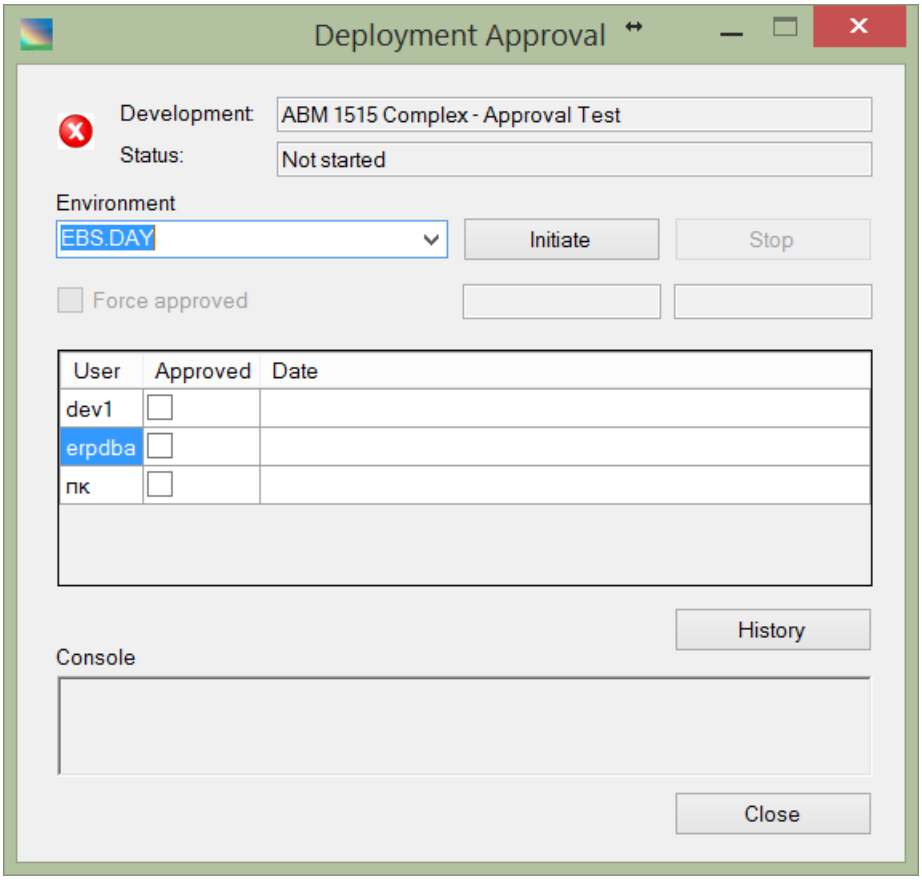
To delete an object from the project, select the object and press "Delete". To restore the object you just deleted, press "Undo".

Adding a Process to the Development Project

The project can contain processes in addition to objects. The process performs specific operation on the server, for example, restart Apache web server or compile apps schema. You can add process to the development project by choosing it in the "Processes" drop down list near the development panel and pressing on the "Add Step" button.

Approving a Development Project

If the project has a project approval rule for the specific environment, Raduga lets you deploy it only if all users defined in the approval rule approve it (see "Defining Project Approval Rules" in the Raduga Administration Guide). To initiate an approval process, click "Approvals" in the "Edit Development" form. The "Deployment Approval" form opens:



The form contains the following fields:

Development The development project name

Status The project's approval status

The list of the approval statuses:

Started	The approval process has started for the current environment
Not_Started	The approval process has not started for the current environment
Not_Needed	Approval is not needed for the current environment
Approved	The project is approved for deployment in the environment
Force_Approved	The project is force-approved by the administrator

Environment The environment in which the project needs to be deployed

Initiate Initiates the project approval process for the current environment

Stop Cancels the project approval process for the current environment

Force Approved Check to force-approve (by administrator only)

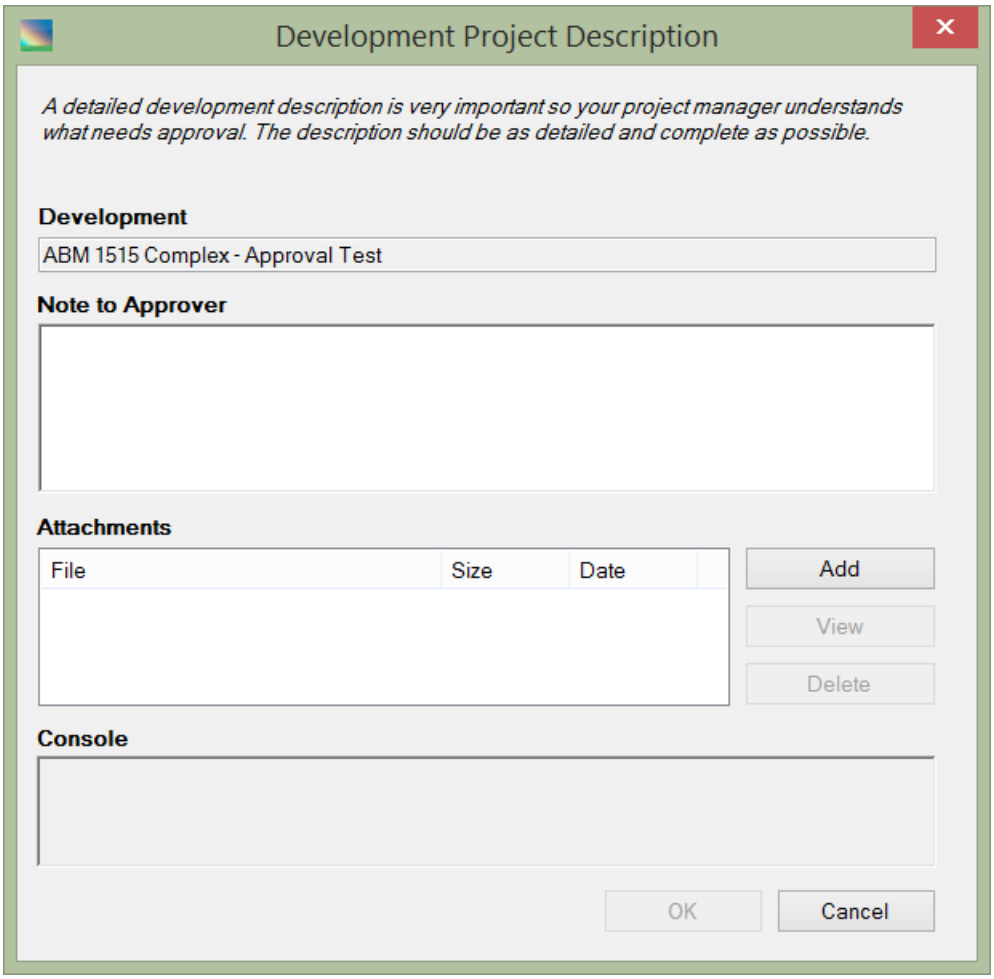
User The Raduga user who needs to approve the project

Approved Shows if the project was approved by the user

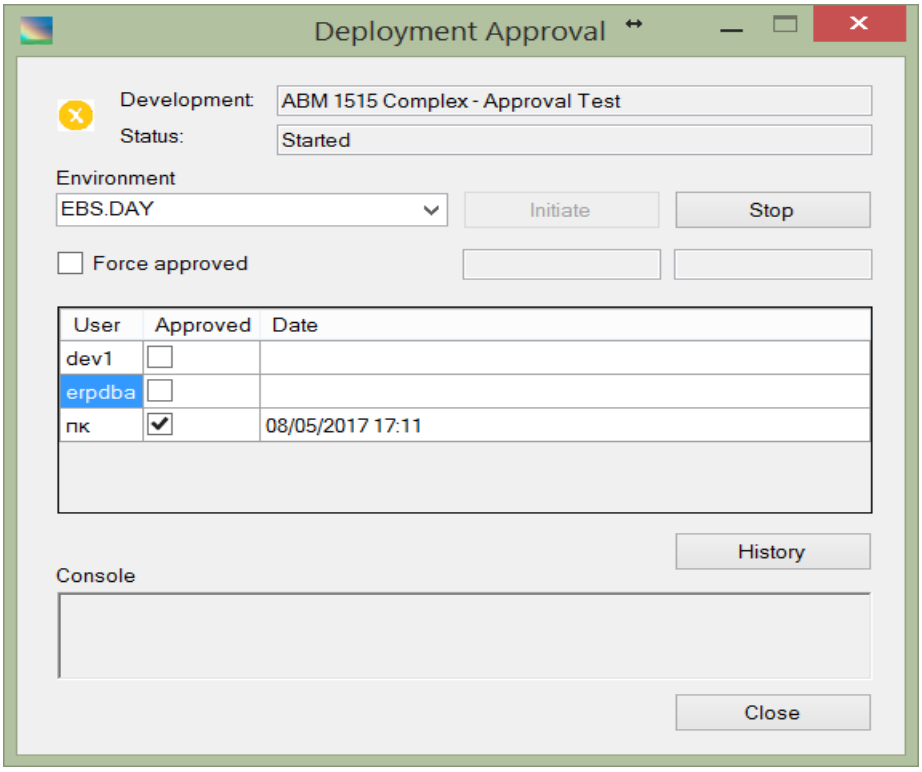
Date The approval date

History Opens the project approval history form

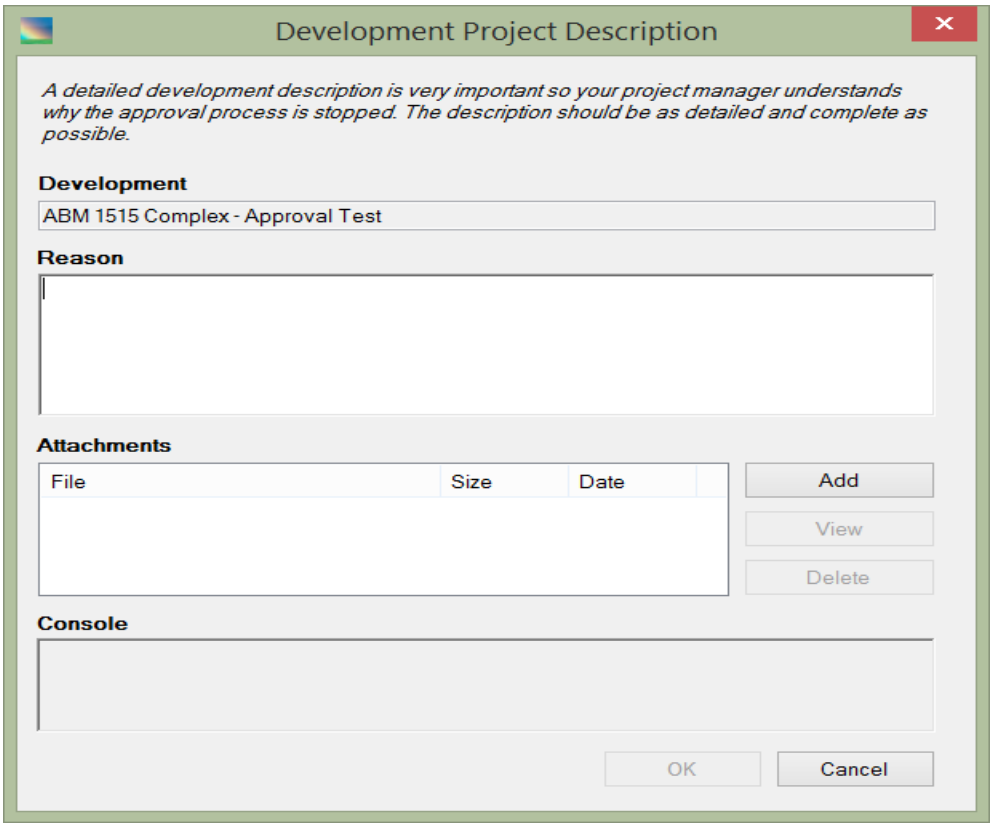
Select an environment and click “Initiate”. Additional “Development Project Description” form opens. Use it to add a detailed approval note so your project manager understands what needs approval. Click “Add” to attach files to the development project:



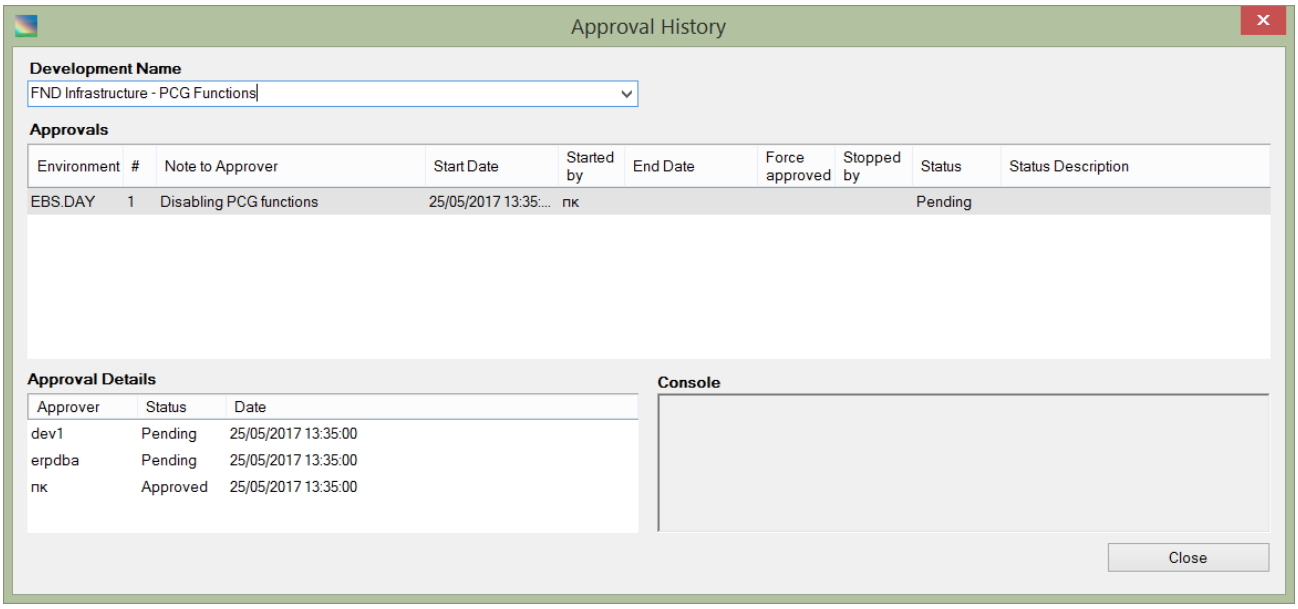
Click “OK”. This will initiate the development project approval process for the selected environment.
If you are a part of the approval chain Raduga will automatically add your approval to the project:



Click “Stop” to cancel the approval process. Additional “Development Project Description” form opens. Use it to add a detailed reason for cancelling the approval chain. Click “Add” to attach files to the development project:



Click “History” to open the project approval history form:



There are two built-in constants that regulate the approvals behavior:

util.PROJ_LOCKED_BY_APPROVER (valid values: Y/N, default: N) - if set to “Y”, the development project is automatically locked for changes when it is approved for deployment. Only administrator or development team leader can unlock locked projects.

util.PROJ_APPROVAL_HOURS - an approval expiration time in hours. If set to 0 (default) the approval never expires.

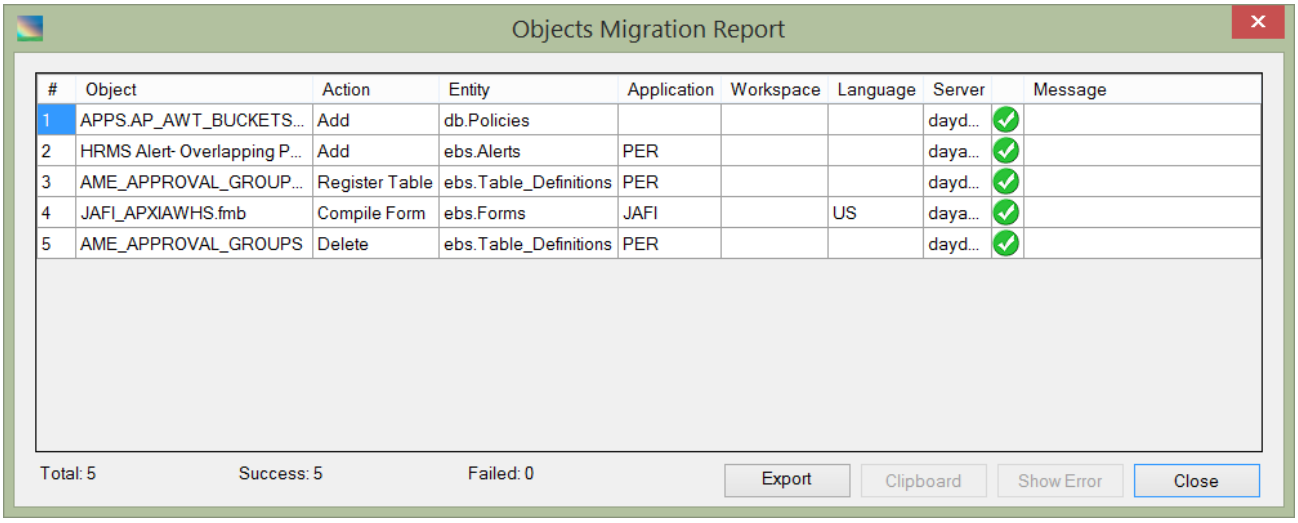
Deploying a Development Project

Once the project is approved it can be deployed in the environment. When you deploy a project, Raduga executes all processes and installs all project objects in the current environment. Deployment takes place sequentially according to the order of steps in the project. If one of the project steps fails then the deployment stops. The deployment is marked as successful only if all its steps finished successfully.

You can perform a partial deployment by choosing specific steps in the Development project. In this case the partial deployment is marked as successful once all selected steps have completed successfully.

- To deploy a project in the current environment, click “Deploy All”.
- To perform a partial deployment, choose the appropriate objects in the project and click “Deploy Selected”. Alternatively you can right click on the selected project objects and choose “Deploy” in the context menu.

When deployment finishes Raduga displays the “Object Migration Report”:



Successful steps are marked in green. If some of the steps failed during deployment the corresponding lines in the Object Migration Report are marked in red and the "Message" column for that step contains the relevant error message.

To review the deployment log file, including information about deployment status, press "Log" in the Development Project panel.

Adding a Development View

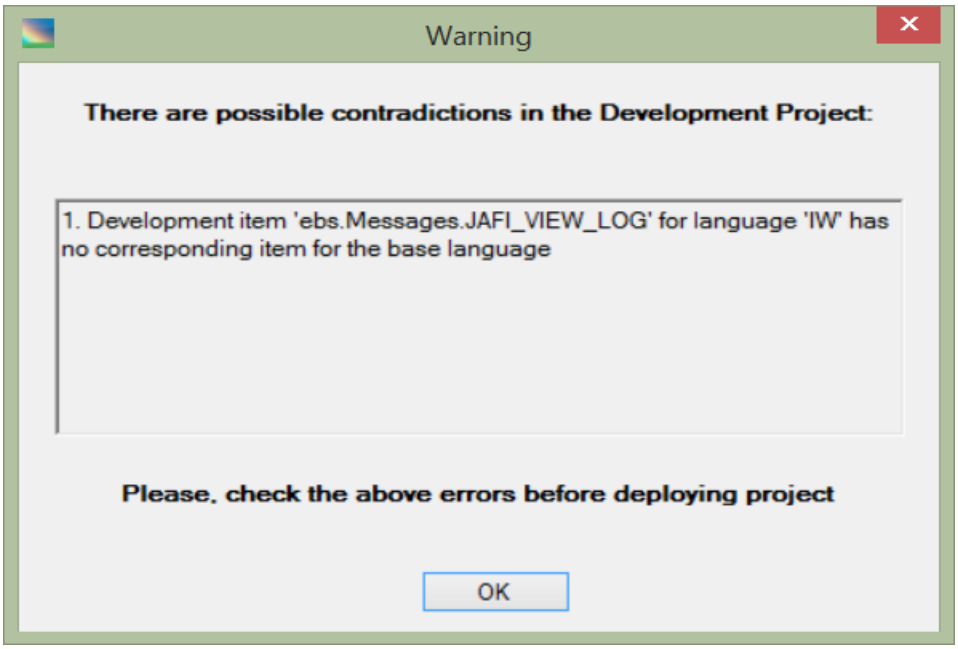
A development view is a way of looking at the Development Project. When you create a development project it is created with a "Main" view. Later you can edit the project and add a development view by typing its name in the "Development View" text box. Raduga automatically creates the new view.

The Development View consists of all objects from the Main View. You can add and delete objects in the new Development View, and changes in this view are not reflected in other views. However, the object directory is common for all views of the project, so if you replace an object in the Development View it will affect all views of the development project.

Use development views when you want to deploy a part of the development project. Instead of partial deployment (when you manually select objects that need to be deployed and press "Deploy Selected") you can create a development view that consists of only the objects that need to be deployed.

Verifying a Development Project

In order to check Development Project consistency, press "Verify". If Raduga finds possible contradictions between the development project items, it displays a warning:



If possible, resolve all inconsistencies before deploying the development project. However, you can ignore the warning if you choose. It will not prevent deploying the project.

Opening the Development Project Folder

To open the folder containing the current project, right-click on the project panel and choose “Open Folder” from the context menu.

- Hover the mouse cursor over a specific object before right-clicking to open the object's folder.
- Hover the mouse cursor over the white area of the development project before right-clicking to open the project's directory.

Archiving Development Project Objects

When you archive, you archive the objects from the current environment, not the objects from the project. You can use the archive option to preserve the current state of objects before overwriting them with further edits or developments.

Press “Backup” and provide a name for the backup project to archive. During the backup process all objects belonging to the current project will be taken from the current environment and added to the newly created backup project.

You can use an archived backup version of the project at some future time to restore the objects to an earlier state, for example if you need to cancel a deployment and return to the pre-deployment version of all objects.

Refreshing a Development Project

The Refresh process reloads the Development Project so you can see the changes other Raduga users may have made. Click “Refresh” to refresh the display.

Synchronizing a Development Project

Press “Synchronize” to replace all project objects with objects taken from the current environment. Alternatively you can right click the development project objects and choose “Synchronize” from the context menu to replace the selected project objects with objects taken from the current environment.

Note that this overwrites the current project objects by replacing them with the equivalent objects from the current environment. You cannot undo a synchronization.

Using the DataLoad Task

In the DataLoad task you can record in a text file all actions performed in Oracle forms.

Recording Data Loader Files

To record a data loader file, choose a file name in the “Record” text box, select an application language in the “Language” drop down list and press “Record”. Raduga launches Oracle Applications and records all actions in a file. The Data Loader file name should include your Raduga user name in upper case. For security reasons, Raduga will show only Data Loader files belonging to the current Raduga user.

After you have finished, press “Refresh” for the remote server. You will be able to see the recorded file in Raduga's right panel. To transfer the file to your private working area, drag it to Raduga's left panel.

You can open the file, to review its contents or add actions and data, with Data Loader Professional (or Classic).

After you save the file, select the environment where you want to “play” the recorded actions, transfer the file to the server, select it in Raduga's right panel and press “Play”. This opens Oracle Applications and “plays” all recorded actions in the selected environment.

Raduga uses default browser for recording and playing data loader files. To use a non-default browser ask System Administrator to set its name in the “util.BROWSER” constant (see “Defining Constants” section in the “Raduga Developer Guide”).

Important notes

- **Data Loader File Name**: The Data Loader file name should include your Raduga user name in upper case. For security reasons, Raduga will show only Data Loader files belonging to the current Raduga user.

- **Oracle Applications Language**: If you select a language other than English for recording data loader files, the Data Loader program can hang.

- **Server Security**: If server security is enabled in Oracle Applications (it is disabled by default in version 11i and enabled by default in version 12i) then you will get the error “APP-FND-01542 The Applications Server is not authorized to access this database” in the DataLoader task.

  To disable server security, locate the variable “s_appserverid_authentication” in the context file and set it to OFF. Then run “autoconfig”. When you are done, to enable server security again, change the “s_appserverid_authentication” variable to ON or SECURE in the context file and run “autoconfig” again.

  If you cannot run autoconfig, use this temporary solution:

  - To disable SECURE mode from the command line, enter:

    cd $FND_SECURE
    java oracle.apps.fnd.security.AdminAppServer apps/<apps password> AUTHENTICATION OFF DBC=<DBC File>

  - To activate SECURE mode from the command line, enter:

cd $FND_SECURE
jre oracle.apps.fnd.security.AdminAppServer apps/<apps password> AUTHENTICATION [SECURE|ON] DBC=<DBC File>

- To check the database status from the command line, enter:

  cd $FND_SECURE
  java oracle.apps.fnd.security.AdminAppServer apps/<apps password> STATUS DBC=<DBC File>

- **Signon Profile**: The "Signon Password Case" profile should have the value of "Insensitive" at the site level to ensure that Data Loader can connect to Oracle Applications. **Note** that if you change the "Signon Password Case" profile from "Sensitive" to "Insensitive", you must change the passwords for existing Oracle Applications users for this change to take effect.

- **Cluster or Load Balancer configuration**: In some configurations the Oracle Applications environment consists of several application servers clustered or connected by Load Balancer. In this case the recorded data loader file may physically reside on one of the application servers and not appear on the Raduga server panel after recording. To locate the recorded file, choose in sequence each application server in the "Server" drop down control (in the upper right part of Raduga window) and press "Login". This will refresh Raduga's right panel to show files from the selected application server. Continue until you see the recorded file in the list.

Useful tips

The data loader program is very sensitive to environmental issues and in some circumstances can hang. To successfully play a data loader file, follow the following rules:

- **Close all internet and java applications and clear Internet Explorer and Java cache.** You can do it by pressing "Clear Cache" on the main Raduga window.

- **Update the DataLoader file**: Add application user and password in the options of Data Loader Professional (or Classic).

- **During the data loading**: Do not perform any actions on the computer

Recording FRD Trace Files

To record FRD trace files, check the “FRD Trace” check box, enter a file name in the “Record” text box, select an application language in the “Language” drop down list, and press “Record”. Raduga launches Oracle Applications and records all actions in an FRD file, according to Metalink (Doc ID 438652.1).

Note that FRD Trace file name should include the Raduga user name in upper case. For security reasons, Raduga will show only FRD Trace files belonging to the current Raduga user.

Using the Version Control System

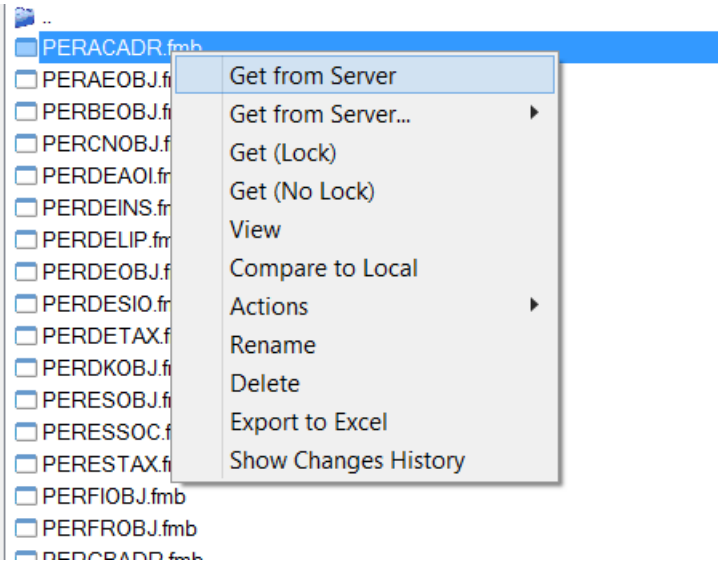
When you get an object from the server, the version control system puts a lock on the object until you complete your work and check the object back in again. This mechanism prevents multiple developers from making simultaneous changes to the same object.

If you try to get an object that is locked by somebody else, Raduga displays a message telling you the object is locked by the other user. You see the same message if you are trying to put or deploy an object that is locked by another developer. The user who created the lock, or an administrator, must release the lock before you can continue to work with the object.

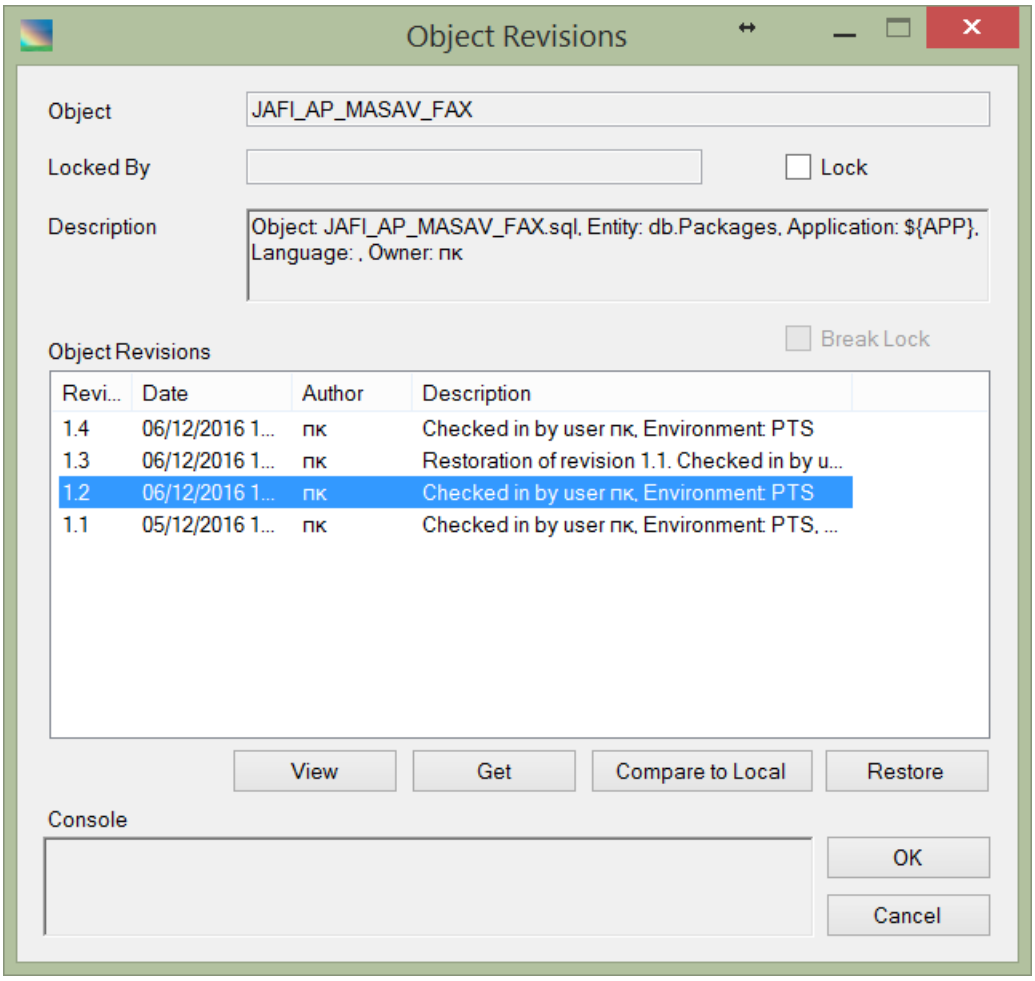
To get an object from the server for viewing only without creating a lock, right-click on the object and select “Get (No Lock)” from the context menu. **Note**: the object is not locked and other developers can make changes to it while you are looking at it.

To get an object from the server and create a lock, right-click on the object and select “Get (Lock)” from the context menu. Other developers will not be able to make changes to the object until you release it.

When you get an object by dragging it to the left Raduga panel or selecting “Get from Server” from the context menu, Raduga will put or will not put a lock on the object depending on the value of the “Lock Files” checkbox in the “Private Configuration” panel (See “Changing User Preferences”).



The version control system saves the history of all changes developers make to objects. To see the change history for an object, right-click on the object and select “Show Changes History” from the context menu. This opens the Object Revisions window:



This window provides information about the object, including its revision history.

Object The original object name

Locked By User holding the object (if any)

Unlock Check box to lock/unlock the object.

Description Object details (entity, application, language).

Break Lock Check box to break another user's lock (available to the Raduga administrator only).

Object Revisions A list of object changes with date, author and additional information.

Get Select a specific line from the "Object Revisions" list and click "Get" to retrieve the specific object's version.

Compare to Local Select a specific line from the "Object Revisions" list and click "Compare to Local" to compare this object's version to the local object.

Restore Select a specific line from the "Object Revisions" list and click "Restore" to restore the specific

object's version. The object with this version is retrieved from the Revisions Control System and becomes the current object.

If the object is not locked you can use the Object Revisions window in order to lock it. If the object is locked by you, you can unlock it by checking the "Unlock" check box.

A Raduga Administrator can break the lock held by any user by checking the "Break Lock" check box.

Retrieving Previous Revisions

In the Object Revisions window you can see the information about object revisions, their date, author and a short description. You can use this data in order to retrieve the older versions of the objects.

In the image above, you can retrieve revision 1.2 of the "JAFI_AP_MASAV_FAX" database package (current revision is 1.4). You can achieve this in two ways:

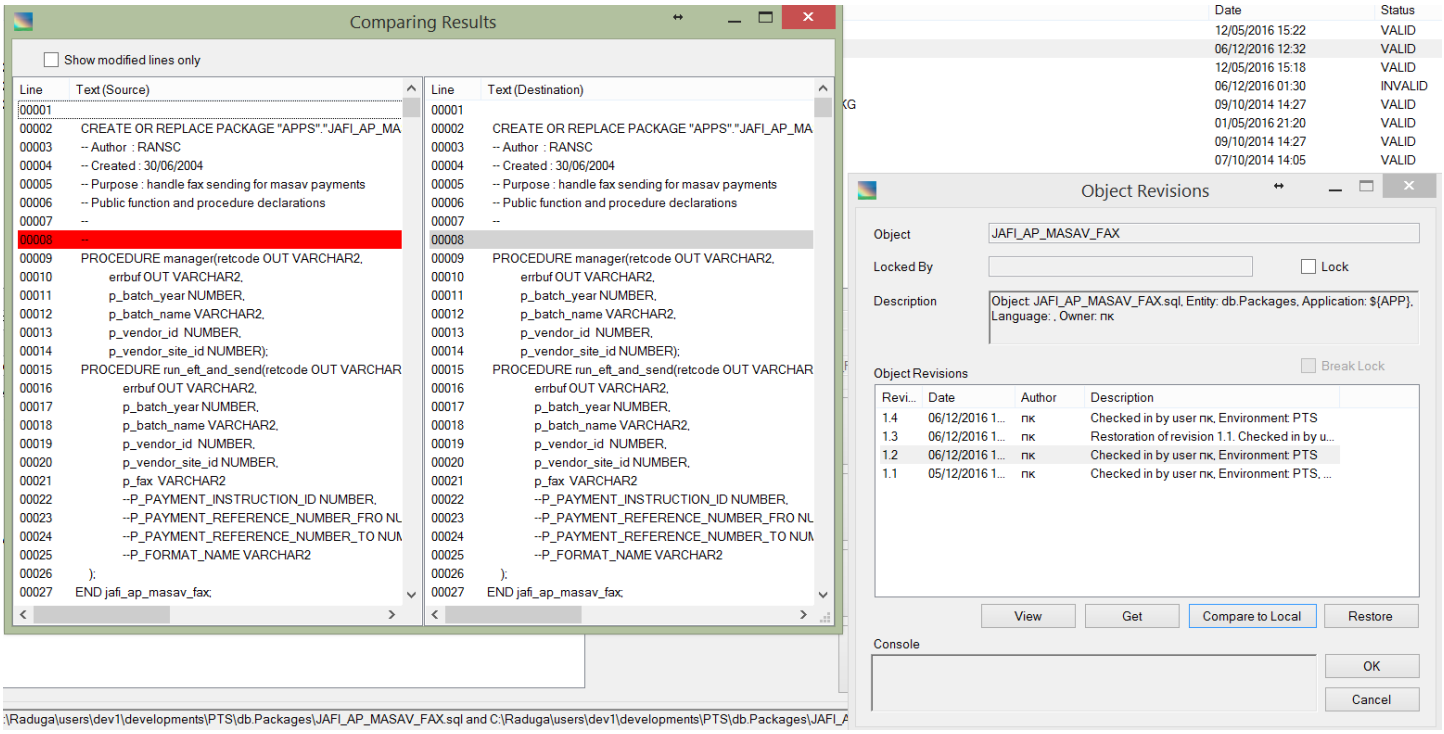
1. Select the line with revision 1.2 and click "Get". This will retrieve version 1.2 of the object and place it in your private working area. The current object's version will remain 1.4.
2. Select the line with revision 1.2 and press "Restore". This will retrieve version 1.2 of the object and make it current. The current object's version will become 1.5 and it will be the same as the 1.2 version of the same object.

Viewing Previous Revisions

In the image above, you can view the previous revisions of the "JAFI_AP_MASAV_FAX" database package. Select the line with the revision you want to view and click "View". This will retrieve the selected version of the object and open it in the editor.

Comparing Revisions

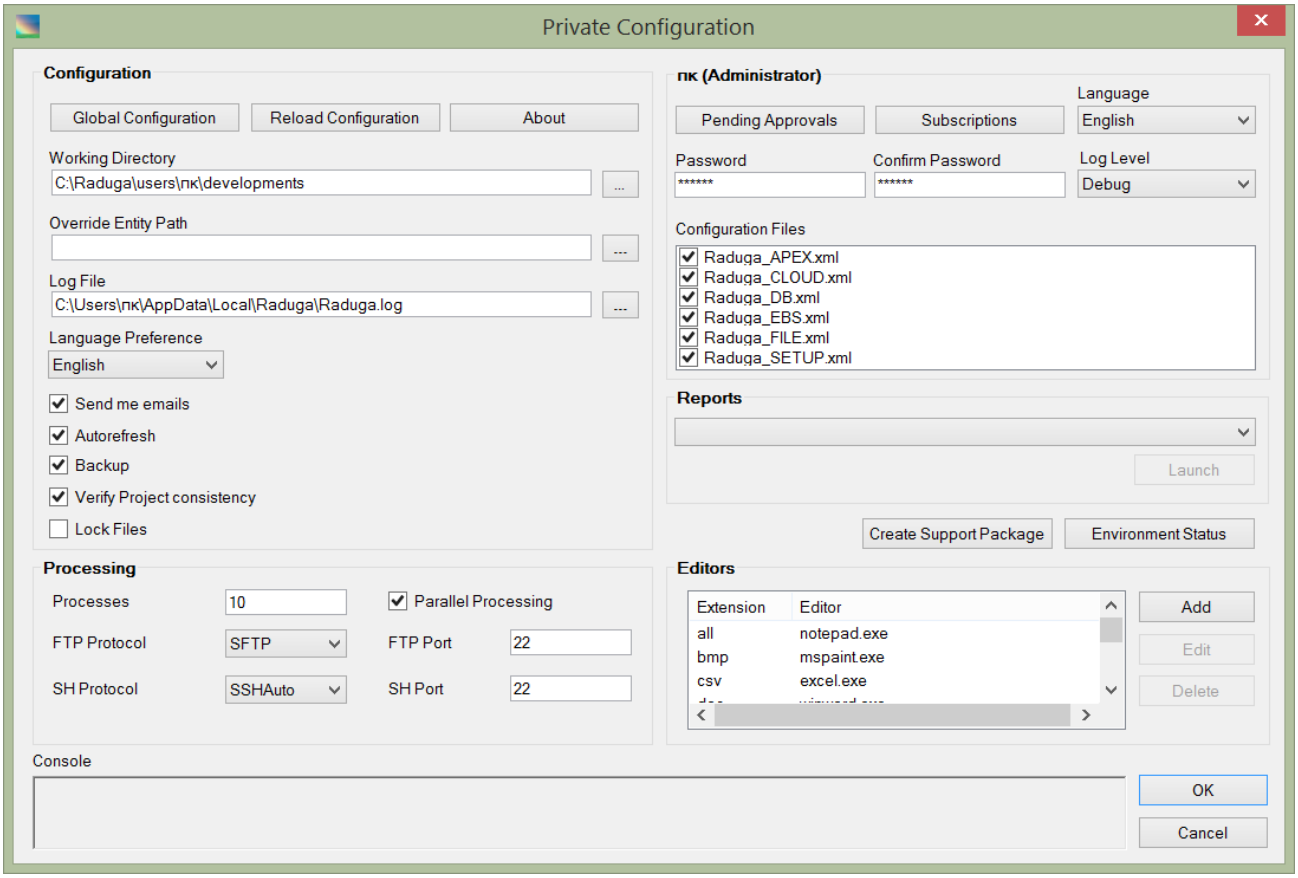
In the image above, you can compare the previous revisions of the "JAFI_AP_MASAV_FAX" database package with its local version. Select the line with the revision you want to compare and click "Compare to Local". This will retrieve the selected version of the object and compare it in to the local object.



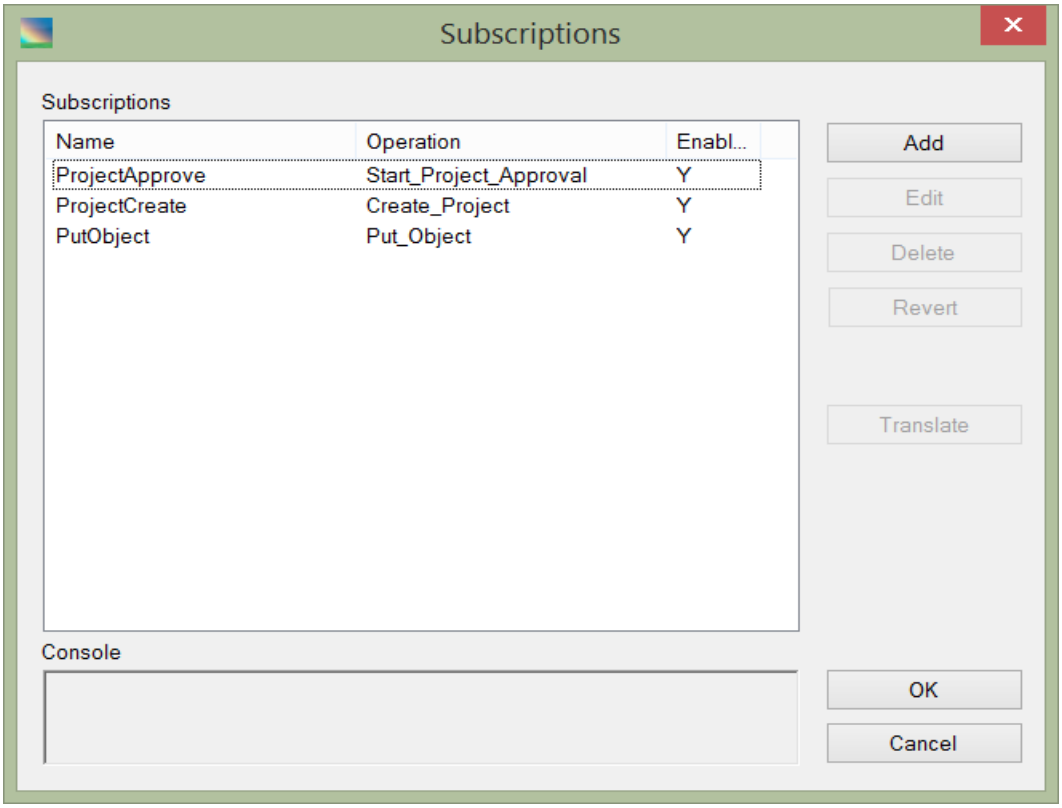
Other Actions

Defining Subscriptions

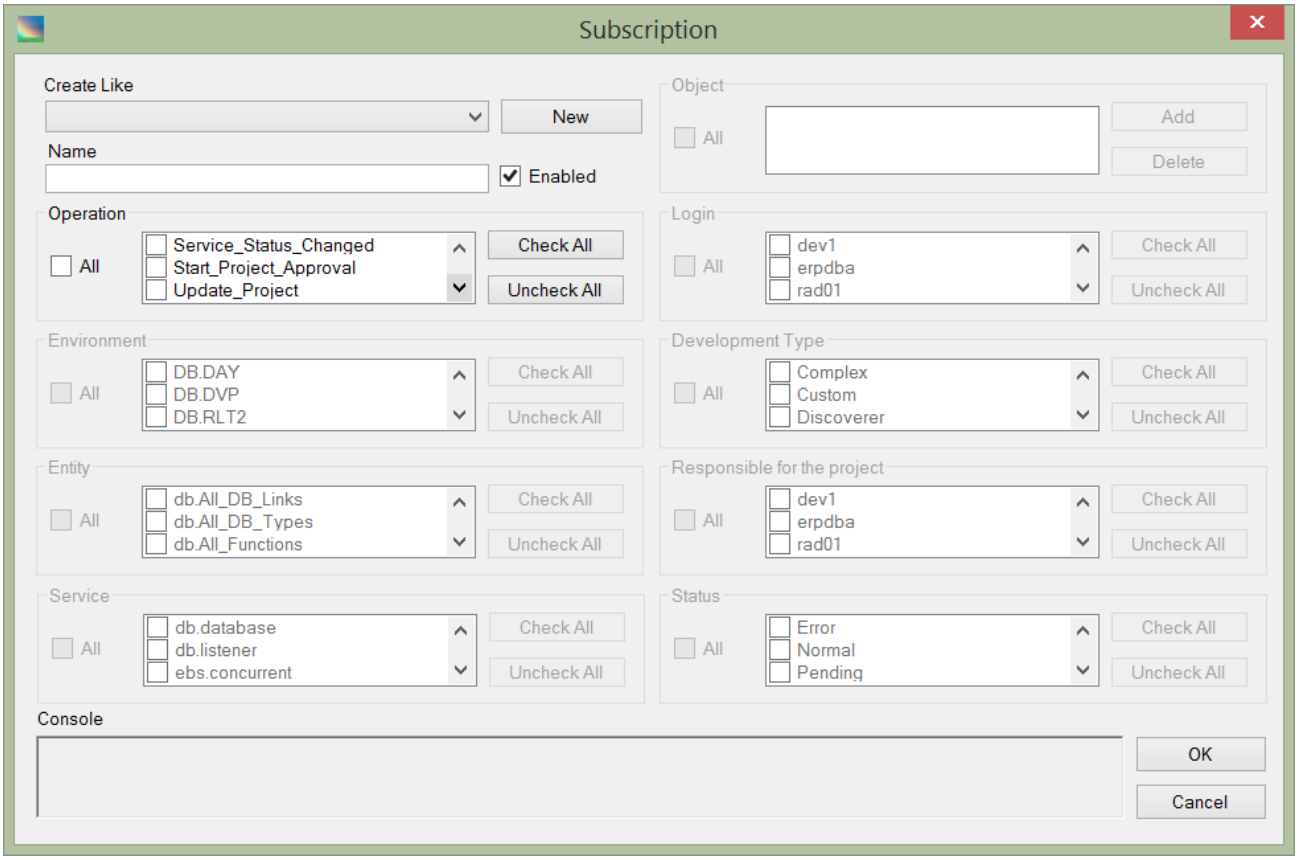
You can define a subscription to email notifications that result from certain actions that occur in the system. For example, you can configure Raduga to send you an email when a project is deployed in an environment or an object is sent to the server. There are many other events you can be notified about. To define a subscription click “Admin” button in the main Raduga form. In the “Private Configuration” form click “Subscriptions”:



The “Subscriptions” form opens. In the “Subscriptions” form click “Add” to create a new subscription or “Edit” to modify the existing one.



The “Subscription” form:



The form contains the following fields:

Create Like The list of existing subscriptions that can be used as a template for creating the new one

New Click to clear the fields and create a new subscription

Name The subscription name

Enabled Select/Unselect to enable/disable the subscription

Operation A list of events that trigger the notification (select "All" to choose all events):

Create_Project	The new project is created
Deploy_Project	Project is deployed in an environment
Get_Object	Object is taken from the environment
Put_Object	Object is sent to the environment
Restore_Object	Object's previous version is restored
Start_Project_Approval	Project's approval chain is initiated
Update_Project	Project is updated
Project_Approved	Project is approved for deployment
Project_Rejected	Project is rejected
Service_Status_Changed	Service status has changed
Login_Created	New Raduga user is created
Login_Updated	Raduga user is modified

Check All Select all items

Uncheck All Deselect all items

Environment A list of environments that must be involved in the event to trigger the notification (select "All" to choose all environments)

Entity A list of entities that must be involved in the event to trigger the notification (select "All" to choose all entities)

Object A list of objects that must be involved in the event to trigger the notification (select "All" to choose all objects). The wildcard characters * or % can be used to define the object's mask.

Login A list of users who must be involved in the event to trigger the notification (select "All" to choose all users)

The event	Notification will be sent if…
Create_Project	The selected login is the project's owner
Deploy_Project	The selected login is the project's owner
Get_Object	The selected login performs the operation
Put_Object	The selected login performs the operation
Restore_Object	The selected login performs the operation
Start_Project_Approval	The selected login is approver
Update_Project	The selected login is the project's owner
Project_Approved	The selected login is the project's owner
Project_Rejected	The selected login is the project's owner

Development Type A list of development types that must be involved in the project event to trigger the notification (select "All" to choose all development types)

Responsible for the project
A list of users responsible for the project who must be involved in the project event to trigger the notification (select "All" to choose all users)

The event	Notification will be sent if…
Create_Project	The selected login is the project's responsible owner
Deploy_Project	The selected login is the project's responsible owner
Update_Project	The selected login is the project's responsible owner
Project_Approved	The selected login is the project's responsible owner
Project_Rejected	The selected login is the project's responsible owner

Service A list of services that must be involved in the "Service_Status_Changed" event to trigger the notification (select "All" to choose all services)

Status A list of service statuses that must be involved in the "Service_Status_Changed" event to trigger the notification (select "All" to choose all users)

In the "Name" field type a unique name for the subscription. Choose the relevant operation. Depending on the operation, different subscription fields become available for editing. Choose all relevant environments, entities, objects, users and development types to filter the events triggering the notification. Note that the object field can contain wildcard characters.

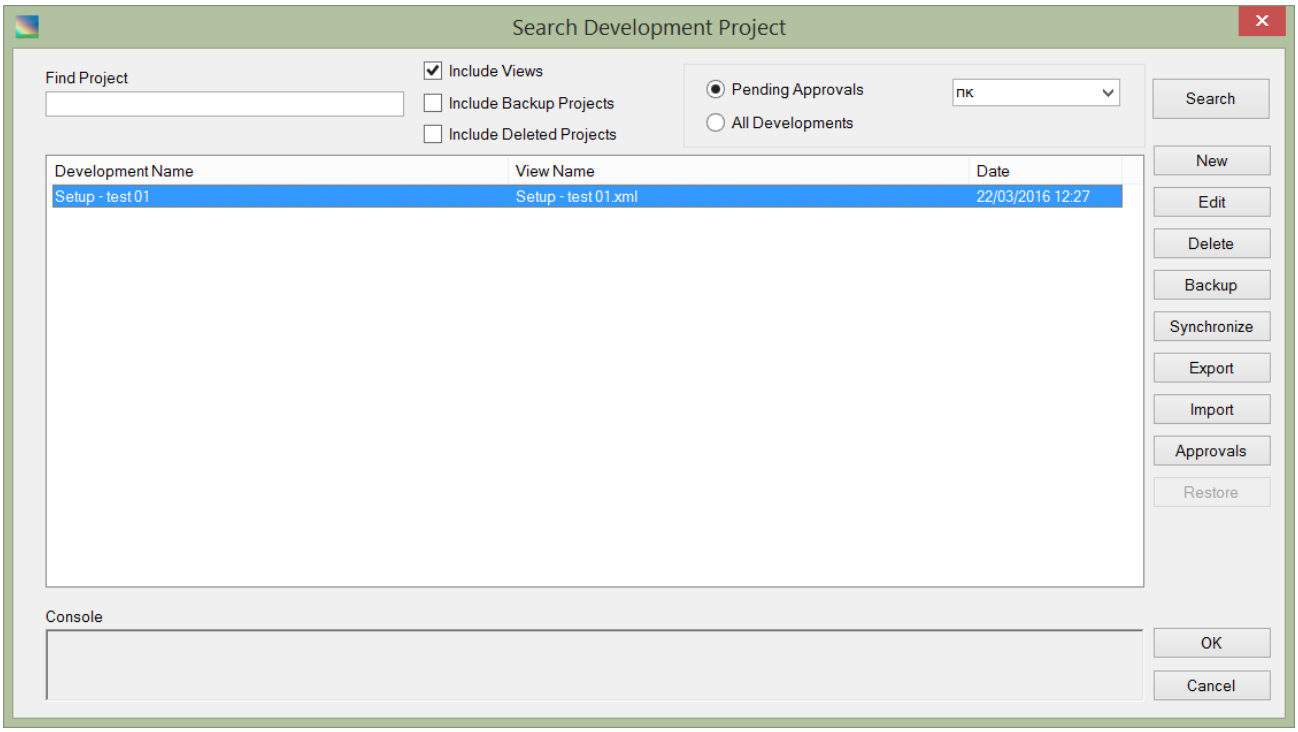
Click OK to save the subscription.

Note that when a new Raduga user is created it has the default "Start_Project_Approval" subscription. This subscription causes a notification to be sent to the user when a development project needs his approval. You can disable this subscription if you don't want to be notified about project approvals.

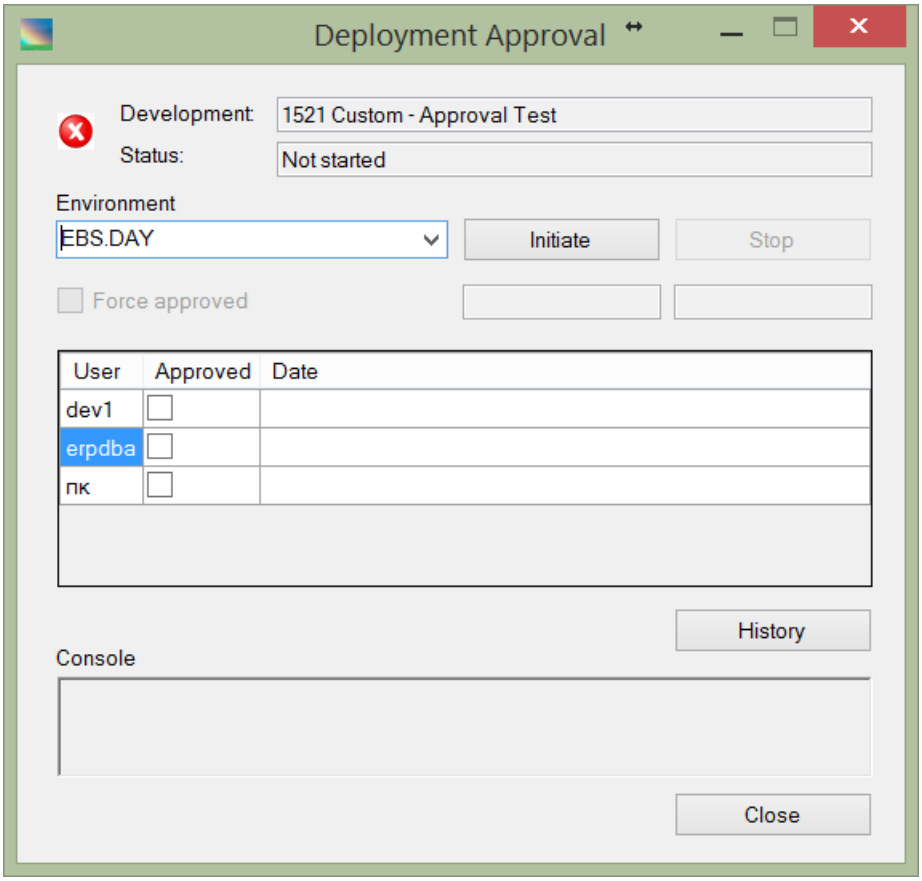
Other subscriptions which are assigned by default to a new user are the "Create_Project", "Project_Approved" and "Project_Rejected" subscriptions. These subscriptions cause a notification to be sent to the user when he is a responsible owner of the development project. You can disable these subscription if you don't want to be notified about the project events.

Approving Projects

If you are defined as an approver for the development projects you may need to approve the projects for deployment in specific environment. To see a list of the projects waiting for your approval, click "Admin" in the main Raduga form. In the "Private Configuration" form click "Pending Approvals". The "Search Development Project" window shows a list of your pending projects:



To approve a project, select it in the “Search Development Project” form and click “Approvals”. The “Deployment Approval” form opens:



The form contains the following fields:

Development The development project name

Status The project's approval status

The list of the approval statuses:

Started	The approval process has started for the current environment
Not_Started	The approval process has not started for the current environment
Not_Needed	Approval is not needed for the current environment
Approved	The project is approved for deployment in the environment
Force_Approved	The project is force-approved by the administrator

Environment The environment in which the project needs to be deployed

Initiate Initiates the project approval process for the current environment

Stop Cancels the project approval process for the current environment

Force Approved Check to force-approve (by administrator only)

User The Raduga user who needs to approve the project

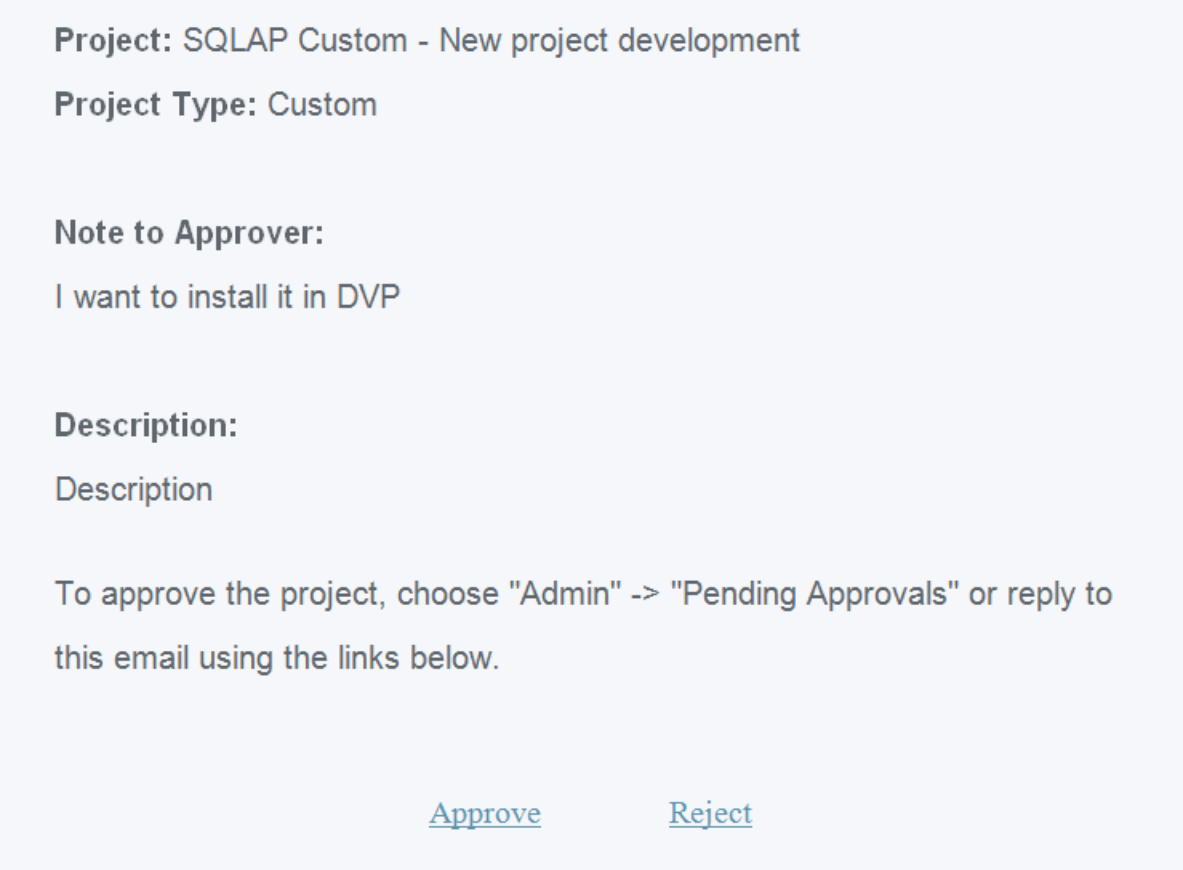
Approved Shows if the project was approved by the user

Date The approval date

History The project approval history

Check the "Approved" field to approve the project for the deployment in the current environment and click "OK".

Another way to approve the project is to reply to the email sent by Raduga to the approver:



Click “Approve” to approve the development project or “Reject” to reject the request.

Comparing Objects

You can compare any Raduga object to the corresponding object on the server, in a private working area or in a Development Project. To do so, right-click the object in one of the Raduga panels and select “Compare to Server/Local/Development” from the context menu. Raduga finds the corresponding object in the location you specified and compares it with the selected object.

To compare local objects with any other local files, press “Compare to …”. Both objects you want to compare need to exist in the system.

Some objects that Raduga deals with are binary objects. Raduga can still try to compare them and show the results, but the content of binary objects in the comparison window may be unreadable.

Changing the Server or the User

In general, Raduga connects to the remote server and the remote user according to the configuration defined by the DBA. So the end user does not need to know to which remote server it is necessary to connect in order to access a specific object.

However, in certain cases it may be necessary to change the remote server and/or remote user. To do that, type the appropriate values in the “Server”, “User” and “Password” fields, or choose corresponding values from the drop down lists available in these fields. To connect to the new server, press “Login”. You will be connected to the new server with new credentials and the remote server panel will be automatically refreshed.

Viewing Object Properties

To see full information about an object, right-click the object and choose “Properties” from the context menu. The “File Properties” window will open. In this window you can review and update object properties.

Note: In general you don’t need to update object properties. The only case when you may need to do that is when you create a new object in a private working area and you want to incorporate it in Raduga. In this case, open the object's properties window and choose suitable values for all writable fields.

Launching an Application

Pressing “Launch” button will start the application using the URL defined by the DBA. The URL can contain the application language and a user name taken from the Raduga current values in order to re-direct the application to the appropriate login screen.

Defining the environment URL is a customization and can differ from site to site.

Here is an example of a custom URL that can be defined for environment:

http://tstap.jafi.org.il:8000/OA_HTML/AppsLocalLogin.jsp?&langCode=${LANGUAGE}&username=${APPLOGIN}

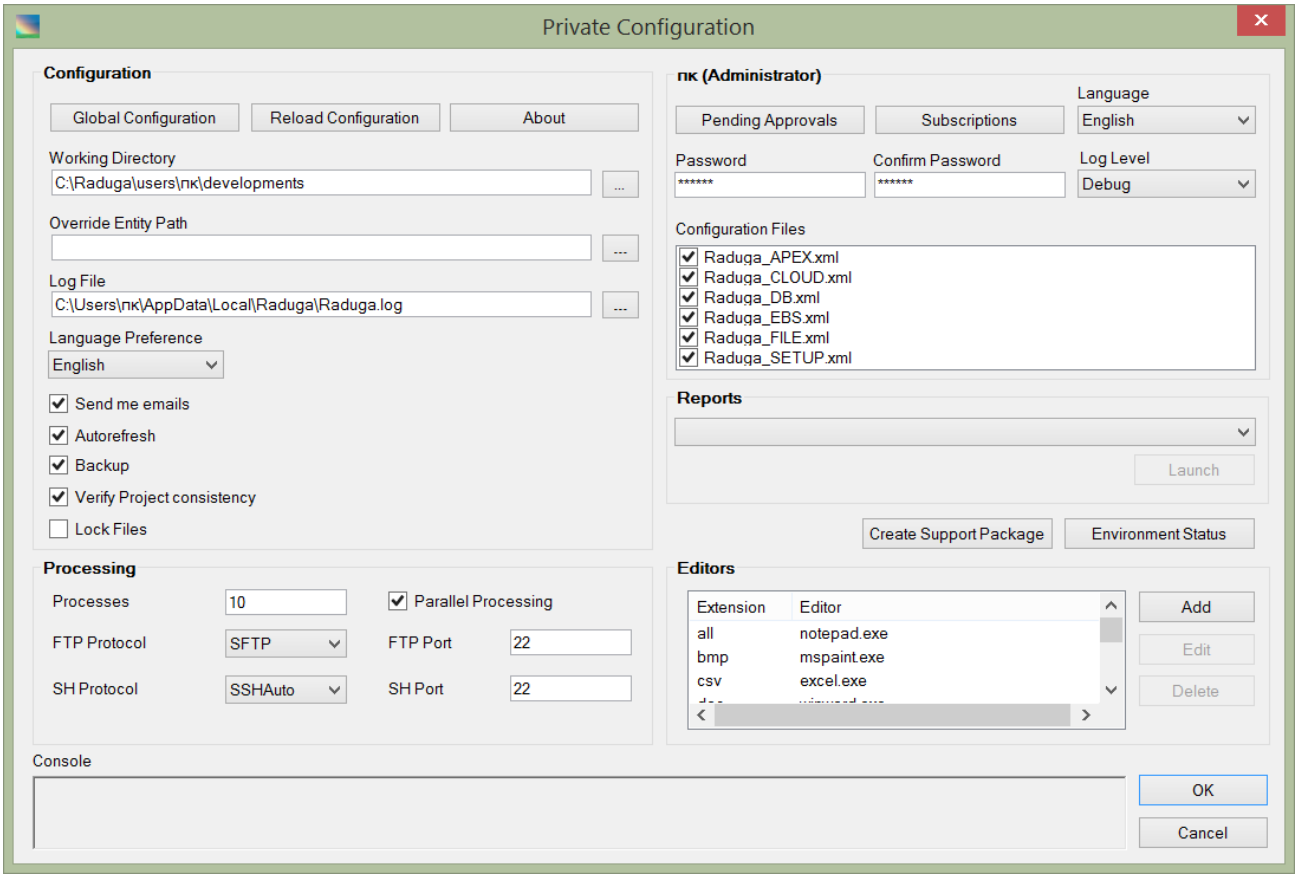
In this example:

${LANGUAGE} Raduga variable corresponding to the current application language

${APPLOGIN} Raduga variable corresponding to the current application login name

Changing User Preferences

A Raduga user can adjust the Raduga's application behavior according to his personal preferences. To change the Raduga private configuration parameters, press “Admin”. The “Private Configuration” window opens:



The following controls and information are available:

Global Configuration — Available to Administrators. Click to open the global Raduga configuration form. On this form Raduga administrators and DBAs can define custom environments, entities, entity types, constants, services, processes and reports.

Reload Configuration — Click to reload the Raduga configuration from the configuration files.

About — Click to show the current Raduga version, copyright and license information.

Working Directory — A private working area where all local objects are stored. Raduga recommends (and it is also the default Raduga configuration) storing all local objects in the following directory structure:

<Working Directory>\<environment>\<entity>\<application>\<language>

The <application> and <language> folders are optional because not all entities require them.

Override Entity Path — Overrides default entity path. The default entity path is comprised from the entity base path, environment name, entity name, application name and language. If you put value in the “Override Entity Path” textbox, it will be used as a default path for all entities.

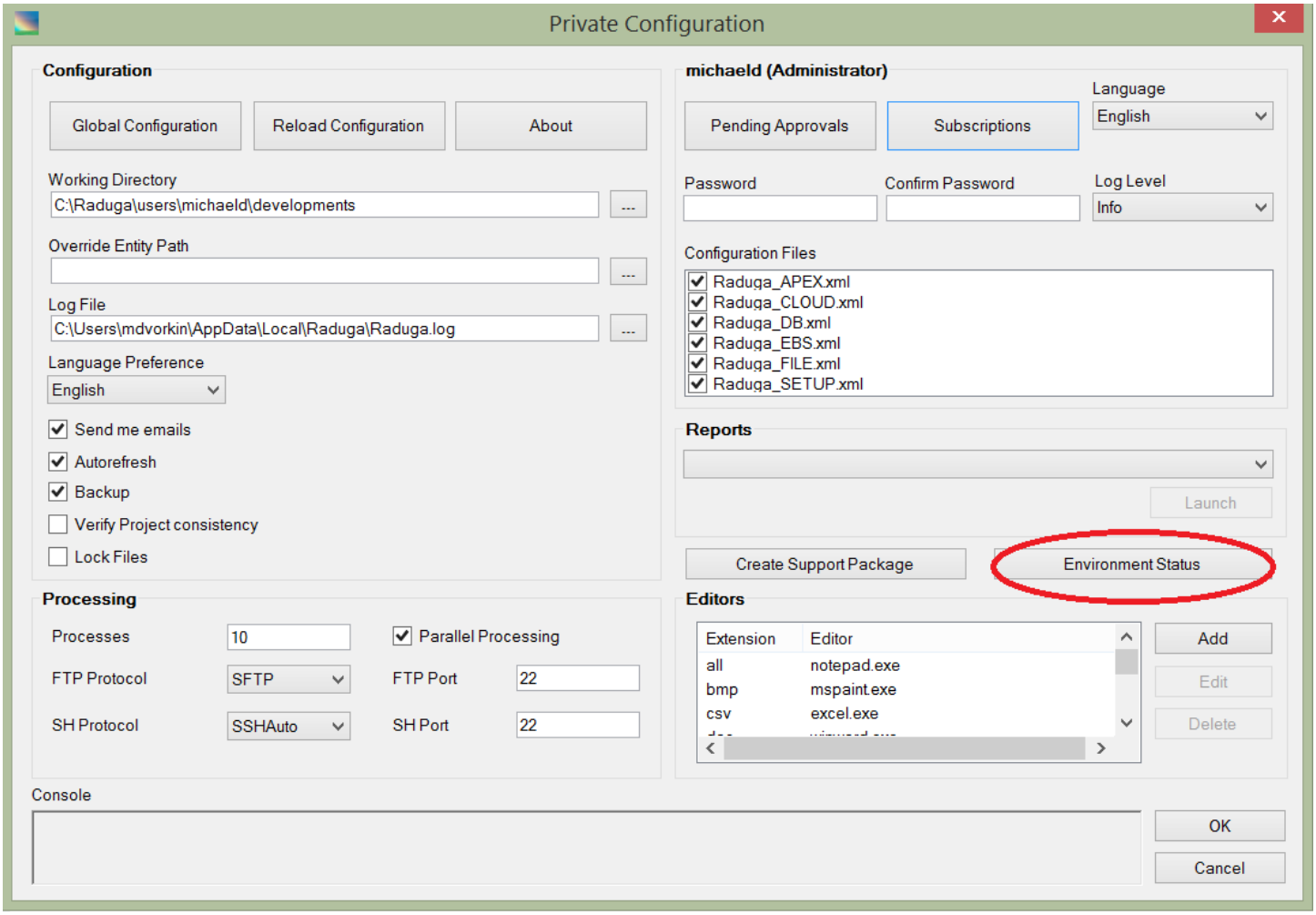
Log File — A Raduga log file. It can be placed on a local computer to avoid conflicts with other

	users' log files.
Language Preference	Choose the email language preference.
Send me emails	Check to get emails from Raduga.
Autorefresh	Check this check box to have the local and remote Raduga panels refresh automatically after you change the current environment, entity, application or language.
Backup	Check this check box to have Raduga back up local objects before overwriting them during a “get” operation.
Verify Project consistency	Check this check box to have Raduga verify development project consistency before each deployment.
Lock Files	Check this check box to lock objects automatically when you get them from the server.
Pending Approvals	Click to open the list of development projects waiting for your approval.
Subscriptions	Click to open a list of subscriptions for email notifications about different events that occur in Raduga.
Language	Set the language for the entire Raduga application. There is a limited set of languages available to Raduga at the moment; however, we will add additional languages in the future.
Log Level	Set the level of detail you want to record in system logs. The options are: *“Error”* - only errors are displayed in the application console and log file *“Info”* - errors and informational messages are shown *“Debug”* - detailed debugging information is sent to the console and log file Raduga erases the log file each time the application starts. The file contains all messages that match the selected log level. To read the log file, click “View Log” at the bottom right of the main window.
Password	To change your password, enter the new password here and in the **Confirmation** field.
Configuration files	A list of configuration files used by Raduga. Unchecking the configuration file will exclude its entities from Raduga entities list.
Processes	Set the maximum number of parallel threads that Raduga can launch on the client workstation.
Parallel Processing	Check this check box to enable parallel processing on the client workstation.
FTP Protocol	Select the preferred FTP protocol. “FTP Port” is updated automatically after you change the FTP Protocol; however, you can manually adjust the port if needed. If Raduga cannot

	connect to the destination server using the preferred FTP protocol and port it will try to use other protocols.
SH Protocol	Choose your preferred SH protocol. “SH Port” is updated automatically after you change the SH Protocol; however, you can manually adjust the SH port if needed. If Raduga is not able to connect to the destination server using the preferred SH protocol and port it will try to use other protocols. If you want to restrict Raduga to the selected FTP and SH protocol and port you can define a “TryAdditionalProtocol” registry string parameter under HKEY_LOCAL_MASHINE\Software\Raduga6 and set the parameter to “N”.
Reports	Choose a report you want to run and press “Launch”. This will start the Raduga reporting engine that will execute the report on the database of your choice.
Create Support Package	Click to create a Raduga support package containing all information necessary for the Raduga support team.
Editors	This is a list of programs Raduga uses to open files of various extension types. You can adjust this list to your personal needs.

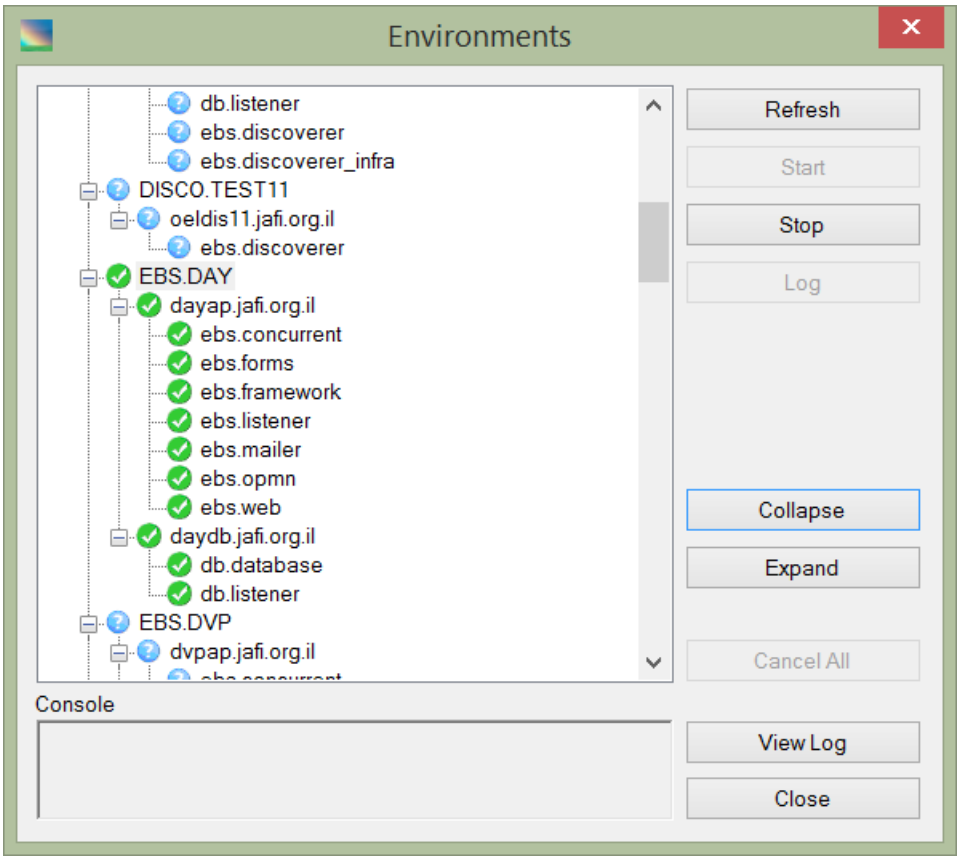
Monitoring Environments

You can monitor each Raduga environment's status, and start and stop environments. To open a list of environments with their statuses, press "Status" in the "Global Configuration" form or "Environment Status" in the "Private Configuration" form.



The "Environments" form opens. It displays a list of all environments defined in Raduga. Each environment consists of the physical servers hosting the environment's services. The status of each environment, server or service can be:

- Unknown
- Partially available
- Available
- Not available



The following controls and information are available:

Environments Tree	List of environments with their statuses. You can expand each environment's node to see its servers and services with their respective statuses.
Refresh All	Click to refresh all statuses.
Cancel All	Click to stop all running processes.
Refresh	Click to refresh the selected node with all its child nodes.
Start	Start all services of the selected node, including all its child nodes.
Stop	Stop all services of the selected node, including all its child nodes.
Log	View the log file of the selected service.
Collapse	Collapse the selected node and all its child nodes.
Expand	Expand the selected node and all its child nodes.
View Log	View the Raduga log file.

Sending Notifications and Appointments

You can configure Raduga to send emails and appointments directly from Oracle Applications. For that ask the Raduga Administrator to enable the Raduga Notification Service (See: “Notifications and Appointments” section in the Raduga Administration Guide). When the Raduga Notification service is enabled for the Oracle Applications environment, it creates the following tables:

RDG_NOTIFICATIONS

```
NOTIFICATION_ID     NOT NULL NUMBER          -- Unique Notification ID
TYPE                VARCHAR2(50)             -- Notification type. Can be: MAIL or APPOINTMENT
SUBJECT             VARCHAR2(300)            -- Notification subject
BODY                CLOB                     -- Notification body
RIGHT_TO_LEFT       CHAR(1)                  -- Right to left format. Can be: Y or N
LOCATION            VARCHAR2(100)            -- Meeting location (for appointment)
START_TIME          DATE                     -- Meeting start time (for appointment)
END_TIME            DATE                     -- Meeting end time (for appointment)
ALL_DAY_EVENT       CHAR(1)                  -- All day event flag. Can be: Y or N (for appointment)
HIGH_IMPORTANCE     CHAR(1)                  -- High importance flag. Can be: Y or N (for appointment)
OWNER               VARCHAR2(100)            -- Notification sender’s email address
TEMPLATE_NAME       VARCHAR2(100)            -- Email template used for the notification
```

RDG_NOTIF_PARTICIPANTS

```
NOTIFICATION_ID     NOT NULL NUMBER          -- Unique Notification ID
EMAIL               NOT NULL VARCHAR2(100)   -- Recipient’s email address
TYPE                VARCHAR2(10)             -- Recipient type. Can be: TO, CC, BCC (for mail) or
                                             -- REQUIRED, OPTIONAL (for appointment)
STATUS              VARCHAR2(1000)           -- Notification status. Can be empty, IN PROCESS, SENT or
                                             -- can contain an error message
STATUS_DATE         DATE                     -- Status change date
```

RDG_NOTIF_ATTACHMENTS

```
NOTIFICATION_ID     NOT NULL NUMBER          -- Unique Notification ID
FILE_NAME           NOT NULL VARCHAR2(300)   -- Attachment file name
MIME_TYPE           VARCHAR2(100)            -- Mime type. Can be: text/plain, application/excel,
                                             -- application/msword etc.
FILE_CONTENT        BLOB                     -- Attachment content
```

You can use these tables for sending emails and appointments directly from Oracle Applications. Whenever new records are added to these tables Raduga will send the notifications to the recipients.

Email notifications are generated using the Raduga default email template MailTemplate.html , located in the Raduga configuration directory. You can customize the template by creating a CustomMailTemplate.html file in the same place. The email template must contain a `<%Body%>` tag that will be replaced by the actual message body. A separate email

template may be created for each notification. Populate the column “TEMPLATE_NAME” in the RDG_NOTIFICATIONS table to use it for sending the notification.

Here are two examples of creating custom notifications:

Example 1. Sending a notice of appointment

```
begin

insert into RDG_NOTIFICATIONS
(
        NOTIFICATION_ID,
        TYPE,
        SUBJECT,
        BODY,
        RIGHT_TO_LEFT,
        LOCATION,
        START_TIME,
        END_TIME,
        ALL_DAY_EVENT,
        HIGH_IMPORTANCE,
        OWNER,
        TEMPLATE_NAME
)
values (
        RDG_NOTIF_SEQ.nextval,
        'APPOINTMENT',
        'Development Team Meeting',
        'Please, be on time',
        'N',
        'Room 112',
        to_date('17/06/2016 10:00', 'DD/MM/RRRR HH24:MI'),
        to_date('17/06/2016 11:30', 'DD/MM/RRRR HH24:MI'),
        'N',
        'N',
        'manager@it.org',
        '');

insert into RDG_NOTIF_PARTICIPANTS
(
        NOTIFICATION_ID,
        EMAIL,
        TYPE,
        STATUS,
        STATUS_DATE
)
values (
        RDG_NOTIF_SEQ.currval,
        'developer@it.org',
```

```
        'REQUIRED',
        null,
        sysdate);

insert into RDG_NOTIF_ATTACHMENTS
(
        NOTIFICATION_ID,
        FILE_NAME,
        MIME_TYPE,
        FILE_CONTENT
)
values (
        RDG_NOTIF_SEQ.currval,
        'plan.txt',
        'text/plain',
        UTL_RAW.CAST_TO_RAW('1. Development review. 2. Future projects'));

commit;

end;
```

Example 2. Sending an email

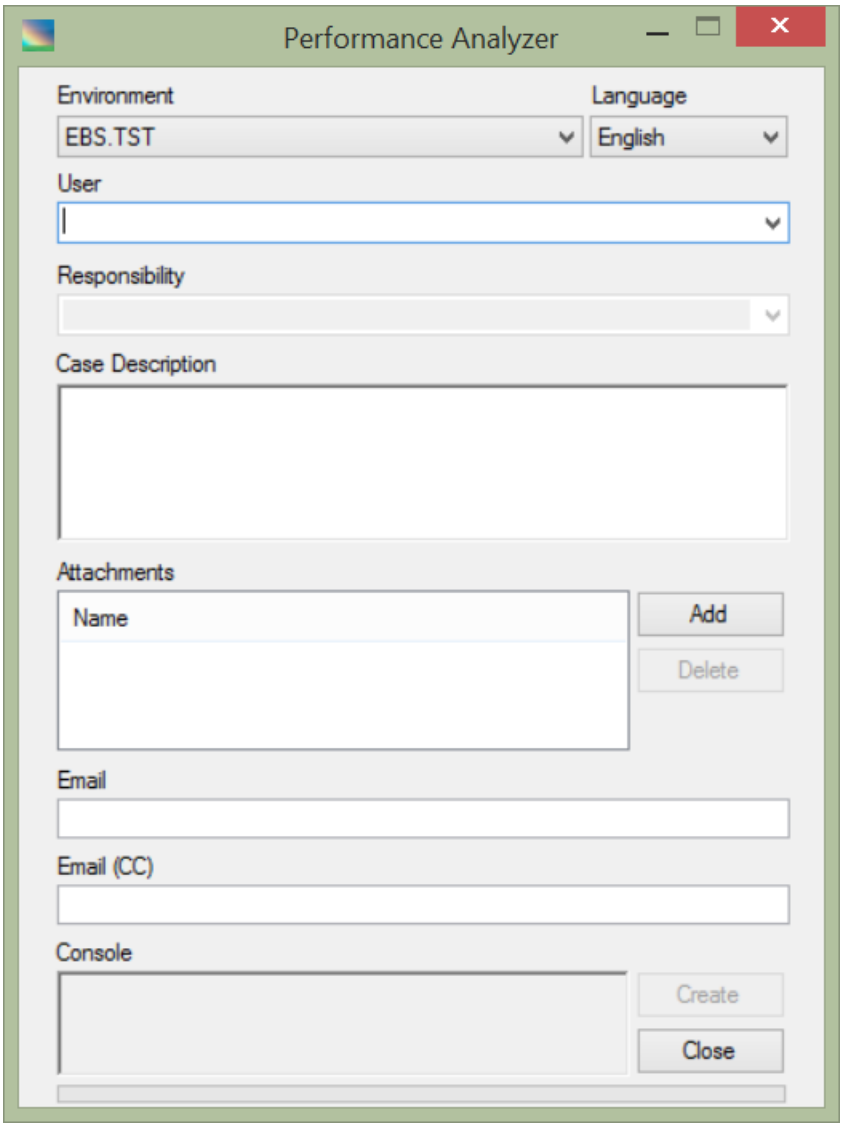
```
begin

insert into RDG_NOTIFICATIONS
(
        NOTIFICATION_ID,
        TYPE,
        SUBJECT,
        BODY,
        RIGHT_TO_LEFT,
        LOCATION,
        START_TIME,
        END_TIME,
        ALL_DAY_EVENT,
        HIGH_IMPORTANCE,
        OWNER,
        TEMPLATE_NAME
)
values (
        RDG_NOTIF_SEQ.nextval,
        'MAIL',
        'Development Specs',
        'Please review the attached document.',
        'N',
        '',
        '',
        '',
```

```
        '',
        '',
        'manager@it.org',
        '');

insert into RDG_NOTIF_PARTICIPANTS
(
        NOTIFICATION_ID,
        EMAIL,
        TYPE,
        STATUS,
        STATUS_DATE
)
values (
        RDG_NOTIF_SEQ.currval,
        'developer@it.org',
        'TO',
        null,
        sysdate);

insert into RDG_NOTIF_ATTACHMENTS
(
        NOTIFICATION_ID,
        FILE_NAME,
        MIME_TYPE,
        FILE_CONTENT
)
values (
        RDG_NOTIF_SEQ.currval,
        'Document.txt',
        'text/plain',
        UTL_RAW.CAST_TO_RAW('Development specifications'));

commit;

end;
```

Client Performance Analyzer (CPA)

Using CPA

Client Performance Analyzer is a standalone utility that lets you gather performance statistics and send the results to DBA. It can be used by the end users during peak hours to identify the causes of the system slowness. CPA gathers statistics about a client computer (CPU, disk and memory load), network speed and server state (database and server processes)



To analyze current performance, fill in necessary values and click on “Create”

Environment The environment that experiences performance issues

Language Choose preferred language

User The applicative user

Responsibility User's responsibility (optional)

Case Description Fill in a short description of the performance problem (optional)

Attachments Attach screenshots to the case (optional)

Email Destination email (the performance analysis will be sent to this email)

Email (CC) CC email (the performance analysis will be sent to this email)

Create Click on this button to start performance analysis

Close Exit the application

Configuring CPA

The following Raduga constants can be used to configure initial values of Client Performance Analyzer

util.CPA_ENV Initial default environment

util.CPA_EMAIL Destination email

util.CPA_CC_EMAIL Destination CC email

util.CPA_DB_METRICS Comma delimited list of database metrics used by CPA

Default list:

'SQL Service Response Time', 'Buffer Cache Hit Ratio', 'Memory Sorts Ratio', 'Redo Allocation Hit Ratio', 'Soft Parse Ratio', 'Host CPU Utilization (%)', 'Current Open Cursors Count', 'Database Wait Time Ratio', 'Database CPU Time Ratio', 'Row Cache Hit Ratio', 'Library Cache Hit Ratio', 'Shared Pool Free %', 'PGA Cache Hit %', 'Current OS Load', 'Session Count', 'Average Synchronous Single-Block Read Latency'

util.CPA_DB_STATISTICS Comma delimited list of session statistics used by CPA

Default list:

'application wait time','concurrency wait time','file io wait time','non-idle wait time','CPU used by this session','DB time','RowCR - row contention','user I/O wait time'

Command Line Mode

Raduga lets you perform operations through the command line in batch mode. Open a command prompt window (Start → Run → Type “cmd” → Press “Enter”) and change directory to the Raduga installation directory:

Example (the path can be different on your computer):

cd "C:\Program Files\LazyDeploy\Raduga\bin"

To launch Raduga command line help, use this command:

Raduga --help

The Raduga options display:

--batch	Batch Mode
--parfile="<FILE>"	Parameters File
--login="<USER>"	Login
--pass="<PASS>"	Password
--loginlang="<LANG>"	Login Language
--winlogon	Use Windows Credentials
--env="<ENV>"	Environment
--entity="<ENTITY>"	Entity
--task="<TASK>"	Task (FTP, Deploy, DataLoad)
--app="<APP>"	Application
--envlang="<LANG>"	Environment Language
--project="<PROJECT>"	Project
--object="<OBJECT>"	Object
--command="<CMD>"	Command (list, get, put, deploy, load)
--force	Force Overwrite Mode
--localdir="<DIR>"	Local Directory
--remotedir="<DIR>"	Remote Directory
--remoteserver="<SERVER>"	Remote Server
--remoteuser="<USER>"	Remote User
--remotepass="<PASS>"	Remote Password
--ftp="<PROTOCOL>"	FTP Protocol (FTP, SFTP)
--ftpport=<PORT>	FTP Port
--binary="<Y|N>"	Binary mode
--sh="<PROTOCOL>"	SH Protocol (Raw, Telnet, SSHAuto)
--shport=<PORT>	SH Port
--uservars="<VARS>"	User Variables (a string of comma separated values)
--log="<LOGFILE>"	Log File Name
--help	Help

There are 5 commands (list, get, put, deploy, load) that can be used in batch mode. To perform the command you need to supply all required parameters. Raduga will perform the command and exit.

Examples of Raduga command line parameters:

Get "HRMS Alert- Active Direct Reports" Alert from EBS.TST environment into directory C:\TEMP

Raduga --batch --login="user" --pass="password" --env="EBS.TST" --entity="ebs.Alerts" --task="Deploy" --app="PER" --object="HRMS Alert- Active Direct Reports" --command="get" --localdir="C:\TEMP"

Start Raduga in a GUI mode with specific startup values

Raduga --winlogon --env="EBS.TST" --entity="ebs.Forms" --task="Deploy" --app="PER" --envlang="US" --lang="English"

Import "HRMS Alert- Active Direct Reports" alert into EBS.TST environment from directory C:\TEMP

Raduga --batch --login="user" --pass="password" --env="EBS.TST" --entity="ebs.Alerts" --task="Deploy" --app="PER" --object="HRMS Alert- Active Direct Reports.ldt" --localdir="C:\TEMP" --command="put" --force

Import all "AP Invoice*" objects from the project "SQLAP Workflow - Adding Optional Approvers" into the EBS.TST environment

Raduga --batch --winlogon --env="EBS.TST" --task="Deploy" --command="deploy" --project="SQLAP Workflow - Adding Optional Approvers" --object="AP Invoice*" --force

Load data from the "DL_USER.fld" file into the EBS.TST environment

Raduga --batch --winlogon --env="EBS.TST" --command="load" --object="DL_USER.fld"

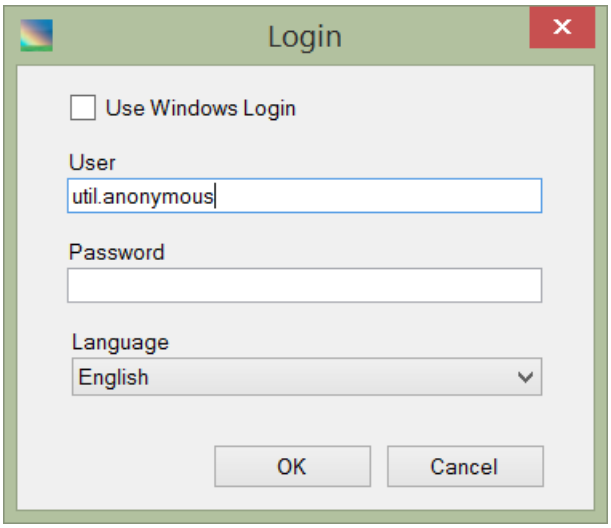
Load Raduga command line parameters from the Raduga parameters file

Raduga --parfile="C:\TEMP\RadugaOptions.txt"

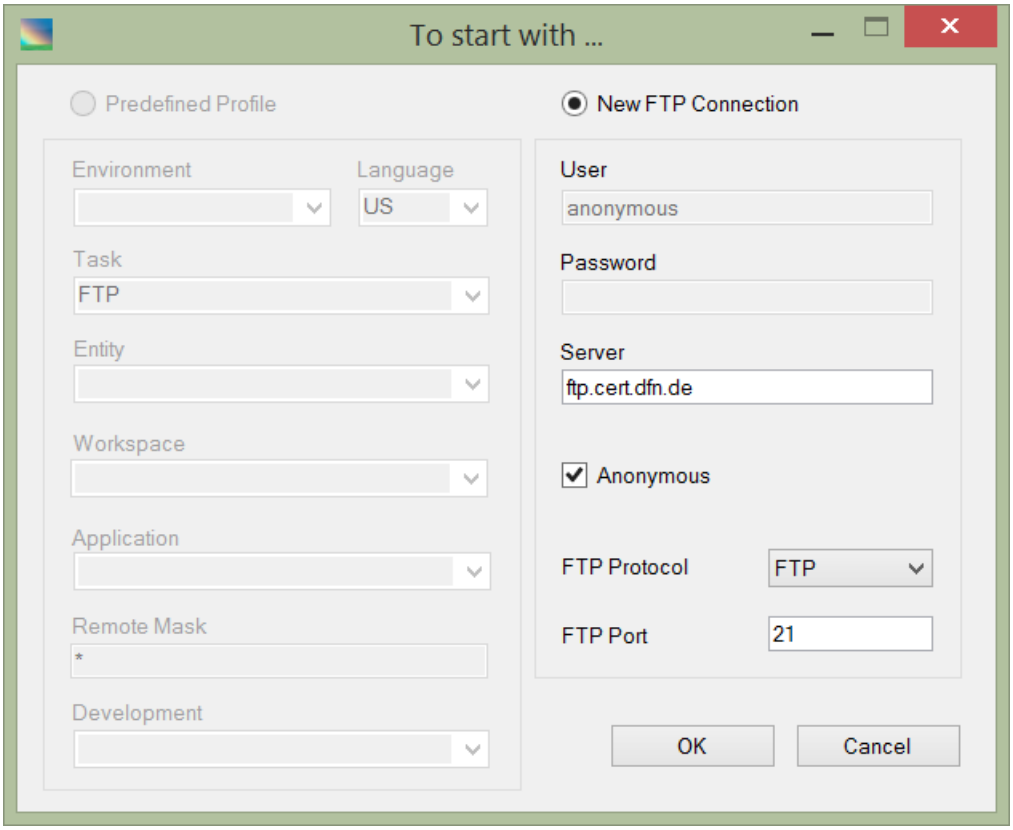
The sample RadugaOptions.txt file can be found in the Raduga installation directory.

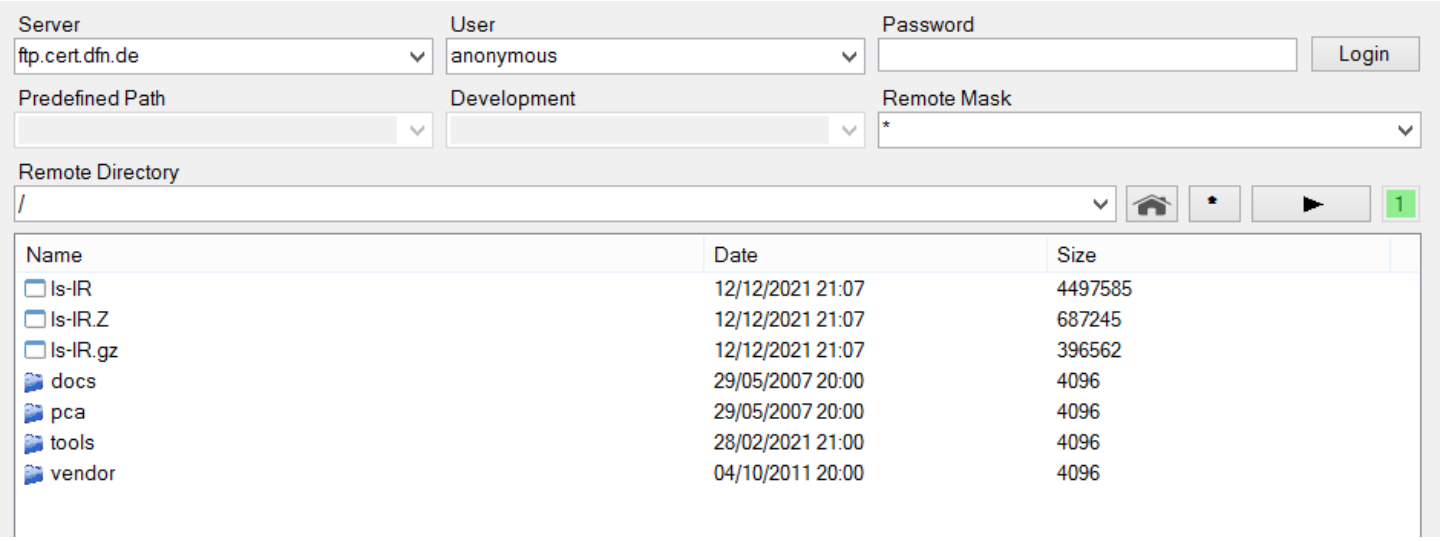
Using the util.anonymous user

You can use the “util.anonymous” user to connect to Raduga without a password, however while working as that user you do not have permissions in any environment. If necessary (though not recommended) the Raduga administrator can add permissions to the “util.anonymous” user.



The “util.anonymous” user has blank password and can be used for accessing remote FTP servers. When using the “util.anonymous” user to connect to the remote server supply the remote server name, remote user ID and password and press “OK”:





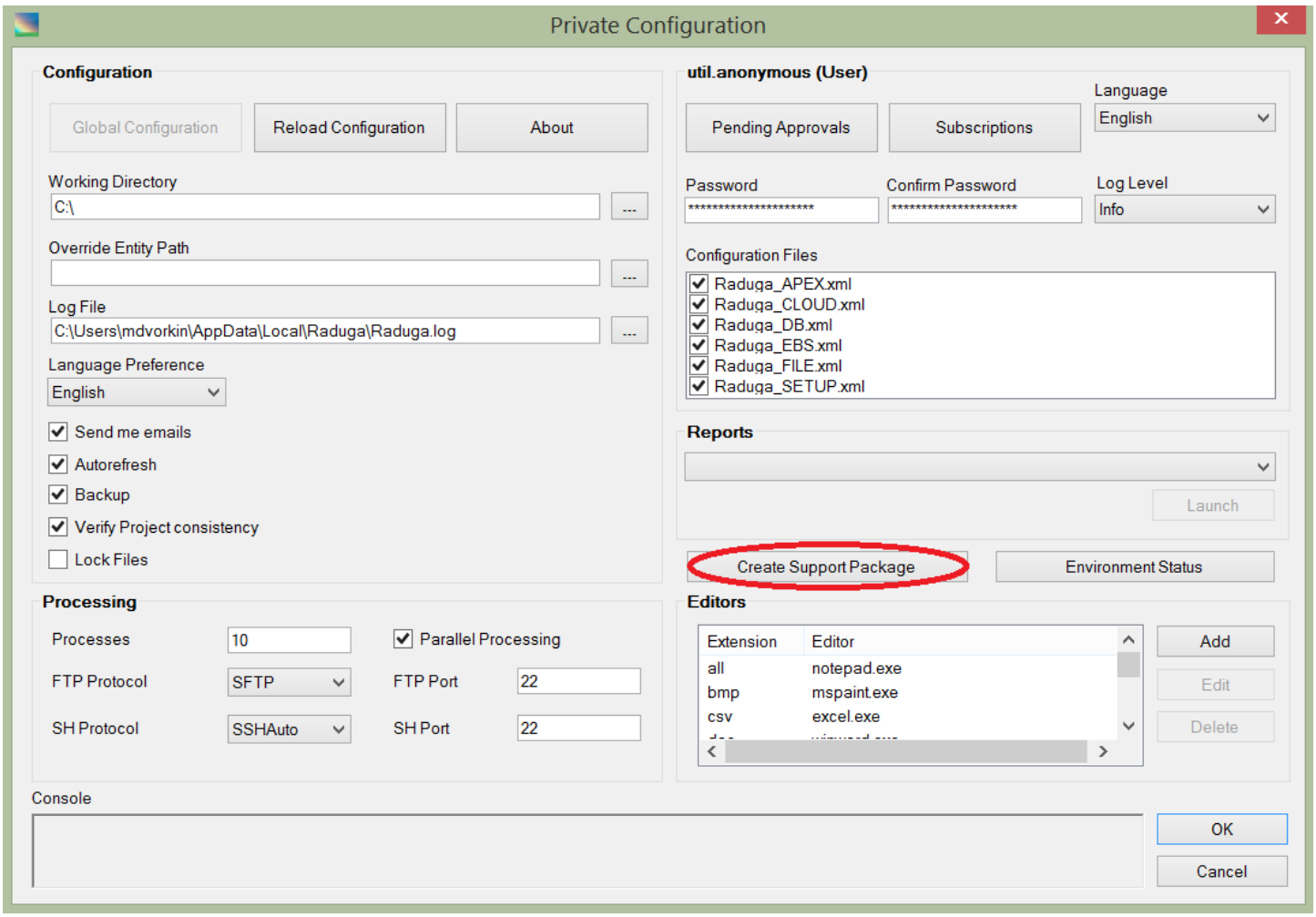
The “util.anonymous” user does not have a preferences file, so Raduga does not remember the last server/user/password for the “util.anonymous” user. This is necessary for security reasons because every end user having access to Raduga is able to login as the “util.anonymous” user.

Working with Support

In order to get effective tech support you need to provide a clear problem description and all Raduga configuration and log files.

Enable debug log level in the Raduga Private Configuration window (See: “Changing User Preferences” section of this manual) and reproduce the problem at this log level.

Create the Raduga Support Package “RadugaSupportPackage.zip” by pressing the “Create Support Package” button in Private Configuration form (See: “Changing User Preferences” section of this manual) and send it to the Raduga support team.



<u>The Raduga Support Package contains the following information:</u>

What is your Raduga user name?

What is your current context?

If the Raduga Support Package cannot be created, all values can be taken from the Raduga Main window. It is also possible to provide a screenshot.

- Environment
- Task
- Language
- Entity
- Application
- Local Mask
- Local Path
- Remote Server
- Remote User
- Predefined Path
- Project
- Remote Mask
- Remote Path
- FTP/SH Protocol and Port

What are the Raduga registry values?

If the Raduga Support Package cannot be created, export all registry values under HKEY_LOCAL_MACHINE\Software\Raduga6 and KEY_CURRENT_USER\Software\Raduga6 manually.

To export Raduga registry values

1. Open the registry editor (Start → Run → regedit).
2. Go to the HKEY_LOCAL_MACHINE\Software\Raduga6 path.
3. Right-click on the “Raduga6” key and choose “Export” from the context menu
4. In the opened registry file export window choose a file location (for example, C:\TEMP) and provide the file name (for example, Raduga_1.reg)
5. Press “Save” to save the file
6. Provide the exported “Raduga_1.reg” file to tech support.

Follow the same process to export the HKEY_CURRENT_USER\Software\Raduga6 registry key. The output file name can be named C:\TEMP\Raduga_2.reg

What is your client operating system?

If the Raduga Support Package cannot be created, open the control panel, choose the “System” icon and provide a screenshot.

What are the Raduga configuration, preferences and log files?

If the Raduga Support Package cannot be created, attach all Raduga Configuration, Preferences and Log files:

- Configuration files are the Raduga XML files having a form of “Raduga_<>.xml”. You can find them in the Raduga configuration directory. You can find the Raduga configuration directory path under the HKEY_LOCAL_MACHINE\Software\Raduga6\ConfigDir registry key.

- Your private preferences file (the preferences file has a form of “<Raduga login>.xml”. It is situated in the preferences directory under the following location: [working directory path]\..\pref. The working directory path can be found in the “Private Configuration” window (See: “Changing User Preferences” section of this manual)

- The last Raduga Log file containing the error messages. The log file is created in the log file directory. The log file directory path can be found in the “Private Configuration” window (See: “Changing User Preferences” section of this manual) or under the HKEY_CURRENT_USER\Software\Raduga6\LogFile registry key).

The support team will require answers to the following questions:

What is the error message? (Provide a full error message or a screenshot)

What is the problem? (Provide a clear description of the problem)

When does the problem occur? (Provide a full list of the steps to perform until the problem happens)

What is your server operating system? (Ask your system administrator to run “uname -a” command on the server and provide the output).

Known Issues

Discoverer business areas, folders and item classes are not migrated correctly

Symptom

You migrate a business area, including a folder with joins to other business areas or an item class.
After the migration the reports based on the migrated objects finish with errors.

Explanation

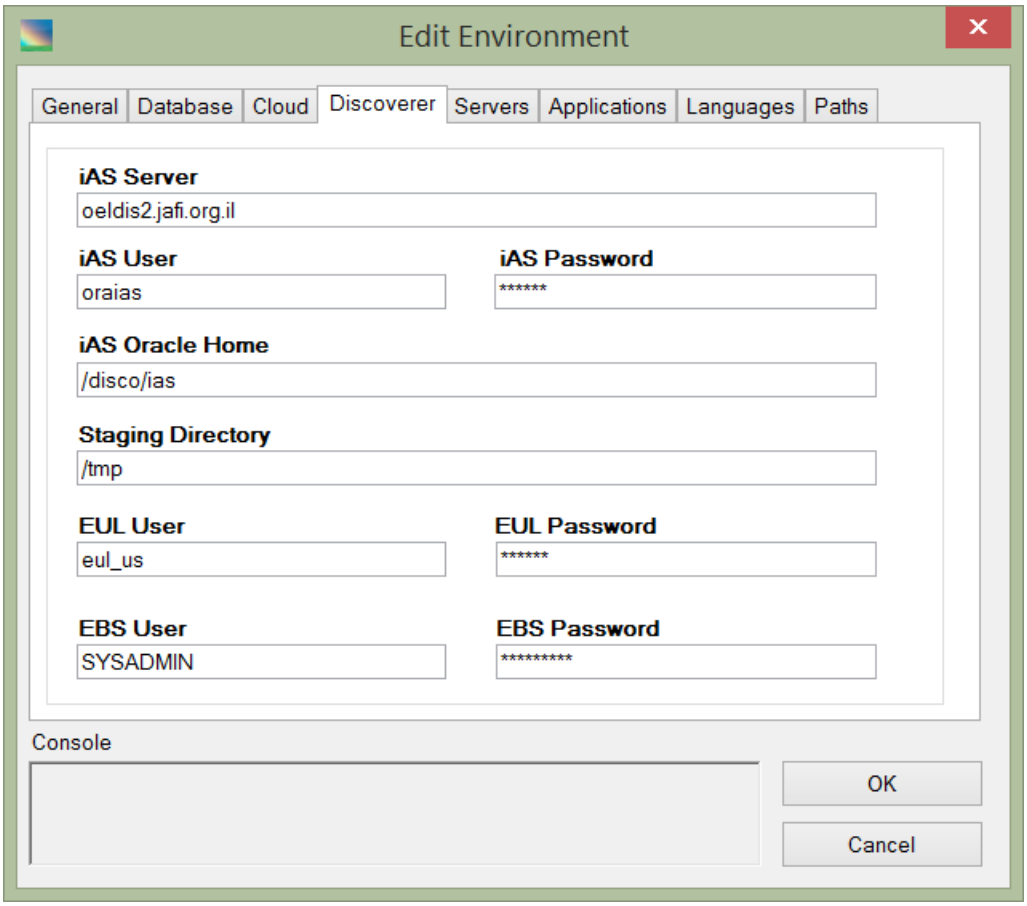
The discoverer command line API must connect to the database using Oracle Applications user in APPS mode to migrate folders, business areas and item classes correctly.

Solution

Define “EBS User” with discoverer admin privilege (usually SYSADMIN). Raduga will automatically grant this user the “all_admin_privs” discoverer privilege as well as the “business_area_admin_access”privilege for discoverer business areas.

You can open the environment definition form using the Global Configuration window (Admin → Global Configuration):

Define the EBS User under the “Discoverer” tab in the “Edit Environment” form:



Discoverer business areas and folders are not found during the migration

Symptom

You migrate a business area or a folder and the process fails with the following error:

Failed to find business area <BA Key>...
Failed to find folder <Folder Key>...

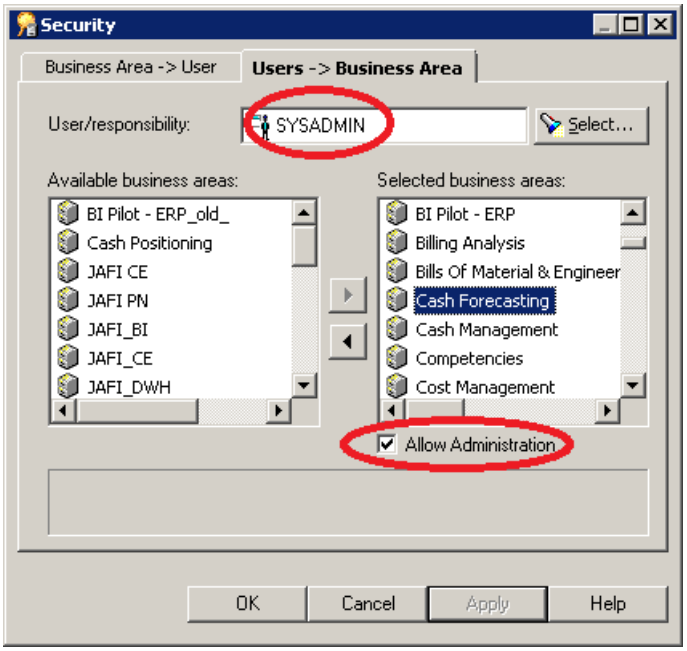
Explanation

The user (usually SYSADMIN) migrating the entity does not have permission for it.

Solution

Add corresponding permission to the Oracle Applications user in the Discoverer Administrator:

Choose Tools → Security.
In the opened dialog choose the "User → Business Area" tab.
Select the user that is used for migrating Business Area or Folder (usually SYSADMIN).
Move the Business Area that is going to be migrated to the "Selected business areas" panel.
Select the "Allow Administration" check box.
Click "Apply"



Data Loader file is not created on the server during the forms recording session

Symptom

You choose "DataLoad" task and press "Record".
After you perform operations in Oracle Applications the "FLD" file is not created on the server.

Explanation
This problem is caused by incorrect Oracle Forms behavior.

Solution
Delete Java temporary files.

- Go to the Control Panel and the choose Java icon.
- In the “General” tab choose “Settings”.
- In the “Temporary File Settings” window, press “Delete Files…”
- Check the “Applications and Applets” and “Trace and Log Files” check boxes (they should be already checked).
- Press “OK”.

Playing the Data Loader file fails with “invalid applications password” error

Symptom
“Cannot complete applications logon. You may have entered an invalid applications password, or there may have been a database connect error.”

Explanation
“Signon Password Case” profile is set to “Sensitive”.

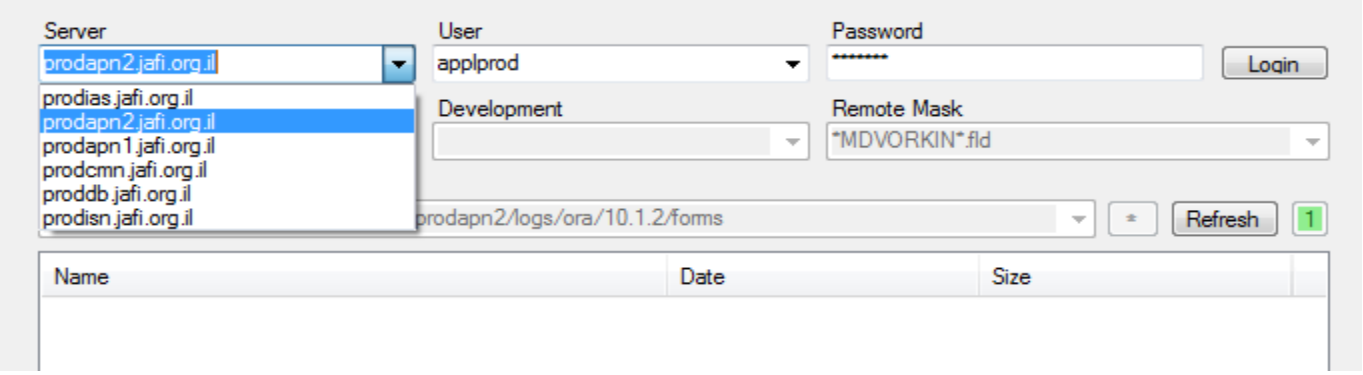
Solution
Change “Signon Password Case” profile to “Insensitive” at the site level.
You may also update the Oracle Applications user password in the DataLoad Professional (Tools → Options, Forms Playback tab)

Data Loader Recorded file is not found on the server after recording session

Symptom
Pressing “Refresh” button does not show the file in the right Raduga panel.

Explanation
In an environment with several web servers and a load balancer the recording file can be placed on a random web server.

Solution
To access the recorded data loader file in the environment with several web servers and a load balancer you need to choose the web servers one by one and press “Login”:



When the right server is chosen the recorded file appears in the right Raduga panel list.

Failure to get ebs.Framewrk_Personalization translated objects from server

Symptom

java.sql.SQLException: ORA-06502: PL/SQL: numeric or value error: character string buffer too small
ORA-06512: at "APPS.JDR_MDS_INTERNAL", line 1390
ORA-06512: at line 1

Explanation

Oracle package JDR_MDS_INTERNAL defines “compref” variable too short

Solution

Ask Oracle support to create a one-off patch as follows:

Change the “compref” variable definition in exportXLIFFDocument function of JDR_MDS_INTERNAL package

Before the change:

compref jdr_attributes_trans.atl_comp_ref%TYPE;

After the change:

compref varchar2(1000);

Failure to refresh APEX objects list

Symptom

APEX_SCHEMA variable is not defined. Please ask Raduga Administrator to define it

Explanation

APEX_SCHEMA variable should be defined for the database OS user

Solution

Define APEX_SCHEMA variable. See “Defining APEX Schema” section of Raduga Administration Guide

For Further Information

For any questions regarding this product, contact us at support@LazyDeploy.com, tel. +79185402272, or visit Raduga's web site: http://www.LazyDeploy.com