

Roundtable White Paper

Creating R-Code Only Deployments

Summary

Question: How do I create an R-Code only deployment if I use RTB (Roundtable)?

RTB's standard deployment process also allows you to create incremental r-code deployments. You create the package in much the same way as you would without Roundtable, except that you can create a package that will contain only changed r-code.

Note: You can eliminate the need to track CRCs by using the Progress Toolkit with RTB. RTB integrates with the toolkit to automatically encrypt your source so that it can be compiled by any Progress license. The entire application can be incrementally deployed then compiled (only objects that need compiling) at the end-user site, even with a run-time license.

CRC Compatible Deployment in General

When r-code is created, its CRC matches the table it is compiled against. If the table then changes, the r-code is no longer valid and cannot be used. The value of the CRC in the table is dependant upon many items, such as the order and number of schema changes applied. If you compile against your own database to generate the r-code, it may not have the same CRC as the table at your end-user's site (even if the schema is an exact match).

You generally create R-code deployments by maintaining a database that mimics the database at the customer's site. You always apply a .df against your database in the same manner that will be applied at the customer's site. You then compile and create your r-code against this CRC compatible database. This way, your r-code CRC value will be correct for the customer's database.

A detailed description of CRC compatible r-code deployments in general can be found at the peg (www.peg.com).

Roundtable Deployments

RTB incremental deployments create a package with just the changed source, the .df files, and some support files. You apply this package against your r-code compatible database (described above). RTB's deployment update procedure (_update.w) allows you to specify an alternate root directory where RTB will place r-code during the compile. This feature enables you to create a small, incremental r-code package that is in CRC sync with the customer.

Steps to Create the R-Code Deployment

1. *Create the deployment in RTB.*

RTB tracks individual sites or groups of sites for you. Use the RTB site tracking capabilities to create a deployment.

2. *Keep deployment in a directory that will be saved like the CRC DB.*

Place the deployment in a directory that you will keep along with your CRC compatible database. You will keep the application as well as the database for r-code generation.

3. *Run `_update.w` against your r-code compatible DB to update the schema.*

Use `_UPDATE.W` to apply the `.DF` files to your r-code compatible database. Since you will apply the same `.df` files against the real end-user DB, they will stay in CRC sync.

4. *Generate an r-code package.*

You will continue with `_UPDATE.W` to compile your application. Before compiling, V8 RTB will ask you to specify an alternate r-code root directory for the deployment (NOTE: In V9.0A and higher, you must enable the prompt for an alternate r-code directory by removing comments around the alternate r-code code blocks – search for “r-code” in the `_update.w` procedure). During the compile, all r-code is moved to this root directory (with relative paths intact).

NOTE: Retain this deployment directory so that you can apply incremental updates to the schema and the application in the future (the support files in `rtb_inst` directory must be retained for the next incremental compile).

5. *Update at the user's site.*

After performing an incremental compile, your “alternate root compile” directory will contain only the needed r-code (including relative paths) that you can zip up and send to the customer. You must run `_update.w` at the customer's site to apply the `.df` files to keep the database in CRC sync with our own.

NOTE: You do not have to use RTB's schema deployment. You can always skip the schema portion of `_update.w` and create your own incremental `.df` files between your deployment workspace DB and the CRC compatible DB (using the Progress Data Admin). You can then apply this incremental `.df` to your CRC compatible DB and to the end-user DB. You then only need `_update.w` to generate the r-code itself.

Non-Compilable Objects

You may also want other types of objects added to your package. For example:

- * Binary objects, such as bitmaps.

- * "Source" objects, such as programs that must be distributed as source (r-code is not created).

The RTB deployment programs look for a file called OBJCOPY in the RTB_INST directory of your deployment. Each file listed in OBJCOPY will be automatically moved to the alternate r-code directory (and therefore included in your package).

In V9, the OBJCOPY file is maintained automatically. If you have an object that you wish to move with your r-code, toggle on the OBJCOPY attribute for the object. RTB will automatically generate the OBJCOPY file in the deployment

In V8, you must create and maintain this file. Each object name must be placed on a separate line and must consist of the relative path and physical name of the object. You must check this object into RTB in a workspace Module with a directory of RTB_INST so that it is deployed with your application.