

WCS Build and Release Process

Aptean Ltd
Copyright © 2011-2026.

Contents

- 1 WCS Build and Release Process.....1**
- 1.1 Scope.....1
- 1.2 Building the programs.....1
- 1.3 Building the Release.....4
- 1.4 Releasing.....5
- 1.5 Sample Release Note.....6

1 WCS Build and Release Process

The objective of this document is to clearly describe the steps required to build and release programs to the client's site for an update to Calidus 3pl-Mobile (the WCS).

1.1 Scope

This process applies to all sites that use Calidus 3pl-Mobile and require releases, but do not utilise an automated release mechanism.

Where the document references updating the Supimix call logging system to particular statuses, the standard Supimix updating procedure applies.

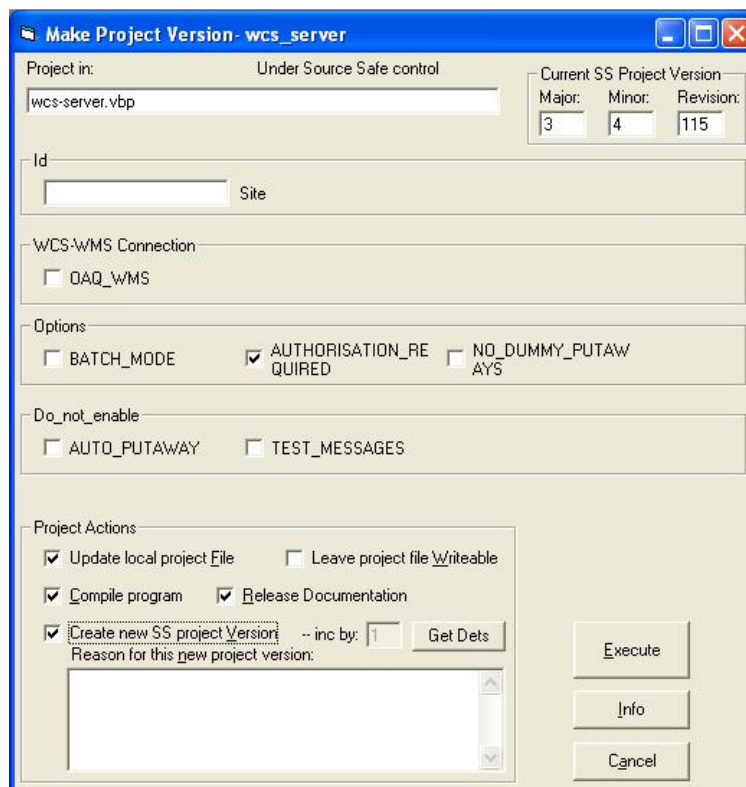
1.1.1 Responsibility

It is the responsibility of the person completing the testing that the programs be released out to appropriate representatives.

1.2 Building the programs

Once the changes for a particular log/number of logs are completed, the affected programs are built on the test server, using a utility for the purpose. The utility, Version Maker, is run directly from the programming environment. The interface presented depends on the program being built:

WCS:



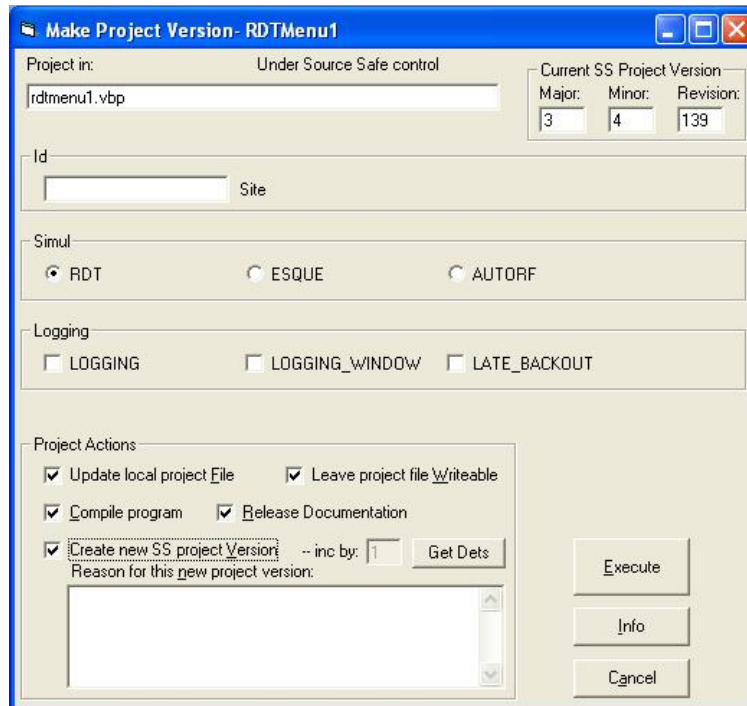
The screen shot shows the settings needed for a build of WCS-Server.exe. To build WCS-Server_OAQ repeat the call with OAQ_WMS ticked (after saving version number change to the .vbp file). Normally the same version number is



wanted for both so ?Create new SS project version? is un-ticked on the OAQ call.

The build is normally preceded by a ?Get Dets? (see button above). This extracts from VSS details of the changes included in the build (delete any duplicates at this stage). This information is automatically written to the release documentation.

RDT:

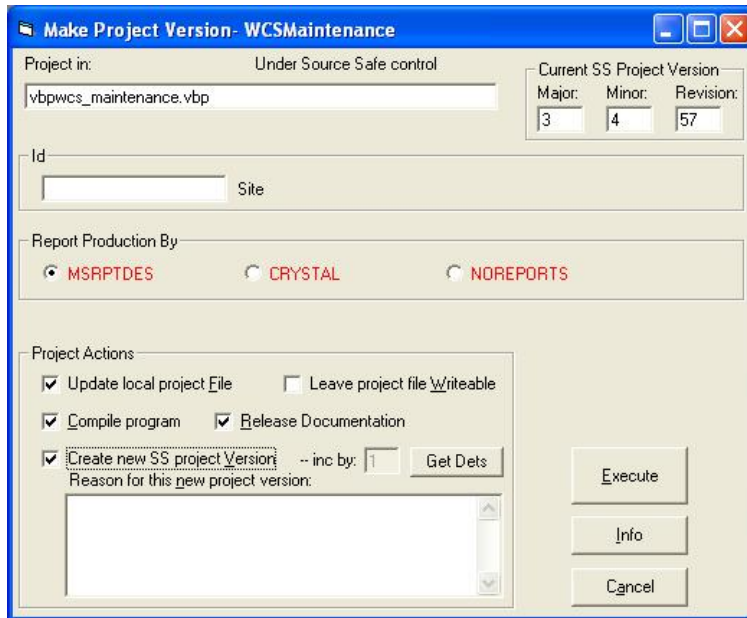


The screen shot shows the settings required to build RDTMenu1.exe. To build Debug.exe click ESQUE and tick LOGGING and LATE_BACKOUT Normally the same version number is wanted for both so ?Create new SS project version? is un-ticked on the Debug call.

The build is normally preceded by a ?Get Dets? (see button above). This extracts from VSS details of the changes included in the build. This information is automatically written to the release documentation.

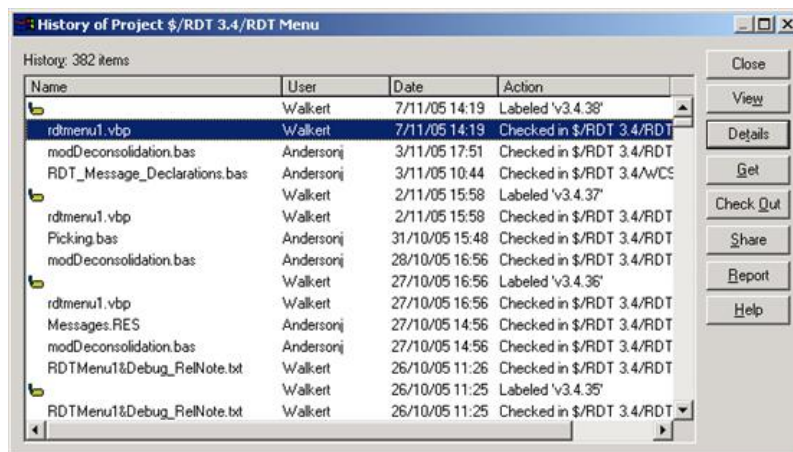
WCS Maintenance:





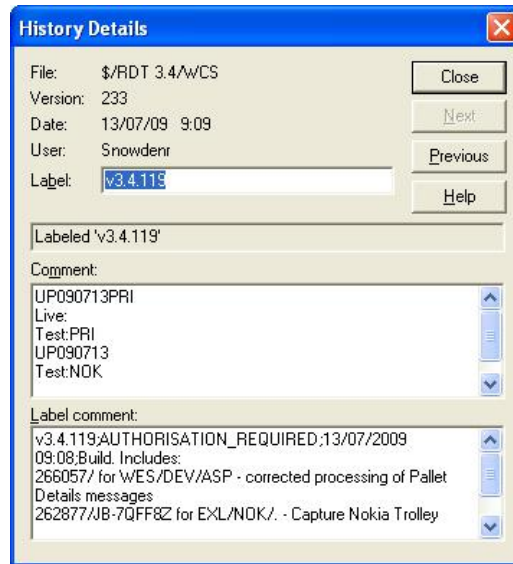
The build is normally preceded by a ?Get Dets? (see button above). This extracts from VSS details of the changes included in the build. This information is automatically written to the release documentation

Creating a new SS project version automatically labels the release in the source control environment.



Each label contains the build description, plus the parameters used to create the build:





The comment should be updated when the build is sent to the client, showing the sites to which the program has been released.

The programs are then moved to a test area and tested.

Update the Status of the Supimix call to ?Test?.

The test will be on a defined test machine, linking to a suitable Calidus-3pl test environment.

Once tested, A Proof of Testing document will have been created in the standard Test Logs directory, and the Supimix call will have been updated to Tested.

1.3 Building the Release

Once the build has been approved for release, all the programs concerned are moved to a releases area.

This releases area is in the network area:

```
P:\RDT\Development\_Installers
```

Under this area, there is a "Releases In Progress" area - the programs should be copied in here while they are being put together in a release.

Note: If the database, Server or RDT process has changed, all three must be released at the same time, to prevent potential issues when a client doesn't release all available patches.

A release notes file is also created, an example of which is listed in Appendix A

It is recommended that the "Changes since last build" section is created by copy and pasting the release notes generated automatically by Version Maker. These will probably contain duplicate lines which should be deleted.

The programs to be released and the release notes should now be included into a zip file, named as follows:

```
UP<YYMMDD>{<CLIENT>}{<SEQUENCE>}.zip
```

CLIENT is optional and should only be used if the release is to a specific client or site only. This is the site from the Supimix call (e.g. CHE, CHE3). Upgrades should only be released to specific clients if:

- The problem requires immediate fixing and it is impractical to upgrade to the latest version, or;



- The site is on a bespoke version of the WCS

SEQUENCE is optional, used only if several releases are made on the same day.

The naming convention for release notes is <RELEASE> relnotes.txt.

When the patch is built, update the status of the Supimix call to 'Ready for Release'. Update the Patch field with the name of the patch file.

1.4 Releasing

Note: The following is a standard release process that should be followed for all new clients. There are some existing clients that require the patch to be released via email.

For a list of current clients and their release mechanisms, see Appendix B

FTP the patch out to clients? identified machine. Existing clients have shortcuts to release to specific machines.

Send an email to the general WCS list and the defined representatives of the specific client, and copy it to the members of the OBS WCS team. The email should identify the release name, and the place the release can be found, and include a copy of the release note.

Copy the release and release note to the client area in the releases area

```
P:\RDT\Development\_Installers
```

Under this area there are specific company areas. For all new installations, this will be the company from the call (e.g. EXL)

Under this area, the site or client is specified. For all new installations, this will be the site from the Supimix call (e.g. CHE, CHE3)

If the release is for all clients in a company (and the company has several sites), the release can be placed in a non-site-specific directory, usually above the site-specific areas. For example:

```
P:\RDT\Development\_Installers
```

```
\EXL                - Company area
\_Sites             - Releases for all sites
\COV                | Site areas
\CHE                |
\CN-ATH            |
```

Update the SourceSafe label to show that the release has been made and which sites have it in test.

Once this is complete, update the status of the Supimix call to ?Acceptance Test? (5), sub-status ?Complete? (70).

For instructions on releasing patches directly onto client servers, please see the C3PL-M Installation Guide, referenced as item 1 in Appendix C



1.5 Sample Release Note

See also [Creating a WCS Patch](#)

Files included in release, plus versions:

Program	Version
rdt1_struct.mdb	ss v107
RDTMenu1.exe	3.4.30
Debug.exe	3.4.30
WCS-Server.exe	3.4.23
WCSMaintenance.exe	3.4.15

Changes since last build:

Log Number	Programs	Description
198788	RDTMenu1/ WCS-Server/ WCS Maintenance	Deconsolidation and Despatch development
206084	WCS-Server	Fix to creation of picking containers when short picking; fix to receiving stock drip feed messages.

Installation Instructions:

Stop all RDTs running on the system.

Stop the WCS server on the TEST system.

Stop all WCS Maintenance sessions accessing the database

Make a backup copy of the old program, if required.

Copy the released versions of this program to the correct area on the TEST machine,

normally C:\Program Files\Warehouse Control Server

If the database structure has been released to you, the database should be converted as described in the WCS Installation guide.

Restart the programs.

Note: There are WMS portions of some of these changes. Please check WebRelease for:

198780/ML-6C2EZD

198790/ML-6C9BKD

198776/ML-6C2DWD

