

# AN418

## Remote Application Update for the RCM33xx

The common method of remotely updating an embedded application is to write directly to parallel flash. This is a potentially dangerous operation because if there is power loss while writing, subsequent execution can not be guaranteed. Our download manager never writes to parallel flash, making it a much more reliable method for updating firmware than other solutions.

The RCM3300 family is the only hardware that supports this software, i.e, RCM3300, RCM3305, RCM3309, RCM3310, RCM3315, RCM3319. This application is not supported on any other Rabbit hardware.

**WARNING:** It is important that the download manager has room on the serial flash to install its partition. Verify that there is room before using the Download Image Generator to download an image to the RCM33xx.

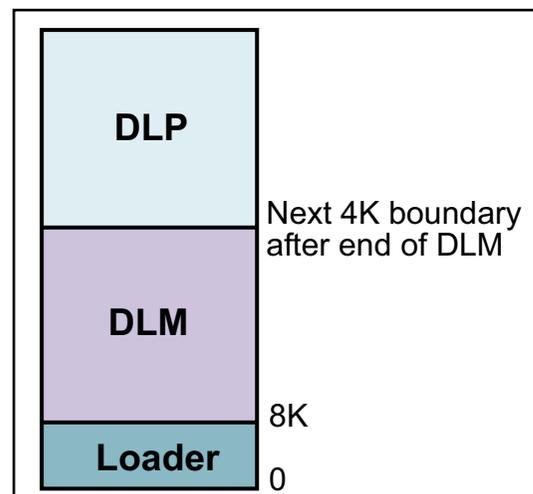
### 1. Overview

A remote application update is accomplished by several different pieces of software working together. On the device itself, there is a flash resident program loader that occupies the lowest 8K of parallel flash. The only responsibility of this loader is to load a program from storage in flash (parallel or serial) to fast RAM and then start the loaded program running. In addition to this loader, the parallel flash also contains two executable images: the download manager (DLM) and the downloaded program (DLP). These two images start on even 4K boundaries and are not runnable from flash; they can only be run in fast RAM.

Figure 1 shows the contents of the parallel flash.

**Figure 1. Parallel Flash Contents**

A separate GUI utility program is used to build these pieces into a single downloadable binary. This utility allows the user to specify the location of the source code for the loader, the DLM and the DLP. It then handles invoking the command line compiler to separately compile the DLM and DLP and then combines those images in the compilation of the loader which results in the full downloadable image. The utility can also invoke the command line RFU to load the full image via a serial connection. The utility is described later in this document: [Section 6 "Download Image Generator Utility."](#)



The DLP can be any application program that has been modified to use the remote upload library and to call a couple of functions. The library that is used in the DLP handles the details of invoking the loader to load the DLM once the user is validated.

The DLM is a program that is capable of receiving a new image and storing it in serial flash for later execution. The DLM can receive new programs via an easy-to-use web interface and will invoke the loader to load and execute a stored program when requested by the user.

**NOTE:** It is important that both the DLM and DLP be compiled with the “Code and BIOS in flash, Run in RAM” compiler option set, and that both must be compiled with separate Instruction and Data space disabled. The loader must be compiled with the “Code and BIOS in Flash” option set, and also must be compiled with separate Instruction and Data spaces disabled. Compile-time errors will be generated if these options are not set correctly.

## 2. Step-by-Step Walk-Through

The following steps will walk you through compiling and downloading a full Loader/DLP/DLM image and compiling and sending a DLP to the DLM. Further information for each step can be found later in this document.

1. Ensure that the serial flash on the supported RCM33xx board is ready to work with the Remote Application Update sample. This can be accomplished by compiling and running the standard sample program: `Samples\RCM3300\SerialFlash\sflash_inspect.c`. Clear pages 0 through 5, inclusive, on the serial flash.
2. If the custom DLP will use a portion of the serial flash for a FAT file system, select “Code and BIOS in Flash, Run in RAM” compile mode and deactivate the “Enable separate instruction and data spaces” check box. Compile and run the standard sample program: `Samples\RCM3300\RemoteApplicationUpdate\dlm_fat_format.c`
3. Open Dynamic C.
4. Select the “File > Project > Open...” menu item. Using the “Open” dialog window that pops up, browse to the main Dynamic C installation directory (if not already there) and open `Applications\RCM3300\RemoteApplicationUpdate\loader.dcp`. This is a Dynamic C Project file.
5. Select the “Options > Project Options” menu item. Click on the “Targetless” tab. On the “Board Selection” tab, select the RCM33x0, RCM33x5 or RCM33x9 board type that is appropriate for your application.
6. Click on the Project Options’s “Communications” tab. Ensure that the “Serial Options” (especially the “Serial Port” and “Use USB to Serial Converter” items) are correct for the COM port that connects to the supported RCM33xx board. Click OK.
7. Select “File > Project > Open...” again. Open `dlmdlp.dcp` (same folder as `loader.dcp`).
8. Select the “Options > Project Options” menu item. Click on the “Targetless” tab. On the “Board Selection” tab, select the RCM33x0, RCM33x5 or RCM33x9 board type that is appropriate for your application.
9. Click on the Project Options’s “Communications” tab. Ensure that the “Serial Options” (especially the “Serial Port” and “Use USB to Serial Converter” items) are correct for the COM port that connects to the supported RCM33xx board. Click OK
10. Select the “File > Project > Close” menu item so Dynamic C will revert to using the default project file (`default.dcp`).

11. Close Dynamic C.
12. If Dynamic C is or has been installed in more than one location on the PC, open the GUI version of the RFU that is provided in the Dynamic C installation. Select the “Setup > File Locations...” menu item. Confirm that the pathlists are correct for each support file for the Dynamic C installation in use. Close the GUI RFU.
13. Open the Download Image Generator (`DownloadImageGenerator.exe`) found in the Utilities subdirectory. It will open on the Build page.
14. Fill in the path for the loader source file (`loader.c`) and the loader project file (`loader.dcp`). Look in the `Applications\Rcm3300\RemoteApplicationUpdate` folder for these files.
15. Fill in the path for the DLM source file (`dml_web.c`) and the DLM project file (`dml_dlp.dcp`). Look in the `Applications\Rcm3300\RemoteApplicationUpdate` folder.
16. Fill in the path for the DLP source file (`dlp_web.c`) and the DLP project file (`dlm_dlp.dcp`). Sample DLP source files are in `Samples\Rcm3300\RemoteApplicationUpdate`. The default DLP project file is in `Applications\Rcm3300\RemoteApplicationUpdate`.
17. Make sure that the network settings are set correctly in `lib\...\tcpip\tcp_config.lib`.
18. Click on the Download Image Generator's “Options” tab. Ensure that the paths to the command line compiler (`dcc1_cmp.exe` and `clrfu.exe`) are correct on the Options page. By default when the Download Image Generator starts, it assumes that it is running in the Utilities subdirectory under the main Dynamic C directory, and fills in these paths accordingly.
19. If using a COM port other than COM1, on the Options page de-select (un-check) the “RFU command Line:” text box’s “Use default” check-box. Then edit the RFU command line to append the appropriate “-s” command line switch. For example, to use COM3 at 115200 baud, append “-s 3:115200” to the end of the RFU command line (excluding the double quotes and noting the spaces surrounding the “-s” switch characters).
20. If using a USB programming cable or a USB to serial converter cable, on the Options page de-select (un-check) the “RFU command Line:” text box’s “Use default” check-box. Then edit the RFU command line to append the “-usb+” command line switch (excluding the double quotes and noting the space preceding the “-usb+” switch characters).
21. Connect a programming cable to the target and click the “Build All” button.
22. Ignore the compiler origin warnings and wait until the “Image Loaded Successfully” message appears at the bottom of the program window. Now disconnect the programming cable from the board and cycle power. The loader will load the DLM, which will take care of some bookkeeping, and then load the DLP. The DLP should be up and running a few seconds after cycling power to the board.
23. Verify that the DLP is loaded by browsing to `[Board's IP address]/reboot.html`. By default, the DLM and DLP use basic authentication, so the web browser should present a user name and password dialog. The default user name is “admin” and the default password is “reboot.”
24. Once the reboot page is loaded, click on the reboot link. This will cause the DLP to shut down, and the loader will reload the DLM. Once the reboot link is clicked, the DLP sends a page to the browser that includes a refresh and redirect tag. After 5 seconds, the browser should refresh and redirect to the main page of the DLM interface.
25. Using the Download Image Generator, change the DLP source file path to `dlp_static.c`, which is located in the `Samples\Rcm3300\RemoteApplicationUpdate` folder. This sample is the same as `static.c` from the `Samples\tcpip\http` directory except that it has been turned into a DLP. Click the Build DLP button.

26. Using the web browser, click on the Upload link from the main page of the DLM.
27. On the upload page, click on the Browse button and navigate to the `Sample\RCM3300\RemoteApplicationUpdate` folder, and choose `dlp_static.dlp`.
28. Click the Upload button, and the binary will be uploaded to the DLM and stored in the serial flash.
29. Click on the Home link to return to the main DLM page. The DLP that was just uploaded will appear as the first item in the list of serial flash images.
30. To start the uploaded DLP, click on the Start button next to the name of the DLP.
31. A page will be shown indicating that the DLP has been successfully loaded. The DLP should be up and running within a few seconds.
32. Set the web browser to the board's IP address. If the browser shows the main DLM index page force it to refresh (click the refresh button or press F5). The web browser should now be showing the same html page as the `static.c` sample shows.

### 3. DLP Essentials

This section describes the basic requirements of a DLP. It is very important that a DLP follow the steps outlined here, and that a program that does not follow these steps is never uploaded to the DLM. The functionality provided in `remoteupload.lib` ensures that a DLP will be able to restart the DLM. If a program is uploaded to the DLM that does not include the functionality provided in `remoteupload.lib`, the deployed hardware may never be reachable remotely again.

Sample DLPs are provided in the `Samples\Rcm3300\RemoteApplicationUpdate` folder.

#### 3.1 Basic Skeleton of a DLP

The **boldface** lines in the following code are what distinguish a DLP from a non-DLP.

```
#define TCPCONFIG 1 // 1. default network configuration
#define DLP_USE_HTTP_INTERFACE // 2. enable http server
#define HTTP_NO_FLASHSPEC // 3. no static resources
#use "dcrtcp.lib" // 4. access necessary libraries
#use "http.lib"
#use "remoteupload.lib"
SSPEC_MIMETABLE_START // 5. required for the http server
    SSPEC_MIME(".html", "text/html"),
    SSPEC_MIME(".cgi", "")
SSPEC_MIMETABLE_END
void main() {
    sock_init(); // 6. initialization functions
    http_init();
    tcp_reserveport(80);
    dlp_init();
    for(;;) { // 7. main program loop
        tcp_tick(NULL);
        http_handler();
        dlp_handler();
    }
}
```

Looking at the code for this DLP, there are several things to note. First, the predefined TCP/IP configuration is chosen (1). For instructions on how to set the configuration, refer to `lib\...\tcpip\tcp_config.lib`. This DLP includes the web interface by defining `DLP_USE_HTTP_INTERFACE` (2). A custom interface may be used for the DLP in addition to or instead of the web interface.

The macro `HTTP_NO_FLASHSPEC` is defined when the HTTP server has no static resources. Defining this macro reclaims the memory that would otherwise be used for the static resource table. Since the DLP in this example has no other web content except for the DLP interface, `HTTP_NO_FLASHSPEC` is defined (3).

If a custom interface is being used and the HTTP server is not needed, it is not necessary to define `HTTP_NO_FLASHSPEC`. If the application being turned into a DLP already makes use of the HTTP

server and it uses static resources, you must not define `HTTP_NO_FLASHSPEC`. In this case, make sure that `SSPEC_MAXSPEC` is large enough to accommodate the two DLP resources: one html page and one CGI. `SSPEC_MAXSPEC` is the number of entries in the dynamic resource table. It defaults to ten.

Next are the `#use` statements for the main Rabbit TCP/IP library, the HTTP library, and the remote upload library (4). The `#use` statement for the HTTP library is only required if using the web interface for the DLP, or if the DLP requires the web server.

The MIME table associates resource content types with a file-type resource. The table is created using the `SSPEC_MIME*` macros; it is only necessary if the DLP is using the web interface (5).

In `main()`, all of the necessary initialization functions are called (6). If the Web interface is not being used, `http_init()` and `tcp_reserveport()` do not need to be called. Once the initialization is complete, the main body of the program drives the network interface(7). If the web interface is not being used, `http_handler()` does not need to be called. Conversely, if the web interface is being used, `http_handler()` must be called. In that case, calling `tcp_tick()` is redundant because it is called inside `http_handler()`.

## 3.2 DLP Specific Macros

There are several macros that can be defined at the top of a DLP or in a DLP project (via the Project Options dialog on the Defines tab page). The following macros are specific to the DLP.

### **DLP\_USE\_HTTP\_INTERFACE**

Specifies that the web interface for the DLP is to be used. Defining this requires that all of the HTTP portions in the skeleton program given in [Section 3.1](#) be present.

### **DLP\_REBOOT\_CGI\_REDIRECTTO (s)**

If the web interface is being used, the page that gets served in the process of rebooting includes a meta refresh tag so that the browser is redirected to the home page of the download manager.

Changing this macro will change where the browser is redirected after a reboot to the download manager. For instance, to redirect to `www.mysite.com:8080`, define this macro as follows:

```
#define DLP_REBOOT_CGI_REDIRECTTO(s) "www.mysite.com:8080"
```

### **DLP\_REBOOT\_CGI\_REFRESH\_SEC**

This macro controls the number of seconds the browser should wait before refreshing when redirecting to the download manager's home page. The default is 5 seconds.

## **DLP\_USE\_CUSTOM\_INTERFACE**

Specifies that a custom interface for the DLP is to be used. A custom interface can be used in addition to or instead of the web interface. If a custom DLP interface is being used, then one or both of the following must be done:

- Set `dlp_do_reboot` to 1 when custom reboot criteria are met and ensure that `dlp_handler()` is called regularly (`dlp_handler()` will see the flag set to 1 and cause a reboot),
- Call `dlp_rebootdml()` directly when reboot criteria are met.

Failure to meet these requirements may result in a deployed program that is not able to reboot back to the DLM.

## **4. DLM Essentials**

This section describes the basic requirements of the DLM. The source code for the DLM is located in `Applications\Rcm3300\RemoteApplicationUpdate\dml_web.c`.

### **4.1 Basic Functionality**

The DLM has two main purposes. First, it is responsible for receiving new DLP uploads from a remote client and storing the DLP image in serial flash, and then invoking the parallel flash resident loader to start a selected DLP image. (The user interface for uploading new images is covered in [Section 7](#).) Second, the DLM informs the user of events and problems via configurable email alerts.

During initialization, the DLM can take several different courses of action. Each time the DLM is restarted, the reason for restart is examined. In the case of restart due to hardware reset or watchdog timeout (which will occur for runtime exceptions in the DLP), the DLM will log the timestamp of the restart to the User block, and will send an email alert if so configured. The DLM differentiates between hardware resets and resets due to a watchdog timeout and can be configured to allow different numbers of each type of reset in different time spans. By default, the DLM will allow 10 hardware resets within 3 days and 1 watchdog timeout within 1 day. If the DLM is restarted more than 10 times due to power failure in 3 days or more than 1 time due to a watchdog timeout in 1 day, the currently active DLP will be marked as invalid and will no longer be allowed to run. The DLM can be configured to automatically fall back to the next newest DLP stored in serial flash. Once a restart limit has been reached, an email alert will be sent (if configured) stating that the limit was reached and that an older DLP may be loaded for execution.

If the download manager is not restarted due to a hardware reset or watchdog timeout, it will next attempt to initialize the serial flash. If serial flash initialization fails for any reason, the DLM will send an email alert (if configured to do so) indicating that serial flash initialization failed. The serial flash must have a large enough available partition. The size of the partition defaults to one megabyte, but can be changed with a configuration macro. If the flash contains a single non-DLM partition which consumes the whole flash, the download manager will not work correctly.

If serial flash initialization is successful, and a restart limit has not been reached, the download manager will attempt to start the current active DLP image from serial flash. If the manager is being started for the first time, or if there are no DLP images stored in serial flash, it will attempt to start the DLP image stored in parallel flash that was built as part of the initial Loader/DLM/DLP image.

Finally, if a restart limit has been reached and the download manager is configured to revert to an older DLP, it will attempt to locate the next newest DLP and invoke the loader to start the image.

## 4.2 DLM Specific Macros

There are several macros that can be defined at the top of the DLM or in the DLM project (via the Project Options dialog on the Defines tab page). The following macros are specific to the DLM.

**DLM\_WATCHDOG\_RESTART\_LIMIT** (Default is 1)

**DLM\_WATCHDOG\_RESTART\_TIME\_UNIT** (Default is DLM\_RESTART\_DAY)

**DLM\_WATCHDOG\_RESTART\_TIME\_UNIT\_COUNT** (Default is 1)

These macros are used together to determine the number of times a DLP can be restarted within a specified number of seconds due to a watchdog timeout, secondary watchdog timeout (if configured) or runtime error.

The **DLM\_WATCHDOG\_RESTART\_TIME\_UNIT** macro can be any of the following:

**DLM\_RESTART\_MINUTE**, **DLM\_RESTART\_HOUR**, **DLM\_RESTART\_DAY**,  
**DLM\_RESTART\_WEEK**, **DLM\_RESTART\_MONTH**, or **DLM\_RESTART\_YEAR**.

The **DLM\_WATCHDOG\_RESTART\_TIME\_UNIT\_COUNT** macro is a multiplier for **DLM\_WATCHDOG\_RESTART\_TIME\_UNIT**. By default, a DLP can be restarted one time within one day due to a watchdog timeout before the DLM decides to take action.

**DLM\_RESTART\_LIMIT** (Default is 10)

**DLM\_RESTART\_TIME\_UNIT** (Default is DLM\_RESTART\_DAY)

**DLM\_RESTART\_TIME\_UNIT\_COUNT** (Default is 3)

These macros are used together to determine the number of times a DLP can be restarted within a specified number of seconds due to a hardware reset or power failure. These macros can be defined with the same values listed in the section above. By default, a DLP can be restarted 10 times within 3 days, before the DLM decides to take action.

**DLM\_REVERT\_TO\_OLD\_DLP** (Default is 1)

If set to one, the DLM will attempt to revert to an older version of the DLP if the current DLP is not runnable.

**DLM\_START\_DLP\_AFTER\_LOAD** (Default is 0)

If set to one, the DLP will be automatically started after a successful load without any further user intervention.

**DLM\_PARTITION\_ONE\_MB** (This is the default)

**DLM\_PARTITION\_TWO\_MB**

**DLM\_PARTITION\_THREE\_MB**

**DLM\_IMAGE\_COUNT** (Default is 3)

The first three of these macros determine how much of the serial flash the download manager will partition for itself. The last macro specifies how the partitioned space will be divided. Define any of the first three, and set the last to an integer. If the macros are not configured, the library will default to partitioning one megabyte and dividing the partition into 3 images.

Make sure that there is room on the serial flash for the partition before installing the download manager on the parallel flash. If the DLM can not partition the serial flash, it will not be able to launch any DLPs.

**DLM\_NEW\_DLP\_RESETS\_RESTARTLIMIT**

If this macro is set to 1, the DLM will reset all restart limits when a new DLP is uploaded.

**DLM\_ALERT\_EMAIL\_ENABLE** (Default is 1)

This macro must be set to 1 in order for the DLM to send email alerts. Alerts are sent by the DLM when possible user intervention is required, e.g., if the DLP has been restarted the maximum number of times.

**DLM\_ALERT\_EMAIL\_ADDRESS\_TO** (Default is myaddress@mydomain.com)

This macro is a string that specifies the email address where DLM alerts are sent. This macro must be defined by the user in order to send email alerts.

**DLM\_ALERT\_EMAIL\_ADDRESS\_FROM** (Default is myaddress@mydomain.com)

This macro is a string that specifies the email address DLM alerts are sent from. This macro must be defined by the user in order to send email alerts.

**SMTP\_SERVER** (Default is mymailserver.mydomain.com)

This macro is a string that specifies the mail server.

**DLM\_ALERT\_EVERY\_DLM\_RESTART**

If this macro is set to 1, the DLM will send an alert each time it is restarted due to either a hardware reset or watchdog timeout. By default, this macro is set to 1 if email alerts are enabled.

**DLM\_ALERT\_EVERY\_DLM\_WDTIMEOUT**

If this macro is set to 1, the DLM will send an alert each time it starts after a watchdog timeout. By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_DLP\_RESTART\_LIMIT**

If this macro is set to 1, the DLM will send an email alert once the DLP has been restarted due to hardware reset or power failure the maximum number of times within the allowed time frame, as set forth by the `DLM_RESTART_LIMIT`, `DLM_RESTART_TIME_UNIT`, and `DLM_RESTART_TIME_UNIT_COUNT` macros. By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_DLP\_WDRESTART\_LIMIT**

If set to 1, the DLM will send an email alert once the DLP has been restarted due to watchdog timeout the maximum number of times within the allowed time frame, as set forth by the `DLM_WATCHDOG_RESTART_LIMIT`, `DLM_WATCHDOG_RESTART_TIME_UNIT`, and `DLM_WATCHDOG_RESTART_TIME_UNIT_COUNT` macros. By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_DLP\_RUNTIME\_ERROR**

If set to 1, the DLM will send an email alert if it is restarted due to a secondary watchdog timeout or runtime error the maximum number of times within the allowed time frame (the watchdog restart time frame is also used for runtime errors). By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_REVERT\_TO\_OLD\_DLP**

If set to 1, the DLM will send an email alert if it reverts to an older version of the DLP (if so configured). By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_DLP\_NOT\_RUNNABLE**

If set to 1, the DLM will send an email alert if the DLP image is not runnable, and will indicate the reason for not being able to run the image. By default, this macro is set to 1 if email alerts are enabled.

#### **DLM\_ALERT\_SFLASH\_INIT\_FAIL**

If set to 1, the DLM will send an email alert if it was unable to initialize serial flash on startup. By default, this macro is set to 1 if email alerts are enabled.

## 5. DLM/DLP Shared Macros

Several macros are used by both the DLM and DLP. Since the DLP and DLM can share the same project file, these macros are best defined in the Defines tab page of the Project Options dialog box so that the macro definitions are visible to the compilation of both the DLM and DLP. These macros can also be changed in `remoteuploaddefs.lib`.

### **RUPL\_USER\_NAME** (Default is “admin”)

This macro is a string that specifies the default user name for authentication when accessing password protected resources via the Web interface.

### **RUPL\_USER\_PASSWORD** (Default is “reboot”)

This macro is a string that specifies the default password for authentication when accessing password protected resources via the Web interface.

### **RUPL\_ADMIN\_GROUP** (Default is 0x0002)

This macro specifies the default user group for accessing resources within the web server.

### **RUPL\_HTTP\_SERVER\_AUTH** (Default is SERVER\_AUTH\_NONE)

This macro specifies the method used for authenticating users in the web interface. This macro can be `SERVER_AUTH_DIGEST` for digest authentication, `SERVER_AUTH_BASIC` for basic authentication, or `SERVER_AUTH_NONE` for no user authentication.

If digest authentication is used, then `USE_HTTP_DIGEST_AUTHENTICATION` must be defined to 1 prior to the `#use` statements for `remoteuploaddefs.lib` and `http.lib`.

### **RUPL\_SAVE\_RESTORE\_NETWORK\_PARAMS** (Default is 0)

If this macro is set to 1, the current network configuration will be saved to the User block when switching between the DLM/DLP, only if the configuration has changed. This allows the run time setting of the device's IP address from either the DLM or DLP and having the setting reflected in the other.

### **RUPL\_USER\_BLOCK\_OFFSET** (Default is 0)

This macro gives the DLM and the DLP the location in the User block where various pieces of configuration information can be stored. The User block will only be used for storing power-on parameters such as IP address, netmask, gateway, etc., and restart information. This macro defaults to 0 which is the start of the User block.

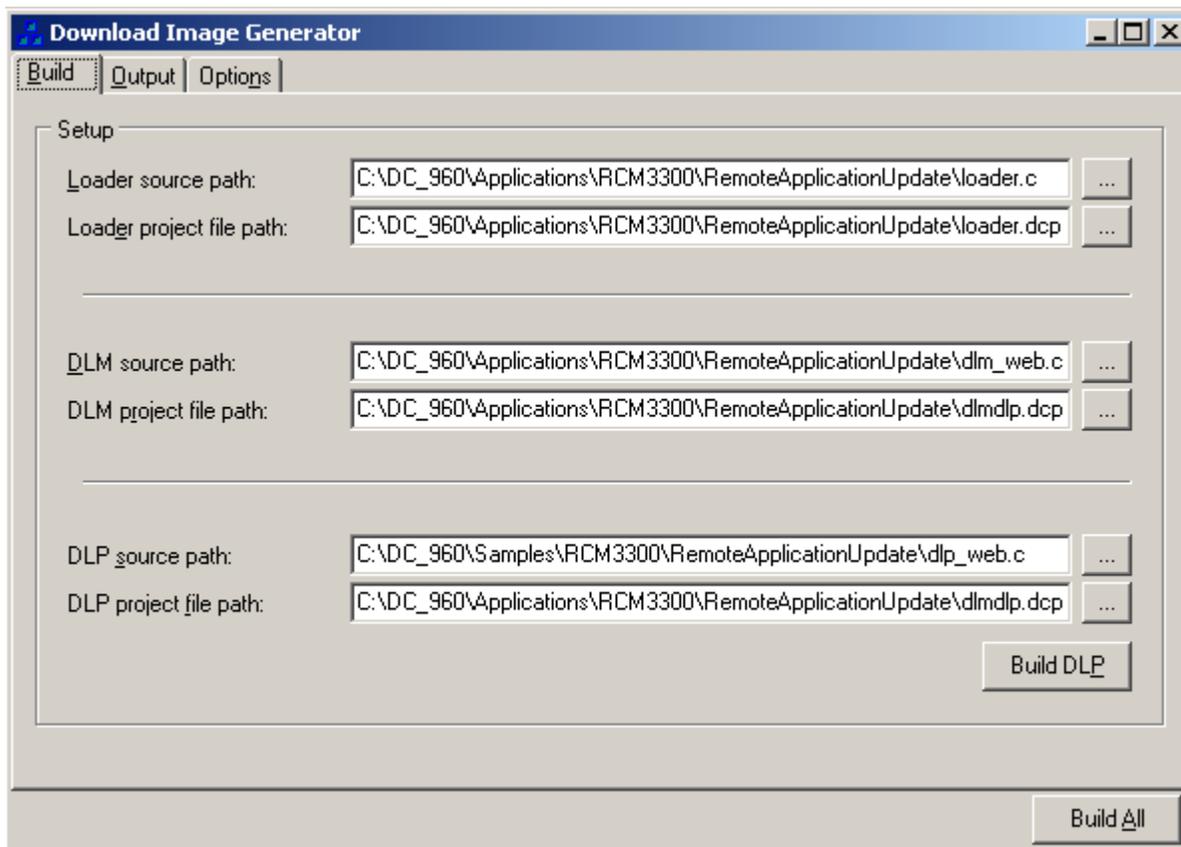
## 6. Download Image Generator Utility

The Download Image Generator Utility (`DownloadImageGenerator.exe`) is located in the Utilities directory under the main Dynamic C directory, and is used for generating binary images for use with the DLM. DLM and DLP images are stored with length and CRC information at the front and back respectively of the executable portion of the image. To build a parallel flash image that contains the loader, DLM, and DLP or a serial flash DLP image, this utility must be used.

### 6.1 Build Page

The Build page is where the pathnames for the loader, the DLM and the DLP are specified

**Figure 2. Build Page of the Download Image Generator Utility**



Loader source path - specifies the path to the loader source code.

Loader project file path - specifies the project file to use when compiling the loader source.

DLM source path - specifies the path to the DLM source code.

DLM project file path - specifies the project file to use when compiling the DLM source. The DLM and DLP can generally use the same project file.

DLP source path - specifies the path to the DLP source code.

DLP project file path - specifies the project file to use when compiling the DLP source. The DLM and DLP can generally use the same project file.

Build DLP - builds a `.d1p` file that can be uploaded to the download manager.

Build All - builds the initial binary image that must be placed on the parallel flash using a serial connection.

### 6.1.1 Building the Initial Loader/DLM/DLP

To put an initial version of the Loader/DLM/DLP combination onto the parallel flash of the desired target, fill in all of the edit boxes on the Build page as well as the Paths edit boxes on the Options page and click the Build All button. This will compile the DLM, DLP and Loader into one downloadable image, and will download the image to the target if the “Upload full image via RFU” checkbox is checked.

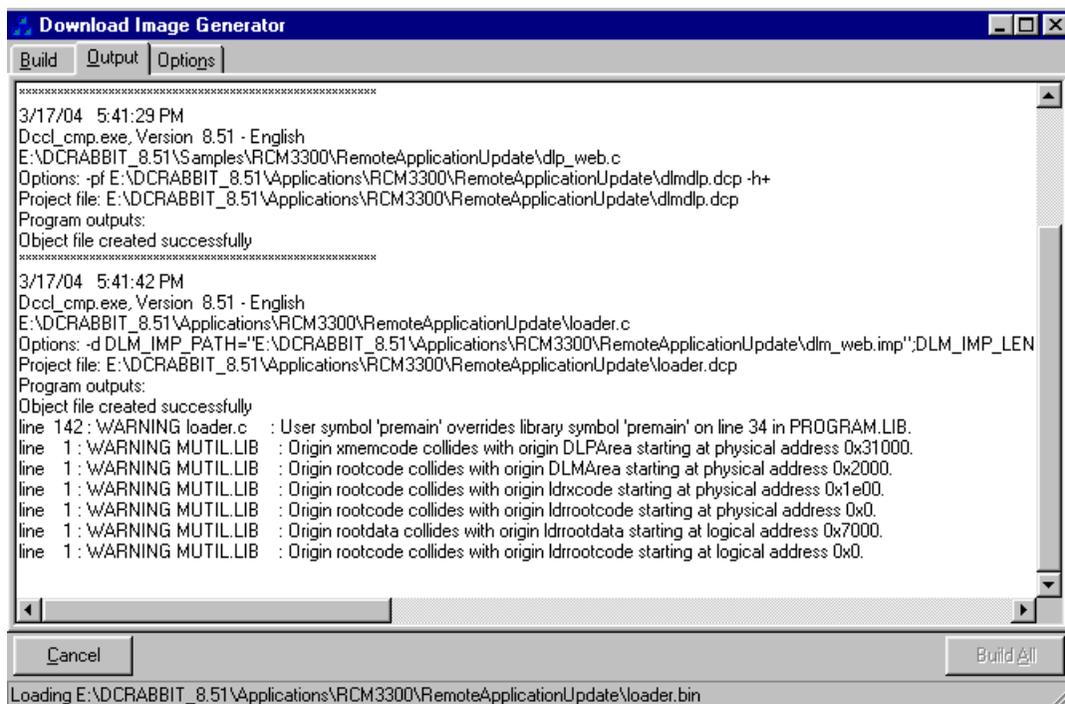
### 6.1.2 Building a DLP

To upload a new DLP to a running DLM, this utility must be used to create the .d1p file for the DLP. The DLM expects that uploaded DLPs will contain the length of the file as the first four bytes of the file, along with a CRC and a few other pieces of information at the tail end of the file. This utility takes care of all of the details that the DLM expects of uploaded files. To create a .d1p file, fill in the path to the DLP source and click the Build DLP button. The newly created .d1p file will be placed in the same directory as the DLP source file.

## 6.2 Output Page

The Output page shows the output from the command line compiler and from the command line RFU.

**Figure 3. Output Page of the Download Image Generator Utility**

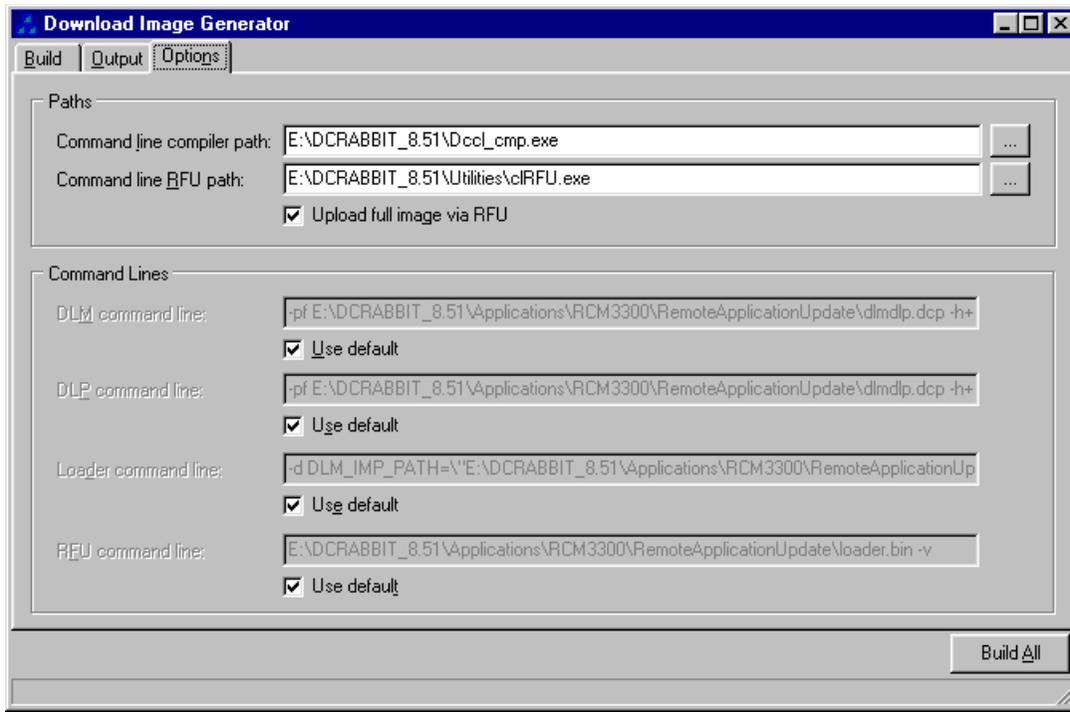


If any compilation errors are encountered, the utility will stop and not attempt to compile any more source or download a .bin file to the target. The origin warnings generated during the compilation of the loader can be ignored.

### 6.3 Options Page

The Options page allows you to change path names for the command line compiler and the command line RFU as well as the command lines that are passed to these programs.

**Figure 4. Options Page of the Download Image Generator Utility**



The utility assumes that it is starting from the Utilities directory under the main directory in Dynamic C and fills in the paths to the command line compiler and command line RFU accordingly. If the utility is being run from somewhere else, these paths will have to be filled in manually. The command line edit boxes will most likely not need to be used, but are provided to give extra control over the way that the command line compiler and command line RFU are invoked.

Command line compiler path - Specifies the path to the command line compiler executable (dcc1\_cmp.exe).

Command line RFU path - Specifies the path the command line RFU (clRFU.exe).

Upload full image via RFU - If this box is checked, the full binary image (Loader/DLM/DLP) will be uploaded via a serial connection using the command line RFU.

DLM command line - Gives the default command line used to compile the DLM. If “Use Default” is unchecked, it is possible to change the command line that is given to the command line compiler.

DLP command line - Gives the default command line used to compile the DLP. If “Use Default” is unchecked, it is possible to change the command line that is given to the command line compiler.

Loader command line - Gives the default command line used to compile the Loader. If “Use Default” is unchecked, it is possible to change the command line that is given to the command line compiler.

RFU command line - Gives the default command line used to invoke the RFU. If “Use Default” is unchecked, it is possible to change the command line that is given to the command line RFU.

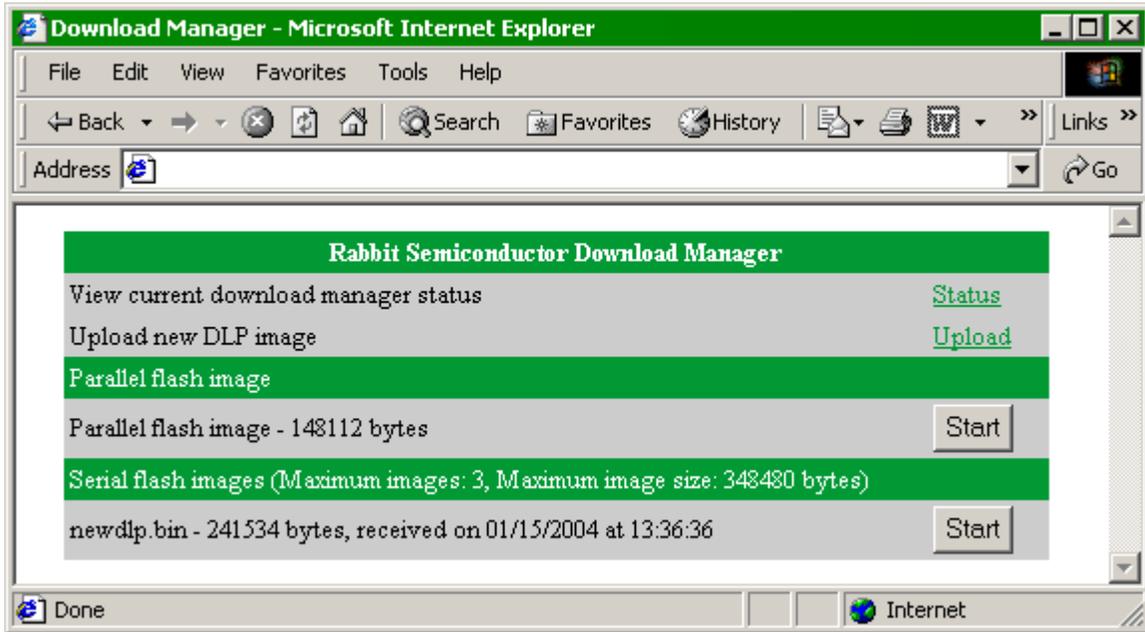
## 7. DLM/DLP Interfaces

This section describes the remote user interface.

### 7.1 DLM Home Page

The main page of the DLM allows the remote user to check the status of the download manager, upload a new DLP image, start the parallel flash resident flash image, or any DLP image stored in serial flash.

**Figure 5. DLM Home Page**

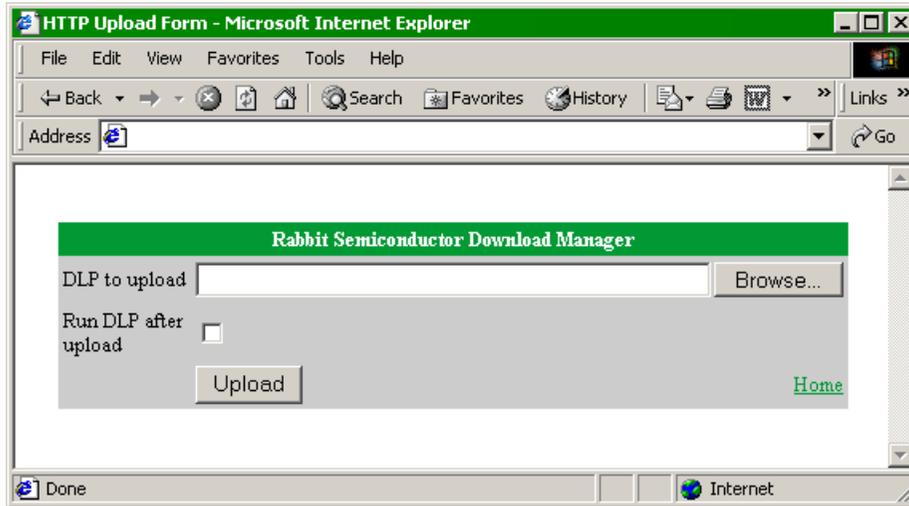


In the above screen shot, the header line above the serial flash image indicates that the download manager is configured to store up to 3 DLP images in its serial flash, each of which can be up to 348480 bytes in size. The download manager can be directed to start any of the available DLP images by clicking on the Start button associated with the desired image.

## 7.2 DLM Upload Page

Clicking on the Upload link from the DLM home page makes the following form appear.

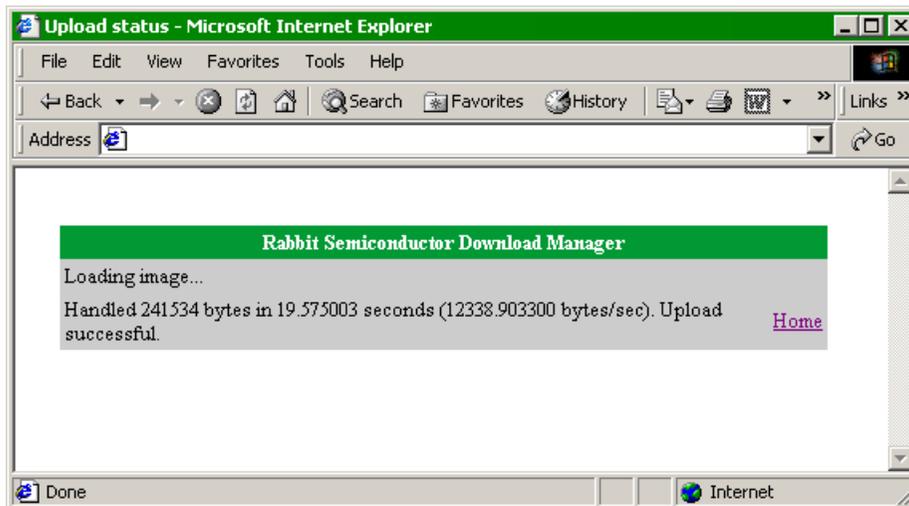
**Figure 6. DLP Upload Form**



This form allows the remote user to select a DLP image (built using the Download Image Generator Utility) from their local computer, and upload it to the download manager. If the “Run DLP after upload” checkbox is checked, the download manager will automatically start the DLP if the upload is successful.

## 7.3 DLM Upload Status Page

Once an image is uploaded, the download manager displays the upload status page.

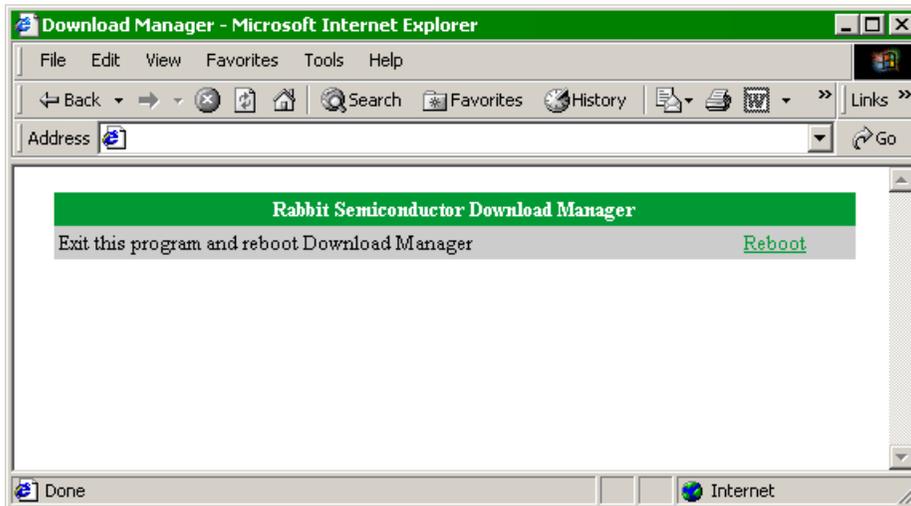


If the upload was successful, the number of bytes received and the elapsed time is shown. If the upload fails, the reason for failure is shown.

## 7.4 DLP Interface

The DLP can use a web interface, a custom interface, or both. These are controlled by configuration macros listed in [Section 3.2](#).

This page is displayed by browsing to [Remote Device IP Address]/reboot.html. Clicking on the Reboot link will cause the DLP to shut down and invokes the parallel flash resident loader to load and execute the DLM.



This next page is displayed after clicking on the Reboot link to let the user know that the DLM will be starting up soon. This page includes a refresh tag and a redirect tag so that the browser can be redirected to the DLM home page after a configureable amount of seconds (see [DLP\\_REBOOT\\_CGI\\_REDIRECTTO\(s\)](#) and [DLP\\_REBOOT\\_CGI\\_REFRESH\\_SEC](#)).

