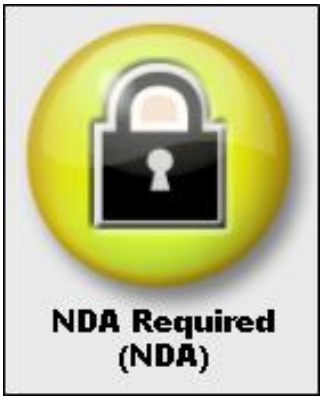
AMI RoboVeB User Manual

AMI RoboVeB User Manual

Document Revision 0.8

Document Owner: Utility Group

May 04, 2016



American Megatrends, Inc.

5555 Oakbrook Parkway

Suite 200

Norcross, GA 30093 (USA)

Legal

Disclaimer

This publication contains proprietary information which is protected by copyright. No part of this publication may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language, or transmitted in any form whatsoever without the prior written consent of the publisher, American Megatrends, Inc. American Megatrends, Inc. retains the right to update, change, modify this publication at any time, without notice.

For Additional Information

Call American Megatrends, Inc. at 1-800-828-9264 for additional information.

Limitations of Liability

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

Limited Warranty

No warranties are made, either expressed or implied, with regard to the contents of this work, its merchantability, or fitness for a particular use. American Megatrends assumes no responsibility for errors and omissions or for the uses made of the material contained herein or reader decisions based on such use.

Trademark and Copyright Acknowledgments

American Megatrends, Inc.

5555 Oakbrook Parkway

Suite 200

Norcross, GA 30093 (USA)

All product names used in this publication are for identification purposes only and are trademarks of their respective companies.

Table of Contents

Document information

Purpose

This document provides the information about the features and use of the product RoboVeB

Audience

The intended audiences are BIOS developers, Generic Chipset Porting Engineers, OEM Porting Engineers, and AMI OEM Customers. An Introduction to RoboVeB

What is RoboVeB?

RoboVeB is a command line tool used to download a project from source control and build the project. The main functionality of this tool would be:

- Download project from source control
- Build the project (either downloaded or residing in a local location)
- Add a component to an existing project from source control or local drive
- Update a component of an existing project from source control or local drive
- Delete a component of an existing project from local drive
- Evaluating component dependency online/offline

Usage

The command line provides the following options to the user:

Roboveb.exe [options]

Command line parameters	Description
-v <.veb file name>	A project with specified name
-d [<dst folder>]	Downloads the project in the provided destination folder
-u <db>:<usr>=<pswrd>	Database, username and password to access source control database
-b [all]	Builds a project
-s	Disables download action
-i	Ignores errors due to missing CIF/unresolved references
-a <AliasName>=<URL>	Database name with database URL
-e	Enables the check for missing Files
-Add <Local/full source control path of component>	Adds a component to a local project.
-Upd <Local/full source control path of component>	Updates a component of a local project.
-Dup <Source path of the component to	Downloads the specified component and skips

download>	the other duplicate component during add/update
-Del <Ref name of a component>	Deletes a component from the project.
-log <log File name>	Adds the download / add component/ delete component / update component logs into the file specified.

Usage Options

Specifying Project file

The -v option is used to specify the project file that is to be loaded and built. It can be path to the .VeB file in source control for downloading. In case the project is in source control, additional parameters such as alias URL, user name and password to access the database are also required. Along with the project file path, the label can also be specified if the user requires the specific labeled version of the project to be downloaded. The project file path along with the label have to be specified in the following format:

```
-v SS:< Alias_name>;< source control path for .veb file>;< label_name>
```

If the project is available locally then the following syntax may be adopted to specify project file,

```
-v <name/path of the .veb file>
```

If operations like Add/Update component and Build were attempted to a project existing locally then name of the .veb file (followed by the switch '–v') alone is sufficient for the intended operation to take place. If download project is attempted on a local .veb file then in this case the whole path of the .veb file needs to be supplied (Downloading local .veb file is discussed briefly in the forthcoming sections).


Note: Semicolon (;) is optional for local operations and it is not required (or has no impact) when provided after the .veb file

Specifying the database info

RoboVeB has the feature of supporting multiple Source Controls similar to VeB. RoboVeB is compatible with multiple Source Control interfaces such as:

- SS: Microsoft Visual Source Safe
- CC: Rational ClearCase
- PVCS: Serena (formerly Merant)
- RSS: AMI Remote Source Control (RSC)
- PCMS: Serena (formerly Merant) Dimensions
- AMISVN

Specifying Alias

If user runs RoboVeB without creating Aliases using VeB, RoboVeB will not be able to retrieve the source control path associated with the specified alias. To avoid this failure, the user must ensure that the aliases are registered using VeB or through command line. User can register the aliases through RoboVeB using -a option. They need to provide the alias name and the full URL of the Database separated by "=". Only then, RoboVeB will retrieve the source control path associated with the specified alias and proceed with downloading the project. Otherwise, RoboVeB would throw an error stating that the source control path is unregistered in the Alias database.


```
-a <Alias>=<Database>:<Protocol>:<URL>;<Alias>=<Database>:<Protocol>:<URL>
```

The user can specify multiple -a options with multiple alias details separated by ";".

Specifying User Name

The -u option specifies the database information required while downloading the project. The user can specify the Alias name followed by ":" and then the user name and password (separated by "=") required to access the database configured for this alias. If the user name and password are unspecified, then RoboVeB will attempt to authenticate the user based on the current user context in which RoboVeB is running. This action is carried out only when downloading project from Server. The user can specify the Alias name followed by the user name and password required to access the database configured for this alias. The user can specify multiple -u options with multiple Alias login information separated by ";". This will be useful, when downloading a project that contains library components residing in another source control database.

```
-u <Alias>:<User Name>=<Password>;<Alias>:<User Name>=<password>
```

Other options

-b [all]
This parameter specifies that the project is needs to be built. "[all]" is an optional parameter. If this parameter is not specified during build, the BIOS image will be generated by compiling and linking only files that were modified since the last complete build.
If this parameter is specified, then the entire project is built completely by compiling and linking all the files in the project source tree. This parameter can be used with the –d option to download and build the project in a single command.

-s
This is optional parameter that can be used to skip downloading a project from server. This will help to build a project from any local drive without downloading it.

-d [<dst folder>]
This is an optional parameter, to specify the destination location, where the project is to be downloaded. If this is not specified and if the project has to be downloaded from source control location then the project will be downloaded in the current location from which RoboVeB is running.
If the destination folder already exists, then the existing contents will not be cleared and the project will be downloaded to the same location.

-i
This parameter will allow the download operation to proceed even if some of the components in the project are missing or inaccessible. It will ignore these errors that occur during the download operation due to missing or inaccessible CIF files. When the download is carried out without this option, the download will halt when it encounters a component that cannot be downloaded.

-e
This parameter will show the error message and will not proceed with the download if the files are missing during the download operation. If this option is not mentioned then the missing file error message will not show after download and it will proceed with the build operation.

-Add [<component location>]
This option can be used to add a new component to a local project. The component can be added from source control or from local drive. Here, in case of adding component getting from source control, <component location> is the full source control path of the component i.e., like SS:<alias>;<source control URL>;<Label>. <Label> is an optional field. In case, if we add the component from local drive, we need to provide the full path of the CIF location.
-Upd [<component location>]
This option can be used to update a component to a local project, provided that component should exists in the project and the updation label and the current label of the component should not be same. The component can be updated from source control or from local drive. Here, in case of updating component from source control, <component location> is the full source control path of the component i.e., like SS:<alias>;<source control URL>;<Label>. <Label> is an optional field. In case of updating from local drive, we need to provide the full path of the CIF location.

-Dup [<component SS path>]
This option can be used to download the component mentioned in the command line and skip duplicate component during add/update operation
-Del [<component Ref name>]
This option can be used to delete a component from a project, provided that component should exists in the project.

-log [<Log File Name>]
This option can be used to save the RoboVeB console log messages to a file under project location. This options saves the download, Add, Update operation logs along with the build log, if BuildLog property of the veb file is not set. If only a file name is specified as log File name, file will be created under project directory.

-consoleLog
This option can be used to print the log messages to the console.

-noExit
This option prevents RoboVeB from exiting automatically.

Sample Usage

Downloading from source control

```
RoboVeB.exe -v SS:AptioV;$/AptioV/Projects/Intel/IeiSugarBay/IeiSugarBay
.veb;(INT)5.008_IeiSugarBay_12 -d D:\Sugarbay -e -consoleLog
```

This will download the specified project from the source control location and will save it in the local directory location. If any missing file is found it will be logged in the console.

Note: If 'No label' was specified then it will try to download the latest version of the project. Also for retrieving latest version, semicolon is optional and it is not required (or has no impact) when provided after the .veb file. The following command has the same effect irrespective of semicolon (for retrieving a project's latest version from Source safe).

```
RoboVeB.exe -v SS:AptioV;$/AptioV/Projects/Intel/IeiSugarBay/IeiSugarBay .veb{with or
without semicolon} -d D:\ Sugarbay -e -consoleLog
```

Downloading from local veb file

```
RoboVeB.exe -v E:\brikland\ThunderRidge.veb;5.008_Brickland_0ACBF020 -d D:\Sugarbay -
e -consoleLog
```

This will get project veb file from the specified location and the components will be downloaded from the source control location in the veb file. Private components may require the appropriate label information in addition to the available source control path. This optional label information can be provided to RoboVeB as part of the command line option. If the label is specified, then the private component of the specified label will be downloaded. If no label is specified then it will download the latest version from the source control. If any missing file is found it will be logged in the console.

Note: The user needs to provide ';' after the local file path followed by an optional label to download the project using a local veb file as shown below,

```
RoboVeB.exe -v E:\brikland\ThunderRidge.veb;5.008_Brickland_0ACBF020 -d D:\ Sugarbay
-e -consoleLog
```

If there is no label, VeB expects atleast a delimiter (;) after the local file path as shown below,

```
RoboVeB.exe -v E:\ThunderRidge.veb; -d D:\Sugarbay -e -consoleLog –noExit
```

If the ';' is not provided then the download operation will not happen.

Specifying Aliases for Downloading

```
RoboVeB.exe -a
AptioV=AMISVN:https://svnbios.ami.com:443/svn/AptioV;Porting_SVN=AMISVN:https://svnbios.ami.com:443/svn/Porting_SVN -u AptioV:UserName=Password -v
SS:AptioV;$/AptioV/Projects/Intel/IeiSugarBay/IeiSugarBay.veb;(INT)5.008_IeiSugarBay_12 -d D:\Sugarbay -consoleLog
```

This will register the aliases and download the specified project from the source control location using the credentials provided.

Building a project from local location

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -i -b all -s -consoleLog
```

This will load the project from the mentioned location and builds the project.

The –b option can be used along with all option to Rebuild the project.

The –s option to skip download and build project from local drive This option can be ignored and full source control path for the project need to be provided to download the project before building it using a single command.


Note: Semicolon (;) is optional for local operations (Build) and it is not required (or has no impact) when provided after the .veb file

Downloading and Building a Project

```
RoboVeB.exe -a
AptioV=AMISVN:https://svnbios.ami.com:443/svn/AptioV;Porting_SVN=AMISVN:https://svnbios.ami.com:443/svn/Porting_SVN -u AptioV:UserName=Password -v
SS:AptioV;$/AptioV/Projects/Intel/IeiSugarBay/IeiSugarBay.veb;(INT)5.008_IeiSugarBay_12 -d D:\Sugarbay -i -b -log log.txt -consoleLog
```

This will register the aliases and download the specified project from the source control location using the credentials provided and then this will load the project from the local location and builds the project.

The –log option will save the download log in the file log.txt, created under project location. If veb file is not having BuildLog property set, the build log also will get appended to this file.

Adding a component to a local project from Source Control

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -s -consoleLog -Add
SS:AptioV;$/AptioV/Source/Interfaces/AmiCompatibilityPkg;5.004_AmiCompatibilityPkg_14
```

This will load the project from the mentioned location and add the AmiCompatibilityPkg of 5.004_AmiCompabilityPkg_14 label to the IeiSugarBay project.

AmiCompatibilityPkg should not be present in that project, before adding it.

Adding a duplicate component to a local project from Source Control

```
RoboVeB.exe -a AptioV=AMISVN:https://svnservertest:8443/svn/AptioV -u
AptioV:username=password -v IeiSugarBay.veb -Add
SS:$/AptioV;AptioV/Source/Interfaces/AmiModulePkg;AmiModulePkg_27.1 -s -d
D:\Projects\5.11_IeiSugarBay_17_INT -Dup
$/AptioV/Source/Interfaces/AmiModulePkg/ReportFV/AmiModulePkg/ReportFV/ReportFV.cif;
$/AptioV/Source/Interfaces/AmiModulePkg/Library -consoleLog -noExit –i
```

This will load the project from the mentioned location and add the AmiModulePkg of AmiModulePkg_27.1 label to the IeiSugarBay project.

If duplicate components are present in the selected cif, the user can provide the SS path of the c component to download along with the –Dup switch as mentioned in the above command. The user can mention more than one component to download by providing the SS path with a ";" and a space.

The SS path can be upto the folder location or upto cif location.

AmiModulePkg should not be present in that project, before adding it.

Adding a component to a local project from local drive

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -s -consoleLog -Add
D:\RoboVeBProduct\Sugarbay_label12
```

This will load the project from the mentioned location and add the AmiCompatibilityPkg of 5.004_AmiCompabilityPkg_14 label to the IeiSugarBay project.

AmiCompatibilityPkg should not be present in that project, before adding it.

Updating a component to a local project from Source Control

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -s -consoleLog -Upd
SS:AptioV;$/AptioV/Source/Interfaces/AmiCompatibilityPkg;5.004_AmiCompatibilityPkg_14
```

This will load the project from the mentioned location and will update the AmiCompatibilityPkg of IeiSugarBay project to label 5.004_AmiCompabilityPkg_14.

AmiCompatibilityPkg should be present in that project and of some other label other than 5.004_AmiCompabilityPkg_14.

Updating a duplicate component to a local project from Source Control

```
RoboVeB.exe -a AptioV=AMISVN:https://svnservertest:8443/svn/AptioV -u
AptioV:username=password -v IeiSugarBay.veb -Upd
SS:$/AptioV;AptioV/Source/Interfaces/AmiModulePkg;AmiModulePkg_27.1 -s -d
D:\Projects\5.11_IeiSugarBay_17_INT -Dup
$/AptioV/Source/Interfaces/AmiModulePkg/ReportFV/AmiModulePkg/ReportFV/ReportFV.cif;
$/AptioV/Source/Interfaces/AmiModulePkg/Library -consoleLog -noExit –i
```

This will load the project from the mentioned location and update the AmiModulePkg of AmiModulePkg_27.1 label to the IeiSugarBay project.

If duplicate components are present in the selected cif, the user can provide the SS path of the component to download along with the –Dup switch as mentioned in the above command. The user can mention more than one component to update by providing the SS path with a ";" and a space.

The SS path can be upto the folder location or upto cif location.

AmiModulePkg should be present in that project and of some other label other than AmiModulePkg_27.1.

Updating a component to a local project from local drive

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -s -consoleLog -Upd
D:\RoboVeBProduct\Sugarbay_label12
```

This will load the project from the mentioned location and will update the AmiCompatibilityPkg of IeiSugarBay project with this component of another project present at "D:\Sugarbay_label12"

AmiCompatibilityPkg should be present in that project, where we want to update it.

Deleting a component to a project

```
RoboVeB.exe -v IeiSugarBay.veb -d D:\Sugarbay -s -consoleLog -Del AmicompabilityPkg
```

This will load the project from the mentioned location and will delete the component AmiCompatibilityPkg from IeiSugarBay project.

AmiCompatibilityPkg should be present in that project, from where we want to delete it.

Note: To support RoboVeB for Linux Environment, the inputs should be enclosed in double quotes as illustrated below:

./RoBoVeB -a "AptioV=AMISVN:https://svnbios.ami.com:443/svn/AptioV;" -u

```
AptioV:username=pwd -v
"SS:AptioV;$/AptioV/Projects/Apm/Apm88xx/Mustang_0ACDY/Storm.veb;5.010_Mustang_
0ACDY005" -d "/home/ubuntu64/Desktop/VeB/Projects/robovebdwnld" -log log.txt
```

Component Dependency Evaluation

Component Dependency evaluation based on command line.

Add component along with its dependency

```
RoboVeB.exe -v TunnelMountain.veb -d D:\TunnelMountain -s -consoleLog -Add
SS:AptioV;$/AptioV/Source/Interfaces/AmiCompatibilityPkg;5.004_AmiCompatibilityPkg_14 -
evalDep
```

This will load the project from the mentioned location and add the AmiCompatibilityPkg of 5.004_AmiCompabilityPkg_14 label to the TunnelMountain project and evaluate the component dependency for whole project. AmiCompatibilityPkg should not be present in that project, before adding it.

Update component along with its dependency

```
RoboVeB.exe -v TunnelMountain.veb -d D:\TunnelMountain -s -consoleLog -Upd
SS:AptioV;$/AptioV/Source/Interfaces/AmiCompatibilityPkg;5.004_AmiCompatibilityPkg_14 -
evalDep
```

This will load the project from the mentioned location and will update the AmiCompatibilityPkg of IeiSugarBay project to label 5.004_AmiCompabilityPkg_14. AmiCompatibilityPkg should be present in that project and of some other label other than 5.004_AmiCompabilityPkg_14. Then evaluate the component dependency for whole project.

Evaluate Dependency Online

```
RoboVeB.exe -v TunnelMountain.veb -d D:\TunnelMountain -s -consoleLog
-evalDep
```

This will load the project from the mentioned location and evaluate the component dependency for whole project.

Evaluate Dependency Offline

```
RoboVeB.exe -v TunnelMountain.veb -d D:\TunnelMountain -s -consoleLog
-evalDepOff
```

This will load the project from the mentioned location and evaluate the component dependency for whole project in offline mode.

1. Appendix A: Building Project from Command line

It is possible to build the BIOS projects from command line also without using RoboVeB or VeB. In order to build the project, the following steps are to be performed for setting up the environment before starting the build:

1. The locations of the tools required for building the project are to be set in the path.
    a. Location of Buildtools (Example: `C:\BuildTools`)
    b. Location of x86 compiler from DDK (Example: `C:\WINDDK\bin\x86`)
    c. Location of x64 compiler from DDK (Example: `C:\WINDDK\bin\win64\x86\amd64`)
2. The environment variables required for building the project are to be set:
    a. Set path=C:\BuildTools;C:\WINDDK\bin\x86;C:\WINDDK\bin\win64\x86\amd64; C:\MASM615\BIN;%path%;
    b. CCX86DIR = Location of x86 compiler from DDK.
        - (Example: C:\WINDDK\3790.1830\bin\x86)
    c. CCX64DIR = Location of x64 compiler from DDK.
        - (Example: C:\WINDDK\3790.1830\bin\win64\x86\amd64)
3. In addition to these environment variables, the following two environment variables may be required, if they are used during the build process. (Note: VeB will be setting these environment variables automatically while launching the build)
    a. VEB_BUILD_DIR= Location of the project build directory
        - (Example: VEB_BUILD_DIR=Build)
    b. VEB=Name of the project file (.veb)
        - (Example: VEB=Greencity)