# embarcadero®

Product Documentation

# Embarcadero® AppWave™ Studio
## User Guide

Version 4.1

Published February, 2013

Embarcadero Technologies, Inc. is a leading provider of award-winning tools for application developers and database professionals so they can design systems right, build them faster and run them better, regardless of their platform or programming language. Ninety of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero products to increase productivity, reduce costs, simplify change management and compliance and accelerate innovation. The company's flagship tools include: Embarcadero® Change Manager™, RAD Studio, DBArtisan®, Delphi®, ER/Studio®, JBuilder® and Rapid SQL®. Founded in 1993, Embarcadero is headquartered in San Francisco, with offices located around the world. Embarcadero is online at www.embarcadero.com.

February,  2013

# Contents

# WELCOME TO APPWAVE STUDIO

The AppWave Studio is an application within the AppWave Browser where you can master an application for AppWave as an App.

> **NOTE:** For the latest versions of AppWave documents and for information that will help you install and upgrade, see the Embarcadero documentation website. Always make sure to consult the README for late breaking updates at http://docs.embarcadero.com/products/appwave.

Apps are the applications that are mastered into a single file that run without installation. The app runs on the desktop, locally, and under 'full power' but does not change the operating system or registry. Apps run within their own space, with no shared DLLs or other system-wide settings that could conflict with other Apps. Application data, including settings and licenses are stored locally on the hard drive separate from the executable.

Once your app is broadcast, an app build is placed in a working folder. This build is a copy of the normal install build which is compressed into one single standalone executable file that runs instantly on any desktop. This process takes the entire structure of an installed app with all the settings (registry and default settings of the application) and combines them into the final standalone build. This "build" can be executed from a network without being installed on that computer.

The advantage of using the app technology is that the app is broadcast on the system and when it is removed, the system is in a state as if it was never installed. Any file or registry keys that are created by the creation of the app are deleted from the file system on removal. If the app modifies a file on the system, the original state is restored.

> **NOTE:** For additional information and support on Embarcadero's products, please refer to the Embarcadero web site at http://www.embarcadero.com.

This chapter covers the following topics:

- AppWave Studio Users

- Mastering Apps

- Benefits of Using Apps

- Enterprise AppWave vs. Public AppWave

## APPWAVE STUDIO USERS

AppWave Studio has two different types of users: Administrators and Users:

**Administrator**: If you have administrative privileges you have access to all the AppWave functionality. When you first log onto the AppWave Browser an Admin tab appears in the header.
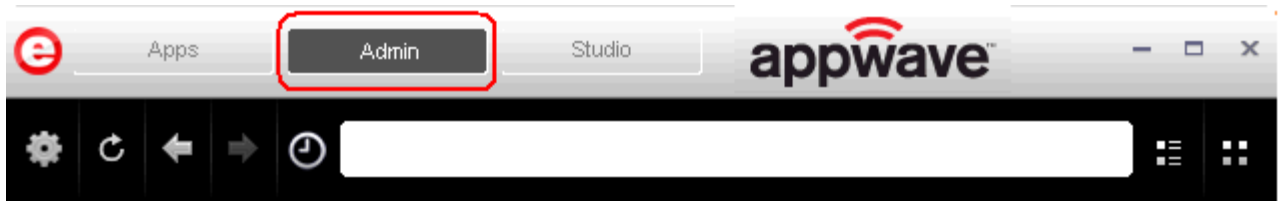
The Administrator:

- Has access to all the administrative functions in AppWave. The Administrator can access the functionality through a web browser or through the Admin tab in the AppWave Browser.

- Has the option to give users access to the AppWave Studio so they can master apps. For more information, please refer to the *Group Management* topic in the *AppWave Administrator Guide.*

- Can collect and analyze product usage on a per user, group, and product basis.

- Can broadcast multiple product configurations and assign them to individuals and groups.

- Can retrieve updates from the public server or other privately deployed servers.

- Manages license inventory, assignments, and connected users. Also ensures the correct information is available for License Automation. For more information, please refer to the **Manage Licenses** topic in the *AppWave Administrator Guide.*

**Users**: When a user opens the AppWave Browser they have access to the App tab where they can view and run apps locally on their own desktop without installation.



The user:

- Is added to the **All Users** group.

- Has access to supported apps as an easy-to-use alternative to installing and configuring apps. These users can create an app from a template in the AppWave Browser. Once the app is mastered, it can then be streamed from the AppWave and then run locally on the desktop without installation.

- If granted authoring permission from the Administrator, has access to the AppWave Studio where they can create Apps two different ways: (1) from an executable that automatically captures file and registry changes; or (2) by building an app manually where they insert required information.

# MASTERING APPS

An app is an application that has been mastered with AppWave and then made available in the AppWave Browser.

> **NOTE:** Some applications are not suitable for use with AppWave Studio and AppWave, for example an application that must run as a Windows service. Since by definition they start automatically as a service when you boot Windows, they are not able to run on a different machine.
>
> Also, an application is not suitable if it is configured to run against a database or other service that is installed on the same machine, for example, an application that locally installs a database and the database connection is hard coded. These applications may have the option to be installed as a network client and can be configured to access a database on the network; that configuration is more likely to work as a Studio App.

> **CAUTION:** During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio will attempt to shutdown any services it detects. But in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the App.

There are several different approaches to creating an app depending on the source of the executable/installer used:

- **Create an App with One Click.** All the mastering support has been done by Embarcadero. This process uses a template that has been created with the information needed to complete the App. This information includes descriptions, display name, product name, etc. All you need to do is supply an application's executable. This creation process starts in the Apps portion of the AppWave Browser. You then complete the creation process in the Studio portion.

  **User**: An Enterprise Desktop Administrator - Only basic programming skills are necessary. The process is streamlined with many of the necessary decisions made for the user.

- **Create AppWave Supported or Custom App from Install.** With this option, you create, edit, and broadcast an app from Install using Embarcadero® AppWave™ Studio. The steps to be completed depend on the application you have chosen to install; some steps may be automatically completed for you.

  **User**: An Independent Software Vendor (ISV) user - The user mastering the application must have the knowledge to make key decisions during the creation process. This includes knowledge about the results of decisions made and when and why to use the default recommendations.

- **Create AppWave Supported or Custom App from Scratch.** An application that does not have the mastering support. This means that you would need to manually insert Product Name, Display Name, Version, Long Description, etc.

  **User**: The user needs registry knowledge. This user also needs an understanding of the installation process including files (location and properties) and environment variables.

- **Once an app has been installed,** you can add another application to the app by recording the installation of the application. For example, you may want the Browser user to have access to a database client they can install in addition to TOAD, so you can click to add one or more applications. You will need only to broadcast one file but the Browser user will see both Apps in the file as unique items in the app catalog. Also, the AppWave Admin can control license usage for each app separately.

  **User**: The user needs registry knowledge. This user also needs an understanding of the installation process including files (location and properties) and environment variables.

- **Create an App with Multiple Versions.** When you master an app using an AppWave Template, that template may contain multiple versions of the application. If that is the case, you have the option to select the version, and to either open or record that version.

  **User**: The user needs registry knowledge. This user also needs an understanding of the installation process including files (location and properties) and environment variables.

- **Create AppWave .** WebApps are applications that are accessible via an Internet Browser, typically executed on an application server either hosted by the enterprise customer directly or accessed as a "cloud" service that is hosted externally. The AppWave Studio is the ideal mastering tool for defining the parameters for the display and execution of WebApps from within AppWave Browser.

  **User:** The user needs an understanding of URLs, cloud services, and how hosted applications work.

For more information, see [Using AppWave Studio](#).

# BENEFITS OF USING APPS

- The Apps are streamed to your computer and not installed so the app doesn't take up space on your computer's hard drive.

- Any files or registry keys that are created during the mastering process, as well as any modifications made to files on the system, are deleted when the system is removed. This means that the original state is restored.

- There is no application conflict – You can determine whether applications are fully integrated with your environment, fully isolated, or any combination between the two. This prevents conflicts between applications. You can run two separate versions of the same application without conflict.

- Application data such as configuration and license information is stored on your hard disk and is separate from the application's executable.

- You can access your Apps from any computer.

- You always have access to the latest version of an app – you don't have to keep track of install patches, service packs, upgrades, etc.

For more information, see: *Understanding App Products* in the *AppWave Browser Apps User Guide*.


# ENTERPRISE APPWAVE VS. PUBLIC APPWAVE

The Enterprise AppWave (also referred to as "private"), determines what content can be received from the Public AppWave. Access to apps is controlled by an AppWave administrator or through license enforcement.

If an app has not been streamed to the Enterprise AppWave from the public it is still possible for the end user to see and access the app. If an app that has not been streamed from public is launched by the end user, the app is streamed to the Enterprise AppWave and then to the end user. An Administrator may choose to stream the app to the Enterprise AppWave to optimize the availability.

The location of the app, whether public, private, or on the local machine, concerns the end user prior to launch because it tells the user what type of delay may be expected. The status of the program loading can be defined by:

- **Unavailable** – in the case where a private server is used, this means the app has not been streamed to the server and the private server cannot currently access the public server.

- **Internet Availability** – This is when the private server does not have the app and it needs to be streamed down from the public server.

- **Remote Availability** – The private server has the app streamed.

- **Available** – The app is available on the local machine.

**Enterprise AppWave**

A key scenario for AppWave is to help companies distribute apps (including open source, third party, and internal apps) through their user base. Rather than waiting for a new OS image with the company-picked set of tools, users can run without installing any company-approved app in minutes. At the same time, the company can manage the number of licenses currently in use and make sure they comply with the vendor rules.

You can install and manage your own server. The difference is allowing the addition of custom and third-party apps with the Embarcadero Apps.

**Public AppWave**

The Public AppWave contains all of the Embarcadero-mastered apps and third-party apps that can be distributed with valid licenses.

This is a public apps-management server managed by Embarcadero. An AppWave Host Administrator determines what is made available on the public server.

The Public AppWave Host has the following major systems in its architecture:

- User Management
- License Management
- Product Catalog Management

# USING APPWAVE STUDIO

AppWave Studio allows you to create an app. Once you have created an app, it can be streamed from the AppWave and then run locally on your desktop without installation. This means that apps can be run on locked-down desktops. There are no conflicts with other apps; no writing to the system areas of their registry; and no modifying shared DLLs.

The following topics are covered:

- Studio Access

- Best Mastering Practices

- Create an App with One Click

- Create AppWave Supported or Custom App from Install

- Install Using a Zip File

- Create AppWave Supported or Custom App from Scratch

- Create an App with a Template That Contains Multiple Versions or Multiple Editions

- Add to App from Install

- Create AppWave Supported WebApps

- Add Native Installer to AppWave Catalog

- Open Project

- Open for Upgrade

## STUDIO ACCESS

When you open the AppWave Browser and only the Apps tab is visible, this indicates that you do not have permission to master apps using Studio.



To access the Studio, the Administrator must grant you permission as an Author. Once this is done, the Studio tab appears in the AppWave Browser.

For more information on how to set up this permission, see Group Management.

Once you have been granted access, click **Studio**. This is where you can master apps and then use them in your organization.

> **TIP:** If the Studio tab does not appear you may need to reconnect to the AppWave Browser
>
> by clicking ⚙ > **Connect**.



## BEST MASTERING PRACTICES

- Always master on Windows XP SP3 (32-bit) to ensure the best compatibility with newer versions of Windows.

- If possible, master in a Virtual Machine environment with a clean OS. Use the "snapshot" functionality to capture a clean state before mastering is begun. This makes it very easy to redo a mastering task or start the next mastering task from a clean state by reverting to previous snapshot.

- Always test the app thoroughly on all supported Windows versions – especially if mastering is not done on Windows XP SP3. If known incompatibilities exist, open the **Advanced Configuration Options** dialog by clicking **Advanced** in Step 4 (Test). Then uncheck the incompatible OS and click **OK**.
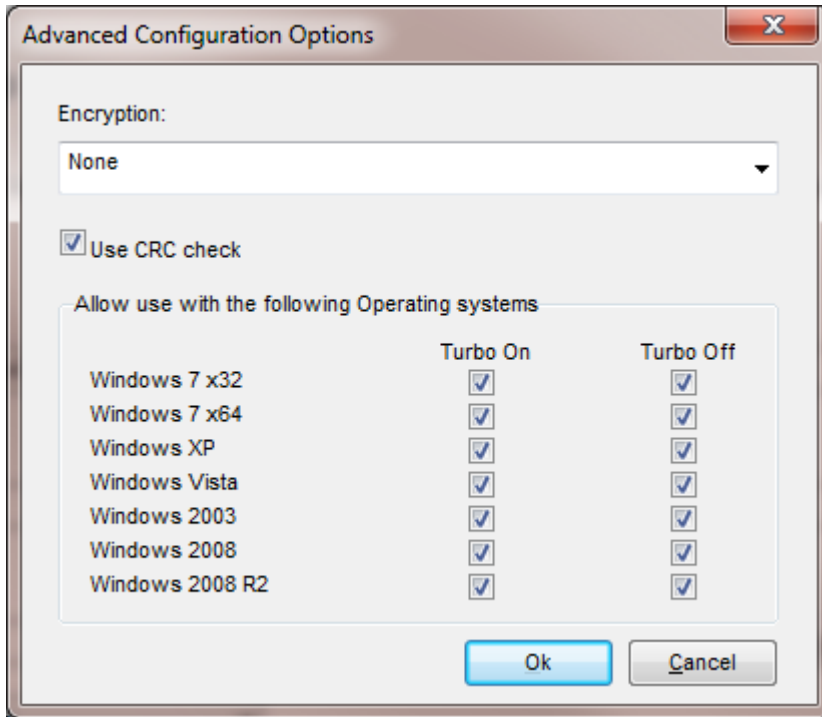


- Make sure to include any runtime dependencies in the app mastering process to guarantee that the app will run across multiple Windows versions. Or you can describe the pre-requisites under the system requirements in Step 2 - the Launch & Brand page.

- When mastering an app, the recommended location for installing the app is C:\application_name_version. Avoid installing in the default Program Files location as problems may occur when launching the app on a 64-bit system.

- Before mastering an app in the AppWave Browser, you must ensure that no other apps are currently running. This is necessary to avoid adding unwanted artifact files such as files, registry entries, and/or environment variables. These unwanted artifacts could cause recording failures.

  If there are any apps running you will get a warning dialog when you click Master. For information on how to stop all applications currently running, see Preparation.

- Folders that point to a network location such as My Documents should be avoided. It is recommended that you set the network folder to a local folder.

- During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio attempts to shut down any services it detects. But in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the app. For more information see: Stopping Services.

- When mastering an app using a template without installing, the app needs to already be on the system and installed with all the default options. If the app is installed in the Program Files folder and has this path in the configuration files for its usage or as part of other required components such as Java, there may be problems running the app on 64-bits systems.

- You cannot master Apps on a Windows 64-bit operating system.

- When mastering an app on the Windows 7 OS, create registry keys under HKEY_CLASSES_ROOT or in one of the subdirectories of HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, and HKEY_USERS.

  If the properties for the key you created are set to "All Applications" and "Permanently Copy Host Version" an error message (APM_E_APP_CONFIG_INSTALL_REGISTRY) will be generated.

- It is important to note that some applications will not work as a streamed app because of their product architecture. Two examples are listed below:

  - An applications that must run as a Windows service. Because they start automatically as a service when you start Windows, they cannot run on a different machine.

  - Applications that are configured to run against a database or other services that are installed on the same machine. An example would be an application that locally installs a database and the database connection is hard coded. If an application has the option of installing as a network client and can be configured to access a database on the network; that configuration is more likely to work as a Studio App.

# CREATE AN APP WITH ONE CLICK

You can use the AppWave Browser to create an app with one click. This process uses an app, available on the Apps tab, that has an AppWave template. AppWave templates have been created by Embarcadero for Embarcadero and third-party apps and are available with AppWave. This template tells AppWave how to master an app and contains branding information and information on which filter to run on the installer for the App, so unnecessary files can be filtered.

The apps with an AppWave template are considered "premium apps" and contain licensing information. This information allows for license automation which activates third-party licenses. AppWave then manages those licenses in the same way that Embarcadero application licenses are managed.

On the Apps tab, the Apps that can be mastered are indicated by a designation of "Master".
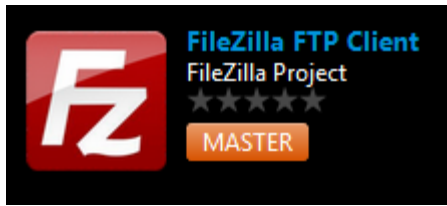
> **TIP:** When you create an app with One Click you start from the Apps tab in the AppWave Browser and then complete it in AppWave Studio.

There are four major steps:

- <u>Licensing</u>
- <u>Preparation</u>
- <u>Setup</u>
- <u>Recording</u>
- <u>Running the App</u>

## LICENSING

The Administrator sets up the licensing for third-party apps through AppWave. Once the licenses are set up and activated, when the app is mastered, the license automation completes the licensing process.

For complete step-by-step instructions for this Administrative function, see: <u>Licensing Third Party Applications</u>.

## PREPARATION

Before mastering an app in AppWave Browser, you must ensure that no other apps are currently running. If there are any apps running you will get the following warning dialog when you start the mastering process in the AppWave Studio:
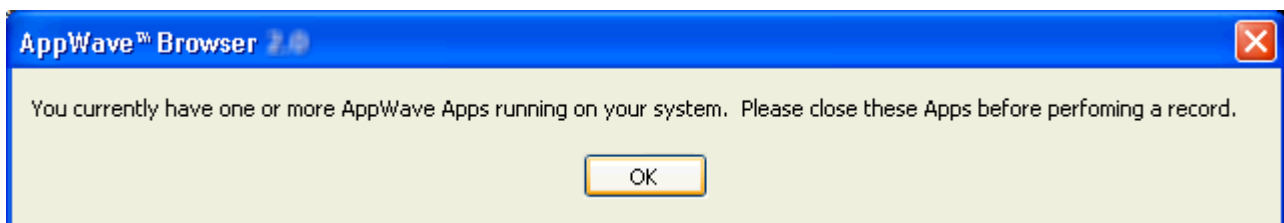


Close each app from its application interface, usually **File > Exit**.

## SETUP

To see how the One Click works, we will use the *FileZilla Client* application.

1  Download and save the FileZilla_3.5.3_win32-setup.exe from `http://filezilla-project.org`.

2  In **Apps,** search (🔍) for "AppWave Template" and all apps with templates are listed. This template is indicated by a Master icon ( MASTER ) located to the right of the app icon.



> **NOTE:**  If you are searching for a specific app or a type of App, you can use the Search field and all apps matching the search parameters appear in the field.

3  Click **View All** and scroll to the app that you want to master. Note if there are multiple pages of apps scroll to the bottom of the screen and click **Next** to advance through the pages.

4   Click **Master** located to the right of the icon.

  **NOTE:**  When you hover your mouse over the app icon, additional information appears.

5   The **Record installation** dialog opens briefly followed by the **Record <application name>** dialog.
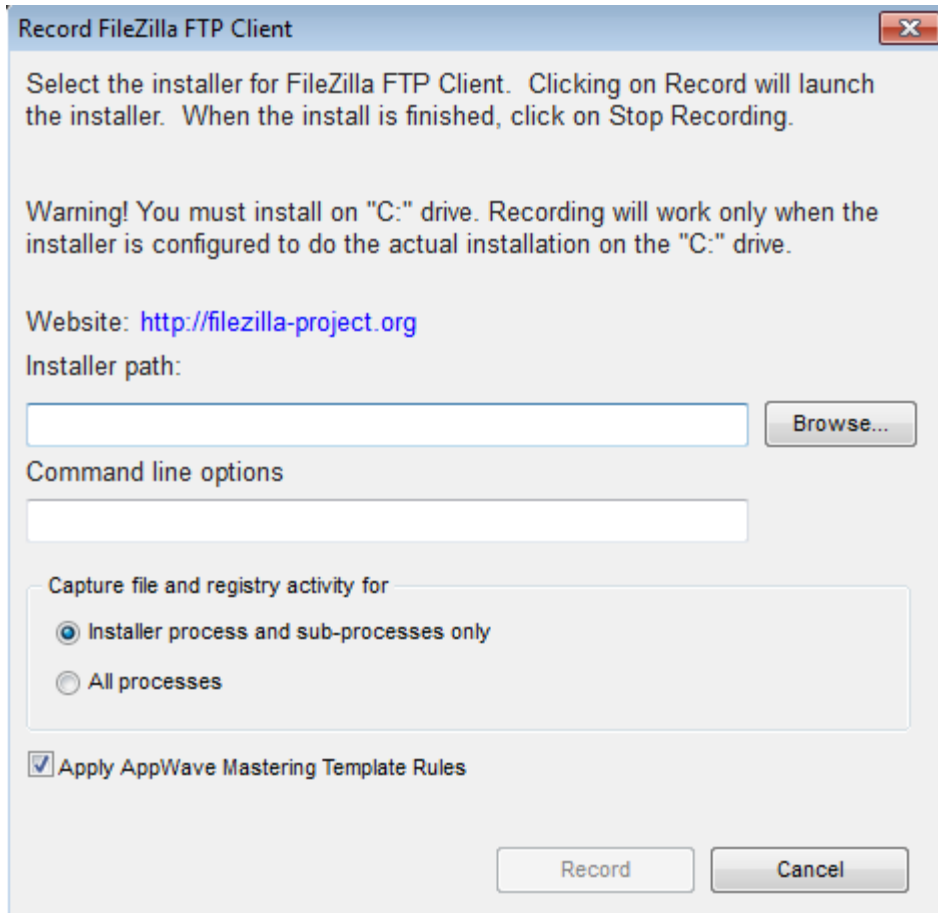


  **NOTE:**  An indicator window opens in the lower right-hand corner of your screen showing that the selected file has finished downloading. The file extension .aw indicates it is the AppWave Template file.
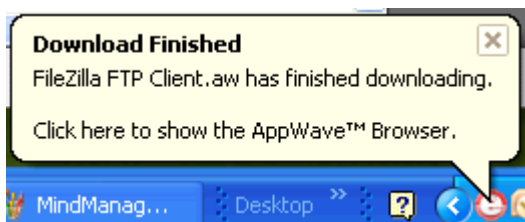
6  Browse to the location of the executable. Notice that the web site information is then automatically populated.

> NOTE:  Click the web site URL if you have not downloaded the executable for the selected app you are going to master.

This dialog has the following fields:

| Field | Description |
|---|---|
| Installer path | Click Browse to find the executable for the app you want to master. |
| Command line options | The information entered here indicates how to launch the file and is the trigger to run a specific command.<br><br>For information on commands you can use, see Using the Command Line. |
| Installer process and sub-processes only | This setting is selected by default. This option captures only the install processes. |
| All Processes | Captures all the processes associated with the App. |
| Apply AppWave Mastering Template Rules | This option is selected by default. As the process goes through the steps to create an App, tasks may be created indicating missing data. This option automatically competes any missing data that is available in the template. |

## RECORDING

1  Once you have completed the information, click **Record** and the install process starts.

2  A **Recording installation** dialog briefly appears and then a **Recording** dialog appears in the upper right corner of your screen. The blinking recording status button indicates that recording is in progress and the AppWave Browser is minimized.
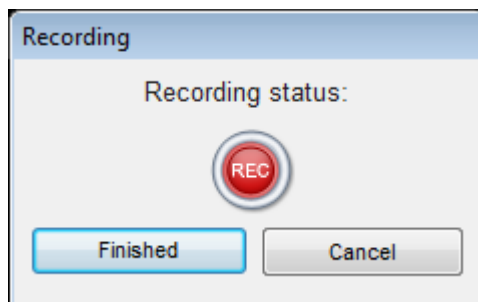
3   An indicator window opens in the lower right-hand corner of your screen showing that your installation has started.



4   Follow the install wizard for your selected app. To ensure compatibility between operating systems, choose a destination folder at the root drive, such as `C:\FileZilla FTP Client`.

5   The licensing of your app depends on what type of AppWave licensing you have purchased as well as the type of app you are creating. For a full explanation of how to license your App, please refer to: <u>Managing Licensing</u>.

   **NOTE:**   In the example we are using here, a License Agreement dialog appears. Click **I Agree** to accept the agreement.

6   On the final panel of the Setup wizard, you may have the option to launch the app. Make your selection and click **Finish** to close the wizard.

   **TIP:**   You may not want to select the Launch option so you can investigate the registry keys that are created. If the roots of these keys are not in the app it is a good idea to include them so they will be managed.

7   Once the installation is complete, click **Finished** in the Recording dialog and the **Record installation** dialog appears. This dialog identifies everything that needs to be installed. A progress bar shows each step of the process.

   **NOTE:**   Click **Details** to see the details of the installation while it is broadcasting or after it completes. If you want to see it after the installation is complete deselect the "Close on successful completion" option and then click **Details**.
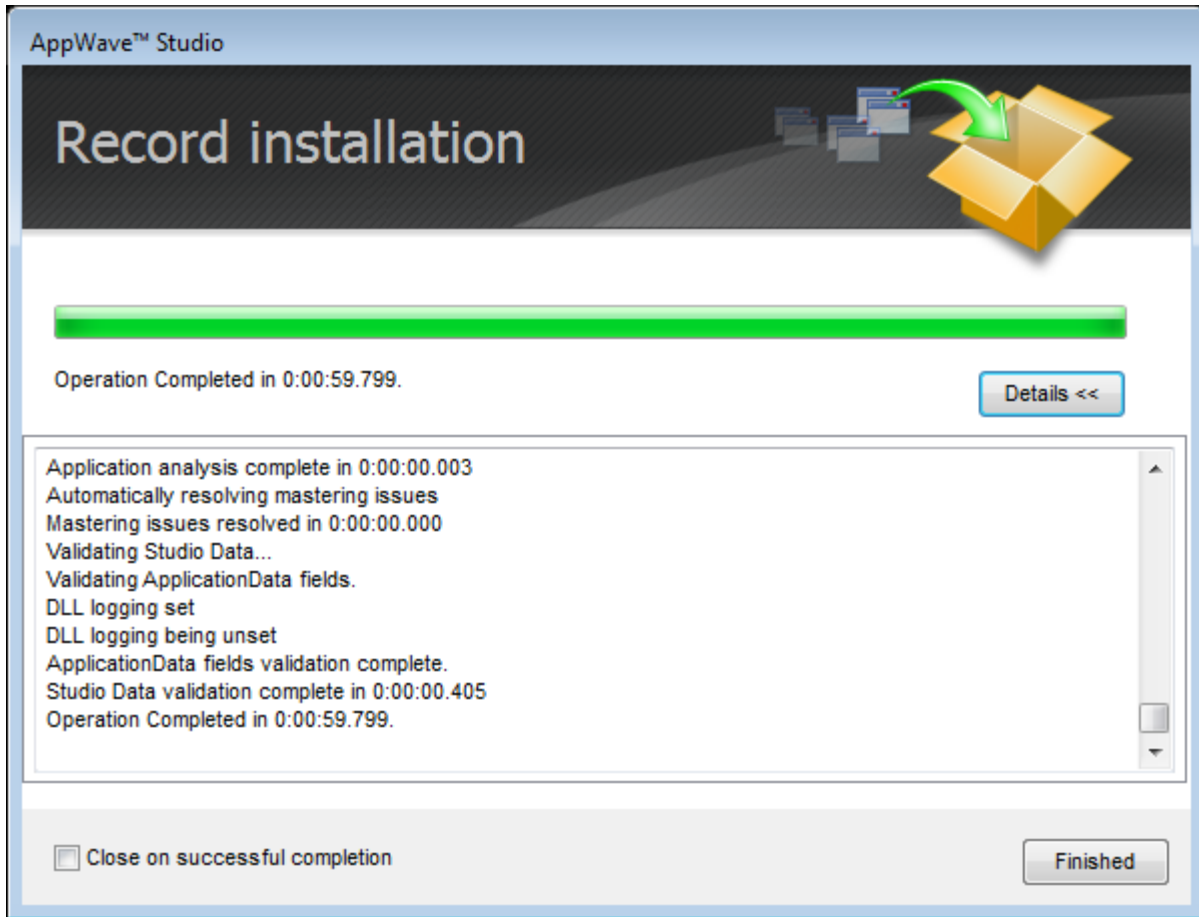
8   If this dialog box does not close automatically, click **Finished** in the **Record Installation** dialog.

> **TIP:**   At this point in the mastering process you are now working in the Studio portion of the AppWave Browser.

9   During the mastering, the AppWave template populates the various fields in the Steps (Launch & Brand, App, Test, or Broadcast). If one of the steps is missing required information, the mastering process opens to that page. If all the information is completed, the Studio opens to the Broadcast step.

> **NOTE:**   If a Tasks pane is open at the bottom of the page, this means there are fields that are missing required information. Click **Fix** next to the task and your cursor is placed in the appropriate task. Once the tasks are corrected the Tasks dialog closes. The fields with an exclamation mark ( ⚠ ) are the fields missing required information.

For the FileZilla example we are using, the Broadcast step opens. For more information on each step, click the links below:

- The <u>Launch & Brand</u> page is where you supply the name of the launch executable and branding information for the App. If this page appears you may need to supply information for the required fields such as the executable, the working folder, Publisher, Display, and Product name.

- The <u>App</u> page is where you can you can configure the app files, registry settings, environment variables and required runtimes. You can also add custom scripts to an App.

- The <u>Test</u> page is where you master and test. This is also where you prefetch the application and launch it to enable the application to start faster. You can change the output folder and edit the file name.

- The <u>Broadcast</u> page is where you broadcast your mastered app to the AppWave. This page appears when the information is complete in the previous five steps and require no action on your part before broadcasting.

## USING THE BROADCAST TASK

The last step in mastering an app is to broadcast your application to the AppWave. The following fields appear on this page.

| Field | Description |
|---|---|
| AppWave URL | The default URL appears in this box. You can edit this URL. |
| Always remaster projects before broadcasting to the AppWave | This is selected by default. This guarantees that the app has the latest updates. |
| Broadcast | Click **Broadcast** and a dialog appears telling you need to save your file before continuing. Click **OK** and the Broadcasting dialog appears. |

1    On the **Broadcast** page, notice that a Task appears at the bottom of the screen. This task indicates that the Prefetch function is recommended in the mastering process. Prefetch is the process of streaming application blocks in anticipation that those blocks are needed by an application, **as** well as to perform typical actions for the first few minutes of using the prograM, such as opening files, saving files, etc. The process of streaming these blocks from the server to AppWave Browser is also called *buffering*.

Only some key blocks of application data from AppWave are necessary for the application to launch and perform basic functions.

Unlike conventionally installed applications, application streaming does not require all application files to be always available.

The prefetch file is created by logging pages that are being streamed when the application is executed. This includes all the functions required by the application when the user opens it for the first time.

The task states:

*"In order to Record the launch of the application for faster start-up time, you need to master the app from the Test tab before broadcasting. Once the launch data was recorded, remastering before broadcast will not remove this information."*

**TIP:**    If you do not want to use the Prefetch option, you can ignore the Task to be fixed and just click **Broadcast**.

**NOTE:**    There is an indicator next to the Broadcast step showing that there is a task that needs to be fixed.



2    In the **Tasks** pane, click **Fix** and a dialog appears instructing you on how to record prefetch information.

3    Click **OK** to close the information dialog.

4    Click the **Test** page and then click [🎁 Master]. The **Master** dialog appears. If you have not saved your file a message opens. Click **Yes** and save the file named FileZilla_FTP_Client.tca.

5    The **Master** dialog opens and masters the App.

6    Once it is complete, check to make sure the "Record the launch of the application for faster start-up time" option is selected. This option is selected by default.

7   Click **Launch.** The End User License Agreement as well as the Recording dialog opens.
    When you click **Accept** the application opens.

8   Once the application opens, click **Finished** in the Recording dialog (this dialog reopens
    when you click Master) and the application closes.

9   In the **Master** dialog, click **Details** to view the results of the recording.



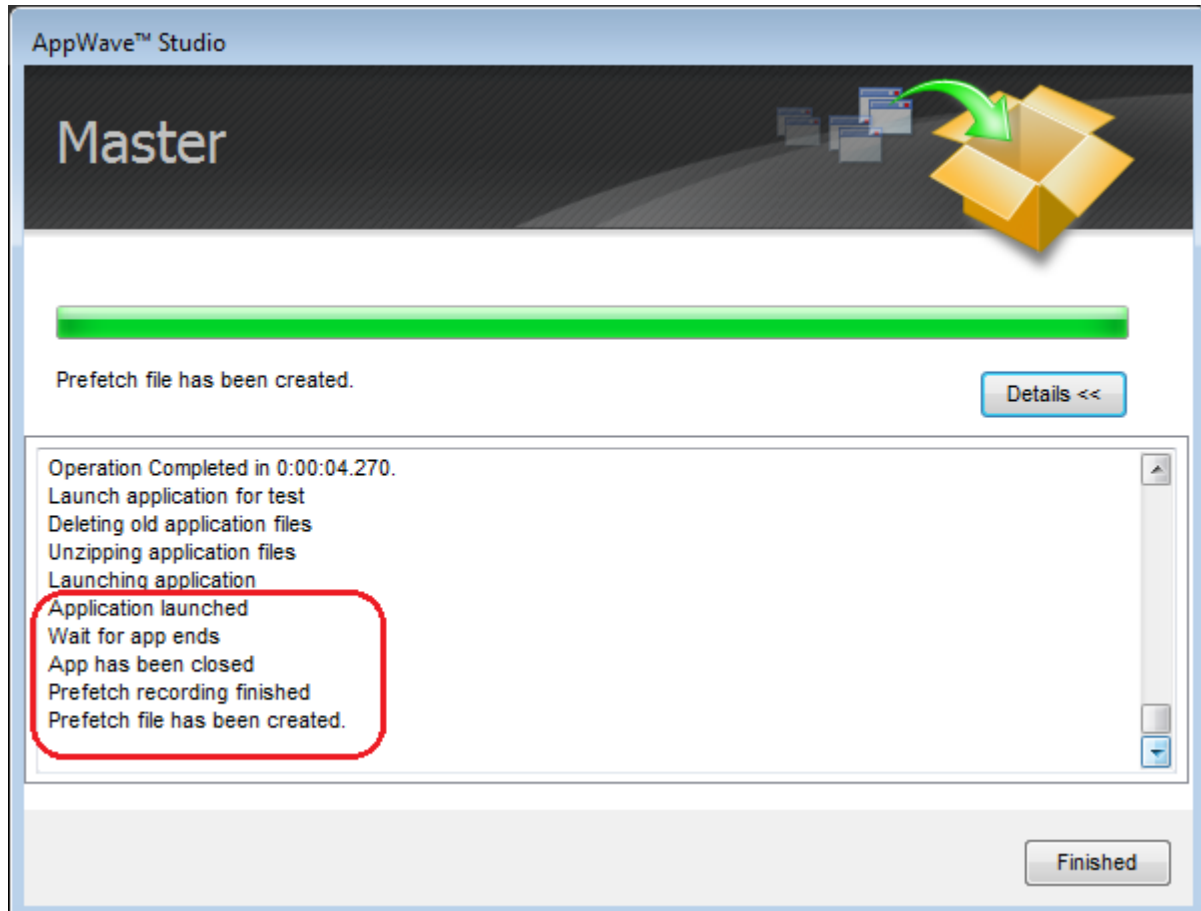10  Click **Finished** in the Master dialog to close it.

11  You must also close the application that you launched during the prefetch operation.

12  Click **Next** to advance to the **Broadcast** page.

13  Make sure the option to **Always remaster projects before broadcasting to the AppWave**
    is selected.

14  Click **Broadcast** and save the changes to your file. A dialog opens asking you to save the
    changes to your current working file. Click **Yes**.

15  Once the file is saved *the* **Broadcast** dialog appears. This dialog tracks the process of the
    broadcasting. Once you get the indication that the broadcast was successful, click **Finished**
    and the **Broadcast** dialog closes.
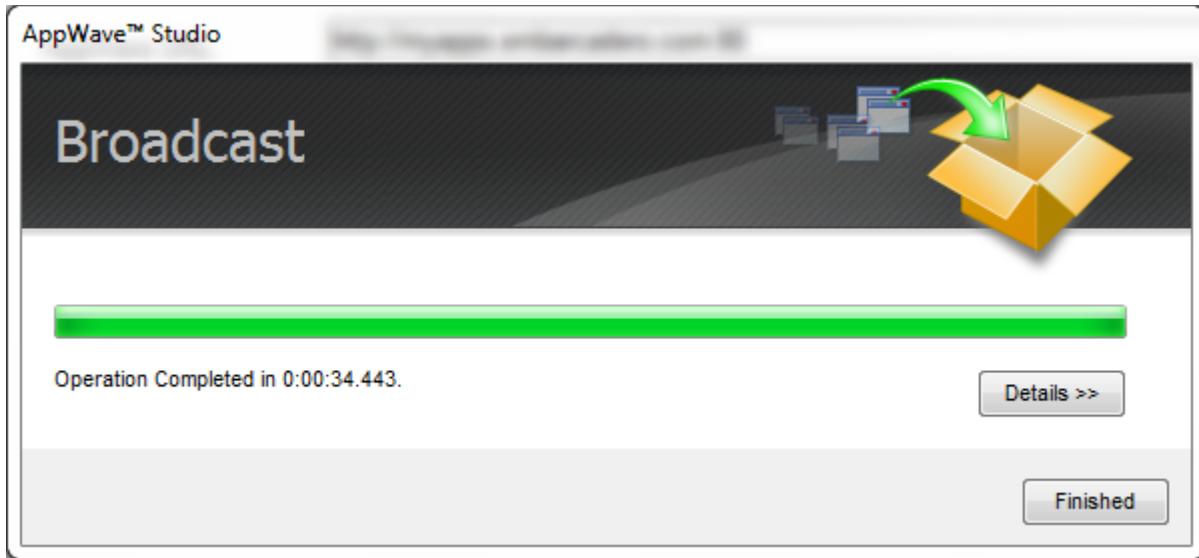
NOTE: A popup also appears in the lower right corner of your screen indicating that new and updated apps are available.
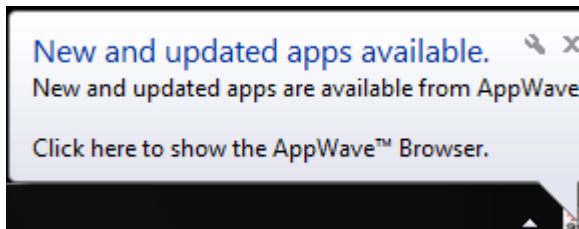


16   Click **Finished** and your app is now mastered with the prefetch information included.

## RUNNING THE APP

Once your app is broadcast, the newly mastered app build is made available in the AppWave Browser.

1   Click **Apps** to return to the AppWave Browser.

2   The following screen shot shows one app has been placed in "embarcadero.comkathyb's Folder" designated by the **1**, as well as the app that was just mastered.

> **TIP:**   If your folder does not appear you may need to reconnect to the AppWave Browser by clicking ⚙ > **Connect**.

> **NOTE:**   You can access details and reviews for this app, as well as write reviews.
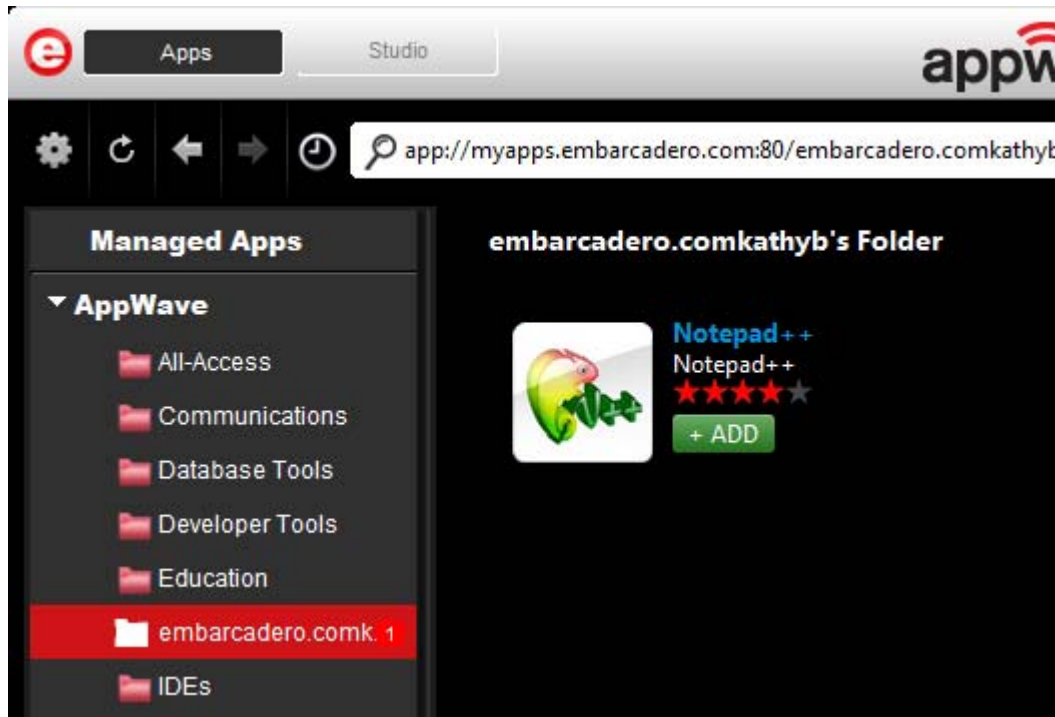


3   Select the app from your folder and click **Add**. In the **Sign In** dialog enter your user name and password from your EDN account and click **Sign in**.

4   If the licenses is available for this app the Licensing Information dialog appears which provides the key necessary to complete the final step in the Licensing Automation process.

If it is a free third-party App, an End User License Agreement dialog appears.

A Progress dialog opens in the lower left corner of your screen. This tracks the downloading, mounting, and streaming progress of your new app.
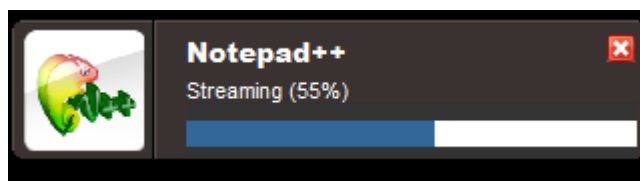


5   Click **Accept** and the app opens.

**CAUTION:** During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio will attempt to shutdown any services it detects, but in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the App. For more information see Stopping Services in the Appendix.

# CREATE APPWAVE SUPPORTED OR CUSTOM APP FROM INSTALL

With this option, you create, edit, and broadcast an app from install using AppWave Studio. This creation process creates .app and .stp files. The .stp file contains the code and other configuration information that comprises most of the app.

This option is only available through the Studio. A user must be granted authoring permission by an Administrator. For more information on this, see: Group Management.

    **NOTE:** The steps that need to be completed depend on the app you have chosen to install; some steps may be automatically completed for you.

- Setup
- Start
- Launch & Brand
- App
- Test
- Broadcast

## START

1  In the AppWave Browser click **Studio** and the **Build an App** page appears with the **Start** step highlighted in red.
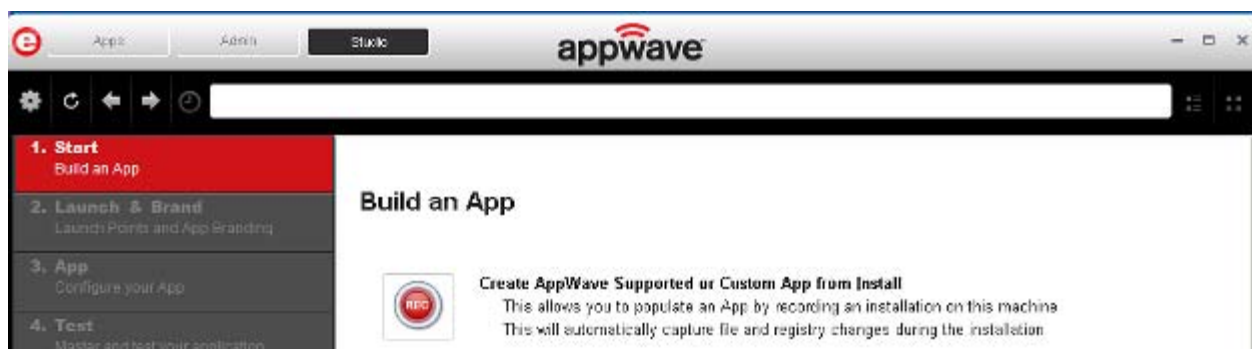
2  Click  or **Create AppWave Supported or Custom App from Install** and the **Record Application Installation** dialog appears.
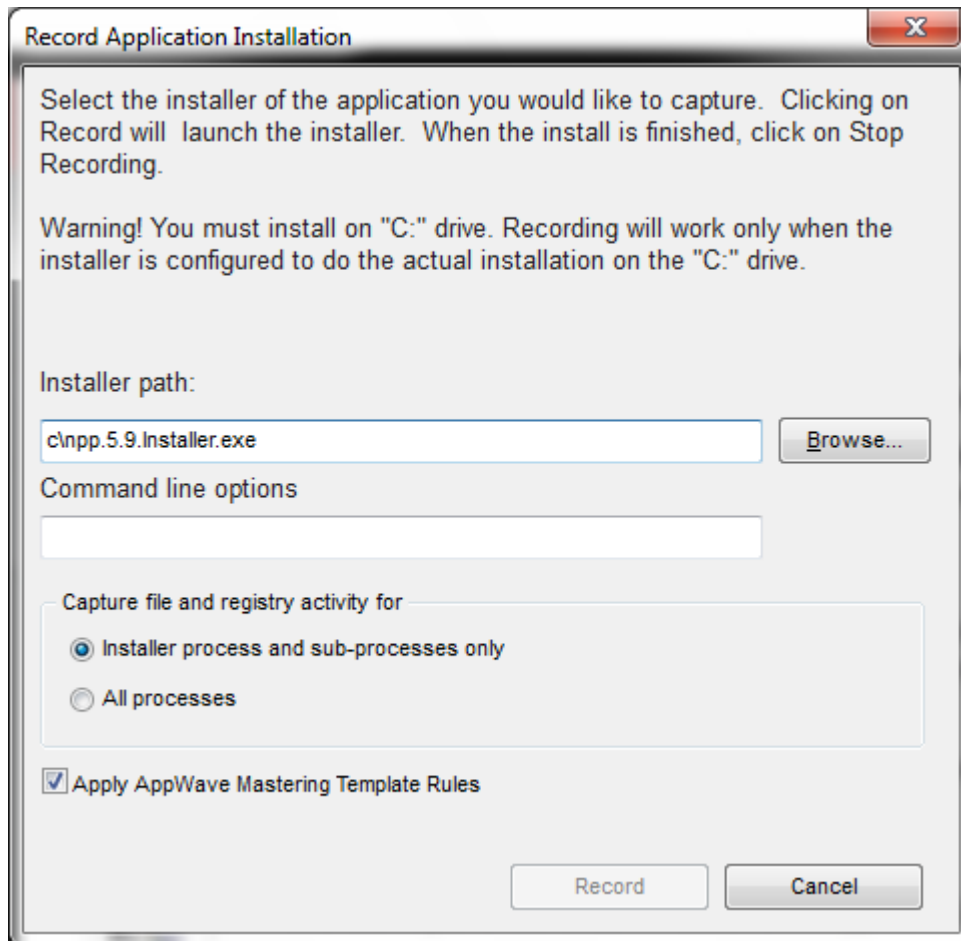


3  Browse to the path where the installer file resides. The executable must be on the root drive or the recording will not work. The example above shows the NotePad++ executable.

> **NOTE:**  Click Install Using a Zip File if your executable is a zip file.

4  In the **Command line option**s field, enter any command options you want to use. For information on commands, see Using the Command Line.

5  In the Capture section you can choose to record only the installer process and sub-processes, or all processes. The first option is the default setting.

6  The **Apply AppWave Mastering Template Rules** option is selected by default. As the process goes through the steps to create an App, warnings may be generated indicating missing data based on the template.

7  Click **Record** and the Recording dialog appears (typically in the upper right-hand corner of the screen). The AppWave Studio is minimized during the recording process.

.
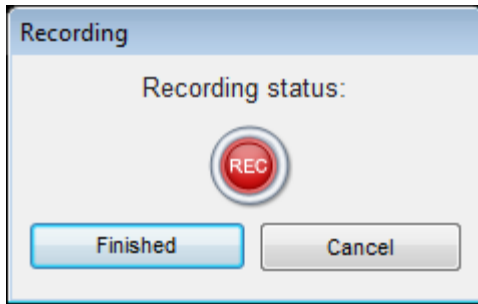


8   Follow the installation wizard for your selected App. There is no need to launch the application after the installation is complete.

9   The licensing of your app depends on what type of AppWave licensing you have purchased as well as the type of app you are creating. For full explanation of how to license your App, please refer to: Managing Licensing.

10  Once the install process is complete, click **Finished** in the **Recording** dialog and the **Record installation** dialog appears.

　　NOTE:   An indicator window opens in the lower right-hand corner of your screen showing that the selected file has finished downloading. The file extension .aw indicates it is the AppWave Template file.
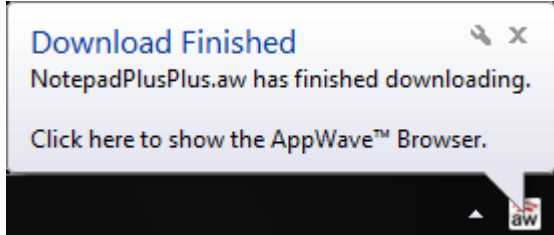


11  Click **Details** to see the details of the installation while it is recording or after it completes. If you want to see it after the recording is complete deselect the "Close on successful completion" option and then click **Details**.

12  The **Record installation** dialog closes automatically if you left the "Close on successful completion" option selected. Otherwise, click **Finished** to close the dialog. The next step that opens depends on the app you are installing.

You are then directed to complete the remaining steps to successfully master your App. The different steps are:

- The <u>Launch & Brand</u> page is where you select the launch executable and working folder and provide branding information for the App.

- The <u>App</u> page is where you can configure your app working with files, registry entries, environment variables, and required runtimes and add custom scripts to your App.

- The <u>Test</u> page is where you master and test before broadcasting to the AppWave.

- The <u>Broadcast</u> page is where you broadcast your mastered app to the AppWave.

For more information, see: <u>Using the Start Task.</u>

## LAUNCH & BRAND

On this page you select the launch executable.The working folder is then populated. You can also set any parameters needed for the app and select an icon that you want to use. It is also where you identify the App. The product information entered here is used in AppWave to identify it.

For more information, click <u>Launch & Brand Fields</u> for detailed information on each field.
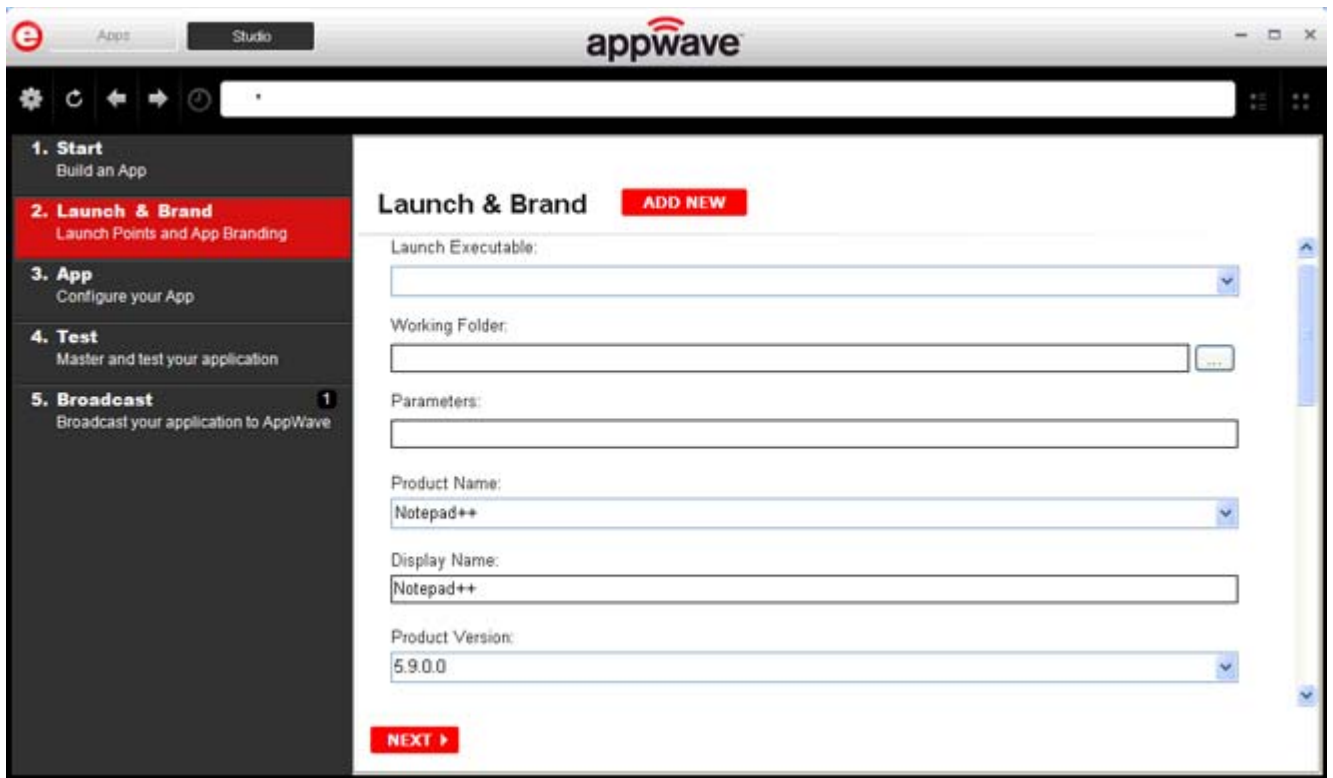
**NOTE:** If a Tasks pane is open at the bottom of the page, this means there are fields that are missing required information. Click **Fix** next to the task and your cursor is placed in the appropriate task. Once the tasks are corrected the **Tasks** dialog closes. The fields with an exclamation mark ( ⚠ ) are the fields missing required information.

## LAUNCH & BRAND FIELDS

1 Complete the following fields:

• **Launch Executable:** The executable (.exe) file that runs the application. Click the drop-down arrow in the Launch Executable and select the executable.

• **Working Folder**: Once you select the desired file, the **Working Folder** is automatically populated.

• **Parameters**: These are requirements used with the command, so the content of the parameter depends on the specific app being mastered. For example, if you were mastering RAD Studio, a command executable would be "bds.exe". But if you wanted to create a Delphi app you would add "-pDelphi" as a parameter for that command.

• **Product Name**: This is a required field. Enter the product name along with the trademark sign. This is an internal name used for the product not displayed to customer.

- **Display Name**: This is the name that is displayed on the AppWave Store and on the AppWave Browser. It should include the trademark sign.

- **Product Version**: If no product version is specified in the installer than it will be filled in with the default value of "1.0". Production version cannot contain build numbers. Format accepted is "a.b.c.d". Valid version numbers would be: 1.0, 1.0.1, 1.0.1.1. Invalid version numbers would be: 1.0.1234, 1.0.1.12345.

- **Product Edition**: This is mandatory if it exists within the app. Sample editions would be: Professional, Home, Ultimate, etc.

- **Publisher**: To locate the vendor name easily, you can use the Copyright phrase usually displayed at the bottom of the official website. For example: "Copyright © 2011 Adobe Systems Incorporated. All rights reserved." Which means the vendor needs to be "Adobe Systems Incorporated" The maximum length of this field is 256 characters.

- **Icon**: Icons must be created using this tool: docs.embarcadero.com/products/appwave/vendor/IconGenerator/uploadIcon.html. The input of the iconize tool is a 130 x 130 pixel graphic.The icon must be placed in the following folder and be in the .png format. C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.png.

- **EULA**: The EULA file must be in the .htm format using the Arial font, size 10. It should be located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.htm

  **NOTE:** If the application require prerequisites (like dotNet Framework, MSVC 2005/2008, J#, MDExplorer, Adobe Flash, etc) then the correspondent EULA file should be included as well.

- **Screen shot**: This is a screen shot of the main window of the application. It must in .png format and located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<screen name>.png.

  **NOTE:** When creating a screen shot, please take into consideration the fact that they will be scaled to 228 x 174 pixels and they must not be distorted; Also, the images used must not be under copyright.

- **Web Site**: Enter the URL for the applications web site.The maximum length is 512 characters.

- **Support Contacts**: List the email first. If that is not available, list the contact page (or form) from the website. Or you can list the phone number.

  **NOTE:** If the contact page contains one entry per country then the link to the page is enough.

- **System Requirements**: If no particular requirements are needed then it must contain the phrase 'Windows XP and higher". If the application has pre-requisites then this field is mandatory. The maximum size of this field is 1024 characters.
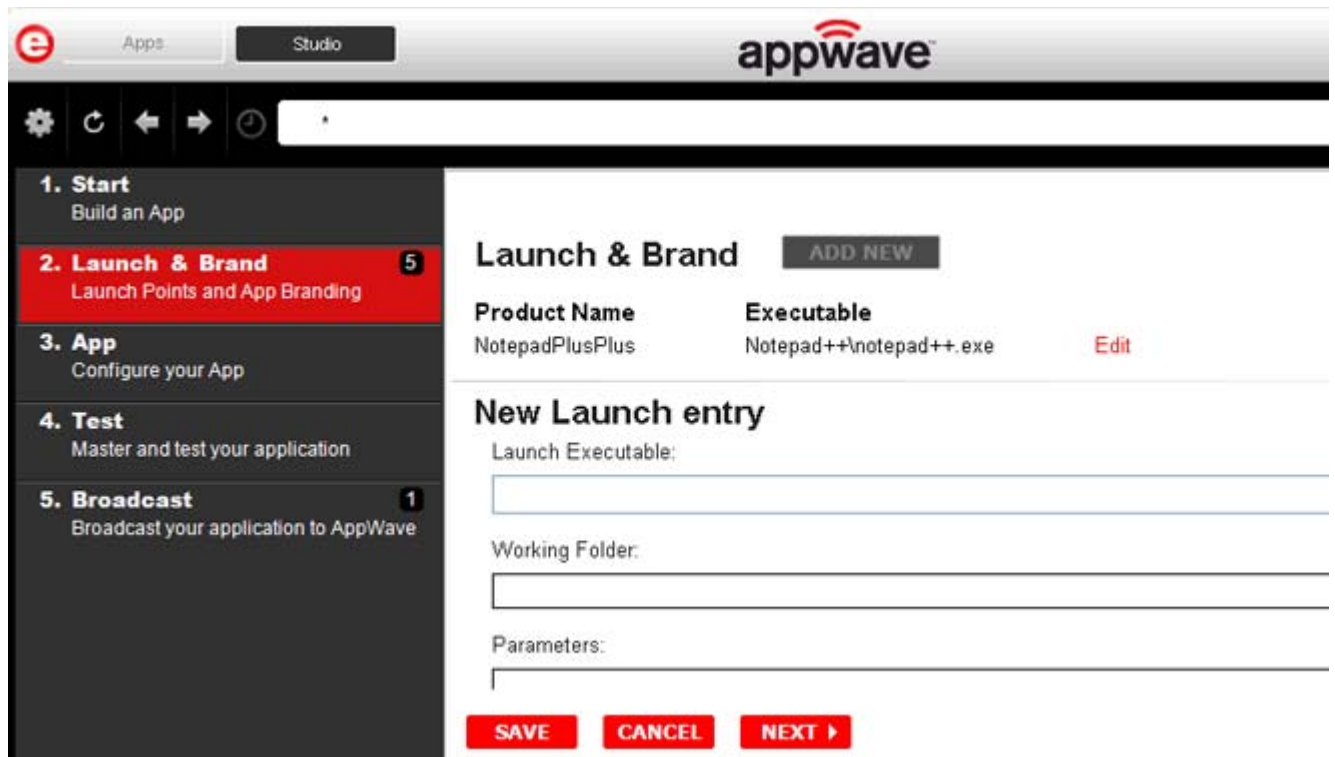
  In case the app requires pre-requisites, this info must be split into two sections:

  - Requirements for running the app: List all the pre-requisites that the app needs to run that Embarcadero does not offer runtime plugins for. An example would be Adobe AIR.

  - Requirements of the native application: List everything the native application needs even if a runtime plug is available. Examples would be: .NET 3.5 SP1 runtime, Adobe AIR runtime, Microsoft Visual C++ 2008 SPI runtime, and Windows installer 3.0.

- **Product Description**: Should not consist of more than two sentences, although there is a 5000 character limit. Include keywords a user might search on such as game, drawing, database, application development, image editor.

  For example, the product description for Photographics is: "Photographics is an image editor that provides powerful tools for designers and developers. The information here cannot be copyrighted."

- **Long Description**: The long description cannot contain part of the Product Description and vice versa. For commercial apps, if the description is taken from the vendor's website, a copyright phrase must be added at the end of this description. For example:  "Copyright © 2011 Adobe Systems Incorporated. All rights reserved." This maximum size of this field is 5,000 characters.

- **Release Date**: This is the date when the application was released, not the date when the app was mastered or published on AppWave Store.

2  If you want to include another application in the App, such as for an application suite that consists of multiple products, click **ADD NEW** and complete the fields for the new launch entry. Click **SAVE** when finished.

   On this page you can add other apps that you want to include in the .tca file. For example, you may want the Browser user to download a database client in addition to TOAD. You would then click **ADD NEW** to add one or more applications. You will need only to broadcast one file but the Browser user will see both apps in the file as unique items in the App catalog.

This page requires the same information: Launch Executable, Working Folder; Parameters; Product Name; Display name, Product Version; Product Edition; Publisher; Icon; EULA; Screen shot; Web Site; Support Contacts; System Requirements; Product Description; Long Product Description; and Release Date. All information designated by an exclamation mark

( ⚠ ) are required fields.

Note that the product name and executable that you are adding to are listed. Click **Edit** to make changes to this product information.

3   Complete all the information. Click **Save** if you want to add another app.

4   Click **Next** to advance to the App step.

For more information on this step, see: Using the Launch & Brand Task.

## APP

1   When you have completed the **Start** and **Launch & Brand** steps, click **App** and the **Configure your App** page opens.

The **Configure your App** page is where you can view what was created when you built a new App. You can edit, add, or delete the following that were created during the mastering process: files, registry settings, and environment variables. You can choose required runtimes and select custom scripts to run when triggered.
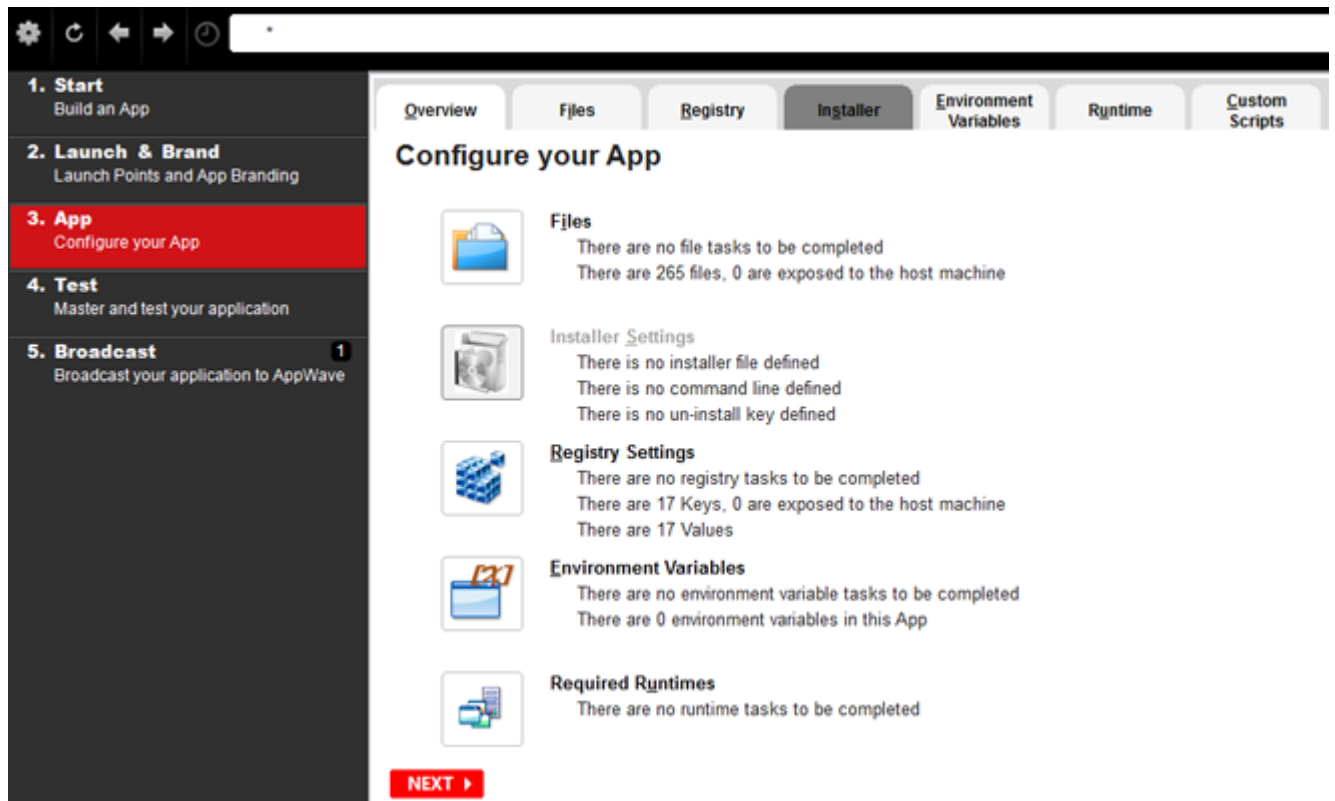
2   Click **Files** and the **Files entries for your App** page opens.

This displays the working folder containing the managed files including the executable. Here you can add or remove files, create folders, and change the properties of files.

> **TIP:**   It's important to know how the application installation is built in order to make changes to the file entries here.
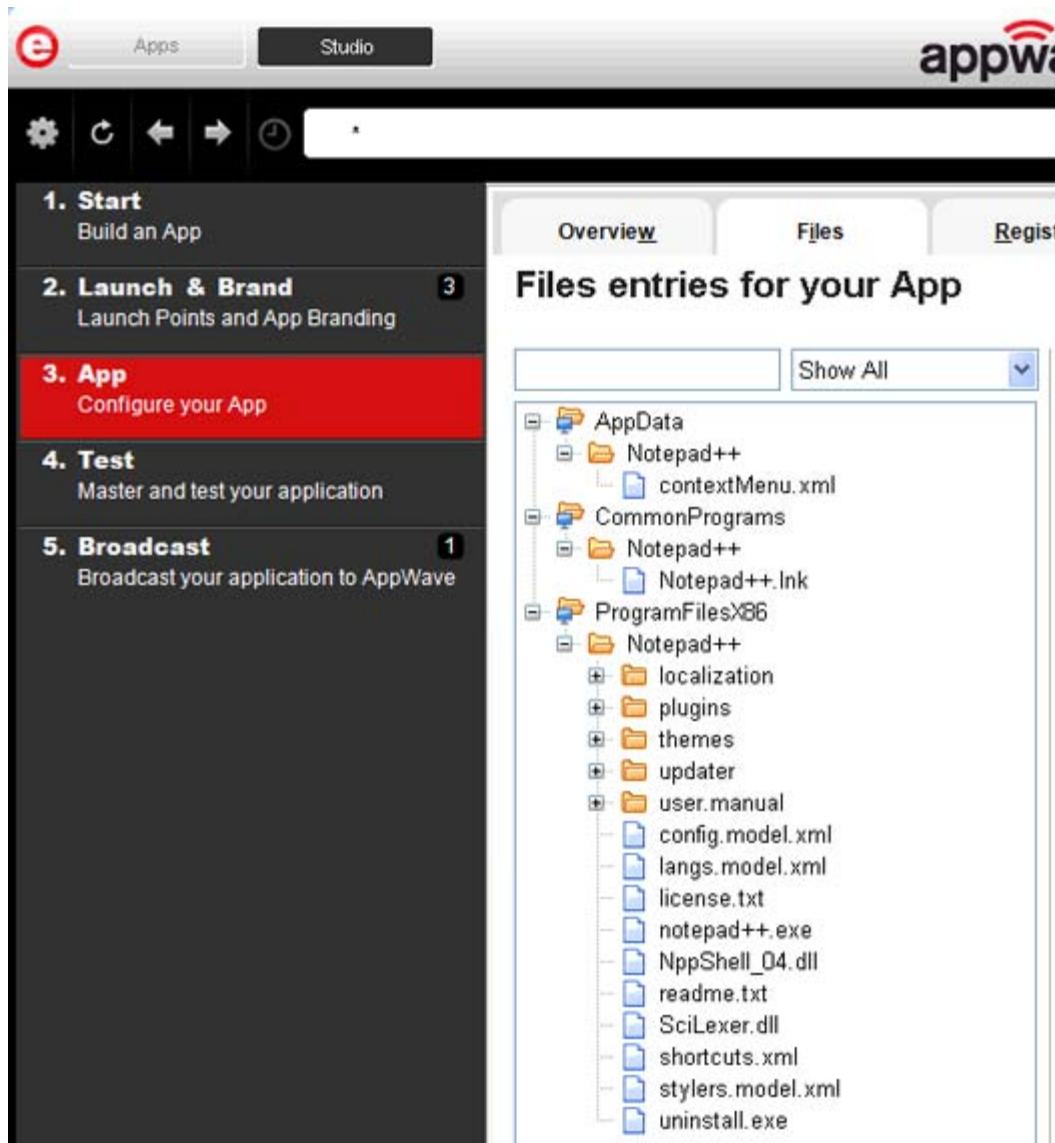
3   Click **Next** to advance to the **Registry entries for your App**. All registries and any changes to those registries are listed. These registry keys are added automatically during the installation recording.

   **NOTE:**   When modifying registry keys it is important to know the system variables. You need to know what variables can be used that will be replaced with the correct path when an app is loaded on a machine. For more information, see: Predefined Virtual Folders.
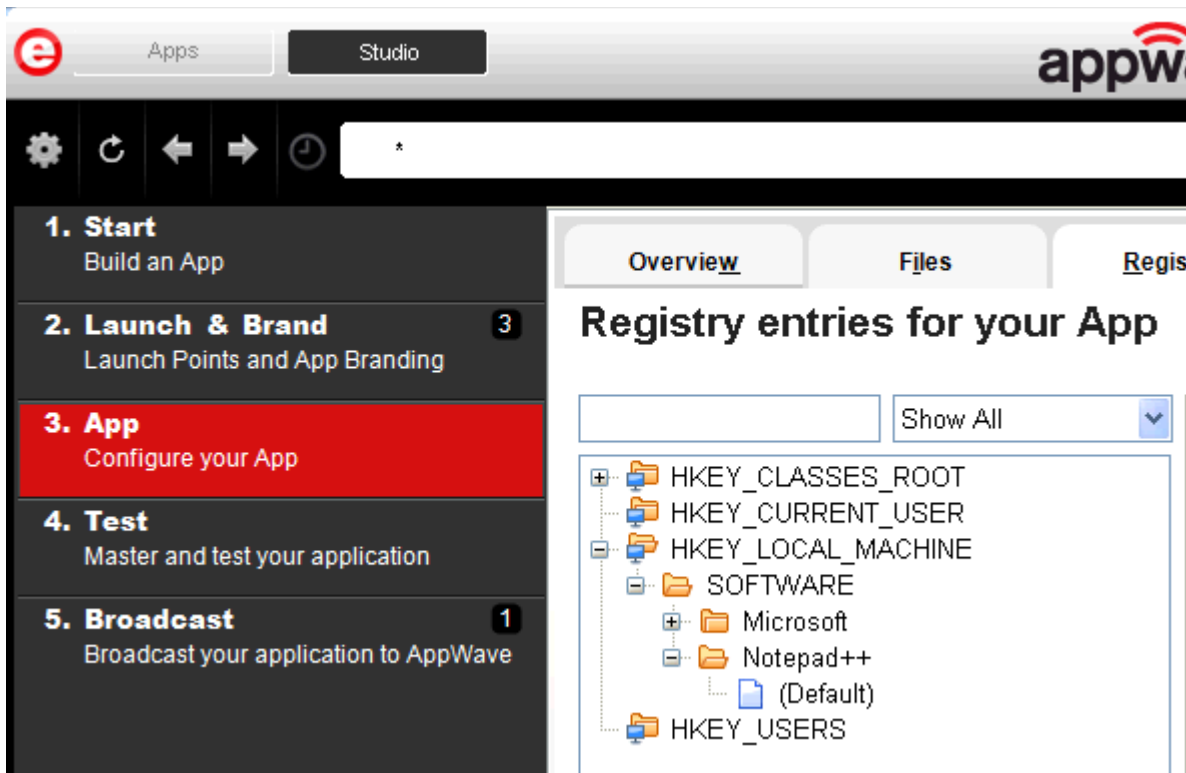
4   You should review these entries before you complete the mastering process. There may be files/registry keys that do not relate directly to the application and have been added in error during the recording. Remove any that obviously do not belong, for example, those relating to the Windows installer.

> **NOTE:**   Also watch for registry keys that directly refer to a user or drive instead of being generic. For example:
>
> C:\xxx\<username>\My Documents instead of %USERPROFILE%\My Documents.

5   Click the Environment Variables tab for the **Environment variables for your app** appears. You can add, edit or remove any environment variables on this page.

6   Click the Runtime tab for the **Runtimes required by your app** page. A default list of runtime plug-ins are listed. You can select the ones required by your App.
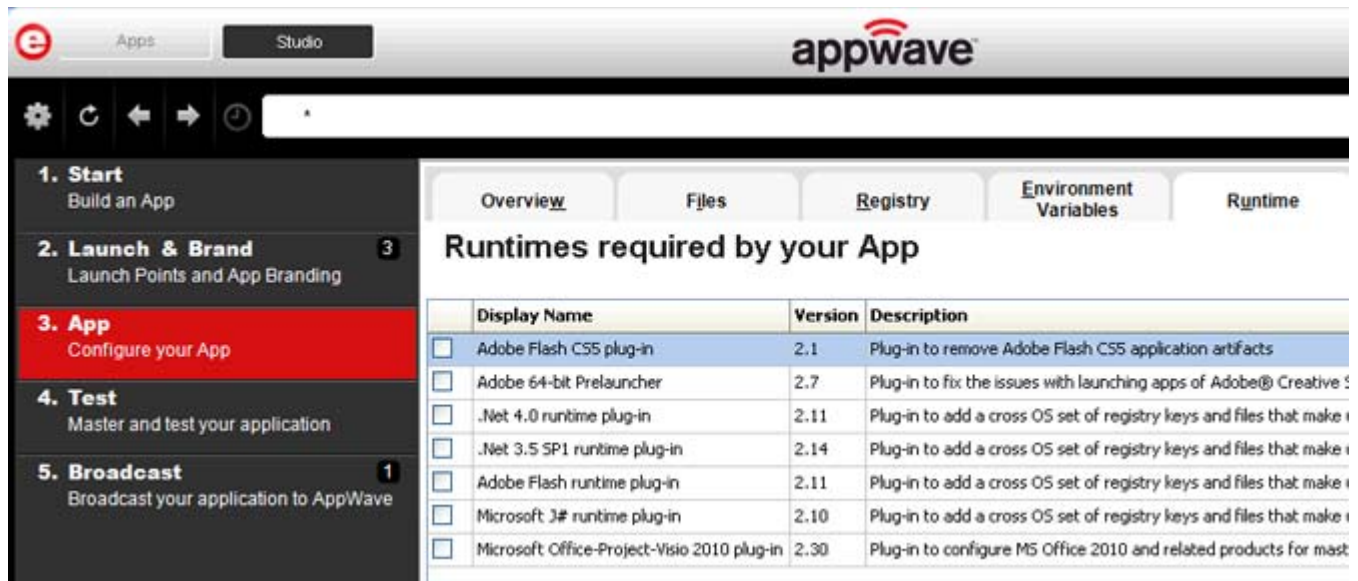
7   Click the Custom Scripts tab to advance to the **Custom Scripts for your app** page. You can attach scripts to key application triggers, such as activation, start, close, and deactivation that can accomplish whatever is necessary. These scripts are typically small executable files and batch files. Executables, batch, and command files are supported to perform acts such as:

• Establishing a connection to a shared network location before the application starts

• Running a file backup/archive process for user created documents after the application has exited.

| Overview | Files | Registry | Environment Variables | Runtime | Custom Scripts |

## Customs Scripts for your App

| Triggers | Name | Enable | | |
|----------|------|--------|------|--------|
| After Activation | None | ☐ | Edit | Delete |
| After Load | None | ☐ | Edit | Delete |
| After Launch | None | ☐ | Edit | Delete |
| After Exit | None | ☐ | Edit | Delete |
| Before Deactivation | None | ☐ | Edit | Delete |
| Before Unload | None | ☐ | Edit | Delete |
| Before Run | None | ☐ | Edit | Delete |

## Trigger: After Activation

**Name**

[                                    ]

**Frequency**

⊙ Fire the first time only    ○ Fire every time

**Script**

Filename: [                                    ] 📁

Parameters: [                                    ]

☐ Use Local Application

**Event**

☐ Run outside of the App

☐ Run elevated (only supported on Windows Vista or later)

☐ Wait for the handler to exit

Time out after: [1        ] [Minutes    ▼]

Success code: [0              ] (Separate multiple codes with commas)

Any other values returned by the handler will be treated as errors

**NEXT ▶**   **SAVE**   **CANCEL**

For more information, see Custom Scripts. For this exercise we won't use any scripts.

For more information on the App step, see: Using the App Task.

## TEST

Click **Test** to master and test your app locally. This option allows you to master an app using the template .tca file that you just created.

This also where you can record the launch of the application for faster start-up time. This prefetch action performs the process of streaming application blocks in anticipation that those blocks are needed by an application. Only some key blocks of application data from AppWave are necessary for the application to launch and perform basic functions.

A .tca file (also known as the Studio file) is a set of information that defines the resources, launch points, and registry settings required to run a product. It also contains information about the product such as 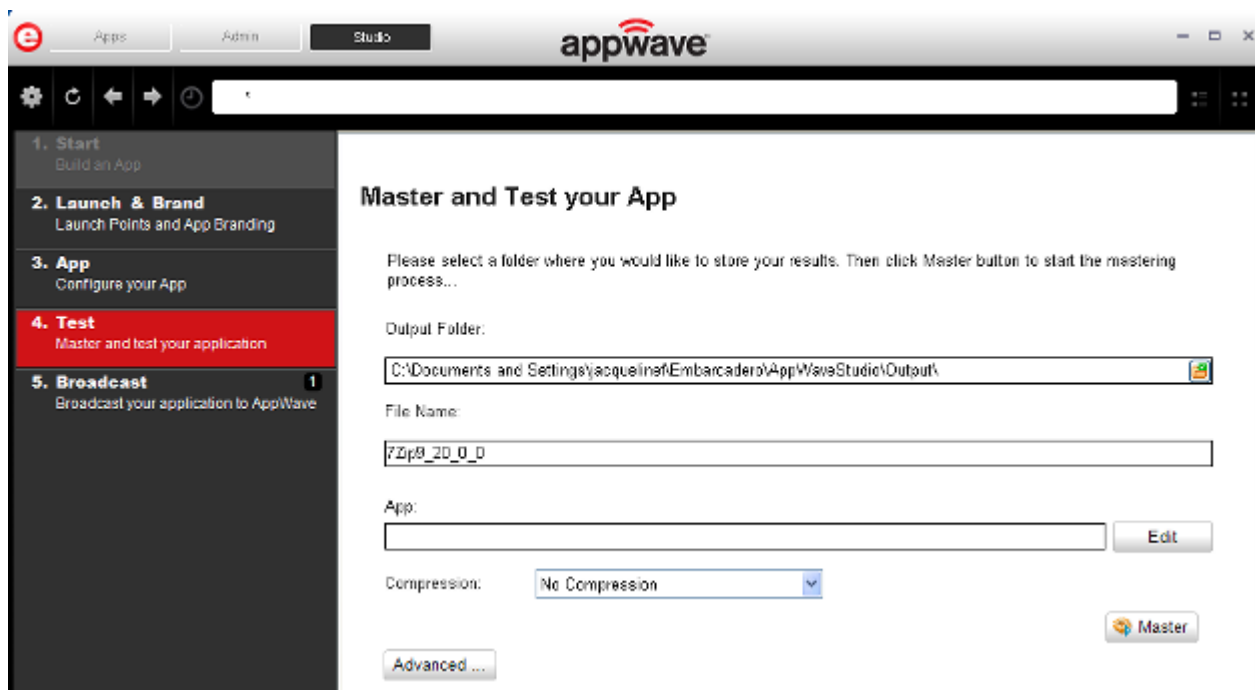the version, name, and editions. The .tca file needs to contain all the files that are associated with it. The .tca file has file system information that actually lists files for the program it needs to master. The files have to be physically present at the directory information that is saved in the .tca data.



1. The **Output Folder** is automatically populated with the path, i.e.,
   `C:\xxxx\<user>\Embarcadero\AppWaveStudio\Output\`.

2. The **File Name** is also automatically populated with the information from the **Launch & Brand** page.

3. In the **App** field if an app file exists for the app (.stp), click **Edit**. Select the App file (.stp) to associate with the .tca file for this App.

   **NOTE:** This .stp file is typically located in the same output directory
   `(c:\xxxx\<user>\Embarcadero\AppWaveStudio\Output\<app name.stp>)`.

4  Click the drop-down arrow in the **Compression** field and select the type you want to use. "Best Compression" is the recommended setting for Compression.

5  If you want to select the type of encryption to use in the App, click Advanced.

6  After your have verified your selections, click **Master**. If you haven't saved your file, you will be asked to save your changes to the current working file.

7  The **Master** dialog appears with a progress bar showing the mastering steps. Click **Details** to view the mastering details.

8  Once the mastering is completed, an option to **Record the launch of the application for faster start-up time** appears in the Master dialog.

9  Click **Launch** to open the app and the **Recording** dialog opens once again.

10  Accept the End User License Agreement that appears and then once the application is open, click **Finished** in the **Recording** dialog.

11  The **Master** dialog reopens.
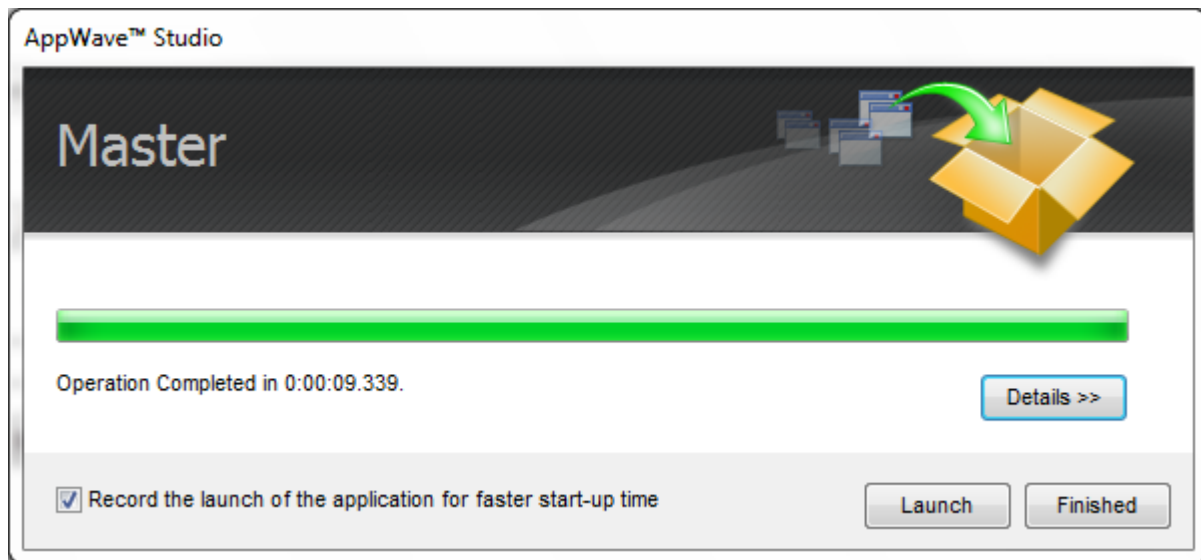


12  Click **Finished** to close the **Master** dialog.

13  Click **Next** to advance to the **Broadcast** step.

For more information on this step, see: Using the Test Task.

## BROADCAST

1   On the **Broadcast** page, notice that a Task appears at the bottom of the screen. This task indicates that the Prefetch function is recommended in the mastering process. Prefetch is the process of streaming application blocks in anticipation that those blocks are needed by an application, **as** well as to perform typical actions for the first few minutes of using the prograM, such as opening files, saving files, etc. The process of streaming these blocks from the server to AppWave Browser is also called *buffering*.

Only some key blocks of application data from AppWave are necessary for the application to launch and perform basic functions.

Unlike conventionally installed applications, application streaming does not require all application files to be always available.

The prefetch file is created by logging pages that are being streamed when the application is executed. This includes all the functions required by the application when the user opens it for the first time.

The task states:

*"In order to Record the launch of the application for faster start-up time, you need to master the app from the Test tab before broadcasting. Once the launch data was recorded, remastering before broadcast will not remove this information."*

> **TIP:**   If you do not want to use the Prefetch option, you can ignore the Task to be fixed and just click **Broadcast**.

> **NOTE:**   There is an indicator next to the Broadcast step showing that there is a task that needs to be fixed.



2   In the **Tasks** pane, click **Fix** and a dialog appears instructing you on how to record prefetch information.

3   Click **OK** to close the information dialog.

4   Click the **Test** page and then click [Master]. The **Master** dialog appears. If you have not saved your file a message opens. Click **Yes** and save the file named FileZilla_FTP_Client.tca.

5   The **Master** dialog opens and masters the App.

6   Once it is complete, check to make sure the "Record the launch of the application for faster start-up time" option is selected. This option is selected by default.

7    Click **Launch.** The End User License Agreement as well as the Recording dialog opens. When you click **Accept** the application opens.

8    Once the application opens, click **Finished** in the Recording dialog (this dialog reopens when you click Master) and the application closes.

  NOTE:    In the **Master** dialog, click **Details** to view the results of the recording.
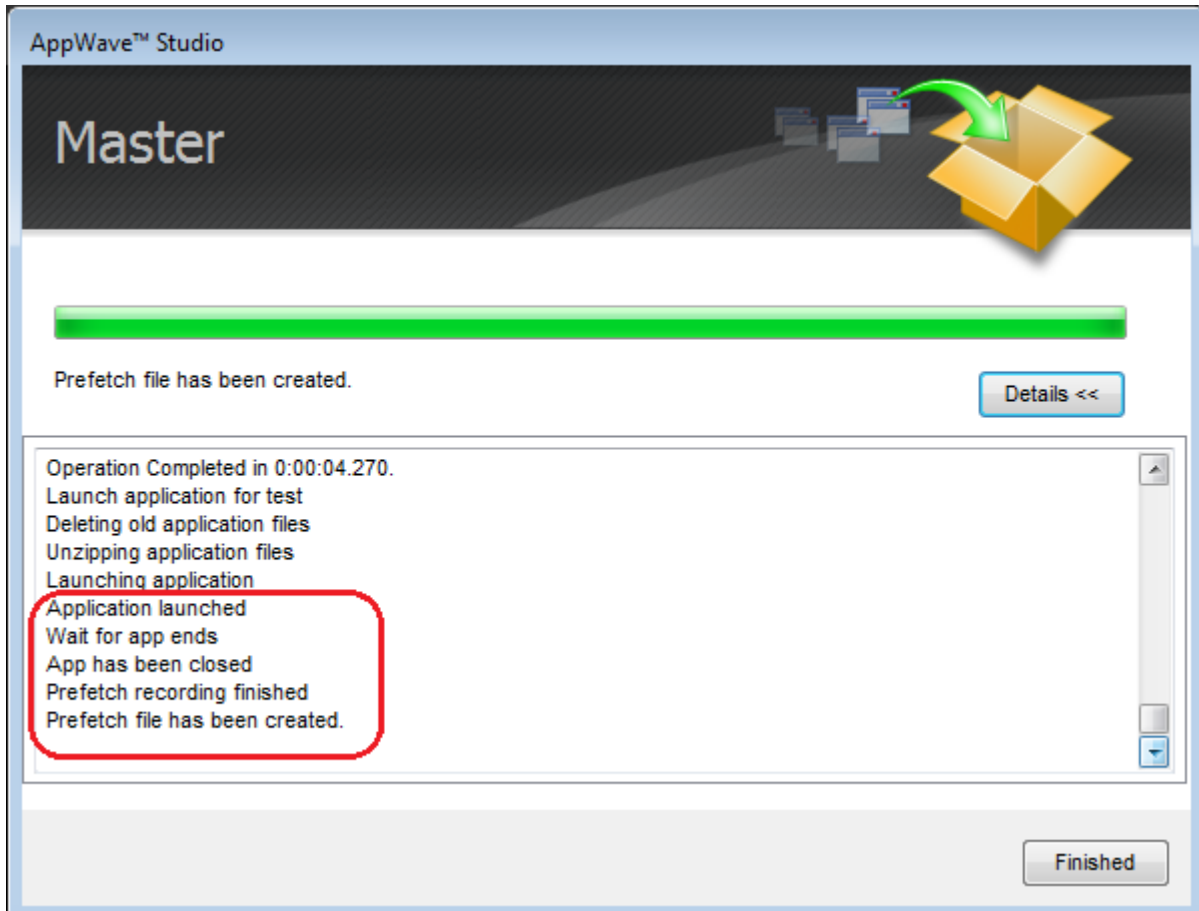


9    On the **Broadcast** page, click **Broadcast** and the Broadcast dialog appears.This dialog tracks the process of the broadcasting. Once you get the indication that the broadcast was successful, click **Finished** and the Broadcast dialog closes.
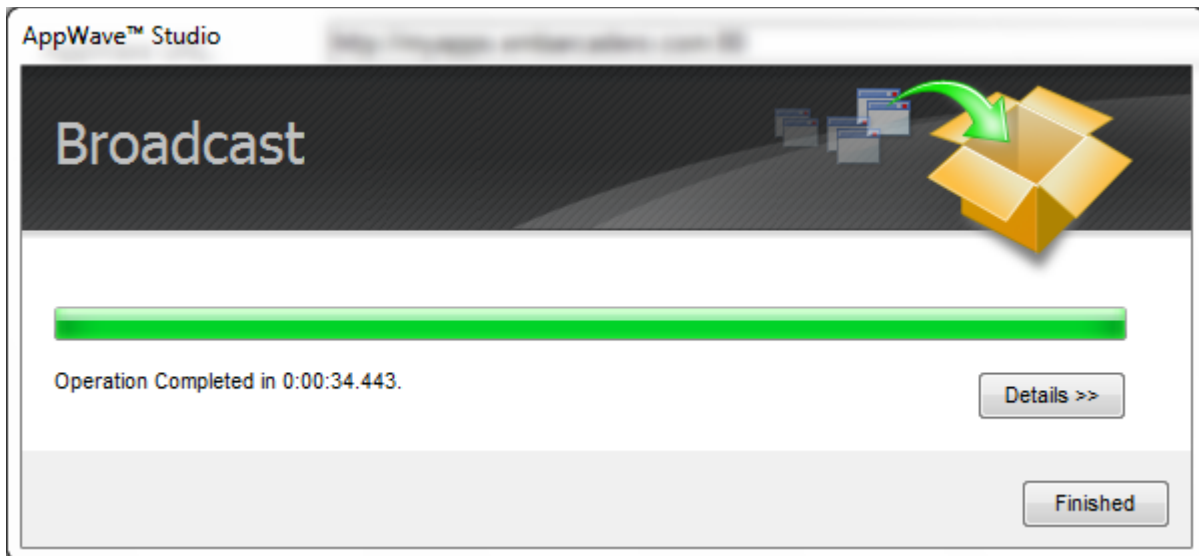
> **NOTE:** A popup also appears in the lower right corner of your screen indicating that new apps are available.
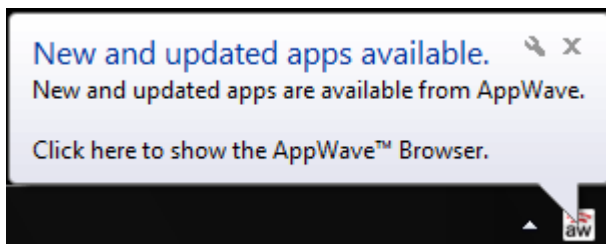


10  Click **Finished** and your app is now mastered

> **NOTE:** The files created have the file extensions of .aw, .app, and .stp and are placed in the C:\<folder>\<user>\Embarcadero\AppWaveStudio\Output folder.

For more information, see: Using the Broadcast Task.

## RUNNING THE NEWLY MASTERED APP

1  Click the **Apps** tab to open the AppWave Browser.

2  After your app is broadcast, the newly mastered app is placed in a working folder. The following screen shot shows one app has been placed in "embarcadero.comkathyb's Folder" designated by the 1, as well as the app that was just mastered.

> **TIP:** If your folder does not appear you may need to reconnect to the AppWave Browser by clicking ⚙ > **Connect**.
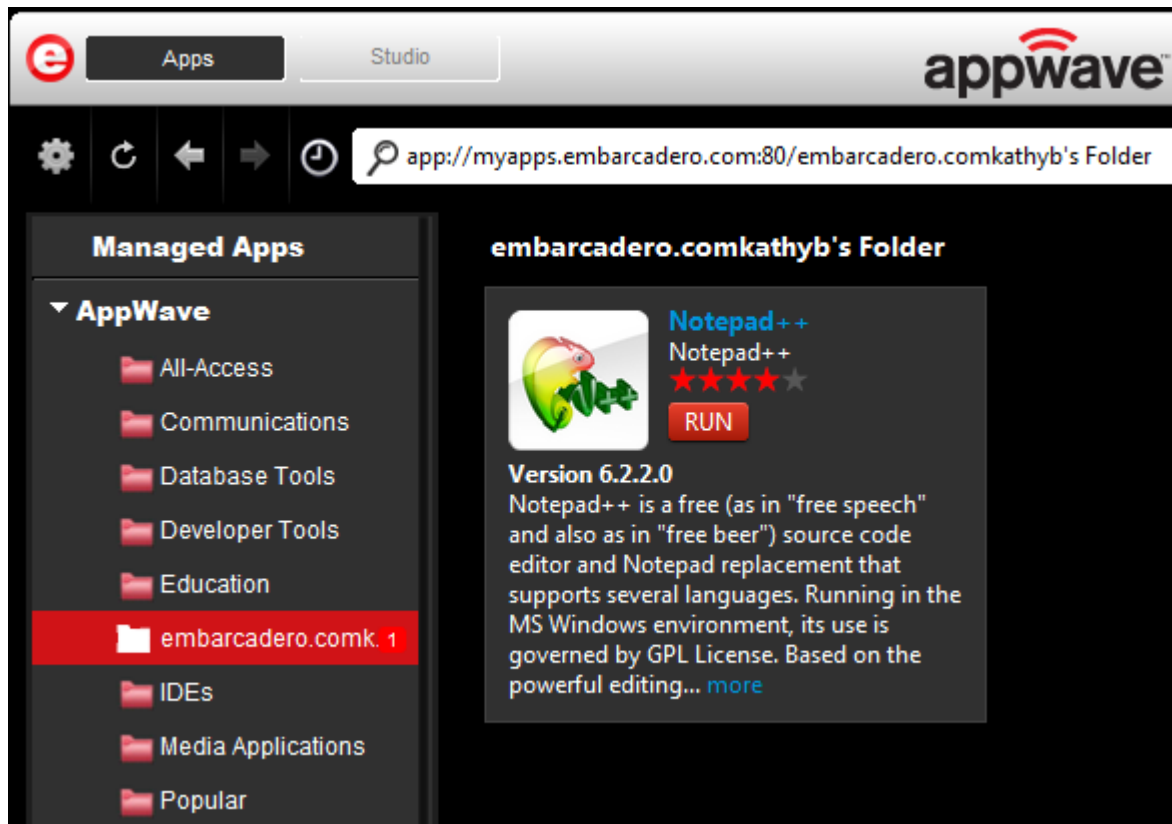
3   Click **Add** and the End User License Agreement dialog opens.

4   Click **Accept**. If no further licensing information is required the app runs.

> **NOTE:**   You can right-click on the icon in AppWave and an information page opens. Here you can view details of the downloaded app, read and/or write reviews, and see what versions are available.

> **CAUTION:**   During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio will attempt to shutdown any services it detects, but in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the App. For more information see Stopping Services in the Appendix.

# INSTALL USING A ZIP FILE

You can install a custom app from a Zip file which contains the application's executable. The following steps walk you through the installation of Embarcadero's RAD Studio XE Architect using a zip file.

1   On the **Start** page select **Create AppWave Supported or Custom App from Install** and the **Record Application Installation** dialog appears.

2   In the Installer path field, browse to the zip file located in the c:\ drive.
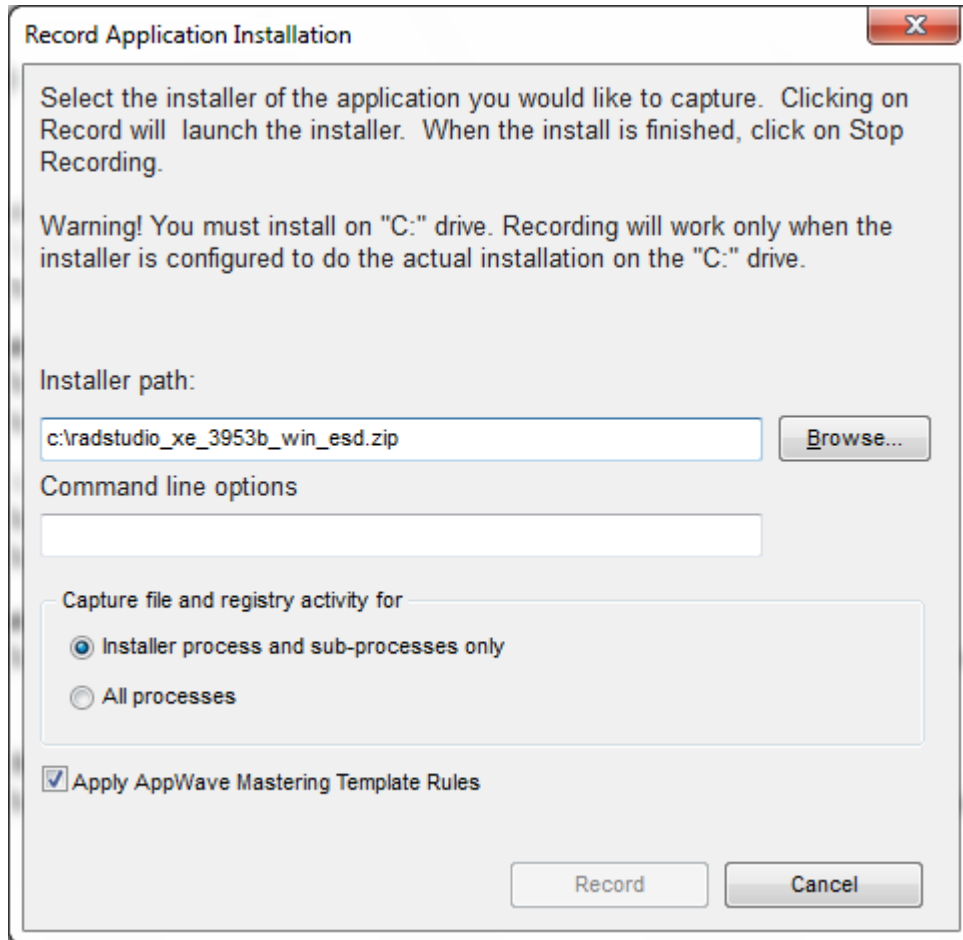


3   Click **Record** and the WinZip dialog appears listing the executables available. The setup executable for RAD Studio is listed in the dialog.
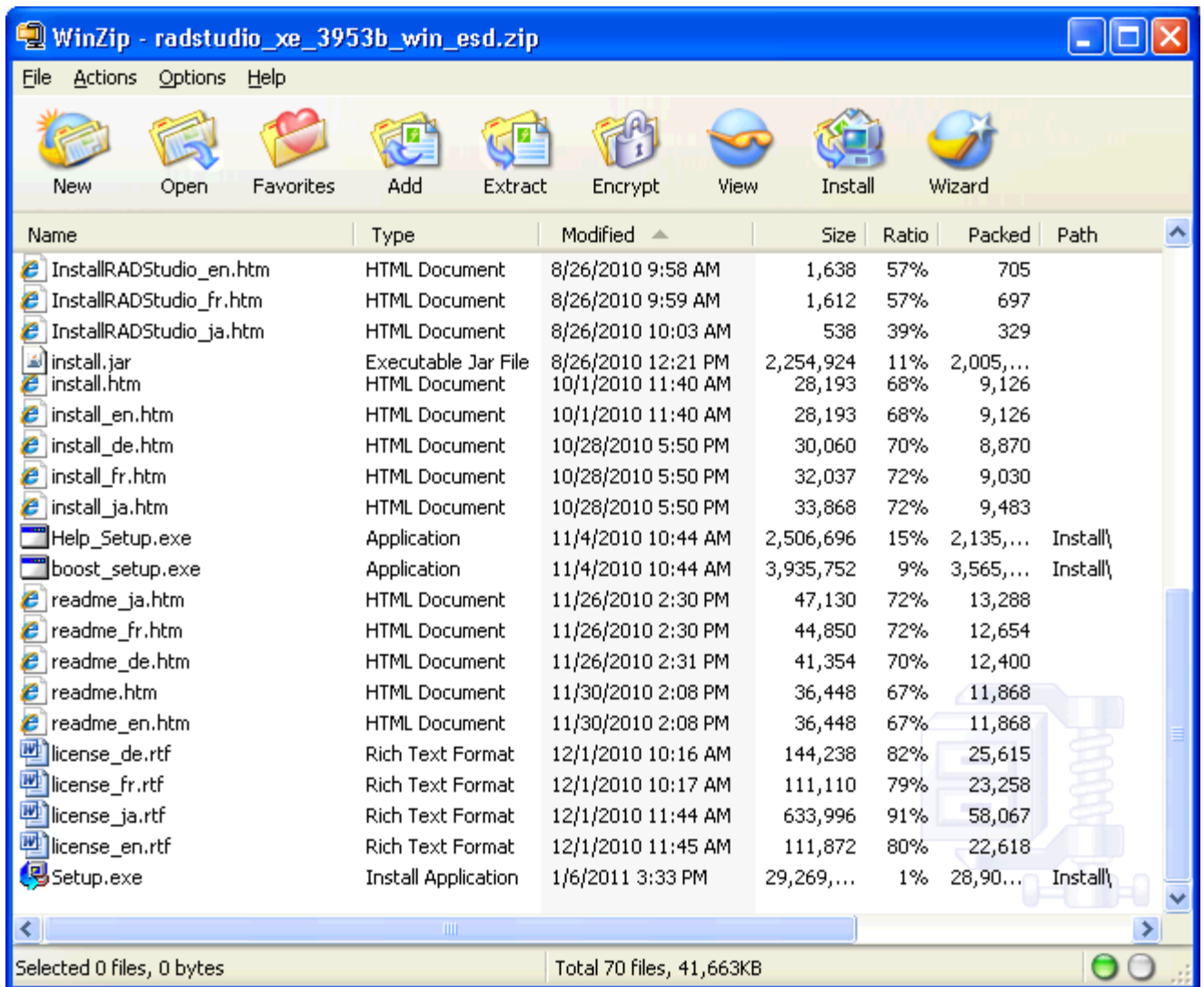
4   Double-click the **boost_setup.exe** and the Install Wizard for that executable opens.

5   Follow the wizard to complete the selected installation.

6   Once you have installed all the desired executables, click **Finished** in the **Recording** dialog and the **Record Installation** dialog appears.

7   Click **Finished** and the Launch & Brand page opens. Complete the rest of the mastering process as described for an executable application.

> **NOTE:**   You also need to close the WinZip dialog.

# CREATE APPWAVE SUPPORTED OR CUSTOM APP FROM SCRATCH

This section walks you through how to create an app from an application's executable file. This is a custom application that does not have any mastering support. This means that you need to manually insert Product Name, Display Name, Version, Long Description, etc. Once your product is mastered and placed in your AppWave, you won't need to master it again.

This option is only available through the AppWave Studio. A user must be granted authoring permission by an Administrator. For more information see: Group Management.

This option provides a way for you to create an app file from a given installer file. It actually extracts all the install files from the installer, and then puts these extracted files in a directory under the OutputLocation directory.

The steps to be completed are:

- Start
- Start
- Launch & Brand
- App
- Test
- Broadcast

## START

1   In the AppWave Browser click **Studio** and the **Build an App** page appears with the **Start** step highlighted in red.
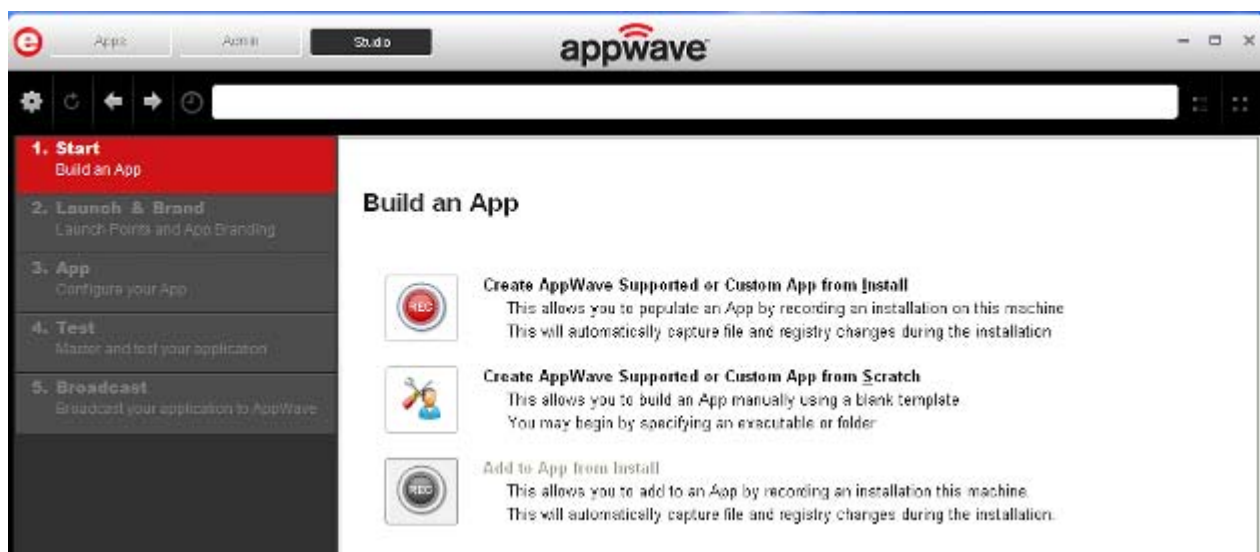
2   Click the **Create AppWave Supported or Custom App from Scratch** link or click 🛠 .

3   The **Select Folders and Files to include** dialog opens. For the **Initial folder name**, enter an existing folder name or create a new one. This is the root folder where the executable will be added.
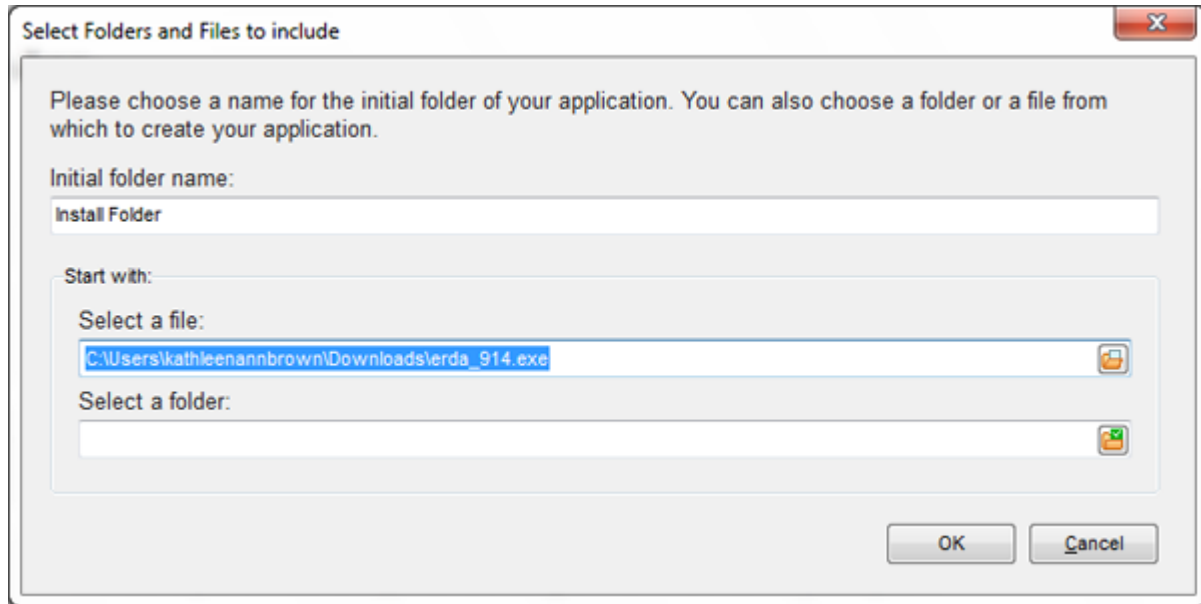


4   In **Select a file**, browse to the Installer path and select the executable. The Installer path must on the root drive. For example: C:\PDFViewer.exe.

5   In **Select a folder** browse to an existing folder. This will contain the executable and files from the creation process.

6   Click **OK** and the **Create From Scratch** dialog appears. You can select the option to close this dialog on successful completion of the operation. The time it takes to complete the operation is also displayed.

   NOTE:   Click **Details** to see the details of the operation while it is creating or after it completes. If you want to see it after the operation is complete deselect the "Close on successful completion" option and then click **Details**.

7   Depending on the option you selected, the Create from Scratch dialog may or may not automatically close. If you selected the "Close on Successful Completion" the Launch & Brand page automatically appears depending on the information provided by the executable. If you did not select that option, click **Finished** to advance to the Launch & Brand page.

For more information on this first step, see: Using the Start Task.

## LAUNCH & BRAND

The Launch & Brand step is where you select the launch executable. The working folder is then populated. You can also set any parameters needed for your App. This is also where you enter critical information for your App. This information for the mastered app appears in the Apps tab of Browser.
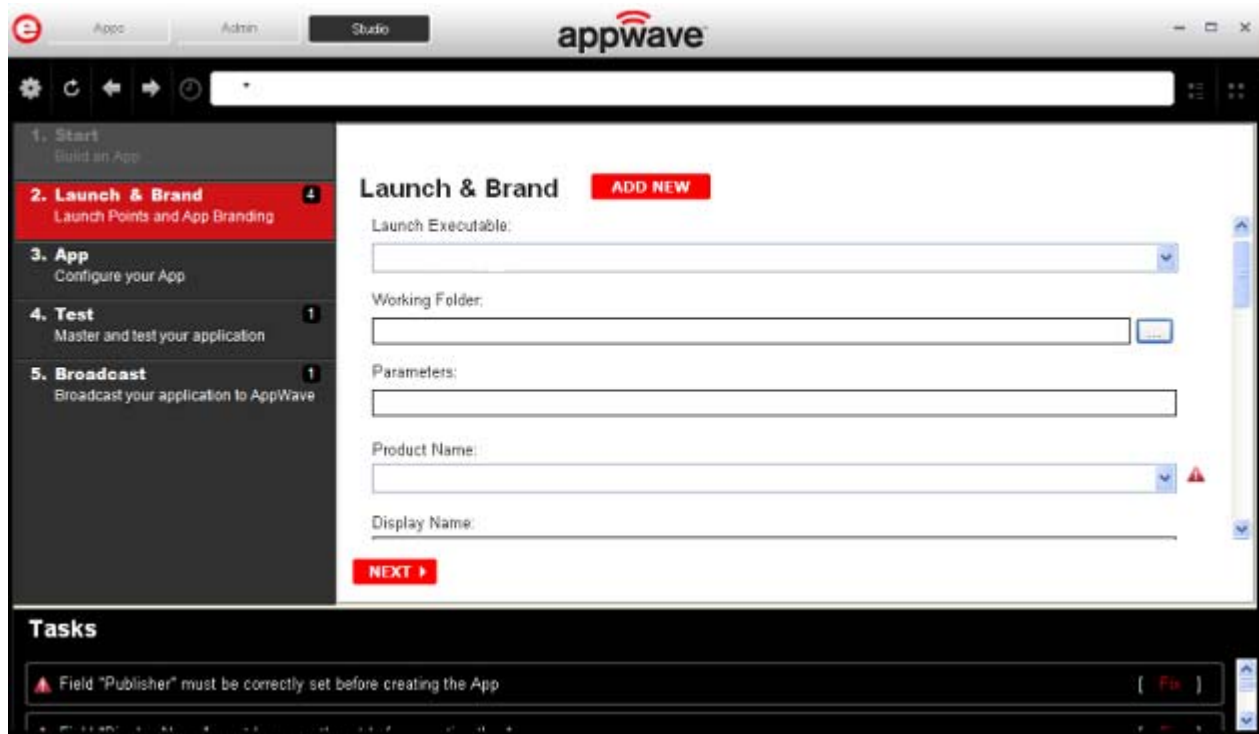


> **NOTE:** If a Tasks pane is open at the bottom of the page, this means there are fields that are missing required information. Click **Fix** next to the task and your cursor is placed in the appropriate task. Once the tasks are corrected the Tasks dialog closes. The fields with an exclamation mark ( ⚠ ) are the required fields missing information.

- **Launch Executable:** The executable (.exe) file that runs the application. Click the drop-down arrow in the Launch Executable and select the executable.

- **Working Folder**: The **Working Folder** is automatically populated from the information you entered in the Initial folder name field.

- **Parameters**: These are requirements used with the command, so the content of the parameter depends on the specific app being mastered. For example, if you were mastering RAD Studio, a command executable would be "bds.exe". But if you wanted to create a Delphi app you would add "-pDelphi" as a parameter for that command.

- **Product Name**: This is a required field. Enter the product name along with the trademark sign. This is an internal name used for the product not displayed to customer.

- **Display Name**: This is the name that is displayed on the AppWave Store and on the AppWave Browser. It should include the trademark sign.

- **Product Version**: If no product version is specified in the installer than it will be filled in with the default value of "1.0". Production version cannot contain build numbers. Format accepted is "a.b.c.d". Valid version numbers would be: 1.0, 1.0.1, 1.0.1.1. Invalid version numbers would be: 1.0.1234, 1.0.1.12345.

- **Product Edition**: This is mandatory if it exists within the app. Sample editions would be: Professional, Home, Ultimate, etc.

- **Publisher**: To locate the vendor name easily, you can use the Copyright phrase usually displayed at the bottom of the official website. For example: "Copyright © 2011 Adobe Systems Incorporated. All rights reserved." Which means the vendor needs to be "Adobe Systems Incorporated" The maximum length of this field is 256 characters.

- **Icon**: Icons must be created using this tool:  docs.embarcadero.com/products/appwave/vendor/IconGenerator/uploadIcon.html. The input of the iconize tool is a 130 x 130 pixel graphic.The icon must be placed in the following folder and be in the .png format. C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.png.

- **EULA**: The EULA file must be in the .htm format using the Arial font, size 10. It should be located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.htm

  **NOTE:**   If the application require prerequisites (like dotNet Framework, MSVC 2005/2008, J#, MDExplorer, Adobe Flash, etc) then the correspondent  EULA file should be included as well.

- **Screen shot**: This is a screen shot of the main window of the application. It must in .png format and located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<screen name>.png.

  **NOTE:**   When creating a screen shot, please take into consideration the fact that they will be scaled to 228 x 174 pixels and they must not be distorted; Also, the images used must not be under copyright.

- **Web Site**: Enter the URL for the applications web site.The maximum length is 512 characters.

- **Support Contacts**: List the email first. If that is not available, list the contact page (or form) from the website. Or you can list the phone number.

  **NOTE:**   If the contact page contains one entry per country then the link to the page is enough.

- **System Requirements**: If no particular requirements are needed then it must contain the phrase 'Windows XP and higher". If the application has pre-requisites then this field is mandatory. The maximum size of this field is 1024 characters.
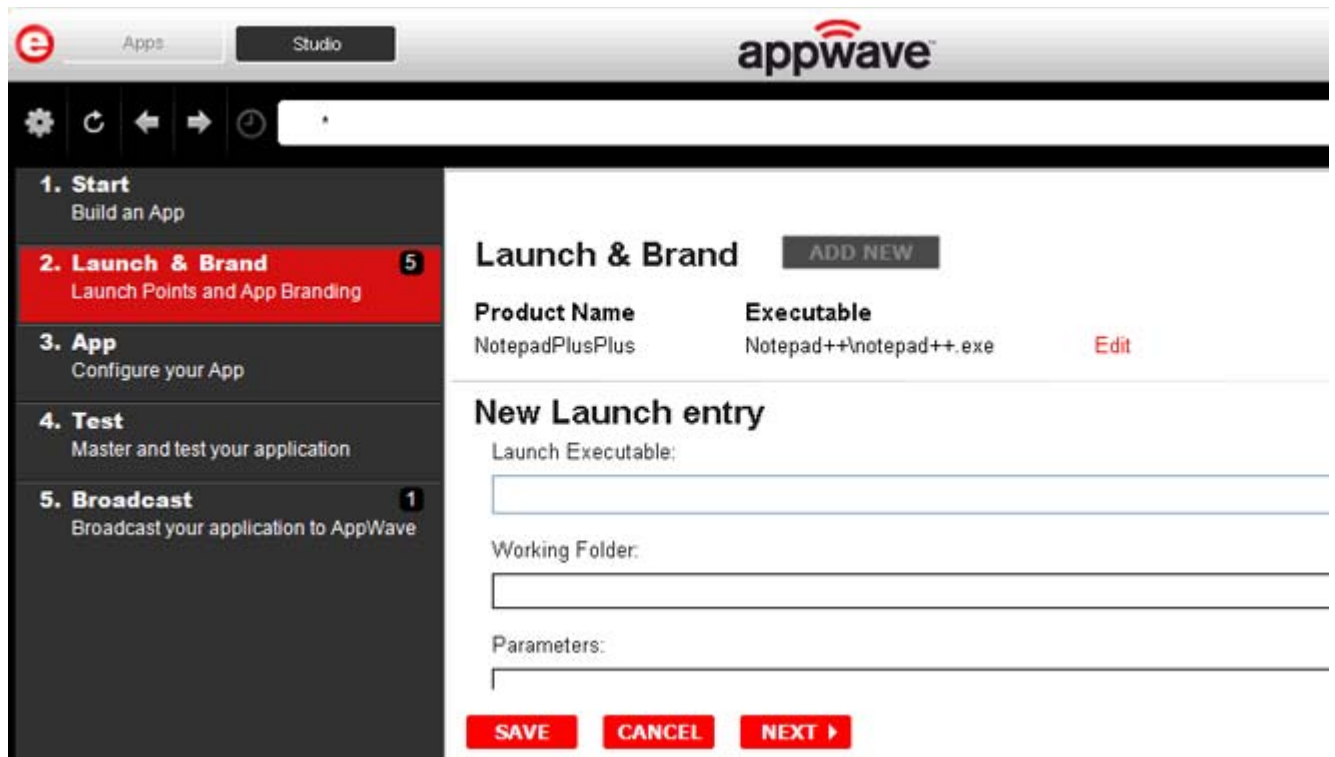
  In case the app requires pre-requisites, this info must be split into two sections:

  - Requirements for running the app: List all the pre-requisites that the app needs to run that Embarcadero does not offer runtime plugins for. An example would be Adobe AIR.

  - Requirements of the native application: List everything the native application needs even if a runtime plug is available. Examples would be: .NET 3.5 SP1 runtime, Adobe AIR runtime, Microsoft Visual C++ 2008 SPI runtime, and Windows installer 3.0.

- **Product Description**: Should not consist of more than two sentences, although there is a 5000 character limit. Include keywords a user might search on such as game, drawing, database, application development, image editor.

  For example, the product description for Photographics is: "Photographics is an image editor that provides powerful tools for designers and developers. The information here cannot be copyrighted."

- **Long Description**: The long description cannot contain part of the Product Description and vice versa. For commercial apps, if the description is taken from the vendor's website, a copyright phrase must be added at the end of this description. For example: "Copyright © 2011 Adobe Systems Incorporated. All rights reserved." This maximum size of this field is 5,000 characters.

- **Release Date**: This is the date when the application was released, not the date when the app was mastered or published on AppWave Store.

1   If you want to include another application in the App, such as for an application suite that consists of multiple products, click **ADD NEW** and complete the fields for the new launch entry. Click **SAVE** when finished.

    On this page you can add other apps that you want to include in the .tca file. For example, you may want the Browser user to download a database client in addition to TOAD. You would then click **ADD NEW** to add one or more applications. You will need only to broadcast one file but the Browser user will see both apps in the file as unique items in the App catalog.

This page requires the same information: Launch Executable, Working Folder; Parameters; Product Name; Display name, Product Version; Product Edition; Publisher; Icon; EULA; Screen shot; Web Site; Support Contacts; System Requirements; Product Description; Long Product Description; and Release Date. All information designated by an exclamation mark

( ⚠ ) are required fields.

Note that the product name and executable that you are adding to are already listed. Click **Edit** to make changes to this product information.

2   Complete all the information. Click **Save** if you want to add another app.

3   Click **Next** to advance to the App step.

For more information on this step, see Using the Launch & Brand Task.

## APP

1  When you have completed the Start and Launch & Brand steps, click **App** and the
   **Configure your App** page opens.
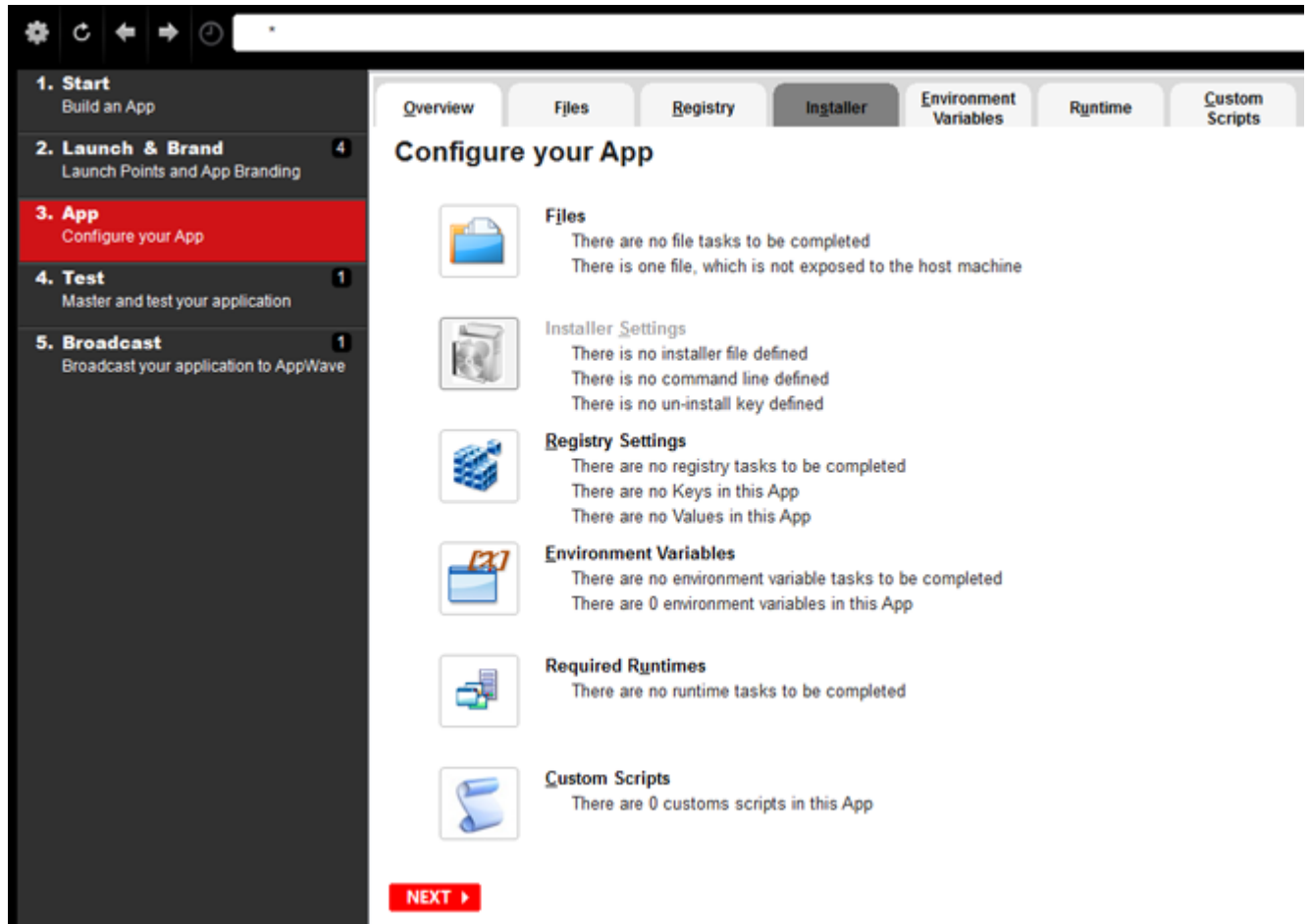


This is where you can view what was created when you built a new App. You can edit, add, or delete the files, registry settings, and environment variables that were created during the mastering process. You can also choose required runtimes and select custom scripts to run when triggered.

2  Click **Files** and the **Files entries for you App** page opens.

3  The folder you created in the first step is shown. Once the managed file is broadcast it will also be shown.
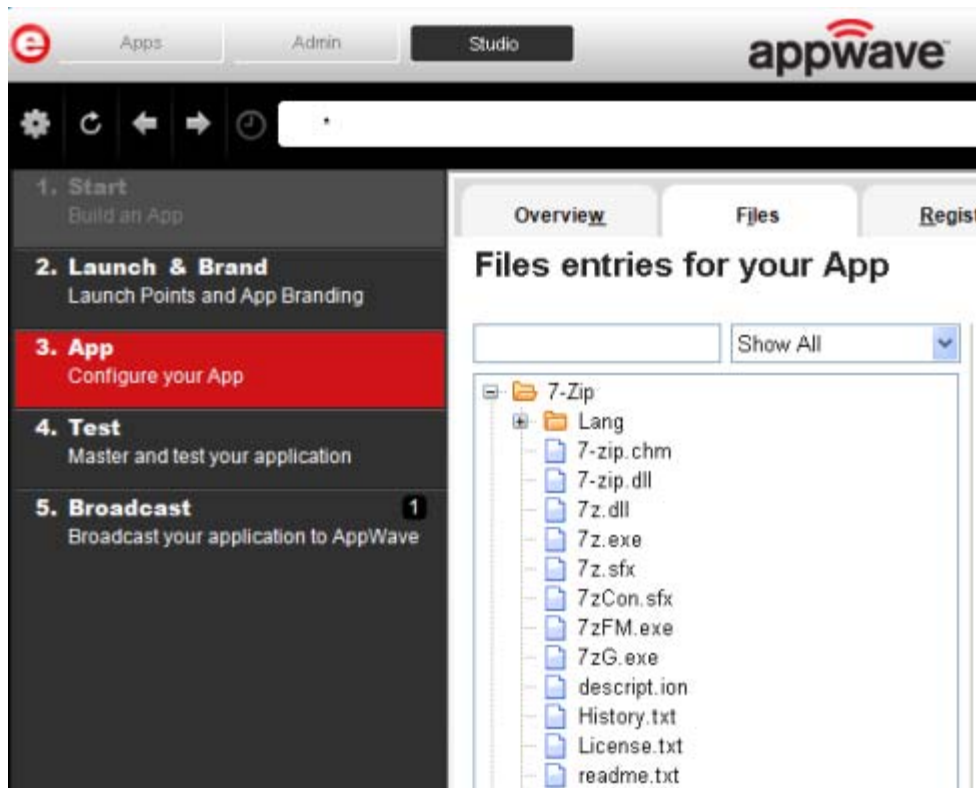
4   Click the Registry tab to advance to the **Registry entries for your App**. All registries and any changes to those registries are listed.

   **NOTE:**   When modifying registry keys it is important to know the system variables. You need to know what variables can be used that will be replaced with the correct path when an app is loaded on a machine. For more information, see: Predefined Virtual Folders.
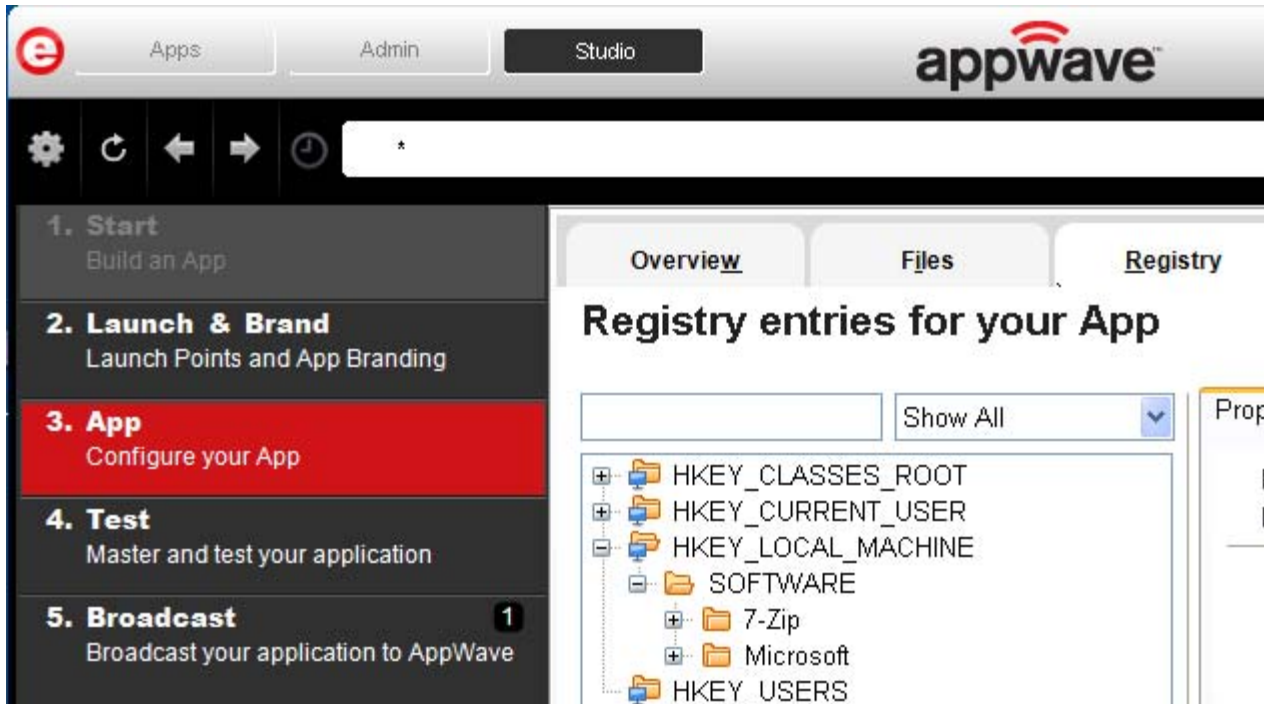
5   Click the Environment Variables tab and the **Environment variables for your app** appears. You can add, edit or remove any Environment variables on this page.

6   Click the Runtime tab to advance to the **Runtimes required by your App** page. A default list of runtime plug-ins are listed. You can select the ones required by your App.
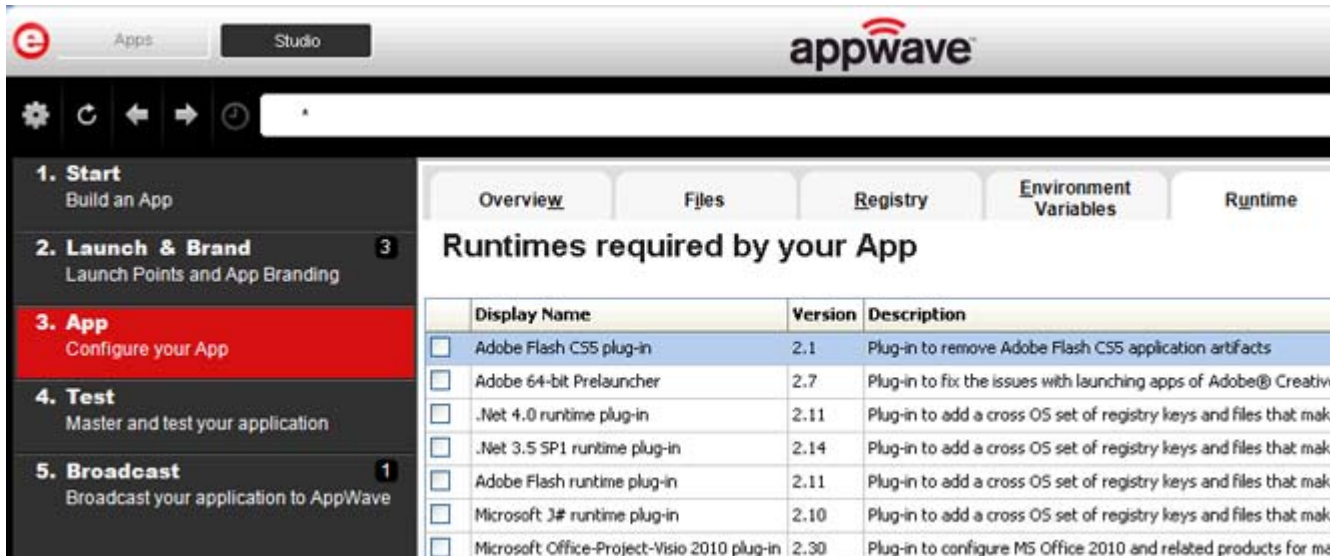
7  Click the Custom Scripts tab to advance to the **Custom Scripts for your App** page. Here you can you can attach scripts to key application triggers, such as activation, start, close, and deactivation that can accomplish whatever is necessary. These scripts are typically small executable files and batch files. Exe, bat, and cmd files are supported to perform acts such as:

- Establish a connection to a shared network location before the application starts

- Run a file backup/archive process for user created documents after the application has exited.

| Overview | Files | Registry | Environment Variables | Runtime | Custom Scripts |
|----------|-------|----------|----------------------|---------|----------------|

## Customs Scripts for your App

| Triggers | Name | | Enable | | |
|----------|------|--|--------|--|--|
| After Activation | None | | ☐ | Edit | Delete |
| After Load | None | | ☐ | Edit | Delete |
| After Launch | None | | ☐ | Edit | Delete |
| After Exit | None | | ☐ | Edit | Delete |
| Before Deactivation | None | | ☐ | Edit | Delete |
| Before Unload | None | | ☐ | Edit | Delete |
| Before Run | None | | ☐ | Edit | Delete |

## Trigger: After Activation

**Name**

**Frequency**

⊙ Fire the first time only        ○ Fire every time

**Script**

Filename:

Parameters:

☐ Use Local Application

**Event**

☐ Run outside of the App

☐ Run elevated (only supported on Windows Vista or later)

☐ Wait for the handler to exit

Time out after:  1    Minutes

Success code:  0    (Separate multiple codes with commas)

Any other values returned by the handler will be treated as errors

You can set different triggers and then define the actions for that trigger.

For more information on this step, see: <u>Using the App Task</u>.

## TEST

Click **Test** to master and test your app locally. This option allows you to master an app using the template .tca file that you just created.

A .tca file (also known as the Studio file) is a set of information that defines the resources, launch points, and registry settings required to run a product. It also contains information about the product such as the version, name, and editions. The .tca file needs to contain all the files that are associated with it. The .tca file has file system information that actually lists files for the program it needs to master. The files have to be physically present at the directory information that is saved in the .tca data.



1   The **Output Folder** is automatically populated with the path, i.e.,
    `C:<xxx>\<user>\Embarcadero\AppWaveStudio\Output\`.

2   The **File Name** is also automatically populated with the information from the **Launch &
    Brand** page.

3   In the **App** field if an app file exists for the app (.stp), click **Edit**. Select the App file (.stp) to
    associate with the .tca file for this App.

   **NOTE:**   This .stp file is typically located in the same output directory
             `(c:\<xxx>\<user>\Embarcadero\AppWaveStudio\Output\<app`
             `name.stp>)`.

4   Click the drop-down arrow in the **Compression** field and select the type you want to use. "Best Compression" is the recommended setting for <u>Compression</u>.

5   If you want to select the type of encryption to use in the App, click <u>Advanced</u>.

6   After your have verified your selections, click **Master (** 🞄 Master **)**. If you haven't saved your file, you will be asked to save your changes to the current working file.

7   The **Master** dialog appears with a progress bar showing the mastering steps. Click **Details** to view the mastering details.

8   Once the mastering is completed, an option to **Record the launch of the application for faster start-up time** appears in the Master dialog.

9   Click **Launch** to open the app and the **Recording** dialog opens once again.

10  Accept the End User License Agreement that appears and then once the application is open and the installation finishes, click **Finished** in the **Recording** dialog.

11  The **Master** dialog reopens. Notice that a prefetch file has been created.
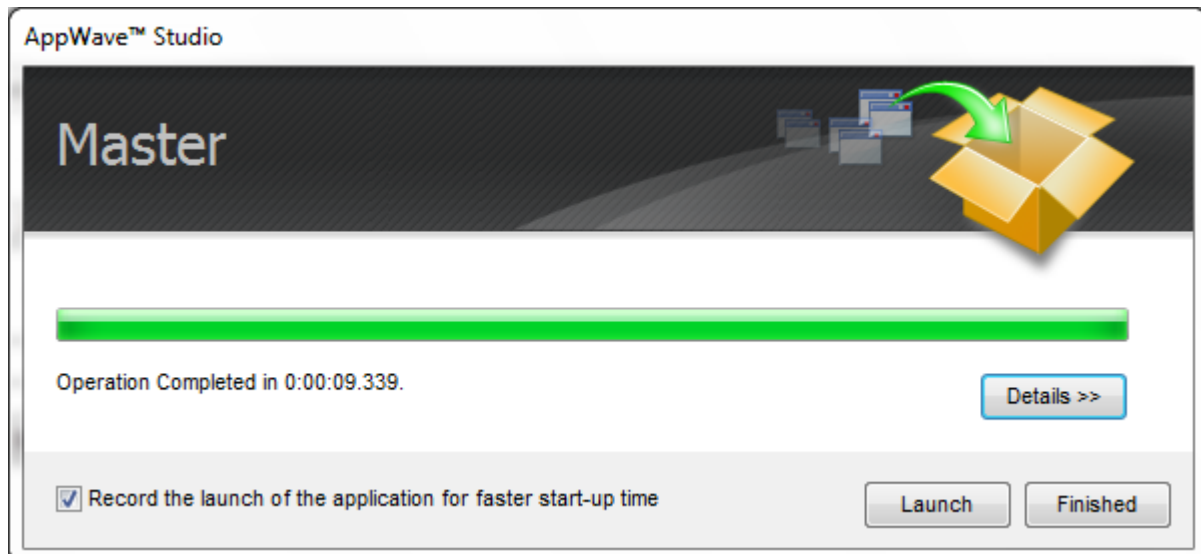


12  Click **Finished** in the **Master** dialog and then click **Next** to advance to the **Broadcast** step.

For more information on this step, see: <u>Using the Test Task</u>.

## BROADCAST

1   From the **Test** page, click **Next** to advance to the **Broadcast** page.

2   Click **Broadcast** and once again save the changes to your file.

3   Once the file is saved *the* **Broadcast** dialog appears.This dialog tracks the process of the broadcasting. Once you get the indication that the broadcast was successful, click **Finished** and the Broadcast dialog closes.
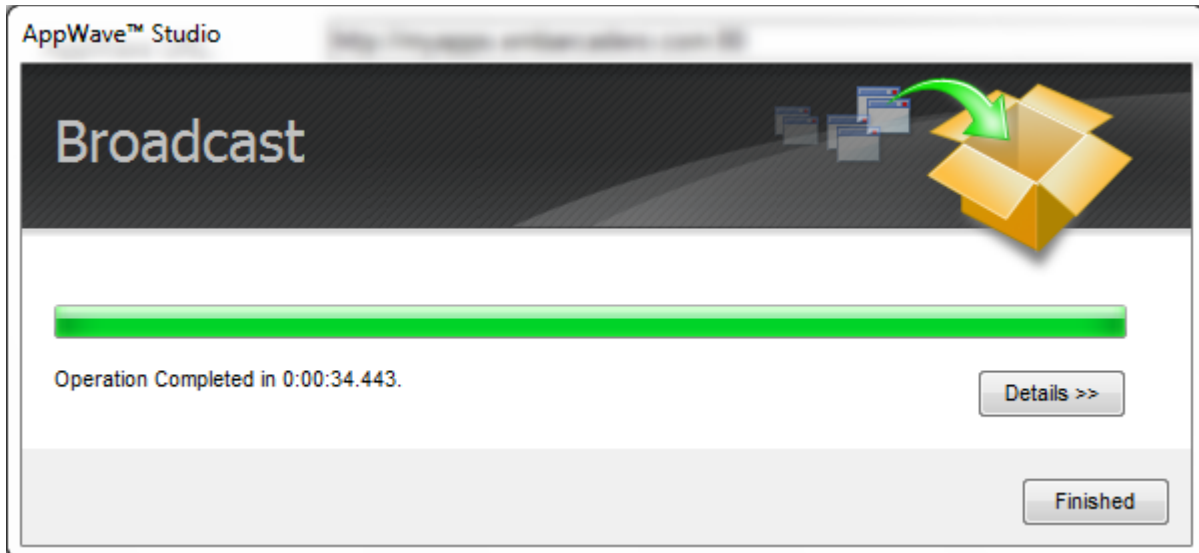


> NOTE:   A popup also appears in the lower right corner of your screen indicating that new apps are available.
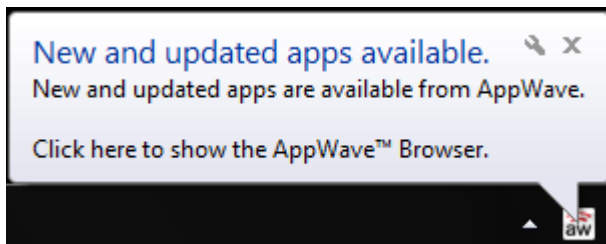


4   Click **Finished** and your app is now mastered with the prefetch information included.

> NOTE:   The files created have the file extensions of .aw, .app, and .stp and are placed in the C:\<xxx>\<user>\Embarcadero\AppWaveStudio\Output folder.

For more information, see: <u>Using the Broadcast Task</u>.

NOTE: If an .stp file already exists for the app you want to broadcast you get a message similar to the following message:
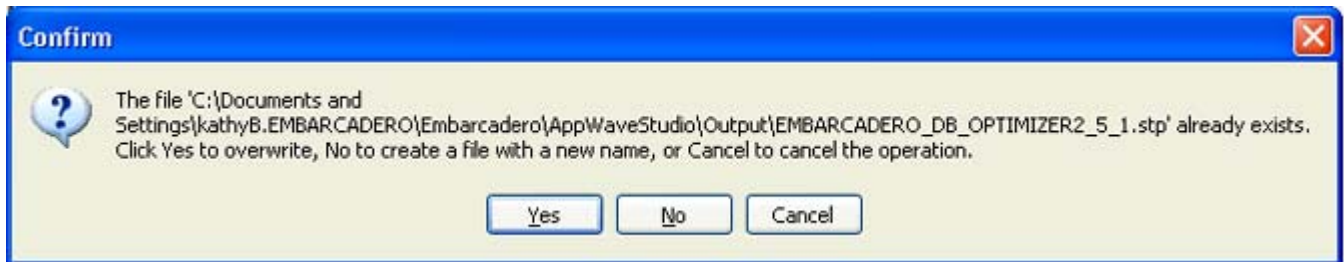


NOTE: If you try to broadcast an app that has the same name, version, and edition of an existing App, you get the following message:
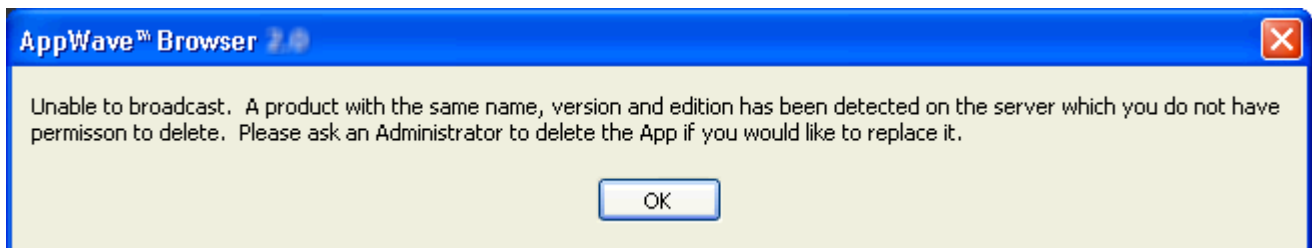


5 Once the file is saved, *t*he **Broadcast** dialog appears.This dialog tracks the process of the broadcasting. Once you get the indication that the broadcast was successful, click **Finished** and the Broadcast dialog closes.
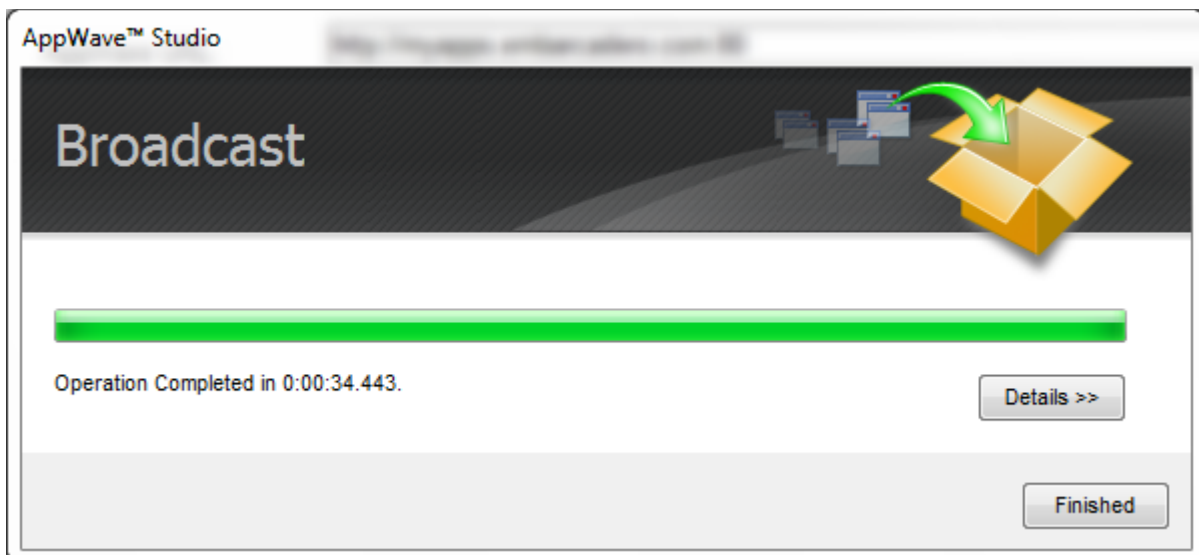
6   Click **Finished** and your app is now mastered with the prefetch information included.

> **NOTE:**   The files created have the file extensions of .aw, .app, and .stp and are placed in the C:\<xxx>/<user>\Embarcadero\AppWaveStudio\Output folder.

A popup also appears in the lower right corner of your screen indicating that new Apps are available.
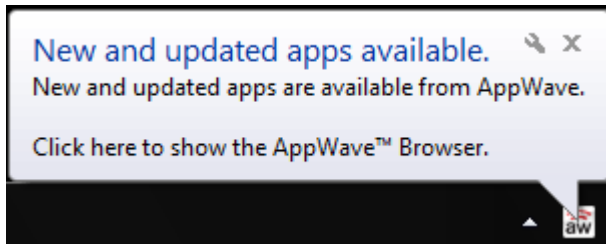


**CAUTION:**   During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio will attempt to shutdown any services it detects, but in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the App.

For more information on this step, see: <u>Using the Broadcast Task</u>

## USING THE APP

1   Once you have created the app using AppWave Studio, it is available on the AppWave Browser.

2   Click **Apps** and select the **AppWave** folder.

3   During the Broadcast process, if a folder doesn't already exist, a folder with your user name (for example "embarcadero.comkathyb's Folder") is created in the AppWave folder.

> **TIP:**   If your folder does not appear you may need to reconnect to the AppWave Browser by clicking  > **Connect**.
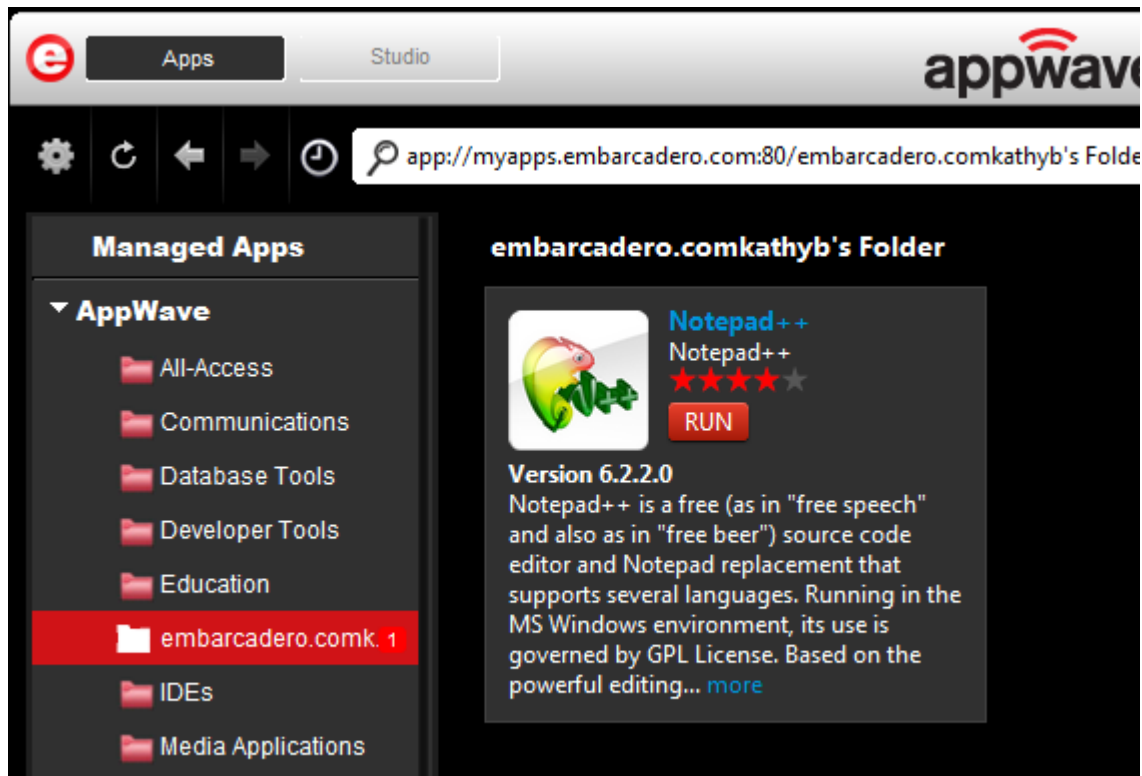
4   After your app is broadcast, the newly mastered app is placed in a working folder. The following screen shot shows one app has been placed in "embarcaderokathyb's Folder" designated by the ![1], as well as the app that was just mastered.

> **NOTE:**   The app runs a little faster if it has first been downloaded to your local disk.

5   Click **Add** to start the App.

> **NOTE:**    For more information on the Apps portion of the App, please see the *AppWave Browser Apps User Guide.*

## CREATE AN APP WITH A TEMPLATE THAT CONTAINS MULTIPLE VERSIONS OR MULTIPLE EDITIONS

When you master an app using an AppWave Template, that template may contain multiple versions (e.g. 1.0, 2.0, 3.0) or multiple editions (e.g. Professional, Personal, Developer) of the app. If that is the case, you have the option to select one of the versions listed, and then either open the app that is already installed or record the version listed.

1   In **Apps,** search ( 🔍 ) for "AppWave Template" and all the apps with that designation are

   listed. This template is indicated by an icon ( MASTER ) located to the right of the app icon.

2   Scroll down and select the app that you want to master. In the following example we selected "Mozilla Firefox" which has an AppWave Template with multiple versions.



3   Click **Master** located to the right of the icon and the **Record installation** dialog appears as the app file template downloads.

   A progress bar appears in the lower-right corner of Studio showing the download of the selected app.
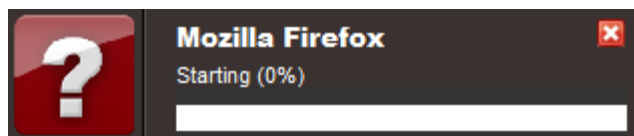


4   Also, an indicator window opens in the lower right-hand corner of your screen showing that the selected file has finished downloading. The file extension .aw indicates it is the AppWave Template file.
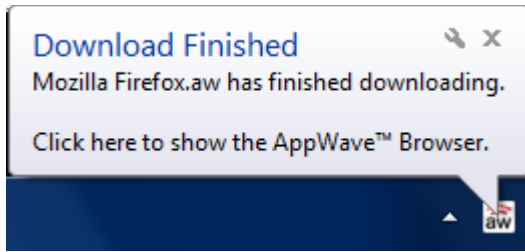
In Studio, the first step of the mastering process opens. However, since the app you want to master has different versions (e.g. 1.0, 2.0, 3.0) or multiple editions (e.g. Professional, Personal, Developer) included in the template, you are given the choice of which version you want to work with on the **Build an App** page. You can also choose to record a new one.
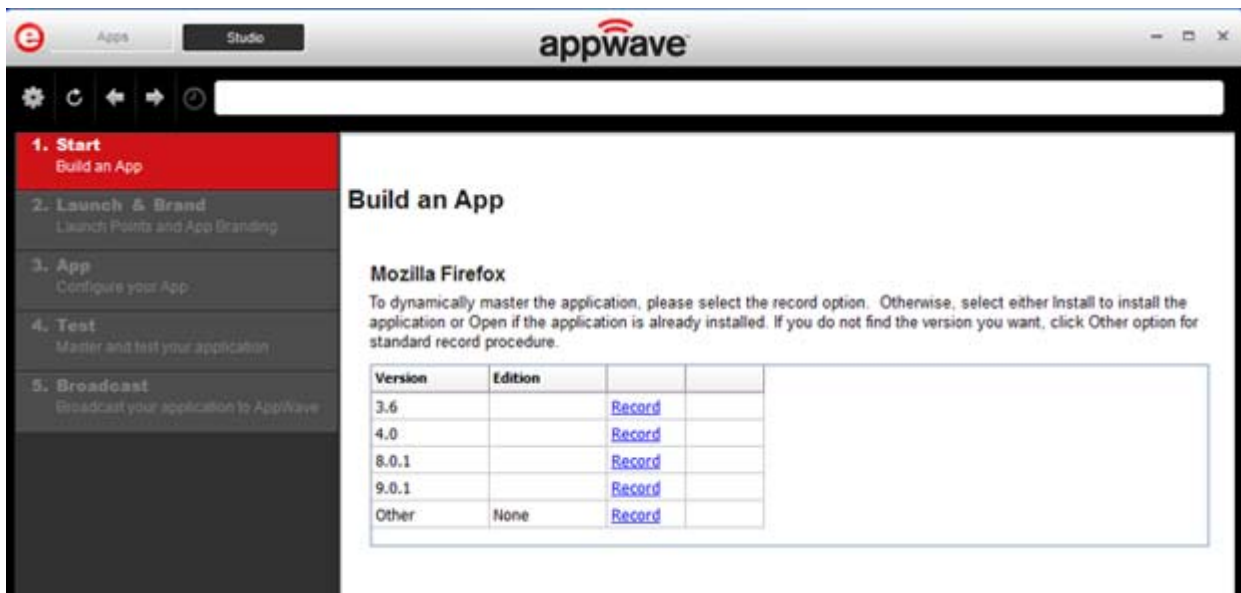
**Record Option**

1  Select a version or edition that you want to master.

2  Click **Record** and the **Record <product name>** dialog appears. Follow the steps to Create AppWave Supported or Custom App from Install.

> NOTE:  Click the web site URL if you have not downloaded the executable for the selected app you are going to master. This executable needs to be saved in the `C:\`directory.

If you select the Record link in the **Other** version, follow the steps to Create AppWave Supported or Custom App from Scratch.
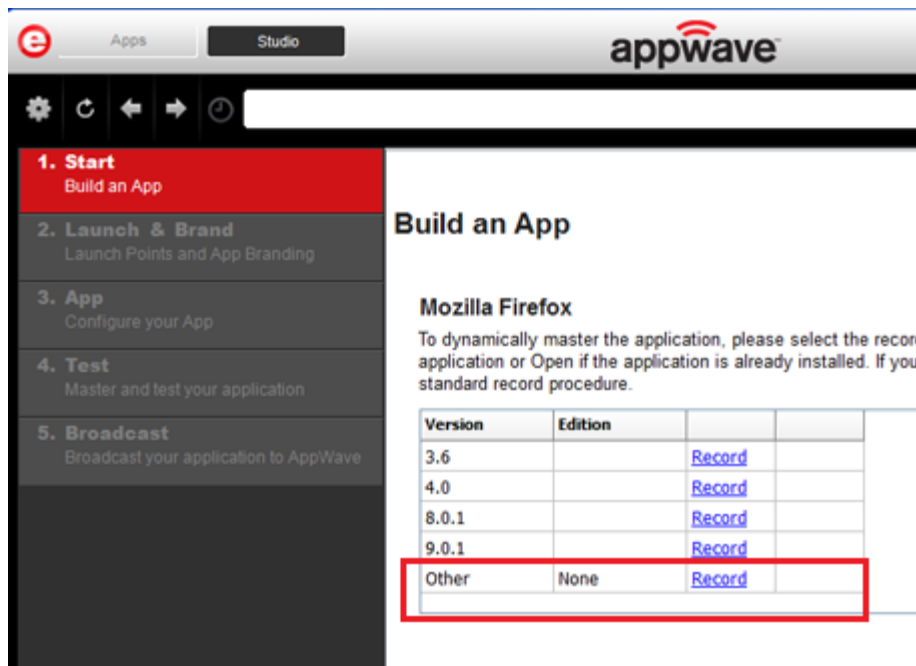


## RUNNING THE APP

Once you have completed the mastering processed and broadcast, the newly mastered app build is placed in a working folder. The app has been placed in "embarcaderokathybB's Folder" and the number next to your name (**1**) designates the number of apps in that folder.

> NOTE:  You can access details and reviews for this App, as well as write reviews.
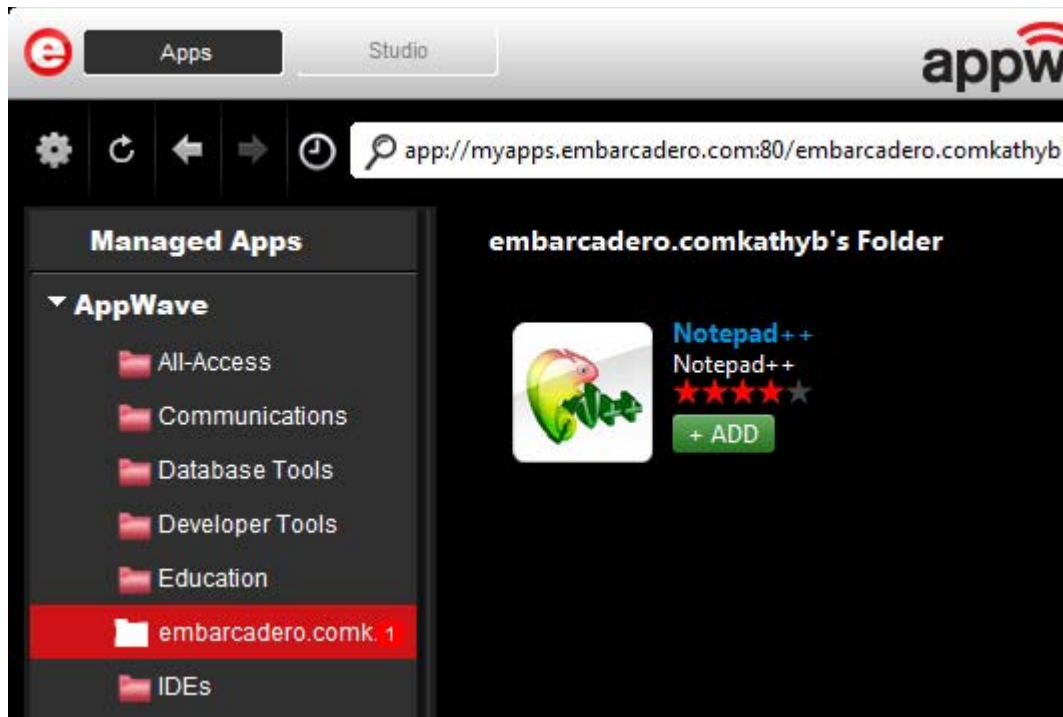
After your app is broadcast, the newly mastered app is placed in a working folder.

> **NOTE:** The app runs a little faster if it has first been downloaded to your local disk.

## ADD TO APP FROM INSTALL

Once you have created an app you can add other applications to it that will automatically be applied when the app is launched. This is useful when you have an application that has patches you want to apply to it or if you have an application that depends on other applications to run, such as TOAD and a database client. When the AppWave Browser launches this complex app, the complementary applications are also applied. The AppWave Browser user sees only one app, the main app in the AppWave catalog.

1  Create an app for the main application as described previously. and save your work or open an existing project to which you want to add other applications.

2  On the Studio tab, click the **Start** step and then click **Add to App from Install**. The **Record Application Installation** dialog appears.
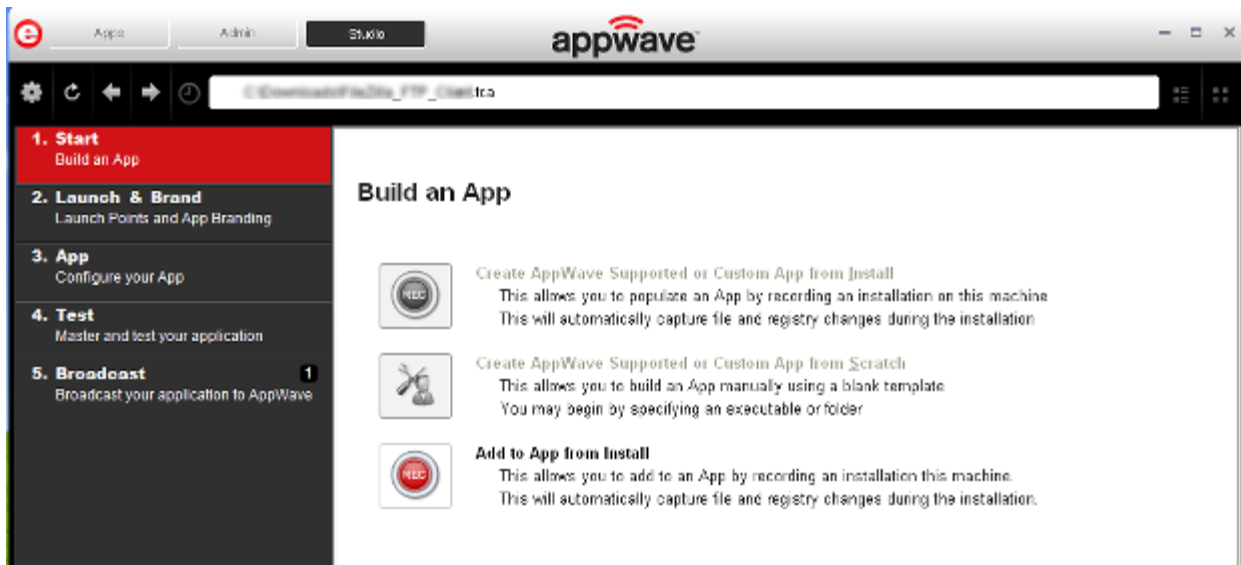
3   Enter the **Installer path** and any **Command line options** required and then click **Record**.

4   When the installation has completed, click **Finished** in the **Recording** dialog.

5   Continue to add applications to the project as required by repeating steps 2 through 4.

6   Complete any Tasks required and then save your project and broadcast the app.

## CREATE APPWAVE SUPPORTED WEBAPPS

This option allows you to create an app that is accessible via an internet browser. Mastering a WebApp does not require recording as for other apps, but you must define the required parameters.

1   After you have installed the AppWave Browser and have mastering permission from the Administrator, click **Studio** to open the Start page.

2   Click **Create AppWave Supported WebApp** or click  . The New Web App dialog briefly appears. This retrieves the PADE plugin information from the server. This plugin helps users master an app by automatically manipulating the file and registry entries.

> **NOTE:** You do not have launch the AppWave Browser as an Administrator in order to master this app.
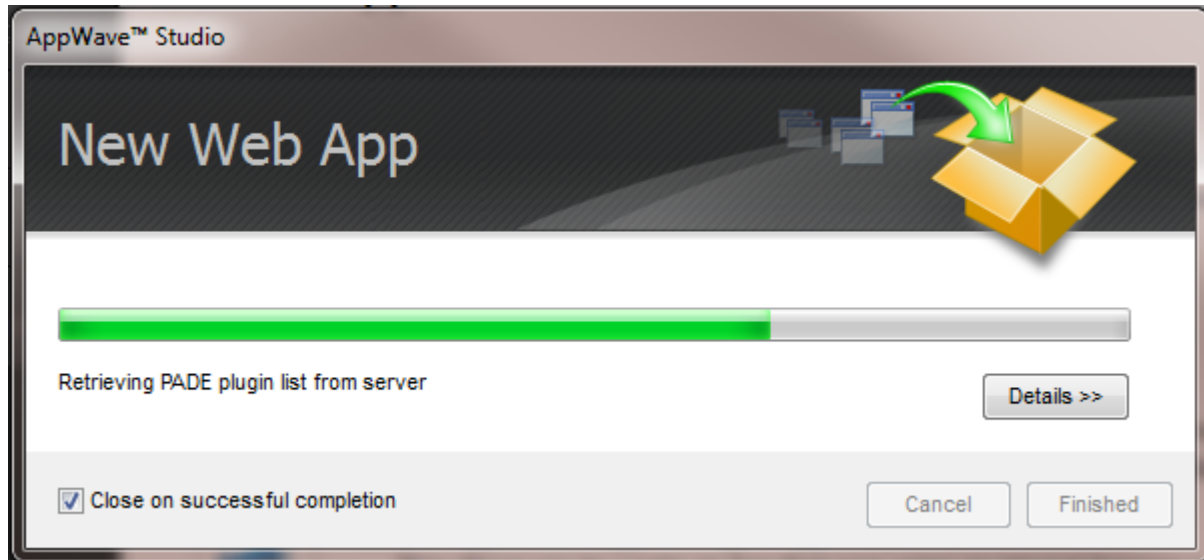


3   Once the initial dialog closes the **New WebApp** page opens (which is also referred to as the Start page). Enter the title of the URL. This is a required field.

4   Next select a URL. You have several selection options:

- Click Browse and a list of recent URL's open. You can select from System Favorites or System Recent URLs. Select one for the WebApp access point and click **OK**.

OR

- Click **File** and select a file from the dialog box.

OR

- Enter a URL

5   Under **Options** click the drop-down arrow and select the browser you want to use. This is a required field.

6   In the Browser Parameters you can define WebApp browser frame parameters. For example:

- the size of browser window on startup (with option to maximize on startup),

- show/hide the URL field (eg. use headless Chrome to "lock down" user from leaving the browser window, and

- show/hide back/forward navigation buttons).

7    Click Next and the Launch & Brand page opens.

## LAUNCH & BRAND

1    The WebApp URL field is automatically populated.

2    If the Preferred Web Browser field is not automatically populated, select one from the Browser dialog. This is a required field.

3    Click the drop-down in Product Name and select the name from the list. Once you have made this selection, the Display Name field is automatically populated.The Product name is a required field.

4    Click the drop-down in Product Version and select the appropriate version number. If nothing appears in the drop-box you can manually enter the version. This is a required field.

5    The **Publisher** is a required field, so enter *the name of the publisher for your WebApp.*

6    **Icon**: Icons must be created using this tool:   docs.embarcadero.com/products/appwave/ vendor/IconGenerator/uploadIcon.html. The input of the iconize tool is a 130 x 130 pixel graphic.The icon must be placed in the following folder and be in the .png format. C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.png.

7    **EULA**: The EULA file must be in the .htm format using the Arial font, size 10. It should be located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<app name>.htm

    **NOTE:**    If the application require prerequisites (like dotNet Framework, MSVC 2005/2008, J#, MDExplorer, Adobe Flash, etc) then the correspondent  EULA file should be included as well.

8    **Screen shot**: This is a screen shot of the main window of the application. It must in .png format and located in the following folder: C:\Documents and Settings\<user name>\Embarcadero\AppWaveStudio\Output\<screen name>.png.

    **NOTE:**    When creating a screen shot, please take into consideration the fact that they will be scaled to 228 x 174 pixels and they must not be distorted; Also, the images used must not be under copyright.

9    **Web Site**: Enter the URL for the applications web site.The maximum length is 512 characters.

10   **Support Contacts**: List the email first. If that is not available, list the contact page (or form) from the website. Or you can list the phone number.

    **NOTE:**    If the contact page contains one entry per country then the link to the page is enough.

11 **System Requirements**: If no particular requirements are needed then it must contain the phrase 'Windows XP and higher". If the application has Pre-requisites then this field is mandatory. The maximum size of this field is 1024 characters.

In case the app requires pre-requisites, this info must be split into two sections:

- Requirements for running the app: List all the pre-requisites that the app needs to run that Embarcadero does not offer runtime plugins for. An example would be Adobe AIR.

- Requirements of the native application: List everything the native application needs even if a runtime plug is available. Examples would be: .NET 3.5 SP1 runtime, Adobe AIR runtime, Microsoft Visual C++ 2008 SPI runtime, and Windows installer 3.0.

12 **Product Description**: Should not consist of more than two sentences, although there is a 5000 character limit. Include keywords a user might search on such as game, drawing, database, application development, image editor.

For example, the product description for Photographics is: "Photographics is an image editor that provides powerful tools for designers and developers. The information here cannot be copyrighted."

13 **Long Description**: The long description cannot contain part of the Product Description and vice versa. For commercial apps, if the description is taken from the vendor's website, a copyright phrase must be added at the end of this description. For example:  "Copyright © 2011 Adobe Systems Incorporated. All rights reserved." This maximum size of this field is 5,000 characters.

14 **Release Date**: This is the date when the application was released, not the date when the app was mastered or published on AppWave Store.

> NOTE: The Tasks pane located at the bottom of the page lists fields that require information. You need to select the product name and version number of the installed application. Click the drop-down and select *your WebApp name*. Once that is selected the Display Name field is also populated.

15 Click Next and the App page opens.

## APP

The App page is where you configure your Web App.
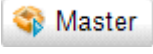
1 Click WebApps or  and the WebApp tab opens. The information that appears here is what you entered on the Start or New WebApps page.

> NOTE: Once you have clicked Next on the Start page, you cannot go back and make changes. Any changes must be made on the WebApps tab

2 Make any necessary changes and click **Next** to advanced to the Test page.

## TEST

1   When the Master and Test your **App** page opens notice that the output folder and file name fields are automatically populated.

2   Click ![Master] and a confirm dialog appears asking if you would like to save your changes.

3   Click **Yes** and a **Save As** dialog appears. Once the file is saved, the **Master** dialog opens and a progress bar tracks the process of mastering. It also displays the time it takes to complete the operation.

> **NOTE:**   The file extension is .tca.

4   Click **Finished** to close the dialog and then click **Next** to advance to the Broadcast page.

> **NOTE:**   You can also choose to launch the mastered application at this point. This is to do a test launch, not to record prefetch.

## BROADCAST

1   On the **Broadcast your application to the AppWave** page leave the AppWave URL default setting.

2   Also, leave the default setting of "Always remaster projects before broadcasting to the AppWave" and click **Broadcast**. The Broadcast dialog appears with a progress bar tracking the broadcasting steps.

3   Click **Finish** and a popup appears.
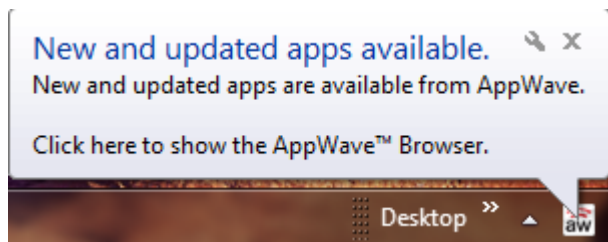


## USING THE APP

1   Once you have mastered the app using AppWave Studio, it is available on the Apps tab.

2   Click **Apps** to open AppWave Browser. Then click **Notifications** and notice that your WebApp is now available**.**
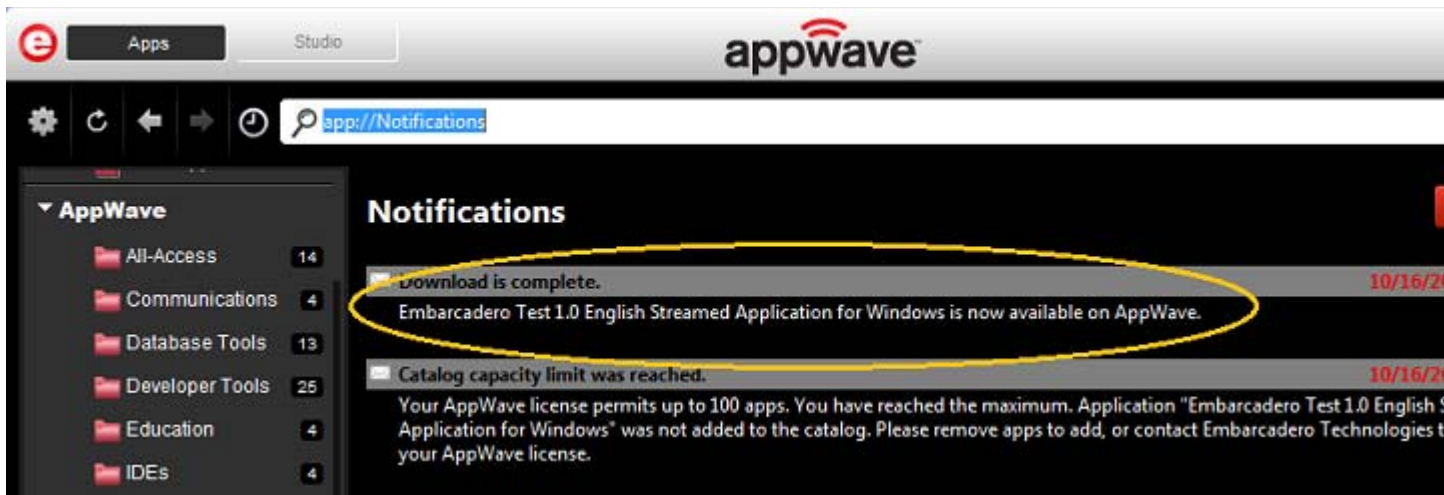
3   Then click AppWave and click your folder to view your WebApp.
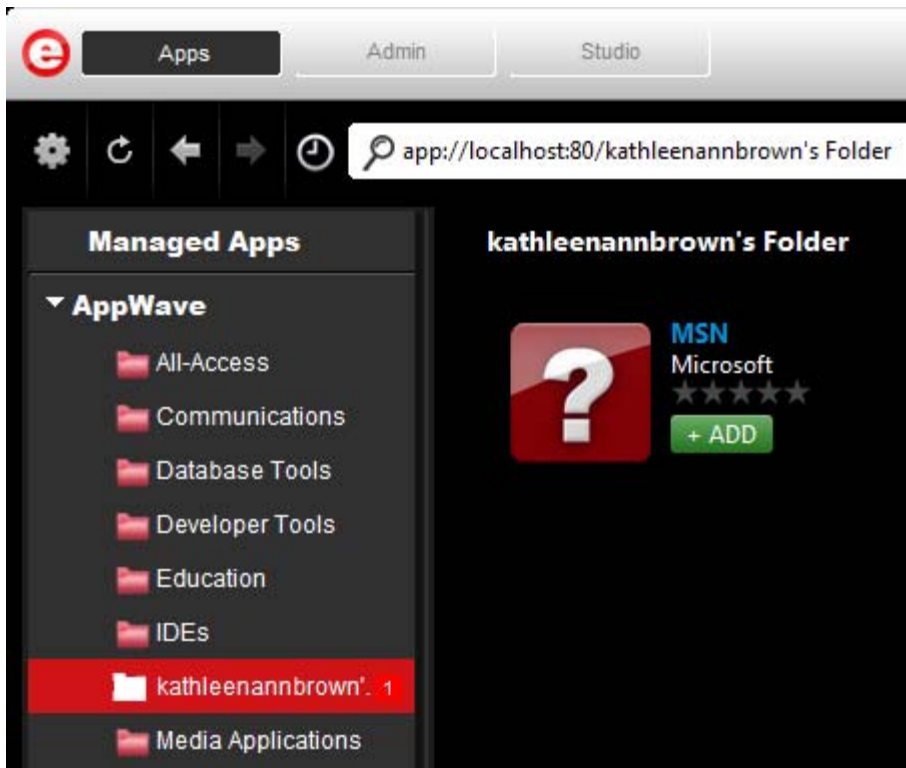


4   Click `+ ADD` and a pop-up opens in the AppWave Browser.
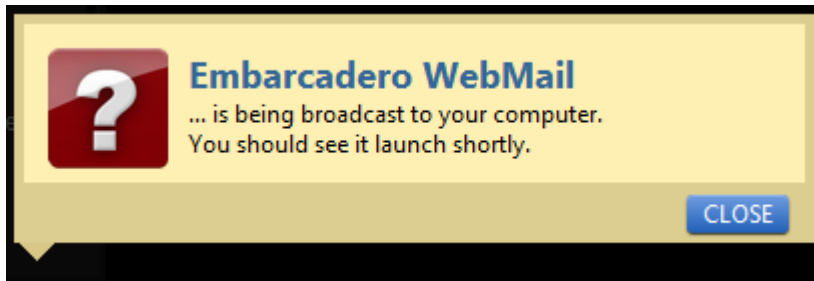
5 Once the pop-up appears the  icon changes to .

6 When the download is complete, click  and the WebApp opens on your desktop.

# ADD NATIVE INSTALLER TO APPWAVE CATALOG

This allows you to add a Native Installer to the AppWave Catalog with the required metadata. This function delivers software that cannot be easily mastered. Some examples of this are: Native 64-bit, printer drivers, and software establishing Windows services.

It can also distribute data and media content packaged up as installers. Examples of this type of content would be: data models, training materials and manuals, and data sets.



1 On the Start page select **Add Native Installer to AppWave Catalog** or click . The **Select Folders and Installer File to Include** opens.

2 Enter an initial folder name and then browse to the installer launch executable. The install command is an executable string that installs the application.

   NOTE: The "Command Line" value indicates how to launch the file and the value is the trigger to run a specific command.

   NOTE: The un-install key is entered later in the mastering process.
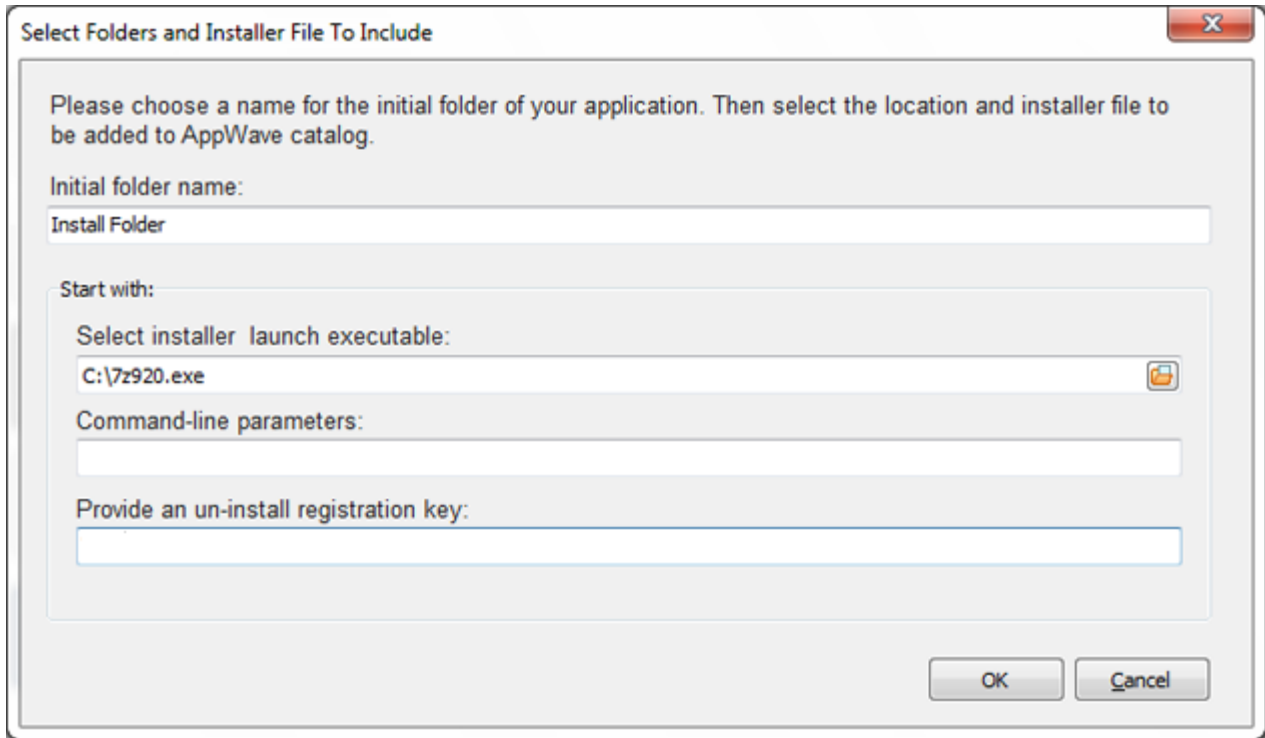
3   **Click OK** and the **Create Installer** dialog opens. Click **Finished** once the creation process is completed.
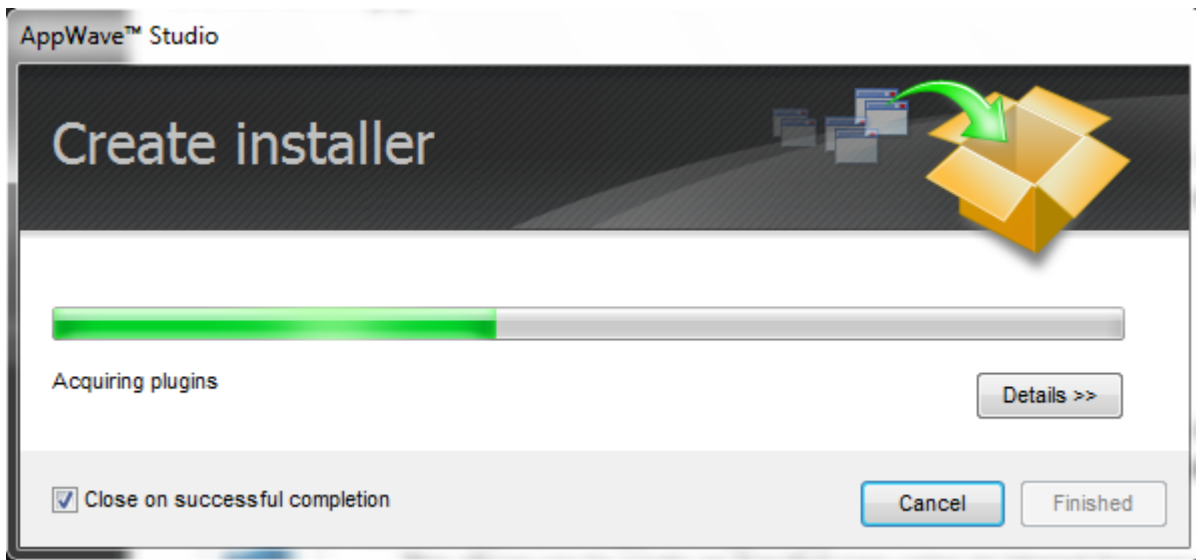
4   The **Launch & Brand** page opens. There may be  a number of Tasks that must be completed. Each required field is marked with a warning icon ( ⚠ ). The following fields are required:

  • Launch Executable

  • Working Folder

  • Product Name

  • Product Version

  • Publisher

5   Click **Next** and then click the Installer tab on the **Configure you App** page.
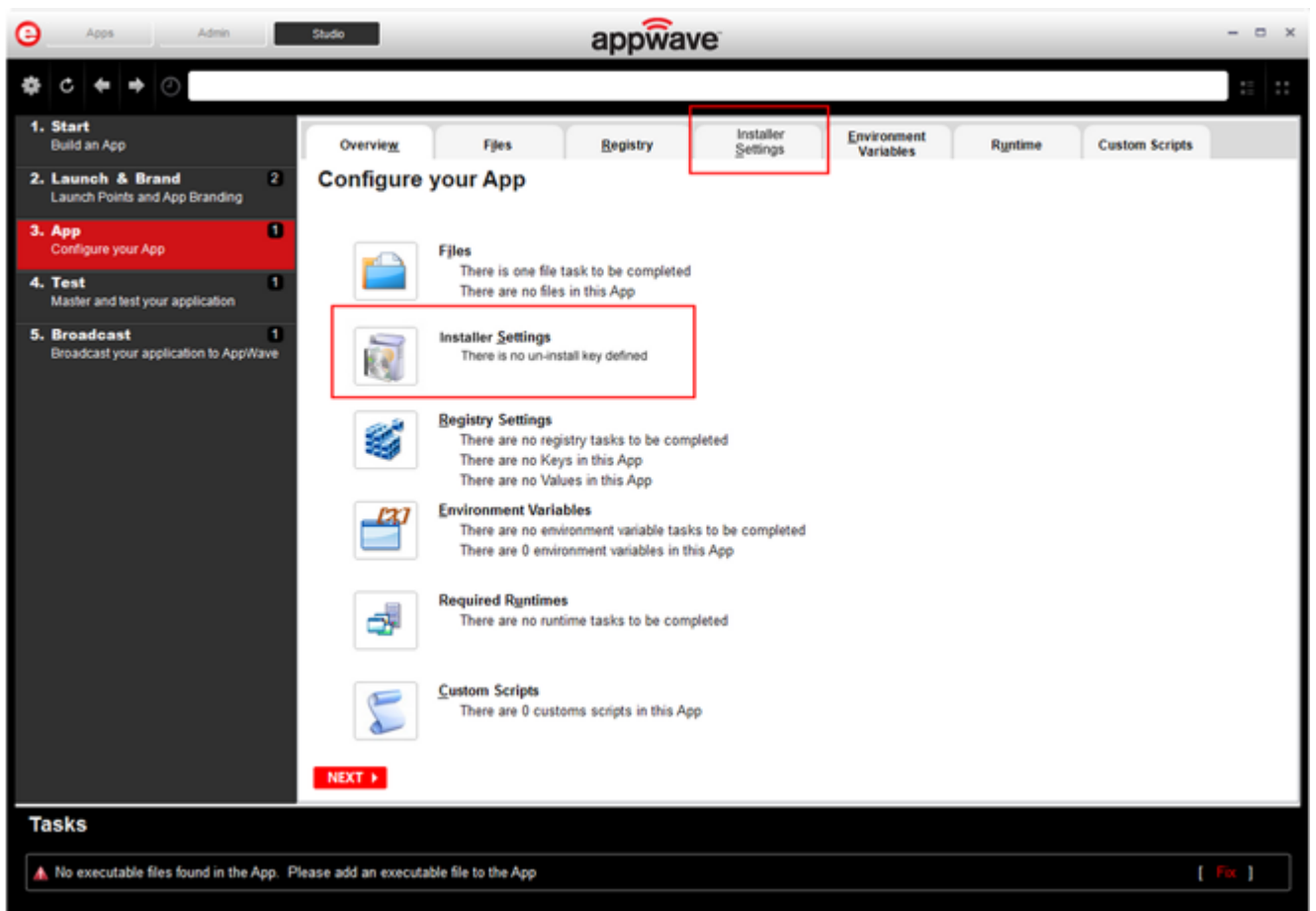


6   The **Select installer launch executable** field is automatically populated.

7   Provide an un-install registration key. This key identifies the registry location that contains the application's uninstall information. This is a required field.

8   Select **Test** to open the **Master and Test your App** page. The Output folder is already populated.

9   Click ![Master] and save the changes to your current working file. Note that the file extension is .tca. Once the file is saved the **Master** dialog opens. This is where the operation masters the installer.

10    Click **Finish** and then advance to the **Broadcast** page.

11   Click **Broadcast** and the Broadcast dialog opens. Once the broadcasting is complete a notification appears in the bottom right corner of the screen. This lets you know that your app is available.

> **NOTE:**    A task appears asking you to do the Prefetch function. You do not have to do the "Prefetch" task when creating an installer.

12   Click the **Apps** tab and note that the installed app is placed in your folder.
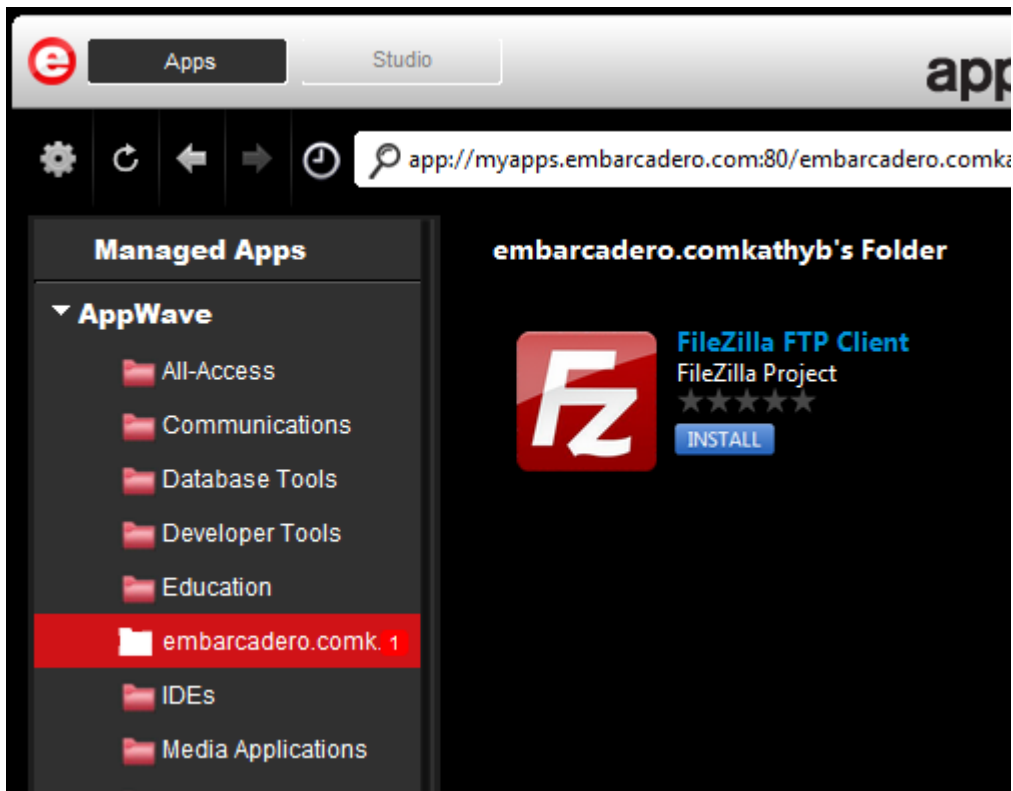


13   Click **Install** and when the installation finishes check that program is registered (Control Panel\Programs and Features) and mark the app as Installed.

14   Click **run** which then execute the entry point target.

# OPEN PROJECT

When you created an App, all the project information is stored in a .tca file. You can revisit this project file and modify the project as needed. Each step in the project is then available for viewing and editing. This option is only available through the AppWave Studio.

For more information see: [Group Management](#).

1   Open the AppWave Browser and click Studio.

2   Click ![gear icon] and select **Open Project**.
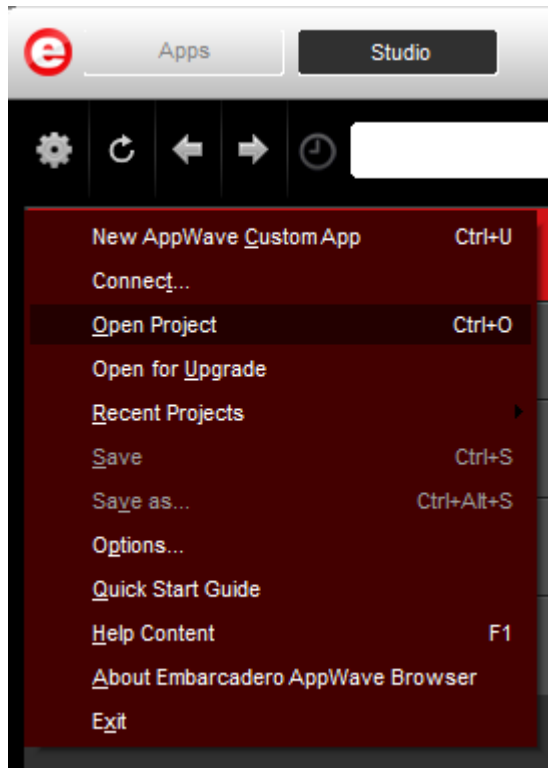


3   In the **Open** dialog, select a Studio file (.tca) and click **Open**. The **Open File** dialog opens.
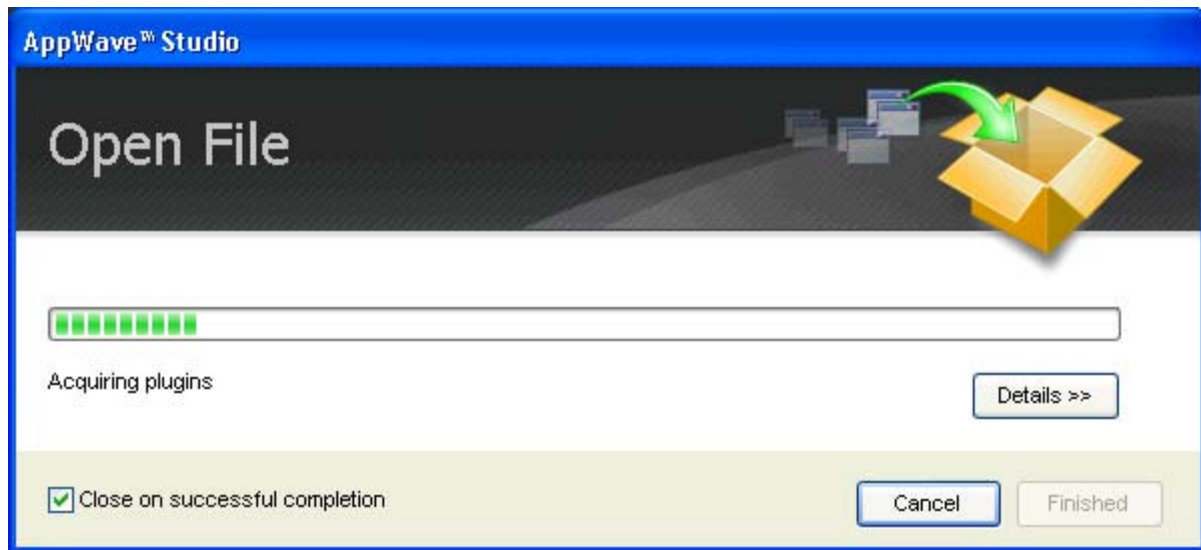
4   Once the file is loaded, the **Broadcast** page opens in Studio.

    NOTE:   The page that opens depends on the app you are working with. If any
information is missing or needs correction Studio opens to the appropriate
page. For example, if the .tca file is missing a Project Version number, it opens
to the Launch & Brand page.

5   You can modify any of the settings you made when you first mastered the application. You
can made changes on the following pages: Launch & Brand, Configure your App, Master
and Test Your App, and Broadcast your application to the AppWave.

6   Click **Broadcast**: The "Always remaster projects before broadcasting to AppWave" option is
selected by default. Best practice is to always remaster a project before broadcasting.

7   Click **Broadcast** and the revised app is made available in AppWave.

## OPEN FOR UPGRADE

This command, available under the Tools and Options menu () takes an existing .app file
and extracts it onto your machine. Once extracted, you can make changes and then remaster it.
It is a very powerful tool when you want to apply a service pack or add some files to an existing
mastered app.

1   Open the AppWave Browser and click **Studio**.

2   Click  > **Open for Upgrade**.

    NOTE:   When you master your apps, the .app file is stored in the c:/xxx/*Embarcadero/
AppWaveStudio/Output* folder.

3   Select the file you want to upgrade and click **Open**. The **Open app for reconstruct** dialog appears. You can view the steps of this process by clicking **Details**.
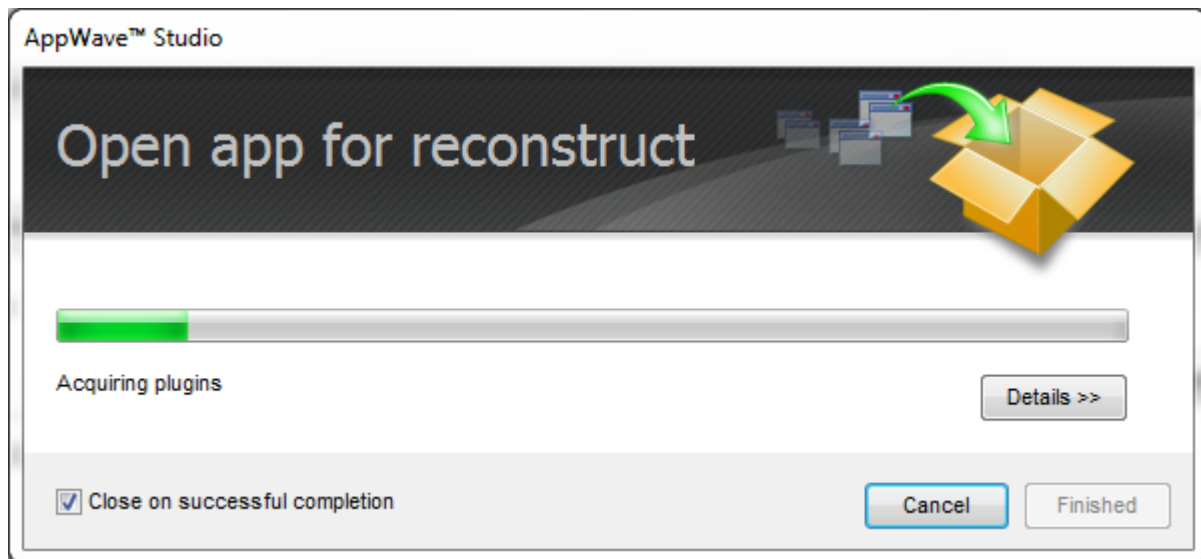


4   Once all the data is processed the dialog box automatically closes and the Broadcast page opens.

5   You can modify any of the settings you made when you first mastered the application. You can made changes on the following pages: Launch & Brand, Configure your App, Master and Test Your App, and Broadcast your application to the AppWave.

6   Once you have made the desired changes, click **Broadcast** on the Broadcast page. The app is remastered with your changes or upgrades.

7   Click **Finished**. Your revised app is made available in the AppWave Browser under your personal folder.

# APPWAVE STUDIO OVERVIEW

The AppWave Studio application is integrated with the AppWave as a feature of the local library and appears in the AppWave Browser as the Studio tab.

The AppWave Studio is used to create an app file for an application. This application can use a mastering template or be created from scratch.

An application can be mastered into a file which can be run without installing it on your computer. This application runs on the desktop, locally but does not change the operating system or registry. It also does not share DLLs or other system-wide settings. The application data, including settings and licenses, are stored locally on the hard drive separate from the executable. The licenses are managed by the Administrator and are administered through Embarcadero's Sanctuary licensing system.

> **NOTE:** Once an app is mastered an "AppWave Template" is automatically created. This template file (.aw) contains all the information necessary to master an app such as production name, version, EULA, etc. It is placed in the output directory (`C:\xxx\<user>\Embarcadero\AppWaveStudio\Output\`).

There are three main ways to master an App: Create an App with One Click, Create App from Install and Create AppWave Supported or Custom App from Scratch. Once your app has been mastered and then broadcast it can then be run on the local machine and be launched from either the local hard drive or the network using the AppWave. You can create and manage your tools in your Local Library and then share them with your organization or with the world.

For complete information about installing the AppWave Browser, see the *AppWave Administrator Guide*.

This chapter gives you an overview and explanation of all the following:

- AppWave Studio Elements.
- Window Control Menu
- Toolbar
- Tools and Options Menu
- Connect
- Options
- Applications within AppWave Browser
- Tasks
- Steps
- Master
- AppWave Studio Error Dialog

For information on how to use AppWave Studio, see: Using AppWave Studio.

# APPWAVE STUDIO ELEMENTS



**Configuration Menu**

**Window Control**

**Apps Tab**

**Studio Tab**

**Search Window/ Displays the .app file**

**You can add another application to a previously mastered app**

**Master installers and add to the catalog**

**Master an web-based app to run on our local system**

**Two main avenues for creating apps**
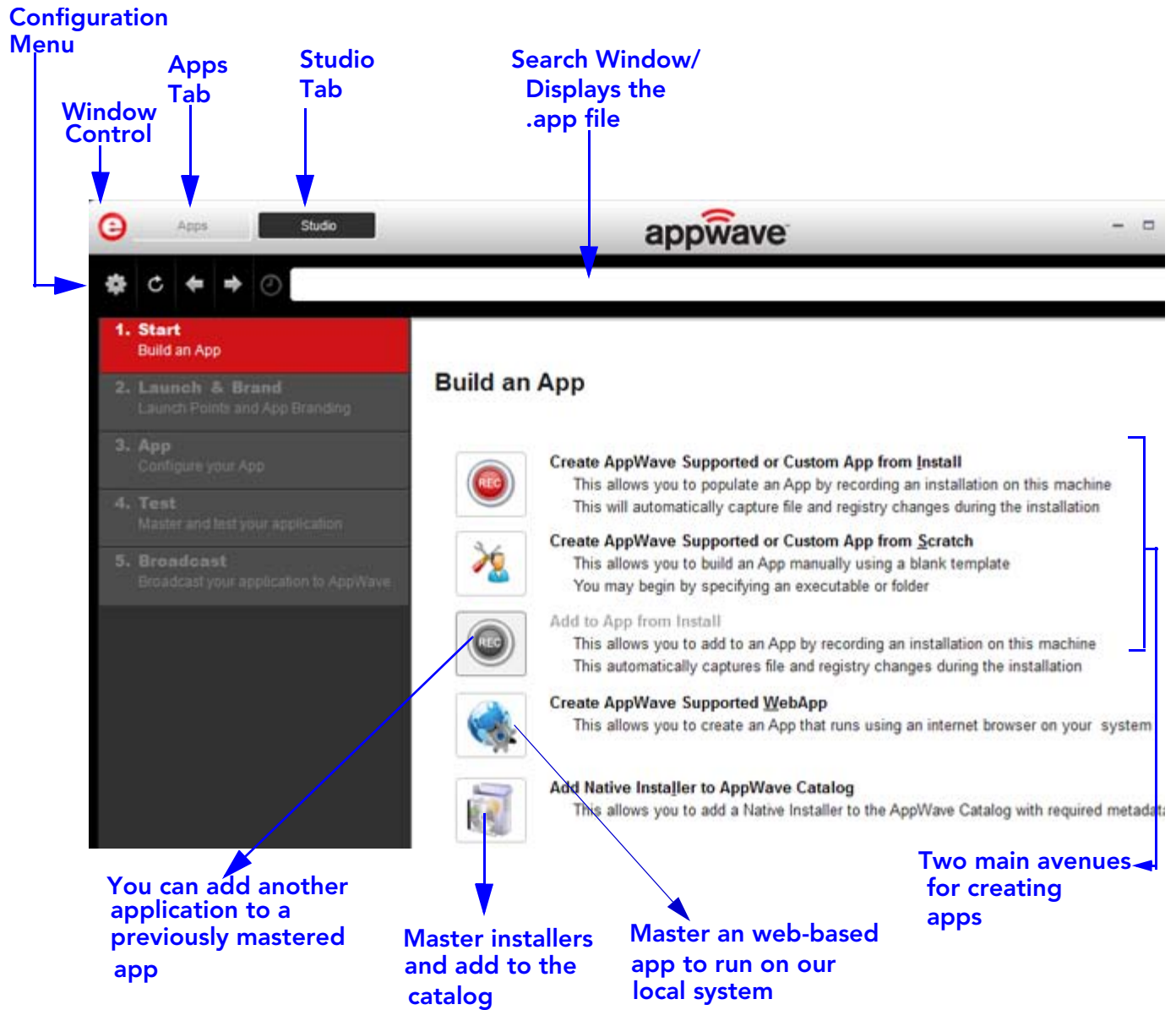
## WINDOW CONTROL MENU

Click the Embarcadero "E" icon () to manipulate the window size of the AppWave Browser.

| Command | Description |
|---|---|
| Restore | Restores the default window size. |
| Minimize | The AppWave Browser window is minimized and now appears on Windows task bar. |
| Maximize | Enlarges the AppWave Browser to full screen. |
| Close | Select Close and the application continues to run in the background. The **Exit** command closes the application. |

**NOTE:** Double-click the window's title bar to maximize it or restore it to its previous size.

# TOOLBAR

The following functions are available on the Studio Tools and Options command:

| Function | Icon | Description |
|---|---|---|
| Tools and Options Menu |  | Provides access to menu options such as connect, opening an existing project, saving your project or opening the online help. |
| Refesh Tool |  | Upates the display. |
| Back |  | Once you have accessed a step and then moved forward, you can move back to that step. |
| Forward |  | Once you have accessed a step and then moved backward, you can move forward to that step. |
| Path | C:\PDF_XChange_Viewer.tca* | Displays the path, and the .tca file name of the application you are currently working with.<br><br>It is also the search window. |

**NOTE:** The other items on the toolbar not listed in this table are used only by the Apps portion of the application.

## TOOLS AND OPTIONS MENU

Click the Tools and Option icon (    ) to access the menu options. The following table explains each menu item:

| Command | Description |
| --- | --- |
| New AppWave Custom App | Takes you to the Build an App page where you can start to create an app from an install executable or from scratch. |
| Connect | Opens the Connect dialog where you can change the connect details using different user credentials or change the URL to connect to a different AppWave. |
| Open Project | Opens a dialog where you can browse to and select a .tca file (Studio Designer File) for an existing project. |
| Open for Upgrade | You can revisit a project file and modify the project as needed by opening the .tca file in Studio. |
| Recent Projects | A pull-right menu displays the most recent projects you have worked on. Select a project and the Open File dialog appears. Once the file has loaded click Finished. |
| Save | Saves the .tca file for the project currently open in AppWave Studio. |
| Save As | Rename the current .tca file in the Save As dialog. |
| Options | Opens the Options dialog where you can set AppWave Browser defaults. |
| Quick Start Guide | Opens the online help file for the AppWave Quick Start Guide. |
| Help Content | Opens the online help file for AppWave Studio. You can also press F1. |
| About Embarcadero AppWave Browser | Displays the current version and build number. |
| Exit | Closes the AppWave Browser application.<br><br>**Note**: You do not get a warning dialog before the application closes. |

## CONNECT

When you click **Connect** in the menu the following dialog appears:



| Options | Description |
| --- | --- |
| AppWave | This is the IP address that you want to connect to. The default setting is displayed. |
| | You can clear the history from this field. |
| Username | Your user name for the AppWave account you are using. |
| Windows Authentication | Your Windows credentials are used to log on. |
| | You need LDAP in order to connect to AppWave using Windows Authentication. Please refer to *Import LDAP Users and Groups* in the *AppWave Administrator Guide*. |
| Password | Your password for the AppWave account you are using. |
| Remember me | Select this option if you want the Browser to remember your password. |

| Options | Description |
|---|---|
| Login | The Loading Data symbol appears in the pane as you are connected. |
| Cancel | Cancels all selections and closes the dialog without taking any action. |
| Clear History | Clears the history in the APpWave Connection dialog. A warning dialog box opens asking you to confirm your action. |

## OPTIONS

Click the Tools and Options icon (    ) and select **Options** to open the Options dialog.

| Option | Description |
|---|---|
| Launch at Startup | Select this option to always have the AppWave Browser launch when your computer starts up. |
| Hide When Minimized | Select this option and when you minimize the AppWave Browser, the icon shows in the notification area but not on the task bar. |
| Do not check for Browser updates | Select this option if you no not want the Browser to automatically check for the latest updates. |
| Cache Folder | This is where the streamed apps are stored. The default folder is: c:\xxx\<user>\Embar-cadero\Product Repository. You can browse to a new folder and select OK to set it as the default. This can be either a local or a network directory. If you change the cache location you must wait until the objects in the current cache location are copied to the new location. |
| App Cache Size | The default cache size is 512 MB. The maximum size shown is the size which is accepted by the machine and the amount of free disk space available on the machine.<br><br>Click **Clear** to clear the cache. A dialog appears telling you when the cache is cleared successfully. When you clear the cache you remove all streamed apps from your local repository. |
| Autogrow | When streaming, if the cache cannot accommodate the size of the full app, the cache can automatically grow to accommodate the new size. The Autogrow default setting is "On". If the Autogrow setting is not selected you will be asked, via the Browser, to increase the size of the cache to meet the size requirements. |

| Option | Description |
|---|---|
| Turbo Option | This is a performance option allowing the AppWave Browser to run more efficiently. If the user doesn't have administrator rights at the time of install, the AppWave Browser installs without the Turbo Option.<br><br>If the Turbo Option is enabled, when you run AppWave Browser, a file system driver is installed which allows AppWave to communicate with your hardware and operating system at a much lower level.<br><br>The AppWave Browser and the AppWave applications you run while Turbo is enabled run faster because of the installation of the file system driver. By default, AppWave Browser runs in this mode if your Windows account has administrative privileges and if you are running AppWave Browser on a 32-bit machine.<br><br>When Turbo is not enabled, AppWave Browser does not require Administrative privileges. This is a zero-install mode for locked-down desktops where your operating system user permissions do not allow you to install drivers. Performance is only slightly less than that provided in the Turbo mode.<br><br>Mastering is not supported when Turbo mode has been disabled. |
| AppWave Name | Displays the name that appears on the Apps tab. The default is the user's name followed by 'Apps'. For example, 'Joe's Apps'. You can edit this name. |
| Save | Saves the changes to your Options settings. |
| Cancel | Discards all changes made to the default settings and closes the Options dialog. |
| Proxy Settings | Click the link to open the Internet Properties dialog where you can describe the proxy server you use to connect to the Internet. AppWave Browser needs to connect to the Internet to access AppWave. |

## APPLICATIONS WITHIN APPWAVE BROWSER

**Apps Tab:** Opens the AppWave Browser application where you can view all groups and the products in each group. You can also view AppWave Store apps, and Notifications. For complete information on the Apps application, see the *AppWave Browser Apps User Guide.*
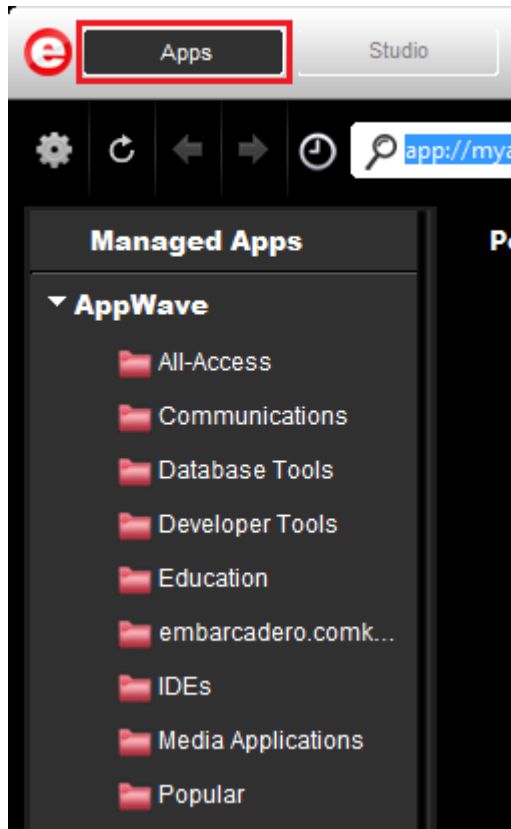


**Studio Tab**: Opens the AppWave Studio application where you can master apps. The steps to create or modify an app are: Start, Launch & Brand, App, Test, and Broadcast.
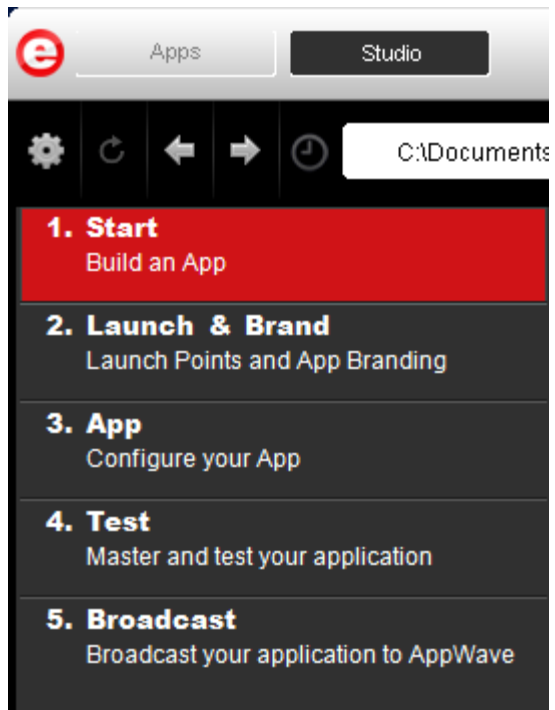
## TASKS

The number icon located to the left of a step indicates the number of tasks that need to be completed before you can advance to the next step. In the screen shot below, the Broadcast step has one task to be completed.
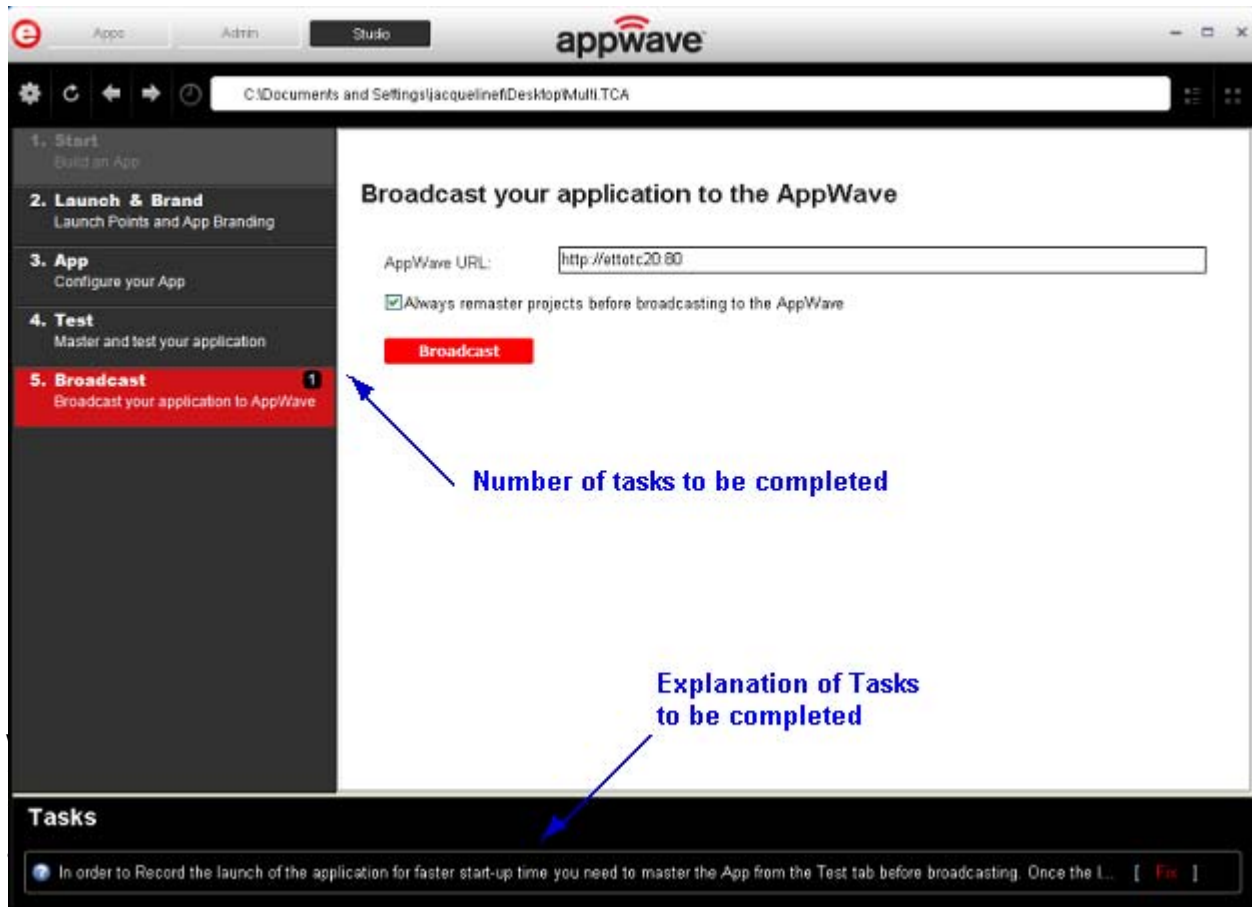
## STEPS

AppWave Studio uses five different steps to create and broadcast your tool. Each step contains elements that must be performed in order to create an App. Many of these steps may be automatically completed during the creation and mastering of your App. The type of application you start with (i.e., from a template, an install, or from scratch) determines the level of detail you must complete.

The five steps in creating an app are defined below.

| Label | Description |
|---|---|
| Using the Start Step | There are five different ways to build a new App:<br><br>• Create from install<br><br>• Create from scratch<br><br>• Create appwave supported WebApp<br><br>• Add to an app from install<br><br>• Add a native installer to the appwave catalog.<br><br>You can also add to an existing App. Each option has different operations that must be completed.<br><br>**Note**: Another option is to Create an App with One Click. This option is initiated in the Apps tab. |
| Using the Launch & Brand Step | The Launch & Brand step is where you select the launch executable and brand the app with product information.<br><br>The working folder is populated, and you can set any parameters needed for the application and select an icon to use. The product information entered here is used by the AppWave Host to identify it.<br><br>This information includes name, version, edition, and publisher. It also includes web site information, system requirements, and product description. You can define multiple entry points for application suites that have more than one application. |

| Label | Description |
|---|---|
| Using the App Step | You can configure the following app elements:<br><br>**Files**: Add, remove, or organize files and folders. You can also change the properties of each file as required.<br><br>**Registry**: Add or create new registry entries. You can also view and change attributes.<br><br>**Environment Variables**: Add and/or edit environment variables.<br><br>**Runtimes**: When a product (for example the .NET framework required by RADStudio) needs an execution runtime, it is provided to those who master applications. Once you make a selection you are requested to save your changes to the current working file in order to analyze required runtime changes.<br><br>**Custom Scripts**: You can attach scripts to key application triggers, such as activation, start, close, and deactivation that can accomplish whatever is necessary. These scripts are typically small executable files and batch files. Exe, bat, and cmd files are supported to perform acts such as<br><br>• Establish a connection to a shared network location before the application starts<br><br>• Run a file backup/archive process for user created documents after the application has exited.<br><br>Two additional tabs are available depending on what you selected on the Build an App page:<br><br>• **Installer Settings**: If you use the **Add Native Installer to AppWave Catalog** feature, this tab is available. You can define the installer file, command line settings and the un-install key.<br><br>• **WebApps**: When you are mastering a WebApp you can edit the required parameters here. |
| Using the Test Step | Opens the **Master and Test Your App** page. This displays the output folder and file name. You can change the output folder and edit the file name.<br><br>When you click **Edit** in the App field, a dialog opens asking you to select the app file (extension .stp) to associate with the .tca file.<br><br>You also choose the type of Compression you want to use and click the Advanced button to select an encryption type.<br><br>For more information see: Understanding the Studio File. |

| Label | Description |
|-------|-------------|
| Using the Broadcast Step | Opens the **Broadcast your application to the AppWave** page. This step requires an AppWave URL, either using the default or one that you select. |
| | You can choose the option to remaster the project before broadcasting. This option is selected by default. |

## USING THE START STEP

When you click **Studio** the Start step is active. This is where you select which type of app you want to create: from Install or from Scratch. You can also choose to add to an app already created.

### Create from Install

When you click **Create AppWave Supported or Custom App from Install** or click the Record Application Installation dialog appears.
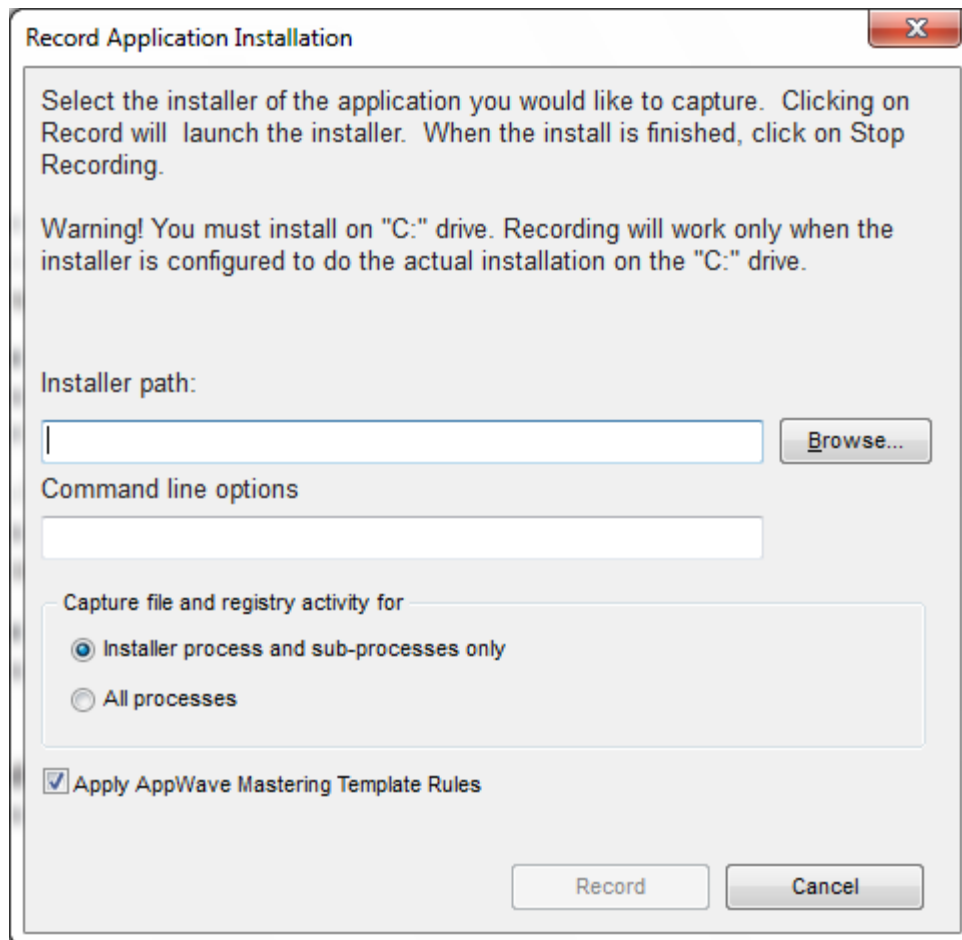
The dialog has the following fields:

| Function | Description |
|---|---|
| Installer Path | Browse to the path were the installer of the application you wish to capture resides.<br><br>You must install on the "C" drive or the recording doesn't work. |
| Command Line options | The "Command Line" value indicates how to launch the file and the value is the trigger to run a specific command.<br><br>For information on commands you can use, see Using the Command Line. |
| Capture file and registry activity for:<br>• Installer process and sub-processes only<br>• All Processes | Records only the install process.<br><br>Records all the processes associated with the application. |
| Apply AppWave Mastering Template Rules | As the process goes through the steps to create an App, tasks may be created indicating missing data. This option automatically completes any missing data. This is the default setting. |
| Record | Click **Record** to launch the installer. The **Recording** dialog appears and the red light blinks indicating that it is recording.<br><br>You can start recording, stop recording, retrieve recording log messages (if any), and configure what process to record (all or just the installer).<br><br>You have two options in the Recording dialog: cancel the recording or finish the recording. |
| Cancel | Cancels the recording process and closes the Record Application Installation dialog. |

**Create AppWave Support or Custom from Scratch**

When you click **Create AppWave Supported or Custom App from Scratch** the **Select Folders and Files to Include** dialog appears.
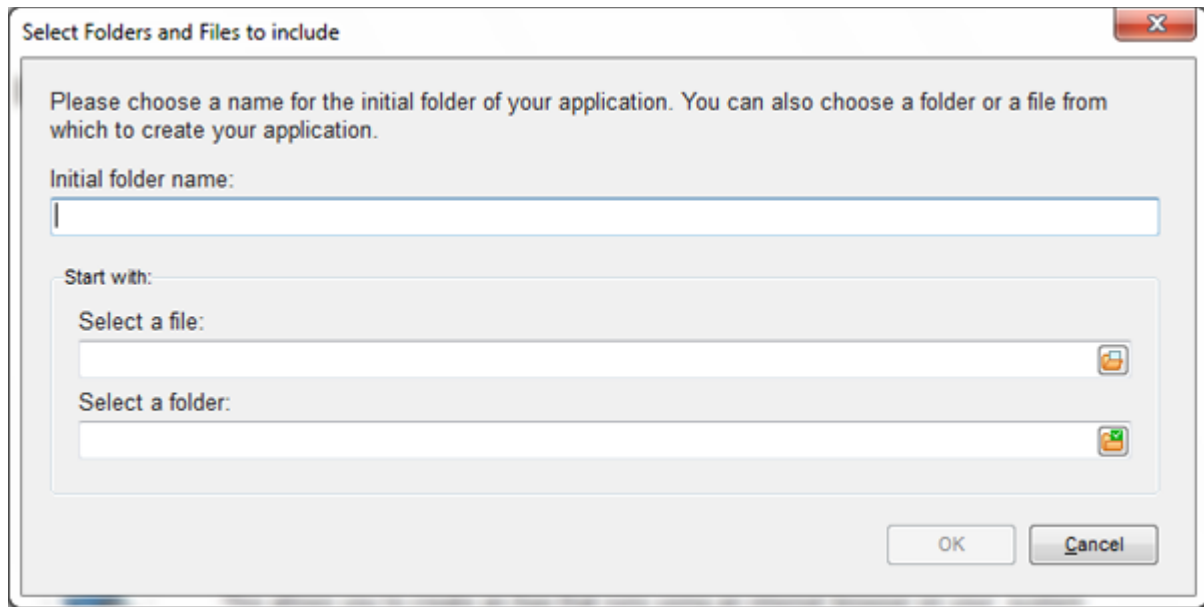
This dialog has the following fields:

| Field | Description |
|---|---|
| Initial Folder Name | Create a name for the initial folder. This folder is the "working" folder for your application. |
| Select a file | Browse to the Installer path and select the executable. The Installer path must on the "C:" drive. For example: `C:\Program Files\Embarcadero\`. |
| Select a folder | Browse to an existing folder or make a new folder. This contains the executable from the "Create from Scratch" process. |

## Add to App from Install

You can add another application to a previously master app by recording the installation of the additional application. For example, you may want the AppWave Browser user to have access to a database client software component installed in addition to the main software application. If so you can do this by using this option.

Another example is when you need to apply update patches to the application you are mastering. Adding one or more recordings in the same mastering process will only create one .app file. You can designate entry points for the additional apps, by using the **ADD NEW** function on the page, but this may not be necessary. Clicking this option starts the **Record from Install** operation.

### Create AppWave Support WebApp

This option allows you to create an app that is accessible via an internet browser. Mastering a WebApp does not require recording as for other apps, but you must define the required parameters.

### Add Native Installer to AppWave Catalog

This allows you to add a Native Installer to the AppWave Catalog with the required metadata. This function delivers software that cannot be easily mastered. Some examples of this are: Native 64-bit, printer drivers, and software establishing Windows services.

It can also distribute data and media content packaged up as installers. Examples of this type of content would be: data models, training materials and manuals, and data sets.

## USING THE LAUNCH & BRAND STEP

The Launch & Brand page opens automatically once you click **Finished** in the Record Installation dialog. The following fields need to be completed before advancing to the next step.

> **NOTE:** When you first open this page, any empty required fields are listed in the lower Tasks panel. Click **Fix** on an issue and you are taken to the step and field where the fix is needed.

| Function | Description |
|---|---|
| ADD NEW | Click **ADD NEW** to add new entry points to the mastered App. For example, when you master a complex application like Microsoft Office, there are several applications installed. You must complete a page for each of the applications so the user can run them. To the AppWave Browser user, the applications appear in the catalog as individual apps. To the AppWave admin, there is one .app file for the suite. However each product in the suite has its own entry in the App catalog. |
| Launch Executable | Select the executable from the drop-down list. This is a required field. |

| Function | Description |
|---|---|
| Working Folder | The working folder field is automatically populated once you select the executable. You can also browse to a different folder. The directory for the program executable cannot be used for the working folder. If so, you can change it's location if necessary.<br><br>This is a required field. |
| Parameters | Enter any parameters that are required by the launch executable. |
| Product Name | Select the Product name from the drop-down list. This is a required field. |
| Display Name | Enter the name of the product you want to appear on the page. |
| Product Version | Select the appropriate production version from the drop-down list. All available versions are listed including AppWave Template. If no product version is available, enter "1.0." |
| Product Edition | Enter edition information so the end user knows exactly which edition was used in the creation of the App. This is a required field. |
| Publisher | Enter the publisher of the product. For Embarcadero products use *Embarcadero Technologies*.This is a required field. |
| Icon | Select the icon file (.ico) or .png file by clicking the drop-down icon and opening the appropriate folder. |
| EULA | Enter the location of the End User License Agreement (EULA). Click the drop-down arrow in the EULA field and select the text file containing the licensing agreement information from the installed application's folder.<br><br>The appearance of the EULA during the install depends on the type of license required for the application you are managing. If a EULA is required, the license agreement appears during the install process and the user must accept this agreement before the license verification process can proceed. |

| Function | Description |
|---|---|
| Screen Shot | Select a screen shot image that you want to appear when a user selects an App. The image of the app appears on the Overview tab.<br><br>The file format is png. |
| Web Site | Enter the URL for the application's web site. |
| Support Contacts | Enter the contact information for the application's company. This information is then available on the Apps tab for the installed application. |
| System Requirements | Enter any system requirements that are needed for the application to run. |
| Product Description | Enter any descriptive information. This information appears with the application in the list of application in the AppWave. This information is then available on the Apps tab for the installed application. |
| Long Product Description | Enter additional descriptive information here.This information appears with the application in the list of application in the AppWave. |
| Release Date | This date is when the application was released to the public for general use. |

## USING THE APP STEP

You can configure your app by viewing what was created when you built a new App. You have access to the following:

| Icon | Name | Description |
|---|---|---|
|  | Files | Click Files to display the file entries for your App.<br><br>This lists all the files that are included in your App. It also displays the properties of each file. The information is displayed in either a list or grid format.<br><br>You can add or remove files and folders, or create a new folder. |

| Icon | Name | Description |
|------|------|-------------|
| | Installer Settings | Displays the settings that required to master the native installer app. |
| | Registry Settings | Displays any existing registry keys that were created when the install file was read. You can add new registry entries and then view and change the attributes of that new entry. |
| | Environment Variables | Lists the environment variables for your App. You can add new variables, edit existing variables, or remove variables. |
| | Runtimes | When a product (for example the .NET framework required by RADStudio) needs an execution runtime, it is provided to those who master applications. Once you make a selection you are requested to save your changes to the current working file in order to analyze the required runtime changes. |
| | Custom Scripts | The AppWave Studio Custom Script Support gives you the ability to attach scripts to key application triggers, such as activation, start, close, and deactivation that can accomplish whatever is necessary. |
| | WebApps | This is where you edit or define the required parameters to master a WebApp. |

FILES

To access this dialog click the app step and the **Configure your App** dialog appears. Click **Files** or to open the **File entries for your App** page. The left-hand panel displays all your files.You can add or remove files. You can also create a new folder under a selected folder.

You have a drop-down box where you can select the Show All (visible only to the App) or Exposed to Host (visible on the host machine).

The Properties and Grid tabs to the right display information on a selected file.
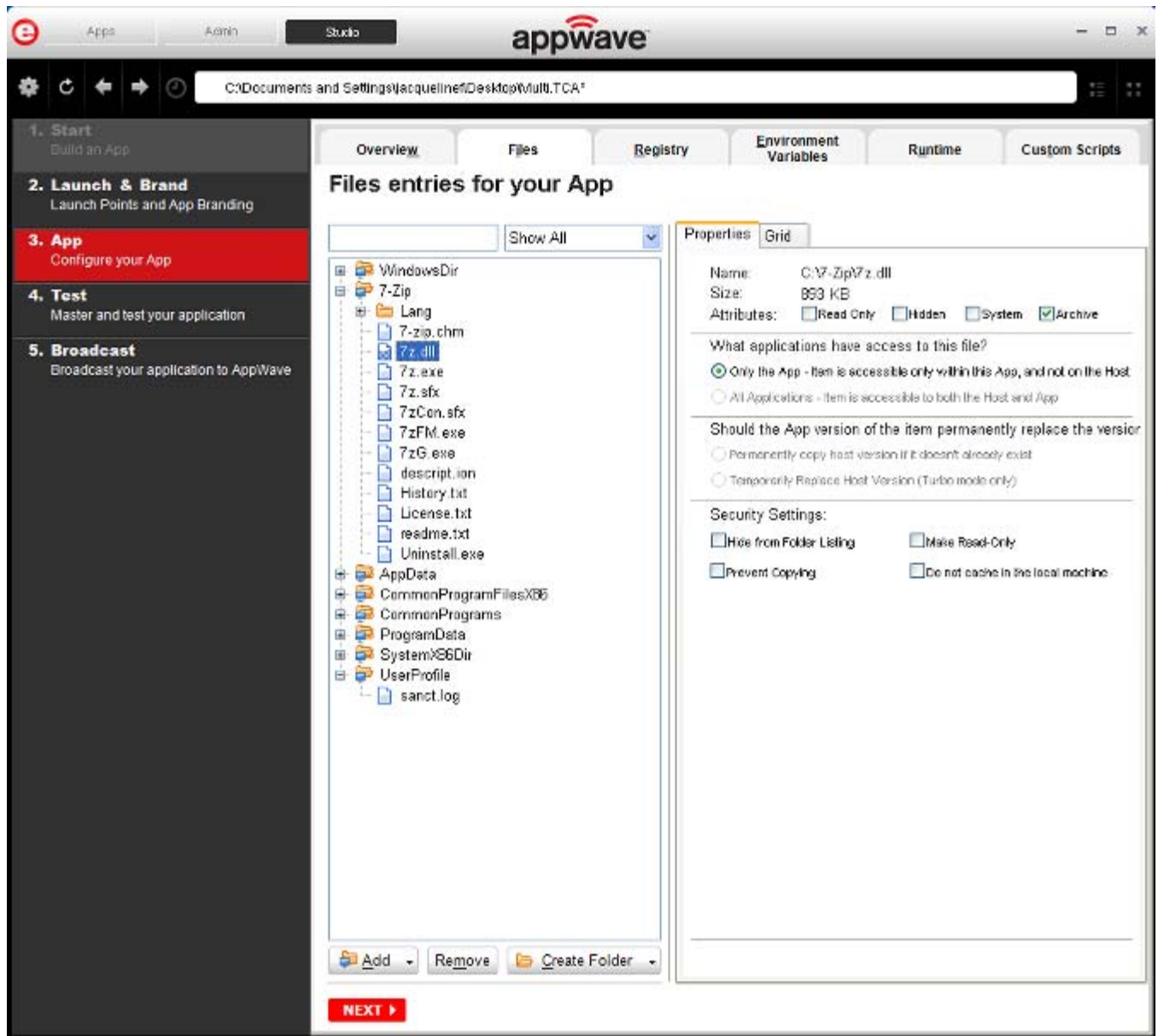
## File Actions

You can add or remove files and folders, or create a new folder. There are additional commands available under **Add**.

| Button | Description |
|---|---|
| Add | When you click the drop-down arrow you have four options.<br><br>**File**: The Open dialog appears. Browse to the file you want to add and click **OK**.<br><br>**Folder**: Opens the **Browse for Folder** dialog. Select the folder you want to add.<br><br>**Virtual Folder**: Opens the **Add a Predefined Virtual Folder** dialog. Select a folder from the drop-down list and click OK.<br><br>Virtual folders are not dependent on location in a hierarchical directory. So a Streaming core allows you to basically mount the file system so that when you want to add to your App, and remaster it, you have the virtual file system present on the hard drive so you can do whatever you want with it.<br><br>**Add Root Folder**: Opens the **Browse for Folder** dialog. Select the folder you want to add. The new folder is added to the top level or root of the tree. |
| Remove | Select a file in the tree and click **Remove**. A Confirm dialog appears asking you to click **OK** to continue. |
| Create Folder | **Create Folder** Select a folder in the tree and click **Create Folder**. Enter a name for the new folder and it is placed as a subfolder in the tree.<br><br>**Create Root Folder**: Enter the name of the new folder to add. The new folder is added to the top level or root of the tree. |

You also have commands available when you right click on a folder or file in the tree.

| Command | Description |
|---|---|
| Add File | The Open dialog appears. Browse to the file you want to add and click **OK**. |
| Add Folder | Opens **Browse for Folder**. Select the folder you want to add. |
| Add Virtual Folder | Opens **Add a Predefined Virtual Folder**. Select a folder from the drop-down list and click **OK**. |

| Command | Description |
|---------|-------------|
| Add Root Folder | Opens **Browse for Folder**. Select the folder you want to add. The new folder is added to the top level or root of the tree. |
| Create Folder | Select a folder in the tree and click **Create Folder**. Enter a name for the new folder and then it is placed as a subfolder in the tree. |
| Create Root Folder | Enter the name of the new folder to add. The new folder is added to the top level or toot of the tree. |
| Remove | Right-click a file in the tree and click **Remove**. A Confirm dialog appears asking you to click **OK** to continue. |
| Apply Visibility to Children | Takes the parent folder visibility settings and applies them to all the children. |

**File Filters**

You have a drop-down box where you can select **Show All** (visible only to the App) or **Expose to Host** (visible on the host machine).

**Properties/Grid Tabs**

Information on a selected file/folder is displayed in two different formats: properties and grid.

The properties tab for a *file* displays the following information:

| Option | Description |
|--------|-------------|
| Name | If you selected a file, the file name and path of the selected file. If a folder is selected, the number of files and/or folders are displayed. |
| Size | If an application, the size of the file is displayed. |
| Attributes | The attributes of a selected file are displayed: Read Only; Hidden, System, Archive (default setting) |
| Only the App | The selected file or folder is only accessible within the app and is not visible to the host. Files or folders with attribute changes are kept private in a cache managed by the Browser. |
| All Applications | The selected file or folder is available to the host and can be accessed by external processes and users. Unless it is hidden from the folder listings, it is displayed by all applications that list the contents of the host's file system. |

| Option | Description |
|---|---|
| Permanently copy host version if it doesn't already exist | Before activation, the app checks to see if this file already exists on the host.  If it does, it uses the existing file. If the file does not exist, it is permanently placed on the machine. |
| Temporarily Replace Host Version (Turbo mode only) | If this file already exists on the host, the app version of the file replaces the existing file as long as the app is active in the Browser. If you remove the app or clear the local Browser cache, the previous version is restored. |
| Hide From Folder Listings | This file is not displayed in the list of folders. Select this option if you want to hide this particular file in a file listing so it is not visible to other users. |
| Prevent Copying | Select this option if you do not want other users to copy the file from a temporary location where it was streamed. However these files are able to be read from outside. |
| Make Read Only | Select this option if you do not want other users to alter the files. |
| Do not cache in the local machine | This file is not permitted to be cached to the local system. This security setting ensures that the file is not located anywhere on the system. |

The properties tab for a *folder* displays the following information:

| Option | Description |
|---|---|
| Name | The file name and path of the selected file. |
| Contains | The number of files and/or folders is displayed. |
| Only the App | The selected file or folder is only accessible within the app and is not visible to the host. Files or folders with attribute changes are kept private in a cache managed by the Browser. |
| All Applications | The selected file or folder is available to the host and app, and can be accessed by external processes and users. Unless it is hidden from the folder listings, it is displayed by any and all applications that list the contents of the host machine's file system. |
| Permanently copy host version if it doesn't already exist | Before activation, the app checks to see if this file already exists on the host. If it does, it uses the existing file. If the file does not exist, it is permanently placed on the machine. |

| Option | Description |
|---|---|
| Temporarily Replace Host Version (Turbo mode only) | If this file already exists on the host , the app version of the file replaces the existing file as long as the app is active in the Browser. If you remove the app or clear the local cache, the previous version is restored. |

The grid format displays the following for both files and folders:

| Option | Description |
|---|---|
| File/Folder Name | The name of the file or folder selected. |
| Size | If a file, the size of the file. If a folder, then the number of files/folders contained in the folder. |
| Application Access | The selected file or folder is available to the host machine and can be accessed by external processes and users. Unless it is hidden from the folder listings, it is displayed by any and all applications that list the contents of the host machine's file system. |
| App Access | The selected file or folder is only accessible within the app and is not visible to the host machine. Files or folders with attribute changes are kept private in a cache managed by the Browser. |
| Path | Path where the file/folder is stored. |

To access the attribute information in the grid format, select a file in the grid and click **Edit**. The Edit selection dialog lists all the attributes for the selected file.

You can delete a file by selecting the file and clicking **Remove**. A confirm dialog appears.

INSTALLER SETTINGS

Click the Installer tab or  on the Overview tab, the installer settings that you entered when you first started the mastering process for a Native Installer are available to edit.

| Command | Description |
|---|---|
| Select installer launch executable | The install command is an executable string that installs the application. |
| Command-line parameters | The "Command Line" value indicates how to launch the file and the value is the trigger to run a specific command. |

| Command | Description |
|---|---|
| Provide an un-install registration key | Enter or edit the key that enables the un-install function. This is a required field. |

REGISTRY SETTINGS

When you click Registry or  on the Overview tab, the **Registry entries for your app** dialog appears. You can:

- Add a registry key from the system registry

- Remove a registry key you have added

- Create a registry key or value

When modifying registry keys it is important to know the system variables. You need to know what variables can be used that will be replaced with the correct path when an app is loaded on a machine. For more information, see: [Predefined Virtual Folders](#).

On the left is a tree of registry setting groups. The base folders are defined in the .tca model:

HKEY_CLASSES_ROOT

HKEY_CURRENT_USER

HKEY_LOCAL_MACHINE

HKEY_USERS

You have a drop-down box where you can select the Show All (visible only to the App) or Exposed to Host (visible on the host machine).

**CAUTION:** It is best practice to review the registry settings before broadcasting your application. Make sure there are no files or registry keys not directly relating to the application. Remove any that have been included in error.

You need to also verify that registry keys refer to a generic path. For example a path should be `%USERPROFILES%\Documents` not `C:\Documents and Settings\<username>\Documents`.

### View and Change Attributes

In the right-hand panel you can view and/or change the attributes of a selected registry entry.

| Command | Description |
|---|---|
| What applications have access to this registry | |

| Command | Description |
|---|---|
| Only the App | These are items accessible only within the App, and not on the Host. The selected file or folder is only accessible within the app and is not visible to the host machine. Files or folders with attribute changes are kept private in a cache managed by the Browser. |
| All Applications | The selected file or folder is available to the host and app, and can be accessed by external processes and users. Unless it is hidden from the folder listings, it is displayed by any and all applications that list the contents of the host machine's file system.<br><br>**Note**: See caution below when mastering an app on the Windows 7 OS. |
| Should the app version of the item permanently replace the version on the Host? | |
| Permanently copy host version if it doesn't already exist | Before activation, the app checks to see if this registry already exists on the host machine. If it does, it uses the existing file.If the file does not exist, it is permanently placed on the machine.<br><br>**Note**: See caution below when mastering an app on the Windows 7 OS. |
| Temporarily Replace Host Version (Turbo mode only) | If this registry already exists on the host machine, the app version of the file replaces the existing file as long as the app is active in the Browser. If you remove the app or clear the local Browser cache, the previous version is restored. |

**CAUTION:** When mastering an app on the Windows 7 OS, create registry keys under HKEY_CLASSES_ROOT or in one of the subdirectories of HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, and HKEY_USERS.

If the properties for the key you created are set to "All Applications," "Permanently Copy Host Version" and "Temporarily Replace Host Version" an error message (APM_E_APP_CONFIG_INSTALL_REGISTRY) will be generated.

To fix this registry problem, you must delete the key you just created.

ENVIRONMENT VARIABLES

When you click Environment Variables or [icon], the **Environment variables for your app** dialog appears. The following table discusses the three options:

| Field | Description |
|---|---|
| Add | Click **Add** and the **New Environment Variable** dialog appears. Enter the variable name and value name here. |
| Edit | Once an environment variable is added, it can be edited. |
| Remove | Select a variable and click **Remove**. A confirm dialog appears. |

RUNTIMES

When you click **Runtime** or [icon] on the Overview tab, the **Runtimes required by your app** dialog appears. You can pick and choose what runtimes you want to include in your App.

NOTE:    The version attribute must match the installation folder name for the specified version.

The table below lists sample runtimes.

| Display Name | Description |
|---|---|
| .Net .35 runtime plug-in | Plug-in to add a cross OS set of registry keys and files that make up the Microsoft .Net 3.5. |
| Microsoft J# runtime plug-in | Plug-in to add a cross OS set of registry keys and files that make up the Microsoft J# runtime. |
| Microsoft Visual C++ 2005 SP1 redistributable runtime plug-in | Plug-in to add a cross OS set of registry keys and files that make up the Microsoft Visual C++ 2005 SP1 |
| Microsoft Visual C++ 2008 SP1 redistributable runtime plug-in | Plug-in to add a cross OS set of registry keys and files that make up the Microsoft Visual 2008 SP1 |

Once you select one of the default settings, you are prompted to save your changes to the current working file in order to analyze the required runtime changes.

You can find a listing of all currently available plug-ins through the AppWave Admin Console. From the **Console Dashboard**, click Manage mastering plug-ins.

CUSTOM SCRIPTS

When you click **Custom Scripts** or  on the Overview tab, the Custom Scripts for your app dialog opens. You can set and edit triggers to create a custom script for the app you are mastering.

These scripts are typically small executable files and batch files. There is a 10 MB size limit for each script. Exe, bat, and cmd files are supported

When you convert a locally installed application into a streamable application, the "native" states of the installed application are captured and recorded in the appset file. When you activate an application, those states are re-created on your system.

Sometimes in order for the application to run properly, it may be necessary to adjust those native states or the runtime environment before, during, or after activation because certain actions cannot be captured in an appset.

An AppEvent is a process or action that uses a trigger to run an executable program or script. In turn, the script or program contains instructions to perform specified runtime action(s) that are required by the application being packaged, such as launching another application. Examples of actions that may need to be performed at runtime are:

- Establish a connection to a shared network location before the application starts

- Run a file backup/archive process for user created documents after the application has exited.

- Configuring plug-ins for office suites and browser applications

- Configuring virtual printers

- Configuring game parameters for a video game

- Starting a background program

You can enable, edit or delete a trigger.

1   Click a trigger or application event on the list, click **Edit**, and the Enable box is activated. Once you have made all your edits and saved your trigger, you can then select the **Enable** option.



> NOTE:   When you select a trigger, notice that the panel below the trigger table now displays the name of the trigger. For example if you selected "After Activation", the next panel is now labeled "Trigger: After Activation".

The following items can be edited:

| Item | Description |
|------|-------------|
| Name | The name of the script. |
| **Frequency** | |
| Fire the first time only | The script runs only the first time the trigger is activated. |
| Fire every time | The script runs every time the trigger is activated. |
| **Script** | |
| Filename | Click the folder icon to locate the script. Exe, bat, and cmd scripts are supported. |
| Parameters | The parameters to use with the script. |
| Use Local Application | Select if the script is located on the local PC or on the company network. |
| **Event** | |
| Run outside of the App | |
| Run elevated (only supported on Windows Vista or later) | This setting causes Windows to elevate the program to administrator level before running it. Some programs require administrative privileges to run properly. |
| Wait for the handler to exit | Waits for the code that has been activated by the trigger to complete before exiting. |
| Time out after | If the code hasn't completed before the time out after period expires, the trigger exits. Enter "0" for no time out. |
| Success codes | Values that if returned by the trigger code indicate that the code ran successfully. Any other values are treated as errors. |

WEBAPPS

When you click **WebApp** or  on the Overview tab, the New WebApp dialog appears. You can set the basic information, what browser to use and any browser parameters.

This is where you can add or edit the settings you entered when you first starting the mastering process:

| Field | Description |
|---|---|
| Basic Information | **Title**: Enter or edit the title that will appear with the mastered app.<br>**URL**: Enter or edit the URL for the WebApp |
| Options | Select the browser you want to use from the drop-down list. Your options are: Default, Chrome, IExplore, or Firefox. |
| Browser Parameters | In the Browser Parameters you can add or edit the browser frame parameters. For example:<br>• the size of browser window on startup (with option to maximize on startup),<br>• show/hide the URL field (eg. use headless Chrome to "lock down" user from leaving the browser window, and<br>• show/hide back/forward navigation buttons). |

## USING THE TEST STEP

Click **Test** to master and test your application locally. This option allows you to master an app file using the given template .tca file. The template .tca file needs to have the files that go with it. The .tca file has file system information that actually lists files for the program it needs to master.

The mastering process verifies the files that are listed in the .tca file data. It then uses these files to complete the mastering of the app file. The template .tca file needs to be a complete and valid .tca file with all the required data (i.e. Publisher, Vendor, etc.) or the mastering fails. The result of this process is the creation of an app OutputName .stp file.

| Field | Description |
|---|---|
| Output Folder | The default output folder is C:\Documents and Set-tings\<user>\Embarcadero\Studio\Output. |
| File Name | The file name that appears is from the information you entered on the Brand panel. It includes the Name and Version of your application. |
| App | You can select the app file to associate with the .tca file for this application. |
| Compression | You have three options for determining the file size during mastering: No, Fast, and Best. |

| Field | Description |
|---|---|
| Advanced | Allows you to select the type of encryption you want to use for the App. |
| Master | After your have set and verified your selections, click **Master**, and the **Mastering App** dialog appears.<br><br>The mastering process changes a Windows-based application into a streamable format. It also gives you the opportunity to launch the app to capture the prefetch information that allows the app to launch quicker. |

## App

You can select an app file (.stp) that corresponds to the Studio file you created.

1   Click **Edit** and the **Select a mastered app** dialog appears.
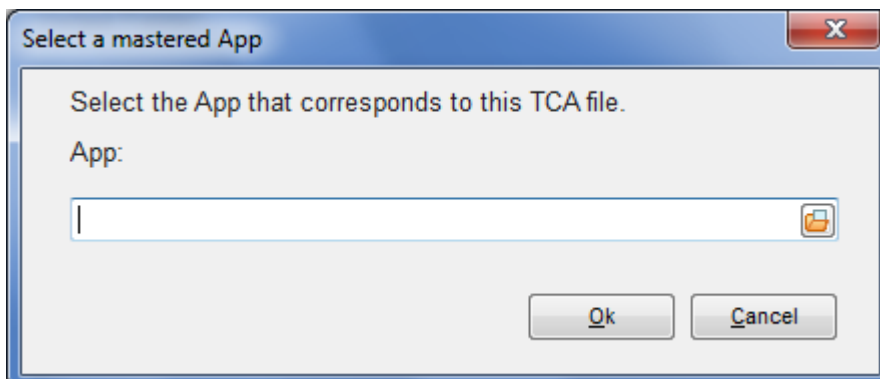


2   Browse to the location of the application's STP File
    (`C:\xxx\<user>\Embarcadero\AppWaveStudio\<application.stp>`).

3   You can also choose the option to update the icon from the App.

4   Click **OK** to close the dialog.

## STP File

This file is the outer layer that encapsulates the other application components. These files are stored in `C:\xxx\<user>\Embarcadero\Product Repository\<application>\<version number>\<product name and version>\`. The other application components include:

| File Extension | Description |
| --- | --- |
| aib | This file contains the metadata that describes the content (or "blueprint") of the STC file. |
| stc | The code and other configuration information that comprises the streaming content of an application. |
| tok | Also known as a virtualization token, token files are comprised of security data. |
| ico | icon file |
| .htm/.txt | EULA file |

## Compression

You have three options for determining the file size during mastering:

- **No Compression**: This is useful for testing, but should not be used for streaming.

- **Fast**: Compresses and decompresses quickly. Use this setting to reduce mastering times. Can be used to stream on faster networks.

- **Best**: Decompresses faster, but it takes longer to compress. This method is recommended because it results in a 20% smaller App. This results in a greater transfer rate when streaming.

## Advanced

The Advanced Button allows you to configure the app. You can select the type of encryption you want to use for the app along with the operating systems to be supported by the app and whether or not the Turbo mode is required for each selected operating system.
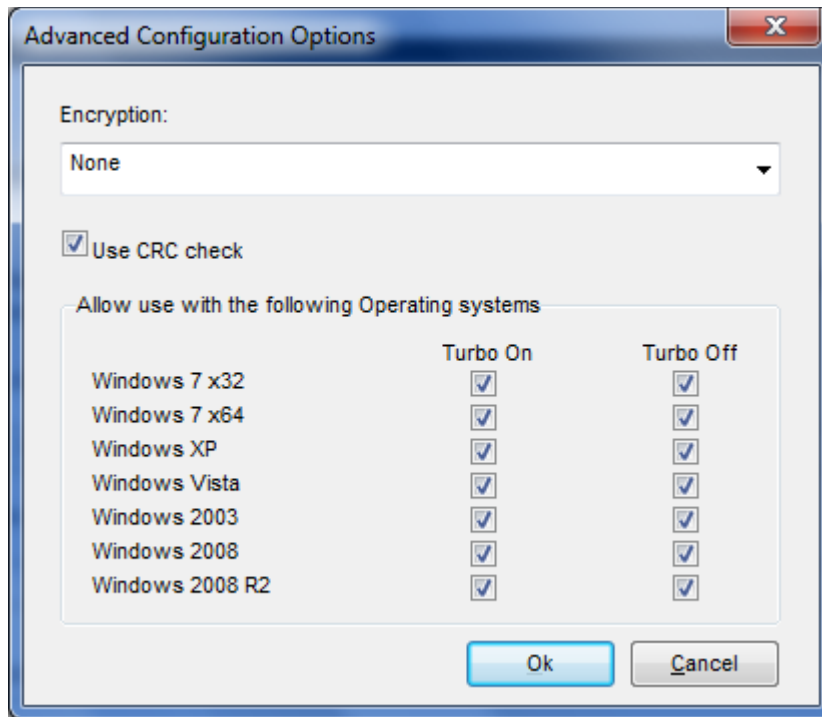
- This encryption provides a high degree of security, ensuring that application data remains safe and private when streaming over the Internet. You have three encryption options:

  - **None**: No encryption is needed.

  - **Bitshift**: An encryption method that shifts the digits to the left or the right.

  - **AES_CFB**: Cipher Feedback mode which makes a block cipher into a self-synchronizing stream cipher.

- Selecting the **Use CRC check** option specifies that the app integrity is verified based on the md5 file, in order to detect possible transfer data error.

- Under the **Allow Use within the Following Operating Systems** option, when the Turbo On option is selected, it indicates that the app is allowed to run on the specified OS and in Turbo mode. When deselected, the check box prevents the app from launching on the specified OS and in Turbo mode.
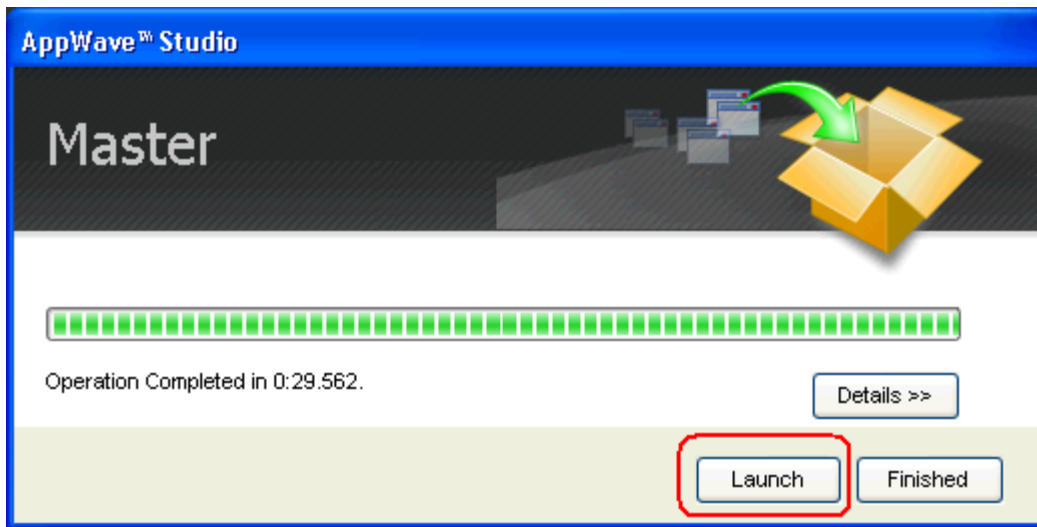
## USING THE BROADCAST STEP

The last step is to broadcast your application to the AppWave. The following fields appear on this page.

| Field | Description |
|---|---|
| AppWave URL | The default URL appears in this box. You can edit this URL. |
| Always remaster projects before broadcasting to the AppWave | This is selected by default. This is to guarantee that the app has the latest updates. |
| Broadcast | Click **Broadcast** and a dialog appears telling you need to save your file before continuing. Click **OK** and the Broadcasting dialog appears. Save the file in *C:\xxx\<username>\Embar-cadero\AppWaveStudio\Output directory.* |

1  Open the **Broadcast** page. Initially you will notice that a Task appears at the bottom of the screen. This task indicates that *"in order to Record the launch of the application for faster start-up time, you need to master the app from the Test tab before broadcasting. Once the launch data was recorded, remastering before broadcast will not remove this information."* Only some key blocks of application data from AppWave are necessary for the application to launch and perform basic functions.

   **NOTE:**  There is also an indicator next to the Broadcast step showing that there is a task that needs to be fixed.

2   Click **Fix** and a dialog appears instructing you on how to record prefetch information. Prefetch is the process of streaming application blocks in anticipation that those blocks are needed by an application.
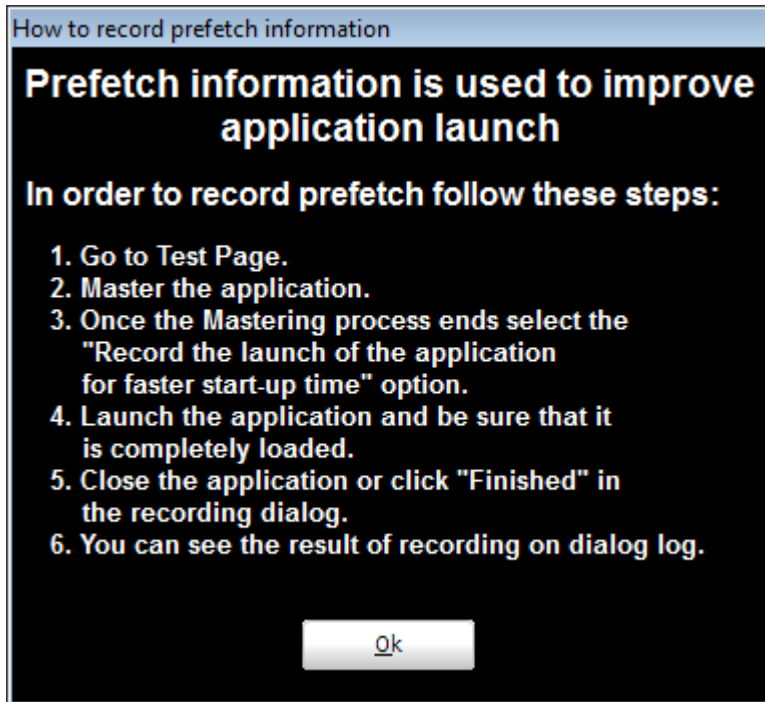


3   Return to the Mastering page and click . The Master dialog appears.

4   Select "Record the launch of the application for faster start-up time" and click **Launch**. This is to ensure that the application is completely loaded.
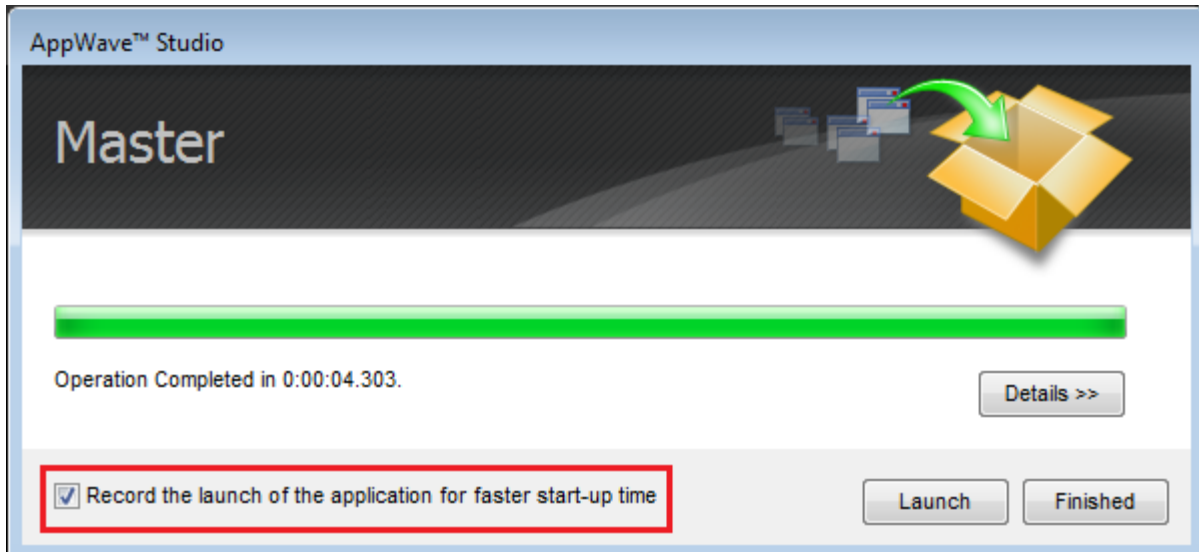
5    When the application has completed loading, close the application or click **Finished** in the recording dialog.

6    Click **Details** to view the results of the recording.
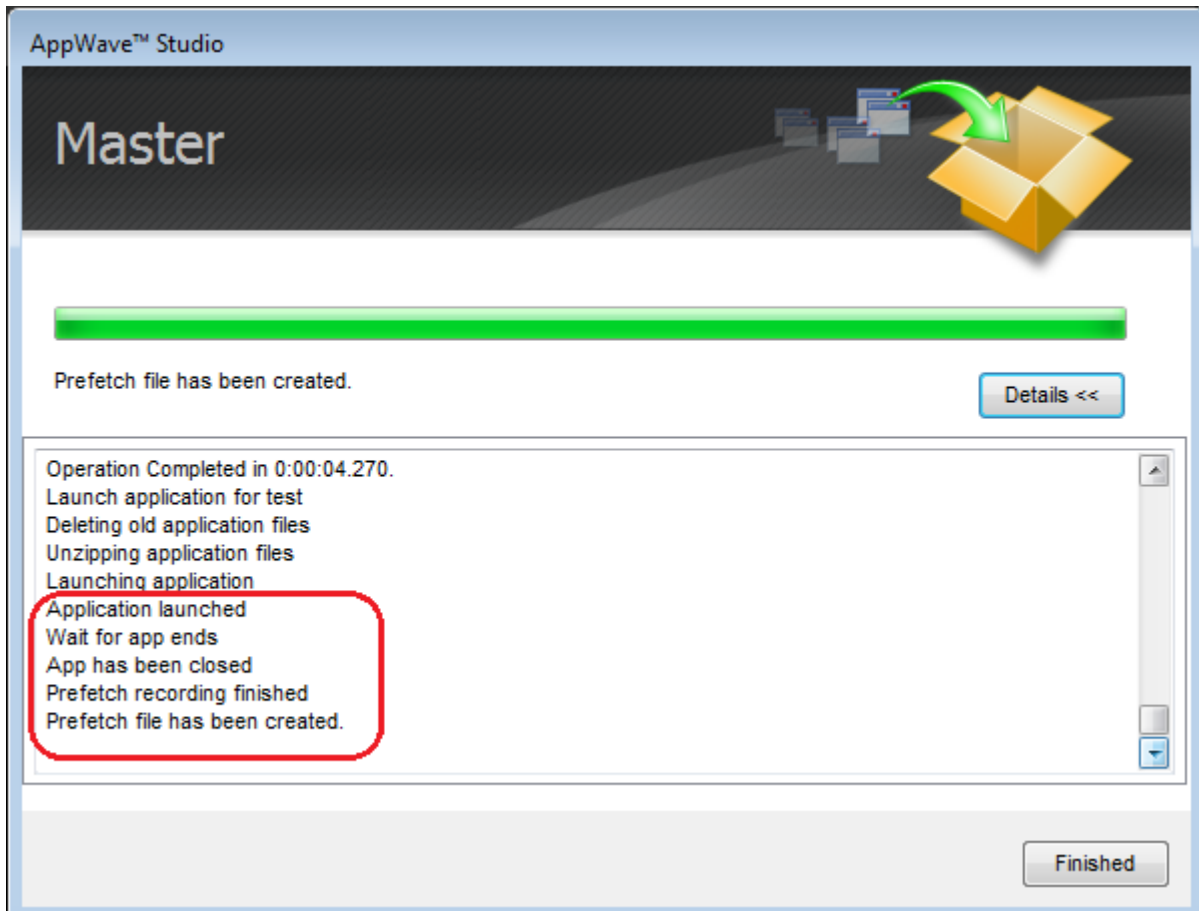
7   Click **Finished** in the Master dialog and then click **Next** to advance to the Broadcast page.

8   Make sure the option to **Always remaster projects before broadcasting to the AppWave** is selected.

9   Click **Broadcast** and save the changes to your file. The **Broadcast** dialog appears where you can track the remastering and broadcast of your app with the prefetch recorded information included.

10  Your app is now mastered with the prefetch information included.

## MASTER

This section is an overview of all the steps involved in mastering an application to create an app executable. Successful mastering depends on a thorough understanding of the components of each element. The app executables provide one-step functionality for each product. This simplifies broadcasting and enables side-by-side versioning of products.

You can master an app file using the given template Studio file. This file needs to also have the files associated with it.

For more information, see:

- Understanding the Studio File

- Before you Master an Application

- Obtaining Mastering Support

- What is Required for App mastering

- Using the Command Line

### UNDERSTANDING THE STUDIO FILE

The app file is mastered using the Studio or .tca file. A  .tca file is a set of information that defines the resources, launch points, and registry settings required to run a product. It also contains information about the product such as the version, name, and editions. The .tca file needs to have the files that go with it and has a file system information that actually lists files for the program it needs to master. The files have to be physically present in the directory that is saved in the .tca data.

The mastering process verifies the files that are listed in the .tca file data. It then uses these files to complete the mastering of the app file. The template .tca file needs to be a complete and valid .tca file with all the required data (i.e. Publisher, Vendor, etc.) or the mastering fails. The result of this process is the creation of an App OutputName .stp file. All the listed arguments are required. The OutputLocation directory needs to have a drive specified, and if the directory does not exist then the program tries to create one. The OutputName argument is used as a name for the .stp file, so there should not be any file named OutputName .stp in the OutputLocation directory.

## BEFORE YOU MASTER AN APPLICATION

- Understand the application

- Know what registry keys are used by the application

- Know what folders are used by the application

- Know what root folders or registry keys were created

- Know where the master files (the .app) and the Studio Files (files used during application creation) are located

- Understand how the application is licensed

- Understand the prerequisites

- Understand the scripting needs

## OBTAINING MASTERING SUPPORT

When you want to master an application, AppWave Studio will:

1   Retrieve the AppWave file (includes brand, launch, required plugins, EULA, and icons) from the AppWave. This information is used to automatically populate the product information;

2   Ensure that the latest version of all required plugins are available;

3   Download a plugin if it is missing;

4   Compare the existing plugin with the version on the mastering computer to ensure the latest version is used.

## WHAT IS REQUIRED FOR APP MASTERING

Creating an app involves:

- Cataloging the product in the Studio, including branding and all the data that makes up the studio metadata file.

- Having the EULA and icons for the application.

- Have the plugins that modify captured studio metadata file (.tca) so that the product is portable and runs on other platforms.

- Prerequisites, those things that are required by the product so the master runs on systems that don't have the prerequisites.

For more information, see:

- [Application Types](#)

- [File Formats](#)

**Application Types**

AppWave works with many types of applications. Two of the most typical are:

- Open source products or companies that allow distribution. These mastered applications can be redistributed as apps.

- Products that can't be redistributed. You then rely on instant mastering. Once the product is mastered and placed in your AppWave you do not have to master it again.

**File Formats**

Two file formats are used to add Applications and app mastering support for the AppWave and AppWave Browser.

- .PADE is a plugin used to help users master an application by automatically manipulating the file and registry entries in the virtual application

    **NOTE:** You can select the .Net v4.x runtime plug-in as part of the mastering process, including the option to exclude the plug-in (if pre-installed on client systems) in case the mastering process decides to include it.

- .AW is an AppWave file that adds Instant mastering support to a server and browser

    The AppWave file (.AW), created by the Plugin Manager, is what is used to add Instant mastering support to a server. The utility also allows the user to choose which plugins are required for Instant mastering of the target application. The AW file is saved with the same name as the APP.

    **NOTE:** The Product Manager is located in the same directory as the AppWaveCommand Line and the AppWaveBrowser executable.

## USING THE COMMAND LINE

The AppWavec.exe command line tool exposes Studio related functionality.  Basic usage details can be found by running the tool without any parameters

**NOTE:** In order to use this option the user must open in CMD C:\xxx\All Users\Application Data\Embarcadero\AppWaveBrowser\AppWavec.exe.

The available options are:

| Option | Description |
|---|---|
| AppWavec -A | Lists all supported functionality. |
| AppWavec -B | Broadcast file(s) to an AppWave. Used to send files to a host. The files can be an app, a template(.aw) or a new pade plugin. |
| AppWavec -C | Converts an endeavors studio project file (.stw) into a Studio project file (.tca). |
| AppWavec -F | Copy files and folders included on a project file (.tca) to a new location. |
| AppWavec -Ma | Create a stp file from a tca file. This option just uses recorder library from appvirt to create a stp file (that is a zip that contains virtualization files). |
| AppWavec -PPL | Creates .pade from a plugin |
| AppWavec -R | Record an installer and create the Studio project file (.tca file) |
| AppWave -X | Creates a new app file from .tca and .stp files. |

1  **AppWavec -A**

Lists all the command line options available and detailed information for each.

2  **AppWavec -B**

The -B or -Broadcast option broadcasts to an AppWave one of four different ways: 1) an .app file; 2) a Studio/STP file combination; 3) a PADE (mastered plugin) file; or 4) an AW file. For the second item, the Studio file has to be the file used to generate the STP file.

- Usage 1: Broadcast a project (Studio and STP files are required) AppWavec (-B|-Broadcast) (-T|-StudioFile) <myProduct .tca> -STP <myProduct .stp> -URL <AppWave> -U [<domain>]<user> [-PWD <password>] [-EDN] [-Debug]

- Usage 2: Broadcast an app (.app file is required)AppWavec (-B|-Broadcast) -APP <myProduct .app> -URL <AppWave> -U [<domain>]<user> [-PWD <password>] [-EDN] [-Debug]

- Usage 3: Broadcast PADE plugins (.pade files)AppWavec (-B|-Broadcast) -PADE <myPlugin .pade> -URL <AppWave> -U [<domain>]<user> [-PWD <password>] [-EDN][-Debug]

- Usage 4: Broadcast AppWave templates (*.aw files) AppWavec (-B|-Broadcast) -AW <myAppWave .aw> -URL <AppWave> -U [<domain>]<user> [-PWD <password>] [-EDN] [-Debug]

| Argument | Description |
| --- | --- |
| -B or -Broadcast | Selects the broadcast option. |
| -T or -StudioFile | The full path and file name of myProduct's .TCA file. |
| -STP | The full path and file name of myProduct's .STP file (created during mastering). |
| -APP | full path and file name of the mastered myProduct's .APP file. |
| -PADE | Full path and file name of the mastered plugin's .PADE file. |
| -AW | Full path and file name of the AppWave file to be broadcast. |
| -URL | The AppWave server URL. |
| -U or -UserName | The UserName for the AppWave server. |
| -PWD | The password for the user (optional). |
| -EDN | Login with public\Embarcadero Developers Network account. |
| -Debug | To output debug messages. |
| myProduct .tca | The full path and filename of your product's Studio file. |
| myProduct .stp | Full path to the STP file created when mastering myProduct. |
| myProduct .app | Full path to the APP file created when broadcasting myProduct. |
| myPlugin .pade | Full path to the PADE (mastered plugin) file. |
| AppWave | The AppWave server URL. |
| user | The username or domain\username for the AppWave login. |
| password | The password for the AppWave. |

### 3  AppWavec -C

Convert an endeavors studio project file (.stw) into a Studio project file (.tca)

- Usage:  AppWavec <-C|-Convert> -S <file + .stw> -O <outputLocation> -N <outputName> [Debug]

| Argument | Description |
|---|---|
| -C or -Convert | Selects the convert option. |
| -S or -SourceFile | The full path and file name of .stw file to convert. |
| -O or -Out | The full path to the output location directory. |
| -N or -Name | The output file name.  Do not include an extension |
| -Debug | To output debut messages. |
| file.stw | The full path and filename of the STW. |
| outputLocation | The directory where the .tca file is output. A complete drive and directory name is required. |
| outputName | The filename for the .tca file produced. This argument is just the name without any file extension. |

4   **AppWavec -F**

Copy files and folders included on a project file (.tca) to a new location.

- Usage:AppWavec <-F|-CopyFiles> -T <template.tca> -O <outputLocation>

| Argument | Description |
|---|---|
| -F or -CopyFiles | Selects the Copy Files option. |
| -T or -StudioFile | The full path and file name of the template Studio file. |
| -0 or -Out | The file output directory. |
| -Debug | To outpu debut messages. |
| template.tca | The full path and filename of the template Studio file. |
| outputLocation | The full path of the directory to save to. |

5 **AppWavec -Ma**

This option masters an App.  All files referenced in the template.tca must exist.  The template.tca file must be a complete, valid studio file with all required data.or mastering will fail.  Mastering creates an App: OutputName.stp file.  All listed arguments are required except -Debug.  The outputName argument is used as a name for the .stp file, so there should be not files named OutputName.stp in the OutputLocation directory.  If the output directory does not exist, it will be created.

- Usage:  AppWavec <-Ma|-Master> -T <template.tca> -O <outputLocation> -U <user> -N <outputName> [Debug]

| Argument | Description |
| --- | --- |
| -Ma or -Master | Selects the mastering option |
| -T or -StudioFile | The template Studio file |
| -0 or -Out | The full path of the output location directory. |
| -N or -Name | The file output name. |
| -U or -Username | The User name. |
| -Debug | To outpu debut messages. |
| template.tca | The full path and filename of the template Studio file. |
| outputLocation | The full path of the directory to save to. |
| outputName | The name for all artifacts created.  Do not include an extension. |

6 **AppWavec -PPL**

This creates pade from a plugin: The -PPL option masters the files of a PADE plugin into a PADE file. It requires the path to the folder containing the plugin files. Two files must be included: the version .dat and the main plugin file that has the same name as the plugin. Optionally additional files may be included. Only the files listed in version .dat will be included in the PADE file.

- Usage:  AppWavec -PPL <-PL|-Plugin> <fullPathToPlugInFolder> [<-O|-Out) <outputLocation> ] [-Debug]

| Argument | Description |
| --- | --- |
| -PPL | Selects the master plugin option. |
| -PL or -Plugin | The full path to the folder containing the plugin files. |

| Argument | Description |
|---|---|
| -O or -Out | The full path of the output location directory. |

7  **AppWavec -R**

Records an installer and creat the studio project file (.tca)

• Usage:  AppWavec

| Argument | Description |
|---|---|
| -R | Record installer |
| -silent | "Installer silent parameter".  The installer is launched with the parameter indicated. |
| -InstallerPath | "path". The path to the installer used. |
| -AppProcesses | "true or false".  Recording all processes on the system or just the installer. |
| -ProjectName | "name" for the generated tca file. |
| -Template | "path" Path to the template to be used |

8  **AppWave -X**

Extract: This option exports an app file (creates an APP file from a Studio/STP file combination). The Studio file (.tca) has to be the file used to generate the STP file.

• Usage: Create *.app file from TCA and STP:AppWavec (-X|-Export) -T <myProduct .tca> -STP <myProduct .stp> -O <outputLocation> -N <outputName> [-Debug]

| Argument | Description |
|---|---|
| -X or -Export | Selects the export option. |
| -T or -StudioFile | The full path and file name of the template Studio file. |
| -STP | The full path and file name of the mastered App. |
| -O or -Out | The file output directory. |
| -N or -Name | The output file name. |
| -Debug | To output debug messages. |
| myProduct .tca | The full path and filename of the template Studio file. |
| myProduct .stp | The full path and filename of the app file to be broadcast. |

| Argument | Description |
|---|---|
| outputLocation | The full path of the directory to save to. |
| outputName | The name of the app file to create. |

# APPWAVE STUDIO ERROR DIALOG

When the AppWave Browser encounters a problem, the error dialog appears.



This dialog has the following actions that can be taken.

| Element | Description |
|---|---|
| Continue Application | This option is available if an error did not close the application. |
| Restart Application | An error closed the AppWave Browser. Click Restart. If you were creating an App, that process was terminated. |
| Close Application | Closes the AppWave Browser application. This action does give you a warning dialog. |
| Send Bug Report | Opens a Send Assist dialog where you can enter your name, email, and details about the error. |
| Show Bug Report | Opens the AppWave Browser .exe dialog which contains the Bug Reports. |

# BUG REPORTS
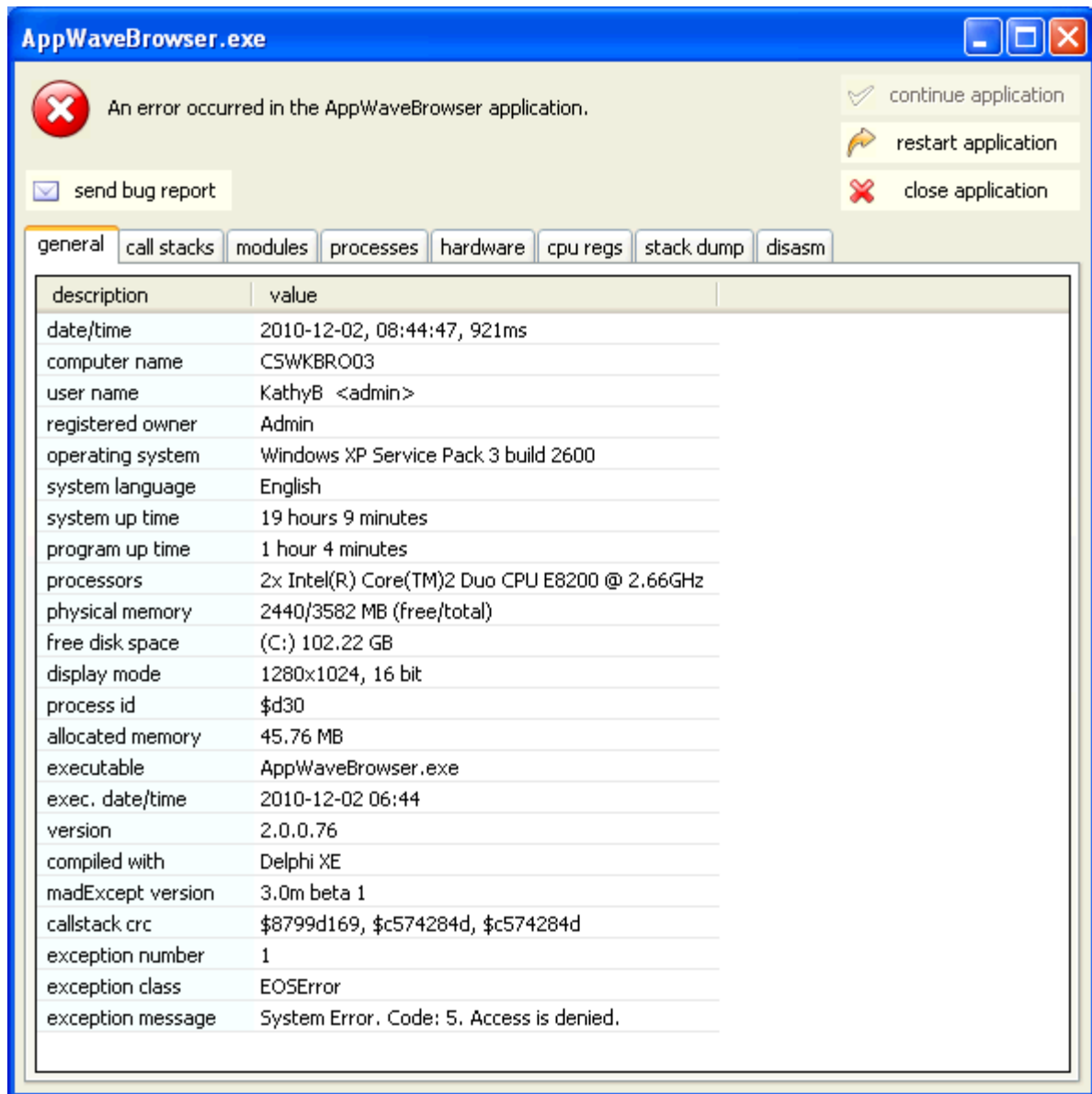
When you click Show Bug Report in the Error dialog the AppWave Browser .exe dialog appears. There are nine different tabs showing information on the bug types.
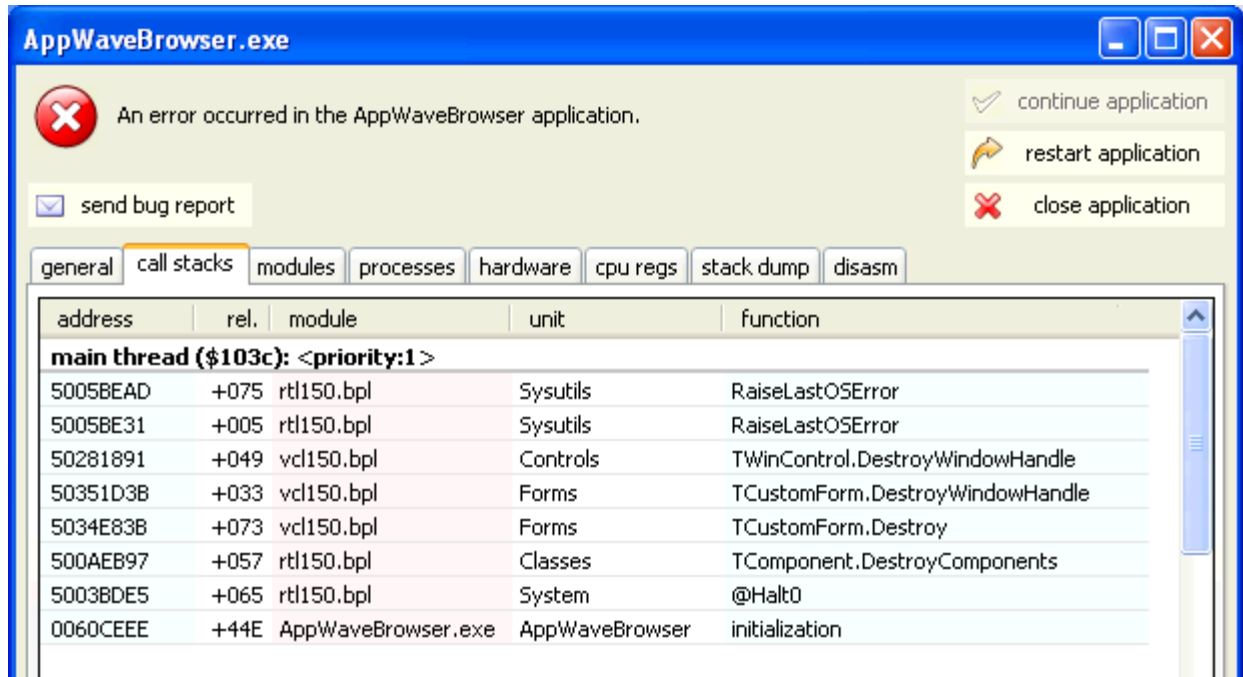
- [General](#)

- [Call Stacks](#)

- [Modules](#)

- [Processes](#)

- [Hardware](#)

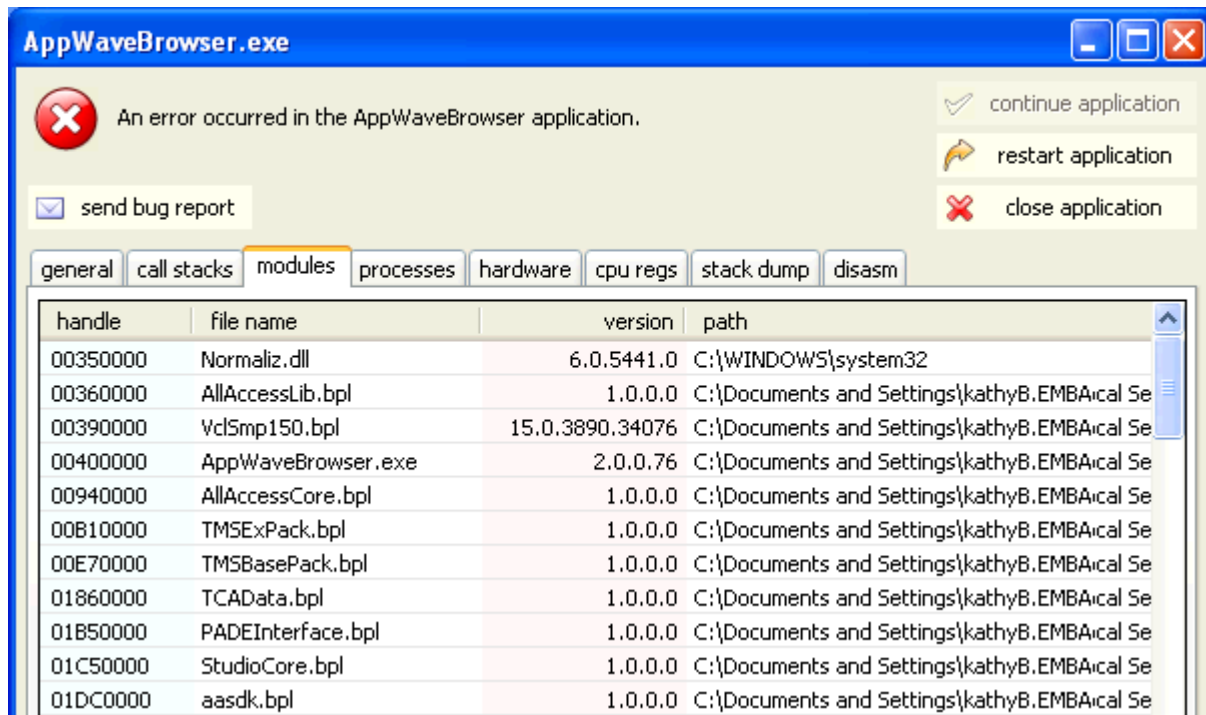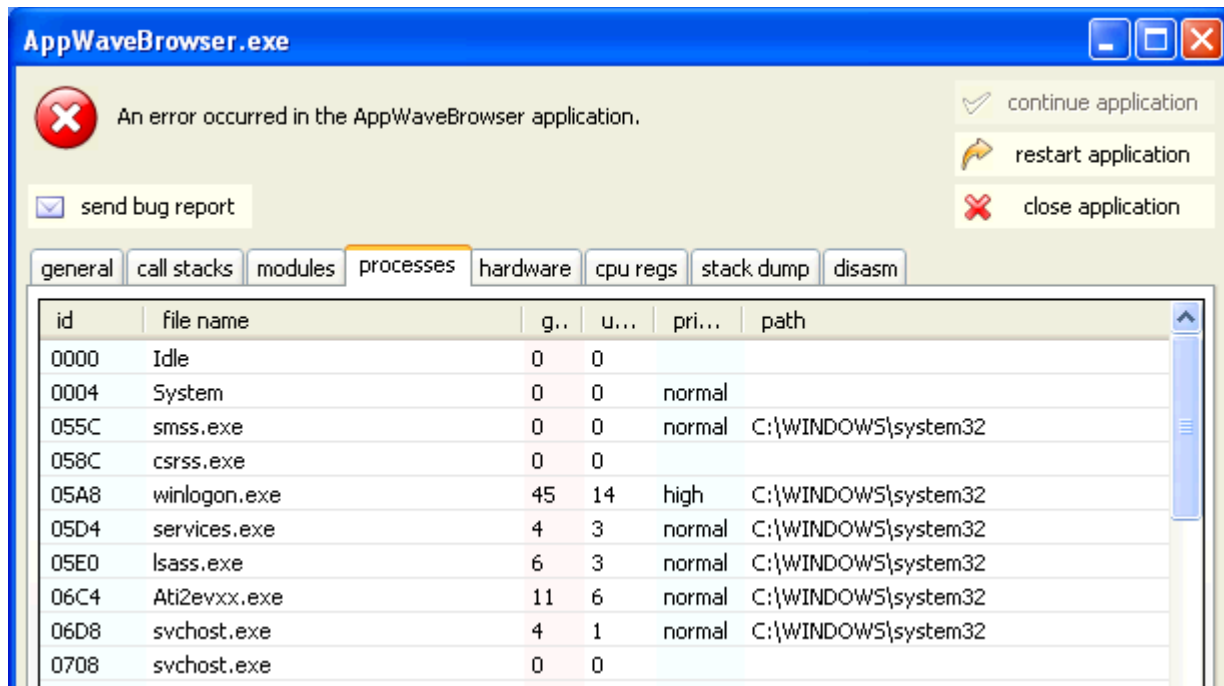- [CPU Regs](#)
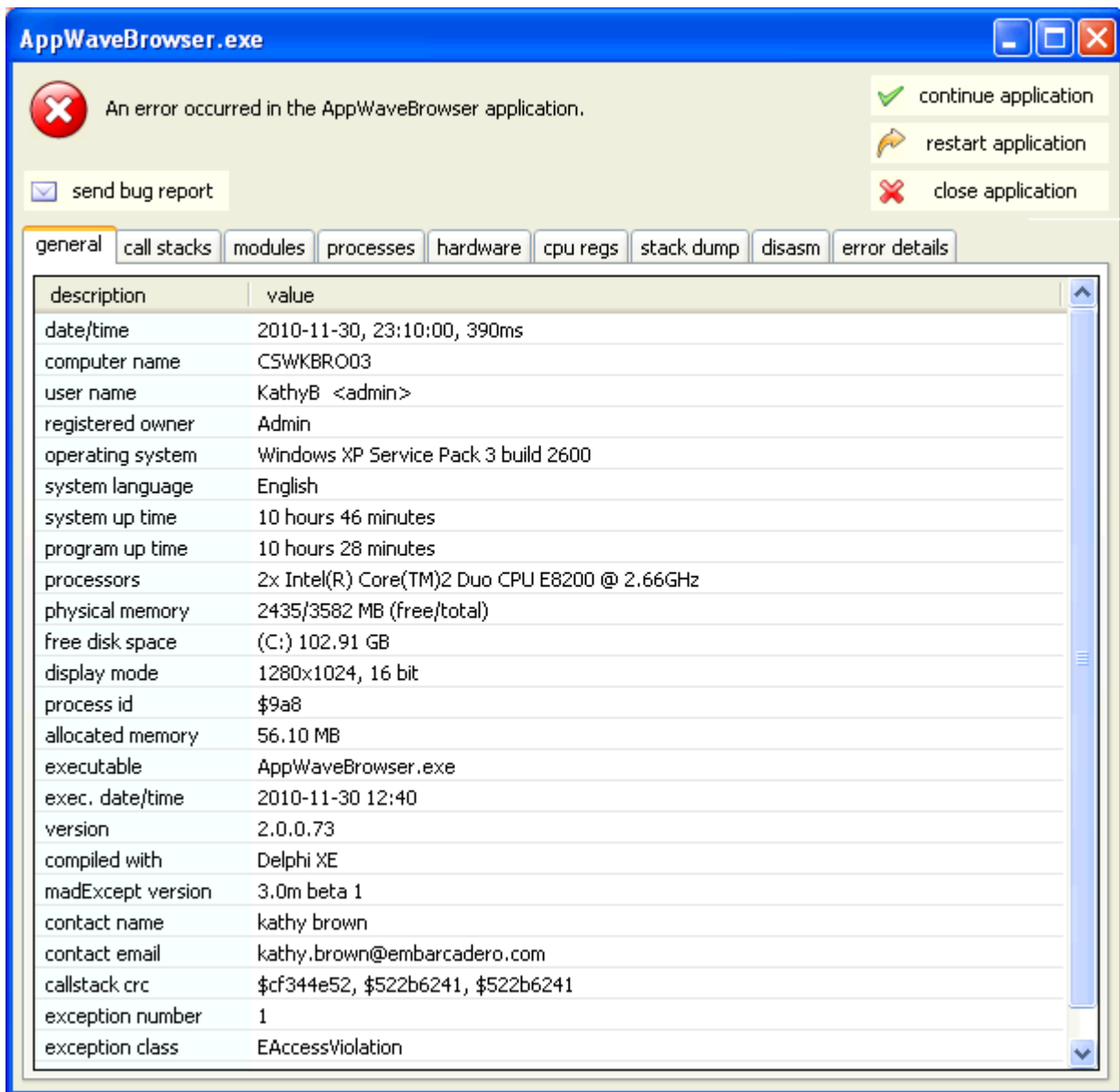
- [Stack Dump](#)

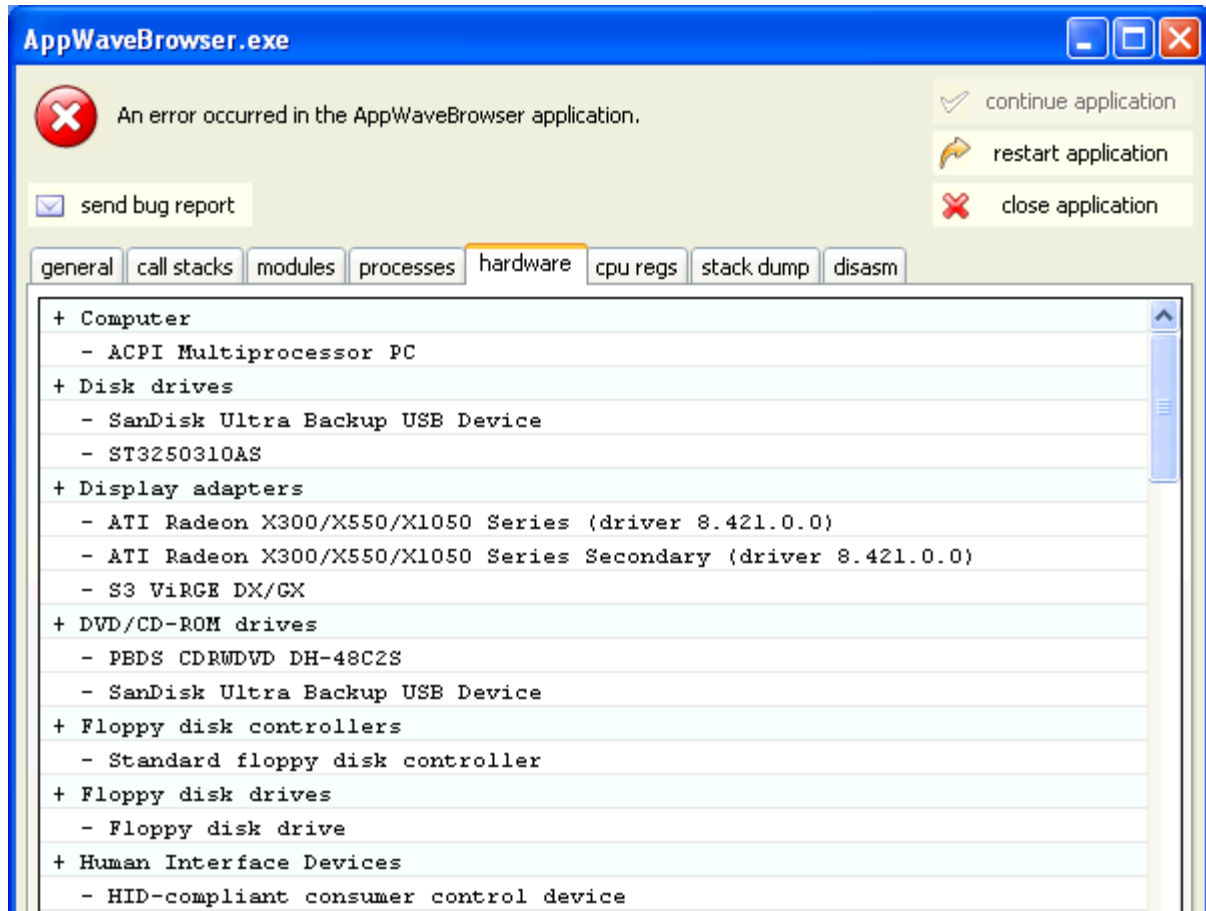- [Disasm](#)

# GENERAL

## CALL STACKS



## MODULES

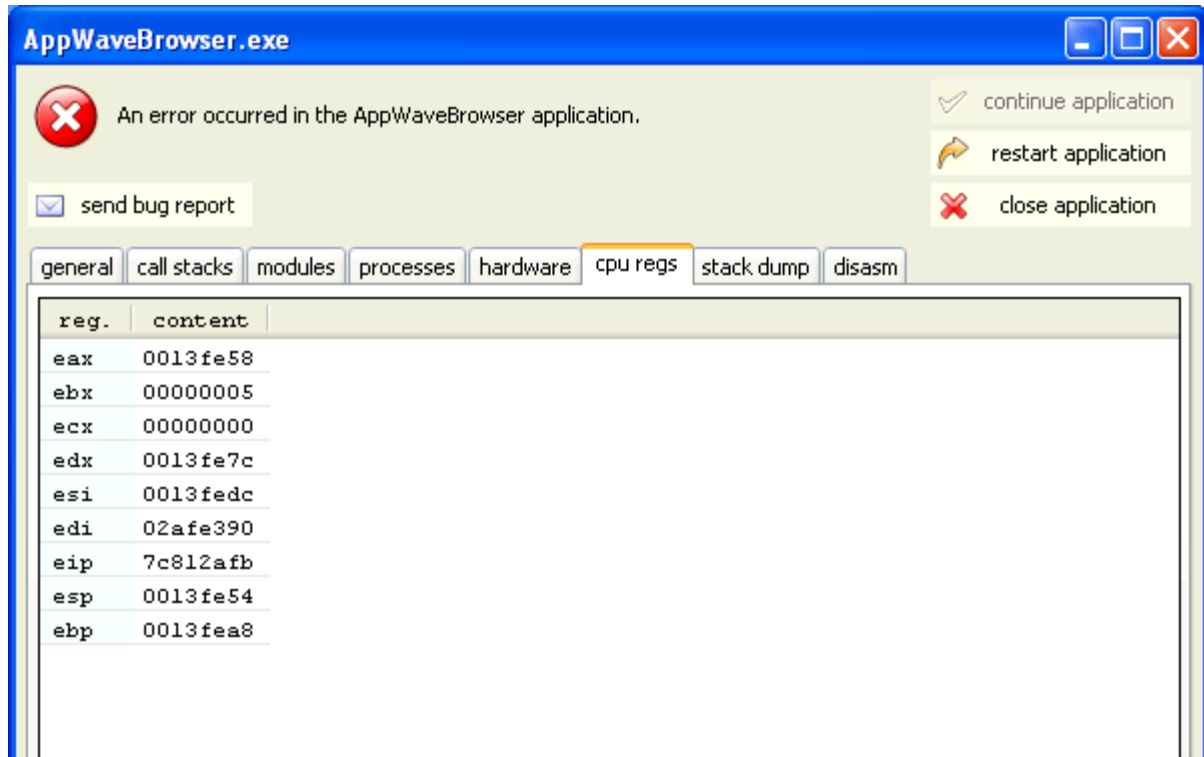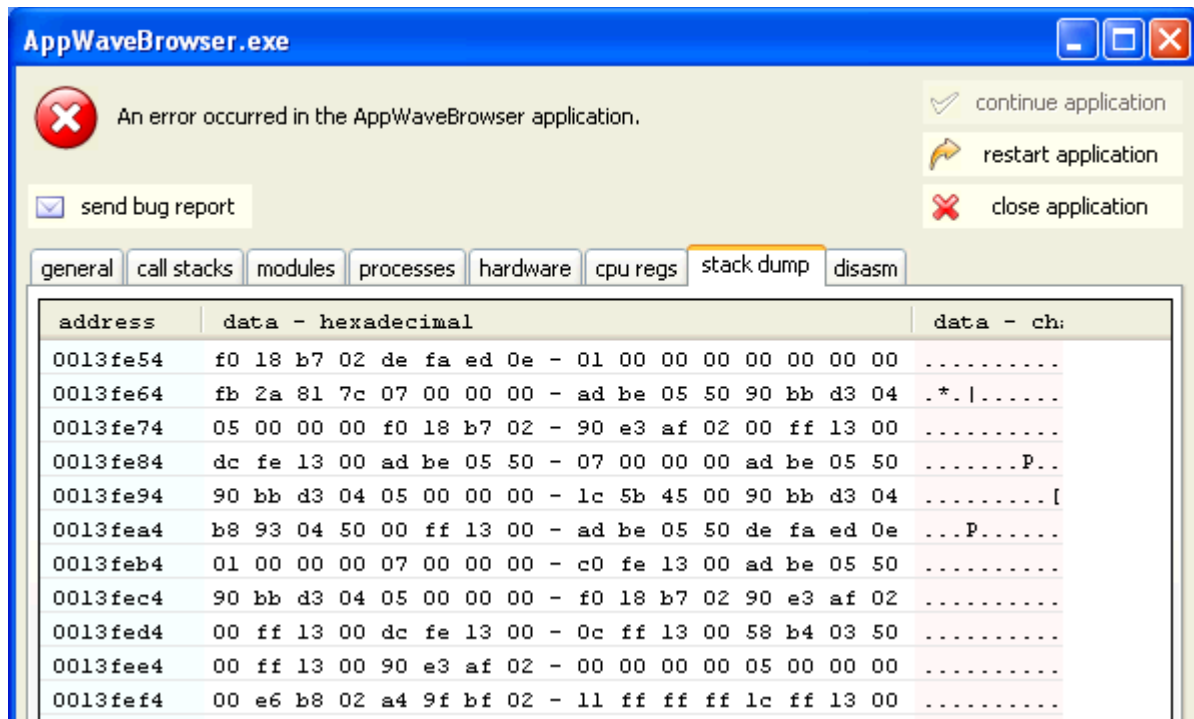## PROCESSES

## HARDWARE

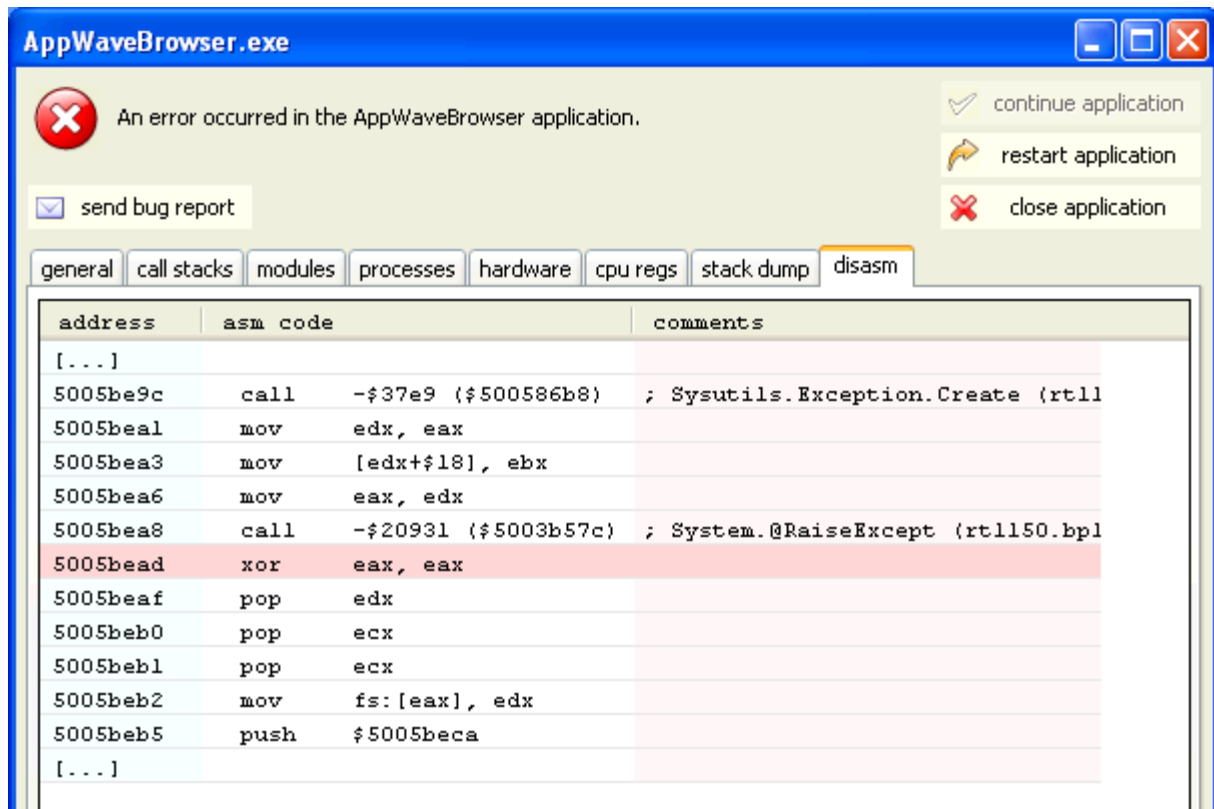## CPU R<small>EGS</small>

## STACK DUMP



## DISASM

This bug report displays malicious behavior or vulnerable code in binary files.

# APPWAVE STUDIO ADMINISTRATOR

The Administrator plays a key role in the AppWave Studio and has the following functions and responsibilities:

| Function | Description |
|---|---|
| Administrative Functions | Has access to all the administrative functions in AppWave. The Administrator can access the functionality through a web browser or through the Admin tab in the AppWave Browser. |
| Group Management | Has the option to give users access to the AppWave Studio so they can master applications. This mastering permission designates a user as an "Author" and gives the user the access. |
| Managing Licensing | Manages license inventory, assignments, and connected users. Also ensures the correct information is available for Licensing Third Party Applications. |
| Updating Apps | You can manage your applications so you always have the latest version. |

## ADMINISTRATIVE FUNCTIONS

This topic is a general overview of the Administrative Functions. For complete information on each function please refer to the *AppWave Administrator Guide.*

| Function | Description |
|---|---|
| Manage Users and Groups | Users access the product repository using AppWave Browser, to cache, run and upgrade products. You control the levels of access that users have to products on an individual or group basis. |
| Manage Folders | Manage Folders allows you to restrict users and groups to a certain set of products available on AppWave. |

| Function | Description |
| --- | --- |
| Manage Apps | Applications from Embarcadero are stored in the AppWave repository; the set of products is called the product catalog. The product catalog helps you manage product caching and versions. |
| Manage Mastering Plugins | PADE Plug-ins are stored in the AppWave repository and are automatically downloaded from the Embarcadero AppWave when the catalog is updated. If you don't have Internet connection, you can also manually update the plug-ins from the Version Updates page. |
| Manage Licenses | AppWave stores product licenses and shares them with users who cache the products using AppWave Browser. |
| System Management | Through the AppWave system management panes you can configure system properties such as what users can see when they connect, how applications are cached and streamed, proxy server details, along with streaming server details and backup licensing host configuration. |
| Reporting | Once you have the server set up and running you can gather statistics to help you manage users, groups, and licenses through the Reports. |

# GROUP MANAGEMENT

As Administrator you have the authority to assign users to groups. Before a user can access the Studio application in the AppWave Browser they have to be assigned to the "Authors" group.

To assign a user to the Authors group:

1   Open the AppWave Host, click Admin and login as administrator.

2   Click **Users** in the **Manage Users & Groups** folder.

3   On the **Users** page, search for name of the user you want to add.



4   Click the user's name and their page opens. Be sure to click on the user's name, do not
    select the check box.
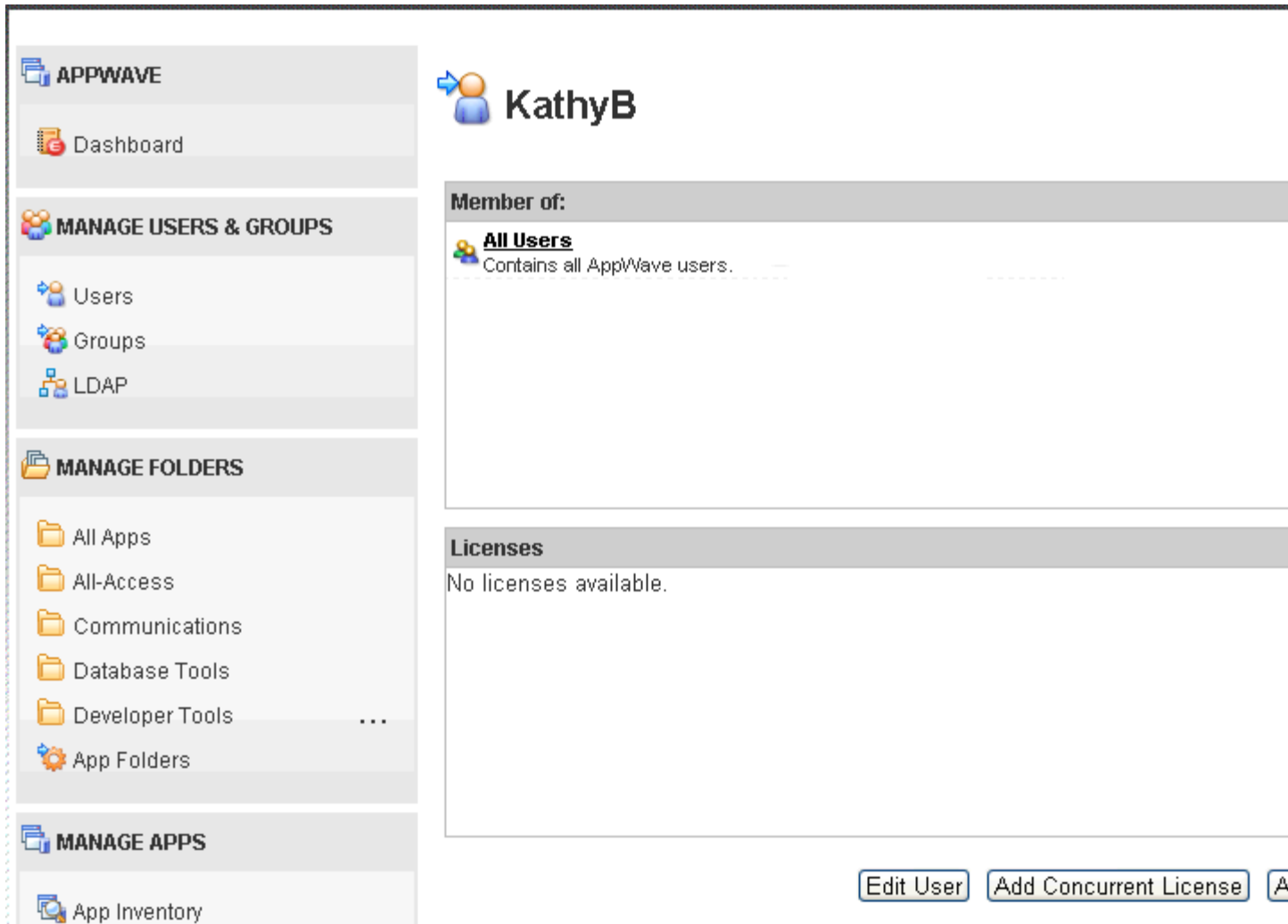
5    Click **Edit User** and the **User Details** page opens.

6    Click **Add Groups** and the **Users and Groups** page opens.

7   Select **Authors** and click **Select**. The user is now listed as a member of the Authors group.

8   Click **Save User** to save your changes.

For more information, please refer to the **Group Management** topic in the *AppWave Administrator Guide.*

## MANAGING LICENSING

The Administrator manages licenses for the Apps that you master and use. If a license exists for the mastered App, it is automatically activated as you walk through the install process.

For complete information on each license type, please refer to the **License Management** topic in the *AppWave Administrator Guide.*

## ADDITIONAL INFORMATION

Any app mastered by a vendor will only ever accept a vendor license. If, however the customer masters a supplied vendor App, then that mastered app can use either a vendor license or an AppWave Product License.

- For free Apps where there is no charge for support, a vendor license can be distributed. The owner (vendor) can then choose to take ownership of the app and the responsibility of maintaining it.

- If a customer has existing licenses from the vendor and wants to upgrade them into AppWave, this requires an AppWave Product License as well.

- A customer has the liberty to use a Vendor license like an AppWave Product License. However, the reverse is not true, because an AppWave Product License cannot grant entitlement to contained applications.

# LICENSING THIRD PARTY APPLICATIONS

When the AppWave Studio user masters a premium application, an application for which Embarcadero provides a mastering template, and then adds the application to AppWave, licenses for the mastered application can be managed by AppWave. AppWave can ensure license compliance by limiting the number of concurrent users to the maximum number of licenses available. Licensing third-party applications is accomplished through the License Automation pages of AppWave.

1   Log onto AppWave as an Administrator.

2   Click **Manage Licenses > License Automation** and all mastered third party applications that support license management are listed.



3   Locate the application to manage and in the **Usage** column, click the edit icon ( ✏ ) and the Licensing information for the application appears.

4    To add a new license or to change the information for an existing license, click **Add Activation Value**. To change the usage information for an existing license, in the Usage column for the relevant license, click the edit icon ( 🖊 )



NOTE:    The fields that appear on this page depend on the application. It may be as simple as the total count of licenses and serial number; or it may ask for company ID, user ID, email information, etc.

5    Adding licenses:

- To add one license, in the **Total Count** field, enter 1 and in the Serial ID field, enter a valid license number.

- To add more than one license click **Bulk Import**, point to the zip file containing the licensing slip files, and then click **Import**.

6    Click **Save** and the Auto Activation Licenses page now shows the number of licenses used and available. It is updated as licenses are used.

When the AppWave Browser user launches a licensed third-party application managed by AppWave, if licenses are available to the user, a dialog opens providing them with a serial number for their use.

For more information on the various licenses, see [Managing Licensing](#).

# UPDATING APPS

You can also ensure you always have the latest version of your applications by subscribing to AppWave notifications and by checking the Notifications folder in AppWave Browser, so you will always know when a new version is available.

For more information see: *Keeping your Apps Current* in the *AppWave Browser Apps User Guide*.

# APPWAVE STUDIO COMPLEX/INTEGRATED APPLICATIONS

With AppWave you can master complex applications (for example, Microsoft Office, Visual Studio, Autodesk Design Suites, Adobe CS5, etc.). Suites, or larger product "bundles", require a deeper level of integration with the operating environment. Once mastered, common components are leveraged and are exposed within the Windows UI as a collection of executables, right-click menu extensions, and file associations.

Mastering complex applications involves four areas:

| Action | Description |
|---|---|
| **File Associations / Shell Integration** | With file association, you have the ability to launch AppWave applications using the file format typically associated with that application. You can also access the functionality typically made available within the right-click menu of a file type.<br><br>It also covers the proximity and visual presence of the AppWave Browser when launching an app directly via file association, license checking, etc.<br><br>File associations and shell integration are performed automatically by AppWave Studio. |
| **Multiple Entry Points** | With Multiple Entry Points you have the ability to:<br>a) Create an application template that includes entry points for multiple applications in a suite/product bundle.<br><br>b) Present multiple application entries via the AppWave Browser from a single App. This may be required for tweaking the branding for an application. |
| **Custom Scripting** | Custom Scripting allows you to define actions/processes to be executed before/during/after certain types of events, for example, setup network connection to a particular network drive before launching an App, or do automatic archiving of certain files on exiting an application. |
| **Admin Elevation** | This is the ability to run Apps that require administrative privileges but those privileges are not available. This behavior is behind the scenes and does not require any action on your part. |

# MASTERING A COMPLEX APP

As an Author, you have the ability to define custom actions at key points during the life of an App. For example:

- before/during/after an app is added to a machine,

- before/during/after an app is loaded,

- before/during/after running an App, etc.

For each of the six known "Trigger Points", you can define the following:

- Trigger name

- Frequency

- Script to execute

- Event

You can enable/disable the defined triggers as well as edit parameters for already created triggers. This functionality is general and affects all entry points created in the app file. It shall be available on the "4. App" section on a new Custom Scripts tab in the Studio Wizard.


# MASTERING A SUITE APPLICATION

The following steps walk you through mastering a suite application such as Microsoft Office: Multiple Entry

1   After the initial Installation Recording process, advance to the  step to select from a list of multiple launch (entry) points discovered during the install process.

    Depending on the application being mastered, this list can be quite large and some knowledge of the application may be req uired in order to select the right launch executable. The list of launch points is made available from a drop-down.

2   Once a launch point has been identified, the rest of the page containing the branding (metadata) information for that launch (entry) point fill in the remaining fields.

3   Upon Save, the page refreshes and the top of the page expands to include the defined launch point in a list (see screenshot below). Author can continue to section 3. **app** or Add New launch point until all required launch points have been defined.

4   The definition of these individual launch points will be exposed in the AppWave Browser as if they were all individual app files with their own separate metadata. They appear in the Catalog exactly the same way as other Apps.

## RUNNING A COMPLEX APP

This section covers the launch of an app from an icon or Start menu using the File Association / Right-click menu options.

When an app is run within AppWave for the first time, your created files become associated with the app. This is similar to how file associations are established for locally installed software.

The Program Group and entry points for the app populate the Start menu (see screenshot below).
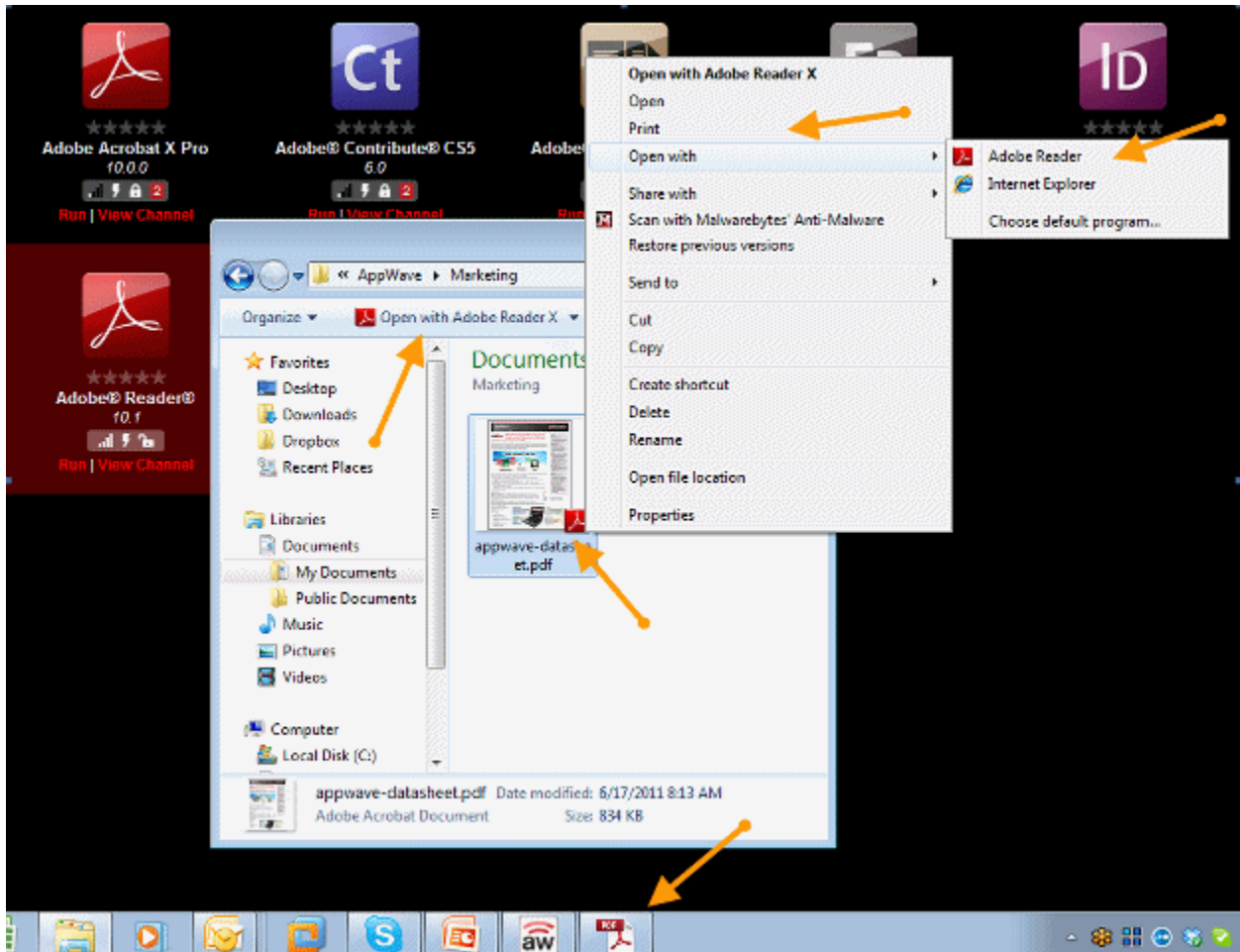


The app also integrates and exposes additional commands into the right-click menu, such as Print, Show, Preview, Send To, etc.

Once the initial file association occurs, you can double-click a document file type and the associated app launches. For example when you double-click a doc/docx file, Microsoft Word launches.

Similarly, if you right-click the doc/docx file, you are presented with options relevant to that file, such as Print, which will open Word and then directly open the Print dialog.

When you exit the App, which has established file associations, the file associations persist.

If you exit AppWave Browser, the file associations are removed.

When an app is launched from a user-created document directly or using the right-click menu options (outside the AppWave Browser), the app does a required license check. If a license is not available (trial has expired or license has otherwise been revoked), the AppWave Browser informs you about the lack of the license.

If a .DOC file is clicked before Word has been run in the AppWave Browser, nothing happens because the associations are not created yet.

If an app is removed from your local cache or the cache is cleared, all File Associations, all Program entries, and Right-click menu options are removed/cleared.

It is possible to run Integrated/Complex Apps without Admin privileges. The recommended install for the AppWave Browser is "Turbo On" mode, but you can execute all apps without Admin privileges. Where this is not possible, during the mastering process the author can specify that the user's rights should be elevated to administrator while running the app. The admin elevation occurs automatically for the user at app launch time.

## DISCOVERY OF APPS IN THE CATALOG

An Integrated/Complex app exposes its entry points in the AppWave Browser as if they were just regular single apps. For example, for Microsoft Office, the entry points in the Catalog would be Word, PowerPoint, and Excel, with their respective icons, metadata descriptions, screenshots, etc.

The individual items of a "suite product" are searchable and can be listed/tiled like any other app in the catalog. No special/automatic grouping of these apps are required.

The metadata content is created in such a way that if you do search for Microsoft Office, all related apps are displayed.

It is possible to bookmark individual apps in the usual way so they are available via your own channel.

You create an app desktop shortcut even though the app is already created in the Start menu's All Program group.

You can "pin an app" (Windows 7) to the taskbar (where the app allows it).

Ratings and Reviews are available for each individual app defined by a launch point in the app file.

# APPWAVE STUDIO APPENDIX

- [Predefined Virtual Folders](#)

- [Install and Uninstall Folder "CleanUp"](#)

- [Files](#)

- [Log Files](#)

- [Stopping Services](#)

- [App Mastering Acceptance Criteria](#)

- [Template Acceptance Criteria](#)

- [Error Messages](#)

## PREDEFINED VIRTUAL FOLDERS

When modifying registry keys it is important to know the system variables. You need to know what variables can be used that will be replaced with the correct path when an app is loaded on a machine.

The following tables list the physical paths for Windows XP, Windows Vista, and Windows 7. This enables the user to check the differences in folder visualization and investigate issues more effectively.

- [Table 1: Template Mapping Samples](#)

- [Table 2: Remarks Table](#)

- [Table 3: Directory Templates for #Defines](#)

- [Table 4: Variable Path Substitution](#)

- [Table 5: System X64 Directory Templates](#)

- [Table 6: Special Windows 7 Folders](#)

# TABLE 1: TEMPLATE MAPPING SAMPLES

| Template Name | Directory Template | Default Path - XP | Default Path - Vista/ Windows 7/Windows 2008 |
|---|---|---|---|
| AppData | ?roamingappdata? | C?\Documents and Settings\<username>\ Application Data | C:\Users\<username>\AppData\Roaming |
| CommonAdminTools | ?commonadmintools? | C:\Documents and Settings\All Users\Start Menu\Programs\Administrative Tools | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Administrative Tools |
| CommonDesktop | ?publicdesktop? | C:\Documents and Settings\All Users\Desktop | C:\Users\Public\Desktop |
| CommonDocuments | ?publicdocuments? | C:\Documents and Settings\All Users\Documents | C:\Users\Public\Documents |
| CommonDownloads | ?publicdownloads? | N/A* | C:\Users\Public\Downloads |
| CommonGamesTasks | ?publicgametasks? | N/A* | C:\ProgramData\Microsoft\Windows\GameExplorer |
| CommonMusic | ?publicmusic? | C:\Documents and Settings\All Users\Documents\ Music | C:\Users\Public\Music |
| CommonOEMLinks | ?commonoemlinks? | C:\Documents and Settings\All Users\OEM Links | C:\ProgramData\OEM Links |
| CommonPictures | ?publicpictures? | C:\Documents and Settings\All Users\Documents\ Pictures | C:\Users\Public\Pictures |
| CommonProgramFiles | ?programfilescommon? | C:\Program Files\Common Files | C:\Program Files\Common Files |
| CommonProgramFiles X64 | ?programfilescommonx64? | See Remarks Table on page 104 | See remark |
| CommonProgramFiles X86 | ?programfilescommonx86? | See Remarks Table on page 104 | See remark |
| CommonSidebarDefaultParts | ?sidebardefaultparts? | N/A* | C:\Program Files\Windows Sidebar\Gadgets |
| CommonStartMenu | ?commonstartmenu? | C:\Documents and Settings\All Users\Start Menu\Programs | C:\ProgramData\Microsoft\Windows\StartMenu\Programs |
| CommonStartup | ?commonstartup? | C:\Documents and Settings\All Users\Start Menu\Programs\StartUp | C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp |

| Template Name | Directory Template | Default Path - XP | Default Path - Vista/Windows 7/Windows 2008 |
|---|---|---|---|
| CommonTemplates | ?commontemplates? | C:\Documents and Settings\All Users\Templates | C:\ProgramData\Templates |
| CommonVideos | ?publicvideos? | C:\Documents and Settings\All Users\Documents\Videos | C:\Users\Public\Videos |
| LocalAppData | ?localappdata? | C:\Documents and Settings\<username>\Local Settings\Application Data | C:\Users\<username>\AppData\Local\ |
| LocalizedResourcesDir | ?localizedresourcesdir? | C:\Windows\resources\0409 (code page) | C:\Windows\resources\0409 (code page) |
| ProgramData | ?programdata? | C:\Documents and Settings\All Users\Application Data | C:\ProgramData |
| ProgramFilesX64 | ?programfilesx64? | See Remarks Table on page 104 | See remark |
| ProgramFilesX86 | ?programfilesx86? | See Remarks Table on page 104 | See remark |
| PublicDir | ?public? | N/A* | C:\Users\Public |
| ResourceDir | ?resourcedir? | C:\Windows\Resources | C:\Windows\Resources |
| SampleMusic | ?samplemusic? | C:\Doucments and Settings\All Users\Documents\Music\Sample Music | C:\Users\Public\Music\Sample Music |
| SamplePictures | ?samplepictures? | C:\Documents and Settings\All Users\Documents\Pictures\Sample Pictures | C:\Users\Public\Pictures\Sample Pictures |
| SamplePlaylists | ?sampleplaylists? | N/A* | C:\Users\{ib;oc\Music\Sample Playlists |
| Sample Videos | ?samplevideos? | N/A* | C:\Users\Public\Videos\Sample Videos |
| SystemDir | ?system? | C:\Windows\System32 | C:\Windows\System32 |
| System Fonts | ?fonts? | C:\WIndows\Fonts | C:\Windows\Fonts |
| SystemX86Dir | ?systemx86? | See Remarks Table on page 104 | see remark |
| UserAdminTools | ?admintools? | C:\Documents and Settings\<username>\Start Menu\Programs\Administrative Tools | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Administrative Tools |

| Template Name | Directory Template | Default Path - XP | Default Path - Vista/ Windows 7/Windows 2008 |
|---|---|---|---|
| UserCDBuring | ?cdburning? | C:\Documents and Settings\<username>\Local Settings\Application Data\Microsoft\CD Burning | C:\Users\<username>\AppData\Local\Microsoft\Windows\Burn\Burn |
| UserContacts | ?contacts? | N/A* | C:\Users\<iserma,e?\Contacts |
| UserCookies | ?cookies? | C:\Documents and Settings\<username>\Cookies | C:\Documents and Settings\<username>\Cookies |
| UserDesktop | ?desktop? | C:\Documents and Settings\<username>\Desktop | C:\Documents and Settings\<username>\Desktop |
| UserDocuments | ?documents? | C:\Documents and Settings\<username>\ Documents | C:\Documents and Settings\<username>\ Documents |
| UserDownloads | ?downloads? | N/A* | C:\Users\<username>\Downloads |
| UserFavorites | ?favorites? | C:\Documents and Settings\<username>\Favorites | C:\Documents and Settings\<username>\Favorites |
| UserGameTasks | ?gametasks? | N/A* | C:\Users\<username>\AppData\Local\Microsoft\Windows\game Explorer |
| UserHistory | ?history? | C:\Documents and Settings\<username>\Local Settings\History | C:\Documents and Settings\<username>\Local Settings\History |
| UserInternetCache | ?internetcache? | C:\Documents and Settings\<username>\Local Settings\Temporary InternetFiles | C:\Documents and Settings\<username>\Local Settings\Temporary InternetFiles |
| UserLinksDir | ?links? | N/A* | C:\Users\<username>\Links |
| UserLocalAppDataLow | ?localappdatalow? | N/A* | C:\Users\<username>\AppData\LocalLow\ |
| UserMusic | ?music? | C:\Documents and Settings\<username>\ Documents\ Music | |
| UserNetworkShortcuts | ?nethood? | C:\Documetns and Settings\<username>\NetHood | C:\Users\<username>\AppData\Roaming\Microsoft\WIndows\Network Shorts |
| UserOriginalImages | ?originalimages? | N/A* | C:\Users\<username>\AppData\Local\Microsoft\Windows Photo Gallery\Original Images |
| UserPhotoAlbums | ?photoalbums? | N/A* | C:\Users\<username>\Pictures\Slide Shows |
| UserPictures | ?pictures? | C:\Documents and Settings\<username>\ Documents\ Pictgures? | C:\Users<nusername>\Pictures |

| Template Name | Directory Template | Default Path - XP | Default Path - Vista/ Windows 7/Windows 2008 |
|---|---|---|---|
| UserPrinterShortcuts | ?printhood? | C:\Documents and Settings\<username>\Printhood | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Printer Shortcuts |
| UserProfile | ?profile? | C:\Documents and Settings\<username>\ | C:\Users\<username> |
| UserProfiles | ?userprofiles? | N/A* | C:\Users |
| UserPrograms | ?programs? | C:\Documents and Settings\<username>\Start Menu\Programs | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs |
| UserQuickLanunch | ?quicklaunch? | C:\Documents and Settings\All Users\Application Data\Microsoft\Internet Explorer\Quick Launch | C:\Users\<username>\AppData\Roaming\Microsoft\Internet Explorer\Quick Launch |
| UserRecent | ?recent? | C:\Documents and Settings\<username>\Recent | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent |
| UserSavedGames | ?savedgames? | N/A* | C:\Users\<username>\Saved Games |
| UserSearches | ?savedsearches? | N/A* | C:\Users\<username>\Saved Searches |
| UserSendTo | ?sendto? | C:\Documents and Settings\<username>\SendTo | C:\Users\<username>\AppData\Roaming \Microsoft\Windows\SendTo |
| UserStartMenu | ?startmenu? | C:\Documents and Settings\<username>\Start Menu | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Program**S** |
| UserStartup | ?startup? | C:\Documents and Settings\<username>\Start Menu\StartUp | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start |
| UserTemplates | ?templates? | C:\Documents and Settings\<username>\Templates | C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Templates |
| UserVideos | ?videos? | C:\Documents and Settings\<username>\ Documents\ Videos | C:\Users\<username>\Videos |
| Windoes | ?windows? | C:\Windows | C:\Windows |

**\* N/A means not supported on the platform.**

## TABLE 2: REMARKS TABLE

| Directory Template | Operating System | Default Path |
|---|---|---|
| ?programfilesx86? | 32 bit | C:\Program Files |
| | 64-bit | C:\Program Files (x86) |
| ?programfilesx64? | 32-bit | N/A* |
| | 64-bit | C:\Program Files |
| ?programfilescommonx86? | 32-bit | C:\Program Files\Common Files |
| | 64-bit | C:\Program Files (x86)\Common Files |
| ?commonprogramfilesx64 | 32-bit | N/A* |
| | 64-bit | C:\Program Files\Common Files |
| ?system? | 32-bit | C:\Windows\System32 |
| | 64-bit | C:\Windows\System32 |
| ?system86? | 32-bit | C:\Windows\System 32 |
| | 64-bit | C:\Windows\Syswow64 |

**\* N/A means not supported on the platform.**

## TABLE 3: DIRECTORY TEMPLATES FOR #DEFINES

| #define | Directory Template |
|---|---|
| #define S_DT_CDDRIVE | T(S_DT_CDDRIVE_A)  // Point to the first CD drive available |
| #define S_DT_FONTS | ?fonts? |
| #define S_DT_DESKTOP | ?desktop? |
| #define S_DT_STARTUP | ?startmenu? |
| #define S_DT_RECENT | ?recent? |
| #define S_DT_SENDTO | ?sendto? |
| #define S_DT_DOCUMENTS | ?documents? |
| #define S_DT_FAVORITES | ?favorites? |
| #define S_DT_NETHOOD | ?nethood? |

| #define | Directory Template |
|---|---|
| #define S_DT_PRINTHOOD | ?printhood? |
| #define S_DT_TEMPLATES | ?templates? |
| #define S_DT_COMMONSTARTUP | ?commonstartup? |
| #define S_DT_COMMONPROGRAMS | ?commonprograms? |
| #define S_DT_COMMONSTARTMENU | ?commonstartmenu? |
| #define S_DT_PUBLICDESKTOP | ?publicdesktop? |
| #define S_DT_PROGRAMDATA | ?programdata? |
| #define S_DT_COMMONTEMPLATES | ?commontemplates? |
| #define S_DT_PUBLICDOCUMENTS | ?publicdocuments? |
| #define S_DT_ROAMINGAPPDATA | ?roamingappdata? |
| #define S_DT_LOCALAPPDATA | ?localappdata? |
| #define S_DT_LOCALAPPDATALOW | ?localappdatalow? |
| #define S_DT_INTERNETCACHE | ?internetcache? |
| #define S_DT_COOKIES | ?cookies? |
| #define S_DT_HISTORY | ?history? |

## TABLE 4: VARIABLE PATH SUBSTITUTION

These are VIRTUAL/UNUSED Directory templates:

| #define | Directory Template |
|---|---|
| S_DT_NETWORKFOLDER | ?networkfolder? |
| S_DT_COMPUTERFOLDER | ?computerfolder? |
| S_DT_INTERNETFOLDER | ?internetfolder? |
| S_DT_CONTROLPANELFOLDER | ?controlpanelfolder? |
| S_DT_PRINTERSFOLDER ? | ?printersfolder? |
| S_DT_SYNCMANAGERFOLDER | ?syncmanagerfolder? |
| S_DT_SYNCSETUPFOLDER | ?syncsetupfolder? |
| S_DT_CONFLICTFOLDER | ?conflictfolder? |
| S_DT_SYNCRESULTSFOLDER | ?syncresultsfolder? |
| S_DT_RECYCLEBINFOLDER | ?recyclebinfolder? |
| S_DT_CONNECTIONSFOLDER | ?connectionsfolder? |
| S_DT_CHANGEREMOVEPROGRAMS | ?changeremoveprograms? |

| #define | Directory Template |
|---|---|
| S_DT_APPUPDATES | ?appupdates? |
| S_DT_ADDNEWPROGRAMS | ?addnewprograms? |
| S_DT_TREEPROPERTIES | ?treeproperties? |
| S_DT_GAMES | ?games? |
| S_DT_RECORDEDTV | ?recordedtv? |
| S_DT_SEARCH_MAPI | ?search_mapi? |
| S_DT_SEARCH_CSC | ?search_csc? |
| S_DT_USERFILES | ?userfiles? |
| S_DT_SEARCHHOME | ?searchhome? |

## TABLE 5: SYSTEM X64 DIRECTORY TEMPLATES

Only 64-bit appsets will have files under this template directory. Windows does not have a separate folder identification for "systemX64",and that is why we don't want to rename the template to "systemX64". The advantage is that if two templates resolve to the same directory path, the template name that is longer can be chosen. If we rename this template to "systemX64", the application installed on a 32-bit machine will have its files incorrectly captured to "systemX64" template directory.

| #define | Directory Template |
|---|---|
| #define S_DT_SYSTEMX64 | ?system? |
| #define S_DT_SYSTEMX86 | ?systemx86? |
| #define S_DT_WINDOWS | ?windows? |
| #define S_DT_PROFILE | ?profile? |
| #define S_DT_PICTURES | ?pictures? |
| #define S_DT_PROGRAMFILESX86 | ?programfilesx86? |
| #define S_DT_PROGRAMFILESCOMMONX86 | ?programfilescommonx86? |
| #define S_DT_PROGRAMFILESX64 | ?programfilesx64? |
| #define S_DT_PROGRAMFILESCOMMONX64 | ?programfilescommonx64? |
| #define S_DT_PROGRAMFILES | ?programfiles? |
| #define S_DT_PROGRAMFILESCOMMON | ?programfilescommon? |
| #define S_DT_ADMINTOOLS | ?admintools? |

| #define | Directory Template |
|---|---|
| #define S_DT_COMMONADMINTOOLS | ?commonadmintools? |
| #define S_DT_MUSIC | ?music? |
| #define S_DT_VIDEOS | ?videos? |
| #define S_DT_PUBLICPICTURES | ?publicpictures? |
| #define S_DT_PUBLICMUSIC | ?publicmusic? |
| #define S_DT_PUBLICVIDEOS | ?publicvideos? |
| #define S_DT_RESOURCEDIR | ?resourcedir? |
| #define S_DT_LOCALIZEDRESOURCESDIR | ?localizedresourcesdir? |
| #define S_DT_COMMONOEMLINKS | ?commonoemlinks? |
| #define S_DT_CDBURNING | ?cdburning? |
| #define S_DT_USERPROFILES | ?userprofiles? |
| #define S_DT_SAMPLEPLAYLISTS | ?sampleplaylists? |
| #define S_DT_SAMPLEMUSIC | ?samplemusic? |
| #define S_DT_SAMPLEPICTURES | ?samplepictures? |
| #define S_DT_SAMPLEVIDEOS | ?samplevideos? |
| #define S_DT_PHOTOALBUMS | ?photoalbums? |
| #define S_DT_PUBLIC | ?public? |
| #define S_DT_DOWNLOADS | ?downloads? |
| #define S_DT_PUBLICDOWNLOADS | ?publicdownloads? |
| #define S_DT_SAVEDSEARCHES | ?savedsearches? |
| #define S_DT_QUICKLAUNCH | ?quicklaunch? |
| #define S_DT_CONTACTS | ?contacts? |
| #define S_DT_SIDEBARDEFAULTPARTS | ?sidebardefaultparts? |
| #define S_DT_PUBLICGAMETASKS | ?publicgametasks? |
| #define S_DT_GAMETASKS | ?gametasks? |
| #define S_DT_SAVEDGAMES | ?savedgames? |
| #define S_DT_LINKS | ?links? |
| #define S_DT_ORIGINALIMAGES | ?originalimages? |

## TABLE 6: SPECIAL WINDOWS 7 FOLDERS

These special folders exist only on Windows 7 or later.

| #define | Directory Template |
|---|---|
| #define S_DT_W7_DEVICEMETADATASTORE | ?w7_devicemetadatastore? |
| #define S_DT_W7_DOCUMENTSLIBRARY | ?w7_documentslibrary? |
| #define S_DT_W7_IMPLICITAPPSHORTCUTS | ?w7_implicitappshortcuts? |
| #define S_DT_W7_LIBRARIES | ?w7_libraries? |
| #define S_DT_W7_PICTURESLIBRARY | ?w7_pictureslibrary? |
| #define S_DT_W7_PUBLICLIBRARIES | ?w7_publiclibraries? |
| #define S_DT_W7_PUBLICRINGTONES | ?w7_publicringtones? |
| #define S_DT_W7_RECORDEDTVLIBRARY | ?w7_recordedtvlibrary? |
| #define S_DT_W7_RINGTONES | ?w7_ringtones? |
| #define S_DT_W7_USERPINNED | ?w7_userpinned? |
| #define S_DT_W7_USERPROGRAMFILES | ?w7_userprogramfiles? |
| #define S_DT_W7_USERPROGRAMFILESCOMMON | ?w7_userprogramfilescommon? |
| #define S_DT_W7_VIDEOSLIBRARY | ?w7_videoslibrary? |

# INSTALL AND UNINSTALL FOLDER "CLEANUP"

When you uninstall and reinstall AppWave Studio, at times it is recommended to fully remove all the files and directories installed by AppWave. This ensures a clean starting place for your next install.

The following files should be removed for the AppWave Studio uninstall:

**Windows XP**

- C:\Documents and Settings\<user>\Local Settings\Application Data\Embarcadero\AppWaveBrowser\plugins

- C:\Documents and Settings\<user>\ AppWaveStudio\Output\ - the default Output folder path *can be changed by the user*

**Windows 7**

- c:\Users\<user>\AppData\Local\Embarcadero\AppWaveBrowser\plugins\

- c:\Users\<user>\Embarcadero\AppWaveStudio\Output\ - this directory *can be changed by the user*

**AppWave**

- Install directory -> c:\Program Files (x86)\Embarcadero\AppWave

- Application repository, only if you want to delete existing binaries -> c:\ProgramData\Embarcadero AppWave\Products

The following files should be removed for the AppWave Browser uninstall:

**On Windows XP:**

- Install directory - "C:\Documents and Settings\<username>\Local Settings\Application Data\Embarcadero\AppWaveBrowser"

- Repository - "C:\Documents and Settings\<username>\Local Settings\Application Data\Embarcadero\Product Repository"

- Streaming Core additional locations:

    - "C:\Documents and Settings\<username>\Application Data\Embarcadero\Product Repository"

    - "C:\Documents and Settings\All Users\Application Data\Embarcadero\StreamingCore"

**On Windows 7:**

- Install directory - "C:\Users\<username>\AppData\Local\Embarcadero\AppWaveBrowser"

- Repository - "C:\Users\<username>\AppData\Local\Embarcadero\Product Repository"

- Streaming Core additional locations:

    - "C:\Users\<username>\AppData\Roaming\Embarcadero\Product Repository"

    - "C:\Users\All Users\Embarcadero\Product Repository"

    - "C:\Users\All Users\Embarcadero\StreamingCore"

# FILES

The files listed below are the mandatory files to be delivered in the app process.

**App File**

.app is the filename extension for AppWave-Enabled application files. It is an application that was mastered with AppWave and must adhere to all of the App Mastering Acceptance Criteria.

App means that applications can be streamed from the AppWave and then run locally on the desktop without installation. This option also allows applications to be run on locked-down desktops. There are no conflicts with other applications, no writing to the system areas of their registry or modifying shared dlls. Application data such as Configuration and license information is stored locally on the hard disk, and is separate from the executable itself.

### .aw File

The AppWave Template file offers a model for mastering an application, and contains branding information and information on which filter to run on the installer for the application. It must respect all of the "Acceptance criteria".

The AppWave Template file (.AW) is created by the PADE Product Manager and is used to add Instant AppWave Mastering support to a server.

### MD5 File

Message-Digest algorithm 5 (MD5) is used to check the integrity of files. The .aw and the .app files must come together with the associated md5 message digest file. Each time one of the two files needs to be changed, the .md5 file must be changed too.

### .msi installer File

MSI files are database files, used by Windows Installer. They contain information about an application which is divided into features and components, and every component may hold files, registry data, shortcuts etc. The MSI file also contains the UI that is to be used for installing, and various other data such as any prerequisites to look for, custom actions to execute, the order of the installation procedure, whether to support Administrative installations, etc. MSI files are the current recommended way of doing installations on Windows.

The .msi files provided must have the ability to run in Unattended mode.

### .stp File

STP File is created after the author masters the application and can be found in the output folder. This file is the outer layer that encapsulates the other application components. It is encrypted and stores the files, folders, and registry settings needed to reproduce the application being mastered on the end users machine.

The .stp file includes:

| File Type | Description |
|---|---|
| Aib (application installation blueprint) | This file contains all the files' metadata (filename, file path, file size, etc) and registry information (registry keys and values) for the mastered application. |
| Stc (streaming content file) | The code and other configuration information that comprises most of the application. |
| Tok (token) | Token Files are comprised of security data and general information about the mastered application |
| Ico | icon file |

| File Type | Description |
|-----------|-------------|
| htm | EULA file |

None of the files contained in the .stp file need to be altered or deleted.

**Studio File**

The Studio or .tca file is the project file in the AppWave Browser.The .tca contains the registry and folder and files structure captured during a software installation along with resources, launch points, and registry settings required to run a product.It also contains information about the product such as the version, name, and editions. Every time a user needs to master an App, a .tca file is created.

# LOG FILES

Log files are the text files which contain all the log entries saved by AppWave and AppWave Browser.

AppWave Browser logs:

- StudioLog.log

- StreamingCore.log

- log4delphi.log

AppWave logs:

- error.log

- info.log

AppWave Browser Log Paths:

- **StudioLog.log:**  C:\Documents and Settings\All Users\Application Data\Embarcadero\Log\AppWaveBrowser\<user>\

- **StreamingCore.log:**  C:\Documents and Settings\All Users\Application Data\Embarcadero\Log\AppWaveBrowser\StreamingCore.log

- **log4delphi.log:**  C:\Documents and Settings\All Users\Application Data\Embarcadero\AppWaveBrowser

- C:\Program Files (x86)\Embarcadero\AppWave\4.0\logs

AppWave Log Paths:

- **error.log:** C:\Program Files\Embarcadero\AppWave\<version>\elc\logs

- **info.log**: C:\Program Files\Embarcadero\AppWave\<version>\elc\logs

# STOPPING SERVICES

During mastering, extra services related to the app you are mastering may be launched. At the end of the mastering process, Studio will attempt to shutdown any services it detects, but in some cases Studio cannot detect those services and you must shut them down manually before broadcasting the App.

You must be logged on as an administrator or a member of the Administrators group in order to complete this procedure. If your computer is connected to a network, network policy settings might also prevent you from completing this procedure.

**To start, stop, pause, resume, or restart a service:**

1 Open Control Panel > Administrative Tools > Services

2 In the details panel, right-click the service, and then click Start, Stop, Pause, Resume, or Restart.

**Important**

- If you stop, start, or restart a service, any dependent services are also affected.

- Changing the default service settings might prevent key services from running correctly. It is especially important to use caution when changing the Startup Type and Log On As settings of services that are configured to start automatically.

- To configure startup parameters for a service, right-click the service, click **Properties**, and then type the parameters in Start parameters before you click Start. These settings are used only once and are not saved. (A backslash (\) is treated as an escape character; type two back slashes for each back slash in a parameter.)

# APP MASTERING ACCEPTANCE CRITERIA

Required Mastering Criteria

Mandatory Fields

Conditionally Required Fields

Recommended Folders

**Required Mastering Criteria**

1 The app-mastering process must always be done with the original installer of the application and not with the converted one.

2 Auto-Update feature of the application must be disabled. If it cannot be disabled then verify that it does not give errors when updating.

3   Remove all empty folders, log files, installer artifacts, computer specific info or temporary files. Attention is needed when empty folders are used for writing files after the application is started;

Some .lnk files are needed by the application(links in help section, etc) so you must be careful when removing them.

When installing : always uncheck "Create shortcut on Desktop, Create Start menu entry", etc.

4   It is recommended that all Apps be mastered with Best Compression and No Encryption.

5   The size of the app file mustn't overcome the size that the native application has on disk;

> **NOTE:**   If you use Best Compression then the size should be approximately one-half the size that the application has on disk.

## Mandatory Fields

1   The filename must be lowercase characters only.  The App, md5 and Readme filename should be in the following format: ,productname><ProductVersion>.

2   The md5 file must match the app file.  If you modify the app file you must rebuild the md5 file.

3   The Product Name and Display Name must respect the manufacturer's identity and trademark/copyright . For example if the name is RadRails on the website, the name must be written the same in the AppWave Browser.

4   Product Version must not contain build number. If no Product Version is specified in the installer or website then it will have the default value set to "1.0"

- Product Version must not contain build number. The accepted format is a.b.c.d (c  and d values are optional)
- Example:  Valid versions : 1.0, 1.0.1, 1.0.1.1;
- Invalid Versions: 1.0.1234, 1.0.1.12345 ( should be modified to be  1.0 and 1.0.1 );

5   The EULA file must take the filename of the .aib and .stc files.

If the application requires prerequisites (like dotNet Framework, MSVC 2005/2008, J#, MDExplorer, Adobe Flash, etc.) then the correspondent  EULA file should be included as well; product Eula must appear first.

6   A screenshot of the main window of the application must be in the .png format. You can check to see if it is the correction format two different places:

- Open the .tca file and verify the following information:

<ProductAppearance Icon="icon.png "><ScreenShots><ScreenShot Name="screen.png"/></ScreenShots></ProductAppearance>

- Open the metadata.xml file:

<version name="1.0"><screenshot url="screen.png"/><buildFileList>

7   The Product Description cannot containt copyrighted material.

8   The long description cannot contain copyrighted material and cannot contain the information in the Product Description.

9   Product Icon must be in a .png file format.

10  Product release date of the app file must match the one of the template.

**Conditionally Required Fields**

These are fields that must be completed if the information is available.

1   System Requirements:  mandatory to contain the name of the prerequisites if the app needs them; if no pre-requisites are needed then it will contain a phrase like "Windows XP and higher" and the information available on vendor's website.

2   Support Contacts

3   Edition

**Recommended Folders**

When mastering an application, it is best practice to install in a folder which is not on a network share.

# TEMPLATE ACCEPTANCE CRITERIA

1   The app executable used to build the template must pass all the App Mastering Acceptance Criteria.

2   All .Apps produced with the template delivered must launch on:

   • Windows 7 32-bit

   • Windows XP 32-bit

   • Windows 7 64-bit,

   **NOTE:**   No matter on which OS the .app was created, the template must work in the same manner on Windows 7 and Windows XP.

3   Naming standard:

   • The .aw file name must be set to productname.aw.

   • The File Name must be lowercase characters only.

   • The mastered application will not launch if certain non-ASCII characters are used. The unsupported characters are: ? (question mark),  : (colon),  \ (backslash),  | (pipe),  / (slash),  * (asterisk),  < (less than),  > (greater than), and  " (double quotes).

4    All the necessary plugins must be included in the template file.

# ERROR MESSAGES

This topic covers some of the more common error messages that may occur as you are using the Studio application.

**File Already Exists**
- If a .stp file already exists for the app you want to broadcast you get the following message:
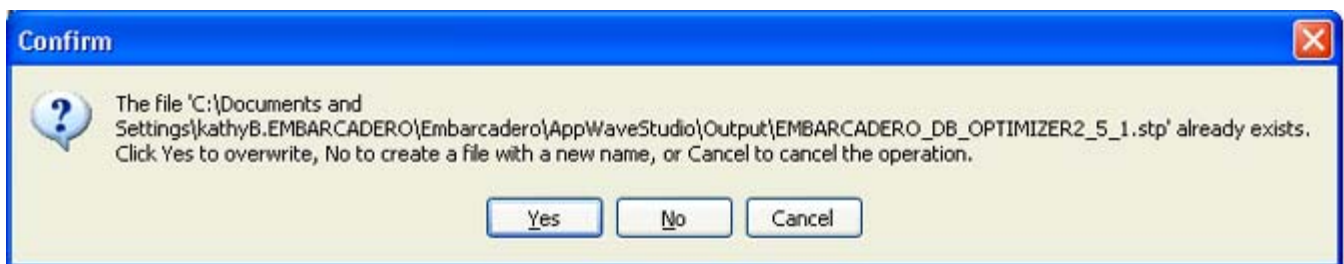


**Unable to Broadcast**
- If you try to broadcast an app that has the same name, version, and edition of an existing App, you get the following message:
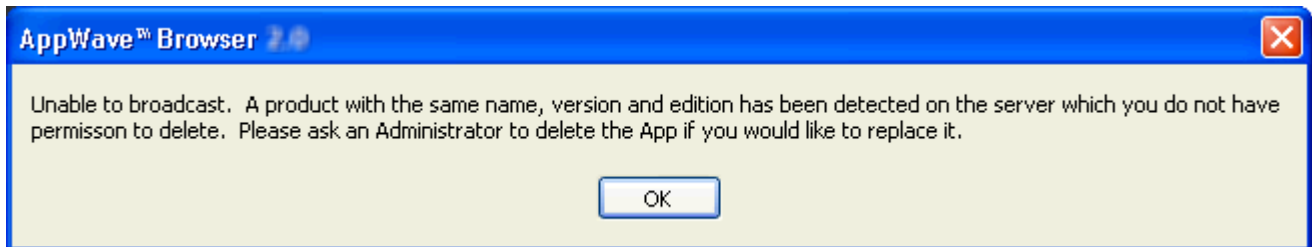


Before you can complete the mastering process, you must contact your Administrator to delete the existing App. When you click OK, you are returned to the Broadcast page.

**Running More than One App**
- Before mastering an app in AppWave Browser you must ensure that no other Apps are currently running. If there are any Apps running you will get the following warning dialog when you click Master:
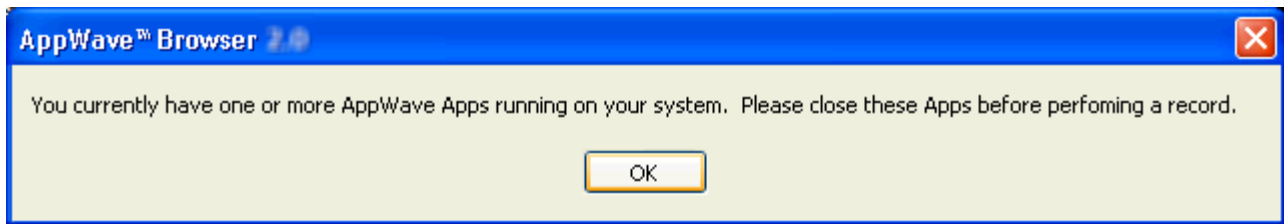
Close each app through its application interface, usually by choosing **File > Exit**.

# Index