

Release Notes

Last edited date: 5/21/01

Table of Contents

1. Universal	1
1.1. SmartCode	1
1.2. CSDChecker Log	1
1.3. Strpack.dll	1
2. Enterprise	1
2.1. Compile All Smart Code	1
2.2. Ctrl + Alt + V	2
3. WebPack	2
3.1. Chaining	2
3.2. CBWFormat HREF#	2
3.3. CBWFormat Check	2
3.4. Align Style	2
4. Teamwork	3
4.1. Checking in CSDs with Non Existing Records and Fields	3
4.2. TeamWork Dialog	3
4.3. Jscript Library File	3
5. Runtime	3
5.1. Close Application	3
6. AppServer	3
6.1. CBAgent	3
6.1.1. Logging Fixed	3
6.1.2. Scaled Down CBAgent (CBAgentM.csd)	3
6.2. Terminal Window Drawing	4
6.3. CBHost	4
6.4. Unexpected Screen Handler	4
6.5. ClientBuilder Services	4
6.6. Scheduler (cmdDispatcher)	4
6.6.1. Example using the 'at' command	5
6.6.2. Example using WinAt Utility	5
6.6.3. Example using Scheduled Tasks	5
6.7. WebPack – Overloading WebPack Connectors	6
7. Emulators	6
7.1. Hummingbird modifications 5250	6

7.2. 3270 Emulators Passing of shift characters	6
7.3. 5250 Emulators passing column characters.....	6
7.4. TN Trace	7
7.4.1. File Naming Convention	7
7.4.2. Level of trace	7

1. Universal

Items listed under the Universal section apply to all ClientBuilder products.

1.1. SmartCode

The **File** menu option that allows users to print SmartCode, JavaScript and/or VBScript has been renamed from **Print SmartCode** to **Print**.

1.2. CSDChecker Log

The text string size per entry in the CSDChecker log has been increased from 1K to 3K. This prevents a GPF from occurring when any individual line entry exceeds 1K.

1.3. Strpack.dll

When using the Strpack.dll, the developer had the option of replacing 'one' instance or 'all' instances of a particular string. However, replacing 'one' instance would cause a GPF. i.e.,

```
call "strpak.replace" with c.this_detail , "1", "xxx", "All"
```

To correct this, the code has been modified as follows:

```
<> = required  
[] = optional
```

```
Added: strpak.replace with  
<String1>, <String2>, <String3>, [(Attributes)]  
Returns the new string in c.result.  
String1 = main string, the string on which to perform the  
replace.  
String2 = replace this text  
String3 = replace with this text.  
Attributes = "All" or "one" Case insensitive (defaults to  
'one')
```

2. Enterprise

2.1. Compile All Smart Code

In the past if users selected **Rebuild all Smartcode** from the **Services** menu, an error message would be generated and the SmartCode would not compile. This has been corrected and the SmartCode will compile without displaying the error message.

2.2. **Ctrl + Alt + V**

When the JavaScript or VBScript editor is opened and in focus, depressing Ctrl + Alt + V no longer causes a GPF. It will open a combo box with the options for selecting either JavaScript or VBScript.

3. WebPack

3.1. Chaining

ClientBuilder WebPack does not support Chaining for the following reasons

1. **Resource Intensive:** In chaining, two or more CSD's are set up in a relay fashion to work with one host session. As such, for each host session working with WebPack, a group of two or more CSD's is required. Currently, ClientBuilder WebPack has a limit of 100 CSD's per AppServer. This, in essence, significantly restricts the number of host sessions available for chaining. For example, if two CSD's are used per host session, then only 50 host sessions will be supported by one AppServer
2. **Configuration.** The configuration for chaining requires each chained "client" CSD to be associated with its "parent" CSD. Currently, the Spawn feature automatically generates multiple instances of a CSD with 3 digit indexes that does not support the chained "parent" reference. For chaining to function properly, each "client" CSD would all have to be manually adjusted to reference the "parent".
3. **Activate Session.** Because chaining uses the "Activate Session" command to pass the host session activity from one CSD to another, multiple "Activate Sessions" being issued by groups of chained CSDs in an AppServer would result in unpredictable behavior. Chaining and Activate Session were originally designed for one group per ClientBuilder environment only.

3.2. **CBWFormat HREF#**

When using the HREF# command, the HTML output would display links, but not in a table format. This has been corrected and using HREF# will display the links in a table format.

3.3. **CBWFormat Check**

The fourth parameter of **CBWFormat, check** is designed to display the default value selected.

Inserting a default value in the fourth parameter instructs the HTML page to display a check in box adjacent to the entry which corresponds to the fourth parameter. If you do not insert a default value, the HTML page will display all four checkboxes as unchecked.

3.4. **Align Style**

In earlier releases, the align attribute did not function properly. This has been corrected in version 6.2.2

4. Teamwork

4.1. *Checking in CSDs with Non Existing Records and Fields*

In the past, if a developer attempted to check an object into the Master.csd with undefined fields/records, they would get a GPF. This has been fixed. Now when a file with undefined fields/records is checked into a Master.csd, the undefined field/records are removed.

4.2. *TeamWork Dialog*

Since the **Done** button on the TeamWork Group Dialog window was never enabled, it has been removed from the window.

4.3. *Jscript Library File*

RowCount, which is a JavaScript version of a SmartCode function, has been streamlined to improve performance. The improvement in performance will be most notable if using the RowCount function with a large string buffer or memory file.

5. Runtime

5.1. *Close Application*

There was an issue with ClientBuilder Runtime where it would not completely shut down the application. Although the user interface would close, the application would continue to run in the background and not shut down completely.

This has been corrected and shutting down ClientBuilder Runtime will completely close and shut down the application.

6. AppServer

6.1. *CBAgent*

6.1.1. **Logging Fixed**

The DoPoll global function was invoking some logging logic irrespective of whether logging was on or off. . This has been corrected.

6.1.2. **Scaled Down CBAgent (CBAgentM.csd)**

A scaled down CBAgent (CBAgentM.csd) is available for expert users.

There is now available a scaled down CBAgent which does not have as many status windows as the full scaled CBAgent. By having fewer status windows the CPU will not have to process and refresh the status tracking information. This allows for more efficient processor time usage and

application optimization. The scaled down version is not for novice users as the tracking information is minimal.

6.2. Terminal Window Drawing

The Cbrt.ini file default setting has been changed to provide the ability to disable the drawing of the terminal window. Disabling this feature improves application performance.

The setting may be changed as follows:

```
Cbrt.ini file
[Window]
EnableShowTerminal=0

[CBSESSIONXXX_DEFAULT]
SHOWTERMINAL=NO
```

The default setting is NO.

6.3. CBHost

To further improve performance, users now have the ability to turn off Dynamic GUI. Having the Dynamic GUI logic enabled in the Cbrt.ini file (even when auto GUI terminal is not specified) may affect application performance. This option may now be turned off in the Cbrt.ini file as follows:

```
[DynaGUI]

EnableDynaGUI=0

To enable DynaGUI, set it to 1
```

6.4. Unexpected Screen Handler

The WebPack Shell has been modified so that it will write a message to the Event Log whenever the Unexpected Screen Handler executes.

6.5. ClientBuilder Services

In the past, if the AutomaticRestart option in the CBSERVER.ini file was set to 'Yes', it would cause the CBRT.exe file to re-launch after a user shut down ClientBuilder Services. This has been corrected, whereby shutting down ClientBuilder Services will not automatically re-launch CBRT.exe

6.6. Scheduler (cmdDispatcher)

Setting the internal scheduler (cmdDispatcher) will shut down ClientBuilder AppServer only. It does not stop ClientBuilder Service. To shut down Services, you must use the command line (or one of Window's utilities such as WinAt.)

Below are examples of how to use scheduling options with Windows Services and WinAt. These are examples only. We suggest that you read the Windows Help files for complete instructions.

6.6.1. Example using the 'at' command

The following examples are provided to show the various ways to set scheduling options.

```
at 11:00 CBCmdDispatcher -p 6400 OpenAllSessions
```

The above command will set AppServer to open all sessions defined in the project associated with IP port 6400 at 11:00 am daily.

```
at 17:00 CBCmdDispatcher -p 6400 CloseAllSessions
```

The above command will set AppServer to close all sessions defined in the project associated with IP port 6400 at 5:00 pm daily.

6.6.2. Example using WinAt Utility

Using the above example scheduling the same session through WinAt would be as follows:

1. Select **Add** from the **Edit** menu
2. In the Command textbox within the Add Command dialog enter:

```
CBCmdDispatcher -p 6400 OpenAllSessions
```
3. Select the frequency in the '**This Occurs**' section: Today, Tomorrow, Every or Next
4. In the **Days** textbox select either the days of the week, or dates of the month for the application to run
5. Select the time to open the application in the **Time on Client** section. WinAt uses a 12 hour clock, so you would have to check PM if the time set is afternoon.
6. Click **OK** to set the time

If you need further information regarding the use of the "at" command in the WinAT Command Scheduler, as well as other ways to interact with it, please consult your Windows Help files.

6.6.3. Example using Scheduled Tasks

To schedule a session through Windows Scheduled Task System Tool perform the following steps:

1. Select **Programs** from the Windows **Start** menu
2. Select **System Tools** from the **Accessories** menu
3. Select **Scheduled Tasks** which will invoke the **Schedule Tasks Window**
4. Double click **Add Scheduled Task** to add a new task. This will invoke the **Scheduled Task Wizard**.

5. Click **Next** and select the ClientBuilder WebPack application that you want windows to run.
6. Click **Next** and insert a name for the task or accept the default selection and select the time interval for which it should run, i.e. daily, weekly, monthly, etc
7. Click **Next**.
8. Select the frequency for which the task should run and click Next
9. Depending on the time frequency selected in step 6, insert the appropriate time, date and month information and click **Next**.
10. Insert the Username and Password click **Next**
11. Select **Finish**

For more information scheduling tasks through Windows, please refer to the Windows Help files.

6.7. WebPack – Overloading WebPack Connectors

When used under heavy loads, ClientBuilder WebPack occasionally gave unexpected results.

If ClientBuilder WebPack sessions reached oversubscribed or near oversubscribed conditions, requests submitted through WebPack connectors would result in unexpected error message.

This issue has been resolved

7. Emulators

7.1. Hummingbird modifications 5250

When Hummingbird 6.2 modified their emulator software to include COM, it caused conflicts with ClientBuilder software. ClientBuilder software has been modified and the conflicts have been resolved.

7.2. ClientBuilder 3270 Emulators Passing of shift characters

There were occasions when the host screen would accept special characters (1234567890 -./,\$^%*()), however the 3270 emulators failed to pass the special characters to the ClientBuilder application. This has been fixed and the special characters are now passed to the ClientBuilder application.

7.3. ClientBuilder 5250 Emulators passing column characters

The host screen would accept the column separators of '[' and ']', however, these characters were not being translated and passed to the ClientBuilder application when using 5250 emulators. This has been fixed and the column characters will now be passed to the ClientBuilder application.

7.4. TN Trace

7.4.1. File Naming Convention

The TN trace files will no longer carry the name 'CBHLLAPI'. The name will now consist of the CSD name appended with the file extension '.trc'. For example, a TN trace file for a csd entitled 'persistence' will be 'persistence.trc.'

The trace files are located in the directory where the ClientBuilder application was installed.

7.4.2. Level of trace

The ini files (Winboth.ini and Cbrt.ini) may be modified to provide four different levels of tracing. You have the option to set varying degrees of trace from minimal (1) to complete (Max) tracing. By inserting the following code within the Winboth.ini or Cbrt.ini file under the CBSESSION3270_DEFAULT or CBSESSION5250_DEFAULT, you can set your trace level.

Option:	Level
Add Trace = 1	First Level
Add Trace = Min	Second Level
Add Trace = Mid or Add Trce = Yes	Third Level
Add Trace = Max	Fourth Level
Trace = 0	Disables the tracing option

Keep in mind that the more information traced, the larger the file becomes. Large trace files will slow down ClientBuilder's response time