

# CASE Tools 3.7.1 Enhancements/Bug Fixes

SBS-CASE Tools™ 3.7.1 contains the following enhancements and bug fixes.

## Enhancements

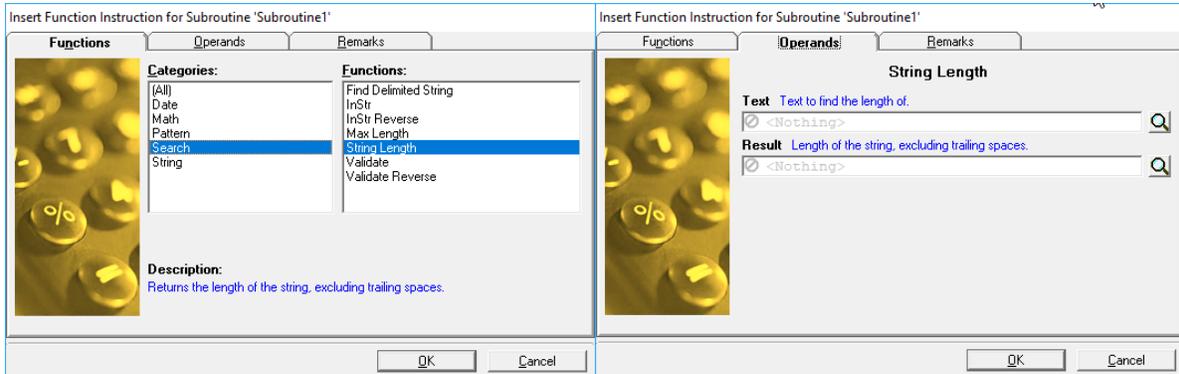
- Length Function
- Check Version Control
- Labels and Text Boxes support word wrap
- Change control attributes without showing panel
- .WAV files
- Queued transactions behavior
- Projects calling projects

## Bug Fixes

- Rules validation
- GoTo Label Length
- Panels – Case sensitivity

# Length Function:

A new “Length” function was added to the “Search” categories of the function instruction.



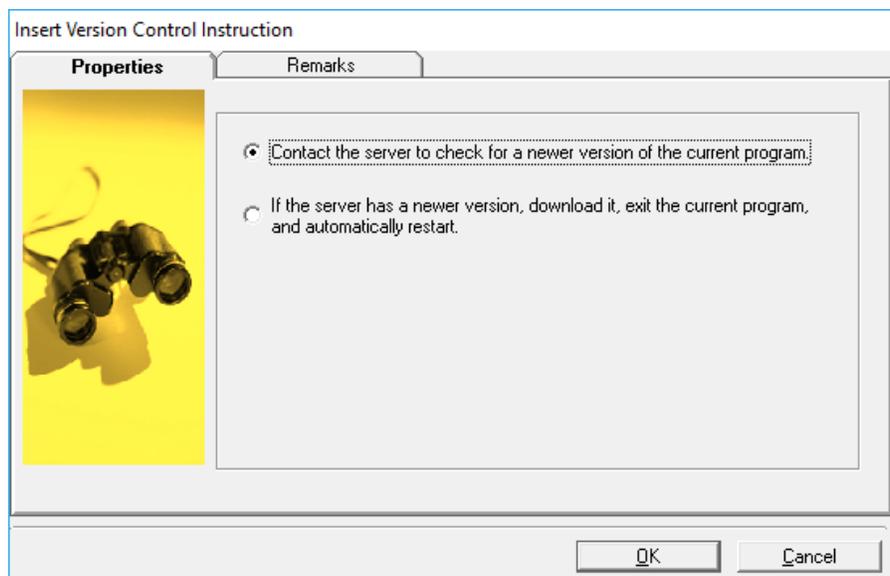
The programmer must provide the name of the variable to be checked and the instruction will return to you the length of the contents of that variable. For example, a Badge ID is defined as 10 characters long. The operator scans their badge which contains the value “ABCDEF”. The results of the length instruction on this scan would be “6”.

## Check Version Control:

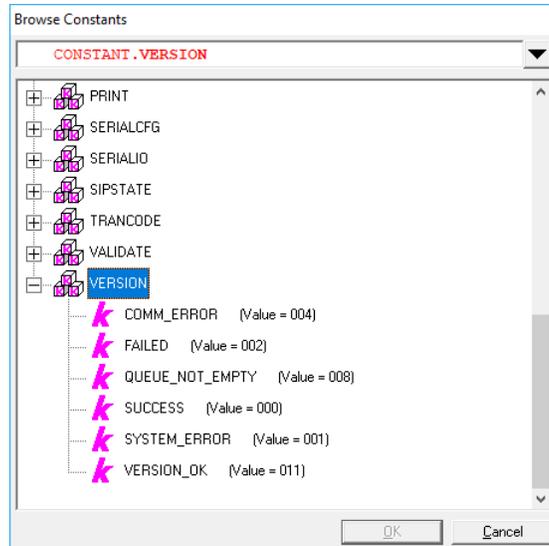
The product has had version control in it for over 2 decades. It is a very old instruction but it always demanded that a newer version be downloaded if one existed. It never gave the programmer the ability to check for a newer version and then download it if they choose to. The other weakness was that if a newer version existed and it started to upgrade to the latest version, the data collection device appeared to be hung to the user but in reality it was just very busy getting the latest version of the application. This situation has also been remedied in 3.7.1. A new system structure named "ServerProgramVersion" was created that will contain the server version number. This information previously was not available to the programmer.

### Checking Only:

To just check to see if a newer version exists the programmer should select the first option "Contact the server to check ...". After the instruction is executed then you should check the system constants associated with "version control" to see if a newer version is available.



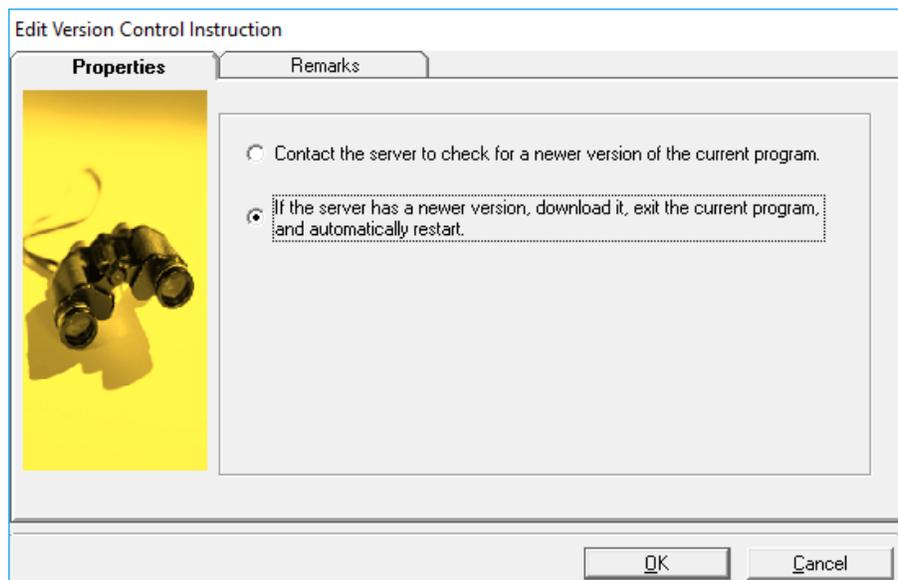
The properties tab allows you to select the option of just checking for a new version or downloading a newer version if it exists.



If a newer version of the application is available then the constant value “Success” will be returned in the System.ErrorNumber variable. If no newer version is available then the constant “Version\_OK” will be returned. The new system structure “ServerProgramVersion” is also updated with the version of the application that has been published to SBS\_Pathfinder.

### Update Version Control:

To download a newer version of the application (if it exists) the programmer should select the second option “If the server has a newer version ...”. After the instruction is executed if you get control back at the next instruction then a newer version was not available. If a newer version was available, then the new version will be downloaded and started and control will start at the top of the application.



If a newer version of the application is available then the constant value "Success" will be returned in the System.ErrorNumber variable. You will not be able to query it because the new version will be downloaded and started. The instruction following the version control will never be executed in that scenario. If no newer version is available then the constant "Version\_OK" will be returned. The new system structure "ServerProgramVersion" is also updated with the version of the application that has been published to SBS\_Pathfinder.

### SBS-CASE Tools™ code snippet

```
/* Check for a newer version - Do NOT apply it
Get Latest Version, Check Only

If SYSTEM.ErrorNumber = CONSTANT.VERSION.VERSION_OK Then
    /* OK - No newer version is available
Else If SYSTEM.ErrorNumber = CONSTANT.VERSION.SUCCESS Then
    /* Success - A newer version is available
    /* Inform operator a new version will now be downloaded
Else If SYSTEM.ErrorNumber = CONSTANT.VERSION.QUEUE_NOT_EMPTY Then
    /* QueueNotEmpty - Version control is not available why application is in Queue mode
Else
    /* ALL Else - Communication/Connectivity Issues - No Version Control Done
End If

/* The system structure "ServerProgramVersion" has been updated

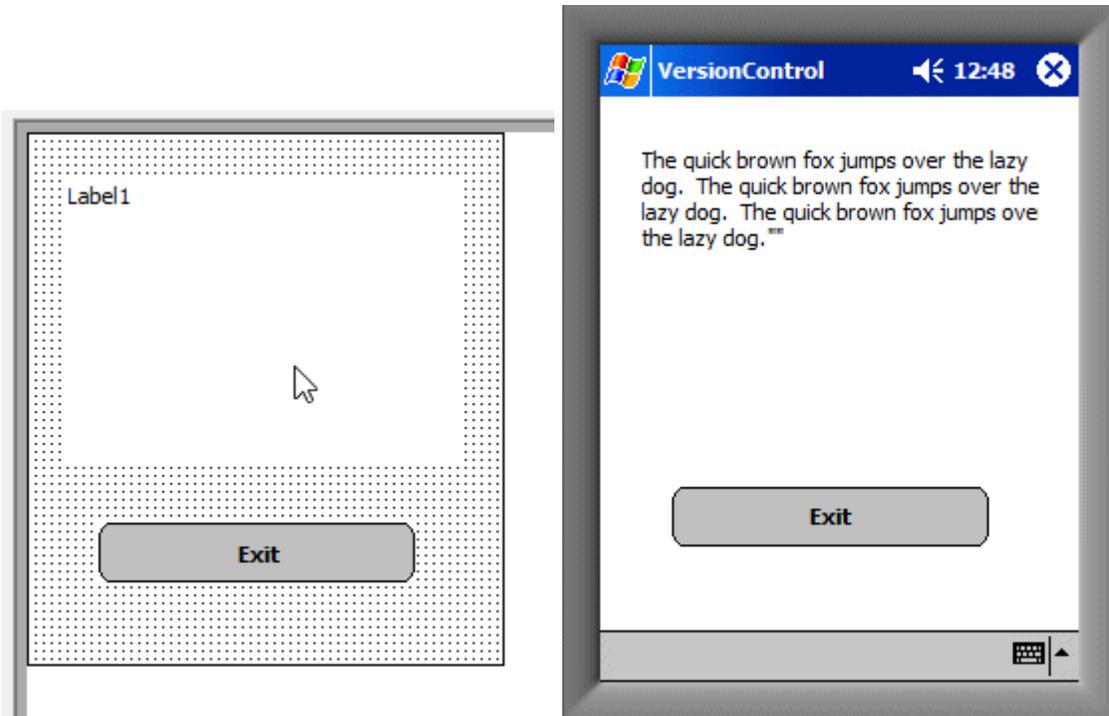
/* Check for a newer version - Apply it if applicable
Get Latest Version

If SYSTEM.ErrorNumber = CONSTANT.VERSION.VERSION_OK Then
    /* OK - No newer version is available
Else If SYSTEM.ErrorNumber = CONSTANT.VERSION.SUCCESS Then
    /* Success - New version was downloaded and started
    /* Since the new version was downloaded and started, thjis instruction will never be executed.
Else If SYSTEM.ErrorNumber = CONSTANT.VERSION.QUEUE_NOT_EMPTY Then
    /* QueueNotEmpty - Version control is not available why application is in Queue mode
Else
    /* ALL Else - Communication/Connectivity Issues - No Version Control Done
End If
```

## Word Wrap:

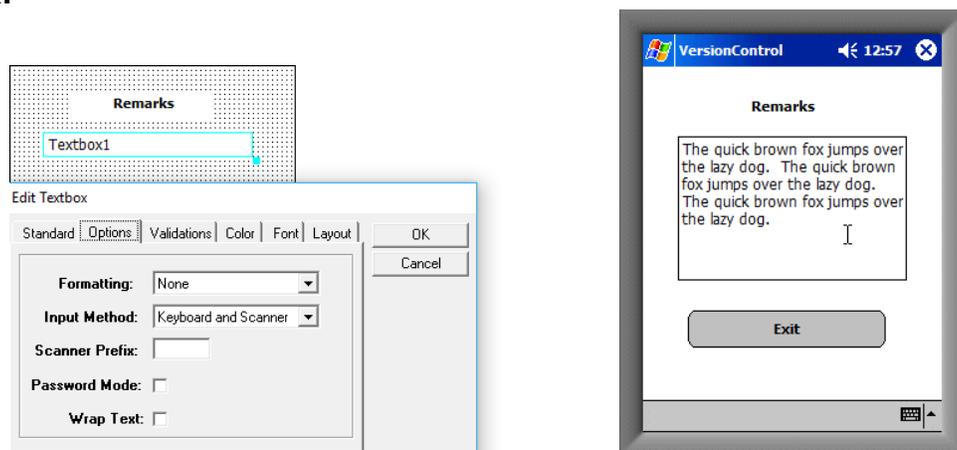
Both the label and the text box control now support word wrap.

### Label:



You must design your label control on the form to occupy the desired space (1 to x lines) and then if the data to be displayed won't fit on a single line, it will be wrapped for you onto another line. The word wrap will only occur for the size of the label that you designed. The label size will not grow automatically.

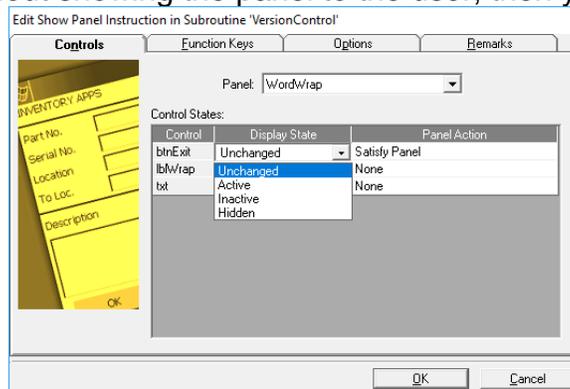
### Text Box:



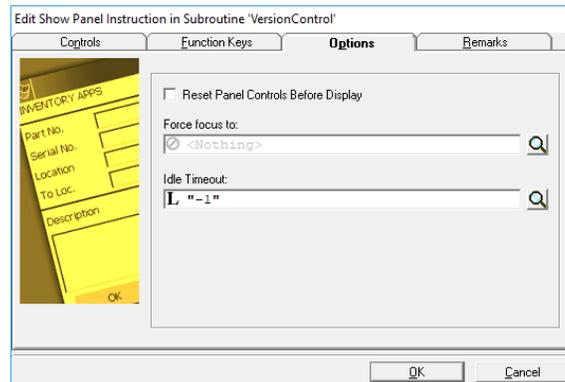
If the programmer desires to have text wrapped into a text box, then they must click "Wrap Text" on the "Options" tab. Once it is selected then the operator can resize the text box to the desired length.

## Change Control Attribute without sending Panel:

If you have the need to change a display state (Active, Inactive or Hidden) on a control on a specific panel without showing the panel to the user, then you must:



1. Change the "Display State" on 1 or more of the controls on the panel.

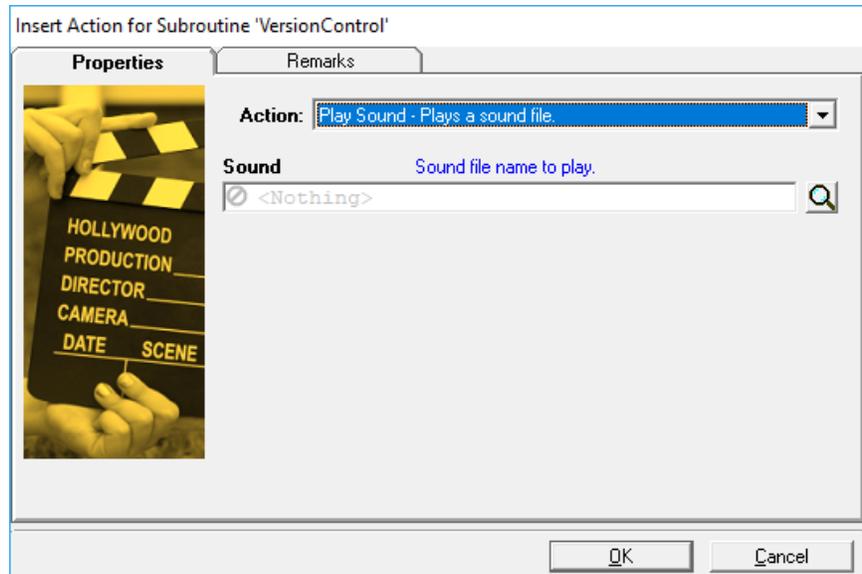


2. Set the "idle timeout" on the "Options" tab to "-1".

In the past without this option, if you toggled the state on a few controls then you would see a flicker on the panel as the entire panel is refreshed as the display state of the individual control changed. At this time, this enhancement was only performed in the TIP driver program. If you were to test your project in the debugger, the panel will still show and then disappear quickly. At a future point in time this will be implemented in the debugger as well.

## .WAV Files:

The programmer can now play custom .WAV files that have been created and downloaded to the data collection device.

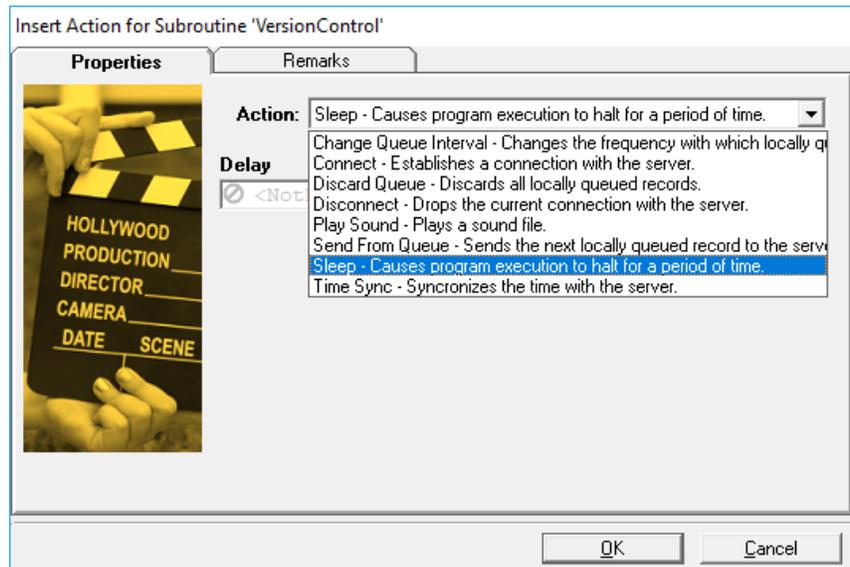


The name and the optional path of the .WAV file must be provided on the “Action” instruction with an “Action” of “Play Sound – Plays a sound file”.

If no path is provided, the default folder for the .WAV file is the folder that the TIP driver is installed in. If you are testing using “debugger” and a path wasn’t provided, the default file folder is the folder where the .TIP is located.

## Queued Transactions Behavior:

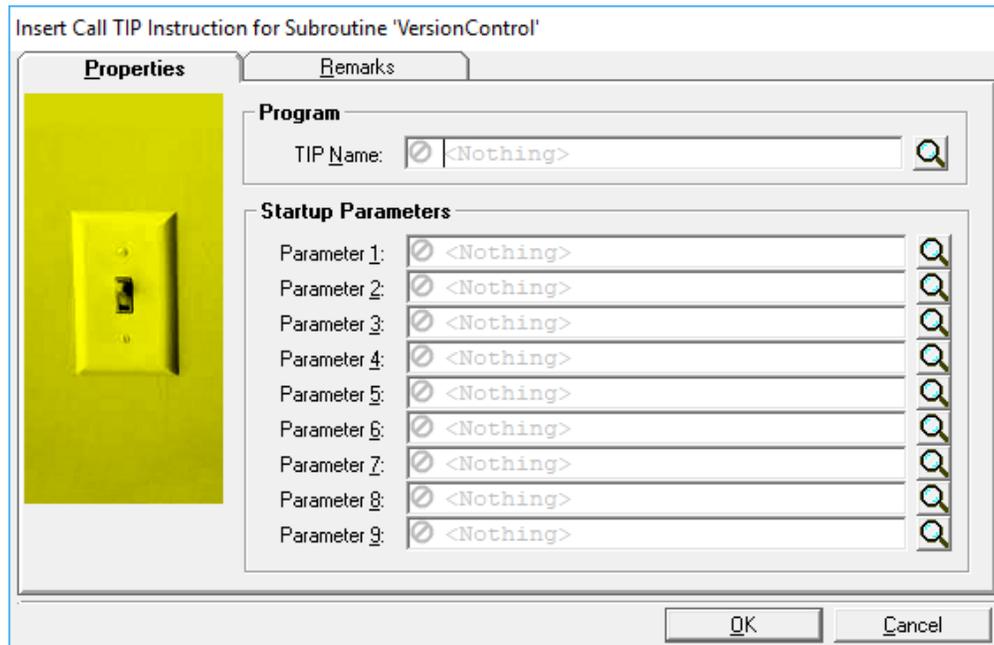
Prior to release 3.6 of SBS-CASE Tools the emptying of transactions located in the queue file was handled automatically by the product itself. The suggestion was made regarding the programmer having control over the queued transactions. The methodology was changed in 3.6 and the programmer was given full control over the queued transactions using the “Action” instruction. This gave the programmer the ultimate in flexibility for control over the queued transactions although it also disabled the automatic processing of locally queued transactions. This broke backwards compatibility with earlier versions. This release restores the backwards compatibility as well as leaves all the functionality for the programmer to have complete control.



This version adds a new updateable system variable names “SYSTEM.QueueInterval”. The default value for this variable is 30 seconds. This means that the TIP Driver every 30 seconds will try to relieve a row from the queue if queued transactions exist. You may change this value up or down if you are not satisfied with the way that the queue is being relieved.

## Project Calling Another Project:

Since the inception of the product back in the 1980's a project was never able to transfer control to another project without human intervention. It required a user to end one project and then select another project from the MRU list and then start that project. Many projects became very large because of this constraint. To remedy this limitation, a new instruction called the "Switch Prg" was created.



The instruction requires you to provide the TIP name of the project that you would like to call. You may also pass up to 9 optional parameters to that project. The TIP name must be a valid TIP name found in the list of TIPS that the TIP Driver will allow you to start manually.

When the called TIP (project) starts, the current project that called it is ended and released from memory. The called project will start from the beginning and any parameters that were passed to it can be found in the parameters of the start up subroutine of the called project.