

Super Security — History

8.00 (2011/06/04)

Features

- Updated installer for the latest Clarion 8.0.

7.40 (2011/05/05)

Features

- Compatible with Clarion 6 through 8.0.

7.30 (2011/01/09)

Features

- Compatible with Clarion 6 through 7.3.

7.20 (2010/06/05)

Features

- Compatible with Clarion 6 through 7.2.

7.02 (2010/04/20)

Bugs

- The installer was aware of Clarion 7.0, but not 7.1.

7.00 (2010/01/29)

New Password!

We no longer officially support Clarion 5.5 and earlier versions!

Features

- Compatible with Clarion 6 through 7.1.
- Two new global embeds: "CheckAccess is beginning" and "CheckAccess is ending".
- PrepareFileNames embed was back-ported into the Clarion legacy chain.
- Installer permits overriding of destination folders.

6.62 (2007/10/29)

Features

- Compatible with Clarion 7-Beta.

- The icons have been reworked:
 - Green checkmark has become Green dot with forward arrow
 - Gray checkmark (access by group, bug individually denied) has become amber triangle (i.e. yield)
 - Red "X" has become red dot with white dash (i.e. "do not enter")
 - New "Clear" icon
- New class property (Security.NoAccDenMessages) prevents "Access Denied" error messages (except override offers).
- SBRW::BrowseEdit (used to view changed values when Auditing) will automatically apply the appropriate display pictures from the fields' settings in the dictionary.
- SecurityRun template also supports expressions beginning with "!". You can even use EVALUATE by beginning it with "=".

Bugs

- Devised improved solution for the intermittent "UpdateButtonInstances" error.
- Needed several calls to CLIP, for those who have changed their Security file STRINGS to CSTRINGS. Problem appeared as redundant entries in SSEC::Program and SSEC::Procedure files.
- Problem with CheckRuntimeAccess, where it could ignore the General Door for procedures with SaveButtons.

6.50 (2004/10/04)

- Concurrent with Super Stuff 6.50.

6.16 (2004/??/??)

Features

- Backdoor template prompt increased to 50 characters, to accommodate larger variable names (e.g. !Glo:SomeBackDoorVariableNameThatIHaveThatsReallyLong).
- The "Suppress Run-Time Security" extension template is now better described as "Tweak Run-Time Security". In addition to suppressing run-time security for a particular procedure, now it can also have a regular window treated as if it's an update form. Also, you still have the option to change the name that the Run-Time security uses to identify the procedure.
- Run-Time security has long attempted to "look ahead" to see if the browse's update procedure was going to prevent Inserts, Changes or Deletes. This worked well if the browse calls the update form directly (which is most common). If your browse calls an intermediate update procedure first, which in turn eventually calls the actual update window, now you can add an extension template to the BrowseUpdateButtons to specify the actual update window. This affects only the intuitive disabling of buttons on the browse, for operations that will be blocked by Run-Time security in the called update window.

6.15 (2004/07/28)

- New Password!

6.11 (2004/06/15)

Bugs

- (ABC) If you browse update buttons were restricting "Change" and not allowing "View", then you would get a compiler error on SELF.Translate.

6.10 (2004/05/15)

Features

- Compatible with Clarion 6.1.

Bugs

- (ABC) The installation program was missing DATA.APP from the SQL example directory (SECUR_DS).

6.06 (2004/04/27)

Bugs

- There was a problem with installation program not placing the new LOGON.TXA in the correct directory.

6.05 (2004/04/27)

Features

- (Clarion) Created SSEC::Logon procedure, just like in ABC. Now you must import CLARION\SUPER\SRC_CLA\SECURITY\LOGON.TXA into your base APP. You can make any desired changes to this logon window. Except for Run-time security, the Super Security legacy chain is now compatible with ClarioNet, et al. For those of you who had a custom STCLSEC.TRN merely for your Logon window, you may want to remove/tweak your copy now.
- (Clarion) Added the "Attempt auto-logon with network username" feature. (This feature pre-existed in the ABC chain.)
- (Clarion) Added the "Auto-Fill UserName from Previous Logon" feature. (This feature pre-existed in the ABC chain.)
- (ABC) Added a Microsoft SQL example in EX_ABC\SECUR_DS. See docs for more information.

Bugs

- (ABC) We left in a debug message in SSEC::Logon. This would appear if you use the network username for the logon. We've fixed it in LOGON.TXA and USERED_?.TXA. If you want to fix it in your own SSEC::Logon, then remove the marked line in the following section of code:

```

        COMPILER('***---***', SSEC::AutoLogonFromNetwork)
L::UserName = p_Security.GetNetUsername()
IF L::UserName
    !MESSAGE(L::UserName)  !***Remove this***
    IF ST::FindUser(L::UserName)
        DO ST::LogonSuccess
            SELF.Response = RequestCompleted
            RETURN LEVEL:Notify
        ELSE
            CLEAR(L::UserName)
        END!IF
    END!IF
END!IF
***---***

```

- *(ABC)* The importable USERED_?.TXA contained SSEC::Logon from an earlier version of Super Security. Importing LOGON.TXA after USERED_?.TXA would have solved the problem, but now the USERED_?.TXAs have been corrected.
- *(Clarion)* When logging on using the backdoor username, the displayed name would always be "[Backdoor]", even if you overrode it.

6.04 (2003/12/22)

Features

- We added a new global embed, so that you can assign runtime filenames for the security files without going to the trouble of overriding the Security class. Look for "[SuperSecurity] Prepare Security Filenames".

Bugs

- If you want to store the Security files in an SQL back-end, then you need to re-import LOGON.TXA to get the new SSEC::Logon procedure. Alternatively, you can manually edit SSEC::Logon and replace ST::FindUser. See "Upgrading From Earlier Versions" in the documentation for the replacement code.

6.03 (2003/12/05)

Bugs

- Fixed problem with generation of DLL exports, which could cause an "Ordinal out of sequence" error.

6.02 (2003/12/04)

Features

- Expanded documentation section "Upgrading from Earlier Versions" for those wishing to tweak their own SSEC::Logon for the new data access methods.

6.01 (2003/12/04)

Features

- Compatible with Clarion 6. This includes replacing UserNo_ and UserName_ with Set/Get_UserNo and Set/Get_UserName. See the documentation section "Upgrading From Earlier Versions" for more information.

- Added convenient template settings droplist of equates from your own MYDOORS.CLW, saving you the trouble of opening the file and copying the value.
- When you are inheriting and overriding the Security object, you can specify an INCLUDE file with your object's definition, and it will automatically be incorporated at the proper location in the code.

Bugs

- Improved handling of "ViewRecord" support.
- Improved security access cache handling when changing users.
- Added more Security class methods to the export list.
- Fixed problem with exporting overridden Security class in a multi-APP system.
- Fixed problem with support for MultiProject.

5.00 (2002/12/01)

Features

- Due to the new features, we've added more fields to the SSEC::User file, along with a new file called SSEC::PwdLog. You must get all of this into your dictionary, along with converting your existing data files. For more information, see the SuperSecurity documentation under the section "Upgrading From Earlier Versions".
- All hard-coded windows have been moved into generated Window procedures, to enable better handling by Clarionet, etc. Due to this change, you *****MUST***** import C:\C55\SUPER\LIBSRC\SECURITY\LOGON.TXA into your base APP. Also, if you are not using global maps, then mark the newly-imported procedure as "Declare Globally" in its procedure properties.
- There's a new property called Security.FullAccess. If you want your users to logon, but for some reason you want to allow unlimited access throughout the program, then set this to True. Its value is checked with each call to Security.CheckAccess(...), so you can change the value on-the-fly.
- There's a new method called Security.PrepareFileNames. It's called from Security.LoadQs, which is executed before Security.Logon in your Main procedure. It's an empty virtual procedure that you can override to set your own variable filenames and owner attributes for the security files. (Some of you may have been using PrepareLogonOpenFiles to handle some of your file opens. This virtual method is now extinct. Please move any code from that method to the new one.) You must override the Security object to do this. For more information, see the Classes tab on the global extension.
- The Logon can happen automatically using the network username (as long as it matches a username in SSEC::User).
- Passwords are no longer restricted to 8 characters. The new default (when importing SECURITY.TXD) is 20 characters, and you can edit the field definition to a maximum of 50 characters. There is also an optional setting for minimum password length.
- You can set passwords to expire, so that the user must enter a new one. This comes from a global default, but can be changed on a per user basis (including no timeout at all). When entering a new password, previous passwords are remembered in SSEC::PwdLog, so the user can be forced to enter a unique password compared to the last "x" passwords and/or last "y" days.
- The new Password features include some new strings in STABSEC.TRN. If you have a custom version of this file in your development directory, make sure that you add these new lines.
- There's a new "Locked" feature to lock-out users from logging on.

- You can set the "Maximum Logon Failures", at which time the user is automatically locked out.
- When the Security system HALTs the program, you can execute some of your own code on the way out. You must override the Security object to do this. For more information, see the Classes tab on the global extension.
- We've improved support for run-time language translation. See "Interface Modification and Translation" in the docs for more information.
- When you're specifying the access door in the template settings, now you have the option to select the door from your own MyDoors.clw.
- It also compiles clean under Clarion 6-EA2 and appears to function correctly. However, we haven't yet addressed all the issues with the new threading model.
- The documentation has been updated to reflect all features in this and prior releases. This is accessible via online help, or you can read STABSEC.PDF in C:\C55\SUPER\DOC\.

Bugs

- The Security.Purge sometimes didn't process all records.
- We cleaned up a few quirks in the "ViewRecord" handling.
- In ????\$SEC.CLW file, the INCLUDE('ABUTIL.TRN') needed the "ONCE" attribute.
- Auditing field changes for MEMO fields generated compile errors.

4.90 (2000/09/26)

Features

- Compatible with C55-cr2.
- This is the last version to support Clarion 4.0!
- Moved location of call to Security.Logon in ThisWindow.Init of Frame procedure from priority 1 to 100. This allows you to perform a task in your frame procedure before the logon occurs.
- Moved security check in browse prior to calling update form. Instead of being in ThisWindow.Run, it's in Browse.Ask. This adds SuperSecurity support for EIP updates, although auditing of actual data changes are not yet supported with EIP. (i.e. It will audit the call to insert, update or delete just fine, even though it can't track the actual change in data.)
- Added feature to run-time security so that the browse looks ahead to the update form to see if the update buttons should be disabled due to run-time security settings in the form. (That's in addition to the check it was already doing for itself.)
- Added support for CapeSoft's MultiProject.

Bugs

- Fixed problem with run-time security update procedures when using Door-oriented security. The current Procedure record would be lost if you added a new door from the door selection browse. Now the Procedure record is saved and restored around the call to SelectDoor. ***You should re-import the run-time support procedures into your application from SUPER\LIBSRC\SECURITY\RUNTIMED.TXA.
- Fixed problems with conflicting code for changing versus viewing a record. Now the optional manager override can occur with the Change and/or View.

- Fixed problem with run-time security button not always working in a multi-DLL system.
- Fixed problem with blank entries appearing in the audit trail files.
- Fixed problem with Insert not allowed, so now blank Auto-Inc record is cancelled.
- The utility LIBSRC\SECURITY\SEC_CONV.APP was using Access:File.Insert, which caused unwanted auto-numbering. It's been changed to ADD(Filename), with appropriate error checking. (This only affects you if you are converting from the really old security files to the current format. See the main docs for more information.)
- Fixed problem with support for CPCS reports and processes.

4.60 (2000/05/03)

Features

- Final Version to Support Clarion 4.
- Tested with Clarion 5.5 Beta-2.
- Run-time security DISABLEs buttons in a browse, depending on the settings for the browse's update form.
- "SuppressRuntimeSecurity" extension template can be used to disable support for a given procedure, or you can change the Procedure name used in the calls to CheckRuntimeAccess. This is useful if you have a procedure that is called for two purposes, and you wish to have different run-time security for each purpose.
- Added run-time security support for Processes, as well as UnivAbcReports and UnivAbcProcesses from CPCS.
- New optional fourth parameter for CheckRuntimeAccess method. Pass TRUE to prevent "Access Denied Message".
- Auditing parameters are CLIP'ed, to accommodate security files where the STRING fields have been changed to CSTRING.

Bugs

- Fixed problem with Security class definition when creating a multi-APP system.
- Inactivity timeout code in the Frame procedure has been moved out of the event handling loop, and into a separate ROUTINE. Therefore, you will no longer use "RETURN LEVEL:Notify" to abort the program. If you are tweaking the inactivity time-out code, you must set "SSEC::RetVal=LEVEL:Notify" in your code, or use HALT.
- Added call to CancelAutoInc, if run-time security prevented Insert with a form.

4.50

Features

- Tested with Clarion 5.5 Beta-1.
- In a multi-APP project, global security settings are output from base APP into STABSEC.INI in the application directory. All other APPs in the project can optionally read these settings. This saves you time the of maintaining consistent settings across all of your APPs.

- You can optionally remember the username from one session to the next. The Logon window automatically fills-in the prior username and position the cursor in the password field. If you enter the backdoor username the remembered username is cleared for the next logon. The username is stored in the WIN.INI file, in a [SuperSecurity] section. You can have a different remembered username for each program.
- You can press Ctrl-H in the in the Logon window to hide the username. This is handy if the customer is looking over your shoulder, and you don't want them to see the backdoor username.
- There is a new condition setting to cause the username on the logon window to be masked (like the password field). You can enter "True", or put some condition, like "INSTRING('BACKDOOR',UPPER(COMMAND()),1,1)".
- You users have the option to change password in Logon window. After they have entered their username and password, they can hit the "Password" button instead of "OK". It will then prompt them for their new password, plus verification.
- Calls to CheckAccess are cached, to speed repeated calls. The cache is forgotten whenever a new user logs on. (This includes a manager override.)
- Inactivity timeout is trapped differently. The HALT command is now generated into your Frame procedure. You can use the "Inactivity Timeout Handling" embed to short-circuit the message and HALT command, or just do some clean-up before it happens.
- New option to logon again after inactivity timeout.
- New methods "SuspendActivityCheck" and "ResumeActivityCheck" for those operations that will take a long time and will not update the activity tracking. This will prevent undesired inactivity timeouts.
- The "Inactivity Frame" setting is gone. Now all frames are treated as inactivity frames.
- If you need to override the Security class, you can derive your class from the Security class, then tell the global security settings to call your class instead of Security. (More on this in the next major update of the documentation.)
- When user hits the Run-Time Security button in the frame (where it doesn't apply), they see a message, rather than nothing happening. The equates for this message were added to STABSEC.TRN. If you have copied this file into your application directory, you must update it with these new equates.

Bugs

- Browse button disable modes are maintained properly now.
- Fixed various problems with Run-Time Security.

4.20

Features

- WindowControlSecurity code moved into routine so that it can be called again without closing and reopening the screen. (This is especially helpful for protected items on the main frame when you are allowing different users to re-logon without exiting the program.)

Bugs

- Fixed problem with ChangeAudit using GROUPs and DIM'ensioned arrays.

4.10

Features

- First SuperSolutions Templates
- Classes Tab in global extension allows developer to specify class other than "Security". Then they can derive their own class from the Security class with their own added functionality.

Bugs

- Inactivity timeout would sometimes immediately exit from the main frame.
- Door lookup in run-time security would cause problems with the primary record position.
- Initiating run-time security in 32-bit would cause a GPF.
- Fixed AccessState in UserEdit. FetchAccess needed leading tilde. You should make the change manually to your UserEdit, or recreate UserEdit by importing the new UserEd_D.TXA. (Does not apply to Levels.)
- Fixed incompatibility with CPCS and UltraTree templates.

4.00

Features

- Clarion 4 ABC Compatibility