

## What steps do I need to follow to successfully upgrade my SB+ 5.4 applications to run in SB+ 6.X for use with a character or SBClient GUI Emulation?

---

This document will take you through each of the steps you will need to perform and explain why each of these steps is being done.

---

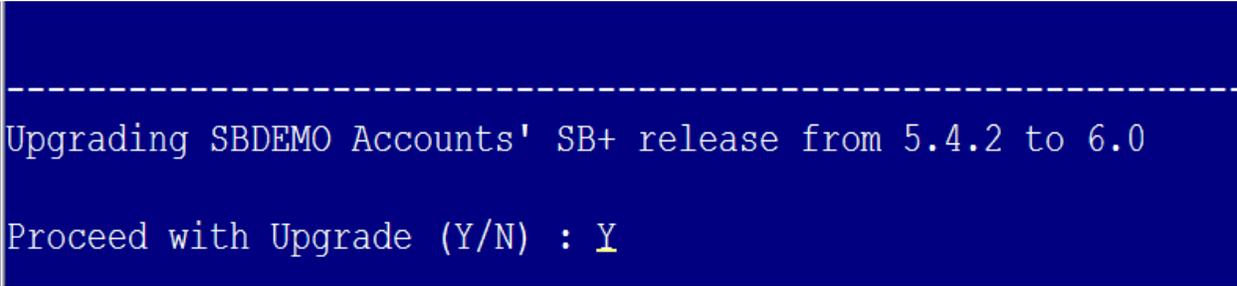
### Before We Begin!!!

There is no easy way to go back to a previous release of SB+ once you have upgraded to SBXA 6.X due to the changes made to the common block. As a precaution a backup of SB+ and applications software running on SB+ should be archived in the case that you have to revert back to the previous version.

### 1 - Upgrade process

When you first log to the application account you will either be presented with the following screen, because SB+ 6.X is installed in the same location as your previous version of SB+ was, or you will see errors about catalog pointers being incorrect.

If you are presented with the screen below you should accept the defaulted 'Y' prompt and allow SB+ to perform its upgrade process for this account.



```
-----  
Upgrading SBDEMO Accounts' SB+ release from 5.4.2 to 6.0  
Proceed with Upgrade (Y/N) : Y
```

**If you see the screen above you should skip steps 2 and 3.**

If your SB+ 6.X installation is in a different location than your previous installation you will receive errors when the LOGIN paragraph attempts to run.

- The SB+ file pointers in your application account will be incorrect.
- The SB+ cataloged program pointers in your application account are wrong.

## 2 - Fixing the SB+ File Pointers

The administration and program files used by SB+ are currently pointing to the wrong location.

This is a common task that needs to be performed and SB+ provides a routine that can be used to quickly resolve the problem. The program is called FIX.SBPLUS.POINTERS and is located in the SB+ administration account in the program file SAPROGS.

- Make sure that your pointer to SAPROGS exists and is correct.

### UniData

```
001   DIR
002   PathToSBAccount\SB\SAPROGS
003   PathToSBAccount\SB\D_SAPROGS
```

**UniVerse** - SAPROGS and SAPROGS.O pointers must exist and point to the same location.

```
001   F
002   PathToSB+Account\SB+\SAPROGS.O
003   PathToSB+Account\SB+\D_SAPROGS.O
```

### Before Proceeding:

- Review the getting started guide section called “Renaming SB Administration Account” in either Section 3 or 4 of the manual depending on whether the installation is for Windows or Unix.
- You can **NOT** use FIX.SBPLUS.POINTERS if the path to your SB Admin account contains duplicate elements. The following are examples of path name which will **NOT** work:
  - C:\Rocket\SB\SBXA\UniData\SB      The string SB is a duplicated path element  
Note – SBXA is not a duplicate string.
  - C:\Rocket\SBXA\UniVerse\SBXA      The string SBXA is a duplicated path element  
Note – SB+ Account renamed to SBXA.
- The U2 Knowledgebase contains several tech notes on the use of FIX.SBPLUS.POINTERS.
- **YOU MUST INCLUDE THE FULL PATH TO THE SB+ ADMINISTRATION ACCOUNT ON THE COMMAND LINE WHEN RUNNING FIX.SBPLUS.POINTER FROM YOUR APPLICATION ACCOUNT**

### Running the Command:

```
RUN SAPROGS FIX.SBPLUS.POINTERS {/full/path/to/SB Admin Account}
```

Verify the pointers to the SB files DM, DMGC, DMGD, DMSH, DMUT, DMSECURITY, DMCONT and DMACCOUNTS are pointing to the correct release after running FIX.SBPLUS.POINTERS.

### 3 - Cataloging the SB+ 6.X Program Files

As with changing the file pointers, the need to re-catalog the SB+ programs is also a common task and the routine CATALOG.SBPLUS has been provided to perform this task.

#### **GLOBAL versus DIRECT catalog on UniData Systems:**

When SB+ was first installed, the user was prompted whether to catalog SB+ globally or locally. CATALOG.SBPLUS remembers their choice and will re-catalog globally or locally accordingly. (DMSYSDEFN GC.CONTROL attribute <10> holds the catalog space prefix for SB+ programs and generated code).

If you want to run multiple versions of SB+ on one system they must either be cataloged using the DIRECT option or they must use a separate catalog space prefix.

Refer to Tech Note 1155272 for alternatives to running CATALOG.SBPLUS

#### **Running the Command:**

```
RUN SAPROGS CATALOG.SBPLUS
```

You will be prompted for the name of the SB+ Definition account.

The default names are:

- UniData            SB.DEFN
- UniVerse         SB+.DEFN

**If you have renamed the definition account please provide the correct account name.**

## 4 – Create Files and Pointers for New SB Administration and Application Files

**XXRESOURCES**            The XXRESOURCE file contains graphics files for themes, toolbars etc used by the new XUI clients.

- Create this file for each system id.
- File must be a DIR type file in UniData or Type 1 or 19 file in UnVerse.
- A global RESOURCES files can be created and referenced in the system control record (/HK.CONTROL - F5 – Additional “Global Resources File”

**XXTHEMES**            The XXTHEMES is where all Themes and the XAML code included in those themes is stored. The file should exist for each system within an application account.

- Create this file for each system id.
- File must be a DIR type file in UniData or Type 1 or 19 file in UnVerse.
- A global THEMES files can be created and referenced in the system control record (/HK.CONTROL - F5 – Additional “Global Themes File”

**SROC.LOG**            SROC.LOG is used to store server side errors relating to the new XUI environment. The log file is located in the SB Administration account.

- Create a file pointer to the directory file SROC.LOG in the SB Admin account.

## 5 - Compile Basic Programs

- The internal structure of SB+ Named Common has changed in version 6.X and this means that ALL subroutine that use the SB+ common block must be re-compiled and cataloged.
- This includes all generated code for SB+ processes. (See section 7 for generated code changes)
- The TU API Include files that were previously stored in the program file TUBP have been moved to the program file TUINSERT. Any subroutines that contain the string \$INCLUDE TUBP will need to be modified to read \$INCLUDE TUINSERT and then re-compiled and cataloged.

There are about 60 .H include records moved to TUINSERT from TUBP.

## 6 - Support for Generated Code in SB/XA processes?

**SBXA 6.X no longer supports generated code for input and output screens.**

This applies to all emulations – Character, SBClient GUI and the new XUI Browser and Rich clients.

**Removing generated code prior to upgrading to SB 6.X will allow you to use the /GC process. Validation routines in SB 6.X prevent you from modifying Input and Output processes using /GC.**

### Manual Steps to Remove Generated Source Code:

There are several files that may require to be updated when removing generated code for input and output processes from your applications (xx should be replaced with your system id):

- **xxPROCESS**            GEN.NAME        Record<4,2> contains the source code Id  
                              GEN.TYPE        Record<1> contains the process type
- **Global Process**        The same attributes and value positions used as XXPROCESS  
                              Global Process File is defined in – xxCONTROL, PARAMS<6,2>
- **VOC**                    The Catalog Pointer for generated source code id from xxPROCESS<4,2>
- **xxPROGS**                The source code for the Id from xxPROCESS<4,2>  
                              UniData - Object code preceded by an underscore
- **xxPROGS.O**            UniVerse - Object code for the Id from xxPROCESS<4,2>

The following select statement should return all the Input and Output processes that have generated code for the given system id:

```
SELECT CHPROCESS WITH TYPE = "I" "O" AND WITH GEN.NAME NE ""
```

You will need to remove the source code Id from each process returned.

Adding GEN.NAME to the end will return a list of Source Code Id's:

```
SELECT CHPROCESS WITH TYPE = "I" "O" AND WITH GEN.NAME NE "" GEN.NAME
```

You can use this list of Id's to remove catalog pointers from the VOC and Source and Object code from xxPROGS and xxPROGS.O

## 7 - You Must Run REGEN.EXP For All Your SB+ Applications.

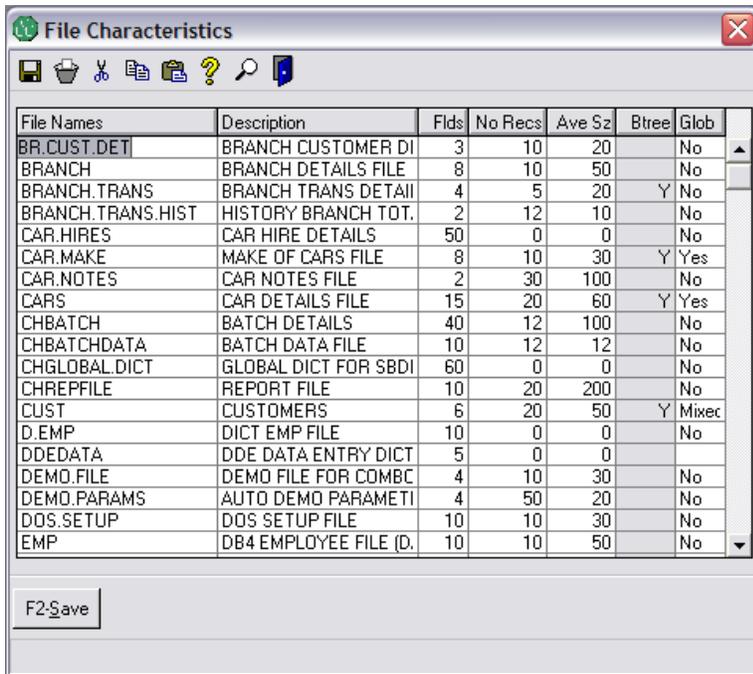
All SB+ definitions and processes contain SB+ internal executable strings that reference elements of the SBPLUS named common. Because the structure of this COMMON area has changed, you must regenerate all of these SB+ executable strings.

### Before you run REGEN.EXP

REGEN.EXP can be run for an individual file or for all files in a system. You will be taking the option to regenerate expressions for **ALL** files.

When you chose **ALL** files, SB+ will read the record **FILES** from the file **xxCONTROL** to obtain a list of files for the current system id. If this is not updated with all the files you use then the expression contained in those files that are not included will not be updated and may cause your application to fail.

You can update this list from the "File Create" process (/FC) by pressing F5 and adding and removing file names as necessary.



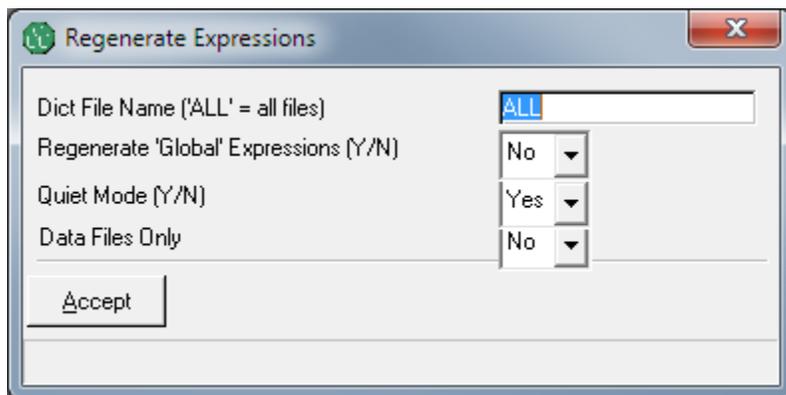
The screenshot shows a window titled "File Characteristics" with a toolbar and a table. The table lists various files with their descriptions and characteristics. At the bottom of the window, there is a button labeled "F2: Save".

File Names	Description	Flds	No Recs	Ave Sz	Btree	Glob
BR.CUST.DET	BRANCH CUSTOMER DI	3	10	20		No
BRANCH	BRANCH DETAILS FILE	8	10	50		No
BRANCH.TRANS	BRANCH TRANS DETAIL	4	5	20	Y	No
BRANCH.TRANS.HIST	HISTORY BRANCH TOT.	2	12	10		No
CAR.HIRES	CAR HIRE DETAILS	50	0	0		No
CAR.MAKE	MAKE OF CARS FILE	8	10	30	Y	Yes
CAR.NOTES	CAR NOTES FILE	2	30	100		No
CARS	CAR DETAILS FILE	15	20	60	Y	Yes
CHBATCH	BATCH DETAILS	40	12	100		No
CHBATCHDATA	BATCH DATA FILE	10	12	12		No
CHGLOBAL.DICT	GLOBAL DICT FOR SBDI	60	0	0		No
CHREPPFILE	REPORT FILE	10	20	200		No
CUST	CUSTOMERS	6	20	50	Y	Mixec
D.EMP	DICT EMP FILE	10	0	0		No
DDEDATA	DDE DATA ENTRY DICT	5	0	0		
DEMO.FILE	DEMO FILE FOR COMBC	4	10	30		No
DEMO.PARAMS	AUTO DEMO PARAMETI	4	50	20		No
DOS.SETUP	DOS SETUP FILE	10	10	30		No
EMP	DB4 EMPLOYEE FILE (D.	10	10	50		No

## Running REGEN.EXP

### Enhancements to REGEN.EXP in 6.x

- When REGEN.EXP is run in “Quiet Mode” all the steps performed are stored in the file **xxLOGS** for each system id ( **xx** is the system id).
- You can run REGEN.EXP for “Data Files Only” - This will not regenerate expressions for the **xxPROCESS** and **xxDEFN** files. This feature was added so that when deploying an application that has already been upgraded, you only need to focus on the data files where customers may have modified versions of screens etc..



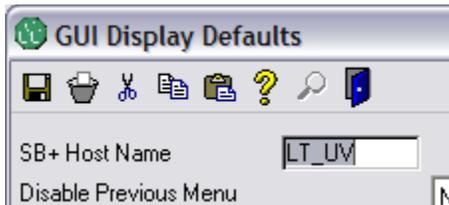
- You should enter ‘ALL’ for file name.
- ‘Y’es for Global Expressions.
- It is recommended to run in Quiet mode and then review the log for all errors.
- Only answer ‘Yes’ to this prompt if all the application definitions have already had the expressions regene.rated

**NB – You should investigate all errors and take corrective action then run REGEN.EXP again!!!**

## 8 - Other Upgrade Steps

**You MUST Add the Host Name to SB+ Control Parameters.**

/SB.PARMS → F8-Display Defaults → F5-GUI Display Defaults



The host name is used when downloading resources such as icons, themes, form definitions and other items that are cached locally on the PC when running in GUI and XUI modes. It is particularly important when working in a development environment and switch between systems that are running the same applications.

**Run /CLEAN.SBHSTATE on every client.**

This will remove the GUI form definitions for SBClient that are stored on your PC.

NB - You do not have to perform this step as long as you have entered an SB+ Host Name in SB+ control Parameters.

**Run CLEAR.FILE XXDRIVERS for every system used.**

The structure of the driver records used in SB/XA 6.X has changed and it is therefore necessary to remove the existing drivers so that new ones are created the first time an input, output... process is run.