SYSPRO Self-healing

SYSPRO 8

Reference Guide

Published: October 2020

CONTENTS

SYSPRO Self-healing

SYSPRO Self-healing

Exploring

Where it fits in?

In **SYSPRO 8**, self-healing refers to the automatic process in a client/server environment where various components of SYSPRO are updated on the client.

This is particularly useful in an environment where a large number of workstations are located in different geographical locations, and where a reinstall on each client would be overly cumbersome.

The self-healing process is activated when the **SYSPRO Installer** application is used for any of the following:

- A first time installation of **SYSPRO 8**
- An update to a new **SYSPRO 8** release
- One or more hotfixes are applied

When a client connects to the server where the self-healing process has been activated, each software component is copied from the server to the client as required. This ensures that the client is updated to match the server software.

This is generally a targeted process in that only the component(s) required at that point in time are copied to the client.

There are some exceptions, where a once-off transfer of a group of files occurs the first time the client has connected (after the server update was applied).

Solving

FAQs

What are managed assemblies?

Managed Assemblies are Microsoft .NET User controls typically developed using the **Microsoft .NET Framework**.

These include standard libraries shipped as part of the SYSPRO product and may also contain components designed by a third party developer to perform custom functionality.

How do I upload Microsoft .NET User controls?

Use the **Upload Files to Server** program (*SYSPRO Ribbon bar > Administration > Customization*) if you are a developer and have created new, or updated existing, Microsoft .NET User controls (i.e. managed assemblies) that require loading onto the server.

This will then transfer your .NET User Control to the server in such a way that the files will also be self-healed to each client.

What are Codejocks?

Codejocks are reusable software components that enhance the Graphical User Interface of Windows Desktop Applications.

Using

Process

Synchronizing Screensets

The following explains how the self-healing functionality synchronizes screensets:

> 💡 This process repeats itself each time a SYSPRO software update takes place on the server.

1. Each time you log in to SYSPRO, the system compares a `VERGUI.TXT` file on the server with a `VERGUI.TXT` file on the client (located in the `\Screens` folder of your SYSPRO installation).

2. If the program detects a difference between the `VER=` entries (usually because new software has been installed on the server) then all the screensets on the client are removed.

3. Once the screensets have been removed on the client, the `VERGUI.TXT` file on the server is transferred to the client.

4. When you select to run any SYSPRO program, the size of the screenset file on the client is compared to the size of the screenset on the server.

   If they are the same size, then the screenset on the client is run and the program continues as normal.

   However, if the size differs (or there is no screenset on the client) then the latest screenset from the server is transferred to the client screenset folder.

   This transfer is handled by SYSPRO automatically. After the screenset has been transferred to the client, it remains on the client for future use.

   All this happens seamlessly and, unless you are running SYSPRO over a relatively slow WAN, you will not notice that it has occurred.

   > 📝 All files that are self-healed are compressed during the transfer to reduce network traffic and provide the best performance.

5. When each screenset is self-healed, its associated controls are also self-healed:
   - List views
   - Tree views
   - Toolbars
   - Status bars
   - Forms, etc.

These are transferred from the `\Base\Bin` folder on the server to the `\Base\Bin` folder on the client.

In addition, the associated files in the `\Base\UI` folder are self-healed at the same time as the files in the `\Base\Bin` folder.

The program controls are prefixed by the screenset name.

> **FOR EXAMPLE:**
>
> The screenset for the **Inventory Query** program (`INVPEN`) is called `INVPEN.RS`. The associated program controls in the `\Base\Bin` folder are named `INVPEN??.INT`.

SYSPRO Client executable

Executable files (such as `SYSPROClient.EXE`) cannot be updated while active in memory.

Therefore, the following process explains how the self-healing process handles the SYSPRO client executable:

1. The release of a new SYSPRO client executable is accompanied by the following set of files to establish whether an update is required:

   - `SYSPROClient_NEW.exe`

     This is the new executable that will be named `SYSPROClient.exe` on the client.

   - `SYSPROClient_SET.exe`

     This is the setup program that will rename the file on the client to `SYSPROClient.exe`.

   - `SYSPROClient_VER.txt`

     This text file contains a `VER=` entry that is a 10 character field which typically contains the last change date in CCYY-MM-DD format.

   These new files are downloaded into the server `\Base` folder, but remain inactive alongside the active `SYSPROClient.EXE`:

2. The next time the user logs into SYSPRO, the self-healing process will verify the `VER=` entries of the `SYSPROClient_VER.TXT` file in both the server and client.

   If these entries differ, the system will transfer the three new files from the server `\Base` folder to the client `\Base` folder.

3. The system detects that this has occurred and will then automatically run `SYSPROClient_SET.EXE` on the client.

   When `SYSPROClient_SET.EXE` is running, the regular `SYSPROClient.EXE` has already exited - and is therefore not locking the standard client executable.

4. The `SYSPROClient_NEW.EXE` file is then copied over the top of `SYSPROClient.EXE`.

5. The user will then be notified that the new client executable has been installed and that they simply need to re-run the standard client executable – this will run the new `SYSPROClient.EXE`.

> 💡 You can force a reinstall of the client executable at any time by editing the `SYSPROClient_VER.txt` file and changing the `VER=` entry.

Codejock and ChartFX controls

> 💡 Codejocks are reusable software components that enhance the Graphical User Interface of Windows Desktop Applications.

The Codejock and ChartFX controls reside in the `\Base\Controls` folder of your SYSPRO installation. These controls contain embedded forms, list views, charts and other components used by the screensets.

The following process explains how the self-healing functionality handles Codejocks and ChartFX Controls:

1. When logging into SYSPRO on a client machine, the menu system checks whether these controls exist.

2. If they don't exist (or are not the correct version) then the controls are copied from the `\Base\Controls` folder on server to the client.

   > 📝 As the SYSPRO executables contain appropriate manifests, there is no requirement to register these components.

SYSPROInteropHostControls

> 💡 These controls support the use of .NET User Controls as well as SYSPRO Reporting Services Document Printing.

There are two `SYSPROInteropHostControl` files that reside in the SYSPRO `\Base` folder:

- `SYSPROInteropHostControl.dll`
- `SYSPROInteropHostControl40.dll.`

If the `SYSPROInteropHostControl.dll` is not found, then SYSPRO will not start. It will create an entry in the **Windows Event log** indicating a side-by-side error message.

You must then register `SYSPROInteropHostControl40.dll` on the client machine using `REGASM.EXE`.

> **FOR EXAMPLE:**
> `regasm /codebase SYSPROInteropHostControl40.dll`

> 📝 It is extremely rare that these controls are changed.

Managed Assemblies

> Managed Assemblies are Microsoft .NET User controls typically developed using the Microsoft .NET Framework.
>
> These include standard libraries shipped as part of the SYSPRO product and may also contain components designed by a third party developer to perform custom functionality.

Managed Assemblies reside in the `\Base\ManagedAssemblies` folder of your SYSPRO installation.

Self-healing is based on changes made to the folder on the server.

If the last modified date has changed (typically because of a third party publishing a new Microsoft .NET user control) then the entire contents of the `\Base\ManagedAssemblies` folder on the server are copied to the `\Base\ManagedAssemblies` folder on the client.

Additional Client Components

There are some additional core client components of *SYSPRO 8* that may require self-healing.

Specific checks for files in the `\Base` folder are used to determine when these components require self-healing.

You will be notified when self-healing has occurred for these components.

When you exit SYSPRO and reload the client, the newly-updated components will be in use.

Workflow Services

The self-healing process for *SYSPRO Workflow Services* involves comparing the version entry contained in the `IMPSWS.IMP` file (located in the `\Base\Store` folder on the SYSPRO server) and `SWSVER.TXT` file (located in the `\Base` folder on the SYSPRO client).

If the version entries differ, self-healing occurs as follows:

1. On the server, all the workflow components are copied from the `\Base` folder to the `\Base\Workflow` folder.

2. Thereafter, all the workflow components in the `\Base\SYSPROWorkflow` folder on the server are copied to the `\Base\SYSPROWorkflow` folder on the client.

Reporting Services and Analytics

SYSPRO Reporting Services and ***SYSPRO Analytics*** are closely bound with regards to self-healing. When SRS initiates a self-healing session, SYSPRO Analytics is self-healed at the same time, and vice verse.

When hotfixes are installed (using the **SYSPRO Installer Application**) any SRS-specific files are installed in the SYSPRO `\Base` folder, while any specific SYSPRO Analytics files are installed in the `\Base\SYSPROAnalytics` folder.

There is a group of files common to both SRS and SYSPRO Analytics (referred to as the common files) which are installed in the SYSPRO `\Base` folder. SRS can directly access the common files, but SYSPRO Analytics requires the common files to be copied to the `\Base\SYSPROAnalytics` folder before they can be accessed.

Update process

Whenever an update occurs:

- Files specific to SYSPRO Analytics are placed directly into the `\Base\SYSPROAnalytics` folder on the server.
- Files specific to SRS or common files are placed into the `\Base` folder on the server.

The update process runs in two phases:

1. The first phase occurs on the server.
2. The second phase occurs between the server and the client.

Server-side healing

The SRS and SYSPRO Analytics self-healing process interrogates the following two files:

- The `IMPHEA.IMP` file located in the `\Base\Store` folder on the server.
- The `HEALVER.TXT` file located in the `\Base\SYSPROAnalytics` folder on the server.

Each of these files has an entry that contains software version information. Self-healing is initiated if the version information contained in these two files differs.

If the version number in `IMPHEA.IMP` is greater than that contained in `HEALVER.TXT` (or if the `HEALVER.TXT` file can't be found) then the following occurs:

- The common files are copied from the `\Base` folder on the server to the `\Base\SYSPROAnalytics` folder on the server.

> This process applies to both SRS and SYSPRO Analytics; they compare the same files and implement the same self-healing procedure.

Client-side healing

Once the self-healing process is completed on the server, the update process compares the software version information entry contained in the following two files:

- The `IMPHEA.IMP` file on the server.
- The `HEALVER.TXT` file located in the `\Base` folder on the client.

Self-healing is initiated if the version information in these two files differs.

If the version number in `IMPHEA.IMP` on the server is greater than what is contained in the `HEALVER.TXT` file on the client (or if the `HEALVER.TXT` file cannot be found on the client) then the following occurs:

- SRS and common files are copied from the `\Base` folder on the server to the `\Base` folder on the client.
- The SYSPRO Analytics files are copied from the `\Base\SYSPROAnalytics` folder on the server to the `\Base\SYSPROAnalytics` folder on the client, including all of the common files that were copied during the first phase of the self-healing process.

SYSPRO Analytics server

The SYSPRO Analytics Server installation can't be updated automatically from the application server.

Updates that are saved on the client must be manually copied to self-heal the program to the SYSPRO Analytics Server. This is achieved by using the **Update** function of the **SYSPRO Analytics Servers** application within the **SYSPRO Analytics Setup** program.

Language Translation

In a client/server environment, the system attempts to self-heal dictionaries from the server to the client.

This is done as follows:

1. The client software determines the file sizes of the following dictionary files:
   - `Base\Lang_??.TXT`
   - `Base\settings\Lang_??_Custom_Dictionary.TXT`
   - `Base\settings\Lang_??_Global_Replace_Dictionary.TXT`

   (where `??` indicates the language code)

2. This information is passed to the server software, which determines the file sizes of the same files on the server (if they exist).

3. Any of these files which exist on the server, but not on the client, are then transferred to the client.

   If the files currently exist on both the server and the client, but the file sizes differ, then the files are transferred from the server to the client. In this way, changes to the dictionaries are automatically self-healed.

> The self-healing process may take a few seconds while the files are being transferred.

SYSPRO

Simplifying your Success

www.syspro.com