Release Process

Aptean Ltd
Copyright © 2011-2025.

Contents

1 Release Process

1.1 Prerequisites

Ensure all code is back in SourceSafe and any new files have been added to SourceSafe in the appropriate sub-solders.

Ensure that all database modifications have been documented in

- `P:\271321 - Vision\_Installers\Releases In Progress\LatestDBChanges.SQL`

A template of this script is available called:

- `P:\271321 - Vision\_Installers\Releases In Progress\LatestDBChanges_Template.SQL`

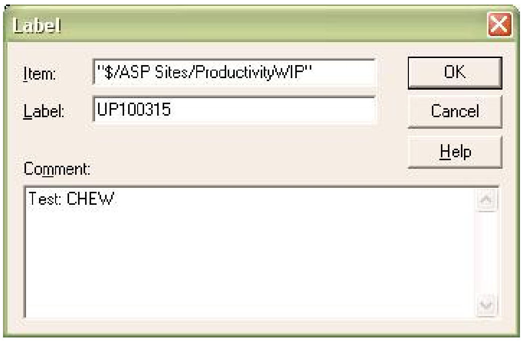
1.2 Create Patch

Copy folder P:\271321 - Vision_Installers\Releases In Progress\Vision_UPYYMMDD to the new release patch name, replacing YYMMDD with the current date. This is referred to as the Patch folder and is denoted as {patch} in the following instructions.

Copy P:\271321 - Vision_Installers\Releases In Progress\LatestDBChanges.SQL to {patch}\Releases\UPYYMMDD.SQL, replacing YYMMDD with the current date. Add this file to SourceSafe, in the Database sub-folder.

Label the programs being released in SourceSafe as follows:

In the left-hand pane of SourceSafe, right-click the 'ProductivityWIP' folder and choose 'Label...' from the pop-up menu.



Item: "$/ASP Sites/ProductivityWIP"

Label: UP100315

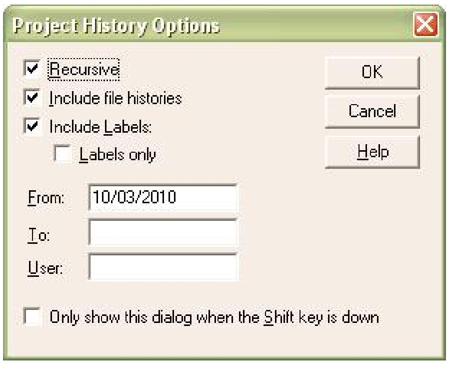
Comment: Test: CHEW

Figure 1 - SourceSafe Label Creation screen

Enter the patch name in the "Label" field, and comment the patch with the client(s) to whom the patch is being released, from the Supimix Client/Site fields.

Find the programs changed since the last release (labelled in SourceSafe) using the SourceSafe history function, as follows:

In the left-hand pane of SourceSafe, right-click the 'ProductivityWIP' folder and choose 'Show History...' from the pop-up menu.



☑ Recursive
☑ Include file histories
☑ Include Labels:
□ Labels only

From: 10/03/2010

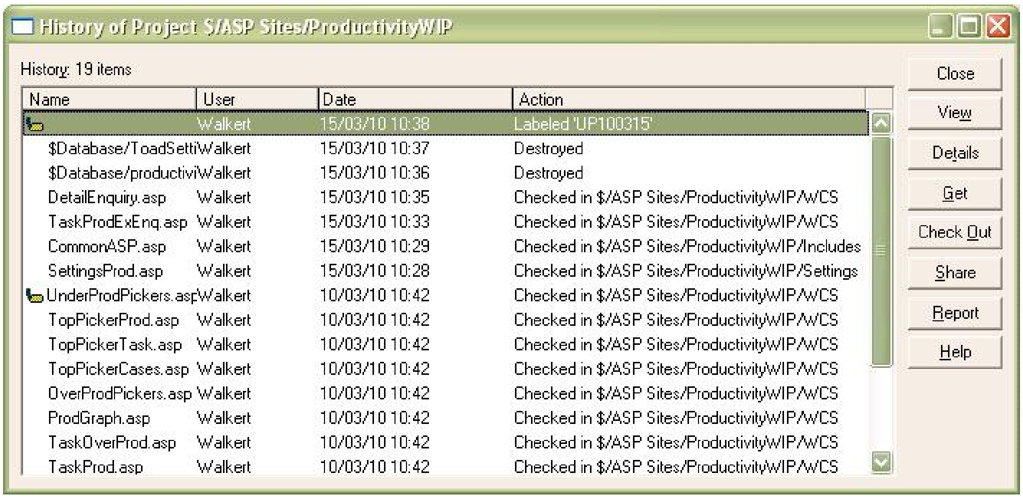
To:

User:

□ Only show this dialog when the Shift key is down

Figure 2 - SourceSafe Project History Options screen

Ensure 'Recursive' is checked. Enter a 'From' date if you know when the last release was completed, to speed things up. SourceSafe will then show a list of all changes made, matching the parameters entered.



History of Project $/ASP Sites/ProductivityWIP

History: 19 items

Name	User	Date	Action
	Walkert	15/03/10 10:38	Labeled 'UP100315'
$Database/ToadSetti	Walkert	15/03/10 10:37	Destroyed
$Database/productivi	Walkert	15/03/10 10:36	Destroyed
DetailEnquiry.asp	Walkert	15/03/10 10:35	Checked in $/ASP Sites/ProductivityWIP/WCS
TaskProdExEnq.asp	Walkert	15/03/10 10:33	Checked in $/ASP Sites/ProductivityWIP/WCS
CommonASP.asp	Walkert	15/03/10 10:29	Checked in $/ASP Sites/ProductivityWIP/Includes
SettingsProd.asp	Walkert	15/03/10 10:28	Checked in $/ASP Sites/ProductivityWIP/Settings
UnderProdPickers.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
TopPickerProd.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
TopPickerTask.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
TopPickerCases.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
OverProdPickers.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
ProdGraph.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
TaskOverProd.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS
TaskProd.asp	Walkert	10/03/10 10:42	Checked in $/ASP Sites/ProductivityWIP/WCS

Figure 3 - SourceSafe History screen

For each program changed or added, copy them into sub-folders of the Patch folder. Each program should go into the subfolder required for the program, as specified in SourceSafe.

Examples: DetailEnquiry.asp exists in a project subfolder called WCS - create a sub-folder called {patch}\WCS and copy the file into this folder.

CommonASP.asp exists in a project subfolder called Includes - create a sub-folder called {patch}\Includes and copy the file into this folder.

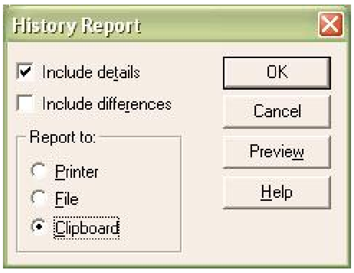
GetSysDetails_MySQL.vbs should be copied to a sub-folder called {patch}\Scripts.

Once all changed or added programs are copied into their respective Patch sub-folders, the release notes should be built.

Rename the file {patch}\Releases\UPYYMMDD_RelNotes.txt, replacing YYMMDD with the current date.

Replace the change content with change descriptions from SourceSafe. To do this:

From the SourceSafe History screen (see Figure 3), click the 'Report' button.



History Report

☑ Include details

□ Include differences

Report to:

○ Printer

○ File

● Clipboard

Figure 4 - SourceSafe History Report Options screen

Ensure 'Include details' is checked, and 'Clipboard' is selected and then click 'OK'.

SourceSafe will drop this history to the clipboard, ready for pasting.

Paste this into the release notes. The format will look as follows:

```
        $/ASP Sites/ProductivityWIP
```

```
*****************  Database  *****************
User: Walkert         Date: 15/03/10  Time:  10:37
Destroyed ToadSettings

*****************  Database  *****************
User: Walkert         Date: 15/03/10  Time:  10:36
Destroyed productivity1

*****************  DetailEnquiry.asp  *****************
User: Walkert         Date: 15/03/10  Time:  10:35
Checked in $/ASP Sites/ProductivityWIP/WCS
Comment:
  272693/SW-7YYJEU for DHL/EMEA/CHEW - Added choice of employee; Added
extended Export functionality (through to HTML-style XLS sheet); Added Task Type
to display; Added selection of All task types; Added ability to choose a
specific date/week/month/quarter or range, including validation and popup
calendar; Added ability to summarise the data reported (as well as a detail
option to report as now).

*****************  TaskProdExEnq.asp  *****************
User: Walkert         Date: 15/03/10  Time:  10:33
Checked in $/ASP Sites/ProductivityWIP/WCS
Comment:
  272693/SW-7YYJEU for DHL/EMEA/CHEW - Added choice of employee; Added
extended Export functionality (through to HTML-style XLS sheet); Added Task Type
to display.

*****************  CommonASP.asp  *****************
User: Walkert         Date: 15/03/10  Time:  10:29
Checked in $/ASP Sites/ProductivityWIP/Includes
Comment:
  272693/SW-7YYJEU for DHL/EMEA/CHEW - Added translation of 'ALL' when
converting task types.

*****************  SettingsProd.asp  *****************
User: Walkert         Date: 15/03/10  Time:  10:28
Checked in $/ASP Sites/ProductivityWIP/Settings
Comment:
  272694/SW-7YYJNV for DHL/EMEA/CHEW - Fixed bug created during
testing regarding setting the user default values for the Productivity Settings
selection.
```

1. Remove the initial project line
2. Remove any deletion notifications.
3. Remove the lines beginning "User:", "Checked in" and "Comment".
4. Remove the asterisks.
5. Remove any blank lines
6. Format the remaining text by joining the lines, removing white space and tabbing and returning at each semi-colon, as follows:

```
DetailEnquiry.asp       272693/SW-7YYJEU for DHL/EMEA/CHEW - Added choice of employee;
                        Added extended Export functionality (through to HTML-style XLS sheet);
                        Added Task Type to display;
                        Added selection of All task types;
                        Added ability to choose a specific date/week/month/quarter or range,
                        including validation and popup calendar;
                        Added ability to summarise the data reported (as well as a detail option to report
TaskProdExEnq.asp       272693/SW-7YYJEU for DHL/EMEA/CHEW - Added choice of employee;
                        Added extended Export functionality (through to HTML-style XLS sheet);
                        Added Task Type to display.
CommonASP.asp           272693/SW-7YYJEU for DHL/EMEA/CHEW - Added translation of 'ALL' when converting tas
SettingsProd.asp        272694/SW-7YYJNV for DHL/EMEA/CHEW - Fixed bug created during testing regarding
                        setting the user default values for the Productivity Settings selection.
```

Save the release notes.

From within the Patch folder, zip all the contents into a release patch, named "Vision_UP{YYMMDD}.zip", where YYMMDD is the current date. Copy this patch to the "P:\271321 - Vision_Installers\Releases In Progress" folder. The working directory {patch} can now be deleted.

1.3 Releasing

Send the patch out to the client - usually this is through FTP. A shortcut will be provided in the "P:\271321 - Vision_Installers\Releases In Progress" folder. This shortcut will connect and set the release folder and file type correctly. Usually this is through the commands:

```
cd Vision
bin
```

Release the patch with the command:

```
send Vision_UP{YYMMDD}.zip
```

where YYMMDD is replaced with the patch number.

Exit FTP with the command 'exit'.

Inform the clients of the release with and email, as follows:

```
"Subject: Vision Release
Attachments: UP{YYMMDD}_RelNotes.txt

All,

A new release patch, Vision_UP{YYMMDD}.zip, is available for your Vision installation.
Release notes are attached.

{Notes}"
```

Replace {YYMMDD} with the patch date and attach the release notes.

Any specific notes, installation or testing instructions can be added at the {notes} point, or in the release notes themselves.

Update the Supimix entry for the changes associated to the release with:

- Status: 5
- Sub-status: 70
- Patch: UP{YYMMDD}

Enter the Analysis and fault codes if these have not already been done.

Enter the full Vision patch name in the comments when confirming - ensure that these are "USE" type comments.

If this is a log rather than a development, send the fix notification email when Supimix has finished updating the log status.

1.4 Installing

Ensure that the users are aware of the installation.

Depending on the size and content of the installation, this may take from 15 minutes to an hour.

- Pause any data mining scripts
- Stop the web server for the Vision application.
- Alternatively, set the system unavailable flag in Global.asa in the Vision folder.
- To install the patch, log on to the Vision server and move to the Vision folder, hereafter referred to as {Vision}.
- Depending on the complexity of the patch, a backup copy of all the programs and the database should be made, as follows:
  - Backup the {Vision} folder to a safe area in a file called VisionBackup_YYYYMMDD.zip.
  - Backup the current database - Many tools can be used for this, but the best tool would be mysqldump, as follows:

```
mysqldump --host={host} --port=5306 --user=root --password={password}
  --result-file=VisionDBBackup_YYYYMMDD.sql Productivity1
```

- Move to the {Releases} subfolder. Get the patch and place it in this subfolder - this is usually through FTP, as before.
- Extract the patch to a subfolder Releases\{Patch}. 💡 **Note:** Alternatively, if the version of windows allows, explore into the zip file instead of extracting it.
- Copy the subfolders of {Vision}\Releases\{Patch} to {Vision} - this will copy all of the required programs, scripts and notes into the correct folders. Note: Each program and component may be released separately by manually copying each item, if necessary.
- Move to the {Vision}\Releases folder - the release notes and database update script (if included) will be there.
- To commit the changes to the Vision database:
  - Start the MySQL Query Browser and log on to the Productivity1 database.
  - Click *File/Open Script...* and open the script {Vision}\Releases\UPYYMMDD.SQL script file, where YYMMDD is the patch number.
  - Click the **Continue** button on the toolbar to make the changes. Note: If there are any problems with the script, the script will pause at the point it failed and display the error. If the error is fixed at this stage and **Continue** is pressed again, the script will continue from the point it left off.
- Delete the {Vision}\Releases\{Patch} folder - it is no longer required. If space is at a premium, also delete the patch zip file.

Once the installation steps above have been followed, the patch has been installed.

Depending on the contents of the patch, it may be necessary to test the installation incrementally - do this by enabling each component it turn and testing, in this order:

- Database and data checks
- Data Mining script checks
- Vision Application checks

The steps to re-enable the system are as follows:

- Start the web server for the Vision application and/or make the system available again (if disabled at the start of the process).
- Re-enable Data Mining.

1.5 Updating Documentation

This *CALIDUS* Assist guide will require modification after release:

- New pages creating for all-new functionality
- Menu pages amended for new pages
- Existing pages modified with new functionality
- New images created and uploaded if appropriate
- If new rules are added, rules documentation changed - see following
- If new screens are added, screens documentation changed - see following

All changes to documentation should be created with a comment as follows:

```
Updated based on version x.y.z
```

The Title and Appendix page should then be updated.

After this, the guide can be exported as PDF. Navigate to the category Category:Vision User Guide Doc then extract to PDF using the link on the left. Save this to

```
P:\271321 - Vision\Documents
```

in format:

```
UG 236343 Vision User Guide vx.y.z.pdf
```

1.5.1 Self-documenting Rules

Rules should always be created on an appropriate menu with descriptive text.

When that is complete, the following SQL can be used to extract the data:

```
SELECT
IF(rm.Menu_Flag = 'Y', IF(rm.rule_parent <> 'SY00001', '**', '*'), IF(rm.rule_parent <> 'SY00001', '**', '
CONCAT('\'\'', rm.Rule_Label, '\'\'') AS Label, ' - ' as delim, rm.Rule_Description
FROM rules_master rm
Where rm.rule_type = 'SYS' and rm.Rule_Parent <> ''
order by IF(rm.menu_flag = 'Y', rm.rule_name, rm.rule_parent), (rm.menu_flag <> 'Y'), rm.rule_name;
```

Note that this does not correctly deal with sub-menus, so this will need some slight modification after production.

Once complete, this can be pasted into the Rules screen

1.5.2 Self-Documenting Screens

Screens should always be created on an appropriate menu with descriptive text.

When that is complete, the following SQL can be used to extract the data:

```
SELECT
Screen_Label, `Screen Description`, Screen_Type, Group_Code
FROM Screens
WHERE Screen_Type = 'Summary'
ORDER BY Screen_Type, Group_Code, Screen_Label;
```

Screen_Type can be:

- Summary
- Single
- Graphs
- Enquiry
- Settings
- Monitor
- Admin

Paste the results into the appropriate screen, formatted as a table, e.g.:

```
{| class="wikitable" border="1"
|- bgcolor="silver"
!Name   !!Description
|-
|colspan="2"| '''Group1'''
|-
| Name1 || Description1
|-
| Name2 || Description2
|-
|colspan="2"| '''Group2'''
|-
| Name1 || Description1
|-
| etc || etc
|}
```

Resulting in:

Name	Description
Group1	
Name1	Description1
Name2	Description2
Group2	
Name1	Description1
etc	etc