



Software Tool House Inc

Meta-Archive for HTML
Software Tool House Inc.

User's Guide

Release 6.01

© 2021 Software Tool House

Meta-Update Release 6.01

Apr 15, 2021



Preface

Audience

This document is intended for Remedy ARS Administrators and developers. It is expected that the reader will have knowledge of the Remedy ARS system, ITSM and be familiar with their tables. It would behoove the reader to be familiar with his ARS server's platform and scripting tools.

Limitation of Liability

This program is provided "as-is". We are in no way liable for any losses arising from your use of this program, the sample scripts, or the documentation. It is your responsibility to evaluate this program. It is your responsibility to backup and protect your data. It is your responsibility to evaluate your use of this program for any particular purpose.

This manual does not represent a commitment to maintain any syntax or operation, nor is it warranted to be complete or accurate.

Copyrights

This program and this manual are copyrighted 1996-2021 by Software Tool House Inc.

Meta-Layer and Meta-Update are trademarks of Software Tool House Inc.

ARS, Remedy are registered trademarks of BMC Corporation.

Solaris is a registered trademark of Sun Microsystems Inc.

Windows is a registered trademark of Microsoft Corporation.

PCRE (Perl Compatible Regular Expression) library is copyrighted © 1997-2021 by University of Cambridge and is distributed under the BSD license.

Updates

This program and this manual may change from time to time. The latest version is available at our web site: www.softwaretoolhouse.com.

Comments

Your comments are welcome! Please see: www.softwaretoolhouse.com/support and click **Comments**, or email us at support@softwaretoolhouse.com. We look forward to hearing from you!



Table of Contents

Preface	2
Audience.....	2
Limitation of Liability.....	2
Copyrights.....	2
Updates.....	2
Comments.....	2
Table of Contents	3
Introduction	6
ITSM Archiving Pain Points.....	6
Trees of data.....	6
Association Checking.....	6
Other Requirements.....	7
Meta-Archive – Implements Archiving your way!.....	7
Installation	10
Pre-requisite: Meta-Update Installation.....	10
Meta-Archive & Meta-Update Distributions.....	10
Expanding the Meta-Archive Distribution.....	10
Expanding the Meta-Archive Distribution.....	11
Meta-Archive Distribution Contents.....	11
Understanding Archiving	14
Overview.....	14
Archiving Targets.....	15
Configuring Archiving	17
Overview.....	17
100-Arch-Cfg.ini.....	18
Query Date Format.....	18
CSV File Delimiter.....	19
Archive Targets.....	19
115-Arch-SrcServer.ini.....	20
Arch-Cfg-ITSM-810-HTML.xlsm.....	22
About ITSM Archive Configuration.....	22
Sheets.....	22
Config sheet.....	24
root sheet.....	25
tree sheet.....	27
assoc sheet.....	29
CMDB sheet.....	31
HTML sheet.....	32
Archive-Cfg-Csv-Tbls.xlsm.....	39



Software Tool House Inc

Introduction



Introduction

Thank you for selecting Meta-Archive. Meta-Archive implements **your** archiving solution.

With Meta-Archive, you have complete, configurable, archiving, to alternate ITSM servers, archive forms, and/or, CSV files – with attachments extracted.

This User's Guide helps you configure Meta-Archive, to add your forms, to change your associations checking, and add ages for types of tickets.

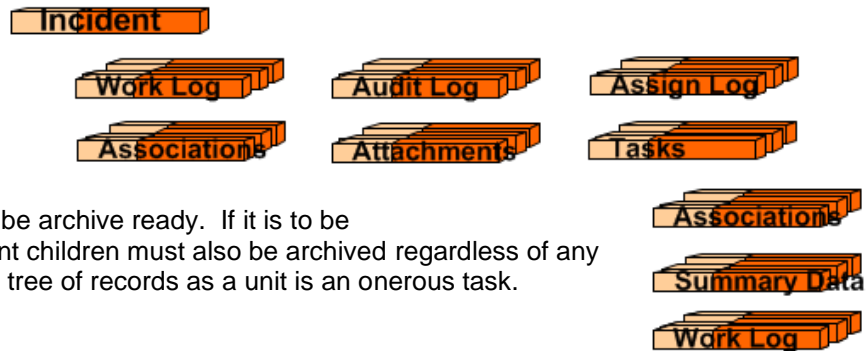
It also shows you how to set up Meta-Update Job Automation when the initial archive load is great.

ITSM Archiving Pain Points

Trees of data

ITSM is a complex application running on a database of some 3,500 tables. These tables are connected by redundant references making an Incident a hierarchical tree of records in thirty odd forms,

That tree of records belonging to a single Incident needs to be handled as a unit.



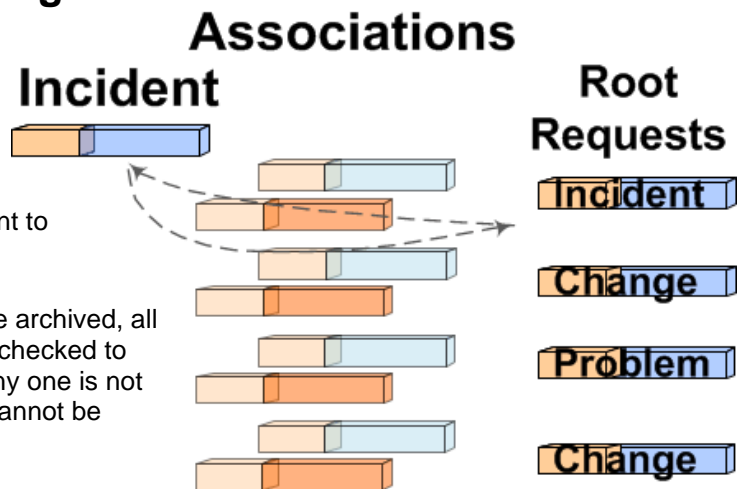
Only the Incident can be archive ready. If it is to be archived, all dependent children must also be archived regardless of any dates. Managing that tree of records as a unit is an onerous task.

Association Checking

If an archive-ready Incident is related to a Change that is not archive-ready, then that Incident should also not be archived.

Those working the change will want to see the Incident.

So, before any root request can be archived, all associated root requests must be checked to see if they are archive ready. If any one is not archive ready, then that Incident cannot be archived.



Other Requirements

- Building the Archive Forms is a manual effort using BMC Developer Studio.
- An archiving solution should be configurable so you can add your own customized forms easily.
- A robust archiving solution should allow for other types of archives. For example, having an ITSM archive server to hold data allows GUI and reporting access to the archived data with no impact on production performance.
- It should be easy to set different requirements for different customers or classes of tickets.

Meta-Archive – Implements Archiving your way!

Meta-Archive handles all these requirements through the BMC Remedy API with no server forms or workflow needed at all.

Installation

Installation

Pre-requisite: Meta-Update Installation

Meta-Archive requires Meta-Update. Meta-Update needs be available and licensing tested. Full Meta-Update installation instructions and usage are in the Meta-Update User's Guide.

Meta-Update installation consists of simply expanding the Meta-Update distribution zip or gzip file. Supplied license files are saved in the Meta-Update bin directory and the SthLicUpd.exe command is run to generate an SthLic.cmd batch file setting Remedy users, addresses, ports, user logins and optionally encrypted passwords.

The generated command file (Windows batch) or shell script may be edited as needed.

Paths to the Meta-Update bin or bin64 directory must be established – See Running below.

Meta-Archive & Meta-Update Distributions

Meta-Archive and Meta-Update do not have to be installed into the same directory. Meta-Archive is run through Meta-Update. That is, the Meta-Update binary is fired and the script that is run is one of the Meta-Archive scripts.

Like all sample scripts, running a Meta-Update script for the section “Do” will print usage and argument help.

Meta-Update finds scripts along a script path and as long as the Meta-Archive scripts can be found, the current working directory can and should be independent of both Meta-Archive and Meta-Update installation directories.

Expanding the Meta-Archive Distribution

Meta-Update Releases are complete distributions including all required binaries, sample scripts, and documentation.

A Meta-Update release is distributed as a single zip or g-zip file. The file may be expanded in your applications area. For example,

On Windows, you could unzip the distribution in:

```
"C:\Program Files\SoftwareToolHouse\"
```

Similarly, on UNIX, you will gunzip the file as you wish and then untar the resultant tar ball.

```
mkdir /apps/STH
chdir /apps/STH
gunzip SthMupd-5.56.tar.gz
tar -xvf SthMupd-5.56.tar
```



Expanding the Meta-Archive Distribution

Meta-Archive Releases are complete distributions including all required binaries, sample scripts, and documentation.

A Meta-Archive release is distributed as a single zip or g-zip file. The file may be expanded in your applications area or any other area. This area does not need to be associated with the Meta-Update area.

For example,

On Windows, you could unzip the distribution in:

```
"C:\Program Files\SoftwareToolHouse\Meta-Archive\"
```

Similarly, on UNIX, you will gunzip the file as you wish and then untar the resultant tar ball.

```
mkdir /apps/STH/Meta-Archive
chdir /apps/STH/Meta-Archive
gunzip SthMeta-Archive-5.56.tar.gz
tar -xvf SthMeta-Archive-5.56.tar
```

Meta-Archive Distribution Contents

Meta-Archive Releases are complete distributions including all required binaries, sample scripts, and documentation.

File Name	Need Client Edit	Description
Arch-Cfg-ITSM-810.xlsx	Yes	This spreadsheet is the main Archiving configuration file.
cfg-arch-root.csv cfg-arch-tree.csv cfg-arch-assoc.csv cfg-arch-excl.csv		You can use it to change the Archive table names, the ages for Archiving and Associations, to add your own tables, to disable modules of ITSM that you are not using. You may create copies for your different customers should they have differing retention requirements. It will automatically generate the CSV files when you are ready.
Archive-Cfg-Csv-Tbls.xlsm cfg-tbl-csv-map.csv	Yes	This is an optional Archiving CSV Naming mapping file. This is only used when archiving to CSV files and only if specified.

100-Arch-Cfg.ini	Yes	<p>This script fragment contains user configurations for Meta-Archive.</p> <p>For example, archive targets of CSVs or identical forms on alternate servers may be selected. You also set Date formats and a global delete flag.</p> <p>As delivered, this is set to Archive to archive forms only, date format of "mdy" (US), no Deletes performed.</p>
115-Arch-SrcServer.ini	Yes	<p>This script fragment is the "Source" (or production) server information.</p> <p>As delivered, this is the same as the "Target" (or archive) server. Identical forms on a single server is not possible.</p>
350-Arch-FormsAdmin.ini	No	<p>This script builds all configured Archive forms.</p> <p>It is run once only per environment or when new forms are added to the configuration.</p> <p>It can be used to check the status of all Archive forms. It will skip any forms already built if run more than once.</p>
500-Arch.ini	No	<p>This script performs archives and restores.</p> <p>During the initial run, when there is a large backlog of records that must be archived, it is expected that Meta-Update Job Automation will be used to fire this script job.</p> <p>For testing of script or configuration changes, this script will be fired manually.</p>
590-Arch-Del.ini	No	<p>Some archive target options require that the deletes be performed separately. If so, this script performs archive deletes as listed the the delete log generated from a Meta-Archive run.</p> <p>This can be added to Meta-Update Job Automation.</p>
Meta-Update_Job_Automation_Users_Guide.pdf		Meta-Update Job Automation User Manual.
000-Jctl.ini	No	This script does Meta-Update Job Automation and controls series of Meta-Update jobs based on a Meta-Update Job Automation spreadsheet.



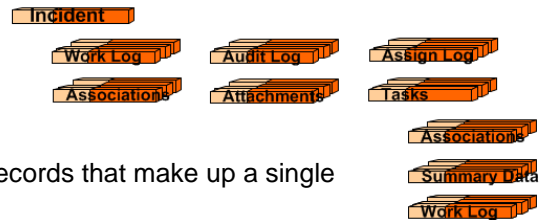
Arch-Jobs-sample.xlsx	Yes	<p>This is a sample Meta-Update Job Automation spreadsheet for Meta-Archive.</p> <p>It is expected that for the initial backlog run that this would be configured for the customer's specific data requirements.</p> <p>It is possible that a single job fired weekly or monthly can perform the Archive once the backlog is done. Then, no different spreadsheet will be needed for the monthly run. If the monthly volume is large enough, a different split-up of the jobs would be needed.</p>
000-Jctl-Sync.ini 910-Arch-CfgRd.ini 920-Arch-Assoc-Do.ini 930-Arch-Upd-Csv.ini	No	<p>These scripts are included by the scripts above.</p>

Understanding Archiving

Overview

In Meta-Archive a “**root request**” would be an archivable data object such as an Incident, a Change, or, a Problem.

The root request is the base of a tree of data in different forms.



In the image to the right, we see some of the data records that make up a single Incident.

In Meta-Archive, these “root requests” and data “trees” are configured in an Excel spreadsheet. Adding your own table is as simple as inserting a row.

Meta-Archive runs date based queries against the root requests. Additional query clauses may be configured.

In the configuration spreadsheet, “tree” rows show the fields needed to generate queries for direct children to two levels. In the above picture for example, a Task Work Log (level 2) can be queried from a Task but not from an Incident. The Task (level 1) itself is queried from data in the Incident.

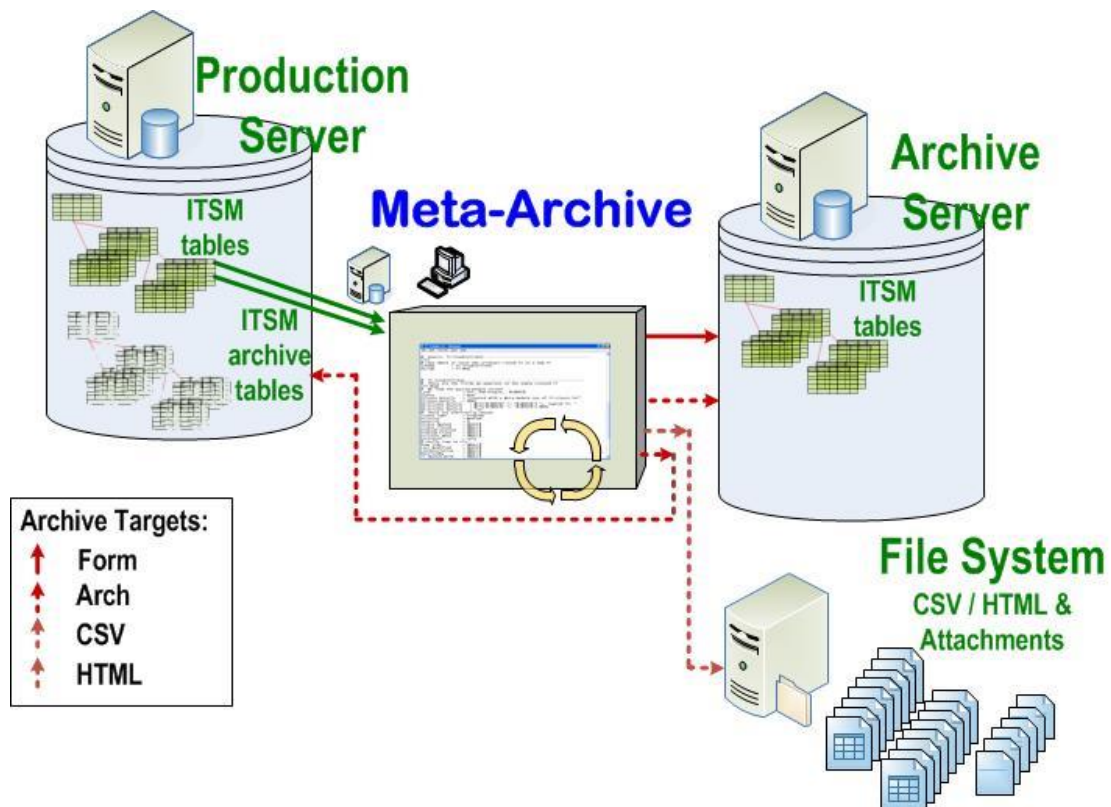
Associations are configured so that each root request is validated first against all its associations and will be archived only if those associations permit it.

Archiving Targets

Meta-Archive offers you choices of where and how the archived data is to be stored.

Target	Description
Archive Forms	An archive form is a specialized BMC Remedy generated copy of a regular form. Meta-Archive can be used to automatically generate archive forms.
Identical Forms	If you use a separate Archive server, the same forms can be used on both servers. This allows the archived data to be viewed with the normal Remedy user interface.
CSV Files	All data from each of the archived tables are set to their own CSV files. Attachments are saved to the file system.
HTML Files	All data from each of the archived tables are set to their own HTML files. Attachments are saved to the file system.

For the CSV archive target, Meta-Archive offers different naming algorithms. This target can always be selected along with the other targets.



If only the CSV or HTML target is selected, deletions are not automatically done when the data is archived. Once the archive job completes, its deletion log is run through a delete script.

Identical forms can only be done on two different source and archive servers.

If this target is selected and the source and archive servers are the same, an error will be thrown and the job will end.

Archive forms can be built only from normal forms on a single server. Archive forms are generally used on a single server.

The database instance containing the archive data and the production data are the same.

So, by archiving data – that is, moving data from the ITSM Incident, Change, Root Request forms to archive tables – indexes are smaller and searches are speeded up, but the database table size is not reduced.

In Meta-Archive, Archive forms can be used when the source and target server are not the same. This is not recommended but not checked.



Configuring Archiving

Overview

There are several files that are expected to be configured by the user once listed in [Distribution Contents](#) above.

File Name	Need Client Edit	Description
100-Arch-Cfg.ini	Yes	<p>This script fragment contains user configurations for Meta-Archive.</p> <p>For example, archive targets of CSVs or identical forms on alternate servers may be selected.</p> <p>You also set Date formats and a global delete flag.</p>
Arch-Cfg-ITSM-810.xlsm cfg-arch-root.csv cfg-arch-tree.csv cfg-arch-assoc.csv cfg-arch-excl.csv	Yes	<p>This spreadsheet is the main Archiving configuration file.</p> <p>You can use it to change the Archive table names, the ages for Archiving and Associations, to add your own tables, to disable modules of ITSM that you are not using.</p> <p>It will automatically generate the CSV files when you are ready.</p>
Archive-Cfg-Csv-Tbls.xlsm cfg-tbl-csv-map.csv	Yes	<p>This is an optional Archiving CSV Naming mapping file. This is only used when archiving to CSV files and only if specified.</p>
115-Arch-SrcServer.ini	Yes	<p>This script fragment is the "Source" (or production) server information.</p> <p>As delivered, this is the same as the "Target" (or archive) server. Identical forms on a single server is not possible.</p>
Arch-Jobs-sample.xlsx	Yes	<p>This is a sample Meta-Update Job Automation spreadsheet for Meta-Archive.</p> <p>See Job Automation below.</p>

100-Arch-Cfg.ini

This file holds the User Configuration settings for Meta-Archive. It is an “Assignment Sections” in Meta-Update script fragment included by other scripts. With it, you can change the behaviour of Meta-Archive.

Simply edit the file in an ASCII editor such as Notepad.exe.

```
100-Arch-Cfg.ini
# Meta-Update and this script are (c) copyright 1996-2015 by Software Tool House Inc.
# Meta-Archive and Meta-Update are trademarks of Software Tool House Inc.
#
# File:      100-Arch-Cfg.ini
#
#           This Meta-Update script fragment contains the User modifiable configurations
#           for Meta-Archive.
#
#           Comments start with a hash at the start of a line and are ignored.
#
# Configuration assignments      These are User configurations and choices.
#                               Please review.  For example, as delivered,
#                               Archiving does NOT delete records.
#
#-----
# Cfg,   QryDateFormat  char   the Remedy Query Date format.  One of:
#                               mdy | dmy | ymd
#
@Cmd    = Ref,   Cfg,   QryDateFormat,   mdy
#-----
# Cfg,   CsvDelim       char: CSV delimiter:  "," or ";"
#
@Cmd    = Ref,   Cfg,   CsvDelim,         ","
#-----
# Cfg,   Del            bool: 0 No deletes; 1 deletes after a tree is archived
#
@Cmd    = Ref,   Cfg,   Del,              0
#-----
# Cfg,   ArchOutCsv     bool: 1 in addition to archive forms, output to CSV is also done
#                               0 no output to CSV is done
#                               The AttachFname setting will be used to store attachments to the file system
# Cfg,   ArchOutArch    bool: 1 output to archive forms on the target server
#                               The Read and Target servers may be the same or different (why though?).
# Cfg,   ArchOutForm    bool: 1 output to identical forms (with possible new names) on the target server
#                               This means that the ReadServer and the Target server MUST (if not new names) be different
#                               servers.  This is checked.
# Cfg,   ArchOutFormNmEq bool: 1 Forms have the same name and MUST be on different servers
#                               0 use Archive Name configs for OutForm - may be the same server
@Cmd    = Ref,   Cfg,   ArchOutCsv,       1
@Cmd    = Ref,   Cfg,   ArchOutArch,      1
@Cmd    = Ref,   Cfg,   ArchOutForm,      0
#
```

Query Date Format

Meta-Archive generates BMC Remedy queries using dates. As delivered, these dates are generated in “mdy” format – that is: month/day/year.

The EU and Australia and New Zealand will need to change this to “**dmy**” and Asian countries to “**ymd**”.



This setting can easily be tested with Meta-Query with these commands in order, “mdy”, “dmy”, “ymd”:

```
SthMqry -c -S "HPD:Help Desk" "' 6' > \"01/31/2015\""  
SthMqry -c -S "HPD:Help Desk" "' 6' > \"31/01/2015\""  
SthMqry -c -S "HPD:Help Desk" "' 6' > \"2015/01/31\""
```

Remedy will throw an error on two of these and return a count of matches correctly in one. Use that one's corresponding setting.

CSV File Delimiter

Meta-Archive's output files use this character as the CSV separator. The default is a comma however some EU countries would need to change this to a semi-colon.

Archive Targets

There are three archive targets:

Target	Description
Archive Forms	An archive form is a specialized BMC Remedy generated copy of a regular form. Meta-Archive can be used to automatically generate archive forms.
Identical Forms	If you use a separate Archive server, the same forms can be used on both servers. This allows the archived data to be viewed with the normal Remedy user interface.
CSV Files	CSV files will be written containing every field from the forms in the tree sheet. Attachments will be extracted to the file system. Because all data is present, restores are possible.
HTML Files	HTML files will be written according to the HTML configurations. Restores are not possible.

To set your archive targets, set the variables to either 1 or 0.

The CSV and HTML targets can be selected with other targets. They will cause CSV / HTML files to be created that contain the archive data. Attachments will be saved to the file system with their original extension but a name that can be found easily for restores.

When you select CSV / HTML, there is an optional configuration file you can use to control the naming of the CSV / HTML files and attachments.

If identical Form is selected, the archive servers must be different. It makes no sense to archive from one form into itself on the same server, Meta-Archive will throw an error if Form is selected and the servers are the same.

If Arch is selected the servers are generally the same. This makes Archive and Identical forms mutually exclusive selections.

115-Arch-SrcServer.ini

This file holds the server access and login details for the source server.

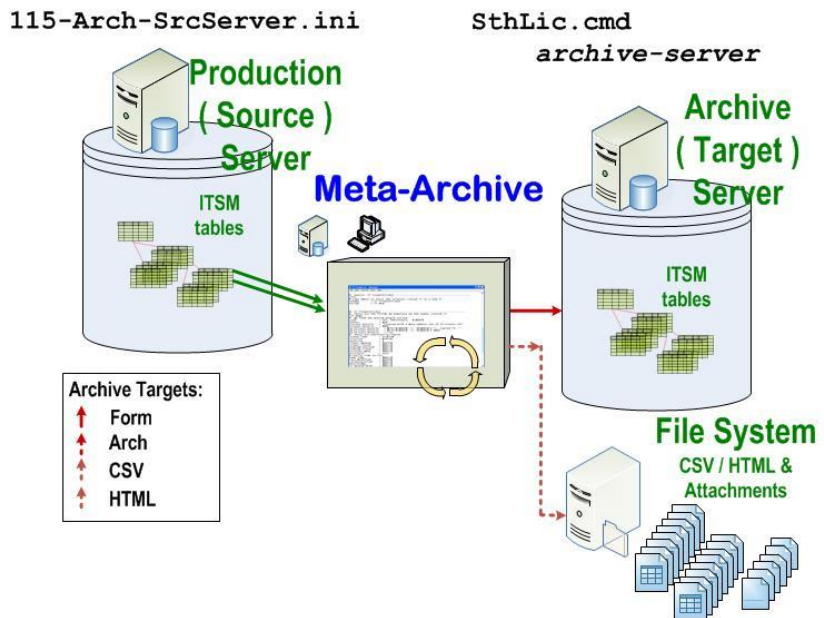
As delivered, this is the same server that is the [Main] or target server and uses the environment variables set by `SthLic.cmd` or `. SthLic.sh`

Simply edit the file in an ASCII editor such as Notepad.exe.

```
115-Arch-SrcServer.ini
# Meta-Update and this script are (c) copyright 1996-2015 by Software Tool House Inc.
# Meta-Archive and Meta-Update are trademarks of Software Tool House Inc.
#
# File:      115-Arch-SrcServer.ini
#
#           This Meta-Update script fragment contains the Archive Source Server addressing and
#           sign-on args for Meta-Archive.
#
#           As delivered, this is the same server as the licensed target server.
#
#           Comments start with a hash at the start of a line and are ignored.
#
Server      = % ENV,  ArsSvrAdmin %
Port        = % ENV,  ArsPort   %
User        = % ENV,  ArsUsr    %
Password    = % ENV,  ArsPwd     %
```

If you are archiving to archive forms on the same server, there is no need to change this file.

If you are archiving on identical forms on an archive server, this references the source server – the one on the left in the image below.

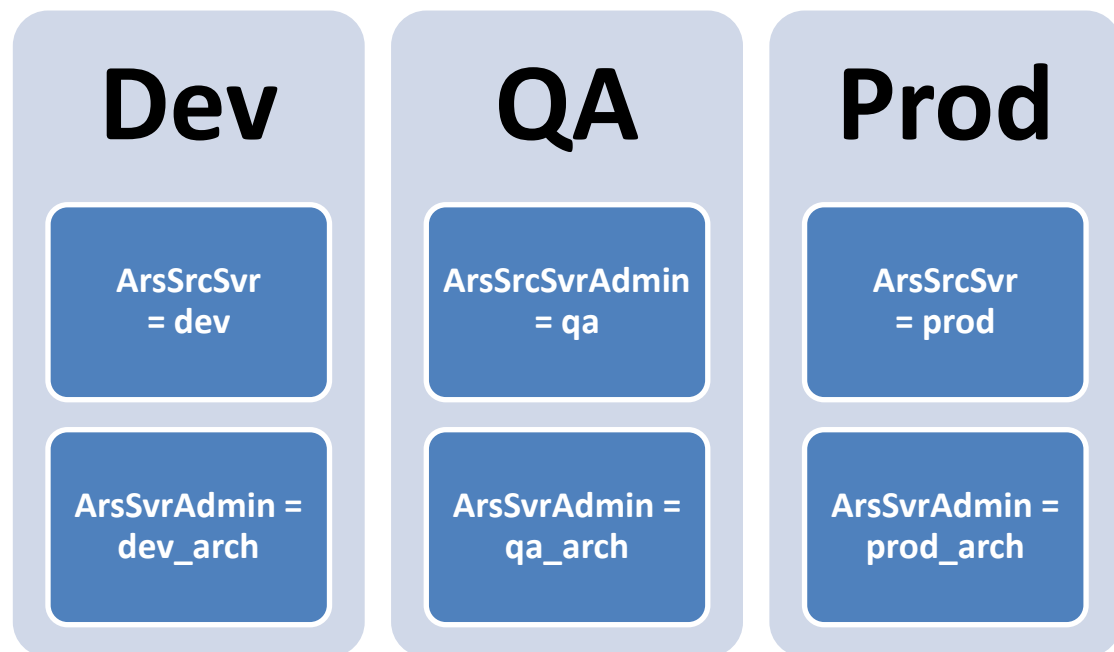




You can modify the above file to change the environment variables referenced. For example, you could use environment variables **ArsSrcSvrAdmin**, **ArsSrcPort**, **ArsSrcUsr**, and, **ArsSrcPwd**. If the password is different than the target archive server, you can use **SthLicUpd.exe** to generate an encrypted password.

You may also use **SthLic.cmd** or **SthLic.sh** so that environments are assigned Remedy servers in “pairs”.

The following table shows three environments and environment variables set to point to the different environments’ servers and archive servers. A total of six servers are set: **dev**, **dev_arch**, **qa**, **qa_arch**, **prod**, **prod_arch**.



About ITSM Archive Configuration

This spreadsheet contains all the major configurations for Meta-Archive.



Macros must be enabled to automatically generate CSV files.

Meta-Archive Introduction	
This spreadsheet is a configuration for the Meta-Archive Scripts for the OOTB ITSM 8.1 for the Incident, Change, Problem, Known Error, Release, Work Order root requests. This sheet may be adapted to include your own forms or for other root requests.	
Contents	
Instructions and Settings	
This explains the other sheets in detail. Please read this page to adapt this for other modules or your own customized tables.	
Config	
Use this sheet to set the archive table names (in formulas) and to create appropriate CSV extract files (with a macro).	
root	
This sheet defines root forms. All Root Form requests to be archived should be listed in this sheet. Records are queried from these forms and all dependent records (given by the "tree" sheet) are archived for requests returned by this query.	
tree	
This sheet describes the tree of forms and linking fields from the root forms. It can specify forms on two levels. The first level is directly related to a root form and the second level is related to a form on the first level. An example is a Task. A Task has its own set of forms with records related to only the Task form, hence these forms are on level 2.	
assoc	
This sheet describes the associations and linking fields from the root forms. A root request can only be archived if and only if all its associated root requests can be archived. This sheets configures association rules and tables.	
exclusions	
This sheet lists individual Customer companies that will be excluded by the script with this configuration file.	
Meta-Update	
Meta-Update is used to automate any ARS / ITSM data ETL - Extract, Transform, Load operation: <ul style="list-style-type: none"> — without staging forms, workflow, or any server changes at all — without any programming skills at all — in a language your ARS admin already speaks 	
Copyrights	
Meta-Archive and Meta-Update are trademarks of Software Tool House Inc. All Meta-Archive and Meta-Update documentation and software including this sheet are © Copyright 1996 through 2014 by Software Tool House, Inc.	
Contents / Instructions / Config / root / tree / assoc / exclusions / Change Log	

Archive configuration for ITSM 8.0.0

Want more information about Meta-Update? Please click the following links:

[Management Summary Presentation](#)

[Learning Meta-Update](#)

[Trial License Request](#)

[Live Install](#)

Sheets

The spreadsheet contains 9 work sheets:

Sheet	Description
Contents	An introductory Help page.
Instructions	Sheet based instructions. Really a summary of this section of the User's Guide.

Config	<p>Contains global data used by formulas in the other sheets. These include a default Archive Table name, and the ITSM Release number.</p> <p>It also presents a button to export all sheets.</p>
root	<p>Describes root forms. Includes archive form names. Ages, query qualifications.</p>
tree	<p>For each root, describes the tree of dependent forms. Specifies linking fields and queries as well as specific HTML data.</p>
assoc	<p>For each root, describes a list of Associations to validate. Includes association specific ages and statuses.</p> <p>These are checked for any root request and if any associations require the root, the request will not be archived but logged instead.</p>
CMDB	<p>Specifies classes and archive names that can be archived.</p> <p>Any classes not listed here are not archived.</p>
HTML	<p>Specifies the HTML formation rules.</p>
exclusions	<p>Specifies companies that should be excluded from any archive runs that use this configuration sheet.</p> <p>A separate sheet will be used for these companies.</p>

It defines all root requests and the attributes related to archiving: ages, statuses, and other query terms.

It defines a two level tree of forms for each root and specifies the linking fields. It specifies associations to be validated before archiving can take place.

For the CMDB, it specifies which classes are to be archived.

Companies to be excluded from this archive configuration are specified as well.

The configurations also specify behavioral settings for CSV and HTML archiving.


Finally, the spreadsheet includes macros to automatically generate the CSV files in the Config sheet.

Config sheet

The Config sheet sets some global values used in formulas and lets you generate all CSV files at the touch of a button.



Macros must be enabled for this button to work.

Archive Table Names	
Please specify a suffix for Source table names used when naming the Archive table.	
Archive Table Name Suffix:	-ARC
ITSM Release	
Some table entries have a formula in the enabled column depending on this release number. Note that this a text value and should conform to BMC release numbering.	
ITSM Release	8.1
Export to CSVs	
Please specify an output file "prefix" such as C:\cfg\cfg-arch	
With this example, files will be: cfg-arch-root.csv, cfg-arch-tree.csv, cfg-arch-cmdb.csv, cfg-arch-html.csv, cfg-arch-assoc.csv, and, cfg-arch-excl.csv - located in folder: C:\cfg\	
Directory C:\cfg\ must exist before the button is pressed.	
Output files' prefix:	e:\Dta\Meta-Archive\cfg-arch
	Press the button to the left to export the files.
With the current setting, these files will be created:	e:\Dta\Meta-Archive\cfg-arch-root.csv e:\Dta\Meta-Archive\cfg-arch-tree.csv e:\Dta\Meta-Archive\cfg-arch-asc.csv e:\Dta\Meta-Archive\cfg-arch-cmdb.csv e:\Dta\Meta-Archive\cfg-arch-html.csv e:\Dta\Meta-Archive\cfg-arch-excl.csv
<p>Contents Instructions Config root tree assoc CMDB HTML exclusions Change Log</p>	

“Archive Table Name Suffix”

is used in formulas in the root and tree “Archive Form” columns. The value set here is appended onto a root or tree form name. Individual Archive Form values may be overridden if desired.

The “ITSM Release”

is used in the root form’s “Enabled” column for the CMDB. BMC introduced a separation of status information from the CMDB in release 8.1.

Status information is key to the Archivable records query, so, two different root requests for the CMDB are specified in the **root** sheet.

Only one of these can be enabled. If you do not wish to have the CMDB archived, disable both root requests.

The “ITSM Release”

is also used in the “Root Form” column of the tree sheet to select the enabled CMDB Root form. It is important to note that all fields in the CMDB tree form are actually the fields of BMC.CORE:BMC_BaseElement and are not dependent on the ITSM release.

Output files’ prefix

The full file path and file name prefix for the CSV files is set here. Once set, you can press the button to generate the CSVs.

The generated file names will take the value entered here and suffix “-root.csv”, “-tree.csv” and so on.



Pressing this button will generate all CSV files as described above.

root sheet

The root sheet is used to define the various ITSM “Root Requests” that will be archived. It also sets an archive age and an archive query qualification.

ITSM Root requests are Incident, Changes, Problems, Work Orders, Releases, Service Requests, and Configuration Items.



Form	Enable	Delete Flag	Archive Qualification	Date Field	Archive Form	Age
HPD:Help Desk	1	1	((('Status' = "Closed" and 'Closed Date' < "@date@") or ('Status' = "Cancelled" and 'Cancelled Date' < "@date@")) and not ('VIP' = "Yes"))	Closed Date	HPD:Help Desk-ARC	1y
CHG:Infrastructure Change	1	1	(('Change Request Status' = "Closed" and 'Closed Date' < "@date@") or ('Change Request Status' = "Cancelled" and 'Last Modified Date' < "@date@") or ('Change Request Status' = "Rejected" and 'Last Modified Date' < "@date@"))	Completed Date	CHG:Infrastructure Change-ARC	1y
PBM:Problem Investigation	1	1	((('Investigation Status' = "Closed" and 'Closed_Date' < "@date@") or ('Investigation Status' = "Cancelled" and ('Last Modified Date' < "@date@") or ('Investigation Status' = "Rejected" and 'Last Modified Date' < "@date@") or ('Investigation Status' = "Completed" and 'Last Modified Date' < "@date@"))	Closed_Date	PBM:Problem Investigation-ARC	1y
PBM:Known Error	1	1	('Known Error Status' = "Closed" or 'Known Error Status' = "Cancelled" or 'Known Error	Archive Date	PBM:Known Error-ARC	1y

The **Enable** column must contain a “1” or this root request, its tree, and its associations are ignored.

If the “**Delete Flag**” is not “1” no data for this root request is deleted. This includes data from the tree of records associated with a root request.

Note that there is also a global; Meta-Archive Delete Flag. Only if both the global Delete Flag and a root request's Delete Flag are set will any deletes take place for this root request and its tree of data.

“Ages” may be specified in any combination (but in order) as follows:

xy xm xd xh xm xs for “x” years, months, days, hours, minutes, seconds.

The first “m” is taken to be months and the second to be minutes. To specify one month, use “1m”. To specify one minute, use, “0m1m”

The Query qualification is used to query the root Form to get a list of “eligible archive records” or “potential archive records”.

Archiving only proceeds if the root request's associations permit it.

You can use the token “@date@” in your query qualification. This will be replaced at run time by an actual date: the time of the run minus the age specified.

If different customers or different classes of tickets have different archive requirements, add appropriate terms to the archive qualification.

You can make copies of this spreadsheet and generate the CSV files for different runs of Meta-Archive. You can also use Meta-Update Job Automation to automate the different runs according to your performance and window parameters.

The “Date Field” is only used if the “@date@” substitution token wasn't found in the qualification. If “@date@” wasn't in the qualification, the date field specified is added to the qualification resulting in:

(Archive Qualification) and ('Date Field' < "@date")

The Configuration Items, or the CMDB, is specified in the Root Sheet as two different forms with a formula in the Enabled column that depends on the ITSM Release.

If you do not want to archive CMDB records, set both CMDB root requests to not Enabled.

Form	Enable	Delete Flag	Archive Qualification	Date Field	Archive Form
AST:BaseElement	1	1	(('AssetLifecycleStatus' = "Delete" or 'AssetLifecycleStatus' = "End of Life" or 'AssetLifecycleStatus' = "Disposed" or 'AssetLifecycleStatus' = "Return to Vendor" or 1=1) and ('ModifiedDate' < "@date@" or 'ReturnDate' < "@date@" or 'Disposal Date' < "@date@"))	ModifiedDate	AST:BaseElement-ARC
BMC.CORE:BMC_BaseElement	0	1	(('AssetLifecycleStatus' = "Delete" or 'AssetLifecycleStatus' = "End of Life" or 'AssetLifecycleStatus' = "Disposed" or 'AssetLifecycleStatus' = "Return to Vendor") and ('ModifiedDate' < "@date@" or 'ReturnDate' < "@date@" or 'DisposalDate' < "@date@"))	ModifiedDate	BMC.CORE:BMC_BaseElement-ARC



The CMDB is different than the other root requests as a CI is an entry in a complex Join form. Every other root request is an entry in a normal form.

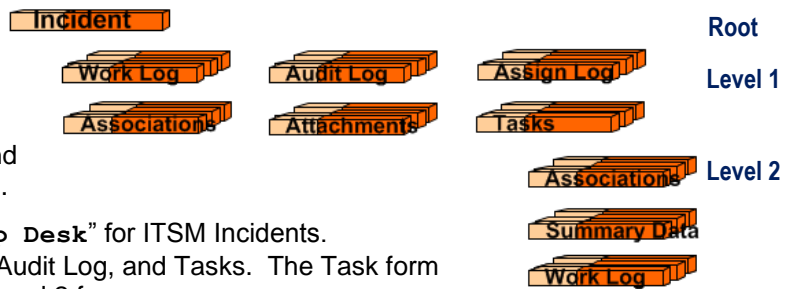
To determine records that can be archived, the root requests' Status field is generally used in the qualification. In ITSM 8.1 the Status of a CI was separated from BaseElement into a new normal form.

The "Archive Form" is a formula that suffixes the "Archive Table Name Suffix" on the "config" sheet to the Root's "Form" name.

tree sheet

For each root request defined in the root sheet, a two level tree of forms are defined in the tree sheet.

The tree sheet describes, for each root request, the set of level one and two forms and how they are related.



In the image, the root is "HPD:Help Desk" for ITSM Incidents.

Level 1 forms would be Work Log, Audit Log, and Tasks. The Task form has its own children which are the level 2 forms.

Tasks and their level 2 forms will be specified twice for different root requests (eg: for Incidents and Changes).

This sheet must have rows added for your customized forms.

Root Form	Enable	Delete Flag	Lvl	Source Form	Upper Form	Source Field	Upper Field	Archive Form	Source Field View
CHG:Infrastruct	1	1	1	AP:Detail	CHG:Infra: Request		1	AP:Detail-ARC	Request
CHG:Infrastruct	1	1	2	AP:Signature	AP:Detail	Approval ID	1	AP:Signature-AR	Approval_ID
CHG:Infrastruct	1	1	1	CFG:Broadcast	CHG:Infra: Request ID02	Infrastructure Change	CFG:Broadcast-ARC	Request_ID02	Request_ID02
CHG:Infrastruct	1	1	1	CFG:Reminders	CHG:Infra: LinktoEntryID		179	CFG:Reminders-ARC	LinktoEntryID
CHG:Infrastruct	1	1	1	CHG:Associations	CHG:Infra: Request ID02	Infrastructure Change	CHG:Association	Request_ID02	Request_ID02
CHG:Infrastruct	1	1	1	CHG:Attachments	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:Attachmen	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	0	1	1	CHG:Change Request Audit	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:Change Request	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:ChangeAPDetailSignature	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:ChangeAPDetail	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:Change-CI-Associations	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:Change-CI-Asso	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:ChangeRequest_AuditLogSystem	CHG:Infra: Original Request ID	Request ID	CHG:ChangeRequest	Request_ID	Request_ID
CHG:Infrastruct	1	1	1	CHG:ChangeRequest_AuditLogSystem	CHG:Infra: Original Request ID	Infrastructure Change	CHG:ChangeRequest	Original_Request_ID	Original_Request_ID
CHG:Infrastruct	1	1	1	CHG:ChangeRiskDerivedFactors	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:ChangeRiskDeri	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:ChangeRiskFactors	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:ChangeRiskFact	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:ChangeRiskFactorsDraft	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:ChangeRiskFact	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:CostAllocation	CHG:Infra: zID Change ID	Infrastructure Change	CHG:CostAllocation-	zID_Change_ID	zID_Change_ID
CHG:Infrastruct	1	1	1	CHG:Impacted Areas	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:Impacted Area	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:Infra. Change Effort Log	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:Infra. Change El	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:InfrastructureChangeAPDetail	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:InfrastructureC	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	CHG:WorkLog	CHG:Infra: Infrastructure Change	Infrastructure Change	CHG:WorkLog-ARC	Infrastructure_Chang	Infrastructure_Chang
CHG:Infrastruct	1	1	1	FIN:Association	CHG:Infra: Instance ID 01		179	FIN:Association-i	Instance_ID_01
CHG:Infrastruct	1	1	2	FIN:Costs	FIN:Associ	179	FIN Record Instance IC	FIN:Costs-ARC	instanceld
CHG:Infrastruct	1	1	1	NTE:Notifier Log	CHG:Infra: Source_Instance_ID		179	NTE:Notifier Log	Source_Instance_ID
CHG:Infrastruct	1	1	1	TMS:Association	CHG:Infra: RootRequestInstance		179	TMS:Association	RootRequestInstance
CHG:Infrastruct	1	1	1	TMS:Flow	CHG:Infra: RootRequestInstance		179	TMS:Flow-ARC	RootRequestInstance

The "Root Form" must be one of the Root Forms defined in the root sheet.

The configuration row is ignored if "Enable" is not set to "1".

The "Lvl" (level) value can be 1 or 2. A Level 2 form must follow the level 1 form it is dependent on. The level 2's "Upper Form" must equal the Level 1's "Source Form" that it is dependent on. A level 1's "Upper Form" must equal the "Root Form".

"Source Form" is the name of the form that will contain records related to its "Upper Form".

In the case of level 1, the "Source Form" records are related to the root request and so the "Upper Form" is the same as the "Root Form". For Level 2 Source Forms, the dependency is to a level 1 form, so its "Upper Form" will be a level 1 "Source Form",

"Upper Field" is the field name or id from the "Upper Form" to use in a query on the "Source Form" with the "Source Field" – for example:

'Source Field' = "-Upper Form's Upper Field's value-"

The "Source Field View" is simply the field's Remedy generated SQL name.

"Query" is an alternative way to link a tree Source Form to its Upper Form. If "Query" is specified, the Upper Field and Source Field will *not* be used to derive the query and will effectively be ignored.

The "Query" value has dollar sign wrapped fields of the upper form replaced with the values from that upper form record.

The above example could also be represented by a "Query" value of

'Source Field' = "\$Upper Field\$"

In the sample configuration file, there are some examples of this columns' use.

"Query SQL" is an SQL translation for the Query value. For the above "Query" example, this could be used for the Query SQL value.

'Source Field' = "\$Upper Field\$"

Both or neither Query and Query SQL must be used together.

The "Archive Form" is a formula that suffixes the "Archive Table Name Suffix" on the "Config" sheet to the "Source Form" name.

Root Form	Enable	Delete Flag	Lvl	Source Form	Upper Form	Source Field	Upper Field
HPD:Help Desk	1	1	1	TMS:Task	HPD:Help Desk	RootRequestInstanceID	179
HPD:Help Desk	1	1	2	TMS:AuditLog	TMS:Task	Audit Record Instance ID	179
HPD:Help Desk	1	1	2	TMS:AuditLogSystem	TMS:Task	Original Request ID	1
HPD:Help Desk	1	1	2	TMS:Metrics	TMS:Task	Task ID	1
CHG:Infrastructure Change	1	1	1	TMS:Task	CHG:Infrastructure Change	RootRequestInstanceID	179
CHG:Infrastructure Change	1	1	2	TMS:AuditLog	TMS:Task	Original Request ID	179
CHG:Infrastructure Change	1	1	2	TMS:Metrics	TMS:Task	Task ID	1
CHG:Infrastructure Change	1	1	2	TMS:MetricsSummary	TMS:Task	Task ID	1

The extract above shows how TMS:Task is related to both HPD:Help Desk and CHG:Infrastructure Change on level 1. Once an Incident or Change is loaded, all related Tasks can be queried. Once a Task is loaded, the level 2 forms like TMS:AuditLog can be queried.

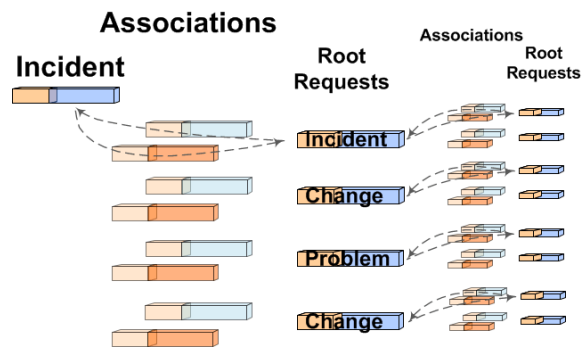
All fields for the CMDB should be based on **BMC.CORE:BMC_BaseElement** – even for ITSM 8.1 and higher, and even for the root **AST:BaseElement**.

assoc sheet

The assoc sheet describes the set of associations to other root requests.

This ensures records can only be archived when not associated to other root requests that are not ready for archiving.

For example, an Incident is ready for archival but it is associated with a Change that is not ready to be archived, then that Incident should not be archived until the Change can be archived.



The assoc sheet is used to generate Remedy and SQL queries to validate that all of the associated root requests' status and date fields indicate that they are ready for archive.

Root Form	Enabled	Src Asc Form	Tgt Form	Age	Src Tgt Form Field	Src Tgt Form Value	Tgt Asc Form	Src Asc Form Field	Src Asc Form ID Field	Src Asc Form ID Field	Src Asc Form ID Field	Tgt Form ID Field	Tgt Form Date Field	Tgt Form Status Field	Status Value Names	Status Values	
CHG:Infrastructure Change	1	CHG:Associations	AST:BaseElement	1y	Request Type01	Configuration Item	AST:BaseElement	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Reconciliation Identity	Modified Date	AssetLife Cycle Status	End of Life, Disposed	6, 11
CHG:Infrastructure Change	0	CHG:Associations	BMC.CORE:BMC_BaseElement	1y	Request Type01	Configuration Item	BMC.CORE:BMC_BaseElement	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Reconciliation Identity	ModifiedDate	AssetLife Cycle Status	End of Life, Disposed	6, 11
CHG:Infrastructure Change	1	CHG:Associations	CHG:Infrastructure Change	1y	Request Type01	Infrastructure Change	CHG:Associations	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Infrastructure Change ID	Completed Date	Change Request Status	Closed, Cancelled	11, 12
CHG:Infrastructure	1	CHG:Associations	HPD:Help Desk	1y	Request Type01	Incident	HPD:Associations	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Incident Number	Closed Date	Status	Closed, Cancelled	5, 6
CHG:Infrastructure	1	CHG:Associations	PBM:Problem	1y	Request Type01	Problem	PBM:Problem	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Problem Number	Closed Date	Investigation Status	Cancelled	7000, 8000
CHG:Infrastructure	1	CHG:Associations	PBM:Known Error	1y	Request Type01	Known Error	PBM:Known Error	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Known Error Number	Archive Date	Known Status	No Action	3, 4, 5, 6
CHG:Infrastructure	1	CHG:Associations	PBM:Solution	1y	Request Type01	Solution	PBM:Solution DB	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Solution Number	Archive Date	Status	Archive, Cancelled	4, 5
CHG:Infrastructure Change	1	CHG:Associations	RMS:Release	1y	Request Type01	Release	RMS:Associations	Request ID02	Request ID01	Request ID02	Request ID01	Infrastructure Change ID	Release ID	Last Modified Date	Release Status	Cancelled, Closed	7000, 8000
HPD:Help Desk	1	HPD:Associations	AST:BaseElement	1y	Request Type01	Configuration Item	AST:BaseElement	Request ID02	Request ID01	Request ID02	Request ID01	Incident Number	Reconciliation Identity	Modified Date	AssetLife Cycle Status	End of Life, Disposed	6, 11
HPD:Help Desk	0	HPD:Associations	BMC.CORE:BMC_BaseElement	1y	Request Type01	Configuration Item	BMC.CORE:BMC_BaseElement	Request ID02	Request ID01	Request ID02	Request ID01	Incident Number	Reconciliation Identity	ModifiedDate	AssetLife Cycle Status	End of Life, Disposed	6, 11
HPD:Help Desk	1	HPD:Associations	HPD:Help Desk	1y	Request Type01	Incident	HPD:Associations	Request ID02	Request ID01	Request ID02	Request ID01	Incident Number	Incident Number	Closed Date	Status	Closed, Cancelled	5, 6
HPD:Help Desk	1	HPD:Associations	PBM:Problem	1y	Request Type01	Problem	PBM:Problem	Request ID02	Request ID01	Request ID02	Request ID01	Incident Number	Problem Number	Closed Date	Investigation Status	Cancelled	7000, 8000

“Root Form” is one of the Forms in the root sheet. A Remedy query is generated for each entry in this sheet for each record returned by a root query. If a “Root Form” on the root sheet is not enabled, the assoc configuration for this root will be ignored.

“Enable” must contain a “1” to not ignore this record.

Note that there are two records for each association to the CMDB. The Enabled column on these records is a formula based on the ITSM Release. Notwithstanding the two different root forms, all qualifications for the CMDB are based on the BMC.CORE:BMC_BaseElement form.

“Src Asc Form” is the name of the form used in making associations from the current root form to other root forms.

In ITSM, this is generally an “Associations” table. For Change, CHG:Associations, for Incident, HPD:Associations.

“Tgt Form” is the root form that is the associated root. Consider that the associations are from the “Src Asc Form” to the “Tgt Form”.

“Age” is set by formula to the same value as the “Age” of the root form in the root sheet. These may be adjusted as needed.

“**Src Tgt Form Field**” identifies the field that distinguishes the different roots that an Association form can point to. As held in the “**Tgt Form**” field. In ITSM this is generally an “enum: or “selection” field.

“**Src Tgt Form Value**” identifies the value that the “**Src Tgt Form Field**” field will have for this entry – generally the enum value corresponding to the “**Tgt Form**”.

“**Query SQL and**” allows an optional value to be “anded” to the SQL query generated to count any associations not old enough or in the right status. The text will be wrapped in parentheses and preceded by “and”. Consider this as a restriction to what can be counted when querying the associated root.

For the CMDB for example, you may want only the production dataset involved so that an automated update to a different dataset will not prevent a ticket associated to it to be archived. Further, some tickets may be associated to CIs that are Active and that association will need to be ignored.

We could set “**Query SQL and**” to something like this:

```
DatasetId = 'BMC:Asset' and ClassId not in ('BMC_BusinessService')
```

A query is generated for CHG:Associations for each root along the lines of:

```
'Request Type01' = "Configuration Item" and  
'Request ID02' = "CRQ_CAL_1000022"
```

For each record from CHG:Associations, an SQL query is generated like:

```
select count(*)  
from BMC_CORE_BMC_BaseElement  
where ReconciliationIdentity = 'OI-3B4B98E90B13442EA985FD2AAD4F8C12'  
and (DatasetId = 'BMC:Asset' and  
ClassId not in ('BMC_BusinessService')  
)  
and not ( AssetLifecycleStatus in ( 6, 8, 11 )  
and ModifiedDate <= 1417078378 )
```

If the returned count is greater than 0, the root cannot be archived.

CMDB sheet

The CMDB sheet simply lists the BMC Class forms that will participate in Archiving and the Archive form names for those classes.

Class Form	Enable	Archive Form
BMC.CORE:BMC_ConnectivityCollection	1	BMC.CORE:BMC_ConnectivityCollection-ARC
BMC.CORE:BMC_IPConnectivitySubnet	1	BMC.CORE:BMC_IPConnectivitySubnet-ARC
BMC.CORE:BMC_Role	1	BMC.CORE:BMC_Role-ARC
BMC.CORE:BMC_Share	1	BMC.CORE:BMC_Share-ARC
BMC.CORE:BMC_TapeDrive	1	BMC.CORE:BMC_TapeDrive-ARC
BMC.CORE:BMC_ElementLocation	1	BMC.CORE:BMC_ElementLocation-ARC
BMC.CORE:BMC_UserCommunity	1	BMC.CORE:BMC_UserCommunity-ARC
BMC.CORE:BMC_Activity	1	BMC.CORE:BMC_Activity-ARC
BMC.CORE:BMC_ApplicationService	1	BMC.CORE:BMC_ApplicationService-ARC
BMC.CORE:BMC_Cluster	1	BMC.CORE:BMC_Cluster-ARC
BMC.CORE:BMC_DataBaseStorage	1	BMC.CORE:BMC_DataBaseStorage-ARC
BMC.CORE:BMC_DiskPartition	1	BMC.CORE:BMC_DiskPartition-ARC
BMC.CORE:BMC_FloppyDrive	1	BMC.CORE:BMC_FloppyDrive-ARC
BMC.CORE:BMC_HardwarePackage	1	BMC.CORE:BMC_HardwarePackage-ARC
BMC.CORE:BMC_HardwareSystemComponent	1	BMC.CORE:BMC_HardwareSystemComponent-ARC
BMC.CORE:BMC_IPEndpoint	1	BMC.CORE:BMC_IPEndpoint-ARC
BMC.CORE:BMC_Patch	1	BMC.CORE:BMC_Patch-ARC
BMC.CORE:BMC_PhysicalLocation	1	BMC.CORE:BMC_PhysicalLocation-ARC
BMC.CORE:BMC_SystemSoftware	1	BMC.CORE:BMC_SystemSoftware-ARC
BMC.CORE:BMC_ConcreteCollection	1	BMC.CORE:BMC_ConcreteCollection-ARC
BMC.CORE:BMC_ResourceAllocationSettingData	1	BMC.CORE:BMC_ResourceAllocationSettingData-ARC
BMC.CORE:BMC_FileSystem	1	BMC.CORE:BMC_FileSystem-ARC
BMC.CORE:BMC_LogicalSystemComponent	1	BMC.CORE:BMC_LogicalSystemComponent-ARC
BMC.CORE:BMC_Organization	1	BMC.CORE:BMC_Organization-ARC
BMC.CORE:BMC_Package	1	BMC.CORE:BMC_Package-ARC

“**Class Form**” is the name of the BMC Join form that holds data for the class. This usually starts with “**BMC.CORE**”. It is not the Asset Management Join – starting with “**AST:**”

“**Enable**” must contain a “1” to not ignore this record.

The “**Archive Form**” is a formula that suffixes the “**Archive Table Name Suffix**” on the “**Config**” sheet to the “**Class Form**” name.

The “**Skip Fields**” is normally not used. It is meant to correct situations where the Archive Server and the Source Server are not identical. It is only used in Identical Forms archives.

“**Skip Fields**” contains a comma separated list of field names or ids belonging to the class that will not be copied to the archive records – in any of the forms that make up the class.

Note that you will be making an incomplete archive if you use this column and on a restore these fields’ values will not be restored. Archiving to another server should be on identical forms.

Spaces are significant and should not be in the string. An example:

“**NormalizationStatus,1030007480**”

HTML sheet

The HTML sheet gives the HTML generation parameters for the tree of forms that will be archived to HTML files.

Forms in this sheet must also be defined and Enabled in the root and tree sheets. Only forms accumulating tables or creating Detail or Summary files need be mentioned in this sheet.

	Root Form	Enable	Lvl	Source Form	Upper Form	Detail?	Id Field	Detail title	Detail open	Detail record	Detail close	Detail fields	Detail table usage	Detail table
1														
2	CHG:Infra	1	0			1	Infra	Archived Change	ptn\chg.ptn	Request			Request ID,Submitter,Submit Date,Assign	
3	CHG:Infra	1	1	AP:Detail	CHG:Inf	1							Form=AP:Tag=APROV, post="</table>"	
4	CHG:Infra	1	2	AP:Signature	AP:Det	1							Tag=SIG, post="</table>" loop	
5	CHG:Infra	1	1	CFG:Broadcast										
6	CHG:Infra	1	1	CFG:Reminders										
7	CHG:Infra	1	1	CHG:Associations										
8	CHG:Infra	1	1	CHG:Attachments										
9	CHG:Infra	0	1	CHG:Change Request Audit										
10	CHG:Infra	0	1	CHG:ChangeAPDetailSignature										
11	CHG:Infra	1	1	CHG:Change-CI-Associations										
12	CHG:Infra	1	1	CHG:ChangeRequest_AuditLogS										
13	CHG:Infra	1	1	CHG:ChangeRequest_AuditLogS										
14	CHG:Infra	1	1	CHG:ChangeRiskDerivedFactors										
15	CHG:Infra	1	1	CHG:ChangeRiskFactors										
16	CHG:Infra	1	1	CHG:ChangeRiskFactorsDraft										
17	CHG:Infra	1	1	CHG:CostAllocation										
18	CHG:Infra	1	1	CHG:Impacted Areas										
19	CHG:Infra	1	1	CHG:Infra. Change Effort Log										
20	CHG:Infra	1	1	CHG:InfrastructureChangeAPDet										
21	CHG:Infra	1	1	CHG:WorkLog				WorkLog \$1\$					Tag=WL, post="</table>" loop	
22	CHG:Infra	1	1	FIN:Association										
23	CHG:Infra	1	2	FIN:Costs										
24	CHG:Infra	1	1	NTE:Notifier Log										
25	CHG:Infra	1	1	TMS:Association										
26	CHG:Infra	1	1	TMS:Flow										
27	CHG:Infra	1	1	TMS:Task										
28	CHG:Infra	0	2	TMS:AuditLogSystem										
29	CHG:Infra	1	2	TMS:Metrics										
30	CHG:Infra	0	2	TMS:MetricsSummary										
31	CHG:Infra	0	2	TMS:Relationships										
32	CHG:Infra	0	2	TMS:SummaryData										
33	CHG:Infra	0	2	TMS:TaskEffort										
34	CHG:Infra	0	2	TMS:WorkInfo										

“Lvl” is used here for both the root form and the level 1 and 2 tree forms for which HTML is to be generated. Set “Lvl” to zero for root forms.

“Enable” must be set to 1 to not ignore this record.

“Source Form” and “Upper Form” are the same as on the “tree” sheet. They should be blank for root forms.



“**Index fields**” can only be used in a root form row. It specifies the fields and optional labels that will be used to create a table of all archived roots as an index.html file and one index.csv file per root request.

It is a comma separated list of entries that are either a single field name, or a single field name followed by an '@' and then followed by a table label for that field's values.

Here's an example of an “Index fields” value for Incidents and the table generated.

Entry ID,Status,Description@Summary,Owner Group,Assigned Group

<i>Incident Number</i>	<i>Date</i>	<i>Entry ID</i>	<i>Status</i>	<i>Summary</i>	<i>Owner Group</i>	<i>Assigned Group</i>
INC_CAL_1000025	2008/11/06 22:14:17	INC0000000000027	Closed	Setup and Install Computer System Hardware.	Service Desk	Service Desk
INC_CAL_1000021	2008/11/06 22:14:17	INC0000000000023	Closed	Need drivers available on Network for Print Server.	Backoffice Support	Backoffice Support
INC_CAL_1000015	2008/11/06 22:14:17	INC0000000000017	Closed	Backup Dev and Sandbox Server.	Backoffice Support	Backoffice Support
INC_CAL_1000006	2015/11/08 17:57:52	INC0000000000008	Closed	Install Wireless Connectivity to Network.	Backoffice Support	Backoffice Support
INC_CAL_1000023	2008/11/06 22:14:17	INC0000000000025	Closed	Prepare System for Distribution.	Service Desk	Service Desk
INC_CAL_1000018	2008/11/06 22:14:17	INC0000000000020	Cancelled	Install latest Security Patch on server.	Backoffice Support	Backoffice Support
INC_CAL_1000008	2008/11/06 22:14:16	INC0000000000010	Closed	Setup Access Badge for Employee.	Service Desk	Service Desk

“**Detail?**” must be set to 1 to generate an HTML Detail file.

This file always has a fixed name starting with the Archive Job run prefix (changes depending on Run Automation) followed by the name of the root, the Id Field value for the root, the name of the level 1 form, the Id Field value for that level 1 form record. It is suffixed with “-dtl.html”

“**Id Field**” identifies a field to be used for generating the file name and title string. If not specified, the Request ID or field id 1 is used.

“**Detail title**” is used to generate the HTML page title. If blank, a default title will be generated. Field Names from the root record may be specified with \$ as in Remedy workflow. The values will be used in the title. Some examples may help:

Archived Change: \$Infrastructure Change ID\$

**Archived Change: \$Infrastructure Change ID\$ - \$3\$ -
\$Categorization Tier 1\$ - \$Categorization Tier 2\$ -
\$Categorization Tier 3\$**

“**Detail record**” this is a “pattern file” used when the creating an HTML file. If a pattern file is used, that pattern file must have all substitutions enclosed in dollar signs. Substitutions permitted depend on the “lv1” or may be global tags set by Meta-Archive. This is discussed in more detail below.

Pattern files are found along the **SthScriptPath** environment variable. If running Meta-Archive on Linux, the path separator must be changed to a forward slash. The sample pattern files specify a backslash as the directory separator.

“**Detail fields**” is only used if there is no pattern file specified. A two column table is generated containing the field names and the values. If blank, all fields are used. If specified, the order of the fields is the order that the HTML table will contain.

"Detail table" is used on level 1 or level 2 forms only. It accumulates an HTML table – in a Meta-Update reference – for all records on this level related to the upper table.

It does not change the generated HTML for this Source Form.

For level 1, this table can then be used on the root form. For level 2, the table can be used on a level 1 form and also be accumulated on a level 1 table as a column.

The "Detail table" value has some components that must have these keywords:

- Tag=** This can be referenced in a pattern file as `$html, tag$`
- Max=** This sets a limit on the number of entries to be part of the table. If zero or missing, all entries are used.
- Pre=** This is used to initialize the text for the table and generally will contain the table HTML tag and the first row of field headers. If missing, all fields will be used.

The **Pre=** value can contain two special keywords: **@Attachments** and **@Links**.

If a form has attachment fields, the **@Attachments** will be replaced by an html link to the saved attachment file.

@Links will be replaced with the link to the records generated HTML file(s).

- Post=** This is used to terminate the text for the table and generally will contain the table HTML end tag. If missing, a simple "`</table>`" will be used.

- Loop=** This is used to terminate the text for the table row containing the data. If missing, the Fields list will be used. Wrap the fields needed in dollar signs.

If this is a level 1 form accumulating a level 2 table, that level two table is part of the level 1 table. Use the special keyword "**@Table**" in the **Loop=** value to indicate where the level 2 table text is to be inserted.

The "**Detail table usage**" value must also be set for a level 1 table to accumulate text from a level 2 table.

- Fields=()** This is a list of fields in the form used if there is no **Loop=** specified. It is a list of fields separated by commas. This is only used if **Loop=** is missing. If this is missing all fields will be part of the row.

The following example will accumulate all Work Log records as a table which can then be used in a root's pattern file with the tag `$html, WL$`

```
Tag=WL, Post="</table>"
```

```
Loop="<tr><td>$Work Log Submit Date$</td><td>$Work Log
Submitter$</td><td>$Work Log Type$</td><td>$Detailed
Description$</td></tr>"
```

```
Pre="<table width=80% border=1><tr
valign=top><th>Date</th><th>Submitter</th><th>Type</th><th>Note
s</th></tr>"
```

```
max=0,
```

```
fields(Work Log Submit Date,Work Log Submitter,Work Log
Type,Detailed Description)
```

An example of usage in a pattern file:

```
<h3>Work Logs</h3>
```

\$html WL\$

“**Detail table usage**” is only used on a level 1 form. It indicates that the level one form will use a level 2 forms table text in its table text. It also indicates that the level 2 table be included in the level 1’s generated HTML page.

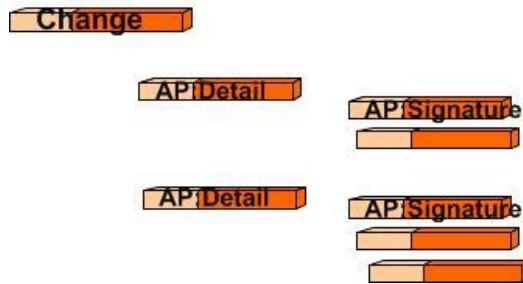
The “**Detail table usage**” value has some components that must have these keywords:

- Tag=** This is the level 2 “Detail table” tag. The data on the level 2 table is accumulated before the level 1 table’s HTML file is written.
- Form=** This is the level 2 table’s “Source Form” name.
- Name=** This is HTML text to use for the name of the level 2 table.

The following example will show the use of these two fields to create a blended table in the root requests, HTML file, as well as a simple table in the level 1’s html file.

A change request can require multiple approvals. Each approval may need several signatures.

A single change record may have two approvals needed. The first approval needs two signatures, the second three



A single change record - CHG:Infrastructure Change – has two approval records and five signatures. However, the first two signatures are related to the first approval – and not related to the second, and the next three signatures are related to the second approval – and not the first.

An HTML table on the Change itself must be organized as an embedded table to show the relationships.

Root Form	Enable	Lvl	Source Form	Upper Form	Detail?	Id Field	Detail title	Detail open	Detail record	Detail close	Detail fields	Detail table usage	Detail table
CHG:Infrastructure Change	1	0			1	Infrastructure Change ID	Archived Change: \$Infrastructure Change ID\$	ptn\chg.ptn			Request ID,Submitter,Submit Date.Ass		
CHG:Infrastructure Change	1	1	AP:Detail	CHG:Infrastructure Change	1							Form=AP:Signature, Tag=SIG, Name=<h3>Signatures</h3>	Tag=APROV, Post="</table>" Loop="<tr><td>\$Status-Dtl\$</td><td>\$Create-Date-
CHG:Infrastructure Change	1	2	AP:Signature	AP:Detail	1								Tag=SIG, post="</table>" loop="<tr><td>\$Approval Status\$</td><td>\$Approvers\$</td><td>\$Create-Date-

The above images shows the HTML configuration for a Change, its approvals, and signatures: CHG:Infrastructure Change (a root or level 0), AP:Detail (on level 1), and its child, AP:Signature, on

The "Detail table" configuration for the level 2 table, AP:Signature, accumulates all the AP:Signature records related to a single parent record in AP:Detail constantly adding to a string of HTML text. That text can be used in both the HTML generated for the level 1 record and the HTML generated for the root Change record.

Tag=SIG,

Post="</table>"

Loop="<tr><td>\$Approval Status\$</td><td>\$Approvers\$</td><td>\$Create-Date-Sig\$</td><td>\$Signature Due Date\$</td><td>\$Signature Method\$</td></tr>"

Pre="<table width=100% border=1><tr valign=top><th>Status</th><th>Approver</th><th>Date</th><th>Signature</th><th>Due Date</th><th>Method</th></tr>"

Max=10,

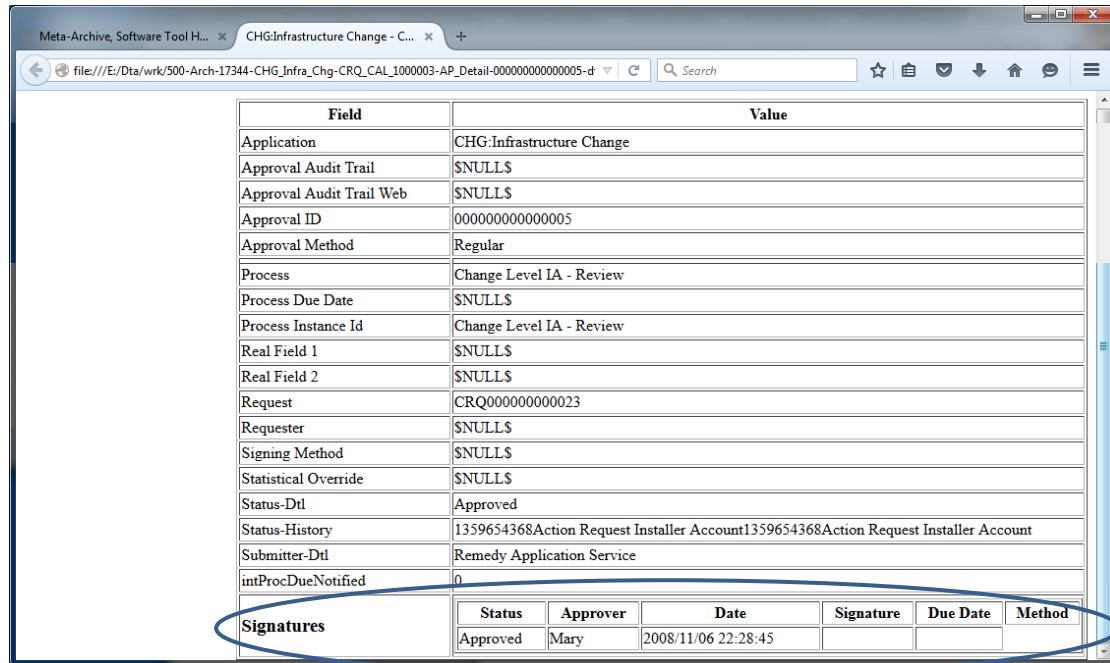
Fields (Approval Status,Approvers,Create-Date-Sig,Signature Due Date,Signature Method)

The “Detail table usage” configuration of the level 1 table, AP:Detail defines the use of the level 2 signatures table.

Form=AP:Signature, Tag=SIG, Name=<h3>Signatures</h3>

It is the Tag “SIG” that connects the level 1 and level 2 html text.

This is a view of the HTML page generated for a single AP:Detail record – with the “Detail table usage” and “Detail table” configured but no pattern files or detail fields configured.



Field	Value												
Application	CHG:Infrastructure Change												
Approval Audit Trail	\$NULL\$												
Approval Audit Trail Web	\$NULL\$												
Approval ID	000000000000005												
Approval Method	Regular												
Process	Change Level IA - Review												
Process Due Date	\$NULL\$												
Process Instance Id	Change Level IA - Review												
Real Field 1	\$NULL\$												
Real Field 2	\$NULL\$												
Request	CRQ000000000023												
Requester	\$NULL\$												
Signing Method	\$NULL\$												
Statistical Override	\$NULL\$												
Status-Dtl	Approved												
Status-History	1359654368Action Request Installer Account1359654368Action Request Installer Account												
Submitter-Dtl	Remedy Application Service												
intProcDueNotified	0												
Signatures	<table border="1"> <thead> <tr> <th>Status</th> <th>Approver</th> <th>Date</th> <th>Signature</th> <th>Due Date</th> <th>Method</th> </tr> </thead> <tbody> <tr> <td>Approved</td> <td>Mary</td> <td>2008/11/06 22:28:45</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Status	Approver	Date	Signature	Due Date	Method	Approved	Mary	2008/11/06 22:28:45			
Status	Approver	Date	Signature	Due Date	Method								
Approved	Mary	2008/11/06 22:28:45											

In the generated level 1 HTML above, the Signature table (highlighted in blue) is added as a row to the list of fields and values. The “Name” specified in the “Detail table usage” is set as the field name, and the accumulated HTML table text for the records of the level 2 table as the value.

Because the level 1 form, also accumulates HTML table text, the level 2 HTML table is incorporated into that level 1 table with an instance in each row, rather than as a “field” above. This then is referenced in the root forms HTML.

Tag=APROV,

Post="</table>"

Loop="<tr><td>\$Status-Dtl\$</td><td>\$Create-Date-Dtl\$</td><td>\$Priority\$</td><td>\$Completed?\$</td><td>\$Process\$</td><td>@Table</td></tr>"

Pre="<table width=80% border=1><tr valign=top><th>Status</th><th>Date</th><th>Priority</th><th>Completed?</th><th>Process</th><th><big>Signatures</big></th></tr>"

Max=10,

Fields (Status-Dtl, Create-Date-Dtl, Priority, Completed?, Process)

The AP:Signature table – with the Meta-Update reference \$html, SIG\$ - is brought in by the @Table keyword in the Loop= for accumulating rows.

In the Change request configuration a pattern file is used to select and order specific fields of that pattern.

The references \$html, WL\$ and \$html, APROV\$ are used to position the accumulated Work Log and Approvals table. The Approval table has a column containing the accumulated Signature records.

In the Change root's pattern file, you will find this:

```
<tr valign=top><td><h3>Work Logs</h3></td>
      <td>$html, WL$</td></tr>
<tr valign=top><td><h3>Approvals</h3></td>
      <td>$html, APROV$</td></tr>
```

The screenshot shows a web application interface for a change request. The main content area displays the following details:

- Description:** Setup new user with basic system permissions.
- Category:** Request - Account - Create
- Company:** Calbro Services
- Organization:** Information Technology
- Assigned Support Organization:** Calbro Services IT Support Frontoffice Support Ian Plyment
- CAB Manager (Change Co-ord):** Mary Mann
- Environment:** \$NULL\$
- Name:** Unser, Joe J.Unser@calbroservices.com 1 212 555-5454 (66)
- Location:** Calbro Services, Information Technology, IT Support Service Desk
- Company:** Calbro Services
- Urgency / Impact / Priority:** 3-Medium / 2-Significant/Large / Medium
- Reason For Change:** Other
- Dates:**
 - Actual Start Date: 2008/10/04 07:00:00
 - Earliest Start Date: 2008/10/01 07:00:00
 - Scheduled Start Date: 2008/10/05 07:00:00
 - Requested Start Date: 2008/10/05 07:00:00
 - Actual End Date: 2008/10/09 15:00:00
 - Scheduled End Date: 2008/10/09 07:00:00
 - Requested End Date: 2008/10/10 07:00:00
 - RFC Date: 2008/10/02 07:00:00

Below the details are two tables:

Work Logs			
Date	Submitter	Type	Notes
2008/11/06 22:28:45	Remedy Application Service	General Information	Work Log Entry from Ian.

Approvals													
Status	Date	Priority	Completed?	Process	Signatures								
Approved	2008/11/06 22:28:45	Normal		Change Level IA - Review	<table border="1"> <thead> <tr> <th>Status</th> <th>Approver</th> <th>Date</th> <th>Signature Due Method</th> </tr> </thead> <tbody> <tr> <td>Approved</td> <td>Mary</td> <td>2008/11/06 22:28:45</td> <td></td> </tr> </tbody> </table>	Status	Approver	Date	Signature Due Method	Approved	Mary	2008/11/06 22:28:45	
Status	Approver	Date	Signature Due Method										
Approved	Mary	2008/11/06 22:28:45											

While will then show the two level 1 table (in brown) and the embedded level 2 table (in blue). That embedded level 2 table text is the same as the one used in the level 1 HTML page above.



Archive-Cfg-Csv-Tbls.xlsm

This file is only used when the target is CSV and / or HTML files. It is used to produce shorter file names by abbreviating table names.

Meta-Archive constructs output file names by concatenating these parts:

File name string	Description
Prefix	The prefix is set by Job Automation or defaults to "500- Arch-pid "- where "pid" is the process id of the archive job.
hyphen	
Root form name	All form names have special characters removed.
hyphen	
Level 1 form name	This is the form name with all special characters removed.
hyphen	
Level 2 form name	This is the form name with all special characters removed.

Attachments additionally have these parts suffixed to the above.

File name string	Description
hyphen	
Request Id	The records' request id – field '1',
Field Id	The Attachment field's id.

For CSV file output ".csv" is always concatenated to the above. Attachments retain their original extension. Within the CSV file, the attachment value is the original attachment source file name.

Examples for Incident might be:

```
500-Arch-9138-HPD_Help_Desk.csv
500-Arch-9138-HPD_Help_Desk-HPD_WorkLog.csv
500-Arch-9138-HPD_Help_Desk-HPD_WorkLog-INC0001079816-1000000351.log
500-Arch-9138-HPD_Help_Desk-HPD_WorkLog-INC0001079816-1000000352.zip
```

Or with Job Automation:

```
100-Inc-HPD_Help_Desk.csv
100-Inc-HPD_Help_Desk-HPD_WorkLog.csv
100-Inc-HPD_Help_Desk-HPD_WorkLog-INC0001079816-1000000351.log
100-Inc-HPD_Help_Desk-HPD_WorkLog-INC0001079816-1000000352.zip
```

The file **Archive-Cfg-Csv-Tbls.xlsm** may be used to change these file names.



Software Tool House Inc