UNIT 4 BUSINESS SOFTWARE

EMBRACING CHANGE
Business World UK Product

Server Process Toolkit for Interfaces

Configuration Guide

Version 3.2

CONTENTS

1 Introduction

This document provides a description of the various functions (methods) available in the Server Process Toolkit for Interfaces (ITK) and on how to configure those methods on a UNIT4 Business World (BW) system.

This manual is for ITKv3 which has been developed in C#[1] and ACT[2] for use on more recent milestones of BW and is still a work in progress because some of the methods have not been developed yet. These methods are greyed out as follows:

Method that has not yet been implemented.

If you are still using ITKv2 which was developed in VB6[3] and AgrComSrv[4], then please refer to the version of the Configuration Guide that accompanied that software.

See section 1.2 for more details on the differences between ITKv3 and ITKv2.

How to install ITK is detailed in a separate Installation Guide.

1.1 Overview

UNIT4 Business World (BW) has an extensive set of data import and data export facilities which can cater for most of a customer's needs without any programming being required. These facilities include:

- Standard import routines with 'query' facilities (CS15, GL07, LG04, etc.)
- ASQL[5] queries used to import data (COPY IN, COPY FROM)
- ASQL queries used to export data (COPY TO)
- ARW[6] reports to export data (more complex formats)

Six shortcomings of the above are:

- The ASQL COPY commands can only handle two types of file:
  - One fixed length record per line where each field is in the same position on each record and there are no separators between fields,
  - One variable length record per line with each field separated by a delimiting character, usually a 'Tab' (ASCII[7] code 9), but can also be a "@", "£", "$", "%", "&", "." Or ";" character.
- When an interface uses a non-standard import file that file is not automatically deleted at the end of the run, so it is possible for a user to process the same file twice.

[1] A Microsoft developed .NET programming language, pronounced "See-sharp".
[2] Agresso Customisation Tool: a library used to interact with any object in newer Agresso versions.
[3] Microsoft Visual Basic 6.
[4] Agresso COM Server: a library used to interact with server processes in older Agresso versions.
[5] Agresso Structured Query Language
[6] Agresso Report Writer
[7] See section 5.2.

- On an update of an existing row “Master file” import routines such as CS15 do not allow you to blank out columns or remove addresses or relation values (where available).
- The ASQL COPY TO command will fail if the file it is trying to write to already exists.
- Whilst an ARW report can create extremely complex output files, including CSV and XML the file is usually placed in the ‘Report Results’ folder with a name indistinguishable from that of any normal report. There is an ARW command that overrides the default name of an output file but this is not very flexible.
- An ARW can only create one output file and many standard server processes do not give you the ability to run additional reports or run a query.
- If an ARW is exporting suppliers (say) as XML and the receiving systems schema can only accept one supplier per file then ARW’s ability to only creating one file is a problem.

ITK addresses these issues by providing the following methods:

- A start-of-process method that will convert “Character Separated Value” (CSV) files to Tab Separated Value (TSV) files that can then be read by the ‘COPY IN’ ASQL command in the process’ query.
- A start-of-process method that will convert “Counted Fields per Record” (CFR) files to Tab Separated Value (TSV) files that can then be read by the ‘COPY IN’ ASQL command in the process’ query.
  See ‘CFR to TSV File Conversion’ for a description of this file format.
- A start-of-process method that will convert “Form-feed Separated Record” (FSR) files to Tab Separated Value (TSV) files that can then be read by the ‘COPY IN’ ASQL command in the process’ query.
  See ‘CFR to TSV File Conversion’ for a description of this file format.
- A start-of-process method that will delete export files.
- An end-of-process method that will delete (or move to the “scratch” folder) non-standard import files
- A calling-the-report or start-of-process method that will blank columns that have been changed to a particular value and also remove addresses and/or relation values that have been set to the same value.
- An end-of-process method that copies or moves output files created by an ARW (for instance) out of the Report Results folder and give them a new name.
- An end-of-process method that runs additional reports.
- A start-of-process or end-of-process method that runs an AG16 query.
- A start-of-process or end-of-process method that calls a command-line tool with parameters. The command-line tool can be a script (bat, cmd, vbs, etc) or a program (in-house, UNIT4 or third party).

- A number of methods end-of-process can split XML output into multiple files at a defined element and also make the payload “XML safe”

In some versions of BW when you use any of the above batch end methods on a server process that has produced no printout then you may get an error:
“Error -2147417848: Method ‘~’ of object ‘~’ failed”,
ITK provides a start-of-process method to resolve this problem.

1.2 Differences in ITKv3

ITKv3 is different to ITKv2 in a number of ways:

1. As mentioned in the introduction it is a work in progress because three of the ITKv2 methods have yet to be implemented (at version 3.1.0).
2. The rewrite from VB6/AgrComSrv to C#/ACT has resulted in a number of benefits:
    a. You are no longer restricted to running a process on a 32bit server queue in order to use ITK.
    b. It is no longer a requirement to register a dll on the Business Server as a part of the installation of ITK, and so (with one possible exception) the entire installation can be carried out from within the Desktop client.
3. The old ‘load_bi’, ‘load_rep’ and ‘load_be’ process parameters were BW’s mechanism for calling VB6 server exits and:
    a. Therefore these parameters are not used in ITKv3 and **MUST** be removed from any report variants that previously used ITKv2.
    b. Are replaced with new process parameters: ‘itk_start’, ‘itk_rep’ and ‘itk_stop’ respectively. Note that in the current version of ITKv3 the ‘itk_rep’ parameter is not used because the methods that use it have not yet been implemented.
4. The ‘FSR Import File Conversion (ConvFsr)’ method now accepts ‘num_headers’ and ‘num_footers’ parameters.
5. In ITKv2 only the ‘Copy or Move Output Files as a Process Ends (Export)’ method would replace file tags (see section 4.3). In ITKv3 the ‘Remove Output Files as a Process Starts (DelFile)’, ‘Run a Query as a Process Starts (BiQuery)’ and ‘Run a Query as a Process Ends (BeQuery)’ methods replace file tags in addition to ‘Export’.
6. Because the “itk…” parameters are “normal” process parameters, it has been possible to enhance ITK in version 3.1 that allows multiple values of these parameters in the same variant, see section 2.1 for more details.

2 Configuration

In order to make use of an ITK method on a standard server process you must create a variant of that process that has some additional process parameters. In order to make use of an ITK method on a user defined report you must add some additional process parameters to its definition or a variant of the process. In general these parameters vary from method to method, but they all have one thing in common:

- If you wish to use a start-of-process method then you must create a parameter with its 'ParamID' set to "itk_start" and its 'Default value' must be the name of the method you wish to use.
- If you wish to use an end-of-process method then you must create a parameter with its 'ParamID' set to "itk_stop" and its 'Default value' must be the name of the method you wish to use.

A server process variant or user defined report can have an 'itk_start' parameter, an 'itk_stop' parameter or both present.

WARNING

If you are modifying a report variant or user defined report to use ITKv3 instead of ITKv2 then you must replace the existing 'load_bi' and 'load_be' process parameters with 'itk_start' and 'itk_stop' respectively and change their 'Default value' cells accordingly. NOTE then at this release there are no valid methods on 'load_rep'/'itk_rep'.

WARNING

Due to a restriction of ACT, ITK methods can only be used on server processes, therefore if you wish to use a method on a simple report then you must convert that report into a server process. See section 4.1 for details of how to do this.

2.1 Multiple ITK… Parameters

As mentioned above the "itk_start" or "itk_stop" parameter must be used to run an ITK method. A significant enhancement to this is that it is now possible to run multiple methods on the same event. Currently this has only been explicitly implemented for the "Extra Report (Extra)" method, but the intention is to make this possible for as many of the existing methods as possible. At the current release the only methods that can be safely defined with an "itk…" parameter with a numeric suffix are:

Method	Description
Extra	Run an additional report as a process ends
Export	Copy or move output files as a process ends

For the first ITK method on an event the corresponding 'itk_*xxxx*' parameter should be used (where "*xxxx*" is either "start" or "stop"), for the second a parameter 'itk_*xxxx*_2' should be used for the third a parameter 'itk_*xxxx*_3' and so on up to a maximum of 20. When a method is defined by an "itk…" parameter with a numeric suffix then all of its additional parameters must also have the same numeric suffix. For example if you wish to run two extra reports on a process and then export one of the report files then you might define the following parameters:

ParamID	Param name	Default value
itk_stop	First On Stop action	Extra
x_rep	Report to run	MYREP1
itk_stop_2	Second On Stop action	Extra
x_rep_2	Report to run	MYREP2
x_test_2	Report help-table	helptab0
itk_stop_3	Third On Stop action	Export
move_1	Move file to	c:\temp\...

Note that the value of the suffix of the 'move…' parameter (i.e. one) is unrelated to the value of the suffix of its owning 'itk_stop…' parameter (e.g. three); for an explanation of why this is the case, please refer to the 'Copy or Move Output Files as a Process Ends (Export)' method in section 3.7.

3 The Methods

3.1 CSV Import File Conversion (ConvCsv)

3.1.1 Introduction

See section 5.1 for an overview of using queries to import data.

For many years there has been a de-facto standard for data transfer files known as character separated value (CSV) traditionally the character used was a comma and some people still use CSV to mean “comma separated value”, but other characters such as the pipe “|” and tilde “~” are not uncommon.

Unfortunately this format cannot be read by the ASQL COPY IN command. This fact has been a minor implementation issue on a number of occasions.

This method will take one or more existing CSV files in the folder defined by the “AGRESSO_IMPORT” environment variable and translate them into new files where the fields are tab-delimited, and processes text fields so that they are not enclosed by apostrophes or speech marks and removes any doubled apostrophes or speech marks within them. It then changes the value of the “file_name” parameters so that the SQL query can process the tab-delimited versions of the CSV files and finally it moves the original CSV files to the folder defined by the “AGRESSO_SCRATCH” environment variable, or the folder defined by the ‘TEMP’ Windows environment variable if AGRESSO_SCRATCH is not set. The new tab-separated files will have a “tsv” extension and will be present in the AGRESSO_IMPORT folder.

If a file has header records and/or footer records then an additional field will be inserted at the beginning of each record of the new “tsv” file, this field will be of the form:

“*Xn*”

Where *X* is

- “H” on a header record,
- “D” on a detail record
- “F” on a footer record

And *n* is a sequence number which starts from one for each record type. For example:

- “H1” is the first header record,
- “D1” is the first detail record,
- “D235” is the two hundredth and thirty fifth detail record
- “F1” is the first footer record

If a file has neither header records nor footer records then this extra field is not created.

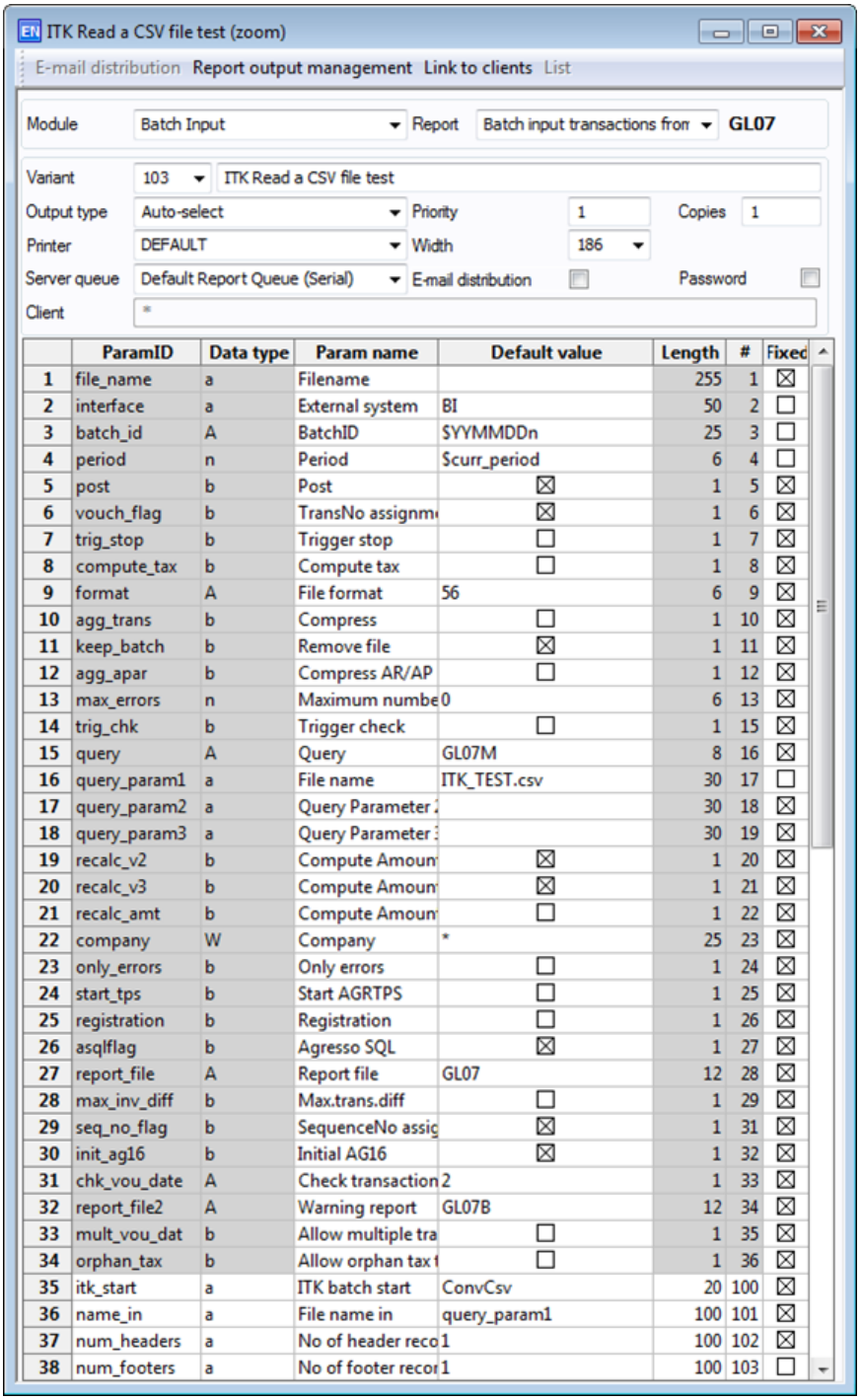
💣 WARNING

The contents of a header or footer record will be put into the first data field of the output record with an appropriate number of trailing tab characters. So when you use the “COPY IN” command it is important that the table column you are copying the first data field to is large enough to hold any possible header or footer data. For example in the query referred to in section 3.1.3 the ‘account’ column is defined as “SPACE(25)” so that it is large enough to hold the date from the header (10 characters would have been enough for this) or the

total debit amount from the footer (the line amount is 15 characters so 18 or 20 should have been enough but you can't be too careful!).
Similarly the table column that the additional field that the program generates is going to be copied into ('rec_type' in the query below) must be big enough to hold at least the expected number of detail records, if you could have a million detail records then the column must be at least eight characters long to hold "D1000000".

3.1.2 Set Up

In order to process CSV files in a data import routine you must set up a variant with some existing parameters changed and some new parameters added. The example is for GL07 but the same principle applies to any other data import routine.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	file_name	a	Filename		255	1	☒
2	interface	a	External system	BI	50	2	☐
3	batch_id	A	BatchID	$YYMMDDn	25	3	☐
4	period	n	Period	$curr_period	6	4	☐
5	post	b	Post	☒	1	5	☒
6	vouch_flag	b	TransNo assignm	☒	1	6	☒
7	trig_stop	b	Trigger stop	☐	1	7	☒
8	compute_tax	b	Compute tax	☐	1	8	☒
9	format	A	File format	56	6	9	☒
10	agg_trans	b	Compress	☐	1	10	☒
11	keep_batch	b	Remove file	☒	1	11	☒
12	agg_apar	b	Compress AR/AP	☐	1	12	☒
13	max_errors	n	Maximum numbe	0	6	13	☒
14	trig_chk	b	Trigger check	☐	1	15	☒
15	query	A	Query	GL07M	8	16	☒
16	query_param1	a	File name	ITK_TEST.csv	30	17	☐
17	query_param2	a	Query Parameter		30	18	☒
18	query_param3	a	Query Parameter		30	19	☒
19	recalc_v2	b	Compute Amoun	☒	1	20	☒
20	recalc_v3	b	Compute Amoun	☒	1	21	☒
21	recalc_amt	b	Compute Amoun	☐	1	22	☒
22	company	W	Company	*	25	23	☒
23	only_errors	b	Only errors	☐	1	24	☒
24	start_tps	b	Start AGRTPS	☐	1	25	☒
25	registration	b	Registration	☐	1	26	☒
26	asqlflag	b	Agresso SQL	☒	1	27	☒
27	report_file	A	Report file	GL07	12	28	☒
28	max_inv_diff	b	Max.trans.diff	☐	1	29	☒
29	seq_no_flag	b	SequenceNo assig	☒	1	31	☒
30	init_ag16	b	Initial AG16	☒	1	32	☒
31	chk_vou_date	A	Check transaction	2	1	33	☒
32	report_file2	A	Warning report	GL07B	12	34	☒
33	mult_vou_dat	b	Allow multiple tra	☐	1	35	☒
34	orphan_tax	b	Allow orphan tax	☐	1	36	☒
35	itk_start	a	ITK batch start	ConvCsv	20	100	☒
36	name_in	a	File name in	query_param1	100	101	☒
37	num_headers	a	No of header reco	1	100	102	☒
38	num_footers	a	No of footer recor	1	100	103	☐

Parameter ‘file_name’

The ‘Default value’ must be blank and ‘Fixed’ must be checked so that GL07 does not try to read the file as a standard GL07 file.

Parameter ‘query’

The ‘Default value’ must be set to the appropriate query’s name.

Parameter ‘query_param1’ (or similar)

Change the ‘Param name’ to “File name” and unfix it.

Parameter ‘asqlflag’

If your query is written in ASQL then change the ‘Default value’ to a 1.

New Parameter ‘itk_start’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	“ConvCsv”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘name_in’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to convert (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	One more than the ‘#’ value of the ‘itk_start’ parameter.
Fixed	Checked

New Parameter ‘num_headers’

This parameter is optional so if none of your files have header records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	“a”
Default value	• If all files in ‘name_in’ have the same number of header records then set ‘Default value’ to that single number. • If the files in ‘name_in’ have different numbers of header records then set ‘Default value’ to a comma separated list of numbers of header records, each value corresponds to the file in the same position in ‘name_in’ and there must be the same number of values as there are in ‘name_in’.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'num_footers'

This parameter is optional so if none of your files have footer records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	"a"
Default value	• If all files in 'name_in' have the same number of footer records then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of footer records then set 'Default value' to a comma separated list of numbers of footer records, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'separator'

This parameter is optional so if your file uses a comma as its field separator then it can either be omitted or can be included and its value set to a comma. If present it must be set as follows:

Column	Value
Data type	"a"
Default value	• If all files in 'name_in' have the same field separator character then set 'Default value' to that single character. • If the files in 'name_in' have different field separator characters then set 'Default value' to a comma separated list of single characters, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

💣 WARNING

If you wish to explicitly define a comma as the field separator character then you must enclose it in apostrophes thus: ','. Failure to do this will result in ITK miscounting the number of values in this parameter.

New Parameter 'asc_only'

This parameter is optional so if all characters in the import file can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then it can either be omitted or can be included and its value set to "N".

If the parameter is present it must be set as follows:

Column	Value
Data type	"A"
Default value	• If all the files in 'name_in' could contain non-ASCII characters and you want them removed then set 'Default value' to "Y". • If all of the files in 'name_in' can only contain ASCII characters or they could contain non-ASCII characters but you don't want them removed then set 'Default value' to "N". • If you want to remove non-ASCII characters from only some of the files in 'name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, , each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

3.1.3 Example File and Query

As an example here is one of the files and queries used during testing; it posts a simple GL journal and relates to the report variant screen shot above. The file header contains a date to be used as the transaction date and the footer contains a total-debit figure that the query will "throw" an error on if it is incorrect

The File

Line	Contents
1	01/03/2005
2	1032,100,'','','','','',1234.56,'Here is a narrative'
3	'CN002',120,'EX9804-2',,'','A-00','2','',-1234.56,'Here''s another narrative'
4	1234.56

The SQL Query

See section 4.2.

The Results

(AGRESSO DEMO) General Ledger transactions

	T	Trans.date	Account	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Text	Amount
1	B	01/03/2005	1032	100							Here is a narrative	1,234.56
2	B	01/03/2005	CN002	120	EX9804-2			A-00	2		Here's another narrative	-1,234.56
Σ												0.00

3.2 CFR Import File Conversion (ConvCfr)

3.2.1 Introduction

See section 5.1 for an overview of using queries to import data.

Some main-frame systems generate files that have a fixed number of fields per record and each field is on a separate line; there is no delimiting character at the end of a record, so the only way to know when you have reached the end of a record is to "count the fields". I have coined the acronym CFR (Counted Fields per Record) to describe this type of file.

Unfortunately this format cannot be read by the COPY IN command. This fact has been a minor implementation issue on a number of occasions.

This method will take one or more existing CFR files in the folder defined by the "AGRESSO_IMPORT" environment variable and translate them into new files where the fields are tab-delimited. It then changes the value of the "file_name" parameters so that the SQL query can process the tab-delimited versions of the CFR files and finally it moves the original CFR files to the folder defined by the "AGRESSO_SCRATCH" environment variable, or the folder defined by the 'TEMP' Windows environment variable if AGRESSO_SCRATCH is not set. The new tab-separated files will have a "tsv" extension and will be present in the AGRESSO_IMPORT folder.

The content of each line on the input file is not inspected by this method and so could contain more than one data item; if this is the case then your SQL query must break the field into its constituent data items using substring operations ("SUBSTR(…)" in ASQL).

If a file has header records and/or footer records then an additional field will be inserted at the beginning of each record of the new "tsv" file, this field will be of the form:

"*Xn*"

Where *X* is

- "H" on a header record,
- "D" on a detail record
- "F" on a footer record

And *n* is a sequence number which starts from one for each record type. For example:

- "H1" is the first header record,
- "D1" is the first detail record,
- "D235" is the two hundredth and thirty fifth detail record
- "F1" is the first footer record

If a file has neither header records nor footer records then this extra field is not created.

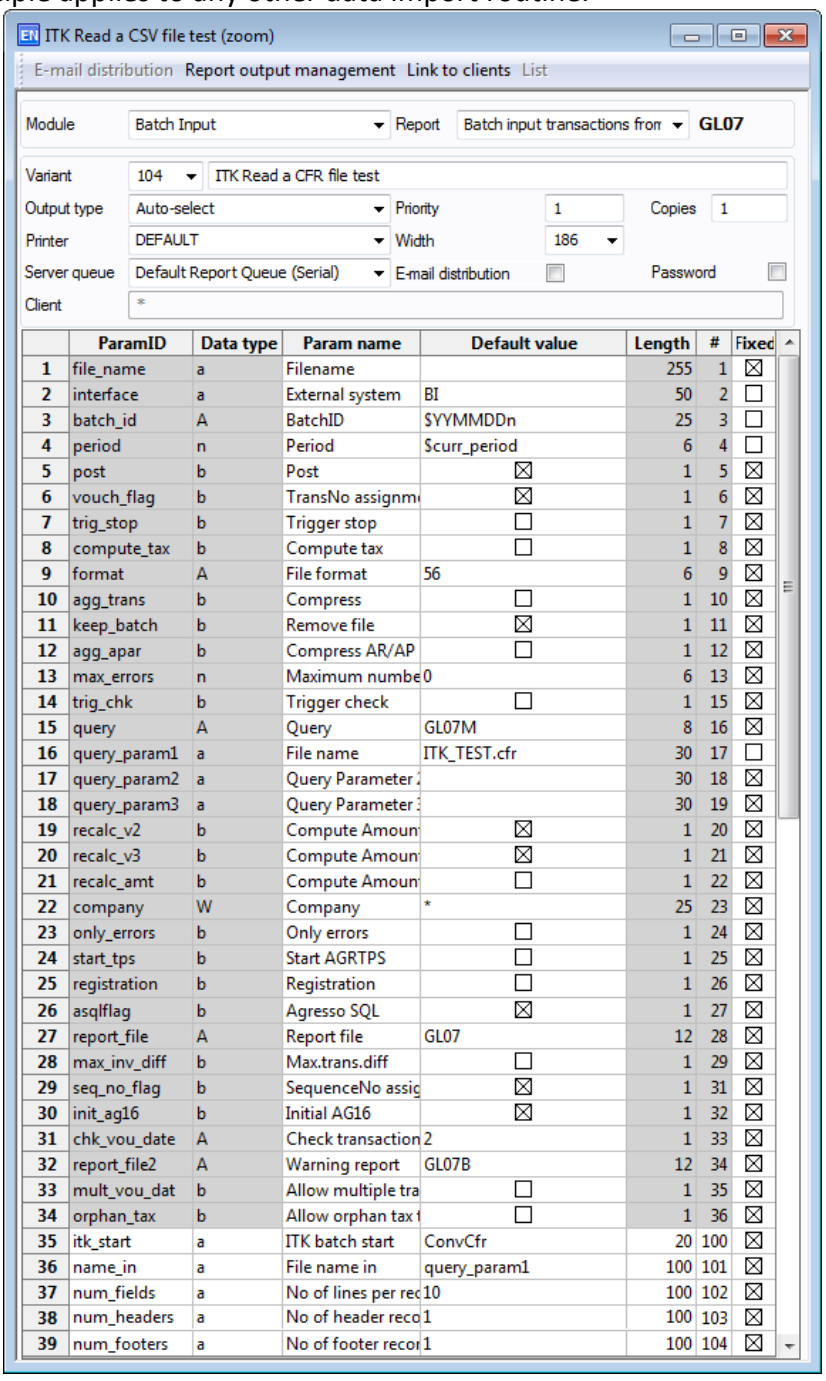
WARNING

The contents of a header or footer record will be put into the first data field of the output record with an appropriate number of trailing tab characters. So when you use the "COPY IN" command it is important that the table column you are copying the first data field to is large enough to hold any possible header or footer data. For example in the query refered to in section 3.2.3 the 'account' column is defined as "SPACE(25)" so that it is large enough to hold the date from the header (10 characters would have been enough for this) or the

total debit amount from the footer (the line amount is 15 characters so 18 or 20 should have been enough but you can't be too careful!).
Similarly the table column that the additional field that the program generates is going to be copied into ('rec_type' in the query below) must be big enough to hold at least the expected number of detail records, if you could have a million detail records then the column must be at least eight characters long to hold "D1000000".

3.2.2 Set Up

In order to process CFR files in a data import routine you must set up a variant with some existing parameters changed and some new parameters added. The example is for GL07 but the same principle applies to any other data import routine.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	file_name	a	Filename		255	1	☒
2	interface	a	External system	BI	50	2	☐
3	batch_id	A	BatchID	$YYMMDDn	25	3	☐
4	period	n	Period	$curr_period	6	4	☐
5	post	b	Post	☒	1	5	☒
6	vouch_flag	b	TransNo assignm	☒	1	6	☒
7	trig_stop	b	Trigger stop	☐	1	7	☒
8	compute_tax	b	Compute tax	☐	1	8	☒
9	format	A	File format	56	6	9	☒
10	agg_trans	b	Compress	☐	1	10	☒
11	keep_batch	b	Remove file	☒	1	11	☒
12	agg_apar	b	Compress AR/AP	☐	1	12	☒
13	max_errors	n	Maximum numbe	0	6	13	☒
14	trig_chk	b	Trigger check	☐	1	15	☒
15	query	A	Query	GL07M	8	16	☒
16	query_param1	a	File name	ITK_TEST.cfr	30	17	☐
17	query_param2	a	Query Parameter		30	18	☒
18	query_param3	a	Query Parameter		30	19	☒
19	recalc_v2	b	Compute Amoun	☒	1	20	☒
20	recalc_v3	b	Compute Amoun	☒	1	21	☒
21	recalc_amt	b	Compute Amoun	☐	1	22	☒
22	company	W	Company	*	25	23	☒
23	only_errors	b	Only errors	☐	1	24	☒
24	start_tps	b	Start AGRTPS	☐	1	25	☒
25	registration	b	Registration	☐	1	26	☒
26	asqlflag	b	Agresso SQL	☒	1	27	☒
27	report_file	A	Report file	GL07	12	28	☒
28	max_inv_diff	b	Max.trans.diff	☐	1	29	☒
29	seq_no_flag	b	SequenceNo assig	☒	1	31	☒
30	init_ag16	b	Initial AG16	☒	1	32	☒
31	chk_vou_date	A	Check transaction	2	1	33	☒
32	report_file2	A	Warning report	GL07B	12	34	☒
33	mult_vou_dat	b	Allow multiple tra	☐	1	35	☒
34	orphan_tax	b	Allow orphan tax t	☐	1	36	☒
35	itk_start	a	ITK batch start	ConvCfr	20	100	☒
36	name_in	a	File name in	query_param1	100	101	☒
37	num_fields	a	No of lines per rec	10	100	102	☒
38	num_headers	a	No of header reco	1	100	103	☒
39	num_footers	a	No of footer recor	1	100	104	☒

Parameter ‘file_name’

The ‘Default value’ must be blank and ‘Fixed’ must be checked so that GL07 does not try to read the file as a standard GL07 file.

Parameter ‘query’

The ‘Default value’ must be set to the appropriate query’s name.

Parameter ‘query_param1’ (or similar)

Change the ‘Param name’ to “File name” and unfix it.

Parameter ‘asqlflag’

If your query is written in ASQL then change the ‘Default value’ to a 1.

New Parameter ‘itk_start’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	“ConvCfr”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘name_in’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to convert (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	To one more than the ‘#’ value of the ‘itk_start’ parameter.
Fixed	Checked

New Parameter ‘num_fields’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”

Column	Value
Default value	• If all files in 'name_in' have the same number of fields per record then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of fields per record then set 'Default value' to a comma separated list of numbers of fields, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'num_headers'

This parameter is optional so if none of your files have header records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	"a"
Default value	• If all files in 'name_in' have the same number of header records then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of header records then set 'Default value' to a comma separated list of numbers of header records, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'num_footers'

This parameter is optional so if none of your files have footer records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	"a"
Default value	• If all files in 'name_in' have the same number of footer records then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of footer records then set 'Default value' to a comma separated list of numbers of footer records, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'

Column	Value
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'asc_only'

This parameter is optional so if all characters in the import file can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then it can either be omitted or can be included and its value set to "N".

If the parameter is present it must be set as follows:

Column	Value
Data type	"A"
Default value	• If all the files in 'name_in' could contain non-ASCII characters and you want them removed then set 'Default value' to "Y". • If all of the files in 'name_in' can only contain ASCII characters or they could contain non-ASCII characters but you don't want them removed then set 'Default value' to "N". • If you want to remove non-ASCII characters from only some of the files in 'name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, , each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

3.2.3 Example File and Query

As an example here is one of the files and queries used during testing; it posts a simple GL journal and relates to the report variant screen shot above. The file header contains a date to be used as the transaction date and the footer contains a total-debit figure that the query will "throw" an error on if it is incorrect

The File

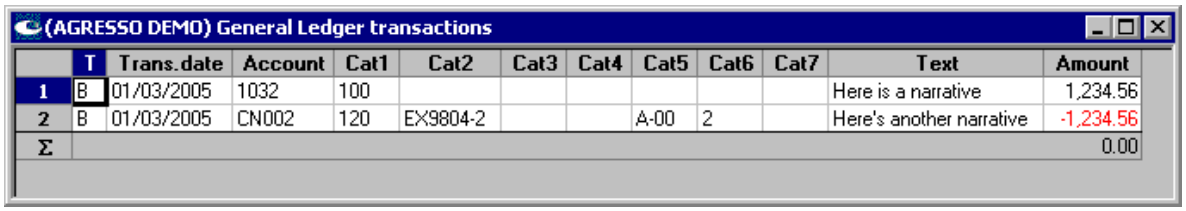
Line	Contents
1	01/03/2005
2	1032
3	100
4	
5	
6	
7	

Line	Contents
8	
9	
10	1234.56
11	Here is a narrative
12	CN002
13	120
14	EX9804-2
15	
16	
17	A-00
18	2
19	
20	-1234.56
21	Here's another narrative
22	1234.56

The SQL Query

See section 4.2.

The Results



	T	Trans.date	Account	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Text	Amount
1	B	01/03/2005	1032	100							Here is a narrative	1,234.56
2	B	01/03/2005	CN002	120	EX9804-2			A-00	2		Here's another narrative	-1,234.56
Σ												0.00

3.3 FSR Import File Conversion (ConvFsr)

3.3.1 Introduction

See section 5.1 for an overview of using queries to import data.

A popular way of exporting data from some systems is to use a non-graphical report writer. Some report writers do not have the ability as ARW does to suppress page throws and so a transfer file generated from such a report writer consists of:

- One page per record,
- Each record consisting of one or more lines with a fixed width, usually 80 or 132 characters
- Each line consists of either:
    - one field
    - multiple fields with the same field on each record being in the same character position in the same line on each record

Multiple lines appear in each record either because there is one field per record or because the number of fields required will not fit into the maximum allowed line width.

Because each record starts with a form-feed character (ASCII[8] code 12) I have coined the acronym FSR (Form-feed Separated Record) to describe this type of file.

Depending partly on the report writer used to create the file the form-feed character at the beginning of each record may always be on a line on its own or it may simply be the first character on the first line of the record.

This method will take one or more existing FSR files in the folder defined by the "AGRESSO_IMPORT" environment variable and translate them into new files where the fields are tab-delimited. It then changes the value of the "file_name" parameters so that the SQL query can process the tab-delimited versions of the FSR files and finally it moves the original FSR files to the folder defined by the "AGRESSO_SCRATCH" environment variable, or the folder defined by the 'TEMP' Windows environment variable if AGRESSO_SCRATCH is not set. The new tab-separated files will have a "tsv" extension and will be present in the AGRESSO_IMPORT folder.

The content of each line on the input file is not inspected by this method and so could contain more than one data item)see above); if this is the case then your SQL query must break the field into its constituent data items using substring operations ("SUBSTR(…)" in ASQL).

If a file has header records and/or footer records then an additional field will be inserted at the beginning of each record of the new "tsv" file, this field will be of the form:

"*Xn*"

Where *X* is

- "H" on a header record,
- "D" on a detail record
- "F" on a footer record

[8] See section 5.2

And *n* is a sequence number which starts from one for each record type. For example:

- "H1" is the first header record,
- "D1" is the first detail record,
- "D235" is the two hundredth and thirty fifth detail record
- "F1" is the first footer record

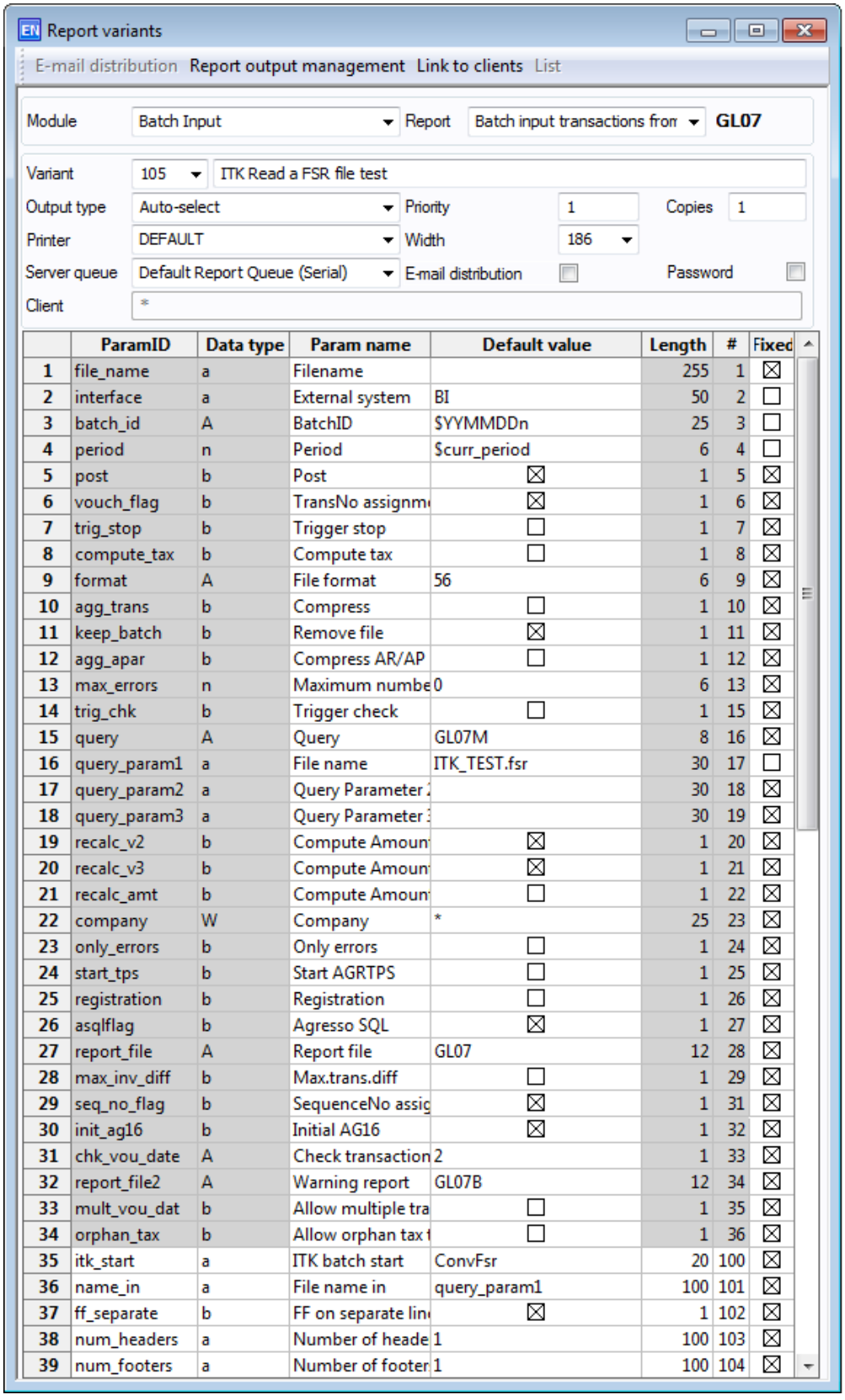
If a file has neither header records nor footer records then this extra field is not created.

💣 WARNING

The contents of a header or footer record will be put into the first data field of the output record with an appropriate number of trailing tab characters. So when you use the "COPY IN" command it is important that the table column you are copying the first data field to is large enough to hold any possible header or footer data. For example in the query referenced in section 3.3.3 the 'account' column is defined as "SPACE(25)" so that it is large enough to hold the date from the header (10 characters would have been enough for this) or the total debit amount from the footer (the line amount is 15 characters so 18 or 20 should have been enough but you can't be too careful!).
Similarly the table column that the additional field that the program generates is going to be copied into ('rec_type' in the query below) must be big enough to hold at least the expected number of detail records, if you could have a million detail records then the column must be at least eight characters long to hold "D1000000".

3.3.2 Set Up

In order to process FSR files in a data import routine you must set up a variant with some existing parameters changed and some new parameters added. The example is for GL07 but the same principle applies to any other data import routine.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	file_name	a	Filename		255	1	☒
2	interface	a	External system	BI	50	2	☐
3	batch_id	A	BatchID	$YYMMDDn	25	3	☐
4	period	n	Period	$curr_period	6	4	☐
5	post	b	Post	☒	1	5	☒
6	vouch_flag	b	TransNo assignm	☒	1	6	☒
7	trig_stop	b	Trigger stop	☐	1	7	☒
8	compute_tax	b	Compute tax	☐	1	8	☒
9	format	A	File format	56	6	9	☒
10	agg_trans	b	Compress	☐	1	10	☒
11	keep_batch	b	Remove file	☒	1	11	☒
12	agg_apar	b	Compress AR/AP	☐	1	12	☒
13	max_errors	n	Maximum numbe	0	6	13	☒
14	trig_chk	b	Trigger check	☐	1	15	☒
15	query	A	Query	GL07M	8	16	☒
16	query_param1	a	File name	ITK_TEST.fsr	30	17	☐
17	query_param2	a	Query Parameter		30	18	☒
18	query_param3	a	Query Parameter		30	19	☒
19	recalc_v2	b	Compute Amoun	☒	1	20	☒
20	recalc_v3	b	Compute Amoun	☒	1	21	☒
21	recalc_amt	b	Compute Amoun	☐	1	22	☒
22	company	W	Company	*	25	23	☒
23	only_errors	b	Only errors	☐	1	24	☒
24	start_tps	b	Start AGRTPS	☐	1	25	☒
25	registration	b	Registration	☐	1	26	☒
26	asqlflag	b	Agresso SQL	☒	1	27	☒
27	report_file	A	Report file	GL07	12	28	☒
28	max_inv_diff	b	Max.trans.diff	☐	1	29	☒
29	seq_no_flag	b	SequenceNo assig	☒	1	31	☒
30	init_ag16	b	Initial AG16	☒	1	32	☒
31	chk_vou_date	A	Check transaction	2	1	33	☒
32	report_file2	A	Warning report	GL07B	12	34	☒
33	mult_vou_dat	b	Allow multiple tra	☐	1	35	☒
34	orphan_tax	b	Allow orphan tax t	☐	1	36	☒
35	itk_start	a	ITK batch start	ConvFsr	20	100	☒
36	name_in	a	File name in	query_param1	100	101	☒
37	ff_separate	b	FF on separate lin	☒	1	102	☒
38	num_headers	a	Number of heade	1	100	103	☒
39	num_footers	a	Number of footer	1	100	104	☒

Parameter ‘file_name’

The ‘Default value’ must be blank and ‘Fixed’ must be checked so that GL07 does not try to read the file as a standard GL07 file.

Parameter ‘query’

The ‘Default value’ must be set to the appropriate query’s name.

Parameter ‘query_param1’ (or similar)

Change the ‘Param name’ to “File name” and unfix it.

Parameter ‘asqlflag’

If your query is written in ASQL then change the ‘Default value’ to a 1.

New Parameter ‘itk_start’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	“ConvFsr”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘name_in’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to convert (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	To one more than the ‘#’ value of the ‘itk_start’ parameter.
Fixed	Checked

New Parameter ‘ff_separate’

This parameter is mandatory and a value of “Y” means that the form-feeds in the files being processed are on lines of their own, a value of “N” means that the form-feeds in the files being processed are the first character of lines containing other data. The parameter must be set as follows:

Column	Value
Data type	“a”

Column	Value
Default value	• If all files in 'name_in' need the same 'ff_separate' value then set 'Default value' to that single value. • If the files in 'name_in' need different 'ff_separate' values then set 'Default value' to a comma separated list of values, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

Note that in the screen shot this parameter has been defined as Data type "b" so that it appears as a checkbox, this is acceptable because there is only one file being processed.

New Parameter 'num_headers'

This parameter is optional so if none of your files have header records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	"a"
Default value	• If all files in 'name_in' have the same number of header records then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of header records then set 'Default value' to a comma separated list of numbers of header records, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'num_footers'

This parameter is optional so if none of your files have footer records it can either be omitted or can be included and its value set to zero. If present it must be set as follows:

Column	Value
Data type	"a"

Column	Value
Default value	• If all files in 'name_in' have the same number of footer records then set 'Default value' to that single number. • If the files in 'name_in' have different numbers of footer records then set 'Default value' to a comma separated list of numbers of footer records, each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'asc_only'

This parameter is optional so if all characters in the import file can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then it can either be omitted or can be included and its value set to "N".

If the parameter is present it must be set as follows:

Column	Value
Data type	"A"
Default value	• If all the files in 'name_in' could contain non-ASCII characters and you want them removed then set 'Default value' to "Y". • If all of the files in 'name_in' can only contain ASCII characters or they could contain non-ASCII characters but you don't want them removed then set 'Default value' to "N". • If you want to remove non-ASCII characters from only some of the files in 'name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, , each value corresponds to the file in the same position in 'name_in' and there must be the same number of values as there are in 'name_in'.
Length	100
#	The next available number.
Fixed	Checked

3.3.3 Example File and Query

As an example here is one of the files and queries used during testing; it posts a simple GL journal and relates to the report variant screen shot above. The file header contains a date to be used as the transaction date and the footer contains a total-debit figure that the query will "throw" an error on if it is incorrect

The File

Line	Contents
1	01/03/2005
2	$^{F}_{F}$
3	1032
4	100
5	
6	
7	
8	
9	
10	
11	1234.56
12	Here is a narrative
13	$^{F}_{F}$
14	CN002
15	120
16	EX9804-2
17	
18	
19	A-00
20	2
21	
22	-1234.56
23	Here's another narrative
24	$^{F}_{F}$
25	1234.56

The SQL Query

See section 4.2.

The Results

(AGRESSO DEMO) General Ledger transactions

	T	Trans.date	Account	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Text	Amount
1	B	01/03/2005	1032	100							Here is a narrative	1,234.56
2	B	01/03/2005	CN002	120	EX9804-2			A-00	2		Here's another narrative	-1,234.56
Σ												0.00

3.4 Remove Import Files as a Process Ends (DelFile or MovFile)

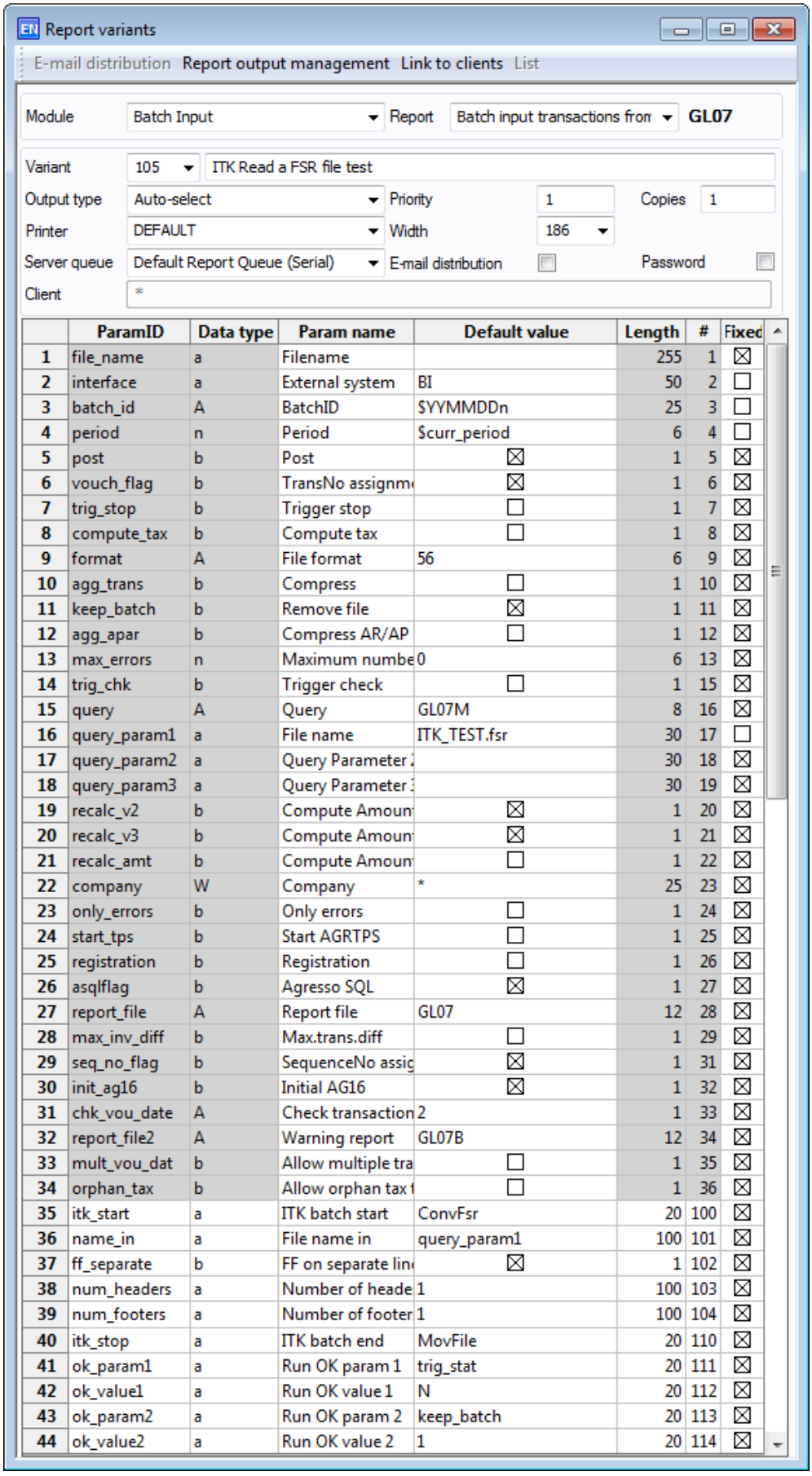
3.4.1 Introduction

When you are using an SQL query to read a file into an import routine the name of the file(s) are usually held in one or more of the 'query_param*n*' process parameters. These files unlike ones whose names are stored in process parameters such as 'file_name' are not removed from the 'Data Import' folder at the end of the run, so it is possible for a user to process the same file(s) twice.

These methods allow you to either delete the file(s) or move them to the BW "Scratch" folder and are typically used in combination with the 'BiQuery', 'ConvCsv', 'ConvCfr' and 'ConvFsr' methods.

3.4.2 Set Up

In order to use one these functions on a process you must set up a variant with some new parameters or add these parameters to your 'User defined report'. The example is for the GL07 and moves the file to scratch but the same principle applies to deleting the files and to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	file_name	a	Filename		255	1	☒
2	interface	a	External system	BI	50	2	☐
3	batch_id	A	BatchID	$YYMMDDn	25	3	☐
4	period	n	Period	$curr_period	6	4	☐
5	post	b	Post	☒	1	5	☒
6	vouch_flag	b	TransNo assignm	☒	1	6	☒
7	trig_stop	b	Trigger stop	☐	1	7	☒
8	compute_tax	b	Compute tax	☐	1	8	☒
9	format	A	File format	56	6	9	☒
10	agg_trans	b	Compress	☐	1	10	☒
11	keep_batch	b	Remove file	☒	1	11	☒
12	agg_apar	b	Compress AR/AP	☐	1	12	☒
13	max_errors	n	Maximum numbe	0	6	13	☒
14	trig_chk	b	Trigger check	☐	1	15	☒
15	query	A	Query	GL07M	8	16	☒
16	query_param1	a	File name	ITK_TEST.fsr	30	17	☐
17	query_param2	a	Query Parameter		30	18	☒
18	query_param3	a	Query Parameter		30	19	☒
19	recalc_v2	b	Compute Amoun	☒	1	20	☒
20	recalc_v3	b	Compute Amoun	☒	1	21	☒
21	recalc_amt	b	Compute Amoun	☐	1	22	☒
22	company	W	Company	*	25	23	☒
23	only_errors	b	Only errors	☐	1	24	☒
24	start_tps	b	Start AGRTPS	☐	1	25	☒
25	registration	b	Registration	☐	1	26	☒
26	asqlflag	b	Agresso SQL	☒	1	27	☒
27	report_file	A	Report file	GL07	12	28	☒
28	max_inv_diff	b	Max.trans.diff	☐	1	29	☒
29	seq_no_flag	b	SequenceNo assig	☒	1	31	☒
30	init_ag16	b	Initial AG16	☒	1	32	☒
31	chk_vou_date	A	Check transaction	2	1	33	☒
32	report_file2	A	Warning report	GL07B	12	34	☒
33	mult_vou_dat	b	Allow multiple tra	☐	1	35	☒
34	orphan_tax	b	Allow orphan tax t	☐	1	36	☒
35	itk_start	a	ITK batch start	ConvFsr	20	100	☒
36	name_in	a	File name in	query_param1	100	101	☒
37	ff_separate	b	FF on separate lin	☒	1	102	☒
38	num_headers	a	Number of heade	1	100	103	☒
39	num_footers	a	Number of footer	1	100	104	☒
40	itk_stop	a	ITK batch end	MovFile	20	110	☒
41	ok_param1	a	Run OK param 1	trig_stat	20	111	☒
42	ok_value1	a	Run OK value 1	N	20	112	☒
43	ok_param2	a	Run OK param 2	keep_batch	20	113	☒
44	ok_value2	a	Run OK value 2	1	20	114	☒

New Parameter ‘itk_stop’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	“DelFile” or “MovFile”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘name_in’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to convert (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	One more than the ‘#’ value of the ‘itk_stop’ parameter.
Fixed	Checked

Note that in the above screen shot the ‘name_in’ parameter is being shared with the ‘itk_start’ “ConvFsr” method so the file that will be moved to scratch is the “TSV” file created by the “ConvFsr” method.

New Parameters ‘ok_param*n*’ and ‘ok_value*n*’

These parameters are optional. If none of these parameters are present then the ‘DelFile’ and ‘MovFile’ methods will carry out their action on the files stored in the ‘name_in’ parameter whether the data import works or not.

If the data import fails and you delete the input file then you may not be able to re-process the data in that file. In order to minimise this problem up to five pairs of optional parameters are provided; ‘ok_param1’ and ‘ok_value1’ to ‘ok_param5’ and ‘ok_value5’…

Some server processes set an internal parameter to a specific value to indicate that no errors have occurred, if you set ‘ok_param1’ to the name of that parameter and ‘ok_value1’ to the value that means that there were no errors then the method will only carry out its action if the named parameter is set to the specified value.

If there are additional parameters that determine whether to delete the file or not even if the run is successful, then these can be defined with up to four additional pairs of parameters. All the defined parameters must have the specified values in order for the files to be deleted or moved.

The “ok_param*n*’ parameters are set up as follows:

Column	Value
Data type	“a”
Default value	The name of the parameter whose value is to be tested against the ‘ok_value*n*’ parameter with the same number..
Length	20
#	The next available number.
Fixed	Checked

The ‘ok_value*n*’ parameters are set up as follows:

Column	Value
Data type	“a”
Default value	The value of the parameter whose name is in the ‘ok_param*n*’ parameter with the same number that indicates that the run was successful.
Length	Long enough to hold the ‘Default value’
#	The next available number.
Fixed	Checked

Processes that set a “run OK” parameter include:

Process	ok_param1	ok_value1	ok_param2	ok_value2
GL07	trig_stat	N	keep_batch	1
SRP01	error#	0 (zero)	only_apar	N

3.5 Blank Columns and Remove Addresses and Relation Values (FldBlnk)

WARNING

The methods ‘FldBlnk’ and ‘FldBlnkPR43’ are not implemented in the current version of ITK.

3.6 Remove Output Files as a Process Starts (DelFile)

3.6.1 Introduction

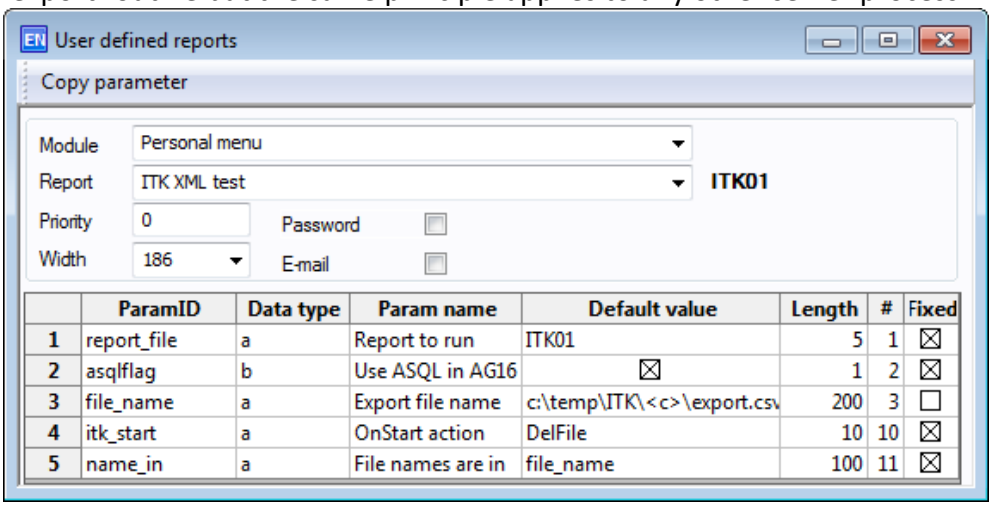
When you are using an SQL query to copy BW data into one or more export files the query may fail if any of the files already exist in the destination folder.

This method allows you to delete the file(s) before the query runs:

- The names of the files to be deleted must be held in one or more process parameters.

3.6.2 Set Up

In order to use this function in a process you must set up a variant with some new parameters or add these parameters to your 'User defined report'. The example is for a bespoke export routine but the same principle applies to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	report_file	a	Report to run	ITK01	5	1	☒
2	asqlflag	b	Use ASQL in AG16	☒	1	2	☒
3	file_name	a	Export file name	c:\temp\ITK\<c>\export.csv	200	3	☐
4	itk_start	a	OnStart action	DelFile	10	10	☒
5	name_in	a	File names are in	file_name	100	11	☒

New Parameter 'itk_start'

This parameter is mandatory and must be set as follows:

Column	Value
Data type	"a"
Default value	"DelFile"
Length	30
#	On a report variant this must be greater than or equal to "100", on a user defined report this can be the next available number.
Fixed	Checked

New Parameter 'name_in'

This parameter is mandatory and must be set as follows:

Column	Value
Data type	"a"

Column	Value
Default value	A comma separated list of the names of the parameters containing the file names to delete (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	To one more than the ‘#’ value of the ‘itk_start’ parameter.
Fixed	Checked

The ‘Default value’ of each of the parameters referred to in the ‘Default value’ of the ‘name_in’ parameter determines a file path and name that you want to delete:

- The value entered can include fixed text and one or more of the file tags described in section 4.3. The ‘Default value’ of each of these parameters is then changed so that your query can use the modified name.
- If the value entered is not a file path then ITK assumes the file is in the folder defined by the “AGRESSO_EXPORT” environment variable.
- If the value entered is a file path then that path will be used.

3.7 Copy or Move Output Files as a Process Ends (Export)

3.7.1 Introduction

Files created with the ASQL 'COPY TO' command can be placed anywhere on the file system using the "FILE=" parameter, with or without the "EXPORT" modifier. However files created by COPY TO have two restrictions:

- All records must have the same format
- The fields in a records must be either fixed width or delimited with one of the following characters:
  - Tab (ASCII[9] code 9)
  - "@" (at sign)
  - "£" (sterling sign)
  - "$" (dollar sign)
  - "%" (percentage symbol)
  - "&" (ampersand)
  - "." (full stop)
  - or ";" (semicolon)

  Using the Tab character is the safest option as this is a non-printing character and is therefore highly unlikely to appear in the data.
  It is possible to produce a CSV file using COPY TO as long as none of the text fields contain apostrophes by producing records containing a single field and concatenating all the data into that one field with commas separating each field and apostrophes around all text fields. E.g.

```
COPY TO EXPORT
       FILE = datafile.csv,
       COLSEP = T,
       SELECT CONCAT('''', CONCAT(client, CONCAT(''',', TO_CHAR(amount))))
```

Often we need to create more complex files with multiple record types and the easiest way to do this is to write an ARW to create the file. However the output of an ARW is saved in the 'Report Results' folder and has the form "*xxxxxy_n*.lis" where:

- *xxxxx* is the process code
- *y* is a sequence letter (a for the first report, b for the second, etc.) and
- *n* is the order number of the run

In order to transmit the file to another system it often needs to be stored elsewhere and ideally with a more meaningful name. ARW has an .OUTPUT command that allows you to specify the file name (and optional path) that your report's output be written to instead of the above "lis" file but you can only use a fixed file name: e.g. "C:\temp\myoutput.txt" with no variable elements, not even the order number of the run

This server exit provides a more flexible solution to the above than the 'OUTPUT command:

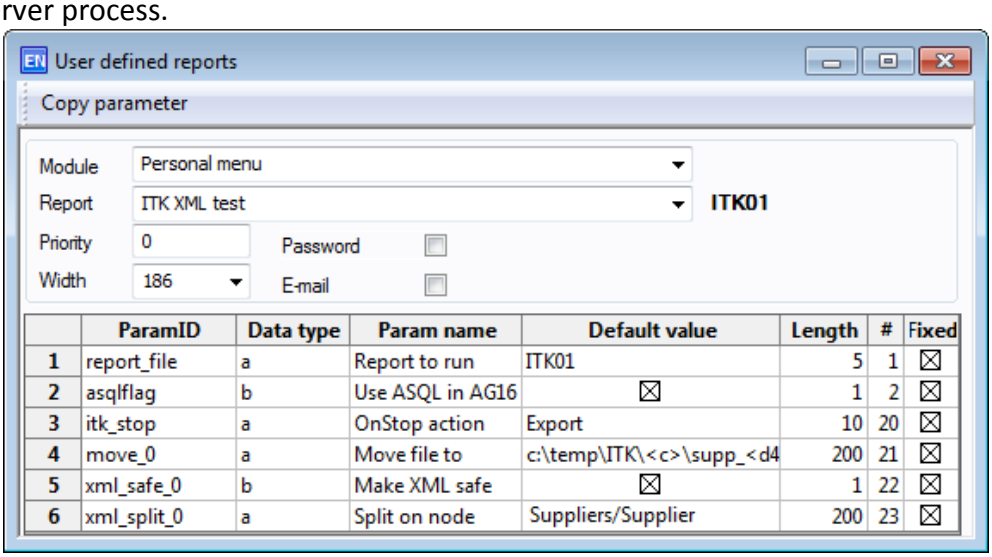
- Variables can be included anywhere in the file name
- The file name is a process parameter instead of hard-coded in the report

[9] See section 5.2.

- If the file name does not include a path then it is assumed to be in the folder defined by the "AGRESSO_EXPORT" environment variable.

3.7.2 Set Up

In order to use this server exit on a process you must set up a variant with some new parameters. The example is for bespoke export routine but the same principle applies to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	report_file	a	Report to run	ITK01	5	1	☒
2	asqlflag	b	Use ASQL in AG16	☒	1	2	☒
3	itk_stop	a	OnStop action	Export	10	20	☒
4	move_0	a	Move file to	c:\temp\ITK\<c>\supp_<d4	200	21	☒
5	xml_safe_0	b	Make XML safe	☒	1	22	☒
6	xml_split_0	a	Split on node	Suppliers/Supplier	200	23	☒

New Parameter 'itk_stop'

This parameter is mandatory and must be set as follows:

Column	Value
Data type	"a"
Default value	"Export"
Length	30
#	On a report variant this must be greater than or equal to "100", on a user defined report this can be the next available number.
Fixed	Checked

New Parameter 'copy_*n*' (or 'move_*n*')

This new parameter is an example of a group of parameters that you can set up to either copy or move one of the process' reports to another location and to give it a new name. All of these parameters have names in the form 'copy_*n*' or 'move_*n*' where *n* is the sequence number of the report file you wish to copy or move. The sequence number of the report file is shown in the '#' column in the **Report printout** screen (CR43). Sequence numbers greater than ten are not currently supported:

- Note that the first report output's sequence number is zero not one.
- If the process has multiple outputs you can leave gaps in the copy_*n* or move_*n* sequence; if you only want to copy the second output file then only define copy_1.
- You can mix 'copy' and 'move' parameters on the same variant as long as they are for different sequence numbers.

At least one of these parameters must be present and must be set as follows:

Column	Value
Data type	“a”
Default value	The name (or path) of the file name that you wish this report output to copied or moved to.
Length	200
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

- The ‘Default value’ entered can include fixed text and one or more of the file tags described in section 4.3.
- If the ‘Default value’ entered is not a file path then the file will be written to the folder defined by the “AGRESSO_EXPORT” environment variable.
- If the ‘Default value’ entered is a file path then that path will be used the name including the file name. If the folder structure in the file name does not exist then it will be created.
- In order to stop this server exit from causing serious damage to your system the ‘Default value’ is not allowed to end with “.dll”, “.exe” or “.ocx”

If the expanded file path and name equals the original file path and name then this is only allowed if at least one of the follow in true:

- The corresponding ‘asc_only_*n*’ is equal to “1”
- The corresponding ‘xml_safe_*n*’ parameter is equal to “1”
- The corresponding ‘xml_split_*n*’ parameter is not blank.

WARNING

The numeric suffix on this parameter and its linked ‘asc_only_*n*’, ‘xml_safe_*n*’ and ‘xml_split_*n*’ parameters has nothing to do with the mechanism for defining multiple ‘itk_stop’ parameters, it is the way you link this action to a particular output of the process.

New Parameter ‘asc_only_*n*’

This parameter is optional so if all characters in the file whose name is in the ‘copy_*n*’ or ‘move_*n*’ parameter with the same sequence number can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then this parameter can either be omitted or it can be included and its value set to unchecked.

If the parameter is present it must be set as follows:

Column	Value
Data type	“b”

Column	Value
Default value	• If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number could contain non-ASCII characters and you want them removed then set 'Default value' to checked • If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number can only contain ASCII characters or it could contain non-ASCII characters but you don't want them removed then set 'Default value' to unchecked.
Length	1
#	The next available number.
Fixed	Checked

You can have an 'asc_only_*n*' parameter for each 'copy_*n*' or 'move_*n*' that is present and its number must match

New Parameter 'blnk_lin_*n*'

This parameter is optional so if you do not wish to remove blank lines in the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number then this parameter can either be omitted or it can be included and its value set to checked.

If the parameter is present it must be set as follows:

Column	Value
Data type	"b"
Default value	• If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number cannot contain blank lines or it could contain blank lines but you don't want them removed then set 'Default value' to checked. • If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number can contain blank lines and you want them removed then set 'Default value' to unchecked
Length	1
#	The next available number.
Fixed	Checked

You can have a 'blnk_lin_*n*' parameter for each 'copy_*n*' or 'move_*n*' that is present and its number must match

New Parameter 'xml_safe_*n*'

This parameter is optional so if whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number does not contain XML or it does but you know that there cannot

be any unsafe characters in it then this parameter can either be omitted or it can be included and its value set to unchecked.

If the parameter is present it must be set as follows:

Column	Value
Data type	"b"
Default value	• If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number contains XML and may contain "unsafe" characters then set 'Default value' to "Y" • If the file whose name is in the 'copy_*n*' or 'move_*n*' parameter does not contain XML or it does but you know that there are cannot be any unsafe characters then set 'Default value' to unchecked.
Length	1
#	The next available number.
Fixed	Checked

You can have a 'xml_safe_*n*' parameter for each 'copy_*n*' or 'move_*n*' that is present and its number must match

If any the following characters are contained in the payload (or value) of an XML element then this will cause programs to fail to read that XML. If any these "unsafe" characters are present they must be replaced by their "entity" symbol to avoid this problem

Unsafe Character	Entity
&	&amp;
<	&lt;
>	&gt;
'	&apos;
"	&quot;

Setting 'xml_safe_*n*' to checked will automatically replace any unsafe characters by their entities

New Parameter 'xml_split_*n*'

This parameter is optional so if whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number does not contain XML or it does but you don't need to split it then this parameter can either be omitted or it can be included and its value set to blank.

If the parameter is present it must be set as follows:

Column	Value
Data type	"a"

Column	Value
Default value	• If the XML file whose name is in the 'copy_*n*' or 'move_*n*' parameter with the same sequence number contains a repeating element and you wish each occurrence of that repeating element to appear in a separate file (due to restrictions in the receiving system) then set 'Default value' to the name of the repeating element without the "<" or ">" characters or any of its attributes, e.g. "Suppliers/Supplier". The value of this parameter is case sensitive. • Otherwise omit this parameter or set 'Default value' to blank.
Length	Up to 200
#	The next available number.
Fixed	Checked

You can have an 'xml_split_*n*' parameter for each 'copy_*n*' or 'move_*n*' that is present and its number must match

If the files will take their name from the 'copy_*n*' or 'move_*n*' parameter but will have a sequence number starting at one, e.g. if the 'copy_*n*' or 'move_*n*' parameters value is "export.xml" then these files will be called "export_1.xml", "export_2.xml", "export_3.xml" and so on.

3.8 Run an Additional Report as a Process Ends (Extra)

3.8.1 Introduction

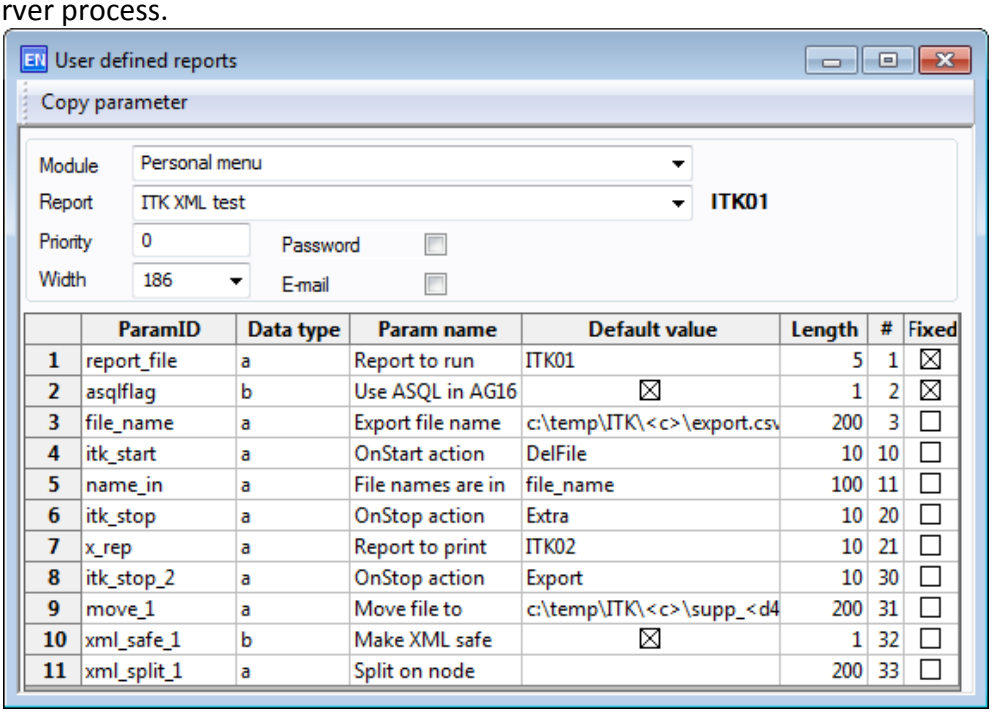
You may wish to add an additional report to a standard server process or user-defined report. This might be simply to produce additional printouts or it may be to add an export file to the process.

The ITKv3 implementation of this method is different to the way it worked in ITKv2:

1. In ITKv2 this method allowed you to run multiple reports; in ITKv3 you must define a separate 'itk_stop' parameter for each additional report that you want to run (see section 2.1 and below for more details).
2. In ITKv2 this method gave you the ability to copy or move the files out of the 'Report Results' folder as well as to run additional report files; in ITKv3 you must use the 'Extra' method to run the reports and the 'Export' method to copy or move the files, again by defining multiple 'itk_stop' parameters.

3.8.2 Set Up

In order to use this server exit on a process you must set up a variant with some additional parameters. The example is for a user defined report but the same principle applies to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	report_file	a	Report to run	ITK01	5	1	☒
2	asqlflag	b	Use ASQL in AG16	☒	1	2	☒
3	file_name	a	Export file name	c:\temp\ITK\<c>\export.csv	200	3	☐
4	itk_start	a	OnStart action	DelFile	10	10	☐
5	name_in	a	File names are in	file_name	100	11	☐
6	itk_stop	a	OnStop action	Extra	10	20	☐
7	x_rep	a	Report to print	ITK02	10	21	☐
8	itk_stop_2	a	OnStop action	Export	10	30	☐
9	move_1	a	Move file to	c:\temp\ITK\<c>\supp_<d4	200	31	☐
10	xml_safe_1	b	Make XML safe	☒	1	32	☐
11	xml_split_1	a	Split on node		200	33	☐

New Parameter 'itk_stop(_*n*)'

This parameter is mandatory and must be set as follows:

Column	Value
Data type	"a"
Default value	"Extra"
Length	30

Column	Value
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘x_rep(_*n*)’

This parameter defines the name of the report to be run:

Column	Value
Data type	“A”
Default value	The name of the report to run, excluding its file type (i.e. “ARW”, “CRW”, “RPT”)
Length	200
#	The next available number.
Fixed	Checked

New Parameter ‘x_test(_*n*)’

This optional parameter can be used to control whether or not the report named in the ‘x_rep’ parameter with the same number should be run or not.

Column	Value
Data type	“a”
Default value	The name of a process parameter that contains the name of the help-table used by the report
Length	100
#	The next available number.
Fixed	Checked

Some reports run from help tables created by the process that they are attached to, the names of these help tables are often stored in parameters created by the process which are then used in the body of the report; these parameters have names like “hlptab”, “helptable”, “helptab01”, etc. If the process fails to create the help table then it generally does not assign the table name(s) to the parameter(s) and when your report runs “`SELECT … FROM $hlptab t WHERE …`” (or whatever) it will become “`SELECT … FROM  t WHERE …`” and you will get a SQL error (in this case something along the lines of “table ‘t’ does not exist”).

If you put the name of the help table parameter into the ‘x_test’ parameter then if that parameter has a blank value the report in the corresponding ‘x_rep’ parameter will not be run. If you are using more than one help table in your report then you can put a comma separated list of the names of the parameter containing the help-table names into the ‘x_test’ parameter and the corresponding report will not be run if any of them have blank values.

In some circumstances the parameter containing the help table name may not be available to the report being called (the report will think its value is blank), using this parameter has the added benefit of overcoming this problem.

Summary of the use of multiple parameters

When multiple 'itk_stop' parameters are defined the names of the additional parameters must conform to the following:

'itk_stop' Parameter	'x_rep' Parameter	'x_test' Parameter
itk_stop	x_rep (or 'x_rep_1')	x_test (or 'x_test_1')
itk_stop_2	x_rep_2	x_test_2
itk_stop_3	x_rep_3	x_test_3
…and so on		

You can enter the 'copy_*n*', 'move_*n*' and 'x_rep_*n*' and 'x_test_*n*' parameters in any order in the variant.

The above screenshot shows a complex example and so deserves further explanation:

ParamID	Default value	Explanation
itk_start	DelFile	Remove an output file.
name_in	file_name	Replace the file tags in the 'file_name' parameter and delete the file if it exists. The process's query will use the modified value of the 'file_name' parameter.
itk_stop	Extra	Run an additional report.
x_rep	ITK02	Run the ITK02 report as well as the main ITK01 report.
		The absence of an 'x_test' parameter means that the ITK02 will run unconditionally.
itk_stop_2	Export	Export one or more report files.
move_1	c:\temp\…	Move the output of ITK02 (the second report) to the file path in 'Default value', replacing the file tags.
xml_safe_1	☒	Make the content of the exported file XML-safe.
xml_split_1		Don't split up the XML file.

3.9 Run a Query as a Process Starts (BiQuery)

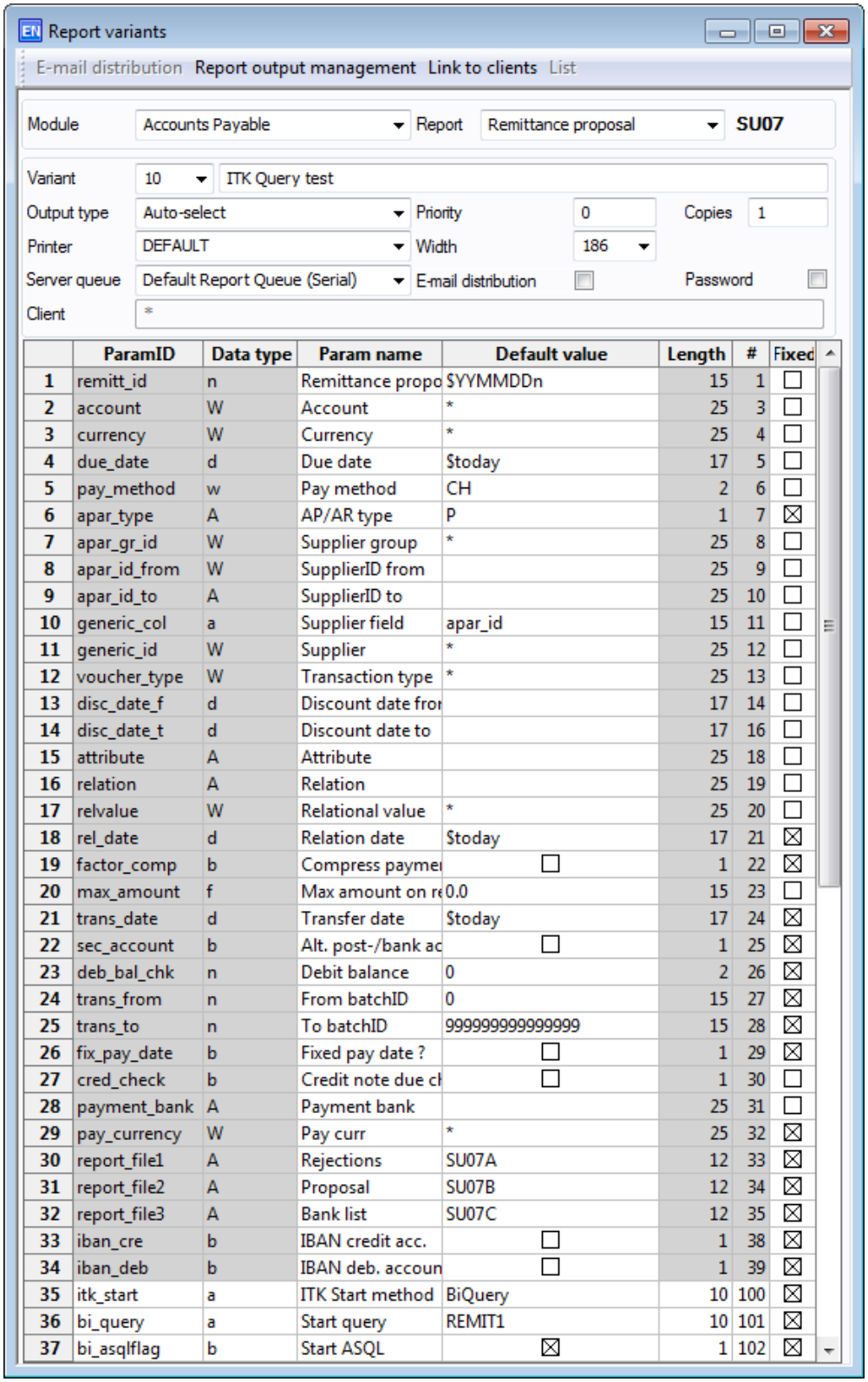
3.9.1 Introduction

Sometimes when configuring a system you need to be able to run a database update before a standard server process or report is run. Even if that process has a query parameter it may be that this query happens too late for your database update and also there is only one query.

The 'RunQuery' method that was available in ITKv2 has been removed.

3.9.2 Set Up

In order to use either of these server exits on a process you must set a variant with some new parameters.
The example is for the Remittance Proposal but the same principle applies to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	remitt_id	n	Remittance propo	$YYMMDDn	15	1	☐
2	account	W	Account	*	25	3	☐
3	currency	W	Currency	*	25	4	☐
4	due_date	d	Due date	$today	17	5	☐
5	pay_method	w	Pay method	CH	2	6	☐
6	apar_type	A	AP/AR type	P	1	7	☒
7	apar_gr_id	W	Supplier group	*	25	8	☐
8	apar_id_from	W	SupplierID from		25	9	☐
9	apar_id_to	A	SupplierID to		25	10	☐
10	generic_col	a	Supplier field	apar_id	15	11	☐
11	generic_id	W	Supplier	*	25	12	☐
12	voucher_type	W	Transaction type	*	25	13	☐
13	disc_date_f	d	Discount date fro		17	14	☐
14	disc_date_t	d	Discount date to		17	16	☐
15	attribute	A	Attribute		25	18	☐
16	relation	A	Relation		25	19	☐
17	relvalue	W	Relational value	*	25	20	☐
18	rel_date	d	Relation date	$today	17	21	☒
19	factor_comp	b	Compress payme	☐	1	22	☒
20	max_amount	f	Max amount on re	0.0	15	23	☐
21	trans_date	d	Transfer date	$today	17	24	☒
22	sec_account	b	Alt. post-/bank ac	☐	1	25	☒
23	deb_bal_chk	n	Debit balance	0	2	26	☒
24	trans_from	n	From batchID	0	15	27	☒
25	trans_to	n	To batchID	999999999999999	15	28	☒
26	fix_pay_date	b	Fixed pay date ?	☐	1	29	☒
27	cred_check	b	Credit note due ch	☐	1	30	☐
28	payment_bank	A	Payment bank		25	31	☐
29	pay_currency	W	Pay curr	*	25	32	☒
30	report_file1	A	Rejections	SU07A	12	33	☒
31	report_file2	A	Proposal	SU07B	12	34	☒
32	report_file3	A	Bank list	SU07C	12	35	☒
33	iban_cre	b	IBAN credit acc.	☐	1	38	☒
34	iban_deb	b	IBAN deb. accoun	☐	1	39	☒
35	itk_start	a	ITK Start method	BiQuery	10	100	☒
36	bi_query	a	Start query	REMIT1	10	101	☒
37	bi_asqlflag	b	Start ASQL	☒	1	102	☒

New Parameter ‘itk_start’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”

Column	Value
Default value	“BiQuery”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘bi_query’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	The name of the AG16 query that you want to be run.
Length	30
#	One more than that of the ‘itk_start’ parameter.
Fixed	Checked

New Parameter ‘bi_asqlflag’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“b”
Default value	• “0” if the AG16 query is written in native SQL • “1” if the AG16 query is written in ASQL
Length	1
#	The next available number.
Fixed	Checked

New Parameter ‘bi_name_in’

If you wish to use any of the ‘bi_asc_only’, ‘bi_xml_safe’ or ‘bi_xml_split’ parameters then you must set this parameter as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to manipulate (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	Next available number.
Fixed	Checked

New Parameter ‘bi_asc_only’

This parameter is optional so if all characters in the export file can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then it can either be omitted or can be included and its value set to “N”.

If the parameter is present it must be set as follows:

Column	Value
Data type	"A"
Default value	• If all the files in 'bi_name_in' could contain non-ASCII characters and you want them removed then set 'Default value' to "Y". • If all of the files in 'bi_name_in' can only contain ASCII characters or they could contain non-ASCII characters but you don't want them removed then set 'Default value' to "N". • If you want to remove non-ASCII characters from only some of the files in 'bi_name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, each value corresponds to the file in the same position in 'bi_name_in' and there must be the same number of values as there are in 'bi_name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'bi_blnk_lin'

This parameter is optional so if you do not wish to remove blank lines in the export file then this parameter can either be omitted or it can be included and its value set to checked.

If the parameter is present it must be set as follows:

Column	Value
Data type	"b"
Default value	• If all the files in 'bi_name_in' cannot contain blank lines or it could contain blank lines but you don't want them removed then set 'Default value' to "Y". • If all the files in 'bi_name_in' can contain blank lines and you want them removed then set 'Default value' to "N". • If you want to remove blank lines from only some of the files in 'bi_name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, each value corresponds to the file in the same position in 'bi_name_in' and there must be the same number of values as there are in 'bi_name_in'.
Length	1
#	The next available number.
Fixed	Checked

New Parameter 'bi_xml_safe'

Set 'Default value' as follows:

- If all the files in ‘bi_name_in’ contain XML and may contain “unsafe” characters then set ‘Default value’ to “Y”
- If none of the files in ‘bi_name_in’ contain XML or they do but you know that there are cannot be any unsafe characters then set ‘Default value’ to “N”. This is the default setting if the parameter is missing or left blank
- If some of the files in ‘bi_name_in’ may contain “unsafe” characters then set ‘Default value’ to a comma separated list of Ys and Ns, , each value corresponds to the file in the same position in ‘bi_name_in’ and there must be the same number of values as there are in ‘bi_name_in’

‘Data’ to ‘A’
‘Length’ to 30
‘#’ to one more than the ‘#’ value of the ‘bi_name_in’ parameter or the ‘bi_asc_only’ parameter if it is present
and ‘Fixed’ to checked

If any the following characters are contained in the payload (or value) of an XML element then this will cause programs to fail to read that XML. If any these “unsafe” characters are present they must be replaced by their “entity” symbol to avoid this problem

Unsafe Character	Entity
&	&amp;
<	&lt;
>	&gt;
'	&apos;
"	&quot;

Setting ‘xml_safe’ to “Y” will automatically replace any unsafe characters by their entities

New Parameter ‘bi_xml_split’

Set ‘Default value’ as follows:

- If all the files in ‘bi_name_in’ contain XML and contain the same repeating element and you wish each occurrence of that repeating element to appear in a separate file (due to restrictions in the receiving system) then set ‘Default value’ to the name of the repeating element without the “<” or “>” characters or any of its attributes, e.g. “Supplier”. The value of this parameter is case sensitive
- If none of the files in ‘bi_name_in’ contain XML or they do but you don’t want to split the repeating elements into separate files then omit this parameter or set ‘Default value’ to blank.
- If some of the files in ‘bi_name_in’ contain XML and you wish to split different repeating elements into different files or not split them at all then set ‘Default value’ to a comma separated list of element names, each value corresponds to the file in the same position in ‘bi_name_in’ and there must be the same number of values as there are in ‘bi_name_in’. A blank element in the list should be represented by two consecutive apostrophes thus “''”

‘Data’ to ‘a’
‘Length’ to 30

‘#’ to one more than the ‘#’ value of the corresponding ‘xml_safe_*n*’ parameter
and ‘Fixed’ to checked

If more than one file is created then each file will have a sequence number starting at one.

Example

If the parameter pointed to by ‘bi_name_in’ contains “C:\temp\BwOut\st_<c><o>.xml” and it is run in client “EN” and its order number is 85 and one file is created in the “C:\temp\BwOut” folder then its name will be “st_EN85.xml”. If three files are created then their names will be “st_EN85_1.xml”, “st_EN85_2.xml” and “st_EN85_3.xml”.

3.10 Run a Query as a Process Ends (BeQuery)

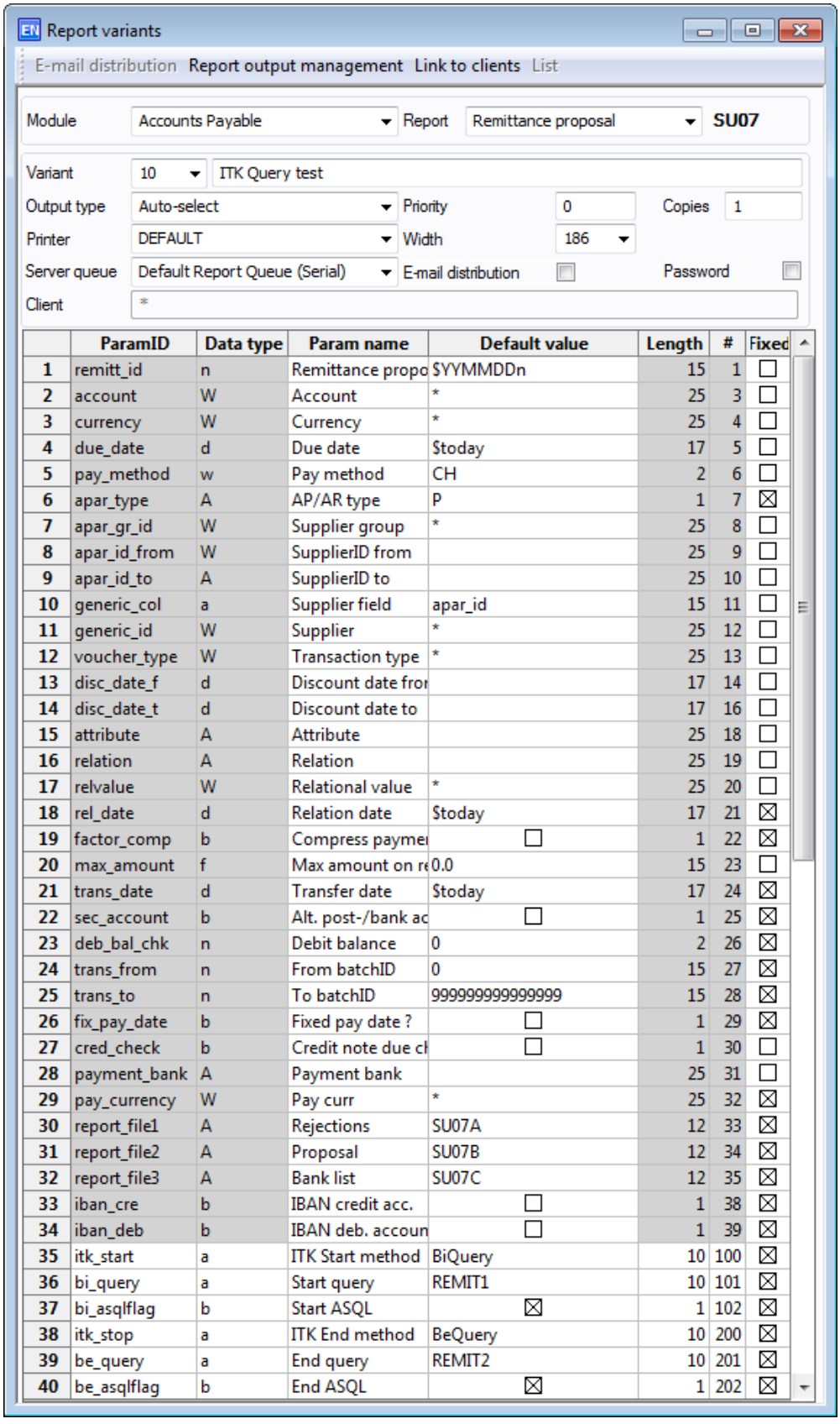
3.10.1 Introduction

Sometimes when configuring a system you need to be able to run a database update at the end of a standard server process or report. Even if that process has a query parameter it may be that this query happens too early for your database update and also there is only one query.

The 'RunQuery' method that was available in ITKv2 has been removed.

3.10.2 Set Up

In order to use either of these server exits on a process you must set a variant with some new parameters.
The example is for the Remittance Proposal but the same principle applies to any other server process.



	ParamID	Data type	Param name	Default value	Length	#	Fixed
1	remitt_id	n	Remittance propo	$YYMMDDn	15	1	☐
2	account	W	Account	*	25	3	☐
3	currency	W	Currency	*	25	4	☐
4	due_date	d	Due date	$today	17	5	☐
5	pay_method	w	Pay method	CH	2	6	☐
6	apar_type	A	AP/AR type	P	1	7	☒
7	apar_gr_id	W	Supplier group	*	25	8	☐
8	apar_id_from	W	SupplierID from		25	9	☐
9	apar_id_to	A	SupplierID to		25	10	☐
10	generic_col	a	Supplier field	apar_id	15	11	☐
11	generic_id	W	Supplier	*	25	12	☐
12	voucher_type	W	Transaction type	*	25	13	☐
13	disc_date_f	d	Discount date fro		17	14	☐
14	disc_date_t	d	Discount date to		17	16	☐
15	attribute	A	Attribute		25	18	☐
16	relation	A	Relation		25	19	☐
17	relvalue	W	Relational value	*	25	20	☐
18	rel_date	d	Relation date	$today	17	21	☒
19	factor_comp	b	Compress payme	☐	1	22	☒
20	max_amount	f	Max amount on re	0.0	15	23	☐
21	trans_date	d	Transfer date	$today	17	24	☒
22	sec_account	b	Alt. post-/bank ac	☐	1	25	☒
23	deb_bal_chk	n	Debit balance	0	2	26	☒
24	trans_from	n	From batchID	0	15	27	☒
25	trans_to	n	To batchID	999999999999999	15	28	☒
26	fix_pay_date	b	Fixed pay date ?	☐	1	29	☒
27	cred_check	b	Credit note due ch	☐	1	30	☐
28	payment_bank	A	Payment bank		25	31	☐
29	pay_currency	W	Pay curr	*	25	32	☒
30	report_file1	A	Rejections	SU07A	12	33	☒
31	report_file2	A	Proposal	SU07B	12	34	☒
32	report_file3	A	Bank list	SU07C	12	35	☒
33	iban_cre	b	IBAN credit acc.	☐	1	38	☒
34	iban_deb	b	IBAN deb. accoun	☐	1	39	☒
35	itk_start	a	ITK Start method	BiQuery	10	100	☒
36	bi_query	a	Start query	REMIT1	10	101	☒
37	bi_asqlflag	b	Start ASQL	☒	1	102	☒
38	itk_stop	a	ITK End method	BeQuery	10	200	☒
39	be_query	a	End query	REMIT2	10	201	☒
40	be_asqlflag	b	End ASQL	☒	1	202	☒

New Parameter 'itk_stop'

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	“BiQuery”
Length	30
#	On a report variant this must be greater than or equal to “100”, on a user defined report this can be the next available number.
Fixed	Checked

New Parameter ‘be_query’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“a”
Default value	The name of the AG16 query that you want to be run.
Length	30
#	One more than that of the ‘itk_start’ parameter.
Fixed	Checked

New Parameter ‘be_asqlflag’

This parameter is mandatory and must be set as follows:

Column	Value
Data type	“b”
Default value	• “0” if the AG16 query is written in native SQL • “1” if the AG16 query is written in ASQL
Length	1
#	The next available number.
Fixed	Checked

New Parameter ‘be_name_in’

If you wish to use any of the ‘bi_asc_only’, ‘bi_xml_safe’ or ‘bi_xml_split’ parameters then you must set this parameter as follows:

Column	Value
Data type	“a”
Default value	A comma separated list of the names of the parameters containing the file names to delete (“query_param1” in the example). Spaces on either side of each comma are ignored.
Length	100
#	Next available number.
Fixed	Checked

New Parameter ‘be_asc_only’

This parameter is optional so if all characters in the export file can only be in the ASCII character set (see section 5.2) or there could be non-ASCII characters and you do not want them removed then it can either be omitted or can be included and its value set to “N”.

If the parameter is present it must be set as follows:

Column	Value
Data type	"A"
Default value	• If all the files in 'be_name_in' could contain non-ASCII characters and you want them removed then set 'Default value' to "Y". • If all of the files in 'be_name_in' can only contain ASCII characters or they could contain non-ASCII characters but you don't want them removed then set 'Default value' to "N". • If you want to remove non-ASCII characters from only some of the files in 'be_name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, each value corresponds to the file in the same position in 'be_name_in' and there must be the same number of values as there are in 'be_name_in'.
Length	100
#	The next available number.
Fixed	Checked

New Parameter 'be_blnk_lin'

This parameter is optional so if you do not wish to remove blank lines in the export file then this parameter can either be omitted or it can be included and its value set to checked.

If the parameter is present it must be set as follows:

Column	Value
Data type	"b"
Default value	• If all the files in 'bei_name_in' cannot contain blank lines or it could contain blank lines but you don't want them removed then set 'Default value' to "Y". • If all the files in 'be_name_in' can contain blank lines and you want them removed then set 'Default value' to "N". • If you want to remove blank lines from only some of the files in 'bei_name_in' then set 'Default value' to a comma separated list of "Y"s and "N"s, each value corresponds to the file in the same position in 'be_name_in' and there must be the same number of values as there are in 'be_name_in'.
Length	1
#	The next available number.
Fixed	Checked

New Parameter 'be_xml_safe'

Set 'Default value' as follows:

- If all the files in 'be_name_in' contain XML and may contain "unsafe" characters then set 'Default value' to "Y"
- If none of the files in 'be_name_in' contain XML or they do but you know that there are cannot be any unsafe characters then set 'Default value' to "N". This is the default setting if the parameter is missing or left blank
- If some of the files in 'be_name_in' may contain "unsafe" characters then set 'Default value' to a comma separated list of Ys and Ns, , each value corresponds to the file in the same position in 'be_name_in' and there must be the same number of values as there are in 'be_name_in'

'Data' to 'A'

'Length' to 30

'#' to one more than the '#' value of the 'be_name_in' parameter or the 'be_asc_only' parameter if it is present

and 'Fixed' to checked

If any the following characters are contained in the payload (or value) of an XML element then this will cause programs to fail to read that XML. If any these "unsafe" characters are present they must be replaced by their "entity" symbol to avoid this problem

Unsafe Character	Entity
&	&amp;
<	<
>	>
'	'
"	"

Setting 'xml_safe' to "Y" will automatically replace any unsafe characters by their entities

New Parameter 'be_xml_split'

Set 'Default value' as follows:

- If all the files in 'be_name_in' contain XML and contain the same repeating element and you wish each occurrence of that repeating element to appear in a separate file (due to restrictions in the receiving system) then set 'Default value' to the name of the repeating element without the "<" or ">" characters or any of its attributes, e.g. "Supplier". The value of this parameter is case sensitive
- If none of the files in 'be_name_in' contain XML or they do but you don't want to split the repeating elements into separate files then omit this parameter or set 'Default value' to blank.
- If some of the files in 'be_name_in' contain XML and you wish to split different repeating elements into different files or not split them at all then set 'Default value' to a comma separated list of element names, each value corresponds to the file in the same position in 'be_name_in' and there must be the same number of values as there are in 'be_name_in'. A blank element in the list should be represented by two consecutive apostrophes thus "'"

'Data' to 'a'
'Length' to 30
'#' to one more than the '#' value of the corresponding 'xml_safe_*n*' parameter
and 'Fixed' to checked

If more than one file is created then each file will have a sequence number starting at one.

Example

If the parameter pointed to by 'be_name_in' contains "C:\temp\BwOut\st_<c><o>.xml" and it is run in client "EN" and its order number is 85 and one file is created in the "C:\temp\BwOut" folder then its name will be "st_EN85.xml". If three files are created then their names will be "st_EN85_1.xml", "st_EN85_2.xml" and "st_EN85_3.xml".

New Parameter 'be_xparam_n'

Some server processes set additional process parameters during their run, examples of this include:

- GL07's "trig_stat" parameter.
- DataLoad's "err_table" parameter

If your AG16 query needs to use the value of one of these internally set parameters then you must specify its name in the value of up to five parameters (be_xparam_1, be_xparam_2, be_xparam_3, be_xparam_4 and be_xparam_5).

Due to technical restrictions if one of the parameters is the name of a database table that you wish to access in your AG16 query then the value of your "be_asqlflag" parameter must not be one

Additional Parameters

These additional parameters can have any 'ParamID' that you wish that does not clash with any other parameters on the server process they are used to transfer values into the AG16 query that you do not want to hardcode or you wish the user to enter. These parameters can be referenced in the query in the usual way. In the screen shot one additional parameter is present 'bi_qryparam1'.

3.11 Run a Command-Line Tool as a Process Starts (BiRun)

WARNING

Method ‘BiRun’ is not implemented in the current version of ITK.

3.12 Run a Command-Line Tool as a Process Ends (BeRun)

WARNING

Method ‘BeRun’ is not implemented in the current version of ITK.

3.13 Access Process Parameters on 5.5 (LoadBi)

WARNING

Method ‘LoadBi’ is not implemented in the current version of ITK.

4 Further Information

4.1 Converting a Report into a Server Process

Due to a technical restriction ITK methods can only be used on server processes, therefore if you wish to use a method on a simple report that report must be converted into a server process. This can be done by turning it into an "AG16" type process whose query doesn't do anything. This can be done by making the following changes:

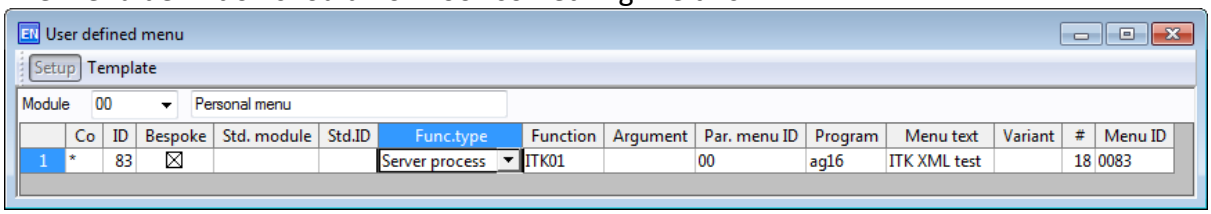
- Modify the report's menu entry,
- Create a new 'SQL query',
- Add additional parameters to its 'User defined report' definition.

4.1.1 The Menu Entry

Use the **Settings ► System administration ► Menus ► User defined menu** screen (AG27) to change the following columns:

- 'Func type' from "Report" to "Server process",
- 'Program' from blank to "ag16".

The menu definition should now look something like this:

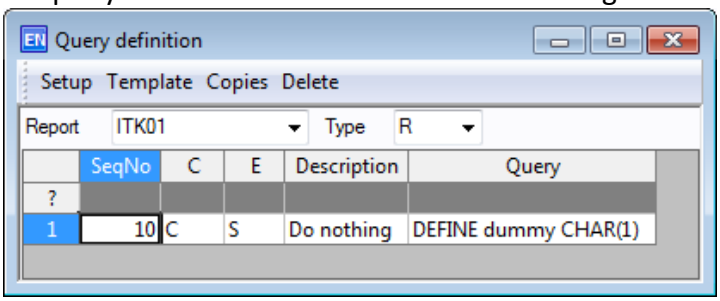


4.1.2 The SQL Query

Use the **Settings ► System administration ► Reports ► SQL queries ► Query definition** screen (AG14) to create a new SQL query with the same name as your report:

- Its 'Type' must be "R" (for "Report"),
- 'SeqNo' should be "10",
- 'C' should be "C",
- 'E' should be "S",
- 'Description' should be "Do nothing",
- 'Query' should be "DEFINE dummy CHAR(1)".

The query definition should now look something like this:



4.1.3 The Additional Parameters

Use the **Settings ► System administration ► Reports ► User defined report** screen (AG35) to add two new parameters to the report:

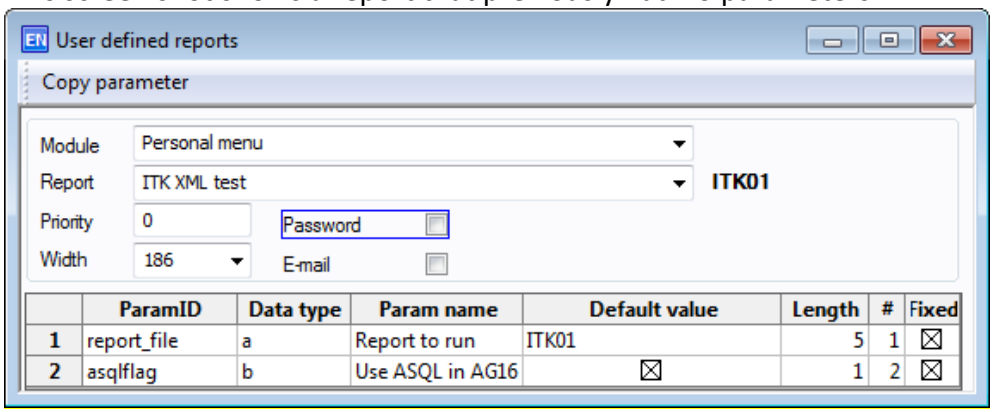
- Parameter one:

Column Name	Value
ParamID	“report_file”
Data type	“a”
Param name	“Report to run” *(or similar)*
Default value	*The name of your report*
Length	*The number of characters in ‘Default value’ (or more)*
#	*The next available number*
Fixed	☒

- Parameter two

Column Name	Value
ParamID	“asqflag"
Data type	“b”
Param name	“Use ASQL in AG16” *(or similar)*
Default value	☒
Length	“1”
#	*The next available number*
Fixed	☒

This screen shot shows a report that previously had no parameters:



Now check that your report still works and if it does you can now add to your user defined report definition the ITK method(s) that you want to use.

4.2 The Query for the Import File Conversions

This query is used in the examples used in 'CSV Import File Conversion (ConvCsv)', 'CFR Import File Conversion (ConvCfr)' and 'FSR Import File Conversion (ConvFsr)' to turn each of the sample files into a GL07 batch.

SeqNo	C	E	Description
10	C	S	Header date
DEFINE file_date CHAR(8)			
20	C	S	Footer total
DEFINE file_dr FLOAT			
30	C	S	Balance amount
DEFINE calc_dr FLOAT			
40	C	S	Error flag
DEFINE error_test IDENT(6)			
50	C	S	Create help table
CREATE TABLE $*tab1 AS SELECT SPACE(25) AS account, description, dim_1, dim_2, dim_3, dim_4, dim_5, dim_6, dim_7, SPACE(15) AS cur_amount, SPACE(5) AS rec_type, sequence_no FROM acrbatchinput WHERE 1 = 2			
60	N	S	Get data
COPY IN IMPORT FILE = '$query_param1', COLSEP = T, TABLE=$*tab1, rec_type, account, dim_1, dim_2, dim_3, dim_4, dim_5, dim_6, dim_7, cur_amount, description			
70	N	S	Remove existing batch data
DELETE FROM acrbatchinput WHERE batch_id = '$batch_id' AND client = '$client' AND interface = '$interface'			
80	N	S	Get date
SELECT CONCAT(SUBSTR(account, 7, 4), CONCAT(SUBSTR(account, 4, 2), SUBSTR(account, 1, 2))) INTO :file_date FROM $*tab1 WHERE rec_type = 'H1'			
90	N	S	Get total dr
SELECT TO_MONEY(account) INTO :file_dr FROM $*tab1 WHERE rec_type = 'F1'			
100	N	S	Remove headers and footers
DELETE FROM $*tab1 WHERE LEFT(rec_type, 1) IN ('H', 'F')			
110	N	S	Get total of debits
SELECT SUM(TO_MONEY(cur_amount)) INTO :calc_dr FROM $*tab1 WHERE TO_MONEY(cur_amount) > 0			
120	N	S	Set column name if footer + balance = zero
SELECT 'col_0' INTO :error_test FROM asysdummy WHERE col_0 = '0' AND :file_dr = :calc_dr			
130	N	S	Generate a syntax error if out of balance
SELECT :error_test INTO :error_test FROM asysdummy WHERE col_0 = '0'			
140	N	S	Sequence no
SEQUENCE ON $*tab1 (sequence_no, 0)			

SeqNo	C	E	Description
150	N	S	Create batch
INSERT INTO acrbatchinput (batch_id, interface, voucher_type, trans_type, client, account, dim_1, dim_2, dim_3, dim_4, dim_5, dim_6, dim_7, tax_code, tax_system, currency, dc_flag, cur_amount, amount, number_1, value_1, value_2, value_3, description, trans_date, sequence_no, voucher_date, voucher_no, period) SELECT '$batch_id', '$interface', 'GL', 'GL', '$client', account, dim_1, dim_2, dim_3, dim_4, dim_5, dim_6, dim_7, '0', ' ', 'GBP', 0, TO_MONEY(cur_amount), TO_MONEY(cur_amount), 0, 0, 0, 0, description, ISO2DATE(:file_date), sequence_no, ISO2DATE(:file_date), 1, $period FROM $*tab1			

4.3 File Name Tags

Some ITK methods allow the names of files being handled to be modified at run time, you do this by entering one or more of the following tags in the 'Default value' of the parameter containing a file name.

Tag	Meaning	Description	Example
<c>	Client	The code of the BW client in which the report was run	EN
<d>	Date	The date on which the report was run ("ddmmyy" format)	141016
<dr>	Date reverse	The date on which the report was run ("yymmdd" format)	161014
<d4>	Date 4 digit year	The date on which the report was run ("ddmmyyyy" format)	14102016
<d4r>	Date 4 digit year reverse	The date on which the report was run ("yyyymmdd" format)	21061014
<dd>	Date day	The day on which the report was run	14
<dmn>	Date month number	The month in which the report was run (numeric format)	10
<dma>	Date month abbrev.	The month in which the report was run (abbreviated format)	Oct
<dmf>	Date month full	The month in which the report was run (full format)	October
<dy2>	Date year 2 digits	The year in which the report was run (short format)	16
<dy4>	Date year 4 digits	The year in which the report was run (full format)	2016
<e>	Export	The path of the BW 'Data Export' folder	C:\Agresso 5.7.1\Data Export
<f>	File name	The default report output name	cu04en_100.lis
<l>	"Lis"	The path of the BW 'Report Results' folder	C:\Agresso 5.7.1\Report Results
<o>	Order number	The report's order number	100

Tag	Meaning	Description	Example
<p|*id*>	Process parameter value	Where *id* is the 'ParamID' of another process parameter of this job, the *id* must **EXACTLY** match that parameter's ID and must not be the name of the current parameter. E.g. <p|copy_1> is not allowed to appear in the "copy_1" parameter	if the process has a parameter 'abc_level' that is set to "BC" then <p|abc_level> would be replaced by "BC"
<r>	Report name	This is the column headed 'Report' in the maintenance of report printouts screen	CU04
<s|*id*>	System parameter value	Where *id* is the name of a client, system or common parameter	<s|MAX_DATE> will be replaced by "31-DEC-2099"
<t>	Time	The time at which the report was run ("hhmmss" format)	130523
<th>	Time hours	The hour in which the report was run	13
<tm>	Time minutes	The minute in which the report was run	05
<ts>	Time seconds	The second in which the report was run	23

Nested tags are not allowed, for instance "<s|<r>_FOLDER>" is illegal and will give unexpected results

Examples

Example One

If the entry in a file name parameter's 'Default value' is "C:\temp\BwOut\st_<c><o>.txt" then if the process is run in client "EN" and its order number is 85 the file will be called "st_EN85.txt" in the "C:\temp\BwOut" folder.

Example 2

If the entry in a file name parameter's 'Default value' is "C:\BwOut\<c>\REMIT<dy4>-<dmn>-<dd>" then if the process is run in client "W1" on 03/08/2005 then the file will be called "REMIT2005-08-03" in the "C:\BwOut\W1" folder.

Example 3

If the following set up is run on process "CU04"

ParamID	Default value
copy_0	<p|folder><p|file>

ParamID	Default value
file	BW-<r>-<o>.XML
folder	<s|XML_OUT_FOLDER>

The if the process is run with order number 1234 in a client where a Common, System or Client parameter “XML_OUT_FOLDER” has the value of “E:\BwOut\” then a file called “BW-CU04-1234.XML” will be created in folder “E:\BwOut”

This last admittedly rather circuitous technique is useful if the report is intended to be run by an IntellAgent event because on some BW versions parameter values must be kept short to avoid IntellAgent stopping with an error (the maximum allowed length varies with BW version).

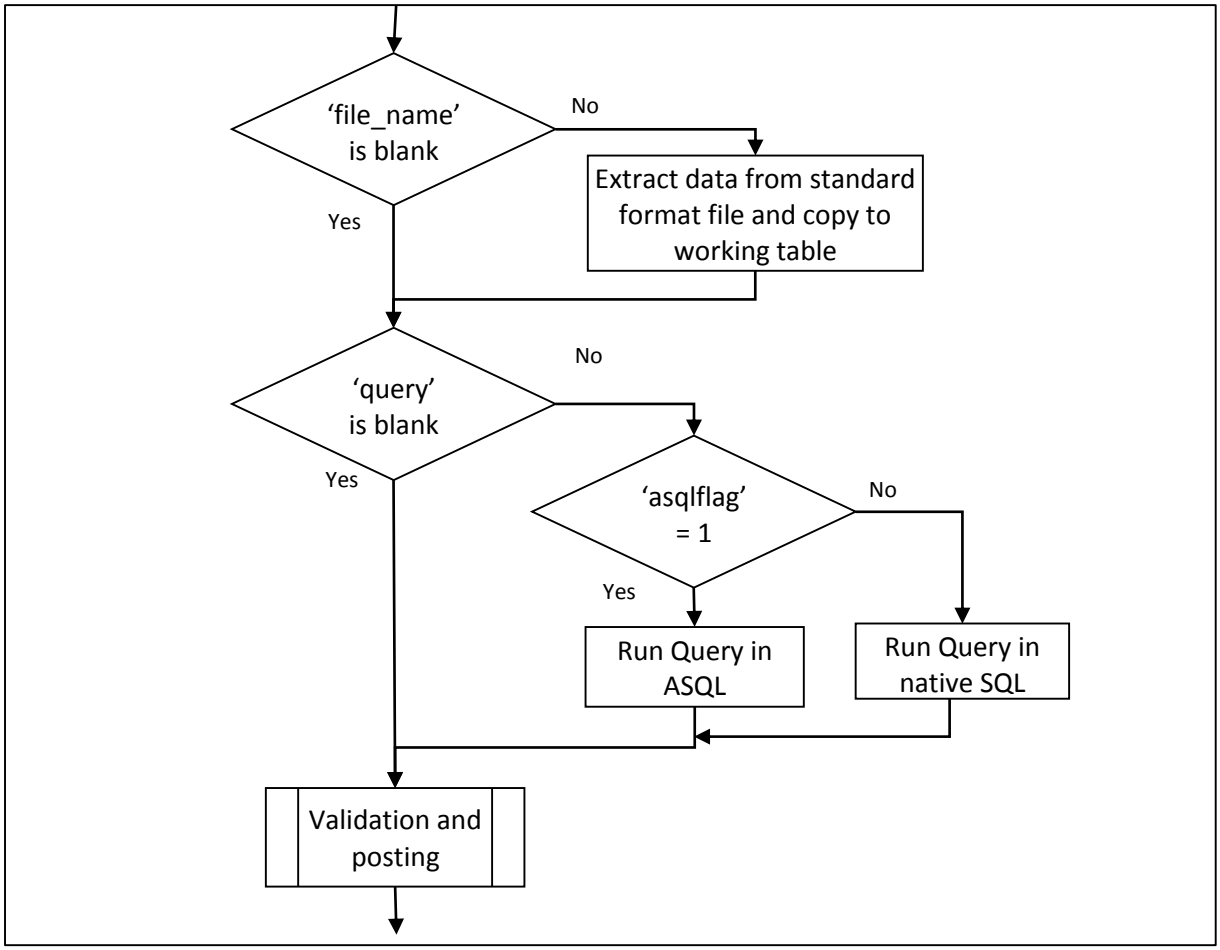
5 Background Information

5.1 SQL Queries in BW Data Import Routines

Many BW data import routines have a process parameter called 'query'. Setting the 'query' parameter instructs the import routine to run a block of SQL written using the **Settings ► System Administration ► Reports ► SQL Queries ► Query definition** screen (AG14). This query is run after the contents of the input file have been copied to the import routine's working table and any validation and posting of this data is carried out. In its simplest form a query can manipulate the data copied from the file so that it can then be correctly processed by the import routine; this manipulation can include, but is not limited to:

- Completing missing data,
- Mapping external codes to BW ones,
- Changing the level of detail of the imported data.

A more advanced use of a query is to allow the import routine to process data contained in a file that does not conform to the standard layout for that import routine. This works because if you leave the process parameter that points to the input file (usually called 'file_name') blank then the import routine will not look for a file and try to copy its contents to its working table, so the first thing that the import routine does is run the nominated query in either ASQL or native SQL syntax.



The ASQL language has a “COPY IN” command that allows you to read the contents of a file into a table. COPY IN can read files consisting of a series of lines (or records) and each record can contain multiple fields. The fields in a record are either

- “fixed width” e.g. characters 1 to 4 are field one, 5 to 7 are field two, etc. on every record
- “delimited” i.e. each field is separated from the next one by a particular character. This character can be either a:
  - Tab (ASCII[10] code 9)
  - “@” (at sign)
  - “£” (sterling sign)
  - “$” (dollar sign)
  - “%” (percentage symbol)
  - “&” (ampersand)
  - “.” (full stop)
  - or “;” (semicolon)

  Using the Tab character is the safest option as this is a non-printing character and is therefore highly unlikely to appear in the data.

The database’s native SQL will have a similar facility to “COPY IN” but whilst this may be less restrictive in the file formats it can read it is not as convenient to use: it may for instance have to be invoked from within a stored procedure

Some import routines do not have a standard file layout at all and rely on you supplying a query to read the file that you wish to import. Examples of this include:

- ‘Import of direct deb. Agreement’ (CU08)
- ‘Import bank statement’ (CB05)
- ‘Batch input of employees/resources’ (PR43)

Queries are usually named after the process that invokes them, hence a query run in ‘Batch input transactions from external system’ (GL07) will be called a “GL07 query”, similarly “CU08 queries”, “LG04 queries”, etc. Queries can also be written that appear as menu items in their own right and these queries are run by a server process called AG16, these queries are “AG16 queries”, because of this all queries are sometimes referred to as AG16 queries whether they are run by AG16, GL07 or some other process.

[10] See section 5.2

5.2 The ASCII Character Set

5.2.1 Overview

The **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (ASCII) character set is a character encoding standard which is used to represent text in computers, telecommunications equipment, and other devices. Most modern character-encoding schemes are based on ASCII, although they support many additional characters.

The first edition of the standard was published in 1963; it underwent a major revision during 1967, and experienced its most recent update during 1986.

The ASCII character set contains:

- Those characters that appear on an American keyboard: numbers, plus upper and lower case English letters, plus a selection of symbols and punctuation marks.
- 33 control codes: these are mostly obsolete printer and modem control characters, but include Tab ($^{H}_{T}$), Line-feed ($^{L}_{F}$), Form-feed ($^{F}_{F}$) and Carriage-return ($^{C}_{R}$).

Being a seven-bit encoding system it only contains 128 characters with codes ranging from 0 to 127, therefore if a character has an encoding higher than 127 it is a non-ASCII character. Globally there are thousands of non-ASCII characters, these include:

- Letters from non-English alphabets,
- Non-English punctuation marks,
- Mathematical symbols,
- "Dingbats" (also known as printer's ornaments),
- Box drawing characters.

WARNING

- Do not confuse the ASCII character set with the "ANSI" character set, which is an eight bit encoding system commonly used by Microsoft® Windows®, this system is sometimes inaccurately (and ambiguously) called "eight bit ASCII" and the characters that it contains depends on which region or "code page" your copy of Windows is using.
- Being an eight bit encoding system it contains a total of 256 characters; the first 128 characters are the same as the ASCII codes but the remaining 128 characters contain mainly region specific codes.
- In America and Western Europe the ANSI character set is more correctly (and unambiguously) called "Windows-1252".

5.2.2 List of the ASCII Character Set

Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character
0	$^{N}_{UL}$	1	$^{S}_{OH}$	2	$^{S}_{TX}$	3	$^{E}_{TX}$	4	$^{E}_{OT}$	5	$^{E}_{NQ}$	6	$^{A}_{CK}$	7	$^{B}_{EL}$
8	$^{B}_{S}$	9	$^{H}_{T}$	10	$^{L}_{F}$	11	$^{V}_{T}$	12	$^{F}_{F}$	13	$^{C}_{R}$	14	$^{S}_{O}$	15	$^{S}_{I}$
16	$^{D}_{LE}$	17	$^{D}_{C1}$	18	$^{D}_{C2}$	19	$^{D}_{C3}$	20	$^{D}_{C4}$	21	$^{N}_{AK}$	22	$^{S}_{YN}$	23	$^{E}_{TB}$
24	$^{C}_{AN}$	25	$^{E}_{M}$	26	$^{S}_{UB}$	27	$^{E}_{SC}$	28	$^{F}_{S}$	29	$^{G}_{S}$	30	$^{R}_{S}$	31	$^{U}_{S}$

Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character	Encoding	Character
32		33	!	34	"	35	#	36	$	37	%	38	&	39	'
40	(	41	)	42	*	43	+	44	,	45	-	46	.	47	/
48	0	49	1	50	2	51	3	52	4	53	5	54	6	55	7
56	8	57	9	58	:	59	;	60	<	61	=	62	>	63	?
64	@	65	A	66	B	67	C	68	D	69	E	70	F	71	G
72	H	73	I	74	J	75	K	76	L	77	M	78	N	79	O
80	P	81	Q	82	R	83	S	84	T	85	U	86	V	87	W
88	X	89	Y	90	Z	91	[	92	\	93	]	94	^	95	_
96	`	97	a	98	b	99	c	100	d	101	e	102	f	103	g
104	h	105	i	106	j	107	k	108	l	109	m	110	n	111	o
112	p	113	q	114	r	115	s	116	t	117	u	118	v	119	w
120	x	121	y	122	z	123	{	124	|	125	}	126	~	127	$^{D}_{EL}$

In particular note that the only currency symbol that is an ASCII character is “$”, so for example:

- The “£” symbol is non-ASCII,
- The “€” symbol is non-ASCII, not least because this currency was still more than ten years in the future at the time when the ASCII character set was last updated.