Integrations (Direct)

Jira — Last update: 31 October 2022

Basis Technologies

Table of Contents

1. Introduction

This Integration Guide is intended to give the reader an overview of the capabilities of the ActiveControl Integration Engine and the out of the box (OOTB) integration scenarios supported. It also contains detailed technical specifications of the currently available communication techniques and a detailed configuration guide.

1.1. Document Audience

The intended audience for this document are the technical teams looking to implement integration between ActiveControl and third party tools, such as ticketing systems. It does not detail how AC can be configured to manage the change process and it assumes a reasonable knowledge of standard change processes with SAP.

1.2. Integration with ActiveControl

ActiveControl offers a variety of ways to integrate inbound and outbound scenarios using documented API's. ActiveControl provides an Integration Framework that can manage outbound interactions with external systems (including queuing, re-sends, error processing and reporting) and inbound integration scenarios – those initiated by a system external to AC – by exposing several fully documented API's and web-services that allow manipulation of AC objects by these systems.
In addition, as AC is a NetWeaver certified product, all standard SAP integration techniques are available, including tRFC and IDoc communication. But for the purposes of this document, it is assumed that web services will be the preferred integration method and these are therefore described in detail in this document.

1.3. Background

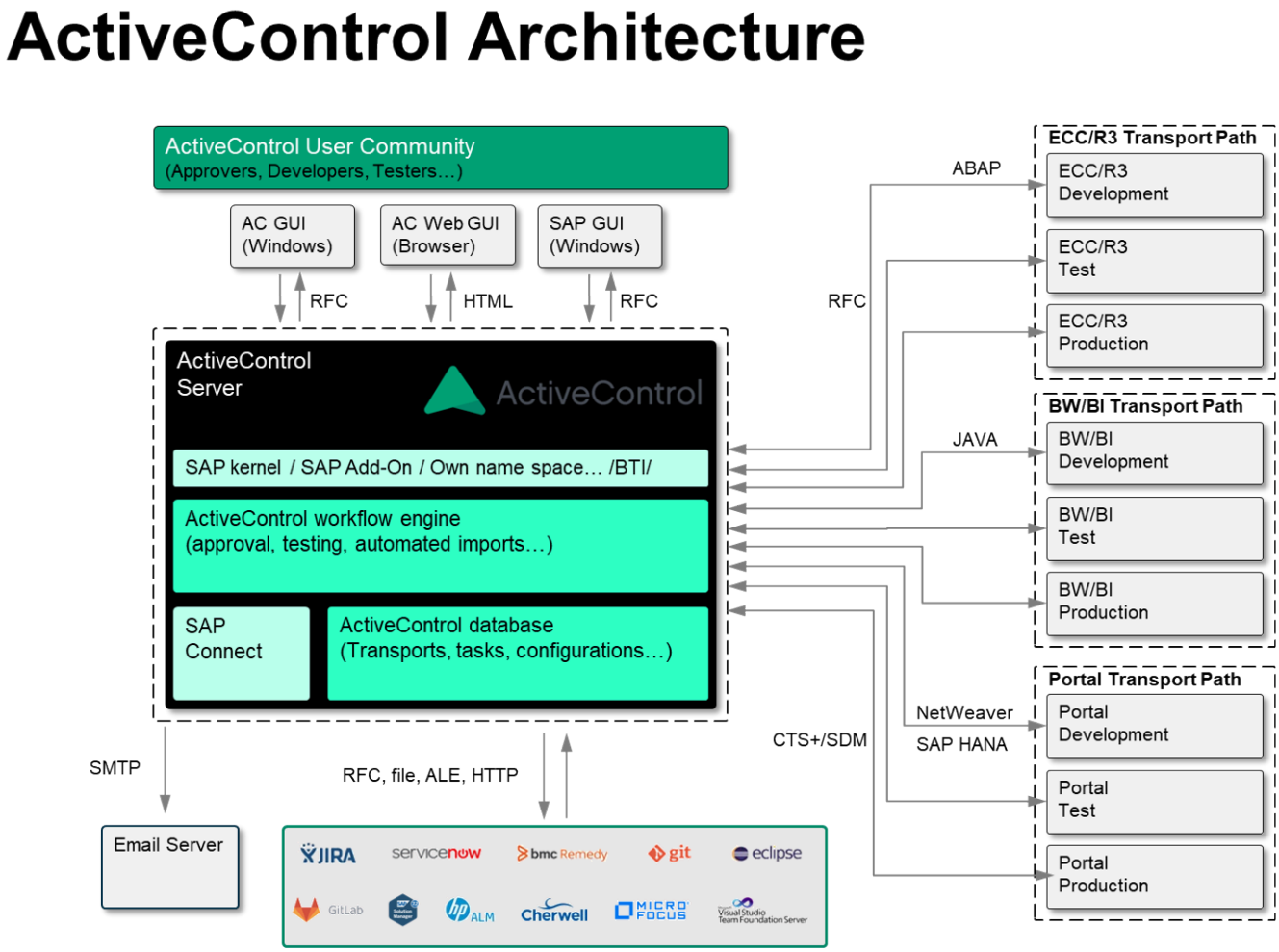
ActiveControl includes an Integration Framework that can be used to deliver data integration between ActiveControl and other 3rd Party Tools.

The Integration Framework can manage outbound interactions with external systems (including queuing, re-sends, error processing and reporting) and inbound integration scenarios – those initiated by a system external to AC – by exposing several fully documented API's that allow manipulation of AC objects by these systems.

Some of the general ActiveControl Integration Framework configuration options are not used and/or required as part of the JIRA integration; these are detailed in this Admin Guide where relevant.

2. Integration Architecture

The architecture of ActiveControl can be broken down into: client software, a controlling SAP system, other participating SAP systems and integration systems. The diagram below illustrates the central role of the controlling SAP system – referred to as the ActiveControl “domain controller”.

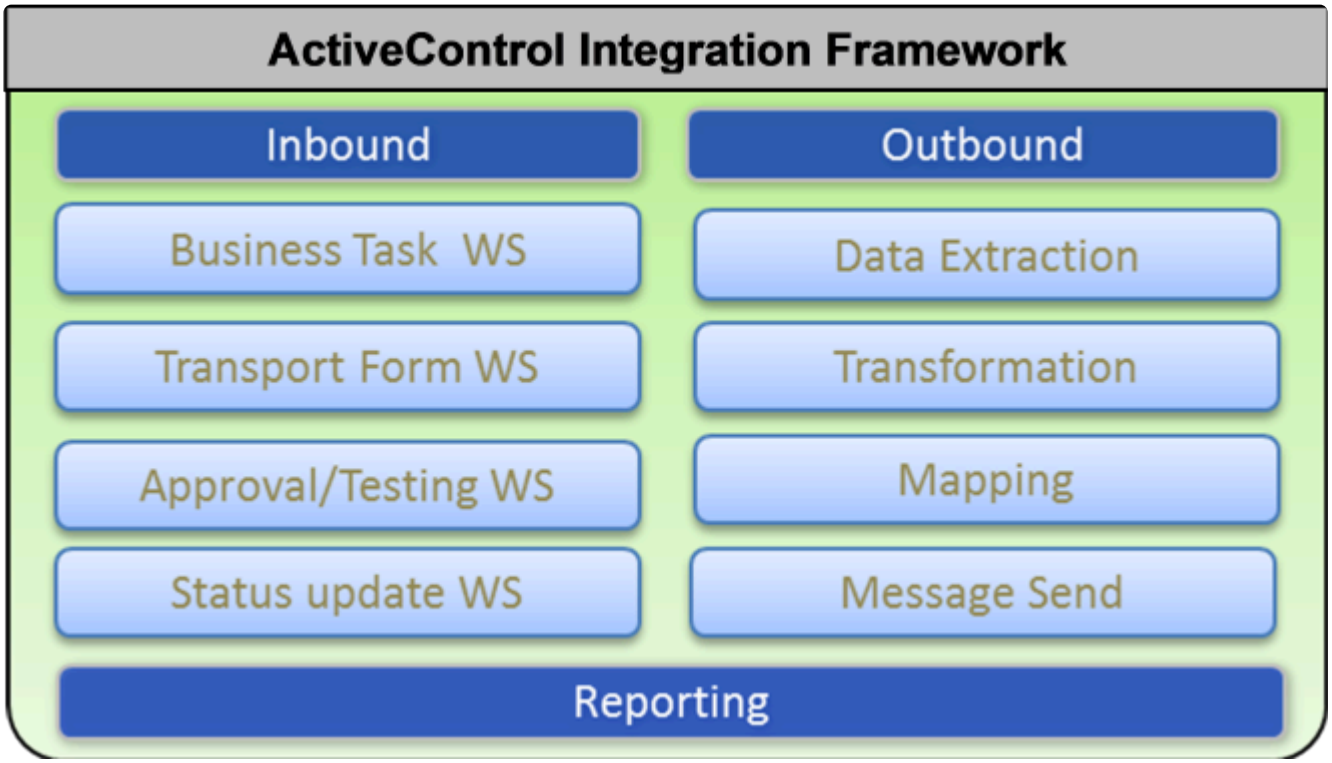


2.1. The Domain Controller

Like the Transport Management System, ActiveControl has the concept of a “domain controller”. The domain controller does not need to be configured in any special way, it is simply the SAP system that the ActiveControl client software connects to and is where ActiveControl configuration and application data is stored.
The server software runs mostly within the ActiveControl domain controller. When necessary, the domain controller connects to the other SAP systems to gather change request information and to perform transports. These connections are made using SAP’s remote function call (RFC) protocol.
The Integration Engine is part of the Domain Controller and manages communication with external products and systems.

2.2. The Integration Framework Architecture

The Integration Framework is divided between inbound and outbound processes. For inbound calls, those made by a third party system into AC, a number of web services are exposed allowing the external system to manipulate ActiveControl objects. Calls to AC web services will return appropriate error messages, but expect the calling system to deal with queuing, service levels and retries for failed integration transactions. For outbound calls there is a configurable framework that includes data extraction, transformation, mapping and sending routines, alongside error detection, correction and reporting, as can be seen in the table below.



ActiveControl provides Data Extraction and Message Send components for some standard scenarios and third party tools, but these can be enhanced by the addition of custom extraction and send routines plugged into the standard framework. So if, for example, you use an in-house ITSM solution, a new send component can be developed and plugged into the integration framework to facilitate communication between it and ActiveControl. All other standard framework functions, such as data extraction, mapping and error correction remain unchanged and can be used as-is.

2.3. Integration Scenarios

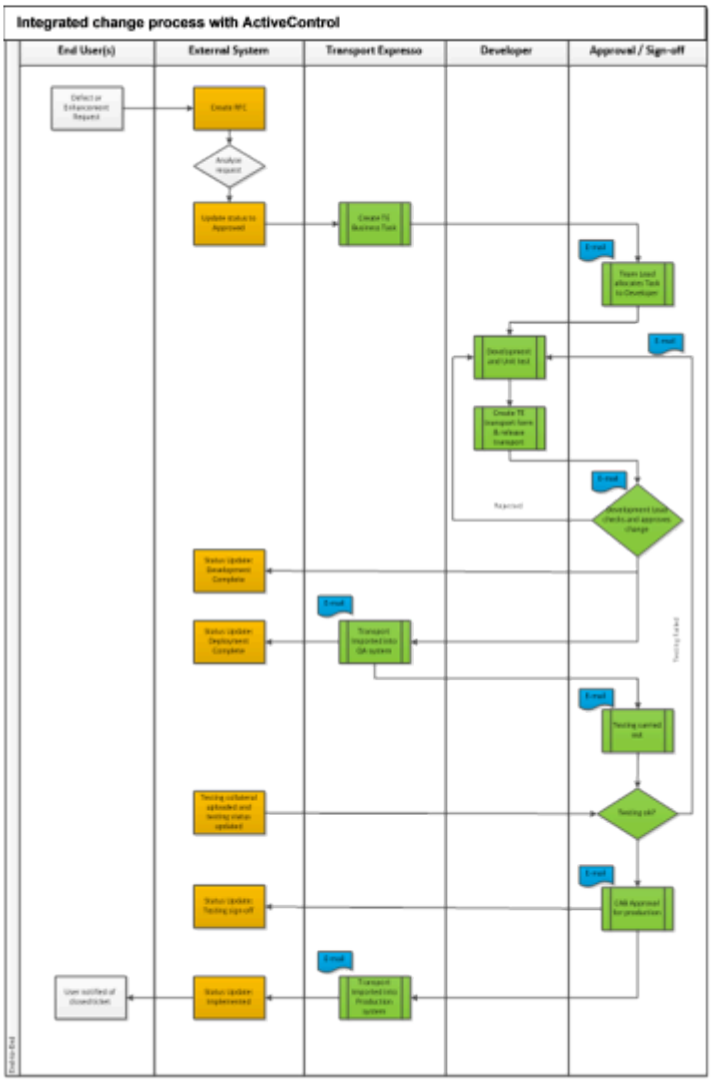
The standard integration scenario is to combine AC and a third party IT Service Management product, and possibly a Test Automation product to create a fully integrated end-to-end process for managing change. This typically requires both inbound and outbound integration:

1. Change created in third party ITSM system
2. Change approved for development in ITSM system
3. Change interfaced to AC (inbound integration)
4. Change managed through AC for deployment to Test and Pre-Prod with updates sent to ITSM system to reflect progress (outbound integration)
5. Change deployed to production through AC and ITSM system updated (outbound integration)
6. Change verified and closed in ITSM system

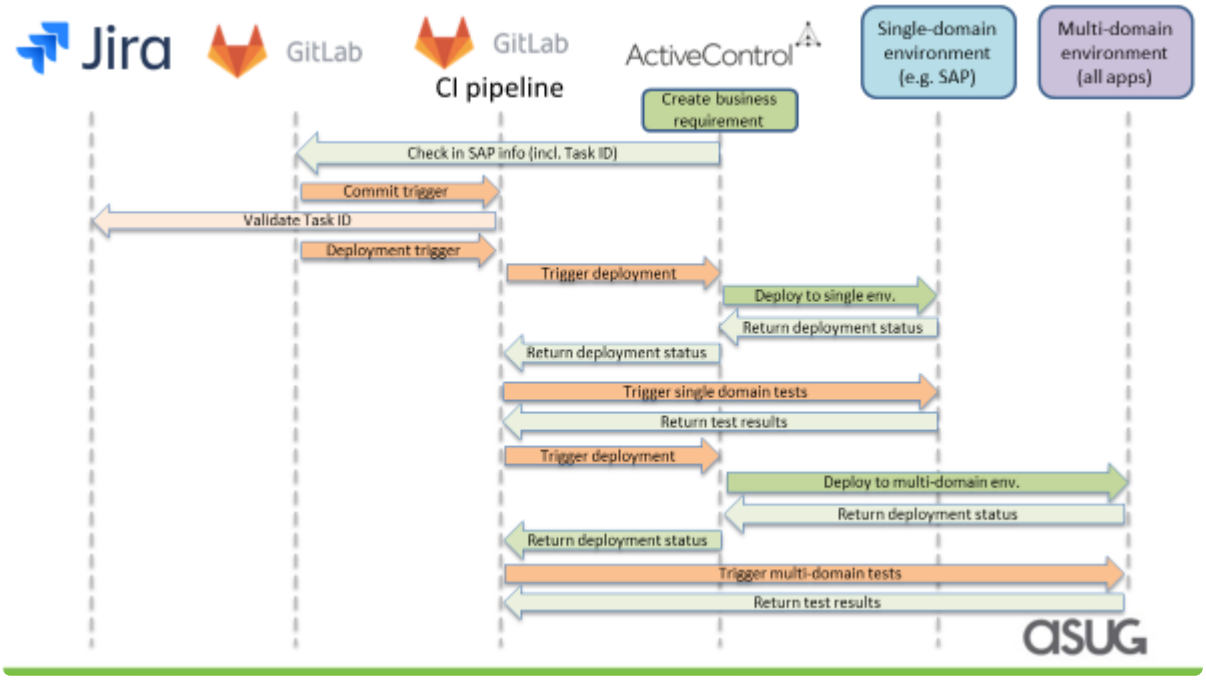
This document will describe, in detail, exactly how such integration can be easily accomplished using the ActiveControl Integration framework, but it should be noted that the framework can be extended for use in many other integration scenarios.

2.4. Integration Process Flow

The ActiveControl Integration Framework provides an open architecture for passing messages into and out of the system in a multitude of ways. Although integration can be set up in many ways, one of the more common scenarios is explained in detail below:



DevOps CI/CD inclusion:



In this scenario we have bi-directional integration between an external ticketing system and ActiveControl. This gives a direct link between the ticketing system and the underlying technical changes that make up the business change. So, whether looked at from the perspective of the ticketing system, or through AC, there is only one version of the 'truth' for all changes across the landscape.
From a more detailed perspective, we can look at the integration scenarios:
1. Once a proposed enhancement or defect resolution is approved and a system change is deemed necessary, the external system creates a Business Task in AC representing the change. The ticket in the ITSM system and the AC Business Task are then tied together for the remainder of the process
2. The creation of the Business task in AC marks the start of the development process. The Task can be allocated to a developer who then performs the development and/configuration, and completes unit testing.
3. Once the developer has finished their work, they release the technical change (the transport) and the development team lead is notified by ActiveControl and approves the change. AC will automatically run a number of configured analysis checks at this point to ensure the change is ok to move on in the process.
4. The change is imported into the Quality Assurance system (maybe after another approval from the Testing manager) and is now ready for testing.
5. AC updates the status of the ticket in the ITSM system to show that it is now in testing or ready to be tested.
6. Test collateral and results can be added to either the ticketing system or AC and the ITSM system automatically updated.
7. CAB approval is sought and AC analysis is completed in real time to report dependencies between changes and the impact of different approval scenarios.
8. Once approved by CAB the status of the change in the ticketing system is updated and the change is imported into the Production system at the appropriate time
9. The ticketing system is updated to show the change has been implemented.

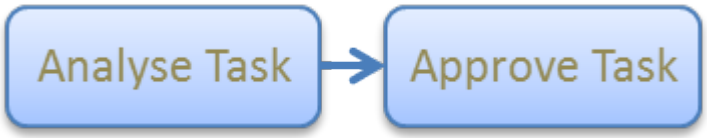
2.5. Inbound Integration

For inbound integration scenarios AC provides several SOAP web services. Currently, these are:
• Create a Business Task
• Change a Business Task
• Read a Business Task
• Analyse a Business Task
• Read the results of an analysis for a Business Task
• Approve a Business Task
• Enter Test Results for a Business Task
Each web service is detailed in the following sections.

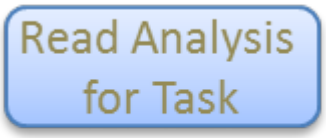
2.6. Inbound Process Flow

The standard inbound integration process flows would be:
• Create/Change a Business Task in ActiveControl
Creating or changing a Business Task requires simple calls to the appropriate web service. When changing a Task, the current field values should be read first to ensure changed data is not overwritten. The process flow should therefore be:

• Approve a Business Task
When approving a Business Task it is important that the Task Analysis is completed to first to ensure that the approval can take place safely. The approval web service will not stop the approval if analysis results are ignored. The process flow for approving a Business Task in AC should therefore be:



The analysis results for a Task can be retrieved for any specific Target/Location by calling the Analysis Read web service.



• Enter test results for a Business Task. When entering test results for a Business Task, it must be decided if this result is simply being saved or saved and approved. Only by using the save and approve will the change move to the following control point in the Path.

2.7. Connector Functionality

From the standard process flow above, it is envisaged that the developed connector will provide the services required on the ITSM side to initiate a web service call to ActiveControl to create a AC Business Task (which will be the representation of the ITSM ticket within AC).

The following functions need to be available within the connector:
1. Initiating integration: The exact conditions required for the integration to be initiated will vary from client to client. This means that a flexible, configurable way needs to be developed to initiate integration. A set of conditions need to be able to be created, including the value(s) of any field on the incident/ change ticket and its status, which when met, initiates the integration to ActiveControl.
2. Default values: When initiating integration to ActiveControl, we need to be able to specify default values for the mandatory fields on the ActiveControl Task. These are:
a. Project
b. Group
c. Type
3. Mapping: The fields on the ITSM ticket need to be able to be mapped to fields in ActiveControl. Any fields on the ITSM ticket, including any customer defined fields, need to be able to be mapped to any field in the ActiveControl Create Task WS, including custom fields.
4. Processing: Once the ITSM ticket meets one of the conditions to fire off integration and the fields have been mapped to the ActiveControl Web Service, the connector should be able to call the AC web service. A system username and password will need to be passed to enable authentication for some clients, other clients may use SSO. Both authentication methods should be available. ActiveControl will either return an error or the internal number of the Task that has been created. The connector needs to be able to update a field on the ITSM ticket with the AC Task number or to store the error message.
5. Error processing: The connector should be able to be configured to try the integration more than once and to store any error messages that are returned. After a configured number of retries, an email (or other notification) needs to be sent to a configurable list of users, informing them of the error and the ITSM ticket involved. An administrator must be able to manually re-send the integration record if the maximum number of retries has been exceeded.

It should be noted that the connector will not be able to cater for all possible scenarios that may be required by a customer. It is really just a starter which may be extended by the client themselves.

2.8. Inbound Integration

There are two inbound calls in the above scenario:
1. Creation of the Business Task in AC
2. Approval of Testing/Entry of test results once testing complete

Both of these calls would be web service calls to standard AC APIs (although alternative techniques are available and are described later in this document). The calling system (i.e. the ticketing system) would be responsible for queuing of messages and ensuring errors were dealt with appropriately. Some mapping may be required depending on the data passed from the ticketing system to AC for classification of the change.

2.9. Outbound Integration

The outbound calls from AC to the external ticketing system can all be based on the Deployment Status of a change within AC. Integration scenarios based on AC status changes are delivered as standard with the AC Integration Engine and therefore require no development.

The steps to set up this type of status based integration are:
1. Complete base AC Integration engine configuration. This includes identifying the end points of the integration and any mapping requirements. The mapping engine can be configured for most standard scenarios, but if complex mapping is required, ActiveControl user exits can be implemented to enhance the standard mapping routines. For more details on AC user exits and how they are implemented, please refer to the ActiveControl Administration Guide.

2. A trigger program should be scheduled to pick up the Task status changes that need to be interfaced to the external system(s). This trigger program selects the appropriate AC records, dependent on the configuration set up above, and passes it through the mapping engine. It then stores the mapped integration transactions into a set of standard tables. See Outbound Configuration section below.
Program Name: /BTI/TE_INTEG_TRIGGER

3. A send program is then scheduled to pick up the mapped transactions and send them out to the configured external systems. It retrieves the required records and then uses the configured send methods for each particular integration scenario to actually push the data out to the receiving systems. If a standard send method is not available for a particular external system (maybe the ticketing system is a ‘home-grown’ application), then custom send methods can be created and utilised in the Integration Framework. See Outbound Configuration section below.
Program Name: /BTI/TE_INTEG_SEND

4. The outcome of the send process is recorded for audit purposes. If successful, any updates configured are made to the AC data objects, alternatively if errors have occurred, the send program will try to re-send (if configured to do so) a certain number of times before marking the transaction in error and sending a notification to the relevant person(s) within the organisation.

5. At any time, the Integration Reporting Console can be used to see the status of all integrations, the status and history of each transaction and can also be used to update the underlying transactional data, if required, to fix errors. See Outbound Configuration section below.
Program Name: /BTI/TE_RINTEG_AUDIT

3. ActiveControl Domain Controller Setup and Configuration

This section guides you through the steps that are needed to configure inbound and outbound integration within ActiveControl.
The Integration configuration is maintained through the SAP standard SM30/31 transactions where table entries can be created and updated.

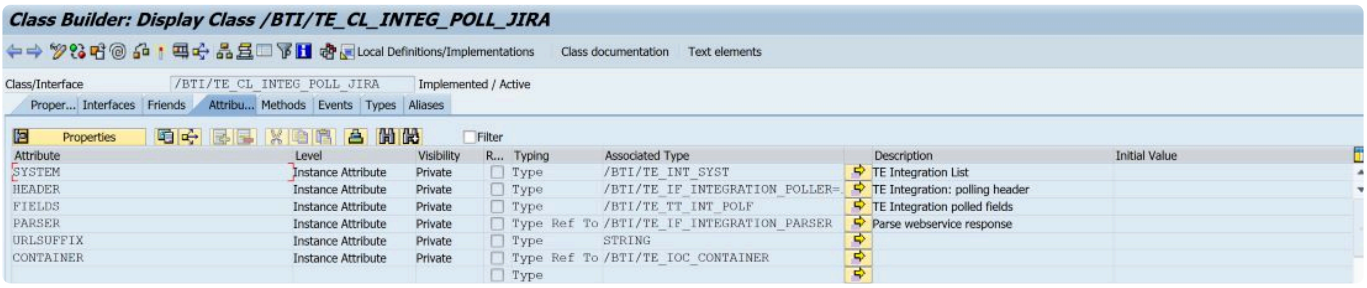
3.1. Bi-directional Integration Process

3.1.1. Class Builder

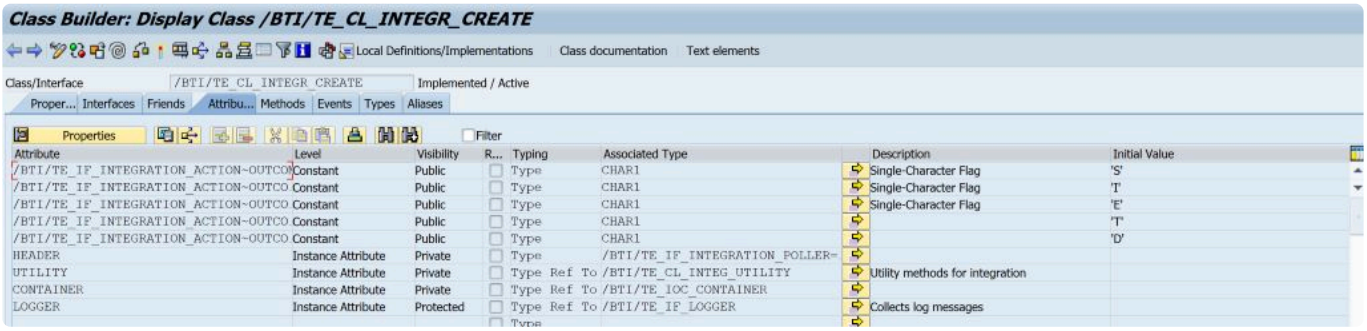
Create/Activate Integration Classes
SE24

CLASS	Notes
/BTI/TE_CL_INTEG_POLL_JIRA	Poll JSON Web service
/BTI/TE_CL_INTEGR_CREATE	Create Business Task action
/BTI/TE_CL_INTEGR_TESTRES	Enter test results action
/BTI/TE_CL_INTEGR_UPDATE	Update Business Task action (post-creation)

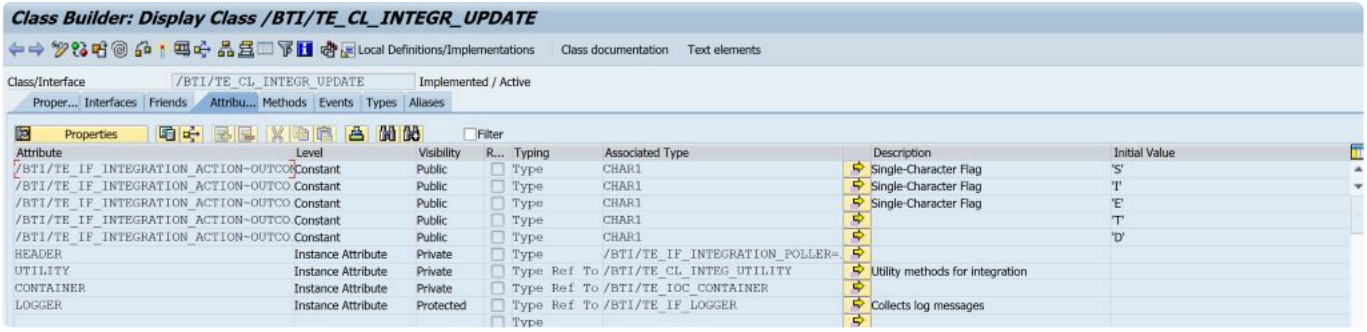
/BTI/TE_CL_INTEG_POLL_JIRA



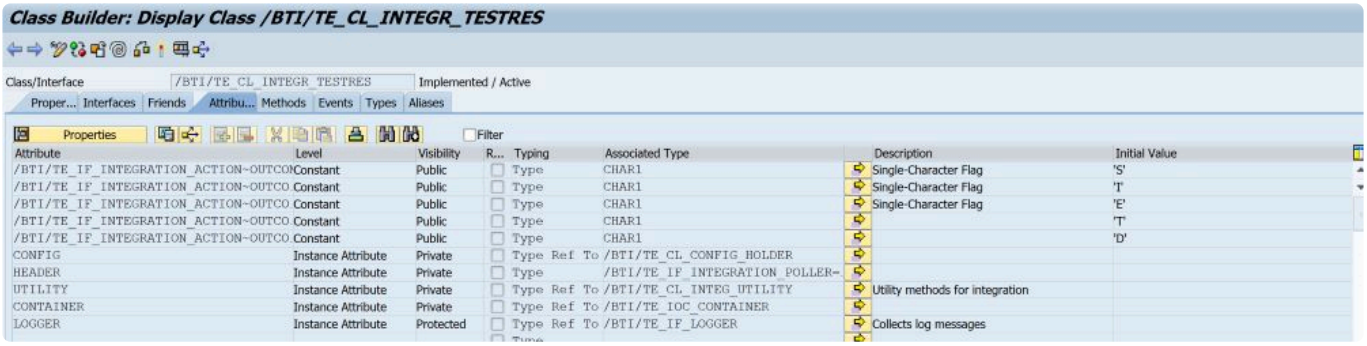
/BTI/TE_CL_INTEGR_CREATE



/BTI/TE_CL_INTEGR_UPDATE



/BTI/TE_CL_INTEGR_TESTRES



Attribute	Level	Visibility	R...	Typing	Associated Type	Description	Initial Value
/BTI/TE_IF_INTEGRATION_ACTION~OUTCON	Constant	Public		Type	CHAR1	Single-Character Flag	'S'
/BTI/TE_IF_INTEGRATION_ACTION~OUTCO	Constant	Public		Type	CHAR1	Single-Character Flag	'I'
/BTI/TE_IF_INTEGRATION_ACTION~OUTCO	Constant	Public		Type	CHAR1	Single-Character Flag	'E'
/BTI/TE_IF_INTEGRATION_ACTION~OUTCO	Constant	Public		Type	CHAR1		'T'
/BTI/TE_IF_INTEGRATION_ACTION~OUTCO	Constant	Public		Type	CHAR1		'D'
CONFIG	Instance Attribute	Private		Type Ref To	/BTI/TE_CL_CONFIG_HOLDER		
HEADER	Instance Attribute	Private		Type	/BTI/TE_IF_INTEGRATION_POLLER~		
UTILITY	Instance Attribute	Private		Type Ref To	/BTI/TE_CL_INTEG_UTILITY	Utility methods for integration	
CONTAINER	Instance Attribute	Private		Type Ref To	/BTI/TE_IOC_CONTAINER		
LOGGER	Instance Attribute	Protected		Type Ref To	/BTI/TE_IF_LOGGER	Collects log messages	

3.1.2. External System(s)

The ActiveControl integration framework can be used to perform outbound integration on potentially any external system. Two tables need to be maintained here, table '/BTI/TE_INT_SYST' is the table that holds all the external system id's and descriptions along with any RFC Destinations that may possibly be needed for example for a Solution Manager system, also table '/BTI/TE_INT_CLAS' needs to be maintained and this holds the class that the framework references.

SM30

/BTI/TE_INT_SYST – AC Integration System List	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXTSYS_ID	Single word identifier for external system. E.g. Jira
EXTSYS_NAME	Full description of external system
RFC_DEST	Some external systems that you want systems with could possibly be SAP systems for example Solution Manager so the RFC destination is held here.
DDC_INT	
TEXT_ID	Task field link
TEXT_ID	Form field link
INT_USER	Integration user from 3rd part tool
INT_PASSWORD	Integration user password from 3rd part tool
INT_PORT	Some external systems that you want systems with could possibly be SAP systems for example Solution Manager, so the RFC destination is held here.

Change View "TE Integration List": Overview

New Entries

TE Integration List

Ext.Sys.No	Ext Sys Id	Description	DDC Ident
1	REMEDY	Transport Expresso ACX	
2	HP-QC	Transport Expresso ACX	
3	HP-SM	Transport Expresso ACX	
4	MS-TFS	Transport Expresso ACX	
5	FOOTPRINTS	Transport Expresso ACX	
6	HP-ALM	Transport Expresso ACX	
7	RATIONAL		
8	SERVICENOW	TRANSPORT EXPRESS D01	
9	SERVICENOW	TRANSPORT EXPRESSO T01	
10	JIRA	ACX ActiveControl Integration System	
11	CA-SERVONE	Transport Expresso ACX	
11	JIRA	BTI_JIRA	
12	TESTIMONY	Testimony	
13	GITLAB	GitLab	

3.1.3. RFC Destination

For API calls to 3rd party tools, AC requires setup of HTTP RFC Destination.

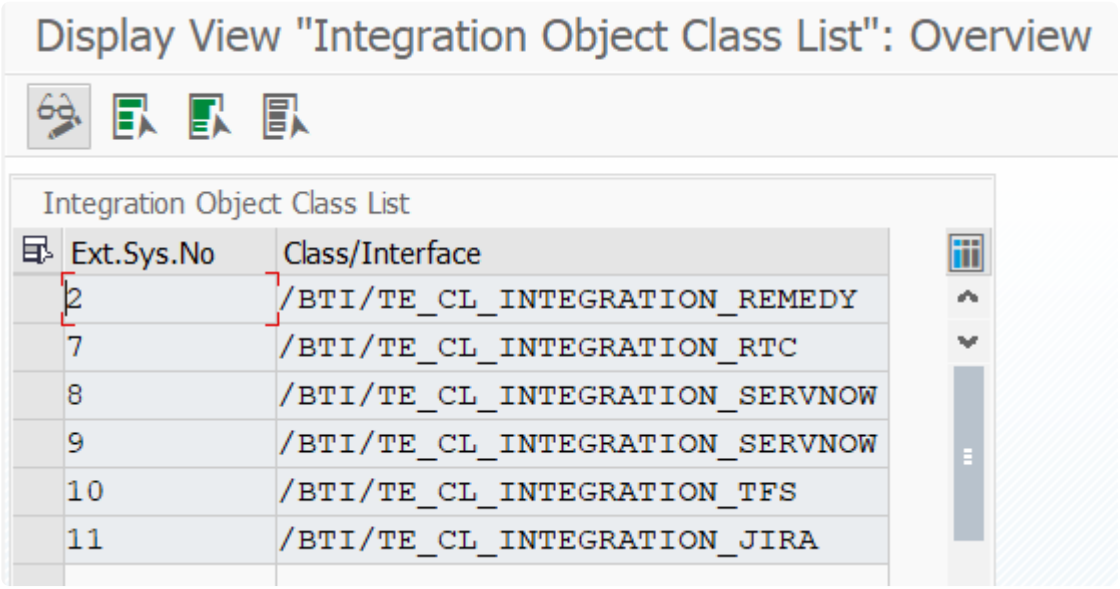
Field	Description
RFC Dest	Name of service and system
Connection Type	G = HTTP External Service
Description	Description
Technical Settings	
Target Host	url for service
Service No.	Port No.
Path Prefix	if proxy is used
Logon & Security	
Security Options	SSL Certificate – Active

Although a connection test is successful, AC Polling Engine will go into error if a system user is required for access to Jira and this integration access to the report filters.
See error handling section on RFC HTTP connection test issues.

3.1.4. Class List

The Class is defined from Class Builder section above.
SM30

/BTI/ TE_INT_CLAS – Integration Object Class List	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
CLASSNAME	Held here is the class name where the bulk of the integration processing is done. AC integration works on the principle of having a class for each external system that we need to integrate with. This is what is called in the integration send program. E.g. /BTI/ TE_CL_INTEGRATION_SOLMAN.



3.1.5. Update Processes

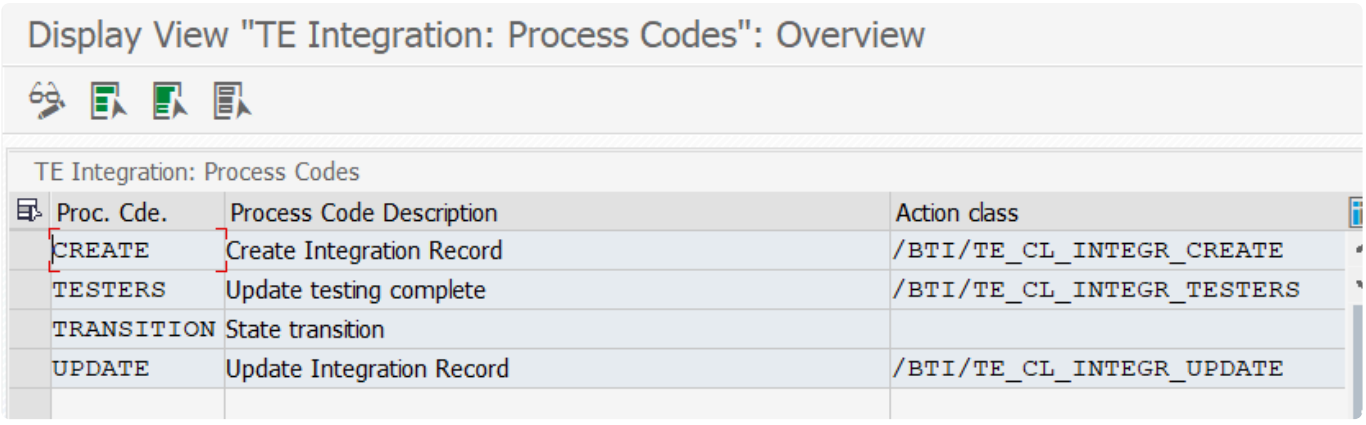
Currently the integration framework is capable of updating external records in two ways in either ‘Create’ mode or ‘Update’ mode, these ‘modes’ are known within the integration framework as process codes and to try and ensure forwards compatibility these have been made configurable but would obviously require code changes if any other process codes were to become available. These process codes are held in table ‘/BTI/TE_INT_PC’. These two process codes would need to be shipped for standard functionality.

SM30

/BTI/TE_INT_PC – Process Codes	
Field	**Description**
PROCESS_CODE	The process codes used by the integration framework to perform some kind of action. The framework gets shipped with two standard process codes CREATE and UPDATE.
CODE_DESCRIPTION	Description of above code.
ACTION_CLASS	The class specified is only used by the polling process, and must implement /BTI/TE_IF_INTEGRATION_ACTION

This existed already, but has a new field INBOUNDACTIONCLASS.
The class specified is only used by the polling process, and must implement /BTI/TE_IF_INTEGRATION_ACTION

Record	**Action Class**
CREATE – Create integration record	/BTI/TE_CL_INTEGR_CREATE
TESTERS – Testing complete	/BTI/TE_CL_INTEGR_TESTERS
UPDATE – Update integration record	/BTI/TE_CL_INTEGR_UPDATE
TRANSITION > State transition	N/A



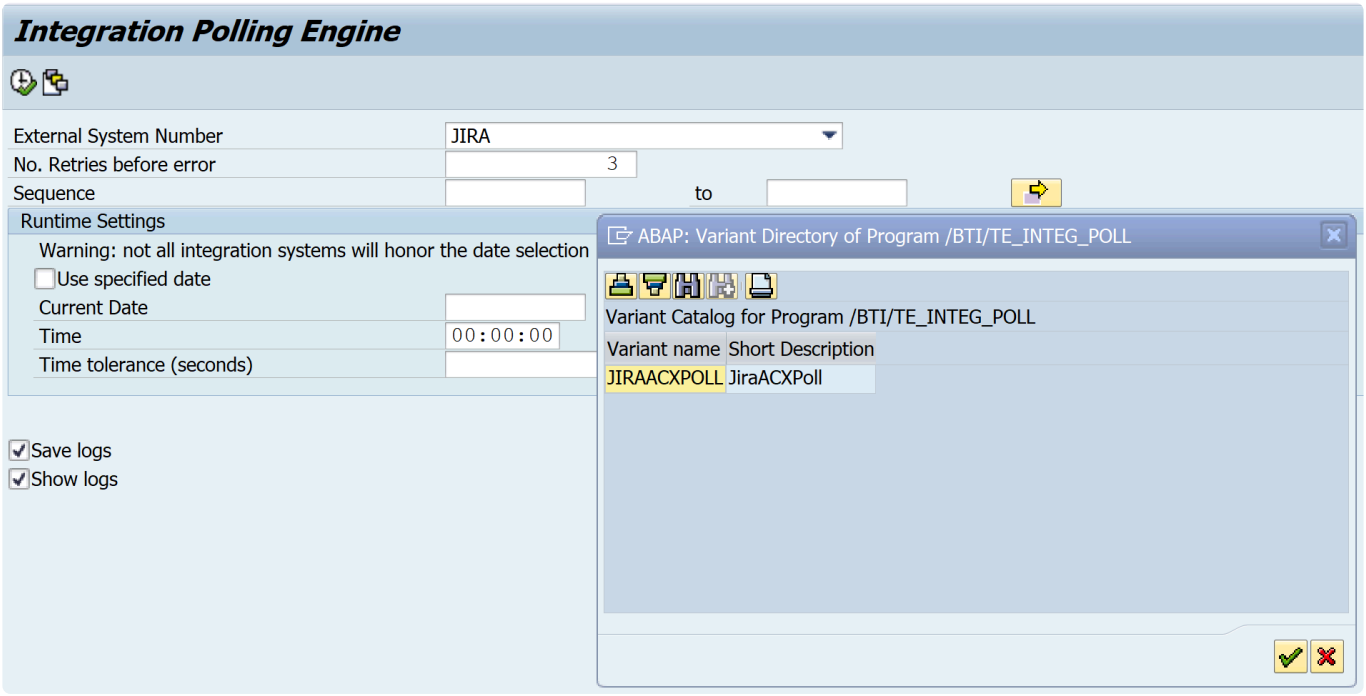
3.2. Inbound Integration Polling

This document explains the extensions made to the AC integration to cater for polling external systems for changes.

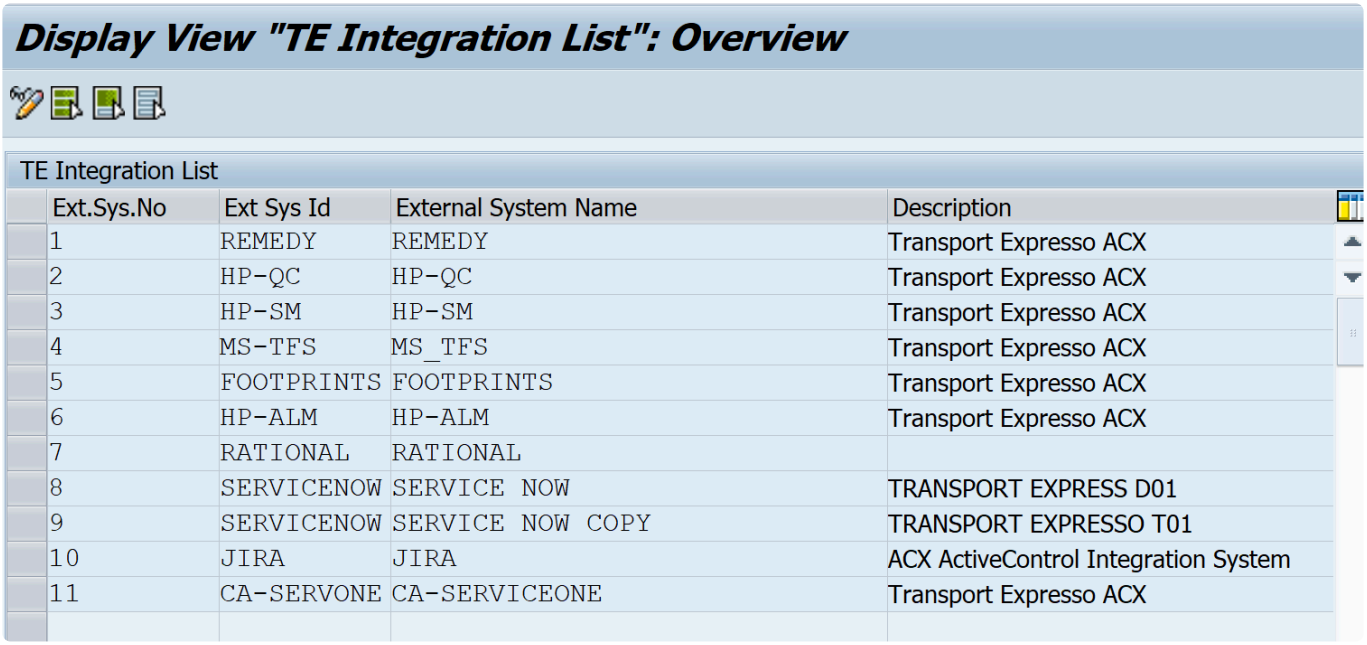
3.2.1. AC Polling Engine

This program is used to poll the external system for new tickets created since the last time the program was run. A variant must be created (and a job scheduled) for the particular external system along with the number of attempts that the program should make to poll the external system before failing.

Engine: SE38 /BTI/TE_INTEG_POLL
Polls an external system. The reference time might be relevant or not depending on the remote system.



To create a Poll Engine you need to create the System list first, see section AC Integration System List



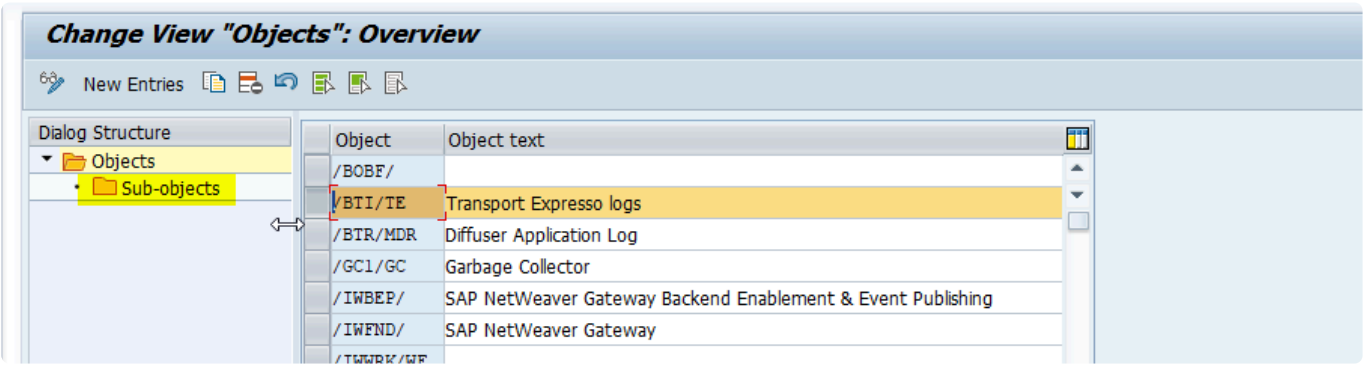
Ext.Sys.No	Ext Sys Id	External System Name	Description
1	REMEDY	REMEDY	Transport Expresso ACX
2	HP-QC	HP-QC	Transport Expresso ACX
3	HP-SM	HP-SM	Transport Expresso ACX
4	MS-TFS	MS_TFS	Transport Expresso ACX
5	FOOTPRINTS	FOOTPRINTS	Transport Expresso ACX
6	HP-ALM	HP-ALM	Transport Expresso ACX
7	RATIONAL	RATIONAL	
8	SERVICENOW	SERVICE NOW	TRANSPORT EXPRESS D01
9	SERVICENOW	SERVICE NOW COPY	TRANSPORT EXPRESSO T01
10	JIRA	JIRA	ACX ActiveControl Integration System
11	CA-SERVONE	CA-SERVICEONE	Transport Expresso ACX

For logging to work, you must go to transaction SLG0, choose /BTI/TE and click on sub-objects. Then click on new entry and create sub-object INTEGRATION_01 and save it. Otherwise, you will get an error when running the polling engine.

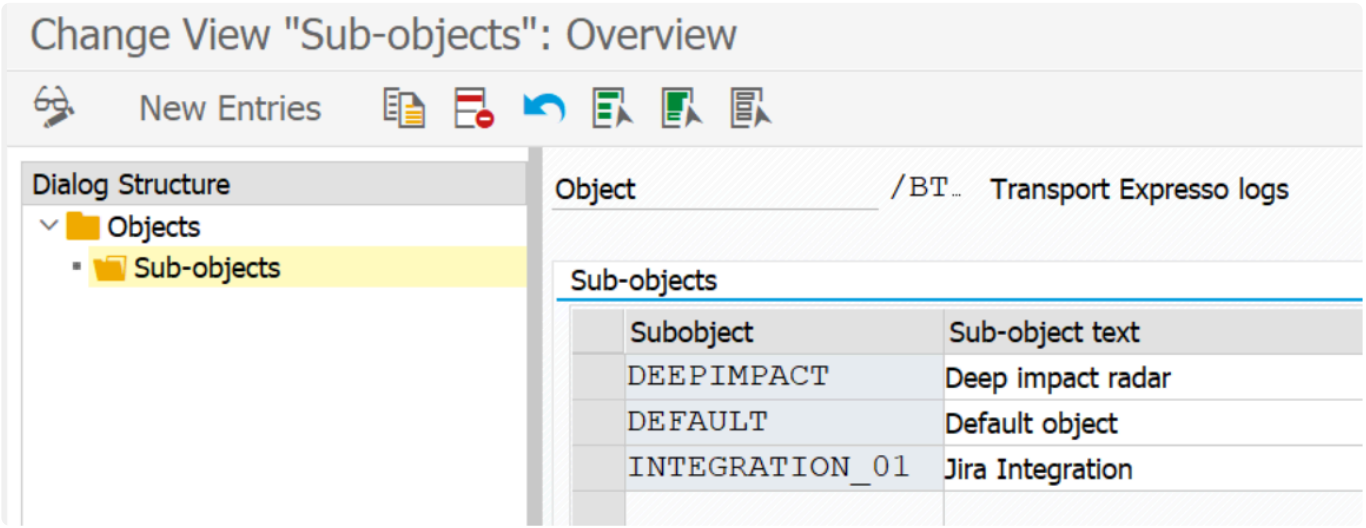
3.2.1.1. Create sub-object INTEGRATION_01

Go to transaction SLG0

Choose /BTI/TE and click on sub-objects.



Click on new entry and create sub-object INTEGRATION_01 and save it.

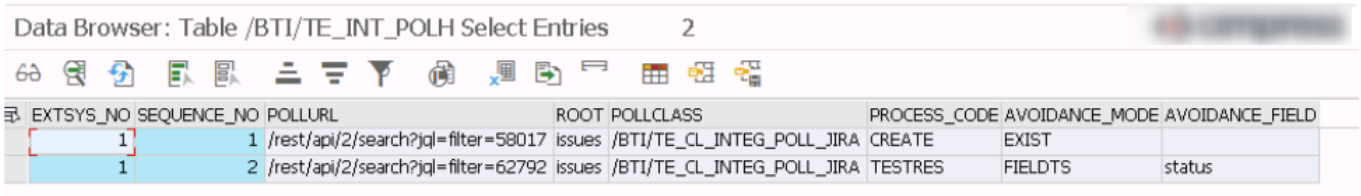


3.2.1.2. Polling Engine Batch Job

Job for: /BTI/TE_INTEG_POLL
A job need to be scheduled via SM36 to run the external system polling.

Program	Variant	Job Name	Frequency	Notes
/BTI/TE_INTEG_POLL	JIRA	TE INTEGRATION 5 MIN	Every 3 minutes	

3.2.2. AC Polling Header


Data Browser: Table /BTI/TE_INT_POLH Select Entries 2

EXTSYS_NO	SEQUENCE_NO	POLLURL	ROOT	POLLCLASS	PROCESS_CODE	AVOIDANCE_MODE	AVOIDANCE_FIELD
1	1	/rest/api/2/search?jql=filter=58017	issues	/BTI/TE_CL_INTEG_POLL_JIRA	CREATE	EXIST	
1	2	/rest/api/2/search?jql=filter=62792	issues	/BTI/TE_CL_INTEG_POLL_JIRA	TESTRES	FIELDTS	status

/BTI/TE_INT_POLH Header for polling items. The RFC destination used comes from /BTI/TE_INT_SYST-INT_PORT

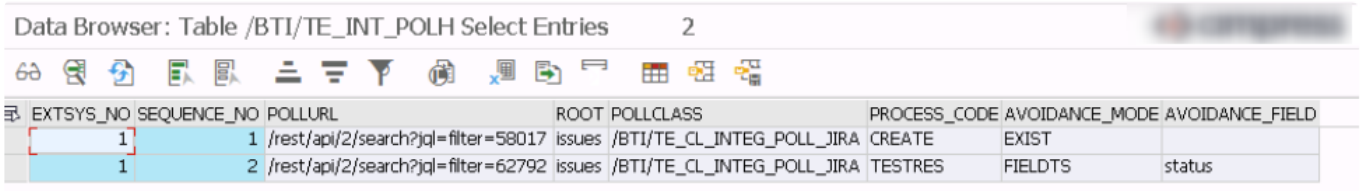
EXTSYS_NO	Integration system number
SEQUENCE_NO	sequence
POLLURL	URL to be polled
ROOT	root node for relevant entries in response
POLLCLASS	Polling class (must implement /BTI/TE_IF_INTEGRATION_POLLER)
PROCESS_CODE	See table /BTI/TE_INT_PC
AVOIDANCE_MODE	Algorythm used to avoid reprocessing. EXIST,TIMESTAMP,FIELDTS
AVOIDANCE_FIELD	Reference field for avoidance mode

When avoidance mode is exists, a check is performed to see if the action is still relevant (i.e. task creation of an existing task)
When is set to timestamp, the record is considered if changed after last time it was processed
When is set to field timestamp, the value is reprocessed if the avoidance field changed since last time.

The action in INT_POLH must match a line in INT_PC with a class implementing interface /BTI/TE_IF_INTEGRATION_ACTION
This is not required for the traditional use in outbound communication.

Also, the URL in the polling should:

- be relative to the RFC destination (no http://.., just a path),
- use Jira's REST API (no /issues/?filter=10600, instead /rest/api/2/search?jql=filter=10600),
- the port and server will be in the RFC destination,
- user and password must be set in the RFC destination,
- Root = issues, default for all REST API integrations.


Data Browser: Table /BTI/TE_INT_POLH Select Entries 2

EXTSYS_NO	SEQUENCE_NO	POLLURL	ROOT	POLLCLASS	PROCESS_CODE	AVOIDANCE_MODE	AVOIDANCE_FIELD
1	1	/rest/api/2/search?jql=filter=58017	issues	/BTI/TE_CL_INTEG_POLL_JIRA	CREATE	EXIST	
1	2	/rest/api/2/search?jql=filter=62792	issues	/BTI/TE_CL_INTEG_POLL_JIRA	TESTRES	FIELDTS	status

3.2.3. AC Polling Fields

/BTI/TE_INT_POLF
Fields for polling items

Task headers

EXTSYS_NO	Integration system number
SEQUENCE_NO	sequence
EXTERNAL_REF	Field in external system or value
TEFIELDREF	Field in AC (structure /BTI/TE_IF_INTEGRATION_PARSER=>TY_TASK)
DATAFORMAT	string, timestamp or constant

Notes: when data format is set to C external_ref is a Constant value to be assigned to the AC field

When the external_ref includes a [] means it’s a table. Must match another in tefieldref

POLF is linked to /BTI/TE_INT_MAPP this is used to map what an external system sends, only the primary AC tables relevant for a Business Task creation. as custom field 12 to the group.

POLF is linked to /BTI/TE_INT_CONV this is used to convert AC against ticket field values.

Data Browser: Table /BTI/TE_INT_POLF Select Entries 23

EXTSYS_NO	SEQUENCE_NO	EXTERNAL_REF	TEFIELDREF	DATAFORMAT
1	1	fields-assignee-emailAddress	HEADER-CF_550	
1	1	fields-assignee-key	HEADER-TESTERID	
1	1	fields-components[1]-name	HEADER-TYPEID	
1	1	fields-created	TIMESTAMP	T
1	1	fields-customfield_10030-value	HEADER-PROJECTID	
1	1	fields-customfield_11046	HEADER-CF_500	
1	1	fields-customfield_16043	HEADER-PRIORITY	
1	1	fields-customfield_16044	HEADER-CF_501	
1	1	fields-customfield_17641-value	HEADER-GROUPID	
1	1	fields-customfield_18240	HEADER-CF_502	
1	1	fields-description	DESCRIPTION	
1	1	fields-summary	HEADER-CAPTION	
1	1	key	HEADER-REFERENCE	
1	2	0	TESTRESULT	C
1	2	00117022100000000029	TARGETROLE	C
1	2	X	APPROVE	C
1	2	changelog-histories[]-created	CHANGES[]-TIMESTAMP	T
1	2	changelog-histories[]-items[]-field	CHANGES[]-ITEMS[]-FIELD	
1	2	fields-assignee-key	HEADER-TESTERID	
1	2	fields-customfield_16044	HEADER-CF_501	
1	2	fields-description	DESCRIPTION	
1	2	fields-resolution-description	TESTRESTEXT	
1	2	key	HEADER-REFERENCE	

Sequence 1 – Initial Create instruction
Sequence 2 – Update instructions
TEFIELDREF = custom field – Jira Epic

3.2.4. User mapping based on email address

As part of an ActiveControl <> JIRA integration, a Tester has to be assigned to the Business Task being automatically created.

ActiveControl integrations traditionally rely on the username on the ITSM system matching that of the SAP username. Based on this, a user field in the ITSM ticket is then assigned as the Default Tester on the Business Task.

This is typically not the case in JIRA – because user accounts in JIRA are email addresses. To mitigate this issue as part of our out-of-the-box JIRA integration, a user exit solution that was built for a customer in 2017 is now part of standard ActiveControl (since AC8.20). These user exits attempt a user mapping based on email address stored against the user in JIRA and in SAP. If these match, then the Tester can be assigned based on the User assignment on the JIRA ticket.

Configuration

1. Add standard user exit /BTI/TE_EXIT_WSCREATESTER_0080 into /BTI/TE_EXITC table in the Domain Controller.

2. Add standard user exit /BTI/TE_EXIT_WSCHNGTESTER_0082 into /BTI/TE_EXITC table in the Domain Controller.

3.3. Outbound Integration Process

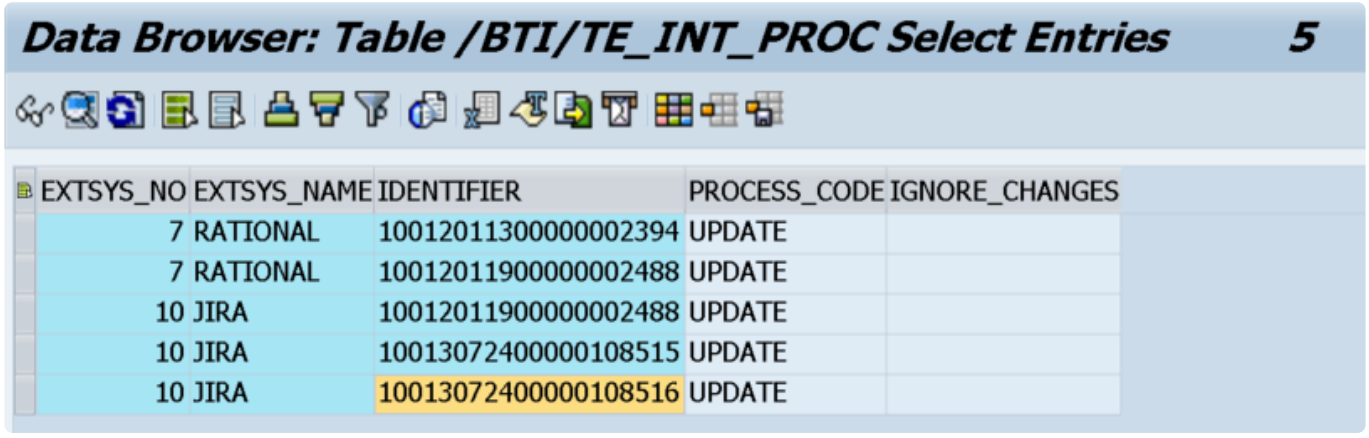
This section guides you through the steps that are needed to configure outbound integration within ActiveControl.
The Integration configuration is maintained through the SAP standard SM30/31 functions where table entries can be created and updated.

3.3.1. Process Identifiers

The standard out of the box integration framework integrates at task level with third party software using task status changes as integration points. A process code will need to be attached to a task deployment or planning status which subsequently needs to be attached to a control point within ActiveControl. Assuming deployment/planning statuses have already been attached to control points within the path, we need to:
To link the process code with a deployment/planning status table '/BTI/TE_INT_PROC' needs to be maintained here the status and process code is attached to the external system that is being integrated with.

/BTI/TE_INT_PROC – Process Identifiers (per system)	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXTSYS_NAME	Full description of external system
IDENTIFIER	This identifier is the crux of the integration framework and denotes a point of integration, more than likely this would be some kind of internal id, in our OOTB example it is a task status. This point of integration is attached to a process code denoted above and this is what would cause an integration to be performed when this identifier is reached.
PROCESS_CODE	The process codes used by the integration framework to perform some kind of action. The framework gets shipped with two standard process codes CREATE and UPDATE.
IGNORE_CHANGES	This flag is set when you wish to ignore previous changes in case the integrated object has skipped through more than one integration point since the integration trigger program was last run.

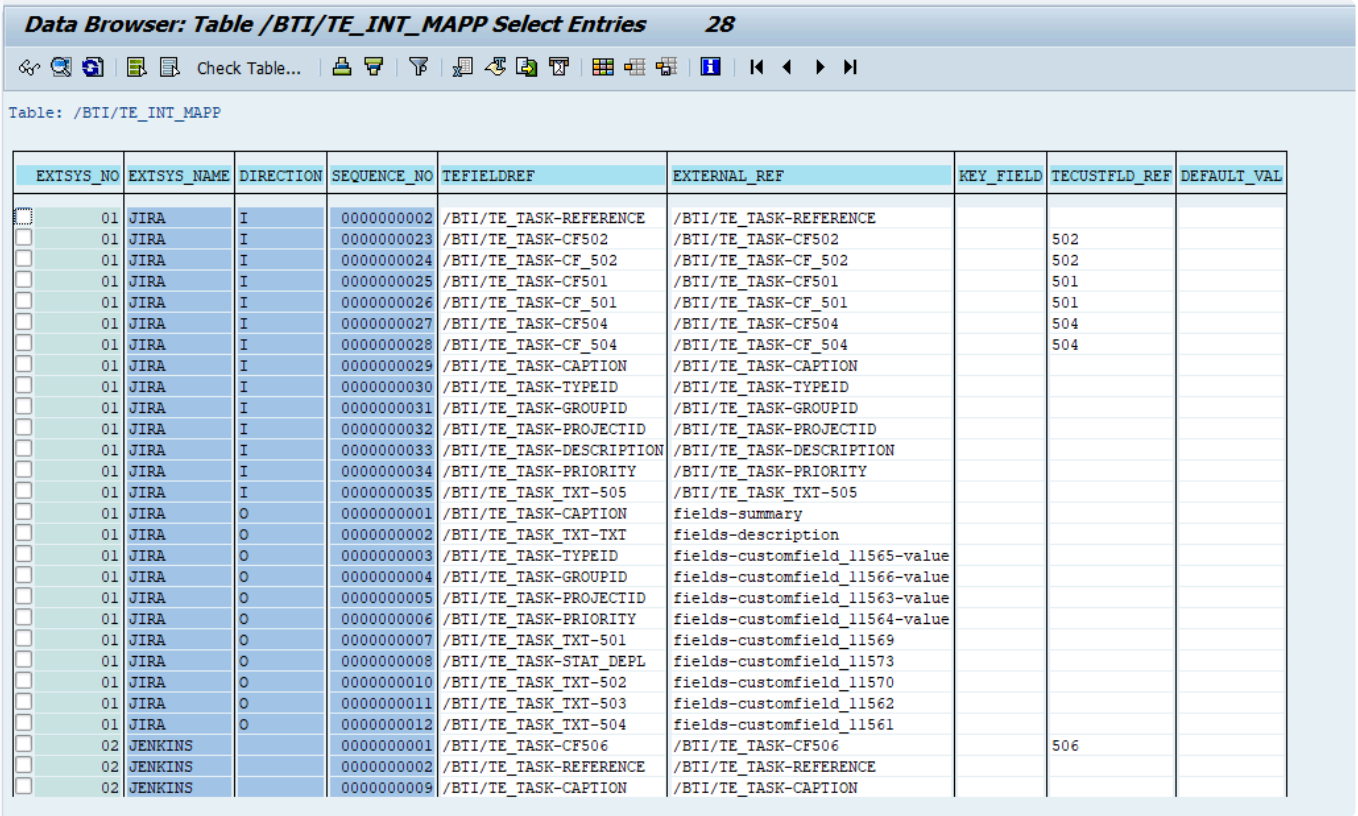
/BTI/TE_INT_PROC = AC deployment status codes – codes found in SE16 table /BTI/TE_TASKSTAT



3.3.2. Mapping

An essential part of the integration framework is mapping ActiveControl fields to the equivalent fields on any external system. This is achieved using the table '/BTI/TE_INT_MAPP'. Ideally, this process will need to be undertaken before the framework can be used. For general fields the AC field should be entered complete with table name into field TEFIELDREF and the external fieldname must be entered in the EXTERNAL_REF field. There is also the functionality to be able to reference any AC Custom fields the custom field ID's would need to be added to TECUSTFIELD_REF, also multiple line itemed fields are able to be handled here such as text fields. Finally, on the mapping table there is a KEY_FIELD field this is used to hold the external system record key in general use a specific non display custom field on the task would be created for this purpose.

/BTI/ TE_INT_MAPP – Integration Mapping	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXTSYS_NAME	Full description of external system
TEFIELDREF	This is the AC Field that needs to be mapped to a field on the external system. This table name is required in the field as well. I.e. /BTI/TE_TASK-PRIORITY
EXTERNAL_REF	This is the fieldname that the frameworks calling web service needs to reference to map across the data.
KEY_FIELD	This field is the link between the AC record, in our task record we have set up a custom field which is hidden from view and in here we store the ID of the created record on the integrated system.
TECUSTFLD_REF	ID of AC Custom field to be mapped.
DEFAULT_VAL	Defaulted Value to be mapped over to the integrated system field.



Data Browser: Table /BTI/TE_INT_MAPP Select Entries 28

Check Table...

Table: /BTI/TE_INT_MAPP

EXTSYS_NO	EXTSYS_NAME	DIRECTION	SEQUENCE_NO	TEFIELDREF	EXTERNAL_REF	KEY_FIELD	TECUSTFLD_REF	DEFAULT_VAL
01	JIRA	I	0000000002	/BTI/TE_TASK-REFERENCE	/BTI/TE_TASK-REFERENCE			
01	JIRA	I	0000000023	/BTI/TE_TASK-CF502	/BTI/TE_TASK-CF502		502	
01	JIRA	I	0000000024	/BTI/TE_TASK-CF_502	/BTI/TE_TASK-CF_502		502	
01	JIRA	I	0000000025	/BTI/TE_TASK-CF501	/BTI/TE_TASK-CF501		501	
01	JIRA	I	0000000026	/BTI/TE_TASK-CF_501	/BTI/TE_TASK-CF_501		501	
01	JIRA	I	0000000027	/BTI/TE_TASK-CF504	/BTI/TE_TASK-CF504		504	
01	JIRA	I	0000000028	/BTI/TE_TASK-CF_504	/BTI/TE_TASK-CF_504		504	
01	JIRA	I	0000000029	/BTI/TE_TASK-CAPTION	/BTI/TE_TASK-CAPTION			
01	JIRA	I	0000000030	/BTI/TE_TASK-TYPEID	/BTI/TE_TASK-TYPEID			
01	JIRA	I	0000000031	/BTI/TE_TASK-GROUPID	/BTI/TE_TASK-GROUPID			
01	JIRA	I	0000000032	/BTI/TE_TASK-PROJECTID	/BTI/TE_TASK-PROJECTID			
01	JIRA	I	0000000033	/BTI/TE_TASK-DESCRIPTION	/BTI/TE_TASK-DESCRIPTION			
01	JIRA	I	0000000034	/BTI/TE_TASK-PRIORITY	/BTI/TE_TASK-PRIORITY			
01	JIRA	I	0000000035	/BTI/TE_TASK_TXT-505	/BTI/TE_TASK_TXT-505			
01	JIRA	O	0000000001	/BTI/TE_TASK-CAPTION	fields-summary			
01	JIRA	O	0000000002	/BTI/TE_TASK_TXT-TXT	fields-description			
01	JIRA	O	0000000003	/BTI/TE_TASK-TYPEID	fields-customfield_11565-value			
01	JIRA	O	0000000004	/BTI/TE_TASK-GROUPID	fields-customfield_11566-value			
01	JIRA	O	0000000005	/BTI/TE_TASK-PROJECTID	fields-customfield_11563-value			
01	JIRA	O	0000000006	/BTI/TE_TASK-PRIORITY	fields-customfield_11564-value			
01	JIRA	O	0000000007	/BTI/TE_TASK_TXT-501	fields-customfield_11569			
01	JIRA	O	0000000008	/BTI/TE_TASK-STAT_DEPL	fields-customfield_11573			
01	JIRA	O	0000000010	/BTI/TE_TASK_TXT-502	fields-customfield_11570			
01	JIRA	O	0000000011	/BTI/TE_TASK_TXT-503	fields-customfield_11562			
01	JIRA	O	0000000012	/BTI/TE_TASK_TXT-504	fields-customfield_11561			
02	JENKINS		0000000001	/BTI/TE_TASK-CF506	/BTI/TE_TASK-CF506		506	
02	JENKINS		0000000002	/BTI/TE_TASK-REFERENCE	/BTI/TE_TASK-REFERENCE			
02	JENKINS		0000000009	/BTI/TE_TASK-CAPTION	/BTI/TE_TASK-CAPTION			

Figure: Example configuration of the JIRA mapping table.

Note that as of ActiveControl 8.60, it is possible to update JIRA fields based on updates made to the corresponding Business Task fields. The configuration required for this optional setup is described in this online Change Note.

3.3.3. Conversions

The integration framework can also take into account value conversions. For instance where a value in ActiveControl could equal one thing maybe its corresponding value in an external could be different although they both mean the same thing. For example:
A AC priority maybe ‘1’ for ‘Low’ however, the same ‘Low’ priority in a Remedy system for example could be ‘4’.
The table ‘/BTI/TE_INT_CONV’ can be used to map the two values together and address these issues.

/BTI/TE_INT_CONV – AC Integration Conversion Table	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXTSYS_NAME	Full description of external system
EXTERNAL_REF	This is the field name of the external field that is on the system to be integrated with.
EXTFLD_ID	This is the AC field value that the conversion needs to take place on.
EXTFLD_VAL	This is the converted value that needs to be fed into the integrated system. For example in our OOTB box example we are performing Solution Manager Integrations only on certain types of task and these types of tasks are set up in solution manager as Support Notification ticket types.

/BTI/TE_INT_CONV = AC Integration Conversation Table – codes found in SE16 table

Data Browser: Table /BTI/TE_INT_CONV Select Entries 31

Check Table...

EXTSYS_NO	EXTSYS_NA...	EXTERNAL_REF	EXTFLD_ID	EXTFLD_VAL
1	JIRA	/BTI/TE_TASK-GROUPID	Core Finance Squad	00117030200000000213
1	JIRA		HR HCM & Data Integration	00117030200000000216
1	JIRA		HR Self Service	00117030200000000216
1	JIRA		Merchant Integration Squad	00117030200000000213
1	JIRA		PMI Squad	00117030200000000213
1	JIRA		SAP Basis & Security	00117022100000000005
1	JIRA		Supply Chain Design & Planning	00117030200000000212
1	JIRA		Supply Chain Execution	00117030200000000212
1	JIRA		Supply Chain Integration	00117030200000000212
1	JIRA	/BTI/TE_TASK-PRIORITY	2	4
1	JIRA		3	3
1	JIRA		4	2
1	JIRA		5	1
1	JIRA		6	4
1	JIRA	/BTI/TE_TASK-PROJECTID	CM	00117022100000000001
1	JIRA		Columbus Aladdin M	00117030200000000233
1	JIRA		Columbus Supply Chain Enhancements	00117030200000000232
1	JIRA		Finance Post Merger Integration	00117030200000000230
1	JIRA		Invoice Automation Project	00117030200000000228
1	JIRA		Maintenance Windsor	00117030200000000231
1	JIRA		RTB: Finance	00117030200000000227
1	JIRA		Sales and Financial integration	00117030200000000229
1	JIRA	/BTI/TE_TASK-TESTERID	__DEFAULT_USER__	epoellinger
1	JIRA	/BTI/TE_TASK-TYPEID	Basis	00117030200000000221
1	JIRA		Development	00117030200000000221
1	JIRA		Finance	00117030200000000221
1	JIRA		HR	00117030200000000221
1	JIRA		Security	00117030200000000221
1	JIRA		Supply Chain and Operations	00117030200000000221
1	JIRA	transition-id	00000000000000000000	201
1	JIRA		00117032900000000785	481

▼ {expand: "transitions",…}
  expand: "transitions"
  ▼ transitions: [{id: "11", name: "To Do",…}, {id: "21", name: "In Progress",…}, {id: "31", name: "Done",…},…]
    ▶ 0: {id: "11", name: "To Do",…}
    ▶ 1: {id: "21", name: "In Progress",…}
    ▶ 2: {id: "31", name: "Done",…}
    ▶ 3: {id: "41", name: "Ready for QA",…}
    ▼ 4: {id: "51", name: "QA in Progress",…}
      fields: {}
      hasScreen: false
      id: "51"
      name: "QA in Progress"
      ▼ to: {self: "https://basistechnologies.atlassian.net/rest/api/2/status/10004", description: "",…}
        description: ""
        iconUrl: "https://basistechnologies.atlassian.net/images/icons/statuses/generic.png"
        id: "10004"
        name: "QA in Progress"
        self: "https://basistechnologies.atlassian.net/rest/api/2/status/10004"
        ▶ statusCategory: {self: "https://basistechnologies.atlassian.net/rest/api/2/statuscategory/4", id: 4,…}
    ▶ 5: {id: "61", name: "Peer Review",…}
    ▶ 6: {id: "71", name: "Backlog",…}

Value used in my sample mapping

Resulting status

3.3.4. Polling URL

As part of the ActiveControl/JIRA integration – it is possible to automatically populate the URL to the Business Task within the ActiveControl WebUI to a custom field in JIRA.

This is achieved via a User Exit solution and some associated configuration.

Configuration Steps

Details on how to setup this URL population can be found in this online Knowledge Article

Please note that this capability was added in ActiveControl 8.20. Customers using the JIRA integration on earlier ActiveControl releases will need to upgrade to use this feature.

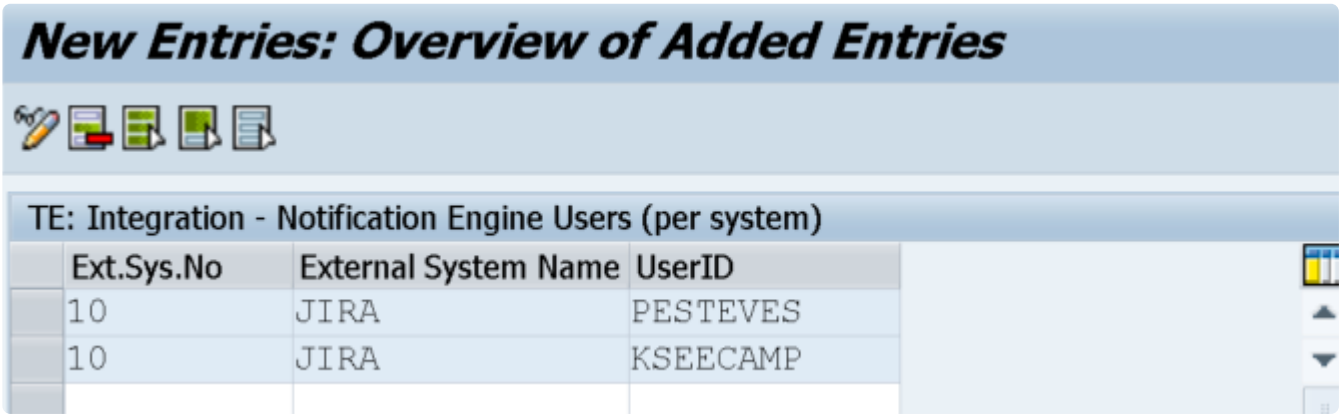
3.3.5. Number Range

SNRO /BTI/TE_RF

3.3.6. Notification Users

It is possible to set up ‘Notification Users’ per external system that can be notified when an integration message has gone into an error status. This is run through the Email Notification Engine and the table that needs to be maintained is ‘/BTI/TE_INT_USR’.

/BTI/TE_INT_USR – Notification Engine Users (per system)	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXTSYS_NAME	Full description of external system
USERID	SAP Logon ID of person that needs to be notified of failed integrations.

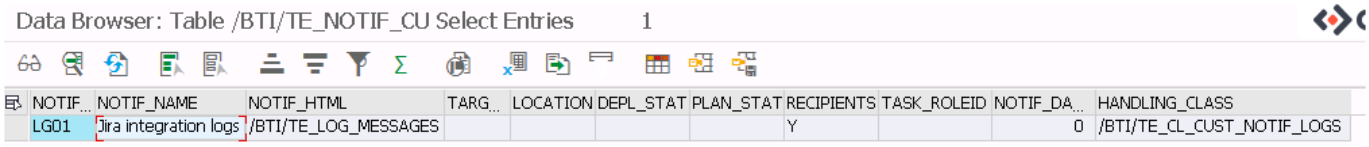


Also see advanced notifications as part of Error Handling and Notifications below.

3.3.7. Notification Logs

This is part of the process for enhanced error handling and notifications.

/BTI/TE_NOTIF_CU needs to have the Recipients flag set

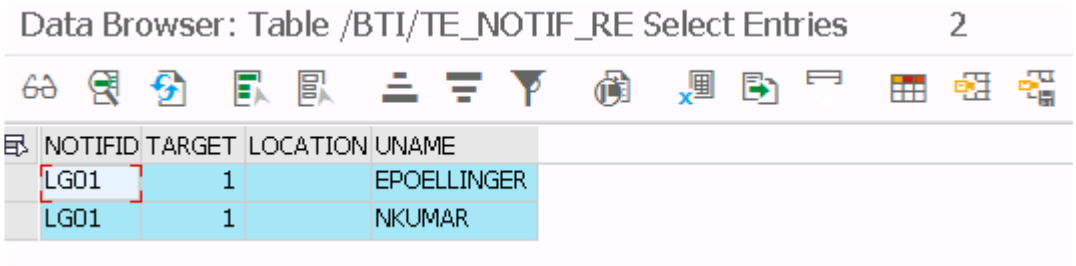

Data Browser: Table /BTI/TE_NOTIF_CU Select Entries 1

NOTIF...	NOTIF_NAME	NOTIF_HTML	TARG...	LOCATION	DEPL_STAT	PLAN_STAT	RECIPIENTS	TASK_ROLEID	NOTIF_DA...	HANDLING_CLASS
LG01	Jira integration logs	/BTI/TE_LOG_MESSAGES					Y		0	/BTI/TE_CL_CUST_NOTIF_LOGS

3.3.8. Notification Recipients

This is part of the process for enhanced error handling and notifications. These users will be notified against specific config values defined in /BTI/TE_NOTIF_CP.

/BTI/TE_NOTIF_RE needs to have valid users set up

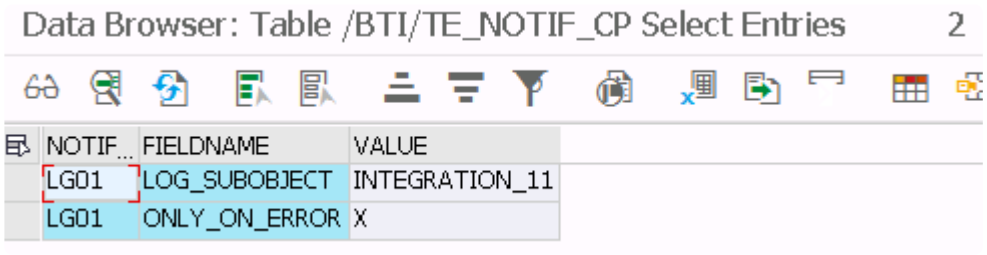


NOTIFID	TARGET	LOCATION	UNAME
LG01	1		EPOELLINGER
LG01	1		NKUMAR

3.3.9. Notification Connector

This is part of the process for enhanced error handling and notifications. The configuration in table /BTI/TE_NOTIF_CP lists the error type to be communicated to the users in above table /BTI/TE_NOTIF_RE.

/BTI/TE_NOTIF_CP



3.3.10. Integration Trigger Engine

A trigger program should be scheduled to pick up the Task status changes that need to be interfaced to the external system(s). This trigger program selects the appropriate AC records, dependent on the configuration set up above, and passes it through the mapping engine. It then stores the mapped integration transactions into a set of standard tables.

Program Name: /BTI/TE_INTEG_TRIGGER

Selection Option	Description
External System	The external system the trigger program is to be run against
Integration Trigger Condition	The trigger program /BTI/TE_INTEG_TRIGGER reads the updated business task based on various conditions defined in table /BTI/TE_INT_COND. Although historically with the Integration Framework it was only possible to trigger the outbound integration based on changes to Business Task [Deployment Status], it is also now possible to trigger based on other updates, such as updates to other Business Task standard or custom fields. If nothing is entered in this field, then the default will be ‘Based on status update(Traditional)’.
Task ID	Task(s) the trigger program will be run against
Task Type	Task Type(s) the trigger program will be run against
Task Reference	Task Reference the trigger program will be fun against
Task Priority	Task Priority the trigger program will be run against
Send previous changes	Select this checkbox if Task status changes is ‘backwards’ in the process and this change should be sent to the external system
Run as though Last Run on	The date and time of the ‘last’ run can be entered manually if this flag is checked
Run Date	The date of the last run (if manually entered)
Run Time	The time of the last run (if manually entered)

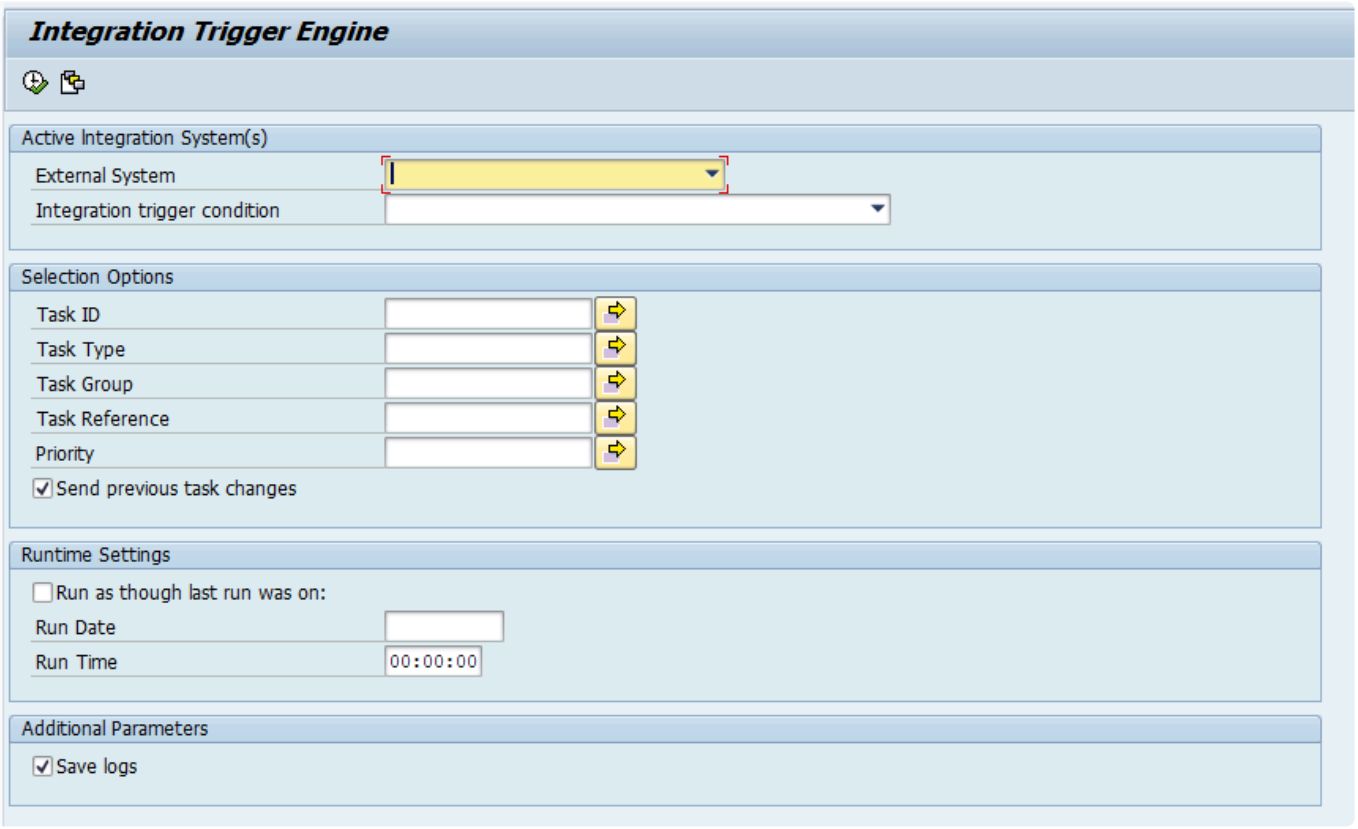


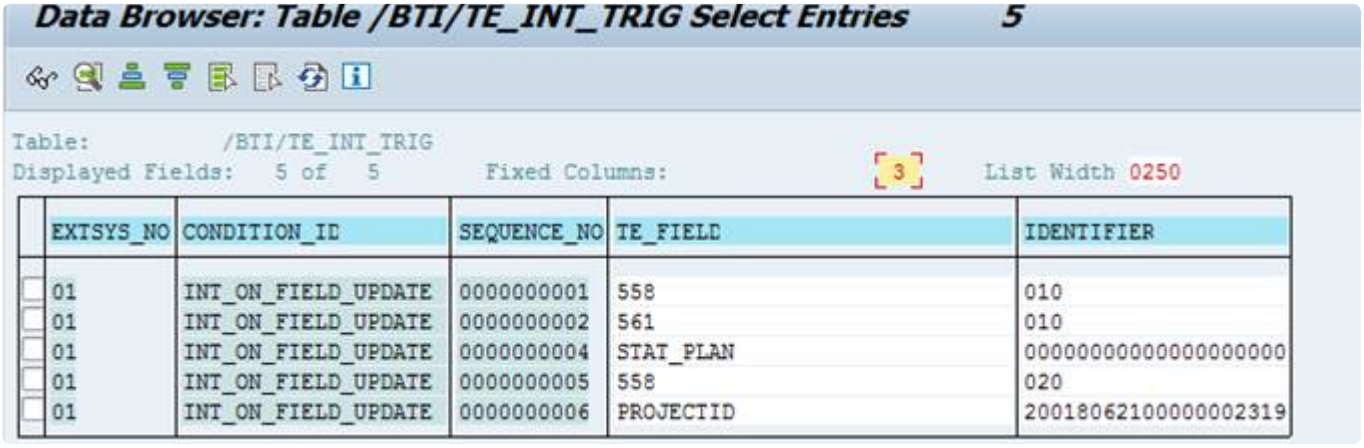
Figure: A variant of of Program /BTI/TE_INTEG_TRIGGER needs to be created and scheduled to run periodically (typically every 5-10 minutes).

Using trigger condition based on the update in standard/custom field of a Busines Task

Table /BTI/TE_INT_TRIG is required to be configured if the triggering strategy is based on the update in standard/custom field of a business task. The condition ID for this strategy is INT_ON_FIELD_UPDATE.

Field	Description
EXT_SYS_NO	Integration System Number
CONDITION ID	INT_ON_FIELD_UPDATE from table /BTI/TE_INT_COND
TE_FIELD	Possible fields are : CAPTION REFERENCE GROUPID TYPEID TESTERID PRIORITY PROJECTID STAT_DEPL STAT_PLAN {Custom Field Number}

IDENTIFIER	Corresponding values that are required to trigger the integration.



Data Browser: Table /BTI/TE_INT_TRIG Select Entries 5

Table: /BTI/TE_INT_TRIG
Displayed Fields: 5 of 5 Fixed Columns: 3 List Width 0250

EXTSYS_NO	CONDITION_ID	SEQUENCE_NO	TE_FIELD	IDENTIFIER
01	INT_ON_FIELD_UPDATE	0000000001	558	010
01	INT_ON_FIELD_UPDATE	0000000002	561	010
01	INT_ON_FIELD_UPDATE	0000000004	STAT_PLAN	00000000000000000000
01	INT_ON_FIELD_UPDATE	0000000005	558	020
01	INT_ON_FIELD_UPDATE	0000000006	PROJECTID	20018062100000002319

Figure: Example configuration of table /BTI/TE_INT_TRIG

3.3.10.1. Trigger Engine Batch Job

Job for: /BTI/TE_INTEG_TRIGGER
A job need to be scheduled via SM36 to run the generation of appropriate items for integration.

Program	Variant	Job Name	Frequency	Notes
/BTI/TE_INTEG_TRIGGER	CPS_TE_JIRA_IN	/BTI/TE_INTEG_TRIGGER	Every 5 minutes	

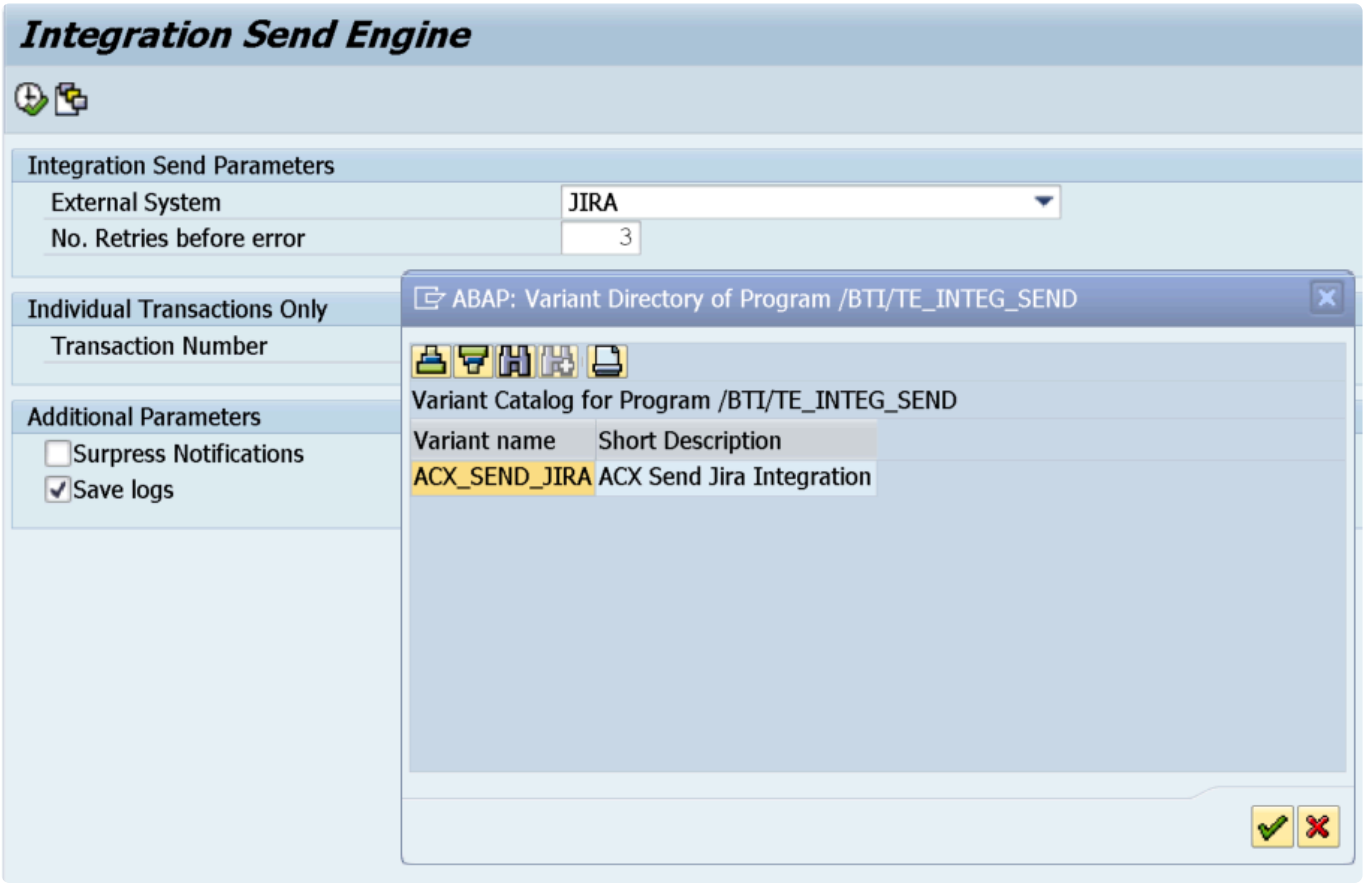
3.3.11. Integration Send Engine

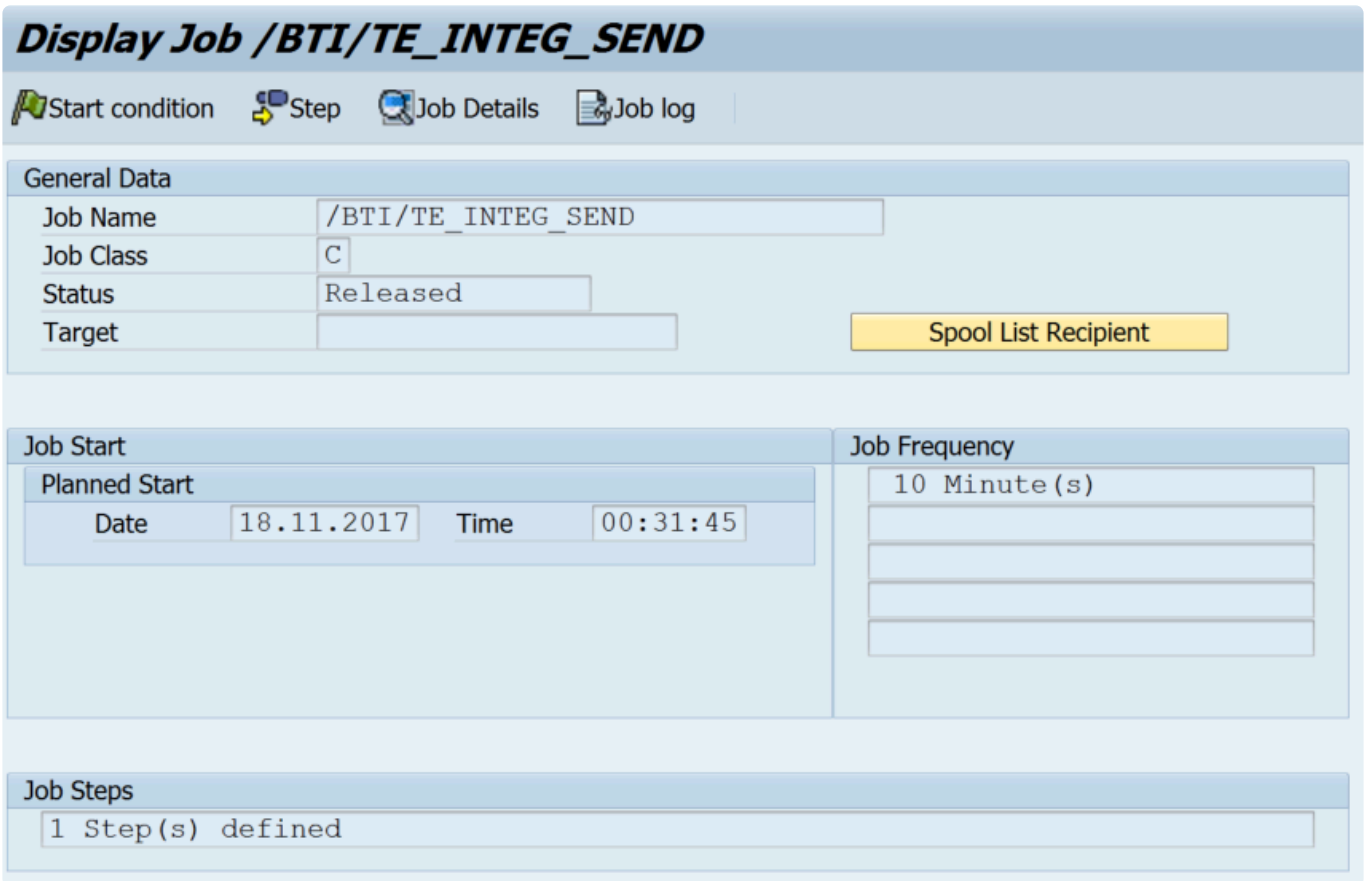
A send program is then scheduled to pick up the mapped transactions and send them out to the configured external systems. It retrieves the required records and then uses the configured send methods for each particular integration scenario to actually push the data out to the receiving systems. If a standard send method is not available for a particular external system (maybe the ticketing system is a ‘home-grown’ application), then custom send methods can be created and utilised in the Integration Framework.

Program Name: /BTI/TE_INTEG_SEND SE38

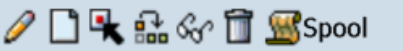
Selection Option	Description
External System	The external system the send program is to be run against
No. of Retries	The number of times the send program will try to send an integration transaction before issuing an error
Transaction Number	Specific integration transactions for the send program to process
Supress Notifications	Makes sure that no notification emails are sent when the transactions are processed

The outcome of the send process is recorded for audit purposes. If successful, any updates configured are made to the AC data objects, alternatively if errors have occurred, the send program will try to re-send (if configured to do so) a certain number of times before marking the transaction in error and sending a notification to the relevant person(s) within the organisation.


Integration Send Engine
Integration Send Parameters
External System
JIRA
No. Retries before error
3
Individual Transactions Only
Transaction Number
Additional Parameters
Surpress Notifications
Save logs
ABAP: Variant Directory of Program /BTI/TE_INTEG_SEND
Variant Catalog for Program /BTI/TE_INTEG_SEND
Variant name
Short Description
ACX_SEND_JIRA
ACX Send Jira Integration


Display Job /BTI/TE_INTEG_SEND
Start condition
Step
Job Details
Job log
General Data
Job Name
/BTI/TE_INTEG_SEND
Job Class
C
Status
Released
Target
Spool List Recipient
Job Start
Planned Start
Date
18.11.2017
Time
00:31:45
Job Frequency
10 Minute(s)
Job Steps
1 Step(s) defined

Step List Overview



No.	Program name/command	Prog. type	Spool list	Parameters	User	Lang.
1	/BTI/TE_INTEG_SEND	ABAP		ACX_SEND_JIRA	TE_BATCH	EN

3.3.11.1. Send Engine Batch Job

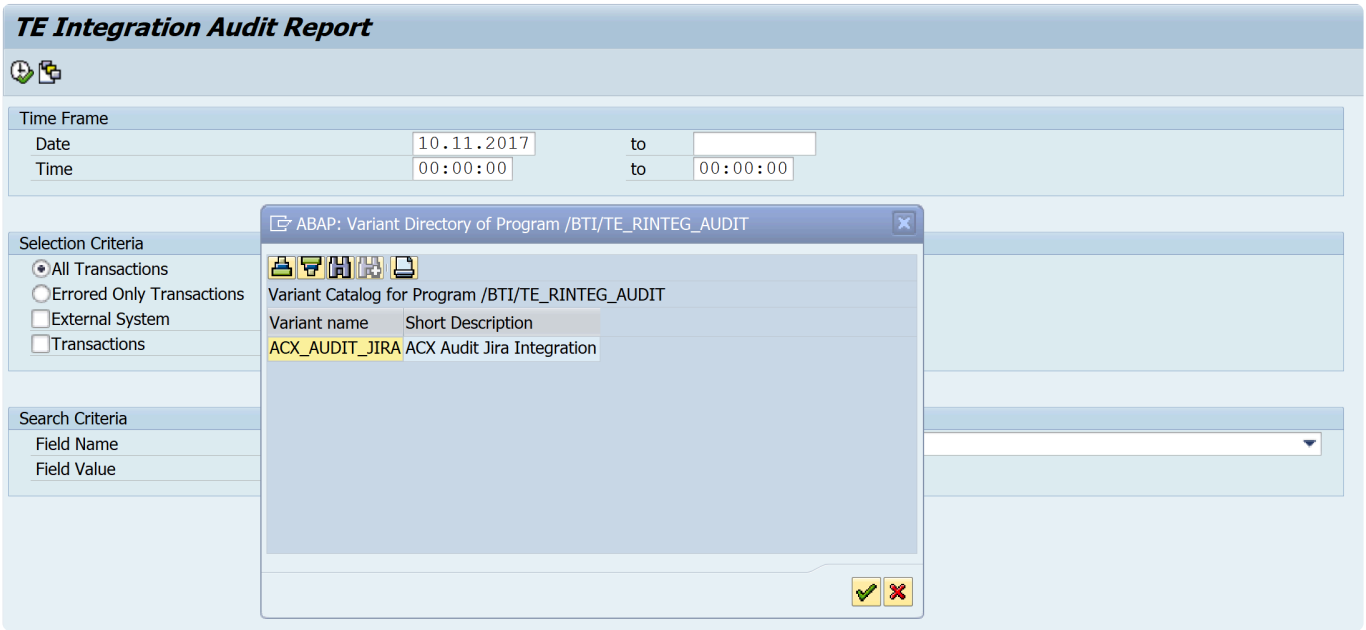
Job for: /BTI/TE_INTEG_SEND
A job need to be scheduled via SM36 to run the actual transmission of information via the integration.

Program	Variant	Job Name	Frequency	Notes
/BTI/TE_INTEG_SEND	CPS_TE_JIRA_SD	/BTI/TE_INTEG_SEND	Every 5 minutes	

3.3.12. Integration Audit Report

At any time, the Integration Reporting Console can be used to see the status of all integrations, the status and history of each transaction and can also be used to update the underlying transactional data, if required, to fix errors.
Program Name: /BTI/TE_RINTEG_AUDIT se38

Selection Option	Description
Date	Date range for the report
Time	Time range for the report
All transactions/ Transactions in error	Select if all transactions should be displayed or just transactions in error
External System	Show only transactions for a specific external system
Transactions	Show only specific transaction numbers
Field Name	The external system field name
Field Value	The value in the external field



3.3.12.1. Audit Report Batch Job

Job for: /BTI/TE_RINTEG_AUDIT
A job need to be scheduled via SM36 to run the actual transmission of information via the integration.

Program	Variant	Job Name	Frequency	Notes
/BTI/TE_RINTEG_AUDIT	CPS_TE_JIRA_SD	/BTI/TE_RINTEG_AUDIT	Every 5 minutes	

4. OTHER CONFIGURATION TABLES

The following are other configuration tables pertaining to the ActiveControl Integration Framework.

Notification Engine
ActiveControl standard Notification Engine (/BTI/TE_RNOTIFICATION_ENGINE) is used to send automated emails. The Notification Engine includes a notification for Integration issues. AC Notification Engine will email the AC Admins when the integration has failed.

Backup
Standard AC Backup program /BTI/TE_RBACKUP_DATA_EXP_NEW is used for taking a regular backup of all AC configuration, including the Integration. In the event of a major issue, ActiveControl can be restored to that point of configuration (and data). Details of how to import backup are detailed in Basis Technologies online FAQs.

None of these tables are utilised by the JIRA Integration, but they are documented here for completeness.

/BTI/TE_INT_CPNT	AC: Integration Control Points [NOT USED AS PART OF JIRA INTEGRATION]
/BTI/TE_INT_FILT	AC: Integration (DDC Only) Filters Table [NOT USED AS PART OF JIRA INTEGRATION]
/BTI/TE_INT_FLDE	AC Integration: Function Module Exit (Complex Field Mapping) [NOT USED AS PART OF JIRA INTEGRATION]
/BTI/TE_INT_LOC	Location for integration approvals [NOT USED AS PART OF JIRA INTEGRATION]
/BTI/TE_INT_MAIL	[NOT USED AS PART OF JIRA INTEGRATION]
/BTI/TE_INT_STLS	AC Integration Status List [NOT USED AS PART OF JIRA INTEGRATION]

4.1. Complex Mapping

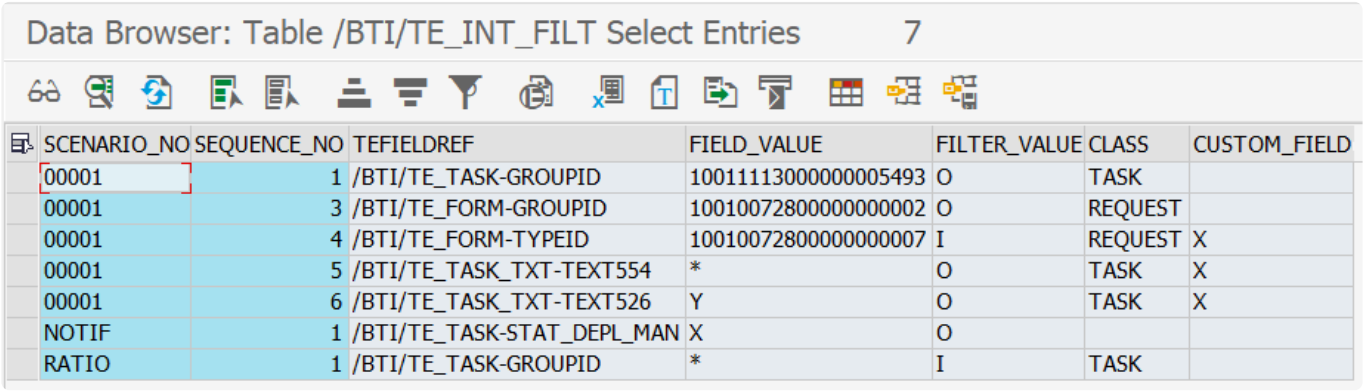
For complex mapping scenarios, a specific function module can be created on the ActiveControl Domain Controller to perform whatever mapping or transformation that may be required.

/BTI/TE_INT_FLDE – Mapping User Exits	
Field	**Description**
EXTSYS_NO	Main external system identifier, this is the identifier of the system that you wish to integrate with we can have as many systems as we want.
EXITFIELDNAME	External field that this refers to An example of this could be: – Remedy – Solution Manager – ServiceNow - Jira – HP ALM - GitLab
EXITFUNCNAME	The function module to be executed to perform this exit

4.2. Filter Values

There's a scenario field in the integration header.
If this is set, the trigger program and the DDC and RTC integration classes will ignore some tasks based on this table's contents.
Some old version of the framework will filter out everything unless you have a dummy entry here
Set the filter values in /BTI/TE_INT_FILT

/BTI/ TE_INT_FILT – Filter for?	
Field	**Description**
SCENARIO_NO.	
SEQUENCE_NO.	
TEFIELDREF	This is the AC Field that needs to be mapped to a field on the external system. This table name is required in the field as well. I.e. /BTI/TE_TASK-PRIORITY
FIELD_VALUE	
FILTER_VALUE	
CLASS	ActiveControl class identifying Transport Form = Request, and Business Task = Task.
CUSTOM_FIELD	


Data Browser: Table /BTI/TE_INT_FILT Select Entries 7

SCENARIO_NO	SEQUENCE_NO	TEFIELDREF	FIELD_VALUE	FILTER_VALUE	CLASS	CUSTOM_FIELD
00001	1	/BTI/TE_TASK-GROUPID	10011113000000005493	O	TASK	
00001	3	/BTI/TE_FORM-GROUPID	10010072800000000002	O	REQUEST	
00001	4	/BTI/TE_FORM-TYPEID	10010072800000000007	I	REQUEST	X
00001	5	/BTI/TE_TASK_TXT-TEXT554	*	O	TASK	X
00001	6	/BTI/TE_TASK_TXT-TEXT526	Y	O	TASK	X
NOTIF	1	/BTI/TE_TASK-STAT_DEPL_MAN	X	O		
RATIO	1	/BTI/TE_TASK-GROUPID	*	I	TASK	

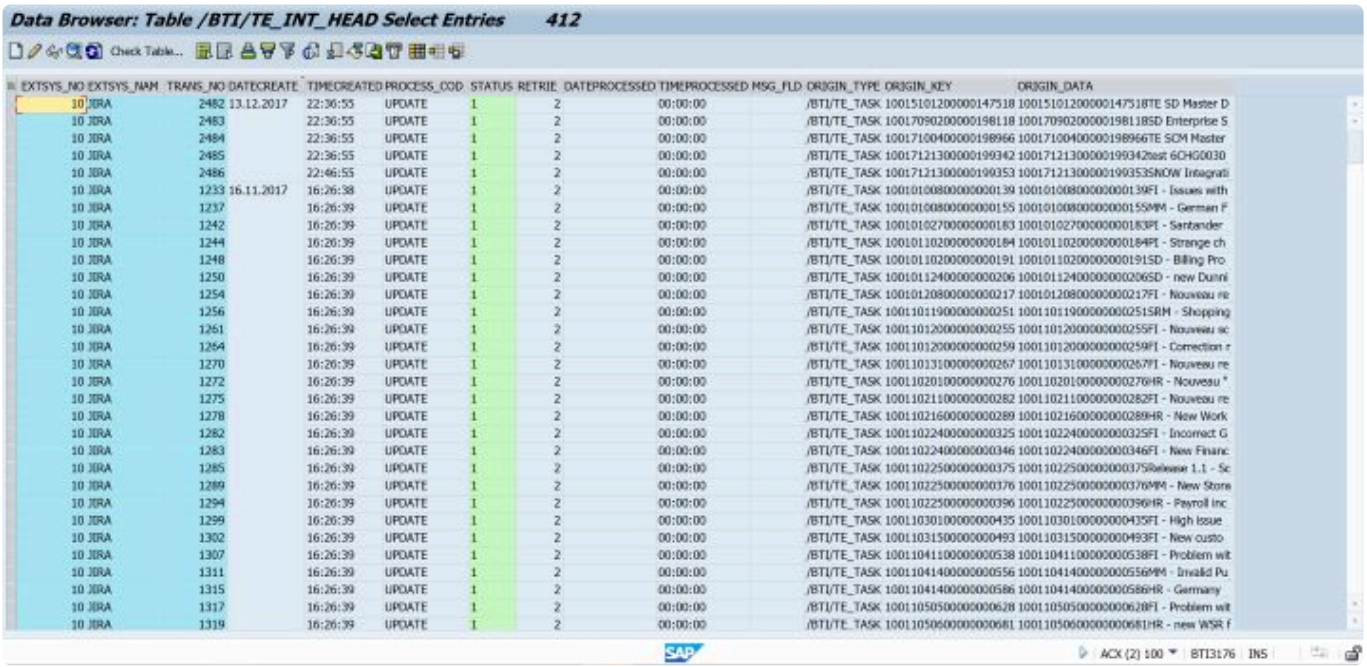
5. New log tables

The following are the main data tables relating to Integration, along with a brief description.

Table	Description
/BTI/ TE_INT_PHEA	Inbound Integration – Header Logging Information Stores the integration items processed successfully, in error or to be retried. Errors are communicated using /BTI/TE_RNOTIFICATION_ENGINE.
/BTI/ TE_INT_HEAD	Outbound Integration – Header Information Stores the integration items processed successfully, in error or to be retried. Errors are communicated using /BTI/TE_RNOTIFICATION_ENGINE.
/BTI/TE_INT_ITEM	Outbound Integration – Item Information Confirms and lists AC Polling Create #1 integration field values mapped, transition # count
/BTI/ TE_INT_MESS	Integration Message Table
/BTI/TE_TEVENTS	Non Integration-specific table, where all event information is stored within AC.

5.1. /BTI/TE_INT_PHEA table

Similar to /BTI/TE_INT_HEAD, stores the integration items processed successfully, in error or to be retried.
They will be reported by the notification engine like the others.

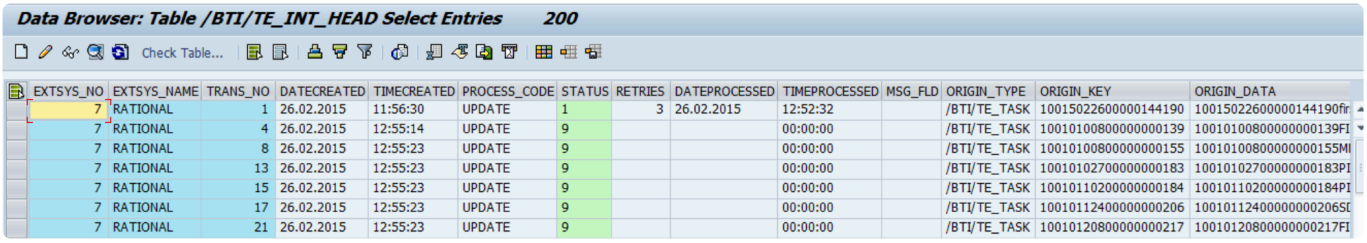


Created sub-object INTEGRATION_01 for object /BTI/TE: R3TR CDAT APPL_LOG

This will enable logs to be created and displayed/sent with notifications.

5.2. Integration Header

Check in tables /bti/te_int_head SE16



Data Browser: Table /BTI/TE_INT_HEAD Select Entries 200

Check Table...

EXTSYS_NO	EXTSYS_NAME	TRANS_NO	DATECREATED	TIMECREATED	PROCESS_CODE	STATUS	RETRIES	DATEPROCESSED	TIMEPROCESSED	MSG_FLD	ORIGIN_TYPE	ORIGIN_KEY	ORIGIN_DATA
7	RATIONAL	1	26.02.2015	11:56:30	UPDATE	1	3	26.02.2015	12:52:32		/BTI/TE_TASK	10015022600000144190	10015022600000144190fr
7	RATIONAL	4	26.02.2015	12:55:14	UPDATE	9			00:00:00		/BTI/TE_TASK	1001010080000000139	1001010080000000139FI
7	RATIONAL	8	26.02.2015	12:55:23	UPDATE	9			00:00:00		/BTI/TE_TASK	1001010080000000155	1001010080000000155MI
7	RATIONAL	13	26.02.2015	12:55:23	UPDATE	9			00:00:00		/BTI/TE_TASK	1001010270000000183	1001010270000000183PI
7	RATIONAL	15	26.02.2015	12:55:23	UPDATE	9			00:00:00		/BTI/TE_TASK	1001011020000000184	1001011020000000184PI
7	RATIONAL	17	26.02.2015	12:55:23	UPDATE	9			00:00:00		/BTI/TE_TASK	1001011240000000206	1001011240000000206SI
7	RATIONAL	21	26.02.2015	12:55:23	UPDATE	9			00:00:00		/BTI/TE_TASK	1001012080000000217	1001012080000000217FI

5.3. Integration Items

/BTI/TE_INT_ITEM

Confirms and lists AC Polling Create #1 integration field values mapped, transition # count

Data Browser: Table /BTI/TE_INT_ITEM Select Entries 199

EXTSYS_NO	EXTSYS_NA...	TRANS_...	SEQUENCE_NO	COUNTER	FLDNAME	FLDVALUE	KEY_FIELD
1	JIRA	117	1		/BTI/TE_TASK-PRIORITY	1	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117030200000000213	
1	JIRA		5		/BTI/TE_TASK-TESTERID	PESTEVES	
1	JIRA		7		transition-id	481	
1	JIRA	114	1		/BTI/TE_TASK-PRIORITY	3	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117022100000000005	
1	JIRA		5		/BTI/TE_TASK-TESTERID	NKUMAR	
1	JIRA		7		transition-id	00117022100000000025	
1	JIRA	112	1		/BTI/TE_TASK-PRIORITY	4	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117022100000000005	
1	JIRA		5		/BTI/TE_TASK-TESTERID	EPOELLINGER	
1	JIRA		7		transition-id	00117022100000000025	
1	JIRA	109	1		/BTI/TE_TASK-PRIORITY	1	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117030200000000213	
1	JIRA		5		/BTI/TE_TASK-TESTERID	PESTEVES	
1	JIRA		7		transition-id	481	
1	JIRA	106	1		/BTI/TE_TASK-PRIORITY	4	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117022100000000005	
1	JIRA		5		/BTI/TE_TASK-TESTERID	EPOELLINGER	
1	JIRA		7		transition-id	00117032900000000785	
1	JIRA	104	1		/BTI/TE_TASK-PRIORITY	2	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117022100000000005	
1	JIRA		5		/BTI/TE_TASK-TESTERID	EPOELLINGER	
1	JIRA	102	1		/BTI/TE_TASK-PRIORITY	4	
1	JIRA		2		/BTI/TE_TASK-TYPEID	00117030200000000221	
1	JIRA		4		/BTI/TE_TASK-GROUPID	00117022100000000005	

5.4. Integration Events

/BTI/TE_TEVENTS

Data Browser: Table /BTI/TE_TEVENTS Select Entries 200

TASK_ID	EVNT_NUM	EVENTCO...	EVENTUSER	EVENTDATE	EVENTTIME	OLD_VAL	NEW_VAL	TARG...	LOCATION
00117022700000000146	13	DS	TE_BATCH	04/04/2017	07:35:21	00117022100000000024	00117032900000000785		
00117031300000000353	8	DS	TE_BATCH		07:35:26	00117022100000000024	00117032900000000785		
00117022300000000117	6	DS	EPOELLINGER	03/31/2017	09:53:28	00117022100000000024	00117032900000000785		
00117022300000000117	7	DS	EPOELLINGER		10:23:50	00117032900000000785	00117022100000000025		
00117022300000000117	8	TS	EPOELLINGER		10:23:49	0	0	3	T
00117022300000000122	8	DS	EPOELLINGER		09:53:28	00117022100000000024	00117032900000000785		
00117022300000000122	9	DS	EPOELLINGER		11:16:03	00117032900000000785	00117033100000000898		
00117022300000000122	10	TS	EPOELLINGER		11:16:03	0	0	3	T
00117022300000000122	11	DS	EPOELLINGER		11:16:45	00117033100000000898	00117022100000000025		
00117022300000000117	4	DS	EPOELLINGER	03/30/2017	16:45:52	00117022100000000020	00117022100000000024		
00117022300000000117	5	TS	EPOELLINGER		16:45:52	0	0	1	T
00117022300000000122	6	DS	EPOELLINGER		16:40:10	00117022100000000020	00117022100000000024		
00117022300000000122	7	TS	EPOELLINGER		16:40:10	0	0	1	T
00117022200000000032	23	DS	EPOELLINGER	03/29/2017	07:06:14	00117022100000000026	00117022100000000020		
00117022200000000032	24	DS	EPOELLINGER		07:44:24	00117022100000000020	00117032900000000787		
00117022200000000032	25	TS	EPOELLINGER		07:44:24	0	0	1	T
00117022200000000032	26	DS	EPOELLINGER		07:44:46	00117032900000000787	00117022100000000024		
00117022200000000032	27	DS	EPOELLINGER		07:46:23	00117022100000000024	00117032900000000785		
00117022300000000122	5	DS	EPOELLINGER		07:07:07	00117022100000000022	00117022100000000020		
00117022700000000146	10	DS	EPOELLINGER		07:07:06	00117022100000000022	00117022100000000020		
00117022700000000146	11	DS	EPOELLINGER		08:29:55	00117022100000000020	00117022100000000024		
00117022700000000146	12	TS	EPOELLINGER		08:29:55	0	0	1	T
00117022700000000150	10	DS	EPOELLINGER		07:05:50	00117022100000000025	00117022100000000020		
00117022800000000200	10	DS	EPOELLINGER		07:07:07	00117022100000000022	00117022100000000020		
00117030700000000258	11	DS	EPOELLINGER		07:07:06	00117022100000000022	00117022100000000020		
00117031000000000312	16	DS	EPOELLINGER		07:05:50	00117022100000000025	00117022100000000022		
00117031000000000312	17	DS	EPOELLINGER		07:07:07	00117022100000000022	00117022100000000020		
00117031300000000344	12	DS	EPOELLINGER		07:07:07	00117022100000000022	00117022100000000020		
00117031300000000353	3	DS	EPOELLINGER		08:10:37	00117022100000000020	00117032900000000787		
00117031300000000353	4	TS	EPOELLINGER		08:10:30	0	0	1	T
00117031300000000353	5	DS	EPOELLINGER		08:29:11	00117032900000000787	00117022100000000020		

SMD (1) 001 | DWNDSAPSMDB01 | INS

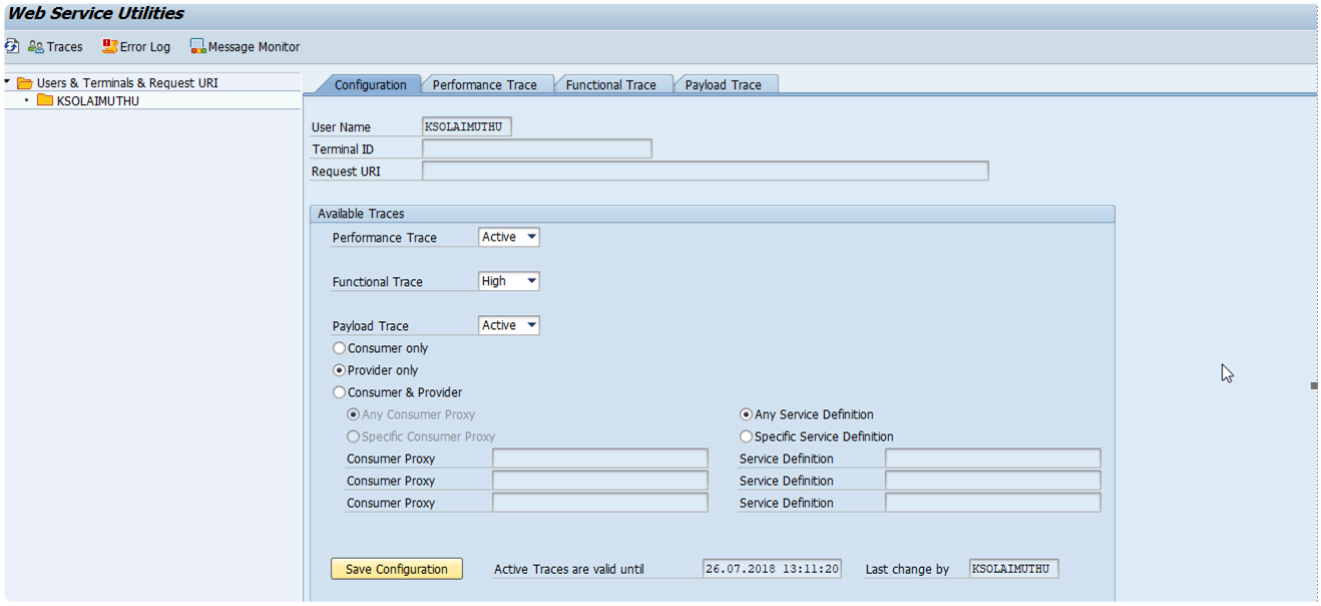
6. Trace for Integration via Proxies

To find if there is a connection established and data transferred between the active control and the external system via proxies, Go to the transaction code SRT_UTIL and click on the ‘Traces’. Give the user ID that is stored in the calling system and enter, the following the configuring screen appears.

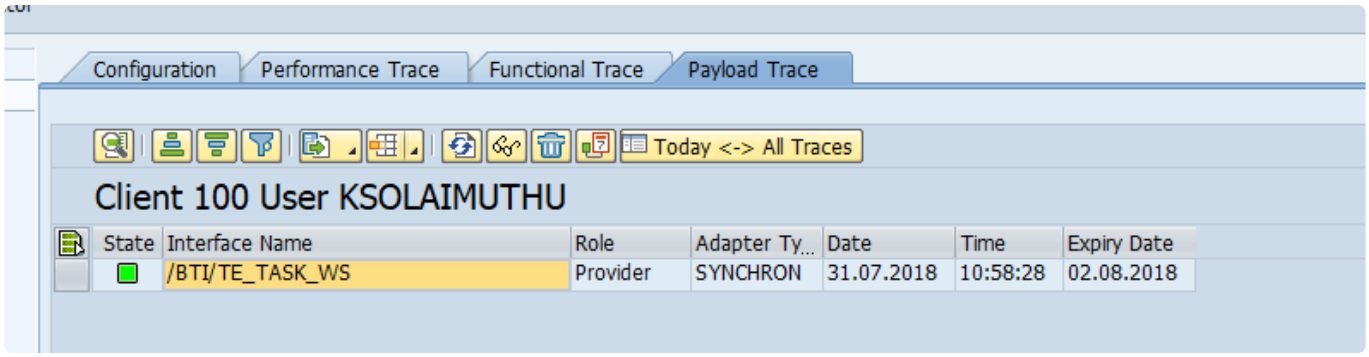
• Consumer only – Choose consumer only if it is a client proxy calling (Outbound).
• Provider only – for external system calling our service (Inbound)

• Set active for performance trace and functional trace if required
• Set active for Payload trace to see the content of the request the calling system sent and response sent from ActiveControl.

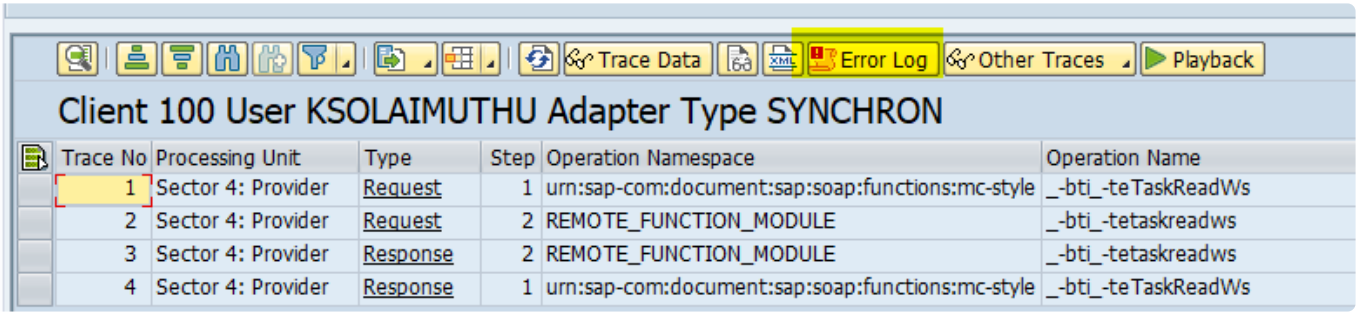
And click ‘Save Configuration’.



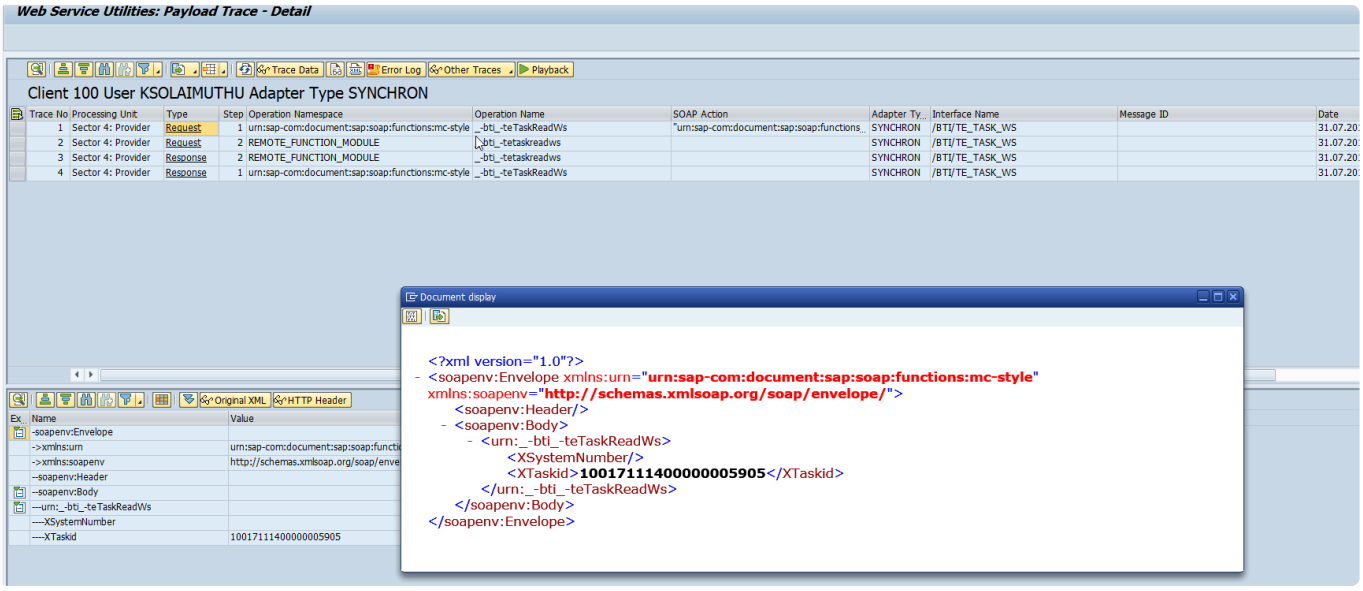
If external system calls our service



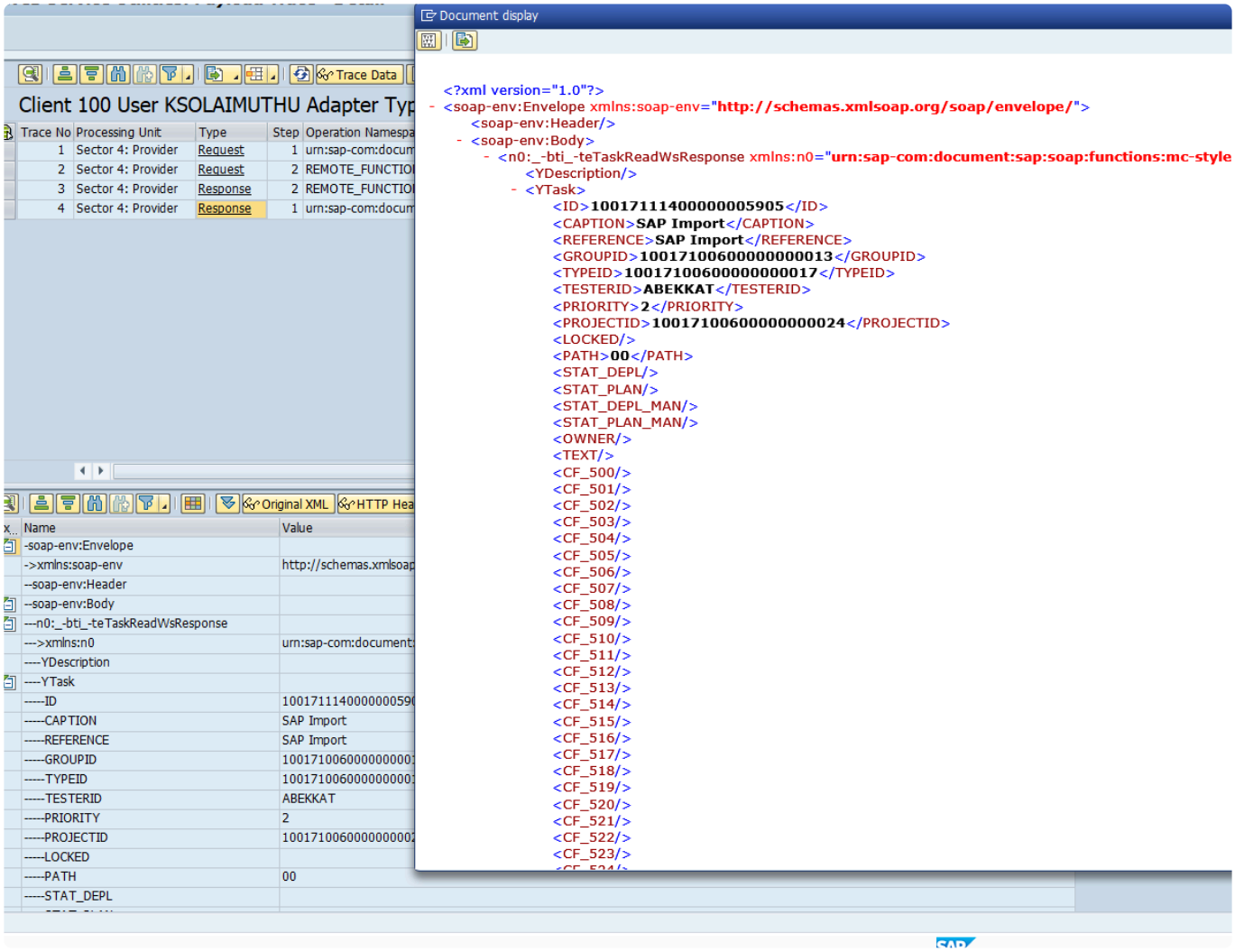
Click on to find more details about the request and response. Also if there is any error, can find in error log



Request:



Response:


Trace Data
Client 100 User KSOLAIMUTHU Adapter Typ
Trace No
Processing Unit
Type
Step
Operation Namespa
1 Sector 4: Provider Request 1 urn:sap-com:docum
2 Sector 4: Provider Request 2 REMOTE_FUNCTIO
3 Sector 4: Provider Response 2 REMOTE_FUNCTIO
4 Sector 4: Provider Response 1 urn:sap-com:docum
Original XML
HTTP Hea
Name
Value
-soap-env:Envelope
->xmlns:soap-env http://schemas.xmlsoap
--soap-env:Header
--soap-env:Body
---n0:_-bti_-teTaskReadWsResponse
--->xmlns:n0 urn:sap-com:document
----YDescription
----YTask
-----ID 1001711140000000590
-----CAPTION SAP Import
-----REFERENCE SAP Import
-----GROUPID 1001710060000000001
-----TYPEID 1001710060000000001
-----TESTERID ABEKKAT
-----PRIORITY 2
-----PROJECTID 1001710060000000002
-----LOCKED
-----PATH 00
-----STAT_DEPL
Document display
<?xml version="1.0"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header/>
<soap-env:Body>
<n0:_-bti_-teTaskReadWsResponse xmlns:n0="urn:sap-com:document:sap:soap:functions:mc-style
<YDescription/>
<YTask>
<ID>1001711140000000590</ID>
<CAPTION>SAP Import</CAPTION>
<REFERENCE>SAP Import</REFERENCE>
<GROUPID>1001710060000000013</GROUPID>
<TYPEID>1001710060000000017</TYPEID>
<TESTERID>ABEKKAT</TESTERID>
<PRIORITY>2</PRIORITY>
<PROJECTID>1001710060000000024</PROJECTID>
<LOCKED/>
<PATH>00</PATH>
<STAT_DEPL/>
<STAT_PLAN/>
<STAT_DEPL_MAN/>
<STAT_PLAN_MAN/>
<OWNER/>
<TEXT/>
<CF_500/>
<CF_501/>
<CF_502/>
<CF_503/>
<CF_504/>
<CF_505/>
<CF_506/>
<CF_507/>
<CF_508/>
<CF_509/>
<CF_510/>
<CF_511/>
<CF_512/>
<CF_513/>
<CF_514/>
<CF_515/>
<CF_516/>
<CF_517/>
<CF_518/>
<CF_519/>
<CF_520/>
<CF_521/>
<CF_522/>
<CF_523/>

7. Troubleshooting Integration Errors

7.1. SAP Error Messages

The following are examples of integration failures, in terms of error message that will seen via SLG1 and likely root-cause.

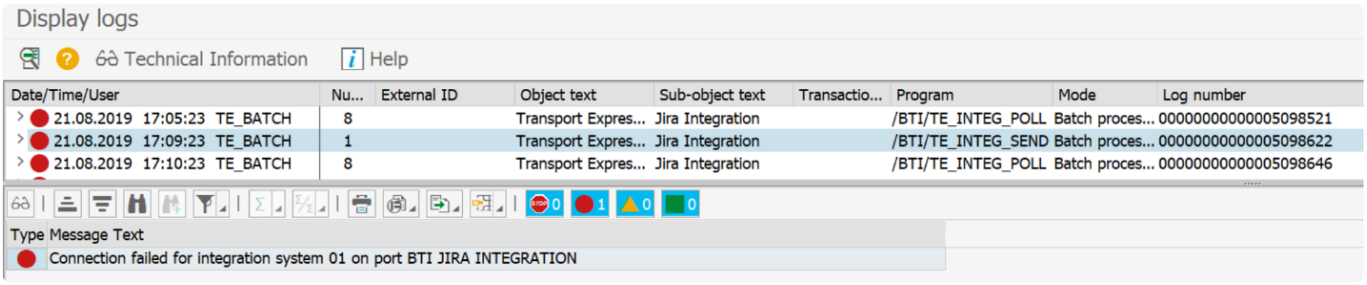
Error	Error Probably Root Cause
Communication failure for system NN polling item. HTTP request receive failed for integration system NN	The integration was not possible due to JIRA unavailability.
Communication failure for system NN polling item. HTTP request receive failed for integration system NN	The integration was not possible due to communication issue.
Connection failed for integration system NN on port NN	There is an issue with the RFC being used for the integration. This can be validated via SM59
Unknown process code & for integration system &	Configuration issue – probably referring to something that does not exist in /BTI/TE_INT_PC
WS call failed for integration system 00 with code 404 not found	There is an issue with the JIRA Query, probably one of the following: • URL is invalid • URL is valid but user has no authority • URL runs but doesn't find expected data e.g. if AC and JIRA are out of sync
AC field [ABCDE] not found for integration system NN polling item 000000000N	Configuration issue. Mismatch in JIRA and ActiveControl, AC field does not exist.
External field & not found for integration system NN polling item NN	Configuration issue. Mismatch in JIRA and ActiveControl, JIRA field does not exist.
No transition to status NNNNN allowed for task {ES-NNNN) in integration system 01.	An unexpected status move was done in JIRA/AC,
Failed to update task NNNNN	Integration was not possible, probably because the AC Business Task was deleted.

/BTI/TE_INTEGRATION 000 & & & &
/BTI/TE_INTEGRATION 002 Connection failed for integration system & on port &
/BTI/TE_INTEGRATION 003 HTTP request send failed for integration system &
/BTI/TE_INTEGRATION 004 HTTP request receive failed for integration system &
/BTI/TE_INTEGRATION 005 WS call failed for integration system & with code & &
/BTI/TE_INTEGRATION 006 Unknown process code & for integration system &
/BTI/TE_INTEGRATION 007 Error building mapping tree for integration system & polling item &
/BTI/TE_INTEGRATION 008 Error building inbound structure for integration system & polling item &
/BTI/TE_INTEGRATION 009 Error deserializing response for integration system & polling item &
/BTI/TE_INTEGRATION 010 Error mapping inbound fields to AC for int. system & polling item &
/BTI/TE_INTEGRATION 011 Trying to map non existing line & of table &
/BTI/TE_INTEGRATION 012 Failed to process task & for integration system & polling item &
/BTI/TE_INTEGRATION 013 Task & skipped for integration system & polling item &

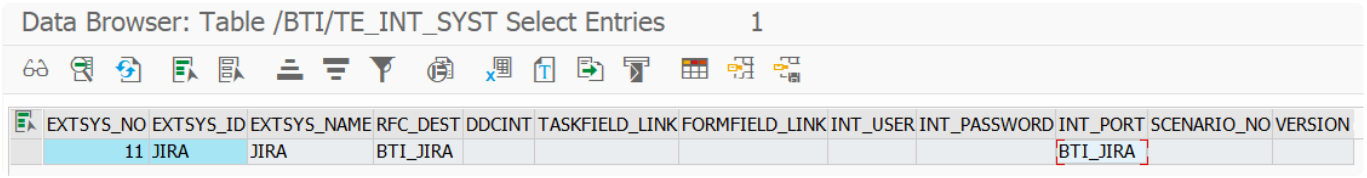
/BTI/TE_INTEGRATION 014 Task & processed successfully for integration system & polling item &
/BTI/TE_INTEGRATION 015 Inconsistent field mapping & treated both as field and structure
/BTI/TE_INTEGRATION 016 WS call failed for integration system & polling item & with code & &
/BTI/TE_INTEGRATION 017 Communication failure for system & polling item &
/BTI/TE_INTEGRATION 018 Creation of task & failed: &
/BTI/TE_INTEGRATION 019 Task & created successfully
/BTI/TE_INTEGRATION 020 Test results entry failed for task & in target & : &
/BTI/TE_INTEGRATION 021 Test results entered for task & in target &
/BTI/TE_INTEGRATION 022 Tests for task & recorded in & locations
/BTI/TE_INTEGRATION 023 & errors approving task & in
/BTI/TE_INTEGRATION 024 No open testseries found for task &
/BTI/TE_INTEGRATION 025 Failed to update task &: &
/BTI/TE_INTEGRATION 026 Task & updated successfully
/BTI/TE_INTEGRATION 027 Starting to poll integration system & item &
/BTI/TE_INTEGRATION 028 Log not initialized subobject & not created for log object /BTI/AC
/BTI/TE_INTEGRATION 029 No transition to status & allowed for task & in integration system &
/BTI/TE_INTEGRATION 030 Transition & might require extra fields for task & in int. system &
/BTI/TE_INTEGRATION 031 AC field & not found for integration system & polling item &
/BTI/TE_INTEGRATION 032 External field & not found for integration system & polling item &
/BTI/TE_INTEGRATION 033 AC table & not found
/BTI/TE_INTEGRATION 034 External table & not found

7.2. Solution 002: Connection failed for integration system [x] on port [x]

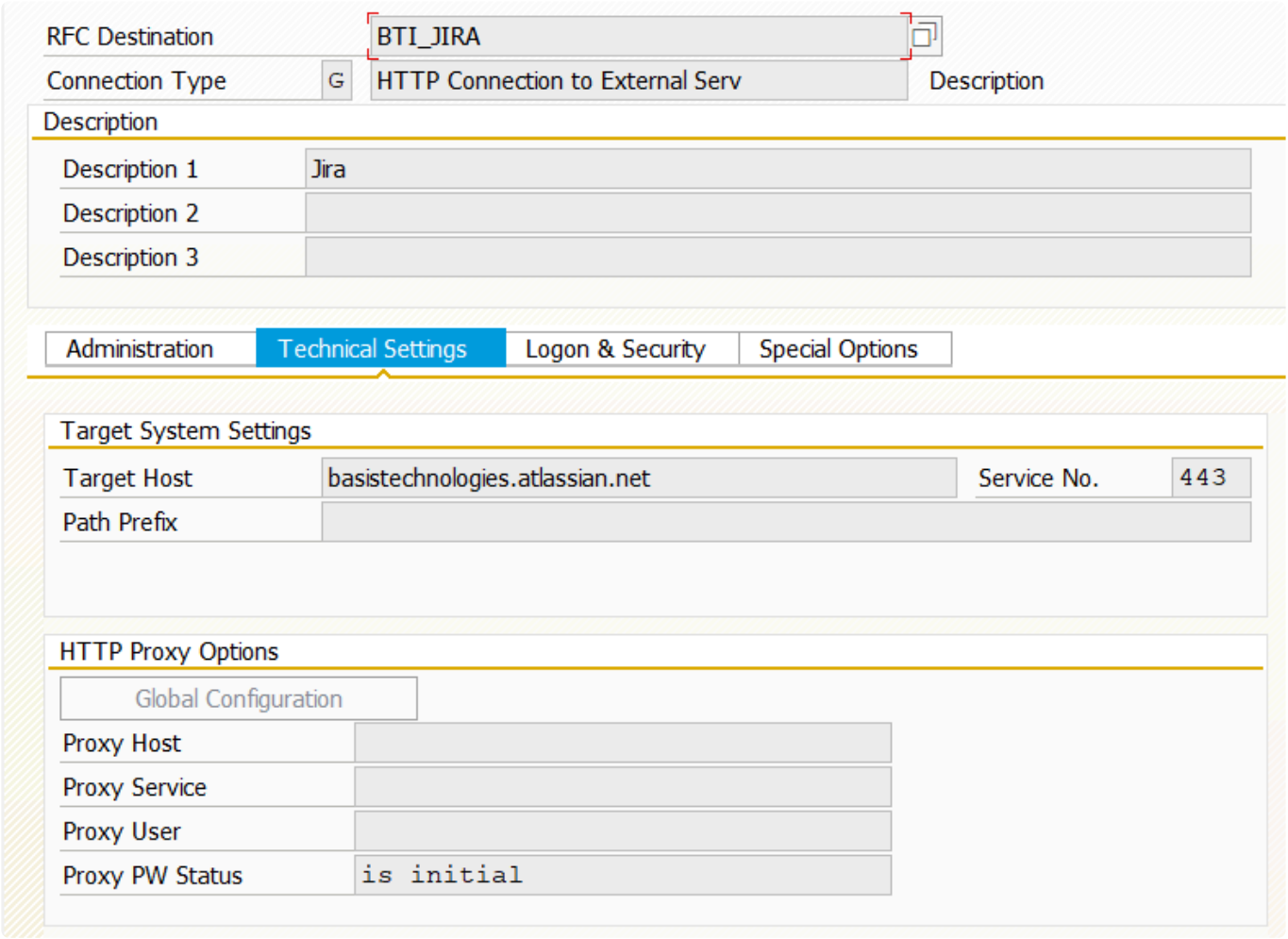
See SLG1 logs for program and error type



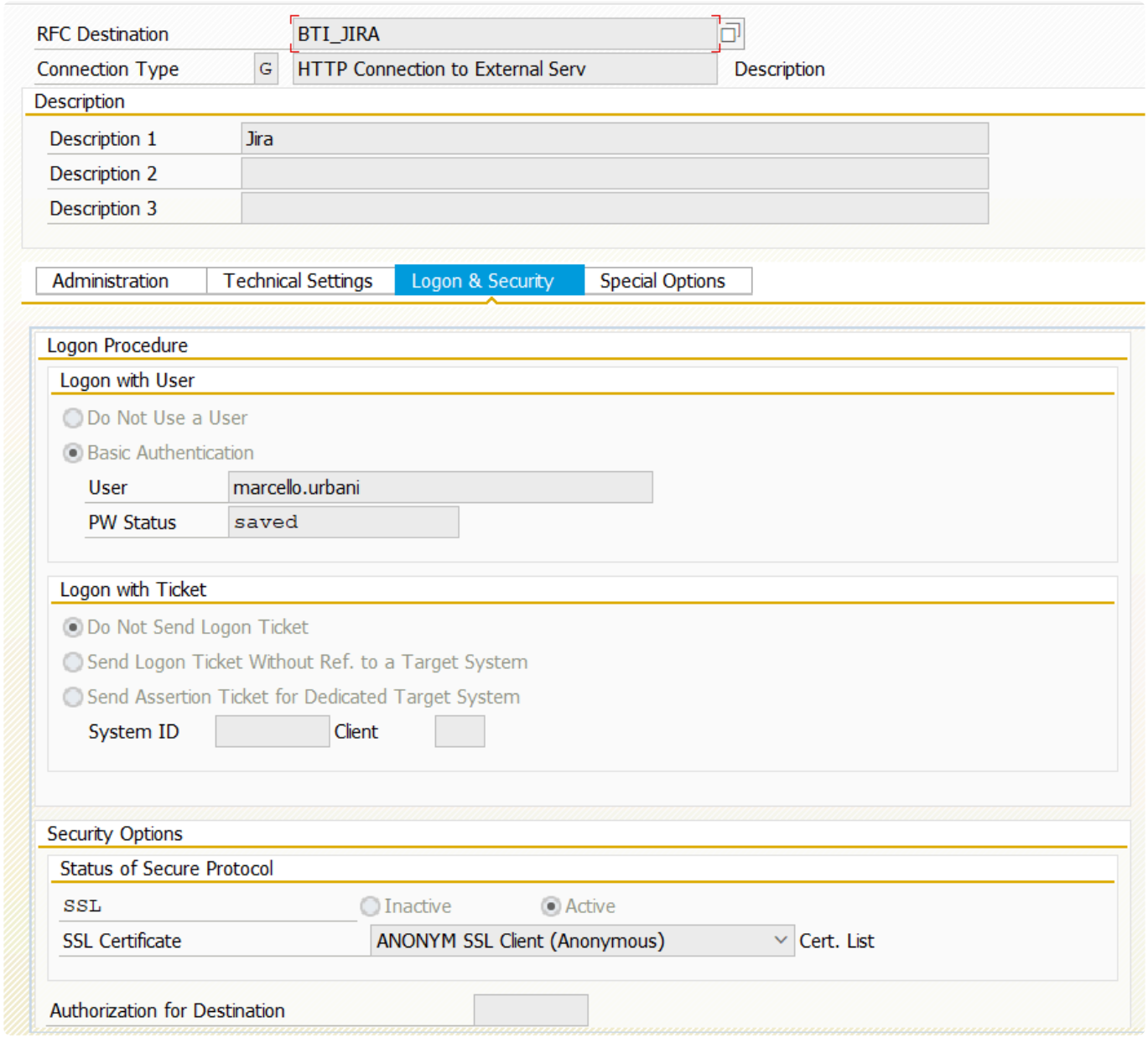
In /BTI/TE_INT_SYST table define the correct INT_PORT.



This port description derives from the HTTP RFC destination created for this integration. See section 4.1 above.

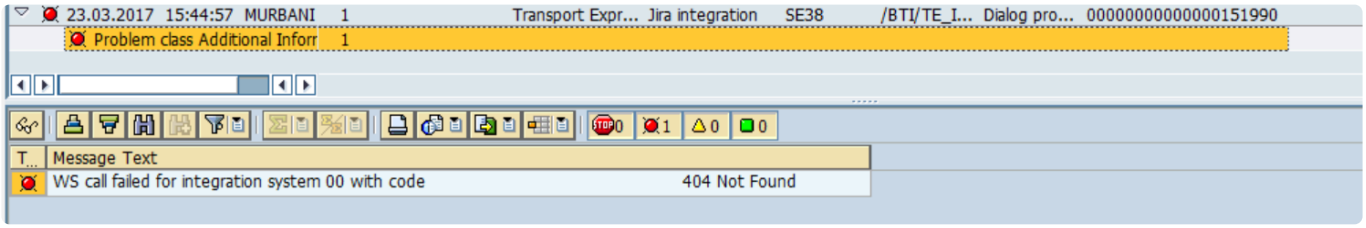


For an integrated HTTP call from SAP to a target ITSM tool you need to define the username and password from the tool for service access and activate SSL security.


RFC Destination
BTI_JIRA
Connection Type
G
HTTP Connection to External Serv
Description
Description
Description 1
Jira
Description 2
Description 3
Administration
Technical Settings
Logon & Security
Special Options
Logon Procedure
Logon with User
Do Not Use a User
Basic Authentication
User
marcello.urbani
PW Status
saved
Logon with Ticket
Do Not Send Logon Ticket
Send Logon Ticket Without Ref. to a Target System
Send Assertion Ticket for Dedicated Target System
System ID
Client
Security Options
Status of Secure Protocol
SSL
Inactive
Active
SSL Certificate
ANONYM SSL Client (Anonymous)
Cert. List
Authorization for Destination

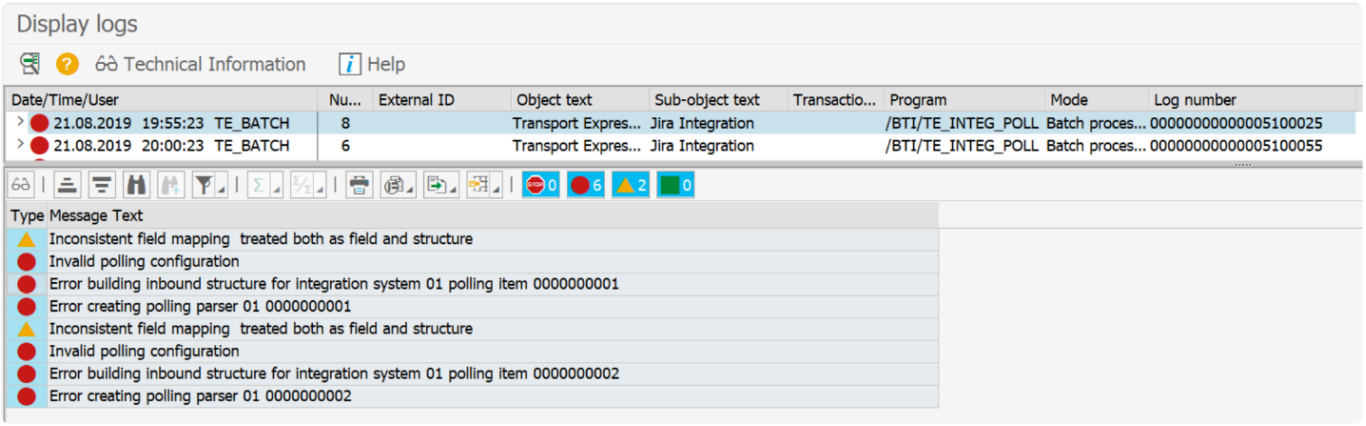
7.3. Solution 005: WS call failed for integration system [X] with code [X] [X]

Message if there is an issue with the JIRA Query. This could be eg
URL is invalid
URL is valid but user has no authority
URL runs but doesn't find expected data – e.g. if AC and JIRA are out of Sync



7.4. Solution 008: Error building inbound structure for integration system [X] polling item [X]

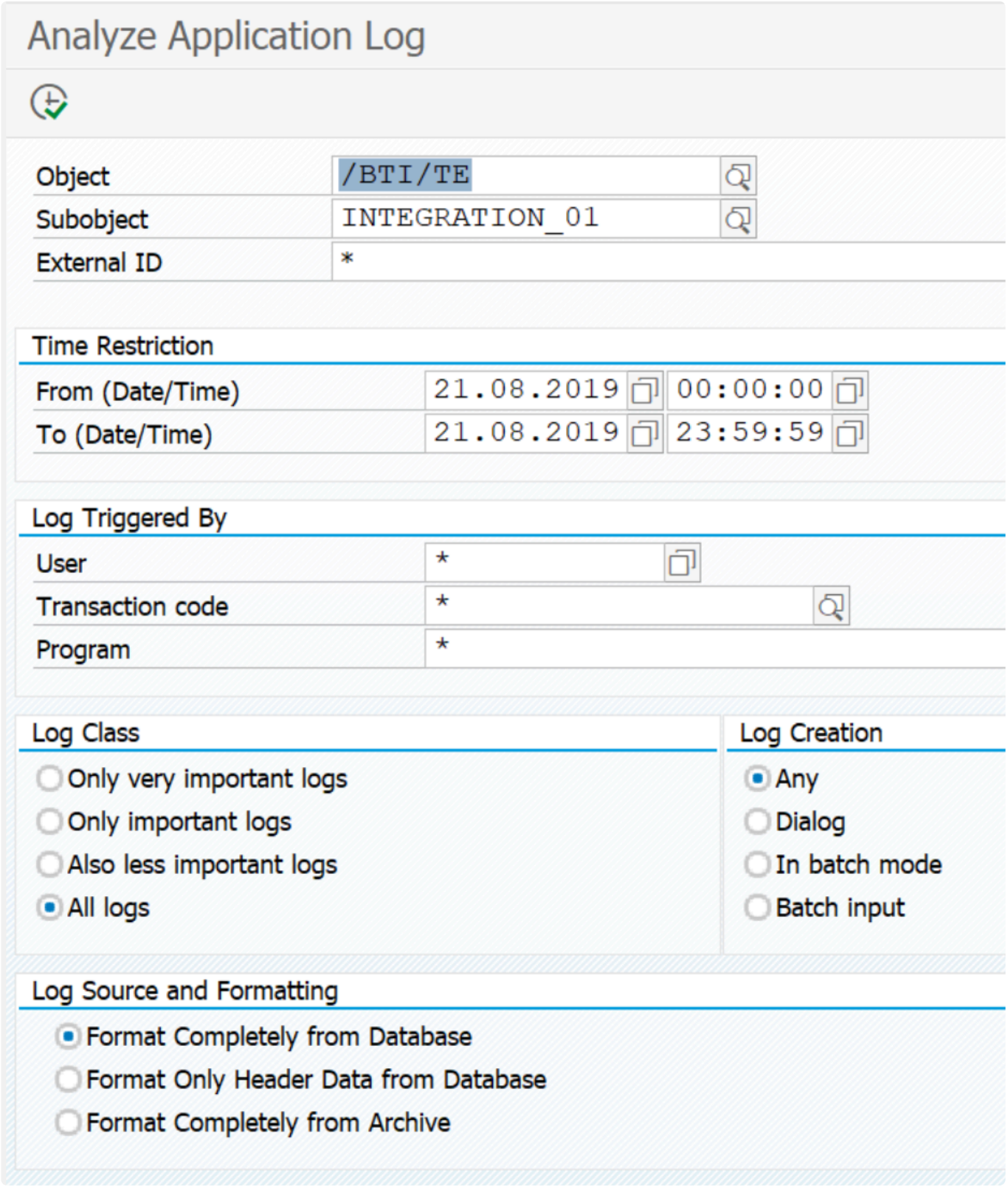
See SLG1 logs for program and error type
In /BTI/TE_INT_POLH table the integration system url defined is incorrect. Reference to item 01 is Create and 02 is Update.

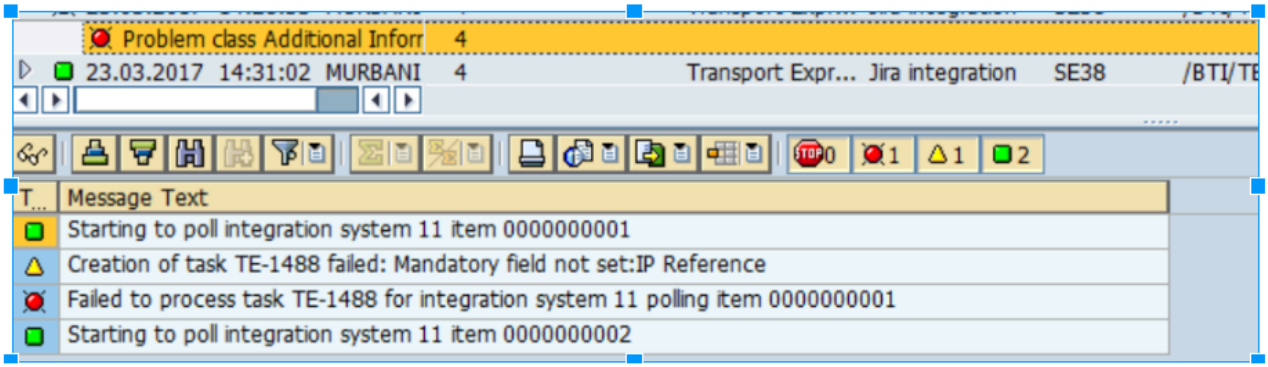


7.5. Solution 012: Failed to process task [X] for integration system [X] polling item &

If a field is marked as mandatory it must be mapped if not then you will get error in the error in SLG1

- Mandatory field and mapped – no error
- Not Mandatory – no error
- Mandatory and not mapped -error




Problem class Additional Inforr 4
23.03.2017 14:31:02 MURBANI 4 Transport Expr... Jira integration SE38 /BTI/TE
0 1 1 2
T... Message Text
Starting to poll integration system 11 item 0000000001
Creation of task TE-1488 failed: Mandatory field not set:IP Reference
Failed to process task TE-1488 for integration system 11 polling item 0000000001
Starting to poll integration system 11 item 0000000002

7.6. Solution 016: WS call failed for integration system [X] polling item [X] with code [X] [X]

Check RFC Destination for correct Target Host, Service No. and Path Prefix (proxy), ensure a target system username and password is defined, SEE 4.1.3 RFC Destination

Check /BTI/TE_INT_SYST that correct RFC Destination is defined, SEE 4.1.2 External System(s)

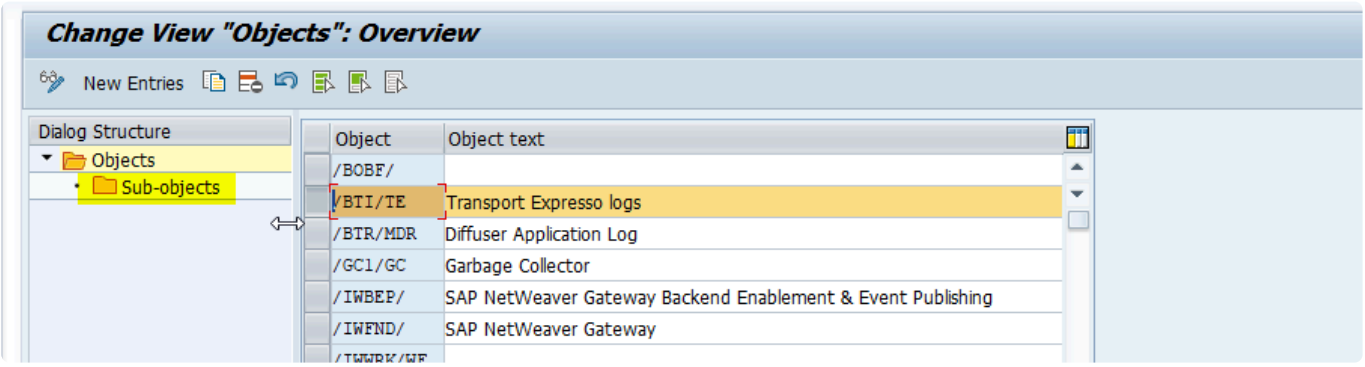
Check /BTI/TE_INT_POLH that the url points to Jira API, e.g. use Jira’s REST API (not /issues/?filter=10600, but /rest/api/2/search?jql=filter=10600), SEE 4.2.2 AC Polling Header

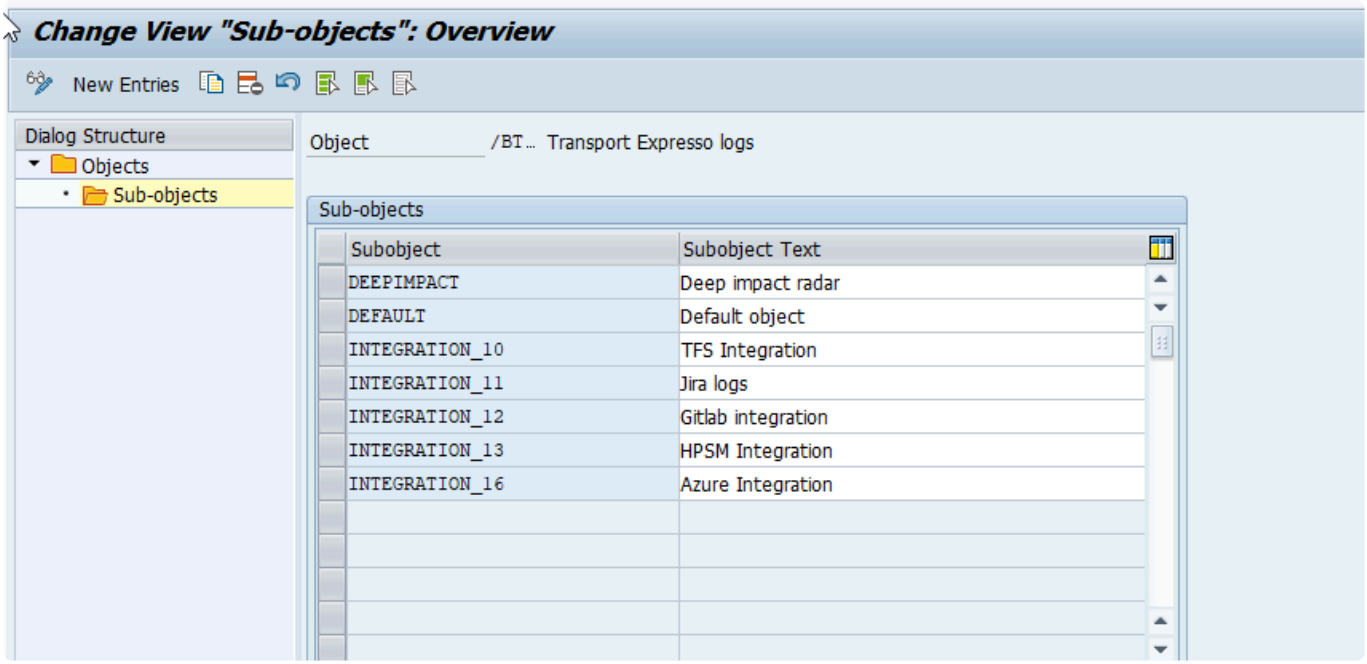
7.7. Solution 017: Communication failure for system [X] polling item

The integration was not possible due to JIRA unavailability.

7.8. Solution 028: Log not initialized sub-object INTEGRATION_01 not created for log object /BTI/TE

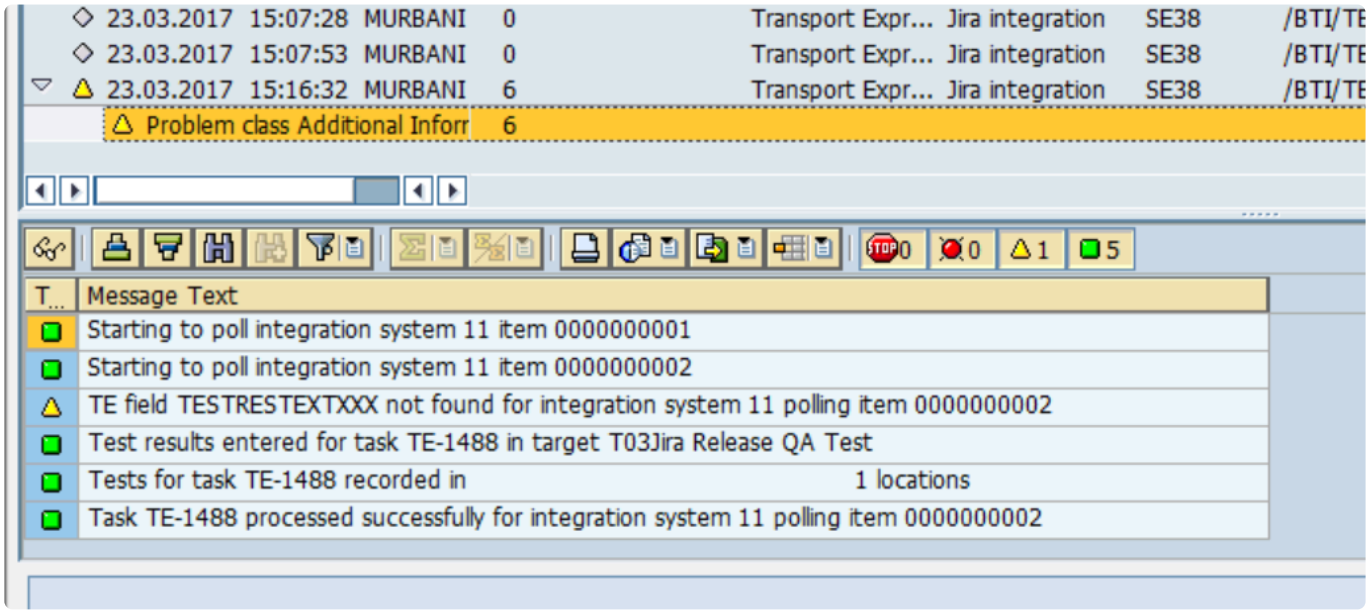
Go to txn SLG0, Choose /BTI/TE and click on sub-objects. Then click on new entry and create sub-object INTEGRATION_01 and save it.





7.9. Solution 031: TE field & not found for integration system & polling item &

Fields in /BTI/TE_INT_POLF – TEFIELDREF – must be valid

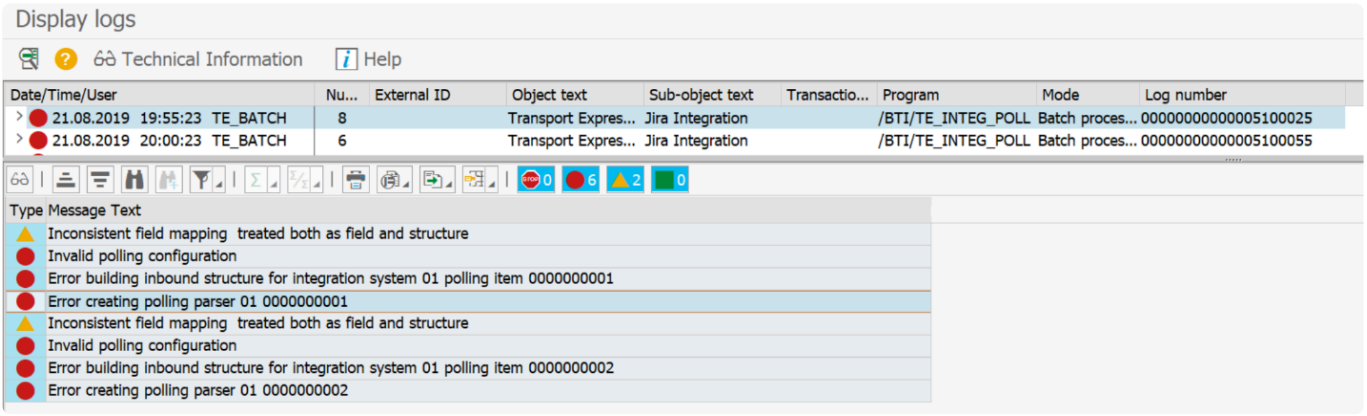


Data Browser: Table /BTI/TE_INT_POLF Select Entries 17

EXTSYS_NO	SEQUENCE_NO	EXTERNAL_REF	TEFIELDREF	DATAFORMAT
11	1	changelog-histories[1]-created	HEADER-CF_511	T
11	1	fields-assignee-emailAddress	HEADER-CF_550	
11	1	fields-assignee-name	HEADER-TESTERID	
11	1	fields-created	TIMESTAMP	T
11	1	fields-description	DESCRIPTION	
11	1	fields-issuetype-id	HEADER-TYPEID	
11	1	fields-project-key	HEADER-PROJECTID	
11	1	fields-summary	HEADER-CAPTION	
11	1	key	HEADER-REFERENCE	
11	2	0	TESTRESULT	C
11	2	10010050500000000002	TARGETROLE	C
11	2	X	APPROVE	C
11	2	changelog-histories[]-created	CHANGES[]-TIMESTAMP	T
11	2	changelog-histories[]-items[]-field	CHANGES[]-ITEMS[]-FIELD	
11	2	fields-resolution-description	TESTRESTEXT	
11	2	fields-resolution-description	TESTRESTEXTXXX	
11	2	key	HEADER-REFERENCE	

7.10. Error creating polling parser 01 0000000001

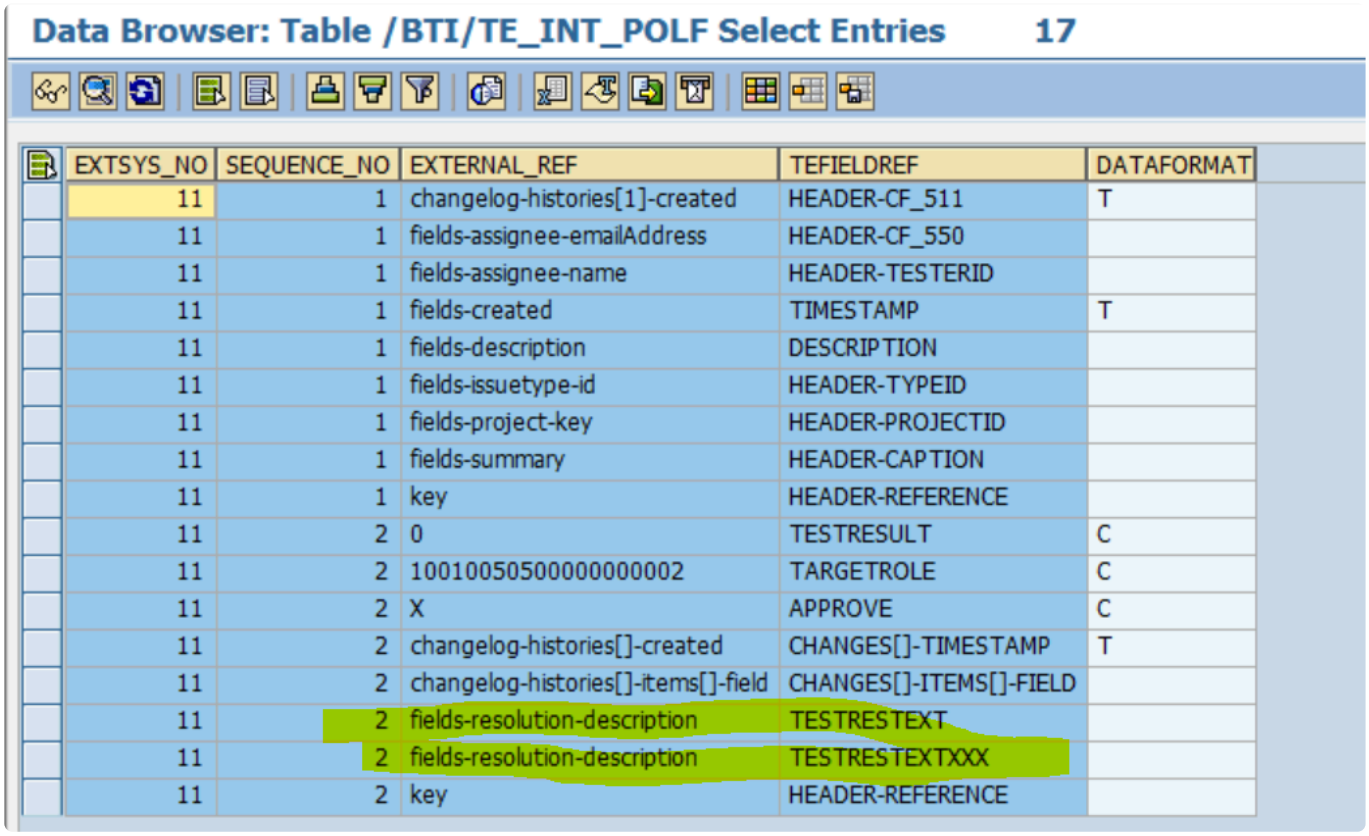
In /BTI/TE_INT_POLF table the field mapping details are incorrect. See section 4.2.3.



7.11. Testing Scenarios

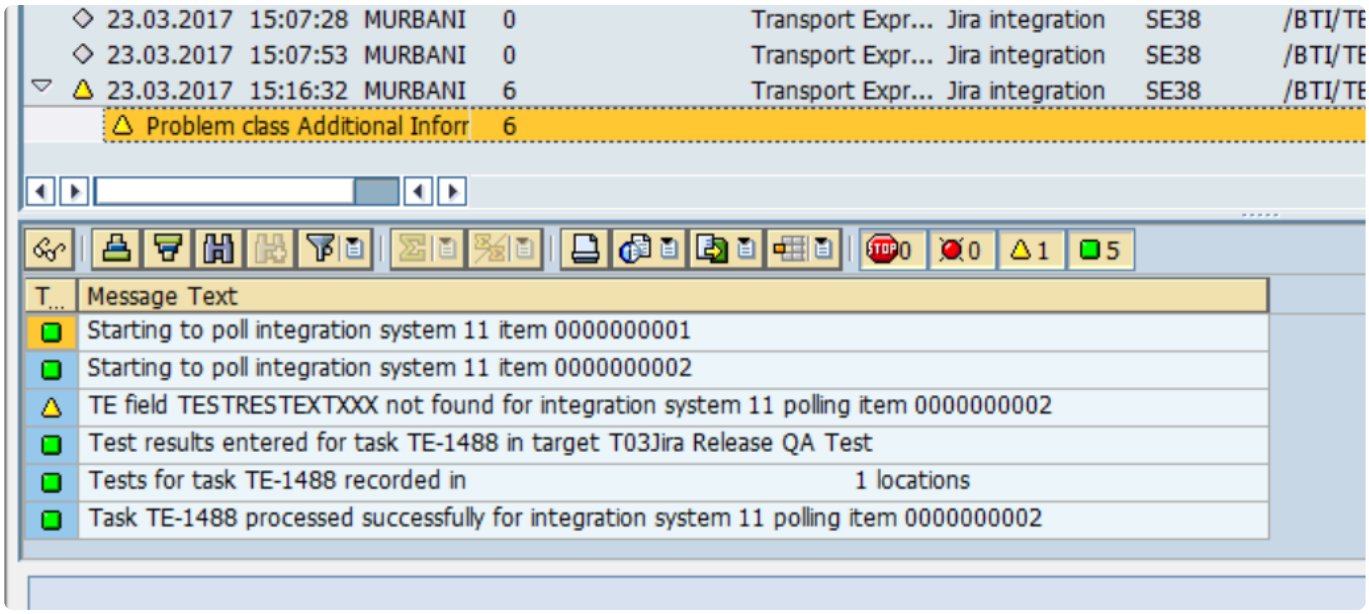
7.11.1. Config Errors

Fields in /BTI/TE_INT_POLF – TEFIELDREF – must be valid



Data Browser: Table /BTI/TE_INT_POLF Select Entries 17

EXTSYS_NO	SEQUENCE_NO	EXTERNAL_REF	TEFIELDREF	DATAFORMAT
11	1	changelog-histories[1]-created	HEADER-CF_511	T
11	1	fields-assignee-emailAddress	HEADER-CF_550	
11	1	fields-assignee-name	HEADER-TESTERID	
11	1	fields-created	TIMESTAMP	T
11	1	fields-description	DESCRIPTION	
11	1	fields-issuetype-id	HEADER-TYPEID	
11	1	fields-project-key	HEADER-PROJECTID	
11	1	fields-summary	HEADER-CAPTION	
11	1	key	HEADER-REFERENCE	
11	2	0	TESTRESULT	C
11	2	10010050500000000002	TARGETROLE	C
11	2	X	APPROVE	C
11	2	changelog-histories[]-created	CHANGES[]-TIMESTAMP	T
11	2	changelog-histories[]-items[]-field	CHANGES[]-ITEMS[]-FIELD	
11	2	fields-resolution-description	TESTRESTEXT	
11	2	fields-resolution-description	TESTRESTEXTXXX	
11	2	key	HEADER-REFERENCE	

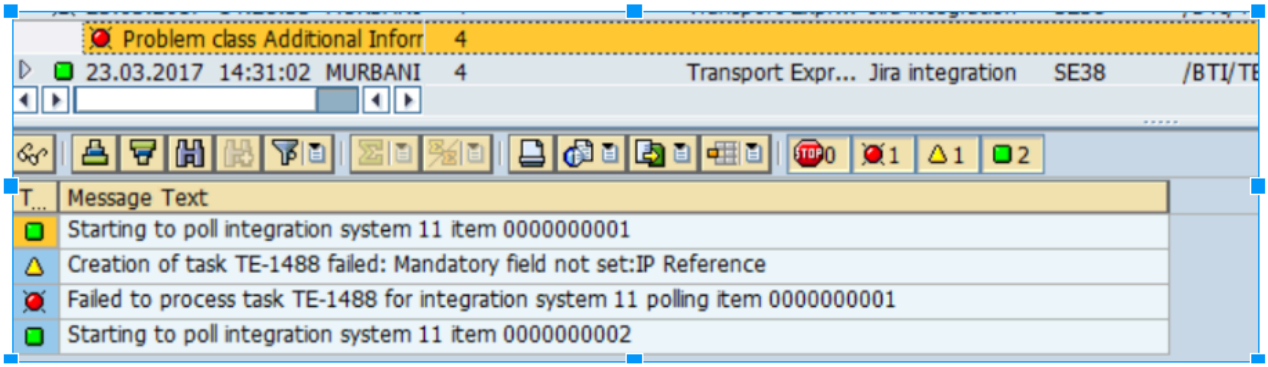


23.03.2017	15:07:28	MURBANI	0	Transport Expr...	Jira integration	SE38	/BTI/TE
23.03.2017	15:07:53	MURBANI	0	Transport Expr...	Jira integration	SE38	/BTI/TE
23.03.2017	15:16:32	MURBANI	6	Transport Expr...	Jira integration	SE38	/BTI/TE
Problem class Additional Inforr			6				

T...	Message Text
	Starting to poll integration system 11 item 0000000001
	Starting to poll integration system 11 item 0000000002
	TE field TESTRESTEXTXXX not found for integration system 11 polling item 0000000002
	Test results entered for task TE-1488 in target T03Jira Release QA Test
	Tests for task TE-1488 recorded in 1 locations
	Task TE-1488 processed successfully for integration system 11 polling item 0000000002

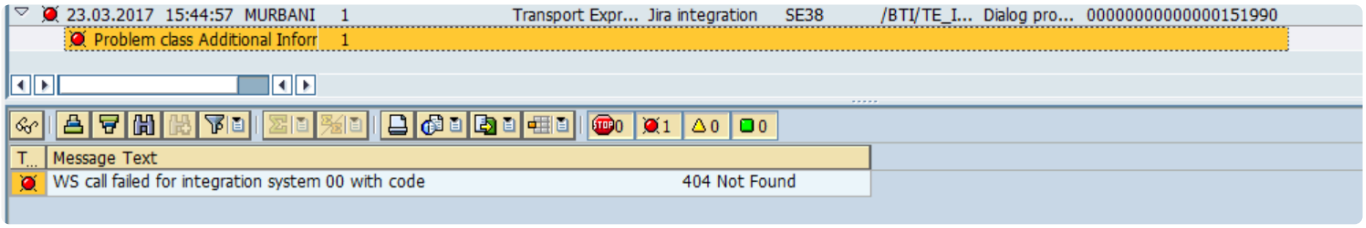
If a field is marked as mandatory field it must be mapped if not then you will get error in the error in SLG1

- Mandatory field and mapped – no error
- Not Mandatory – no error
- Mandatory and not mapped -error


Problem class Additional Inforr 4
23.03.2017 14:31:02 MURBANI 4 Transport Expr... Jira integration SE38 /BTI/TE
0 1 1 2
T... Message Text
Starting to poll integration system 11 item 0000000001
Creation of task TE-1488 failed: Mandatory field not set:IP Reference
Failed to process task TE-1488 for integration system 11 polling item 0000000001
Starting to poll integration system 11 item 0000000002

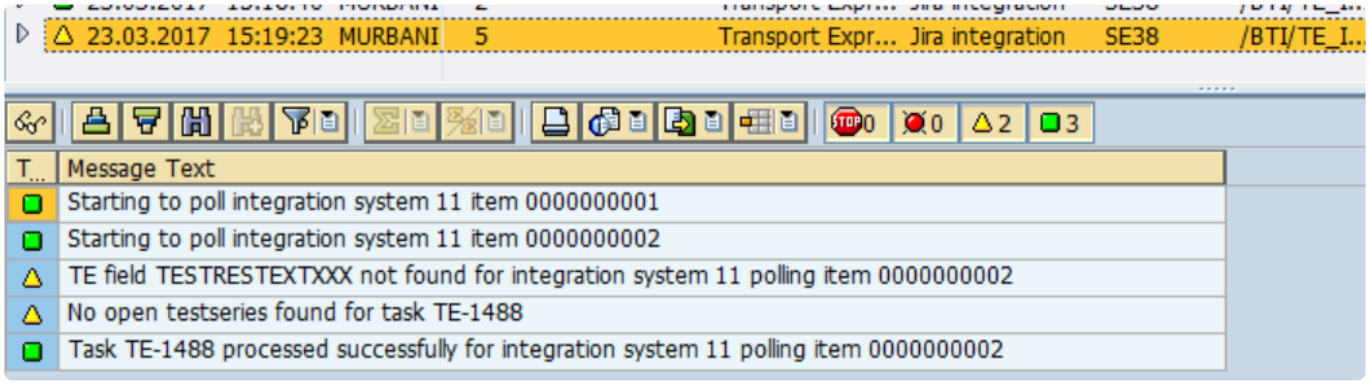
7.11.2. JIRA Query Error

Message if there is an issue with the JIRA Query. This could be eg
URL is invalid
URL is valid but user has no authority
URL runs but doesnt find expected data – eg if AC and JIRA are out of Sync



7.11.3. JIRA AC out of sync

Jira moves to QA in Progress
AC is in the outbox but not in test queue
Mark JIRA as done
Query 2 will pick up the item but an do nothing with it



7.12. System Unavailability

There may be occasion where the Integration is not possible due to system downtime on either ActiveControl or JIRA side.

ActiveControl downtime
If the AC Domain Controller is down or unavailable for whatever reason, when the Domain Controller becomes available again the next run of the integration jobs will perform a catch up of all required activity since the last time the jobs were run.

No manual intervention relating to the Integration should be required of the AC Administrator after AC downtime; as long as the Integration jobs detailed earlier in this document are running, then the catch up will occur automatically.

JIRA downtime
If JIRA is down, then Integration will fail. This will be notified to the configured users.

A new report is available from ActiveControl 7.0 onwards, that lists the failed entries and provide the facility reset their number of retries counter so that the items can be picked up for Sending at the next run.

This report is called via transaction /n/BTI/INTTEG_RESET_ER in the Domain Controller.

Further details of this Report are detailed in this Knowledge Article