Integrations (via Middleware)

APIs / Webhooks — Last update: 28 June 2023

Basis Technologies

Copyright © 2023 Basis Technologies

Table of Contents

1. Introduction 4

2. ActiveControl – Key Concepts 7

2.1. Domain Controller 8

2.2. Inbound / Outbound 9

3. Integration Pre-requisites 10

3.1. SAP User 11

4. SOAP APIs 12

4.1. WSDL 13

4.2. Webservices 14

4.2.1. Create a business task 15

4.2.2. Change a business task 17

4.2.3. Read a business task 19

4.2.4. Start the analysis for a business task 21

4.2.5. Read the results of an analysis run 22

4.2.6. Approve a Business Task 25

4.2.7. Enter the test results for a business task 26

4.2.8. Mapping Internal Values 27

4.2.9. Other Communication Techniques 29

5. RESTful APIs 30

6. HTTP APIs 31

6.1. Get Queue Contents 32

6.2. Lock Queue 38

6.3. Unlock Queue 40

6.4. Save Business Task Test Result 42

7. Web Hooks 45

7.1. Deployment Status Webhook 46

7.1.1. Configuration Steps 48

7.1.1.1. RFC Destination 49

7.1.1.2. Table: /BTI/TE_INT_SYST 51

7.1.1.3. Table: /BTI/TE_INT_CLAS 52

7.1.1.4. Table: /BTI/TE_TVARV 53

7.1.1.5. Table: /BTI/TE_INT_MAPP 54

7.1.1.6. Table: /BTI/TE_INT_PROC 55

7.1.1.7. Table: /BTI/TE_INT_CONV 56

7.1.1.8. Create sub-object INTEGRATION_nn 57

7.1.1.9. Program: /BTI/TE_INTEG_TRIGGER 58

7.1.1.10. Program: /BTI/TE_INTEG_SEND 59
7.2. Debugging the JSON 60

8. Support of Custom Integrations 61

9. Further Information? 62

1. Introduction

As of ActiveControl 8.40, there are essentially three main ways of integrating ActiveControl with other tools:

1. Using the **SAP Integration Framework** that has been available within ActiveControl for the past decade.

2. Using the **Node-Red Integration Framework**, introduced in ActiveControl 8.40 (2021)

3. Utilising standalone **APIs and webhooks**.

Although the focus of this document is (3) – all three will now be summarised in turn.

SAP Integration Framework.

ActiveControl has long offered an Integration Framework – based on SAP ABAP – to enable integration between ActiveControl and 3rd Party tools used by our customers. The following integrations are offered.

Product	Type	Documentation Link
ServiceNow	ITSM	Integration Guide
JIRA	ITSM	Integration Guide
Cherwell	ITSM	Integration Guide available on request
Team Foundation Server	ITSM	Integration Guide
ChaRM	ITSM	Integration Guide
HPSM	Manual Testing	Integration Guide
smartShift	S/4 migrations	Integration Guide
Jenkins	AUtomated Testing	Integration Guide
GitLab	CI/CD	Integration Guide
abapGit	Development	Integration Guide

Of these integrations, the ServiceNow and JIRA integrations are the most mature and well-established at Basis Technologies customers. The others were built to fulfil specific Customer requirements, and are less widely utilised within Basis Technologies customer-base.

Node-RED Integration framework

In 2021,a new Node-Red integration Framework was introduced as part of ActiveControl 8.40. Node-RED is a flow-based development tool for visual programming, for wiring together hardware devices, APIs and online services. Basis Technologies believe that using Node-RED as an integration middleware between ActiveControl and 3rd Party products will enable new integrations to be setup much more easily and rapidly than we have been able to do in the past. Basis Technologies intend to replace the use of the SAP Integration Framework with this new Node-RED integration framework over time.

As ‘templates’ for new integrations are created by Basis Technologies, Integration Guides will become available via https://docs.basistechnologies.com/

Product	Type	Documentation Link
Azure DevOps	ITSM / DevOps	In Progress
JIRA	ITSM	In Progress

SOAP APIs, webservices and webhooks.

There may be instances where Basis Technologies may want to build their own integrations between ActiveControl and other tools, either because Basis Technologies does not already offer an out-of-the-box integration via either our SAP or Node-RED integration frameworks – or because the Customer prefers to use their own middleware such as Tibco, PI etc – or because the tool is proprietary or non-strategic and Basis Technologies do not have the capacity to develop the Integration for our Customers.

This can be achieved using APIs, webservices and webhooks that are available as part of ActiveControl.

The purpose of this document is to detail what is available as part of ActiveControl, for customers that do want to explore this option.

Intended Audience

The intended audience for this document are the technical teams looking to implement integration between ActiveControl and third party tools – but not using our existing SAP and Node-RED integration frameworks.

The document assumes prior knowledge of ActiveControl and also general Integration principles.

Need Support?

Basis Technologies effort to develop or setup Integrations using our SAP or Node-Red Integration Frameworks is a chargeable activity. Similarly, providing Customers with assistance in setting up their own integrations using our APIs, webservices or webhooks is chargeable consulting rather than a maintenance support activity. If a Customer is interested in setting up an Integration and requires Basis Technologies assistance, then they should reach out to their Account Manager in the first instance to discuss options of how best to proceed.

2. ActiveControl – Key Concepts

2.1. 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.

More information on ActiveControl can be found in the seperate Administration Guide.

ActiveControl Domain Controller

2.2. Inbound / Outbound

A typical Integration between ActiveControl and another tool will consist of an Inbound and Outbound component.

Inbound Integration

An inbound integration to ActiveControl will typically consist of such activities as:

1) Automatically creating the Business Task within ActiveControl when the corresponding 3rd Party tool reaches a certain status.
2) Changing a Business Task within ActiveControl as the corresponding 3rd Party tool changes.
2) Signing off a Test Queue, based on a status update made in a 3rd Party tool.
3) Signing off a Test Queue, based on PASS/FAIL information sent from a 3rd Party testing tool.
4) Approving an Inbox or Outbox in ActiveControl, based on a status update made in a 3rd Party tool.

Outbound Integration

An outbound integration from ActiveControl will typically consist of such activities as:

1) Updating a 3rd Party tool status, based on a status (ie Business Task Deployment Status) change within ActiveControl.
2) Reading information from the ActiveControl Business Task, to update a 3rd Part tool. (eg Transport list, Analysis information).

3. Integration Pre-requisites

3.1. SAP User

A SAP user is required within SAP, to support inbound communication with ActiveControl.

The Basis Technologies recommendation is to create a new, seperate Integration user for such purposes (eg username AC_INTEG) to keep it seperate from the other AC_RFC and AC_BATCH users needed as part of an ActiveControl setup.

This integration user should be a SERVICE user with /BTI/TE:CTS_ADMIN and /BTI/TE:CTS_USER role assignments.

4. SOAP APIs

ActiveControl offers various out of the box APIs via which ActiveControl can communicate with 3rd Party tools

As of ActiveControl 9.00, SOAP APIs are available to perform all of the following key functions (however the intention is to supplement these with RESTful APIs in a future relesae of ActiveControl):

- Create a Business Task
- Change a Business Task
- Read a Business Task
- Read the Transports of a Business Task
- Start the analysis for a Business Task
- Read the results of an Analysis run
- Approve a Business task
- Enter the test results for a Business Task

4.1. WSDL

The technical definitions are available as WSDL both for the main definition and test endpoint .

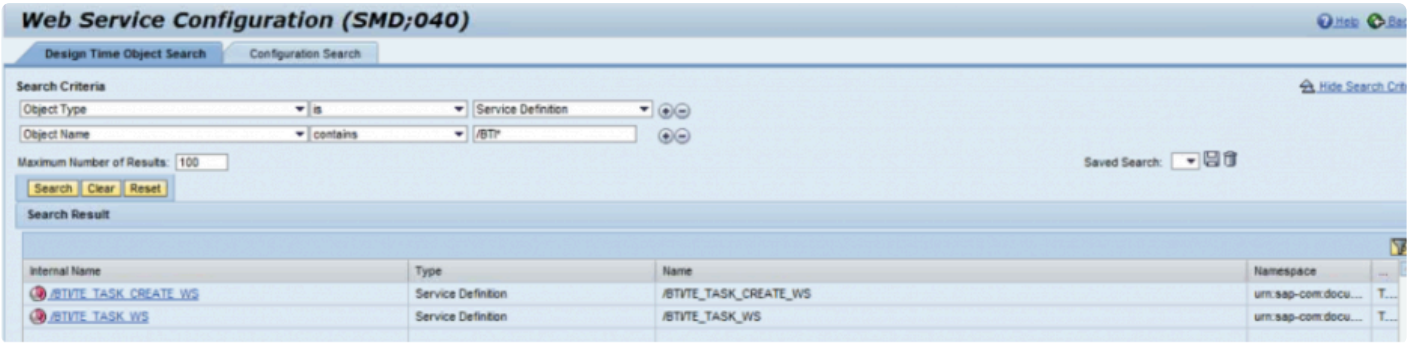
Port type _-BTI_-TE_TASK_WS

The documentation in the rest of this section, which details every action available, has been generated from WSDL.

Obtaining WSDL

Integration WSDL used for inbound integration can be generated from the Customer ActiveControl Controller via transaction SOAMANAGER.

/BTI/TE_TASK_WS is to be configured.



4.2. Webservices

4.2.1. Create a business task

Allows a new business task to be created in ActiveControl with the details specified.

-bti-teTaskCreateWs

Input:

-bti-teTaskCreateWs

parameters type _-bti_-teTaskCreateWs

- **XDescription – optional; type string**
- **XTask type _-bti_-teStXtask**
  Id type char20 – type string with restriction maxLength(20)
  Caption type char100 – type string with restriction maxLength(100)
  Reference type char20 – type string with restriction maxLength(20)
  Groupid type char20 – type string with restriction maxLength(20)
  Typeid type char20 – type string with restriction maxLength(20)
  Testerid type char20 – type string with restriction maxLength(20)
  Priority type char1 – type string with restriction maxLength(1)
  Projectid type char20 – type string with restriction maxLength(20)
  Locked type char1 – type string with restriction maxLength(1)
  Path type numeric2 – type string with restriction maxLength(2) pattern(\d*)
  StatDepl type char20 – type string with restriction maxLength(20)
  StatPlan type char20 – type string with restriction maxLength(20)
  StatDeplMan type char1 – type string with restriction maxLength(1)
  StatPlanMan type char1 – type string with restriction maxLength(1)
- **XtCustfields – optional; type _-bti_-teTextfieldTt**
  item – optional, unbounded; type _-bti_-teTextfield
  Id type char3 – type string with restriction maxLength(3)
  Value type string
- **XtTesters – optional; type _-bti_-teTtXtasktest**
  item – optional, unbounded; type _-bti_-teStXtasktest
  Testerid type char20 – type string with restriction maxLength(20)
  Targetroleid type char20 – type string with restriction maxLength(20)
  Targetid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Systemid type char8 – type string with restriction maxLength(8)
  Mandt type clnt3 – type string with restriction maxLength(3)
  SmtpAddr type char241 – type string with restriction maxLength(241)
  Testername type char30 – type string with restriction maxLength(30)

Output:

-bti-teTaskCreateWsResponse

parameter type _-bti_-teTaskCreateWsResponse

- **YReturn type _-bti_-teStXretrn**
  Msgtyp type char1 – type string with restriction maxLength(1)
  Msgid type char20 – type string with restriction maxLength(20)
  Msgnum type numeric3 – type string with restriction maxLength(3) pattern(\d*)
  Message type char220 – type string with restriction maxLength(220)
  Msgv1 type char50 – type string with restriction maxLength(50)
  Msgv2 type char50 – type string with restriction maxLength(50)
  Msgv3 type char50 – type string with restriction maxLength(50)
  Msgv4 type char50 – type string with restriction maxLength(50)
  Exception type char50 – type string with restriction maxLength(50)
- **YTask type _-bti_-teTask**
  Id type char20 – type string with restriction maxLength(20)
  Caption type char100 – type string with restriction maxLength(100)
  Reference type char20 – type string with restriction maxLength(20)
  Refsearch type char20 – type string with restriction maxLength(20)
  Groupid type char20 – type string with restriction maxLength(20)
  Typeid type char20 – type string with restriction maxLength(20)
  Testerid type char20 – type string with restriction maxLength(20)
  Priority type char1 – type string with restriction maxLength(1)
  Locked type char1 – type string with restriction maxLength(1)
  StatDepl type char20 – type string with restriction maxLength(20)
  StatPlan type char20 – type string with restriction maxLength(20)
  StatDeplMan type char1 – type string with restriction maxLength(1)
  StatPlanMan type char1 – type string with restriction maxLength(1)

4.2.2. Change a business task

Allows an existing business task to be changed in ActiveControl with the details specified.

-bti-teTaskChangeWs

The various XUpdate… fields need to be set to ‘X’ for the corresponding data to be taken into account.
Input:

-bti-teTaskChangeWs

parameters type _-bti_-teTaskChangeWs

- **XDescription – optional; type string**
- **XTask type _-bti_-teStXtask**
  Id type char20 – type string with restriction maxLength(20)
  Caption type char100 – type string with restriction maxLength(100)
  Reference type char20 – type string with restriction maxLength(20)
  Groupid type char20 – type string with restriction maxLength(20)
  Typeid type char20 – type string with restriction maxLength(20)
  Testerid type char20 – type string with restriction maxLength(20)
  Priority type char1 – type string with restriction maxLength(1)
  Projectid type char20 – type string with restriction maxLength(20)
  Locked type char1 – type string with restriction maxLength(1)
  Path type numeric2 – type string with restriction maxLength(2) pattern(\d*)
  StatDepl type char20 – type string with restriction maxLength(20)
  StatPlan type char20 – type string with restriction maxLength(20)
  StatDeplMan type char1 – type string with restriction maxLength(1)
  StatPlanMan type char1 – type string with restriction maxLength(1)
- **XUpdateCustfields type char1 – type string with restriction maxLength(1)**
- **XUpdateDesc type char1 – type string with restriction maxLength(1)**
- **XUpdateTesters type char1 – type string with restriction maxLength(1)**
- **XUpddateTask – optional; type char1 – type string with restriction maxLength(1)**
- **XtCustfields – optional; type _-bti_-teTextfieldTt**
  item – optional, unbounded; type _-bti_-teTextfield
  Id type char3 – type string with restriction maxLength(3)
  Value type string
- **XtTesters – optional; type _-bti_-teTtXtasktest**
  item – optional, unbounded; type _-bti_-teStXtasktest
  Testerid type char20 – type string with restriction maxLength(20)
  Targetroleid type char20 – type string with restriction maxLength(20)
  Targetid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Systemid type char8 – type string with restriction maxLength(8)
  Mandt type clnt3 – type string with restriction maxLength(3)

SmtpAddr type char241 – type string with restriction maxLength(241)
Testername type char30 – type string with restriction maxLength(30)

Output:

-bti-teTaskChangeWsResponse

parameter type _-bti_-teTaskChangeWsResponse

- **YReturn type _-bti_-teStXretrn**
  Msgtyp type char1 – type string with restriction maxLength(1)
  Msgid type char20 – type string with restriction maxLength(20)
  Msgnum type numeric3 – type string with restriction maxLength(3) pattern(\d*)
  Message type char220 – type string with restriction maxLength(220)
  Msgv1 type char50 – type string with restriction maxLength(50)
  Msgv2 type char50 – type string with restriction maxLength(50)
  Msgv3 type char50 – type string with restriction maxLength(50)
  Msgv4 type char50 – type string with restriction maxLength(50)
  Exception type char50 – type string with restriction maxLength(50)

4.2.3. Read a business task

Allows an existing business task in ActiveControl to be read to obtain the details.

-bti-teTaskReadWs

Input:

-bti-teTaskReadWs

parameters type _-bti_-teTaskReadWs

- **XTaskid type char20 – type string with restriction maxLength(20)**

Output:

-bti-teTaskReadWsResponseSource code

parameter type _-bti_-teTaskReadWsResponse

- **YDescription type string**
- **YTask type _-bti_-teStXtask**
  Id type char20 – type string with restriction maxLength(20)
  Caption type char100 – type string with restriction maxLength(100)
  Reference type char20 – type string with restriction maxLength(20)
  Groupid type char20 – type string with restriction maxLength(20)
  Typeid type char20 – type string with restriction maxLength(20)
  Testerid type char20 – type string with restriction maxLength(20)
  Priority type char1 – type string with restriction maxLength(1)
  Projectid type char20 – type string with restriction maxLength(20)
  Locked type char1 – type string with restriction maxLength(1)
  Path type numeric2 – type string with restriction maxLength(2) pattern(\d*)
  StatDepl type char20 – type string with restriction maxLength(20)
  StatPlan type char20 – type string with restriction maxLength(20)
  StatDeplMan type char1 – type string with restriction maxLength(1)
  StatPlanMan type char1 – type string with restriction maxLength(1)
- **YtCustfields type _-bti_-teTextfieldTt**
  item – optional, unbounded; type _-bti_-teTextfield
  Id type char3 – type string with restriction maxLength(3)
  Value type string
- **YtTesters type _-bti_-teTtXtasktest**
  item – optional, unbounded; type _-bti_-teStXtasktest
  Testerid type char20 – type string with restriction maxLength(20)
  Targetroleid type char20 – type string with restriction maxLength(20)

Targetid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
Systemid type char8 – type string with restriction maxLength(8)
Mandt type clnt3 – type string with restriction maxLength(3)
SmtpAddr type char241 – type string with restriction maxLength(241)
Testername type char30 – type string with restriction maxLength(30)

4.2.4. Start the analysis for a business task

Starts the ActiveControl analysis process for a specific business task.

The status and results of the analysis can be queried later via the “Read the results of an analysis run“ web service using the analysis ID returned so it can be determined if it safe to approve.

-bti-teTaskAnalyseWs

Input:

-bti-teTaskAnalyseWs

parameters type _-bti_-teTaskAnalyseWs

- **XAnltypeid – optional; type numeric4 – type string with restriction maxLength(4) pattern(\d)***
- **XLocation type char1 – type string with restriction maxLength(1)**
- **XTarget type numeric4 – type string with restriction maxLength(4) pattern(\d)***
- **XTaskid type char20 – type string with restriction maxLength(20)**
- **XtRequest type _-bti_-teTtRfc010**
- **item – optional, unbounded; type _-bti_-teStRfc010**
- **Trkorr type char20 – type string with restriction maxLength(20)**

Output:

-bti-teTaskAnalyseWsResponse

parameter type _-bti_-teTaskAnalyseWsResponse

- **XtRequest type _-bti_-teTtRfc010**
  item – optional, unbounded; type _-bti_-teStRfc010
  Trkorr type char20 – type string with restriction maxLength(20)
- **YAnalysisId type char20 – type string with restriction maxLength(20)**
- **YReturn type _-bti_-teStXretrn**
  Msgtyp type char1 – type string with restriction maxLength(1)
  Msgid type char20 – type string with restriction maxLength(20)
  Msgnum type numeric3 – type string with restriction maxLength(3) pattern(\d*)
  Message type char220 – type string with restriction maxLength(220)
  Msgv1 type char50 – type string with restriction maxLength(50)
  Msgv2 type char50 – type string with restriction maxLength(50)
  Msgv3 type char50 – type string with restriction maxLength(50)
  Msgv4 type char50 – type string with restriction maxLength(50)
  Exception type char50 – type string with restriction maxLength(50)

4.2.5. Read the results of an analysis run

Reads the results of an analysis run started via the “Start the analysis for a business task” web service. This will return all analysis issues reported for the analysed business task so it can be determined if it safe to approve.

-bti-teTaskReadAnalysisWs

Input:

-bti-teTaskReadAnalysisWs

parameters type _-bti_-teTaskReadAnalysisWs

- **XAnalysisId type char20 – type string with restriction maxLength(20)**
- **XMaxlines type int**
- **YtAnalysisResultColumns type TableOf_-bti_-teStRfc110**
  item – optional, unbounded; type _-bti_-teStRfc110
  Groupid type char10 – type string with restriction maxLength(10)
  Columnid type char10 – type string with restriction maxLength(10)
  Columnname type char50 – type string with restriction maxLength(50)
  Columnwidth type char10 – type string with restriction maxLength(10)
- **YtAnalysisResultGroups type TableOf_-bti_-teStRfc112**
  item – optional, unbounded; type _-bti_-teStRfc112
  Groupid type char10 – type string with restriction maxLength(10)
  Anltypeid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Title type char100 – type string with restriction maxLength(100)
- **YtAnalysisResults type TableOf_-bti_-teStRfc107**
  item – optional, unbounded; type _-bti_-teStRfc107
  Groupid type char10 – type string with restriction maxLength(10)
  Rownum type char10 – type string with restriction maxLength(10)
  Columnid type char10 – type string with restriction maxLength(10)
  Texttype type char1 – type string with restriction maxLength(1)
  Text type char100 – type string with restriction maxLength(100)
- **YtConflictRequests type TableOf_-bti_-teStRfc103**
  item – optional, unbounded; type _-bti_-teStRfc103
  Strkorr type char20 – type string with restriction maxLength(20)
  Reason type char2 – type string with restriction maxLength(2)
  Trkorr type char20 – type string with restriction maxLength(20)
  Reldate type date – type string with restriction maxLength(10) pattern(\d\d\d\d-\d\d-\d\d)
  Reltime type time – type string with restriction maxLength(8) pattern(\d\d:\d\d:\d\d)
  AllowConfirm type char1 – type string with restriction maxLength(1)
  AllowAdminConf type char1 – type string with restriction maxLength(1)
  AllowImport type char1 – type string with restriction maxLength(1)

- **YtStatus type _-bti_-teTtAnlStatus**
  item – optional, unbounded; type _-bti_-teStAnlStatus
  Anltypeid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Description type char50 – type string with restriction maxLength(50)
  Startdate type date – type string with restriction maxLength(10) pattern(\d\d\d\d-\d\d-\d\d)
  Starttime type time – type string with restriction maxLength(8) pattern(\d\d:\d\d:\d\d)
  HasProgress type char1 – type string with restriction maxLength(1)
  Curnum type numeric8 – type string with restriction maxLength(8) pattern(\d*)
  Total type numeric8 – type string with restriction maxLength(8) pattern(\d*)
  Status type char1 – type string with restriction maxLength(1)
  Errors type numeric8 – type string with restriction maxLength(8) pattern(\d*)

Output:

-bti-teTaskReadAnalysisWsResponse

parameter type _-bti_-teTaskReadAnalysisWsResponse

- **YProblems type char1 – type string with restriction maxLength(1)**
- **YRunning type char1 – type string with restriction maxLength(1)**
- **YtAnalysisResultColumns type TableOf_-bti_-teStRfc110**
  tem – optional, unbounded; type _-bti_-teStRfc110
  Groupid type char10 – type string with restriction maxLength(10)
  Columnid type char10 – type string with restriction maxLength(10)
  Columnname type char50 – type string with restriction maxLength(50)
  Columnwidth type char10 – type string with restriction maxLength(10)
- **YtAnalysisResultGroups type TableOf_-bti_-teStRfc112**
  item – optional, unbounded; type _-bti_-teStRfc112
  Groupid type char10 – type string with restriction maxLength(10)
  Anltypeid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Title type char100 – type string with restriction maxLength(100)
- **YtAnalysisResults type TableOf_-bti_-teStRfc107**
  item – optional, unbounded; type _-bti_-teStRfc107
  Groupid type char10 – type string with restriction maxLength(10)
  Rownum type char10 – type string with restriction maxLength(10)
  Columnid type char10 – type string with restriction maxLength(10)
  Texttype type char1 – type string with restriction maxLength(1)
  Text type char100 – type string with restriction maxLength(100)
- **YtConflictRequests type TableOf_-bti_-teStRfc103**
  item – optional, unbounded; type _-bti_-teStRfc103
  Strkorr type char20 – type string with restriction maxLength(20)
  Reason type char2 – type string with restriction maxLength(2)
  Trkorr type char20 – type string with restriction maxLength(20)
  Reldate type date – type string with restriction maxLength(10) pattern(\d\d\d\d-\d\d-\d\d)
  Reltime type time – type string with restriction maxLength(8) pattern(\d\d:\d\d:\d\d)

AllowConfirm type char1 – type string with restriction maxLength(1)
AllowAdminConf type char1 – type string with restriction maxLength(1)
AllowImport type char1 – type string with restriction maxLength(1)

- **YtStatus type _-bti_-teTtAnlStatus**
  item – optional, unbounded; type _-bti_-teStAnlStatus
  Anltypeid type numeric4 – type string with restriction maxLength(4) pattern(\d*)
  Description type char50 – type string with restriction maxLength(50)
  Startdate type date – type string with restriction maxLength(10) pattern(\d\d\d\d-\d\d-\d\d)
  Starttime type time – type string with restriction maxLength(8) pattern(\d\d:\d\d:\d\d)
  HasProgress type char1 – type string with restriction maxLength(1)
  Curnum type numeric8 – type string with restriction maxLength(8) pattern(\d*)
  Total type numeric8 – type string with restriction maxLength(8) pattern(\d*)
  Status type char1 – type string with restriction maxLength(1)
  Errors type numeric8 – type string with restriction maxLength(8) pattern(\d*)

4.2.6. Approve a Business Task

Perform the approval of a business task. Prior to approval, the business task should be analysed to determine whether it is safe to approve. Approval will move the business task and its associated transports to the next location / process control point in ActiveControl.

-bti-teTaskApproveWs

Input:

-bti-teTaskApproveWs

parameters type _-bti_-teTaskApproveWs

- **XAnalysisId type char20 – type string with restriction maxLength(20)**
- **XLocation type char1 – type string with restriction maxLength(1)**
- **XTarget type numeric4 – type string with restriction maxLength(4) pattern(\d**)*
- **XTaskid type char20 – type string with restriction maxLength(20)**
- **XtRequest type _-bti_-teTtRfc010**
  item – optional, unbounded; type _-bti_-teStRfc010
  Trkorr type char20 – type string with restriction maxLength(20)

Output:

-bti-teTaskApproveWsResponse

parameter type _-bti_-teTaskApproveWsResponse

- **XtRequest type _-bti_-teTtRfc010**
  item – optional, unbounded; type _-bti_-teStRfc010
  Trkorr type char20 – type string with restriction maxLength(20)
- **YReturn type _-bti_-teStXretrn**
  Msgtyp type char1 – type string with restriction maxLength(1)
  Msgid type char20 – type string with restriction maxLength(20)
  Msgnum type numeric3 – type string with restriction maxLength(3) pattern(\d*)
  Message type char220 – type string with restriction maxLength(220)
  Msgv1 type char50 – type string with restriction maxLength(50)
  Msgv2 type char50 – type string with restriction maxLength(50)
  Msgv3 type char50 – type string with restriction maxLength(50)
  Msgv4 type char50 – type string with restriction maxLength(50)
  Exception type char50 – type string with restriction maxLength(50)

4.2.7. Enter the test results for a business task

Allows test results to be entered against a business task. ActiveControl uses a test series to log the test results.

Closing the test series will approve the testing and move the task and its associated transports to the next location / process control point in ActiveControl. If the test series is not closed (e.g. due to an unsuccessful test result) the test result are saved but the task and transports will not be approved and will remain open for further testing.

-bti-teTaskTestresWs

Input:

-bti-teTaskTestresWs

parameters type _-bti_-teTaskTestresWs

- **XClose type char1 – type string with restriction maxLength(1)**
- **XComment type string**
- **XLocation type char1 – type string with restriction maxLength(1)**
- **XRescode type char1 – type string with restriction maxLength(1)**
- **XTarget type numeric4 – type string with restriction maxLength(4) pattern(\d**)*
- **XTaskid type char20 – type string with restriction maxLength(20)**

Output:

-bti-teTaskTestresWsResponseSource code

parameter type _-bti_-teTaskTestresWsResponse

- **YReturn type _-bti_-teStXretrn**
  Msgtyp type char1 – type string with restriction maxLength(1)
  Msgid type char20 – type string with restriction maxLength(20)
  Msgnum type numeric3 – type string with restriction maxLength(3) pattern(\d*)
  Message type char220 – type string with restriction maxLength(220)
  Msgv1 type char50 – type string with restriction maxLength(50)
  Msgv2 type char50 – type string with restriction maxLength(50)
  Msgv3 type char50 – type string with restriction maxLength(50)
  Msgv4 type char50 – type string with restriction maxLength(50)
  Exception type char50 – type string with restriction maxLength(50)

4.2.8. Mapping Internal Values

All ActiveControl web services and API's expect the ActiveControl internal field values to be supplied for them to function correctly. These internal ID's can be found in the following configuration and transactional tables:

WS field	Table	Field	Description
Groupid	/BTI/TE_GROUPS	GROUPID	Task Group ID (Use CLASS = TASK)
Typeid	/BTI/TE_TYPE	ID	Task Type ID (Use CLASS = TASK)
Projectid	/BTI/TE_PROJ	ID	Project ID
Statdepl	/BTI/TE_TASKSTAT	STATID	Deployment Status (Use STATTYPE = DS)
Statplan	/BTI/TE_TASKSTAT	STATID	Planning Status Use STATTYPE = PS)
XTarget, Targetid	/BTI/TE_TARG	TARGET	Target ID
Path	/BTI/TE_PATH	PATH	Path ID
XAnalysisId, YAnalysisid	/BTI/TE_ANLTYPE	ANLTYPID	Analysis Type ID
Targetroleid	/BTI/TE_TARGROLE	ID	Target Role ID
Id, Taskid	/BTI/TE_TASK	ID	Task ID
XAnltypeid /BTI/TE_ANLRUN	ANALYSISRUN	ID	Analysis Run ID
Reason	/BTI/TE_ANREASON	REASON	Reason ID

Other mapping and web service fields:

WS Field	Description
XLocation	Location (Possible values: I – Inbox, Q – Import Queue, T – Test Queue, O – Outbox)
Priority	Priority (Possible values: 1 – Low, 2 – Normal, 3 – High, 4 – Urgent)
Locked	Locked (Flag values: X and SPACE)
XRescode	Test Result (Possible values: 0 – Testing Successful, 1 – Problem Found, 2 – Information, 3 – Waiting, 4 – Bypass Testing)
XClose	Close and Approve testing (Flag values: X and SPACE)
XDescription	Task long description
XTask, YTask	Structure for the main task field details
Caption	Task short description / subject

Reference	Task unique reference number (e.g. ticket, defect, change request number)
Testerid	SAP user id of the main task tester
Statdeplman	Flag to indicate that the task deployment status is manually set rather than allowing ActiveControl to set it (Flag values: X and SPACE)
Statplanman	Flag to indicate that the task planning status is manually set rather than allowing ActiveControl to set it (Flag values: X and SPACE)
Systemid	SAP system ID
XtCustfields, YtCustfields	Task custom field values structure formatted as a list of: • Id – Custom field ID • Value – Custom field value
XtTesters, YtTesters	Task testers structure to list the testers for the task
XUpdateCustfields	Flag to indicate whether the task Custom Fields are to be updated (Flag values: X and SPACE)
XUpdateDesc	Flag to indicate whether the task Description are to be updated (Flag values: X and SPACE)
XUpdateTesters	Flag to indicate whether the task Testers are to be updated (Flag values: X and SPACE)
XUpddateTask	Flag to indicate whether the task main fields (in XTask) are to be updated (Flag values: X and SPACE)
YProblems	Flag to indicate whether any analysis issues were found (Flag values: X and SPACE)
YRunning	Flag to indicate whether any analysis is still running or not (Flag values: X and SPACE)
XComment	Test to enter a free text comment during test results entry
XtRequest – Trkorr	A list of SAP transports to be analysed / approved
YReturn	WS call return structure to pass back messages and errors
Msgtyp	Type of message returned from the WS call (Possible values: E – Error, W – Warning, I – Information, S/Blank – Success)
Msgid	ID of the message returned from the WS call
Msgnum	Number of the message returned from the WS call
Message	Message text returned from the WS call
Msgv1 – 4	Further message texts returned from the WS call

4.2.9. Other Communication Techniques

Although the use of web services is the standard communication technique using ActiveControl, as the product resides in the SAP Netweaver stack, other SAP standard communication techniques are available for integration if preferred.

tRFC Communication
All ActiveControl APIs exist as remote enabled function modules within the ABAP environment and can therefore be called using the standard tRFC calls through an appropriate RFC destination. If the external system integrating with ActiveControl is either another SAP system or able to call remote functions directly then this method of communication can be used.

For inbound scenarios, the standard API's can be called directly. For outbound scenarios, new send methods would need to be developed to enable direct calling of the external system.

IDoc Communication
As with TRFC integration above, as the ActiveControl API's are standard function modules, IDoc wrappers can be created to call them and standard IDoc processing configured to control the integration.

For inbound scenarios, the appropriate IDoc wrappers would need to be generated and any IDoc sub-system configuration completed. Once again, for outbound scenarios, new send methods would need to be developed for IDoc communication to be enabled.

5. RESTful APIs

> ! Please note that ActiveControl's suite of RESTful APIs are only available from ActiveControl 9.00 onwards.

Background

For many years, ActiveControl has offered a suite of SOAP APIs to perform certain key tasks as part of an ActiveControl integration.

From ActiveControl 9.00 onwards, a suite of RESTful APIs is also available as part of the ActiveControl Web Platform; this much more extensive suite of APIs is intended to help customers perform more advanced integrations.

Documentation

Information on the available RESTful APIs can be found via your own installation of the ActiveControl Web Platform.

Please refer to this online Knowledge Article for further information.

> * Current out-of-the-box integrations such as JIRA and ServiceNow use the existing SOAP APIs. There is no current plans to switch these existing integrations over to the RESTful APIs.

6. HTTP APIs

6.1. Get Queue Contents

HTTP GET

Example URL:
http://te.basistechnologies.net:8000/bti/te_web_services?action=QUEUE_CONTENTS&TARGETID=168&LOCATION=T

Authentication – Basic (ActiveControl user credentials)

Request Parameters

Name	Value	Description
action	QUEUE_CONTENTS	The get queue content action
TARGETID	168	Look in the /BTI/TE_TARG table for the target id. See example below.
LOCATION	T	The queue location in the target system. I – Inbox Q – Import Queue T – Test queue O – Outbox

Data Browser: Table /BTI/TE_TARG Select Entries 195

Check Table...

TARGET	SYSID	TTEXT	APPROVEIN	APPROVEOUT	MINAP
154	TD0	Target for foreign transport TD0			
155	HD3	S4/Hana for multi(Merge)			
156	HD3	S4/Hana DEV for multi			
157	ACD	Jira integration dev			
158	T03	Jira integration demo QA			
159	D01	RN - D01			
160	TY1	RN - TY1 Target			
161	HD2	CIQ			
162	ACD	ACDC			
163	TQ1	Quality Assurance			
164	ACD	Kali ACD MERGE			
165	ACD	S/4 Project Dev System			
166	T03	S/4 Project Test			
167	ACD	Jenkins demo source			
168	D01	Jenkins demo target			
169	ACD	S/4 Project Dev System MERGE			

Response

XML Payload

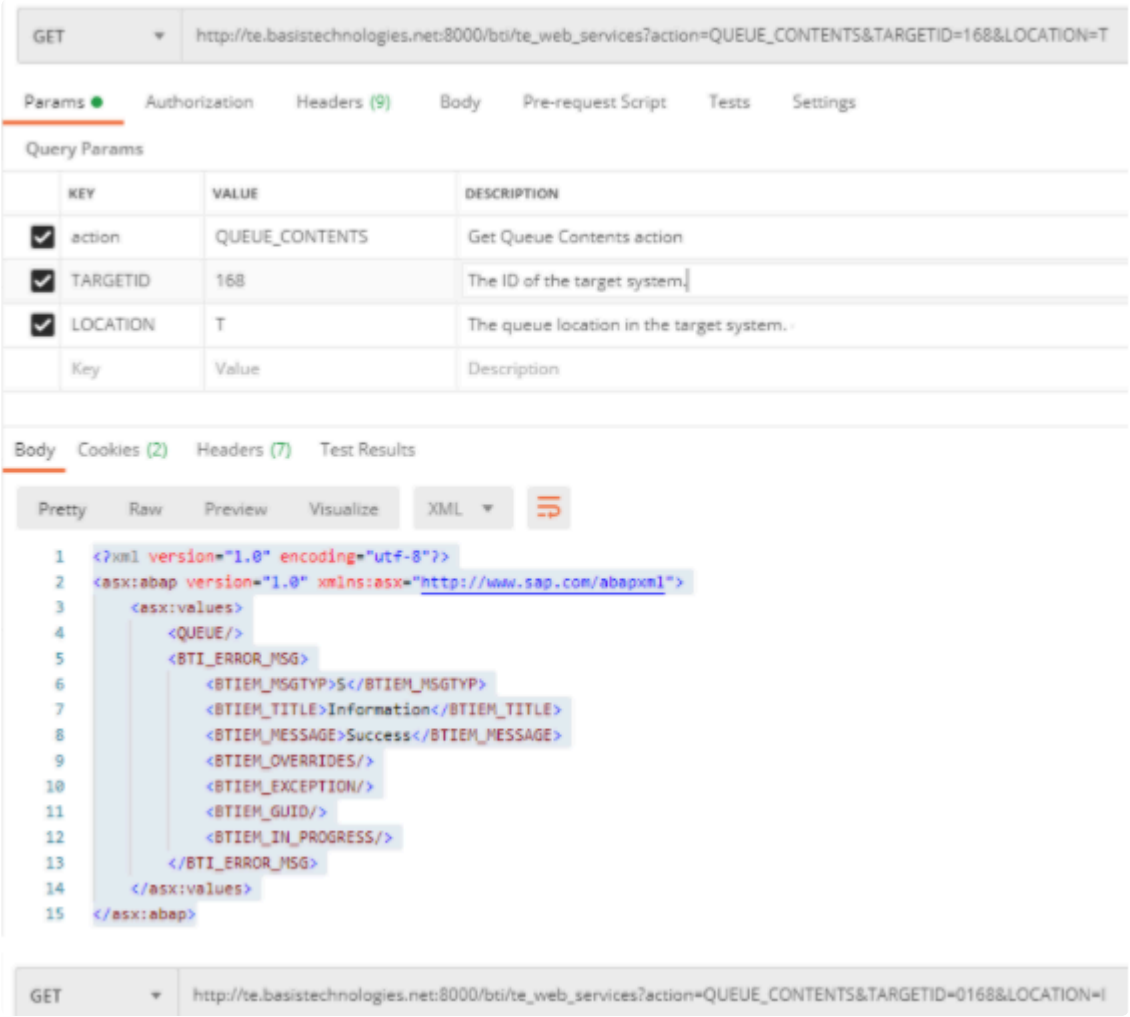
```xml
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
  <asx:values>
    <QUEUE>
      <item> <!-- Zero to many Business Tasks -->
        <ID>10020041500000141320</ID>
        <CAPTION>jenkins demo 1</CAPTION>
        <REFERENCE>jd1</REFERENCE>
        <GROUPID>10017100600000000012</GROUPID>
        <TYPEID>10017100600000000017</TYPEID>
        <TESTERID>MURBANI</TESTERID>
        <PRIORITY>2</PRIORITY>
        <PROJECTID>10017100600000000022</PROJECTID>
        <LOCKED/>
        <PATH>00</PATH>
        <STAT_DEPL>10020041500000141305</STAT_DEPL>
        <STAT_PLAN/>
        <STAT_DEPL_MAN/>
        <STAT_PLAN_MAN/>
```

```xml
        <OWNER/>
        <TEXT/>
        <CF_508/>
        <CF_512/>
        <CF_513>Business task test1</CF_513>
        <TRANSPORTS>
          <item> <!-- Zero to many Transport Forms -->
            <TRKORR>ACDK908578</TRKORR>
            <REQTEXT/>
            <FORMDESCRIPTION/>
            <GROUPID>10017100600000000008</GROUPID>
            <TYPEID>10017100600000000020</TYPEID>
            <TRFUNCTION/>
            <TRCATEGORY/>
            <TRSTATUS/>
            <CLIENT/>
            <REQUESTOR>MURBANI</REQUESTOR>
            <REQNAME/>
            <REQDATE>20200415</REQDATE>
            <REQTIME>104145</REQTIME>
            <RELDATE/>
            <RELTIME/>
            <PATH>55</PATH>
```

```
            <PATH>55</PATH>
            <HASFORM/>
            <OWNER/>
            <COMPLETED/>
            <CLIDEP/>
            <QUEUED/>
            <HIDDEN/>
            <LOCKED/>
            <CONFLICTS/>
            <REFERENCE/>
            <UMODES/>
            <NOEXPLOGS/>
            <EXPORTSTAT/>
            <TSTIMPSTAT/>
            <LOCSTATUS/>
            <MANDT/>
            <RETURNCODE/>
            <TARSYSTEM/>
            <CHANGETYPE/>
            <MANUAL_STEP_STATUS/>
            <PREV_IMP_RESULT/>
            <CONFLICT_STAT>0</CONFLICT_STAT>
            <SCI_STAT>0</SCI_STAT>
            <RISK_STAT>0</RISK_STAT>
            <NUM_OBJECTS>0000000000</NUM_OBJECTS>
            <NUM_KEYS>0000000000</NUM_KEYS>
            <CONF_STAT_RUNID/>
            <SCI_STAT_RUNID/>
            <RISK_STAT_RUNID/>
            <REJECTED/>
            <AUTO_APPR_STATUS/>
            <CF_500>HAMPTON HILL</CF_500>
            <CF_501>N</CF_501>
            <CF_502/>
            <CF_503/>
            <CF_504>N</CF_504>
            <CF_505>N</CF_505>
            <CF_506>N</CF_506>
            <CF_507>N</CF_507>
            <CF_509/>
            <CF_510/>
            <CF_511/>
            <CF_514>Test for 908578</CF_514>
        </item>
    </TRANSPORTS>
```

```
            </TRANSPORTS>
          </item>
        </QUEUE>
        <BTI_ERROR_MSG>
          <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
          <BTIEM_TITLE>Information</BTIEM_TITLE>
          <BTIEM_MESSAGE>Success</BTIEM_MESSAGE>
          <BTIEM_OVERRIDES/>
          <BTIEM_EXCEPTION/>
          <BTIEM_GUID/>
          <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
      </asx:values>
    </asx:abap>
```


Postman Example



GET http://te.basistechnologies.net:8000/bti/te_web_services?action=QUEUE_CONTENTS&TARGETID=0168&LOCATION=I

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Headers Hide auto-generated headers

	KEY	VALUE
☑	Authorization	Basic Z[redacted]
☑	Cookie	sap-usercontext=sap-client=100; SAP_S
☑	Cache-Control	no-cache
☑	Postman-Token	<calculated when request is sent>
☑	Host	<calculated when request is sent>
☑	User-Agent	PostmanRuntime/7.26.5
☑	Accept	*/*
☑	Accept-Encoding	gzip, deflate, br
☑	Connection	keep-alive
	Key	Value

6.2. Lock Queue

HTTP GET

Example URL:
http://te.basistechnologies.net:8000/bti/te_web_services?action=LOCK_TARGET&TARGETID=168

Authentication – Basic (ActiveControl user credentials)

Note – This locks only the import queue for the target system.

Request Parameters

Name	Value	Description
action	LOCK_TARGET	Locks the target system import queue.
TARGETID	168	Look in the /BTI/TE_TARG table for the target id.

Response


```xml
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
    <asx:values>
        <BTI_ERROR_MSG>
            <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
            <BTIEM_TITLE>Information</BTIEM_TITLE>
            <BTIEM_MESSAGE>Target Locked</BTIEM_MESSAGE>
            <BTIEM_OVERRIDES/>
            <BTIEM_EXCEPTION/>
            <BTIEM_GUID/>
            <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
    </asx:values>
</asx:abap>
```

Postman Example

GET http://te.basistechnologies.net:8000/bti/te_web_services?action=LOCK_TARGET&TARGETID=168

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
☑	action	LOCK_TARGET	The LOCK_TARGET action
☑	TARGETID	168	The ID of the target system import queue to lock.
	Key	Value	Description

Body Cookies (3) Headers (6) Test Results

Pretty Raw Preview Visualize XML

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
    <asx:values>
        <BTI_ERROR_MSG>
            <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
            <BTIEM_TITLE>Information</BTIEM_TITLE>
            <BTIEM_MESSAGE>Target Locked</BTIEM_MESSAGE>
            <BTIEM_OVERRIDES/>
            <BTIEM_EXCEPTION/>
            <BTIEM_GUID/>
            <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
    </asx:values>
</asx:abap>
```

6.3. Unlock Queue

HTTP GET

Example URL:
http://te.basistechnologies.net:8000/bti/te_web_services?action=UNLOCK_TARGET&TARGETID=168

Authentication – Basic (ActiveControl user credentials)

Note – This unlocks only the import queue for the target system.

Request Parameters

Name	Value	Description
action	UNLOCK_TARGET	Unlocks the target system import queue.
TARGETID	168	Look in the /BTI/TE_TARG table for the target id.

Response

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
    <asx:values>
        <BTI_ERROR_MSG>
            <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
            <BTIEM_TITLE>Information</BTIEM_TITLE>
            <BTIEM_MESSAGE>Target Unocked</BTIEM_MESSAGE>
            <BTIEM_OVERRIDES/>
            <BTIEM_EXCEPTION/>
            <BTIEM_GUID/>
            <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
    </asx:values>
</asx:abap>
```

Note – the BTIEM_MESSAGE value ‘Target Unocked’ is misspelled. A defect has been submitted to fix the

spelling in a future release.

Postman Example

GET http://te.basistechnologies.net:8000/bti/te_web_services?action=UNLOCK_TARGET&TARGETID=168

Params Authorization Headers (9) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
☑	action	UNLOCK_TARGET	The UNLOCK_TARGET action
☑	TARGETID	168	The ID of the target system import queue to lock.
	Key	Value	Description

Body Cookies (3) Headers (4) Test Results

Pretty Raw Preview Visualize XML

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
3      <asx:values>
4          <BTI_ERROR_MSG>
5              <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
6              <BTIEM_TITLE>Information</BTIEM_TITLE>
7              <BTIEM_MESSAGE>Target Unocked</BTIEM_MESSAGE>
8              <BTIEM_OVERRIDES/>
9              <BTIEM_EXCEPTION/>
10             <BTIEM_GUID/>
11             <BTIEM_IN_PROGRESS/>
12         </BTI_ERROR_MSG>
13     </asx:values>
14 </asx:abap>
```

6.4. Save Business Task Test Result

Purpose

API can be used to entering/saving successful Test Results on a Business Task, as part of automated testing process.

HTTP POST

Example URL:
http://te.basistechnologies.net:8000/bti/te_web_services?action=TARGET_TESTRESULTS&TASKID=10020041500000141320

Authentication – Basic (ActiveControl user credentials)

Request Parameters

Name	Value	Description
action	TARGET_TESTRESULTS	The TARGET_TESTRESULTS action
TASKID	10020041500000141320	The ActiveControl Business Task ID

Response


```xml
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
    <asx:values>
        <BTI_ERROR_MSG>
            <BTIEM_MSGTYP>S</BTIEM_MSGTYP>
            <BTIEM_TITLE>Information</BTIEM_TITLE>
            <BTIEM_MESSAGE></BTIEM_MESSAGE>
            <BTIEM_OVERRIDES/>
            <BTIEM_EXCEPTION/>
            <BTIEM_GUID/>
            <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
    </asx:values>
</asx:abap>
```


Error response example


```xml
<?xml version="1.0" encoding="utf-8"?>
<asx:abap version="1.0" xmlns:asx="http://www.sap.com/abapxml">
    <asx:values>
        <BTI_ERROR_MSG>
            <BTIEM_MSGTYP>E</BTIEM_MSGTYP>
            <BTIEM_TITLE>Error</BTIEM_TITLE>
            <BTIEM_MESSAGE>No task available to record results to</BTIEM_MESSAGE>
            <BTIEM_OVERRIDES/>
            <BTIEM_EXCEPTION/>
            <BTIEM_GUID/>
            <BTIEM_IN_PROGRESS/>
        </BTI_ERROR_MSG>
    </asx:values>
</asx:abap>
```

Postman Example

POST http://te.basistechnologies.net:8000/bti/te_web_services?action=TARGET_TESTRESULTS&TASKID=10020041500000141320

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
action	TARGET_TESTRESULTS	The TARGET_TESTRESULTS action
TASKID	10020041500000141320	The ActiveControl Business Task ID
Key	Value	Description

POST http://te.basistechnologies.net:8000/bti/te_web_services?action=TARGET_TESTRESULTS&TASKID=10020041500000141320

Params Authorization Headers (11) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
queueresultsfile	<TESTRESULTS> - ...	This specifiies the target ID and if the test was successful
attachment	Select Files	Optional file attachment for the actual results
Key	Value	Description

POST http://te.basistechnologies.net:8000/bti/te_web_services?action=TARGET_TESTRESULTS&TASKID=10020041500000141320

Params Authorization Headers (11) Body Pre-request Script Tests Settings

Headers Hide auto-generated headers

KEY	VALUE	DESCRIPTION
Authorization	Basic	
Cookie	sap-usercontext=sap-client=1	
Cache-Control	no-cache	
Postman-Token	<calculated when request is s	
Content-Type	multipart/form-data; boundar	
Content-Length	<calculated when request is s	
Host	<calculated when request is s	
User-Agent	PostmanRuntime/7.26.5	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Key	Value	Description

Notes

SUCCESS – can be either X (for Successful Testing) or left Blank (for Problem Found). It is not possible to pass back results relating to Information, Waiting, Bypass etc)

7. Web Hooks

7.1. Deployment Status Webhook

From ActiveControl 8.40, a CI webhook that supports a POST method to send a payload with Business Task and Transport Form details is available.

This is done via new key CI_INTEGRATION_METHOD_XX (where XX is the number of the Integration). When this is set to POST, ActiveControl will generate a hierarchical payload with the following:

1) All the affected Business Tasks
2) All the live (ie not Complete) Transport Forms associated with these Business Tasks. (whether relevant for the Integration event or not)
3) the location of these transports, along with the relevant statuses.
4) All Custom Field information, for both the Business Task and Transport Forms.

Only header data is sent as part of the Payload. (Ie no Tester / Event / Planning Data or Attachments information is sent). These will bypass AC's mapping framework and pass on the data as-is for consumption on the receiving end.

Sample_nr.json
json · 3 KB


```
[
    {
        "ID": "10018031500000013789",
        "CAPTION": "TESTTE-904",
        "REFERENCE": "TESTTE-904",
        "GROUPID": "10017100600000000013",
        "TYPEID": "10017100600000000017",
        "TESTERID": "KSOLAIMUTHU",
        "PRIORITY": "2",
        "PROJECTID": "10018041700000014297",
        "LOCKED": "",
        "PATH": "00",
        "STAT_DEPL": "10017100600000000025",
        "STAT_PLAN": "",
        "STAT_DEPL_MAN": "X",
        "STAT_PLAN_MAN": "",
        "OWNER": "",
        "TEXT": "tettets tysdkjhds fkjshflk asfdkhjksadf hkjshfkshadflk sjhafk ssdakfhdf sadjkfh sadf sdfakjhsdaf sadfkjh sdfkljsadhf sdfakjf hsadjkhfasd lkdsajfhasdf dfljhsaf sdaflkkjashfsa kfhnsakfh sdfkjhsadf sadf",
        "CF_508": "",
        "CF_512": "",
        "CF_513": "",
        "CF_517": "",
        "CF_518": "",
        "CF_519": "",
        "CF_520": "",
        "CF_521": "",
        "CF_522": "",
        "CF_528": "",
        "CF_529": "",
        "CF_530": "",
        "CF_531": "",
        "CF_532": "",
        "CF_533": "",
        "CF_534": "",
        "TRANSPORTS": [
            {
                "TRKORR": "ACDK901241",
                "REQTEXT": "",
                "FORMDESCRIPTION": "",
                "GROUPID": "10017100600000000009",
                "TYPEID": "10017100600000000020",
                "TRFUNCTION": "",
                "TRCATEGORY": "",
                "TRSTATUS": "",
                "CLIENT": "",
                "REQUESTOR": "KSOLAIMUTHU",
                "REQNAME": "",
                "REQDATE": "20180410",
                "REQTIME": "170345",
                "RELDATE": "",
                "RELTIME": "",
                "PATH": "03",
                "HASFORM": "",
```


Figure: Sample json containing all Busines Task and Transport Form information (both standard and custom fields

The POST version of the hook is triggered by flag CI_INTEGRATION_METHOD_XX set to POST in tvarv (XX is the integration number)

Data Browser: Table /BTI/TE_TVARV Select Entries 1

NAME	NUMB	SIGN	OPTI	LOW	HIGH
CI_INTEGRATION_METHOD_24	1	I	EQ	POST	

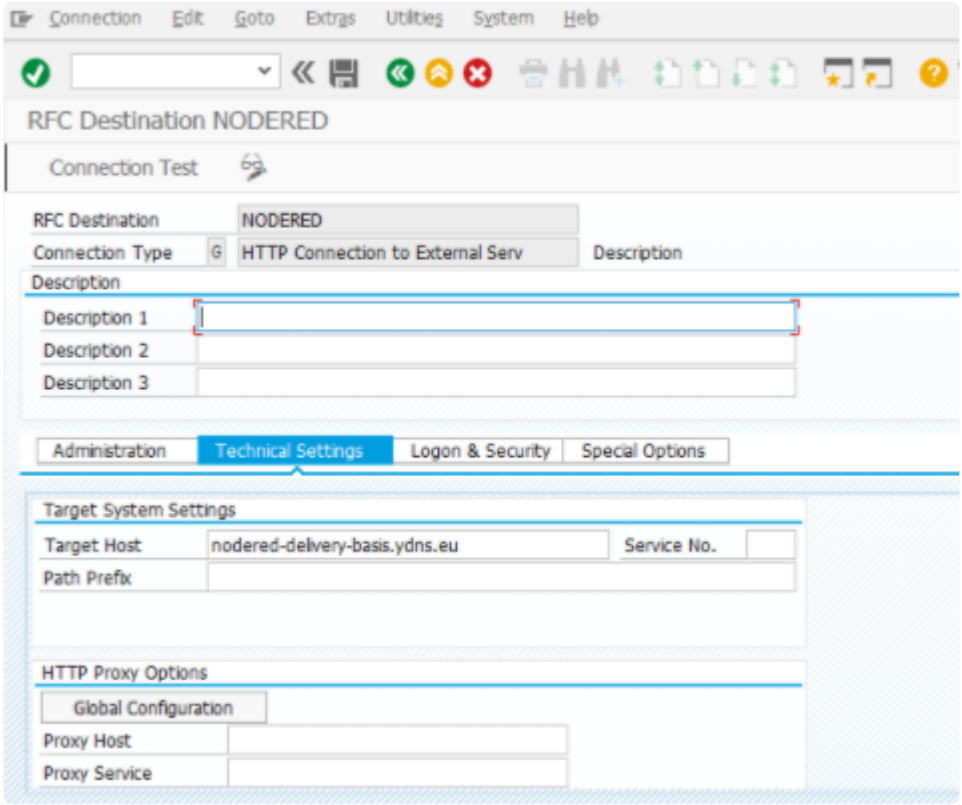
7.1.1. Configuration Steps

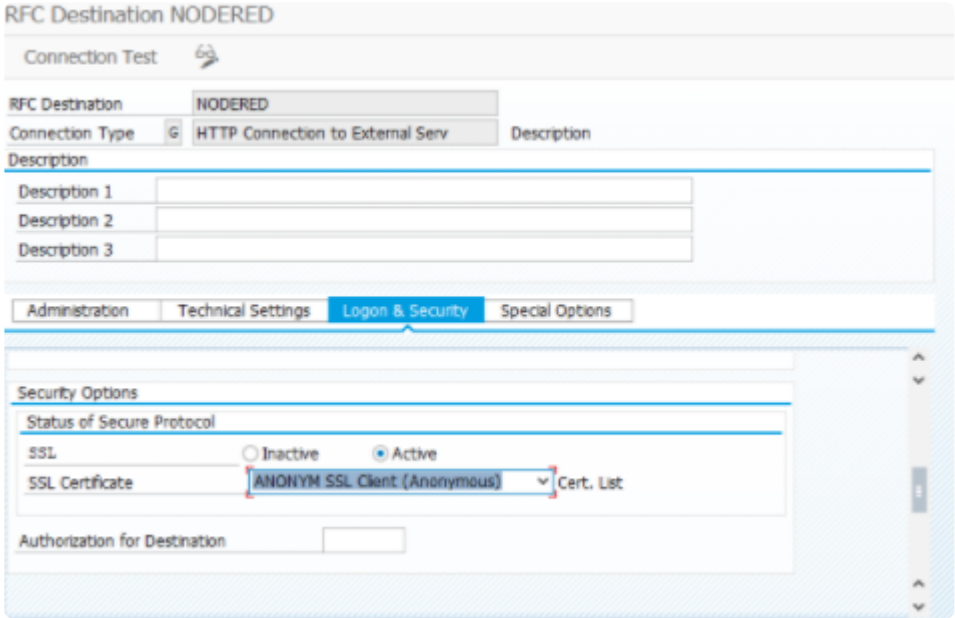
This section details all of the setup steps for the Deployment Status Webhook.

7.1.1.1. RFC Destination

In the ActiveControl Domain Controller – create an RFC Destination (via SM59) for your endpoint application.

In screenshot example, the Target Host is https://nodered-delivery-basis.ydns.eu.





If security basic authentication is required, enter it on the Logon & Security tab.

You will need to check with your local SAP Basis team if an SSL Certificate is required to be installed.

7.1.1.2. Table: /BTI/TE_INT_SYST

Create an entry for the external system in /BTI/TE_INT_SYST table.

Table: /BTI/TE_INT_SYST
Displayed Fields: 9 of 12 Fixed Columns: 2 List Width 0250

EXTSYS_NO	EXTSYS_ID	EXTSYS_NAME	RFC_DEST	DDCINT	TASKFIELD_LINK	FORMFIELD_LINK	INT_USER	INT_PASSWORD
16	NODERED	NODERED	NODERED					

Note: In this case, NODERED is used and the EXTSYS_NO is 16. (because that is the latest available number in our internal system)

7.1.1.3. Table: /BTI/TE_INT_CLAS

Create an entry for the external system in /BTI/TE_INT_CLAS table. Enter the EXTSYS_NO and the CLASSNAME OF /BTI/TE_INTEGRATION_CI.

Table: /BTI/TE_INT_CLAS
Displayed Fields: 2 of 2 Fixed Columns: 1 List Width 0250

EXTSYS_NO	CLASSNAME
16	/BTI/TE_INTEGRATION_CI

7.1.1.4. Table: /BTI/TE_TVARV

Create entries for CI_INTEGRATION_METHOD_nn and CI_INTEGRATION_URI_nn in /BTI/TE_TVARV table.

Table: /BTI/TE_TVARV
Displayed Fields: 6 of 6 Fixed Columns: 2 List Width 0250

NAME	NUMB	SIGN	OPTI	LOW	HIGH
CI_INTEGRATION_METHOD_16	0001		EQ	POST	
CI_INTEGRATION_URI_16	0001			/task/update2	

7.1.1.5. Table: /BTI/TE_INT_MAPP

Create an entry for the external system in /BTI/TE_INT_MAPP table.

Table: /BTI/TE_INT_MAPP
Displayed Fields: 8 of 9 Fixed Columns: 4 List Width 0250

EXTSYS_NO	EXTSYS_NAME	DIRECTION	SEQUENCE_NO	TEFIELDREF	EXTERNAL_REF	KEY_FIELD	TECUSTFLD_REF
16	NODERED		0000000001	/BTI/TE_TASK-STAT_DEPL	/BTI/TE_TASK-STAT_DEPL		

- Enter the EXTSYS_NO as per what was configured in /BTI/TE_INT_SYST table.
- TEFIELDREF: ‘/BTI/TE_TASK-STAT_DEPL’
- EXTERNAL REF: ‘/BIT/TE_TASK-STAT_DEPL’

7.1.1.6. Table: /BTI/TE_INT_PROC

Create an entry for the external system in /BTI/TE_INT_PROC table.

Enter the EXTSYS_NO, External System name and the deployment status ID. You can find the deployment status id in the /BTI/TE_TASKSTAT table.

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

EXTSYS_NO	EXTSYS_NAME	IDENTIFIER	PROCESS_CODE	IGNORE_CHANGES
16	NODERED	10013072400000108516	UPDATE	

7.1.1.7. Table: /BTI/TE_INT_CONV

Create an entry for the external system in /BTI/TE_INT_CONV table. Enter the EXTSYS_NO, External System name, Direction as ‘Outbound’. T

The rest of the fields can have a meaningless entry such as ‘DUMMY’ shown in this example. Conversions will be handled by the endpoint system the Deployment Status Webhook will call.

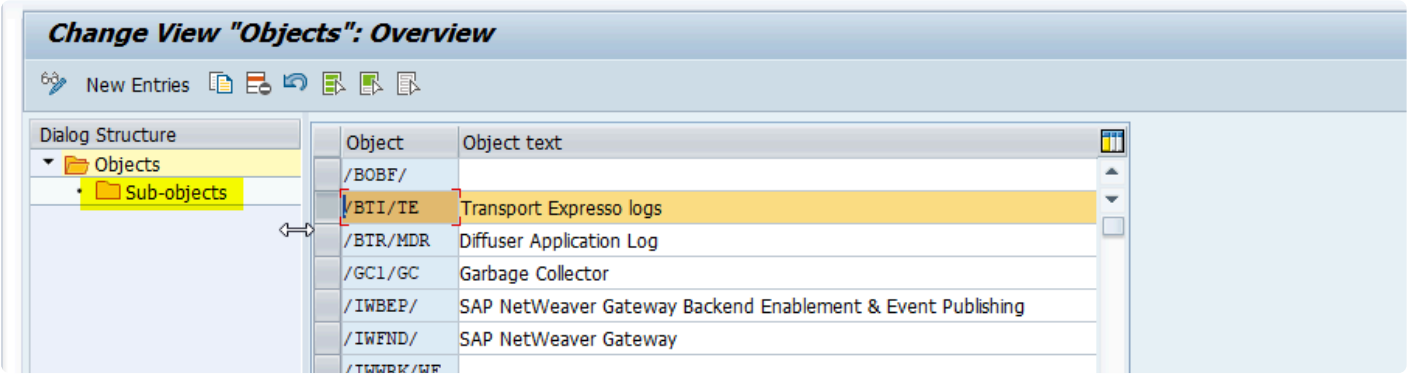
Table: /BTI/TE_INT_CONV
Displayed Fields: 5 of 6 Fixed Columns: 5 List Width 0250

EXTSYS_NO	EXTSYS_NAME	DIRECTION	EXTERNAL_REF	EXTFLD_ID
16	NODERED	O	DUMMY	DUMMY

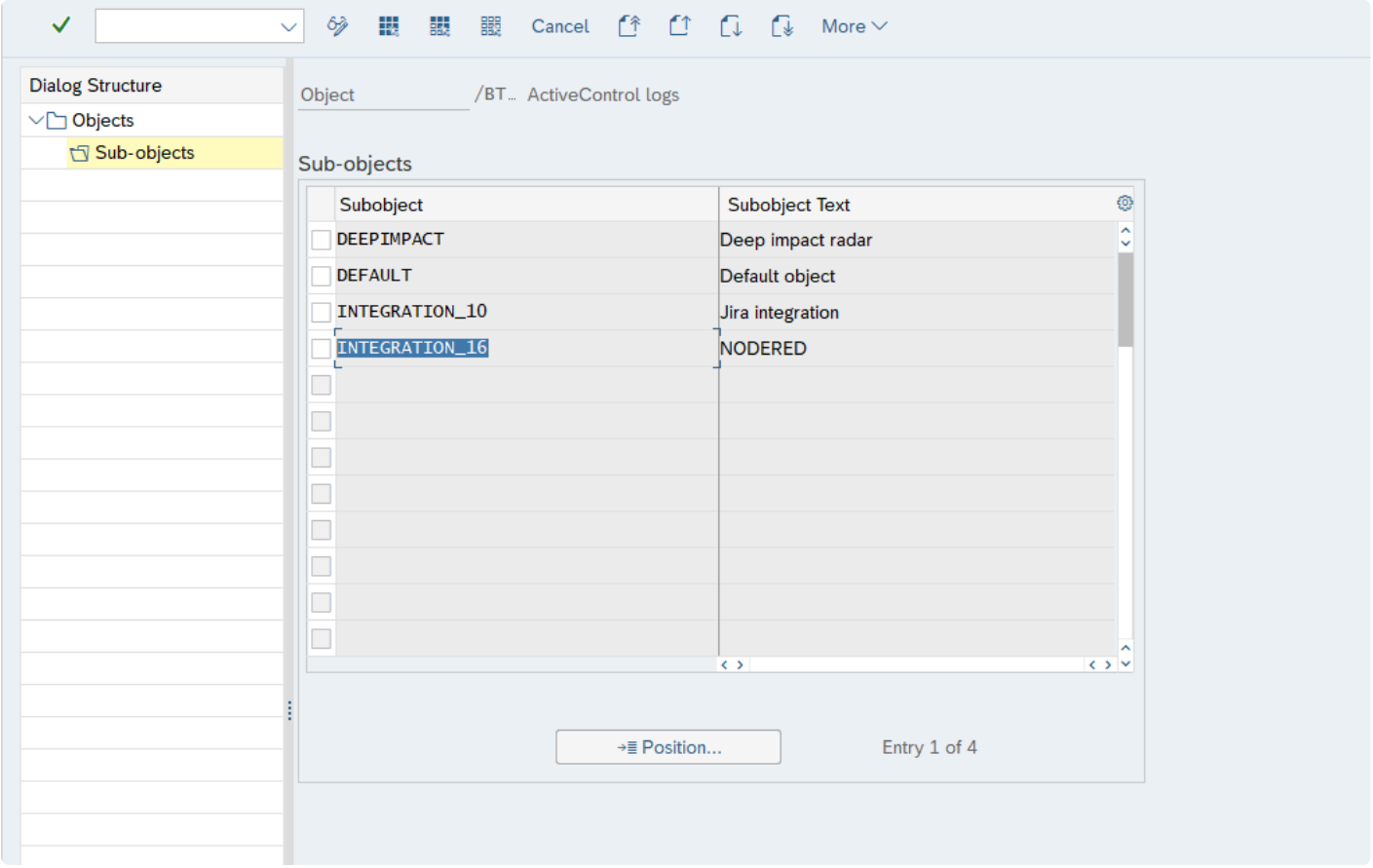
7.1.1.8. Create sub-object INTEGRATION_nn

Go to transaction SLG0

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

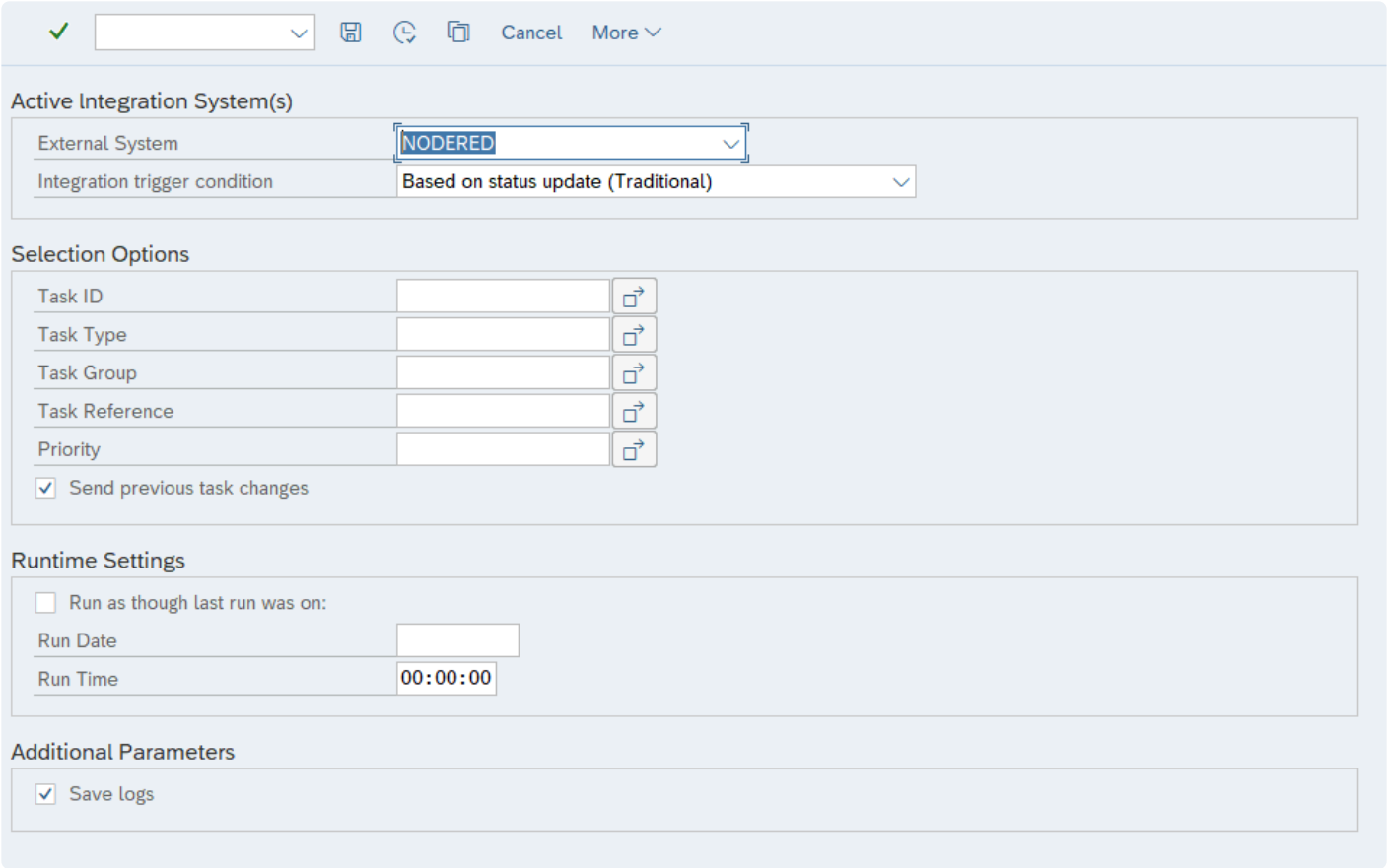


Click on new entry and create sub-object INTEGRATION_nn and save it. (where NN is the number of the integration as per /BTI/TE_INT_SYST



7.1.1.9. Program: /BTI/TE_INTEG_TRIGGER

Create a variant of program /BTI/TE_INTEG_TRIGGER via SE38

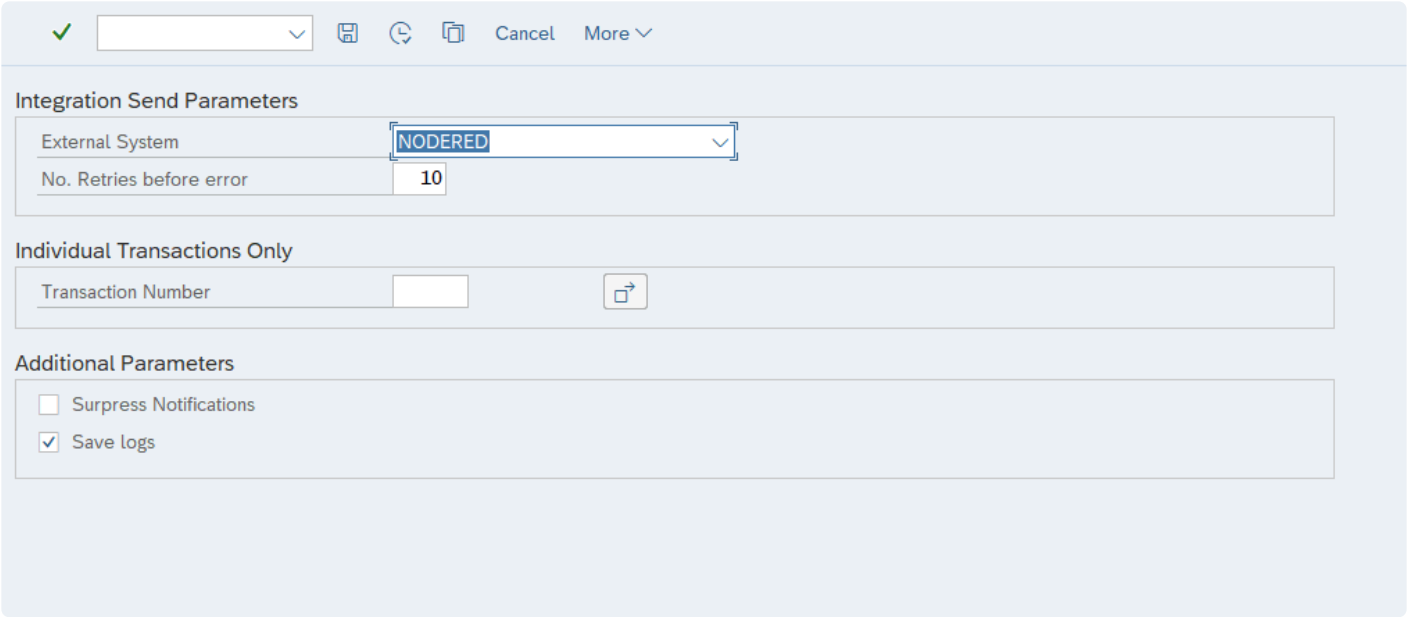


Schedule this program/variant to run periodically (eg every 5 minutes) via SM36.

> **!** It is recommended to create one Job, running the /BTI/TE_INTEG_TRIGGER and then the /BTI/TE_INTEG_SEND detailed in following setup step.

7.1.1.10. Program: /BTI/TE_INTEG_SEND

Create a variant of program /BTI/TE_INTEG_SEND via SE38



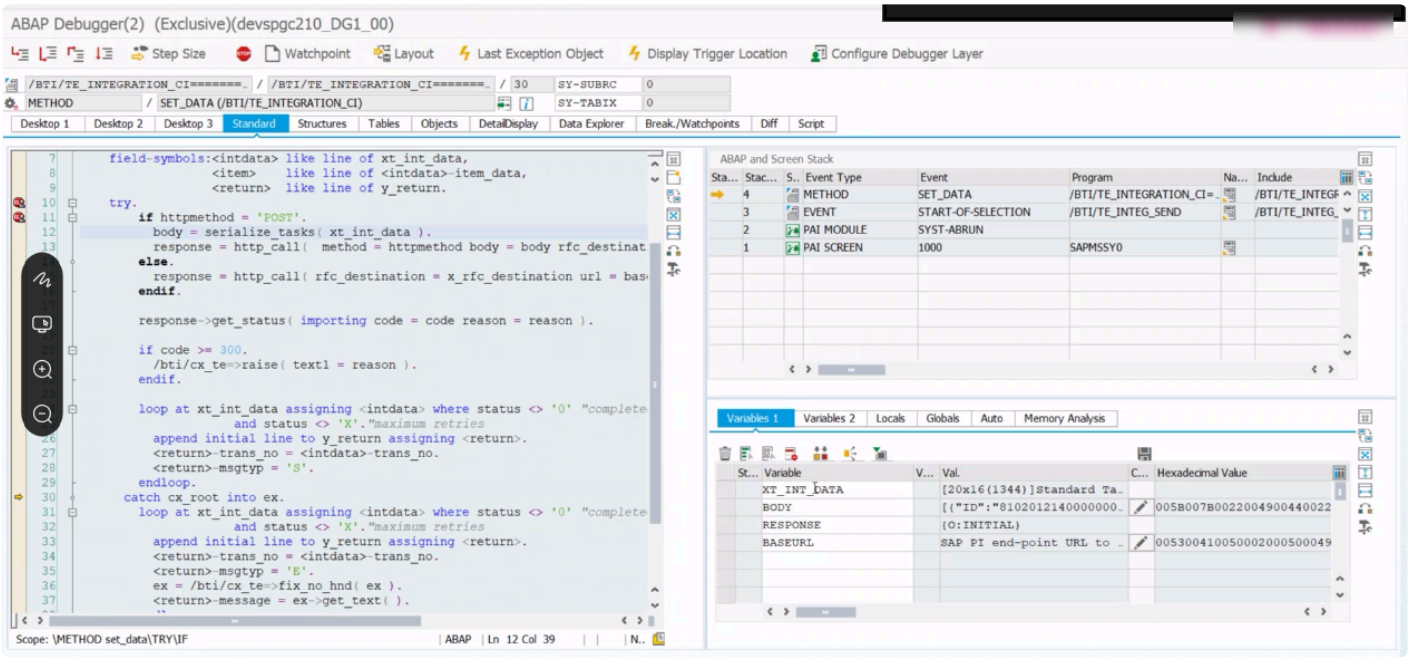
Schedule this program/variant to run periodically (eg every 5 minutes) via SM36.

> ! It is recommended to create one Job, running the /BTI/TE_INTEG_TRIGGER and then the /BTI/TE_INTEG_SEND

7.2. Debugging the JSON

If you want to view the JSON that will be created by this Web Hook in advance of setting up RFC connectivity to the receiving middleware (be that PI, Node-RED or whatever) – you can do this setting a breakpoint within the following

class: /bti/te_integration_ci
method: set_data



8. Support of Custom Integrations

The following matrix summarises the responsibilities & boundaries of support for Custom Integrations built outside of our SAP and Node-RED integration frameworks.

	SAP-side Configuration	SAP-side Application ActiveControl + APIs	SAP-side Customisations User Exits, Z developments	Connectivity (SAP < > 3rd Party Application) RFCs, Firewall, Network Ports etc	3rd Party Application	3rd Party Application Configuration
Any 3rd Party Application	Basis Technologies (only where the configuration was performed by Basis Technologies originally. Otherwise, chargeable services need to be in place)	Basis Technologies	Basis Technologies (only when the Customisations were developed by Basis Technologies)	Customer	Customer	Customer

9. Further Information?

As with ActiveControl integrations setup using our SAP or Node-RED integration frameworks, Basis Technologies would strongly recommend some formal consulting services support in the setup of an ActiveControl Integration using our APIs and Webhooks

If you are interested in finding out more about ActiveControl's integration capabilities, please reach out to your Basis Technologies Acount Manager and they can help to arrange a demo.