

ActiveControl - Administration Guide

8.0 — Last update: 4 November 2020

Basis Technologies

Copyright © 2020 Basis Technologies

Table of Contents

1. Introduction 6

2. Key Concepts of ActiveControl 7

2.1. Architecture of ActiveControl 7

2.1.1. Domain Controller 8

2.1.2. Satellite SAP Systems 8

2.1.3. Accessing ActiveControl 9

2.1.4. Integration Framework 11

2.2. Transport Forms, Business Tasks and Projects 11

2.3. Inboxes, Outboxes and Test Queues 12

2.4. Transport Sequencing and Dependencies 13

3. Installing ActiveControl 15

3.1. Installing the Server Software 15

3.2. Removing the Server Software 15

3.3. Installing the Client Software 16

3.4. Removing the Client Software 16

3.5. RFC Destinations 16

3.6. Next Steps 18

4. Configuring ActiveControl 20

4.1. SAP GUI Configuration 20

4.1.1. Completing Transport Forms and Business Tasks in the SAP GUI 20

4.1.1.1. Switching on SAP GUI Processing 22

4.1.1.2. Configuring SAP GUI Processing 24

4.1.1.3. Transport Form Organizer 24

4.1.1.4. Opening the Windows GUI from SAP GUI 25

4.1.2. Web User Interface 25

4.1.2.1. BSP Services 26

4.1.2.2. Web UI – Metrics 27

4.1.2.3. Web UI – Newsfeed 28

4.1.2.4. Web UI – Following 29

4.1.2.5. Web UI – Project Milestones 29

4.1.2.6. Web UI – System messages 30

4.1.2.7. Web UI – Other Configuration Tables 30

4.1.3. E-mail Notifications 32

4.1.3.1. Standard email notifications 32

4.1.3.2. Custom email notifications 33

4.1.3.3. Reminder Notifications 34

4.1.3.4. Analysis Result standard notifications 35

4.1.3.5. Analysis Result custom notifications 36

4.1.3.6. Import Error (RC8) Notification – Additional Recipients. 36

4.1.4. Authorisations 36

4.1.4.1. User Authorisations 37

4.1.4.2. View Authorisations 39
4.1.4.3. Activity Authorisations 41
4.1.4.4. Legacy – Screen Variants 43
4.2. Windows GUI Configuration 44
4.2.1. Defining Targets 45
4.2.1.1. Target Properties – General 46
4.2.1.2. Target Properties – Import Options 48
4.2.1.3. Target Properties – Import Options II 51
4.2.1.4. Target Properties – Inbox (Pending) Approvers 52
4.2.1.5. Target Properties – Outbox Approvers 52
4.2.1.6. Target Properties – Analysis Types 53
4.2.2. Defining Transport Paths 53
4.2.3. Assigning Targets to a Transport Path 55
4.2.4. Defining Target Roles 55
4.2.5. Defining Transport Schedules 56
4.2.6. Defining Projects 60
4.2.7. Defining Groups 60
4.2.8. Defining Types 61
4.2.9. Defining Text Fields 61
4.2.10. Defining Custom Fields 62
4.2.11. Defining Custom Tabs 64
4.2.12. Defining Mandatory Fields 65
4.2.13. Defining Administrators 65
4.2.14. Defining Priority Approvers 65
4.2.15. Defining Custom Labels 66
4.2.16. Defining Business Task Statuses 66
4.2.17. Other Configuration Options 68
4.2.18. Check Configuration 70

5. Advanced Configuration Topics 72
5.1. Automated Analysis Checks 72
5.1.1. General Analysis 72
5.1.1.1. Check Dependencies (0030) 73
5.1.1.2. Overtake and Regression Checks (0031) 73
5.1.1.3. Check Locked Transport Forms (0032) 74
5.1.1.4. Check Authorisations (0033) 75
5.1.1.5. Check Transport Release (0034) 75
5.1.1.6. Conflict Analysis (0035) 75
5.1.1.7. Check Merge Origin (0036) 76
5.1.1.8. Check Manual Steps (0037) 76
5.1.1.9. Check Manual Activities (0039) 76
5.1.1.10. Check future import date/time (0056) 76
5.1.2. ShiftLeft Analysers 77
5.1.2.1. Analysis Type Configuration 77
5.1.2.2. Risk Guard (0001) 78
5.1.2.3. Dev Enforcer: Security (0004) 82

5.1.2.4. Dev Enforcer: Performance (0006) ... 83
5.1.2.5. Dev Enforcer: Standards (0016) ... 84
5.1.2.6. Dev Enforcer: Naming Conventions (0003) ... 84
5.1.2.7. Check Valid To Date (0007) ... 85
5.1.2.8. Check Don’t Approve Before Date (0008) ... 85
5.1.2.9. Check Releasability (0009) ... 86
5.1.2.10. Check Date (0012) ... 86
5.1.2.11. Conflict Analysis (0005) ... 86
5.1.2.12. BW Conflict Analysis (0013) ... 87
5.1.2.13. CTS+ Conflict Analysis (0011) ... 87
5.1.2.14. MDM Conflict Analysis (0025) ... 88
5.1.2.15. Check Transport Release (0014) ... 88
5.1.2.16. Check Unicode (0015) ... 88
5.1.2.17. Check Unconditional Modes (0021) ... 88
5.1.2.18. Show Future Manual Steps (0022) ... 89
5.1.2.19. Deep Impact Analysis (0023) ... 89
5.1.2.20. Previous Import Errors (0024) ... 90
5.1.2.21. Check Import Order (0026) ... 90
5.1.2.22. Lock Control Point / Import Queue (0027) ... 91
5.1.2.23. Check Own Changes (0028) ... 91
5.1.2.24. Check SAP Objects and OSS Notes (0038) ... 91
5.1.2.25. Check Calendar (0040) ... 92
5.1.2.26. Check for Local Non-Transportable Requests (0041) ... 92
5.1.2.27. Dual Domain Controller Overview (0042) ... 92
5.1.2.28. Test Impact Radar (0043) ... 93
5.1.2.29. Check Unreleased Tasks in Requests (0044) ... 93
5.1.2.30. Check Request Tasks not yet in a TOC (0045) ... 94
5.1.2.31. TOC Analysis (0046) ... 94
5.1.2.32. BPCA (0047) ... 94
5.1.2.33. Changes to Same Objects (0048) ... 95
5.1.2.34. Allowed Objects Check (0049) ... 95
5.1.2.35. Disallowed and Critical Objects Check (0050) ... 96
5.1.2.36. Check Documentation (0051) ... 96
5.1.2.37. Unit Test automation (0052) ... 97
5.1.2.38. Check Component Version (0053) ... 97
5.1.2.39. Changes to same objects in path (0055) ... 97
5.1.2.40. Check Custom Field entered (0057) ... 98
5.1.2.41. SCC1 Client Copy Check (0058) ... 98
5.1.2.42. Check sibling system imports (0059) ... 99
5.1.2.43. Deep Impact Analysis (New) (0060) ... 99
5.1.2.44. Critical Impact Analysis (0061) ... 100
5.1.2.45. Check Interdependencies in Path (0062) ... 101
5.1.2.46. Approver SOD Check (0063) ... 101
5.1.2.47. Check SAP Import (0064) ... 102
5.1.2.48. Integration Status Check (0065) ... 102

5.1.2.49. Version Comparison (0066) 103
5.1.2.50. Impacted Batch Jobs (0067) 104
5.1.3. Custom Analysis Types 105
5.2. Rules Engine 105
5.2.1. Introduction 106
5.2.2. Core Configuration 107
5.2.2.1. /BTI/TE_RE_CONSE 109
5.2.2.2. /BTI/TE_RE_STEP 109
5.2.2.3. /BTI/TE_RE_RULES 111
5.2.2.4. /BTI/TE_RE_STEPC 111
5.2.2.5. /BTI/TE_RE_SYSTP 114
5.2.2.6. /BTI/TE_TVARV 114
5.2.3. Consumers 114
5.2.3.1. Skipping (/BTI/TE_RE_SKPCP) 114
5.2.3.2. Approvals (/BTI/TE_RE_APPRV) 115
5.2.3.3. Consumer Configuration Examples 116
5.2.4. Rules Test Harness 118
5.2.5. Troubleshooting Issues 119
5.2.6. Advanced Topics 120
5.2.6.1. Ranges 120
5.2.6.2. Contains 120
5.2.6.3. Reusable Steps 121
5.2.6.4. Custom Node Types 122
5.2.6.5. Filters 122
5.2.6.6. Calculated Custom Fields (CalcCF) 122
5.2.6.7. Application Tables 123
5.3. Skipping 124
5.4. Merge 125
5.5. Inline Conflict Analysis 128
5.6. Inline Risk Analysis 128
5.7. Backout 130
5.8. Release Orchestration 131
5.9. Partial Testing 132
5.10. Automated Transport Naming Convention 132
5.11. Automated ‘Exception-based’ Approvals 134
5.12. Additional Approvals for specific Objects 134
5.13. Document Categories 135
5.14. Documentation Links 136
5.15. Manual Activities and Non-Transport Deployments 137
5.16. External Transports 138
5.17. Delegation 139
5.18. Non-ABAP / Java transports and CTS+ 140
5.19. Transport of Copies (Testing TOCs) 141
5.20. Transport of Copies (Production TOCs) 142
5.21. Unit Test Automation 143

5.22. Diffuser / Object Linkaging 144
5.23. Test Impact Radar 145
5.24. Integration Framework 147
5.24.1. Integration Scenarios 148
5.24.2. Integration Process Flow 149
5.24.2.1. Inbound Integration Process 150
5.24.2.2. Outbound Integration Process 150
5.24.3. Outbound Integration 152
5.24.3.1. Integration – Configuration Overview 152
5.24.3.2. Detailed Configuration Table Information 152
5.24.3.2.1. External System(s) 153
5.24.3.2.2. Update Processes 153
5.24.3.2.3. Integration Points 154
5.24.3.2.4. Conversions 154
5.24.3.2.5. Notification Users 155
5.24.3.2.6. Mapping 156
5.24.4. Inbound Integration 157
5.24.4.1. Inbound Process Flow 157
5.24.4.2. Web Services 158
5.24.4.2.1. Create a business task 158
5.24.4.2.2. Change a business task 160
5.24.4.2.3. Read a business task 161
5.24.4.2.4. Start the analysis for a business task 162
5.24.4.2.5. Read the results of an analysis run 163
5.24.4.2.6. Approve a Business Task 166
5.24.4.2.7. Enter the test results for a business task 167
5.24.4.2.8. Mapping internal values 168
5.24.4.2.9. Other communication techniques 169
5.24.5. ServiceNow Integration 170
5.24.6. JIRA Integration 170
5.24.7. BMC Remedy Integration 170
5.24.7.1. Web Service User 171
5.24.7.2. Remedy foundation data 173
5.24.7.3. Custom Mappings 176
5.24.7.4. Remedy Web Service trigger code 177
5.24.7.5. Web Service Interface form 178
5.24.7.6. Setup of Web Service user 179
5.24.7.7. Web Service Workflow setup 179
5.24.7.8. Use of values returned from a Web Service call 182
5.24.7.9. Consumption of the Task Create Web Service 182
5.24.7.10. Consumption of the Remedy Create Change Work Info Web Service 183
5.24.8. ChaRM integration 183
5.24.9. Integration based on Custom Field 184
5.25. Airgap 185
5.26. Consolidated Import Queues 189

5.27. Client-based Transport distribution 190
5.28. Additional Metadata 191
5.29. Reports 192
5.30. Domain Controller Utility Programs 195

6. Deployment Considerations 198
6.1. Windows Client and RFC 198
6.2. E-mail Integration 198
6.3. TMS Configuration 198
6.4. User Exits 199
6.5. Windows GUI Command Line Parameters 205
6.6. Language Support 205

7. Housekeeping Activities 207
7.1. Recommended Housekeeping Activities 207
7.2. ActiveControl Data Archiving 209

8. Upgrading ActiveControl 212
8.1. Upgrade Considerations 212
8.2. Upgrade Steps 213

9. Software Support 216
9.1. Online FAQ Forum 216
9.2. Support from Basis Technologies 216
9.3. Updating ActiveControl License Keys 217
9.4. Uninstalling ActiveControl 218

10. Legal Statement 219

1. Introduction

The ActiveControl Administration Guide is aimed at the Administrators within our customer's organisations who are responsible for maintaining ActiveControl.

This Administrator Guide is divided into eight sections:

1. Key Concepts
2. Installation
3. Core Configuration
4. Advanced Configuration
5. Deployment Considerations
6. Housekeeping
7. Upgrading ActiveControl
8. Support Information

2. Key Concepts of ActiveControl

This section of the Administration Guide details some of the key concepts of ActiveControl

1. Architecture
2. Transport Forms, Business Tasks and Projects
3. Inboxes, Outboxes and Test Queues
4. Transport Sequencing & Dependencies

2.1. Architecture of ActiveControl

The architecture of ActiveControl can be broken down into several core components:

1) a main **Domain Controller**

2) other **participating satellite SAP systems**.

3) **Access methods** – consisting of a Windows GUI client software, Web UI and SAP GUI screens.

4) **Integration Framework**

The rest of this section summarises each of these in turn.

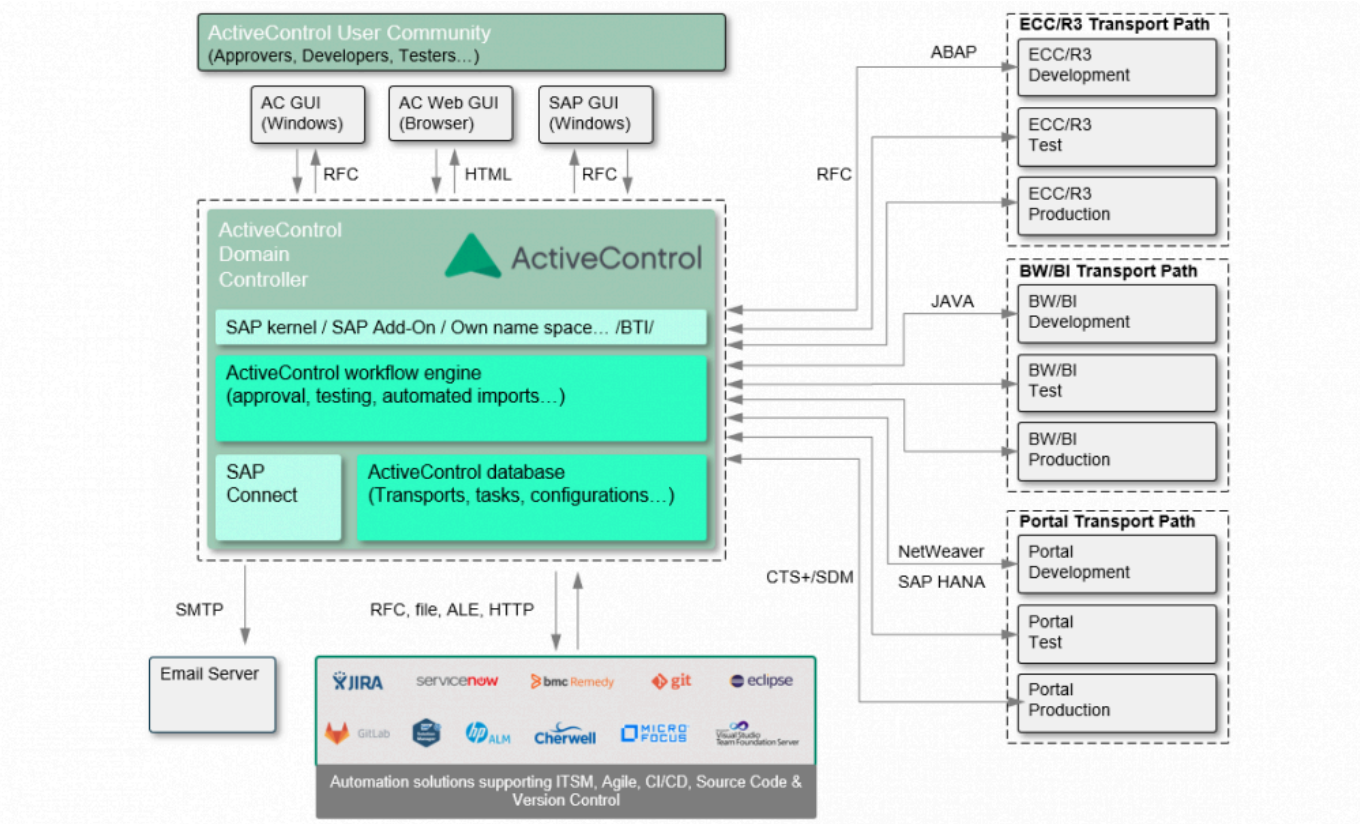


Figure: The ActiveControl architecture

2.1.1. Domain Controller

Like the SAP Transport Management System (TMS) , 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 transport request information and to perform transports. These connections are made using SAP's remote function call (RFC) protocol.

There is no requirement for the ActiveControl domain controller and TMS domain controller to be the same SAP system. In fact, a single ActiveControl domain controller can manage transports for multiple TMS domains.

All ActiveControl users will require a user account in the domain controller system with a valid email address in order to receive email notifications. Each user must also be assigned with the relevant ActiveControl single or composite role(s) to reflect their job function and the access they need to the various functionality within ActiveControl.

> ✻ It is generally recommended that a **Solution Manager system** be used as the ActiveControl domain controller but this is not mandatory; the reality is the domain controller can be any ABAP system with high availability and in which all the ActiveControl users have an SAP user account.

2.1.2. Satellite SAP Systems

The ActiveControl server software must also be installed in each of the participating satellite SAP systems that will be managed by ActiveControl.

It is not necessary for any participating SAP systems to share transport directories with any of the other SAP systems in the ActiveControl system landscape. If transport files are missing, the ActiveControl domain controller automatically requests the transport files, on demand, from the source SAP system and forwards them to the SAP system that the changes are to be imported into.

The only requirement is that the ActiveControl domain controller must be able to connect to each of the participating SAP systems via the remote function call (RFC) protocol. This RFC connectivity is required to gather transport request information, release transport requests or to perform import actions.

Example 1: Multiple Development Streams with Separate Transport Directories

ActiveControl can support a system landscape with three development streams (such as ERP, CRM and BW), where each development stream has its own transport directories. That is, the development,

test, production (etc.) ERP systems share a common set of transport directories, however these transport directories are not shared with any of the CRM or BW systems.

In this example, the ERP development system might be acting as the ActiveControl domain controller for all three development streams, even though none of the SAP systems in the ERP stream share transport directories with either the CRM or BW systems (and vice versa).

Example 2: Single Development Stream with Multiple Transport Directories

ActiveControl also supports a system landscape where the SAP systems of a single development stream do not share the same set of transport directories. It might be that the development and test systems share a common set of transport directories, but for security reasons, the production system has its own set of transport directories.

2.1.3. Accessing ActiveControl

There are three main ways of accessing ActiveControl:

1) **SAP GUI screens**, normally used by 'transport owners' (ie Developers, Functionals, Securities teams) to log their changes within ActiveControl.

2) **Web UI interface**, typically used by the majority of users such as Approvers, Testers and other stakeholders to perform their activities relating to ActiveControl.

3) **Windows GUI client**, generally only deployed to 'heavy users' such as Basis teams, Change Managers and ActiveControl Administrators.

The Windows GUI

The Windows GUI client software is a relatively small application occupying less than 15MB of space. The software can be installed locally to each person's computer, or alternatively to a file server location for easier deployment. The client software should run on any computer capable of running SAP's standard (32-bit or 64-bit) SAP GUI software.

The Windows client software connects to the ActiveControl domain controller via SAP's remote function call (RFC) protocol. ActiveControl does not store any log-on information, including the user passwords.

The logon details of client, user ID and password are the same as if the person were logging into the ActiveControl domain controller via the SAP GUI. Since all of ActiveControl's configuration and application data is client independent, it does not matter which client of the ActiveControl domain controller is specified.

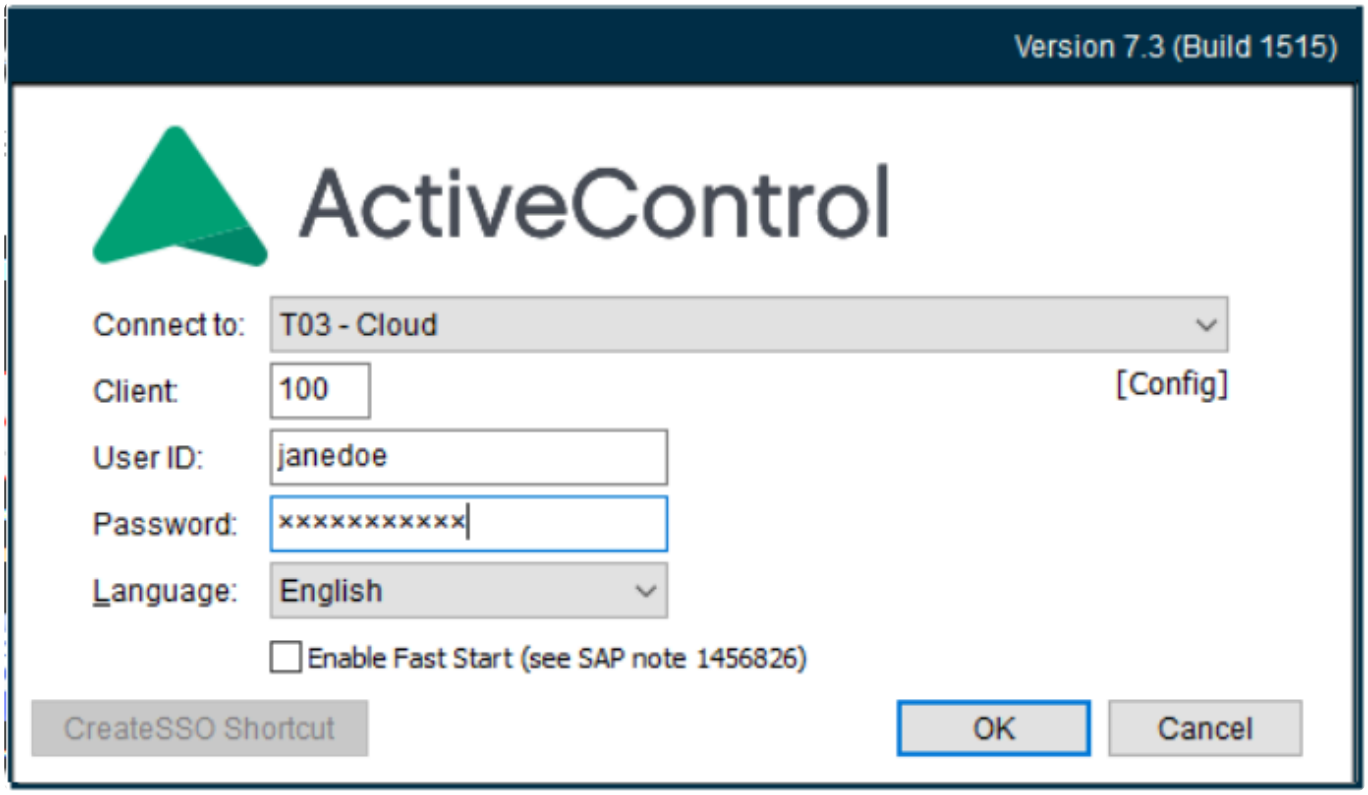


Figure: The Windows GUI login screen

Remote function call is a TCP/IP sockets-based protocol that utilises the 3300-3399 range of port numbers. For example, if the ActiveControl domain controller has system number 10, then the client software will communicate with it using TCP port number 3310.

Although the Windows GUI client software does run on older 32-bit operating systems such as Windows NT 4.0, Windows 98 and Windows XP, only Windows version from Windows 8onwards are officially now supported by Basis Technologies.

Connecting Systems

The ActiveControl Windows GUI uses the saplogon.ini files (for SAP GUI 730 and earlier) and XML file for SAP GUI 7.40 onwards, on the local PC to determine this list of systems to connect to. ActiveControl reports the presence of both saplogon.ini and SAPUILandscape.xml files in:

1) Local ActiveControl folder (typically C:\Program Files (x86)\Basis Technologies\ActiveControl)

2) Windows directory

3) Directories stored in the registry by SAP GUI
Current User Local: HKEY_CURRENT_USER\Software\SAP\SAPLogon\ConfigFilesLastUsed\ConnectionConfigFile

Current User Server: HKEY_CURRENT_USER\Software\SAP\SAPLogon\ConfigFilesLastUsed\ConnectionConfigFileOnServer
Local Machine Local: HKEY_LOCAL_MACHINE\Software\SAP\SAPLogon\ConfigFilesLastUsed\ConnectionConfigFile
Local Machine Server: HKEY_LOCAL_MACHINE\Software\SAP\SAPLogon\ConfigFilesLastUsed\ConnectionConfigFileOnServer

4) SAPLOGON_INI_FILE environment variable (saplogon.ini only)

E.g. “C:\Program Files\Basis Technologies\ActiveControl\ActiveControl.exe” “SAPLOGON_INI_FILE=C:\Windows\saplogon.ini”

5) %APPDATA%\SAP\Common folder (XML only) – this is the default location for the ‘local configuration path’.

> ! Please be aware that ActiveControl will continue to look at old saplogon.ini files if the customer moves over to xml config files and the historical ini. file still exists, so users may need to check that both ActiveControl and SAP are using the same configuration files after doing a SAP GUI or ActiveControl upgrade.

2.1.4. Integration Framework

ActiveControl include an Integration Framework to allow easy bi-directional communication with other ITSM tools.

Out-of-the-box Integrations are already available for tools such as ServiceNow, JIRA, GitLab and HP ALM.

The capabilities and general principles of the ActiveControl Integration Framework are detailed in the Integration Framework section of this Administration Guide. Our existing out-of-the-box Integrations are detailed in separate Administration Guide documentation, available here.

2.2. Transport Forms, Business Tasks and Projects

ActiveControl has a 3-level hierarchy concept, consisting of Transport Forms, Business Tasks and Projects.

Before a transport request can be transported using ActiveControl, a **Transport Form** must be created for the transport request. The Transport Form is a simple form that can typically be completed in 10-15 seconds. It is used to document and categorise the changes made, so that Approvers, Testers and Auditors can immediately understand what the Transport is for.

The Transport Form can be used to associate the technically-oriented transport request with the corresponding business issue and/or requirement to which it relates. Within ActiveControl, this is referred to as a **Business Task**. Although it can be an optional association, it is strongly recommended that all Transport Forms be associated with Business Tasks, as this:

- Groups together technically separate but functionally-related transport requests that collectively implement the changes necessary to solve / deliver individual business issues and requirements.
- Allows for the progress and testing of the solution to a business issue or requirement throughout the system landscape to be monitored and reconciled as the solution makes its way through one or more development streams (such as in the case of branched development systems).
- Provides the business context for business users and team leaders involved in the change process (for example, from an approval perspective).

Projects within ActiveControl are used to define a logical grouping of Business Tasks. Projects are typically used within most customers for grouping Business Tasks and Transport Forms into Releases, Upgrade Projects, SAP Projects etc.

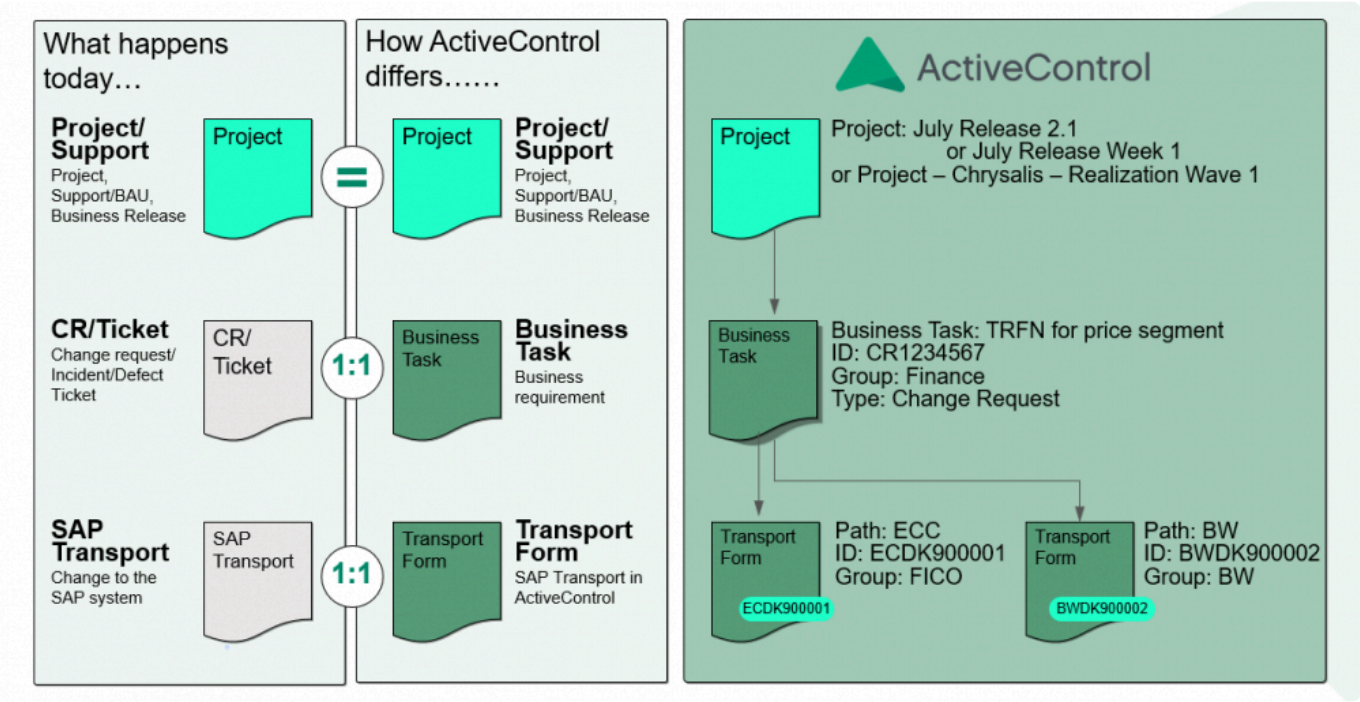
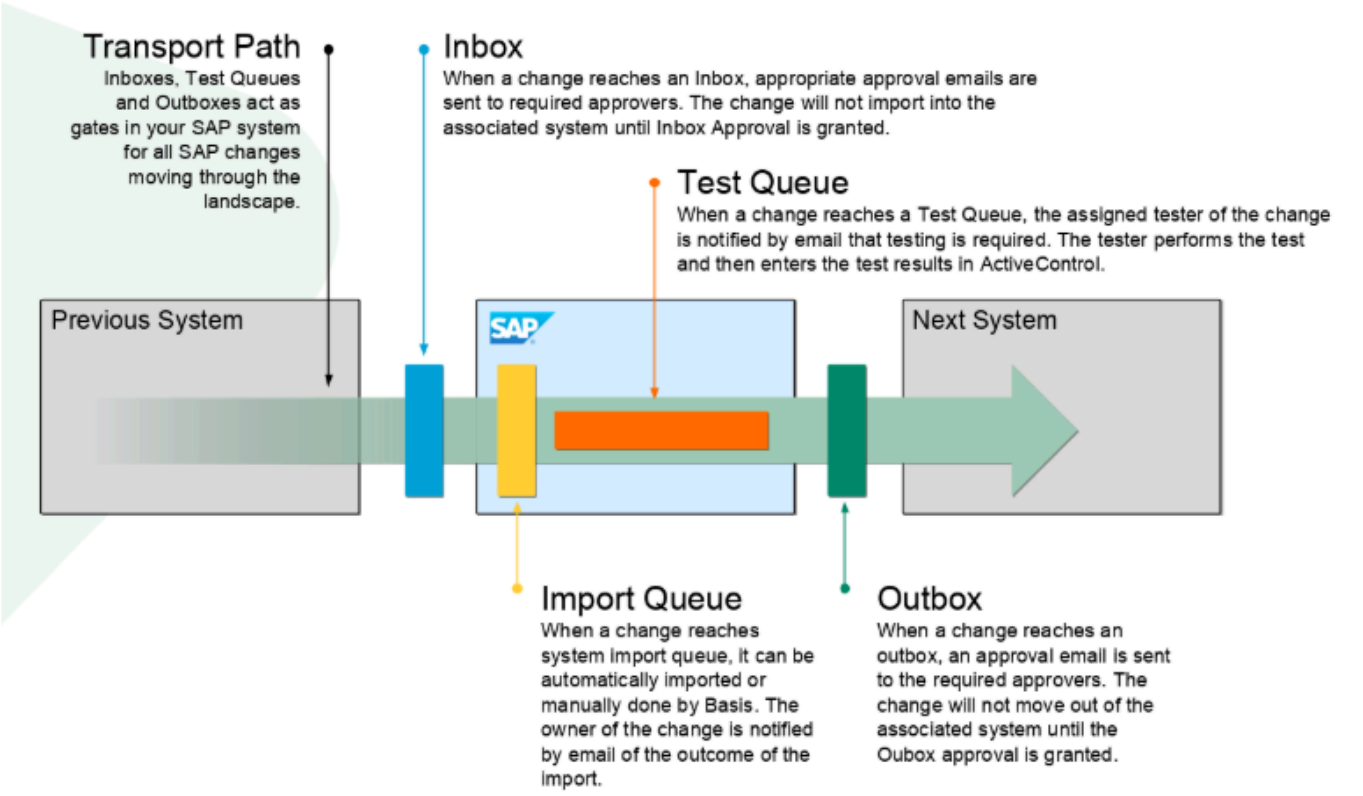


Figure: The ActiveControl 3-level hierarchy: Projects, Business Tasks and Transport Forms

2.3. Inboxes, Outboxes and Test Queues

The ActiveControl workflow consists of **Inbox**, **Test Queue** and **Outbox** control points where an approver(s) is responsible for approving the SAP change / transports. Until this approval is done, the transports won't move forward in the transport path. These Inbox, Test Queue and Outbox control points are completely configurable, depending on an SAP customer's approval requirements.



The ActiveControl approvals workflow: Inboxes, Outboxes and Test Queues.

> Inbox and Outbox approvers are maintained by Administrators within the ActiveControl Windows GUI, and are based on the Transport Form [Group]. Test Queue testers on the other hand, are governed by assignments on individual Business Tasks, done in the SAP GUI, Windows GUI and Web UI by any authorised user.

2.4. Transport Sequencing and Dependencies

The default sequencing within ActiveControl is based on the release date/time of the transports.

For example, if transport A is released at 15:00 on 1st January 2019, and transport B is released at 16:00 on 1st January 2019, and both transports are sitting in an Import Queue, then ActiveControl would ‘order’ Transport A to be imported before transport B.

It is possible to change this default sequencing by hardcoding **Dependencies** between transports. This is useful if transports were inadvertently released in the wrong order, or indeed when there is a need to ensure sequencing of transports across different SAP systems.

Transport Dependencies

ActiveControl allows for inter-dependencies to be defined between transports in the same system (for example, if a functional Customising transport is dependent on a particular Workbench transport), and also for cross-system dependencies between transports being deployed in different SAP development

systems (for example, if a BW transport is dependent on a particular ECC transport)

In both cases, these Dependencies are defined on the [Advanced Options] tab of the Transport Form in the SAP GUI, Windows GUI or Web UI.

Technically, when ActiveControl checks a dependency where transport A is dependent on the prior import of transport B, it checks whether transport B has been applied to the SAP system that transport A is being applied to, or that transport B has been applied to at least one other SAP system with the same role as the SAP system that transport A is being applied to.

> In order to enforce cross-system dependencies, ActiveControl must be able to equate different SAP systems that serve the same purpose, for example an ECC Test system and a BW Test system. This is achieved in ActiveControl by assigning the SAP systems to a common ‘Target Role’. Please refer to the ‘Defining Targets’ and ‘Defining Target Roles’ sections of this Administration Guide for more information about creating target roles and assigning them to target SAP systems.

3. Installing ActiveControl

ActiveControl is an ABAP add-on, and as such is straightforward to install.

The server software is contained within a single transport and the Windows client software is installed using a standard setup program.

This section provides information on how to perform the initial installation of ActiveControl.

3.1. Installing the Server Software

The following steps summarise the activities involved in installing the ActiveControl server software. The steps assume the system ID of your development SAP system is DEV, and also that the reader is already comfortable with the basics of the SAP Correction and Transport System.

Step	Activity
1.	Copy the transport file K9nnnnn.SID to the cofiles transport directory. As with the following transport files, make sure that the file is copied in binary mode and for UNIX systems ensure that the target filename is in uppercase and is not read-only.
2.	Copy the transport file R9nnnnn.SID to the data transport directory.
3.	Add the transport to the import buffer of the SAP system that you have selected to be the ActiveControl domain controller. *tp addtobuffer SIDK9nnnnn DEV*
4.	Import the transport into the ActiveControl domain controller SAP system (note the use of unconditional modes 1 and 8). *tp import SIDK9nnnnn DEV client=000 U18*
5.	Repeat steps 3 and 4 for the other participating SAP systems in your system landscape.

3.2. Removing the Server Software

SAP does not provide a simple method for removing the contents of a transport from a system. To mitigate this, Basis Technologies can provide you with a backup transport to remove the contents of ActiveControl from your SAP systems in the unlikely event that you no longer want to use the product. This backup transport has been created using ActiveControl's innovative backup capability, a powerful feature described in a later section of this Administration Guide.

Please contact support@basistechnologies.com if you require a backup transport to remove ActiveControl from your SAP systems.

3.3. Installing the Client Software

The ActiveControl client software is installed using a Windows setup program. This is done by running the supplied setup program and following the on-screen instructions.

When ActiveControl is installed on a local or network directory it creates the required files in a directory such as **C:\Program Files\Basis Technologies\ActiveControl**. The ActiveControl.exe file will be located within this directory.

The ActiveControl.exe executable requires access to a sub-directory called “View Layouts” (so “C:\Program Files\Basis Technologies\ActiveControl\View Layouts\”. Within this sub-directory are a number of layouts for each type of control point. It is important that when the ActiveControl.exe is run it is run in the installed directory as it requires the “View Layouts” sub-directory to run properly otherwise errors will occur. A shortcut that points to the ActiveControl.exe in this location will work ok.

When creating a shortcut to ActiveControl , you can also choose to default the details for the ActiveControl domain controller that the client software will connect to. As an example, the following shortcut command-line defaults ActiveControl to connect to the SAP system with SAP Logon ID ‘Production Support Development’ and with logon client 100:

…\ActiveControl.exe SAPLOGON_ID=”Production Support Development” CLIENT=100

3.4. Removing the Client Software

The ActiveControl client software can be removed by opening the Windows “Add or Remove Programs” control panel and choosing to remove the program titled “ActiveControl version…” with publisher Basis Technologies.

3.5. RFC Destinations

ActiveControl uses SAP’s remote function call (RFC) technology for communication. RFCs are required in two places:

1) from the Domain Controller to every participating satellite SAP system being managed by ActiveControl

2) from the Development satellite systems back to the Domain Controller.

In essence, an RFC destination must be defined for every target SAP system that the ActiveControl Domain Controller communicates with (including the Domain Controller itself), and also from the satellite Development systems back to the Domain Controller. RFCs are NOT required from non-Development satellite systems such as QA and Production back to the Domain Controller.

RFC Destination Settings

RFC destinations are maintained with SAP transaction SM59. An RFC destination contains the technical information required to call a function (program) remotely. When the remote destination is an SAP system, the technical information specifies the target application server and the logon details (client, user ID, password and language).

RFC destination	TRANSPORT EXPRESS [SID]
Connection type	3 – SAP connection
Load balance	No
Target host	Hostname of an application server of the SAP system
System number	The corresponding system number of the application server
Description	Any suitable description (required)
Language	EN for English
Client	The main client of the SAP system
User	User ID in the specified SAP client – see more information below
Password	Password of the specified SAP user ID

RFC Destinations in the ActiveControl Domain Controller

Each RFC destination required by ActiveControl has the name ‘TRANSPORT EXPRESS SID’ where SID is the system ID of the system you are connecting to from the Domain Controller. For example, if your implementation has development, test and production SAP systems called DEV, TST and PRD; then the following RFC destinations must be created in the domain controller:

- TRANSPORT EXPRESS DEV
- TRANSPORT EXPRESS TST
- TRANSPORT EXPRESS PRD

In addition, an RFC destination should be created on the ActiveControl Domain Controller, pointing back to itself. So if your Domain Controller system SID is SOL, then an RFC Destination called TRANSPORT EXPRESS SOL should be created, pointing back to the SOL system.

RFC Destinations in the Development Satellite Systems.

An RFC destination is required in each Development satellite systems, pointing back to the Domain Controller. This RFC destination must have the name ‘TRANSPORT EXPRESS CONTROLLER’.

RFC Users

When defining RFC destinations, Basis Technologies recommend that you create and assign a special user ID in the satellite system called **AC_RFC**. These users need to be slightly different in the Domain Controller versus the participating satellite systems.

RFC User in the Satellite Systems
In the satellite systems, the AC_RFC user must be a System user. A system user ID is a special type of SAP user ID that cannot be used to access the SAP system via the SAP GUI. Therefore by using a system user ID, potential security holes, such as a remote logon to your production system, are avoided. Another advantage of a system user ID is that the password does not expire.

In the satellite systems, the AC_RFC user must have SAP_BC_TRANSPORT_ADMINISTRATOR and /BTI/TE:CTS_RFC roles assigned to it

RFC User in the Domain Controller
In the ActiveControl domain controller, the AC_RFC user must be setup as a Service user. A system user cannot be used as it does not support popup screens like the Transport Form.

In the domain controller, the AC_RFC user needs to have SAP_BC_TRANSPORT_ADMINISTRATOR, /BTI/TE:CTS_RFC,
/BTI/TE:CTS_ADMIN_USER and /BTI/TE:CTS_ADMIN roles assigned to it.



ActiveControl may update the client field of an RFC destination at runtime. This may be necessary when releasing a transport request or creating a backup request as these actions are client-specific. In these situations, the specified user ID and password must be valid for all possible clients.



Remember, the AC_RFC user must be a System User in the satellite systems, but a Service User in the Domain Controller. Not adhering to these user types is one of the most common reasons for unexpected behaviour during an ActiveControl implementation.

3.6. Next Steps

After installing the ActiveControl client and server software and creating the RFCs between the Domain Controller and the satellite systems, you are essentially now ready to begin configuring ActiveControl. The majority (~90%) of this configuration is performed by authorised Administrators in the Windows GUI client software, with the remaining ~10% done in the SAP GUI of either the Domain Controller or satellite Development systems.

When you start the Windows client software, choose the SAP Logon system entry for your ActiveControl domain controller and then your logon details for that SAP system.

The first time that you start the ActiveControl Windows GUI after installing (or upgrading), you may be asked to enter a license key for the ActiveControl domain controller. The license key is determined using the system ID and installation number of the SAP system acting as the ActiveControl domain controller. If you have not already been provided with an ActiveControl License Key, please request one via support@basistechnologies.com

The ActiveControl configuration can only be changed by authorised ActiveControl Administrator(s). The first person to logon to ActiveControl and upload the License Key following its installation is automatically defined as an Administrator. The list of Administrators can be maintained thereafter as required.

The next section of this Administration Guide will explain the core configuration of ActiveControl required in the SAP GUI and Windows GUI.

4. Configuring ActiveControl

This section of the Administration Guide details the steps to configure the core functionality of ActiveControl. It is split into two sections:

1) SAP GUI configuration

2) Windows GUI configuration

> The separate ActiveControl Quick Setup Guide offers a simple step-by-step summary of how to configure ActiveControl. For most new Administrators, the Quick Setup Guide is a good place to start to gain familiarity with an ActiveControl implementation, and also to remind yourself what has been installed and configured.

4.1. SAP GUI Configuration

4.1.1. Completing Transport Forms and Business Tasks in the SAP GUI

ActiveControl has the ability to trigger a **Transport Form popup** in the SAP GUI of a satellite Development system, at one or two places:

1) when the transport is released via SE09/SE10 etc, or

2) when objects are attached to a transport via SAP transactions such as SE38, SE16, PFCG etc

Via the Transport Form popup, the Transport Owner (ie Development, Functional, Securities and Basis teams) can log some key information about their transport, and then also associate it with a new or existing Business Task. It is via these screens in the SAP GUI that most ActiveControl customers enforce all SAP change to be registered within ActiveControl at the start of the Development process.

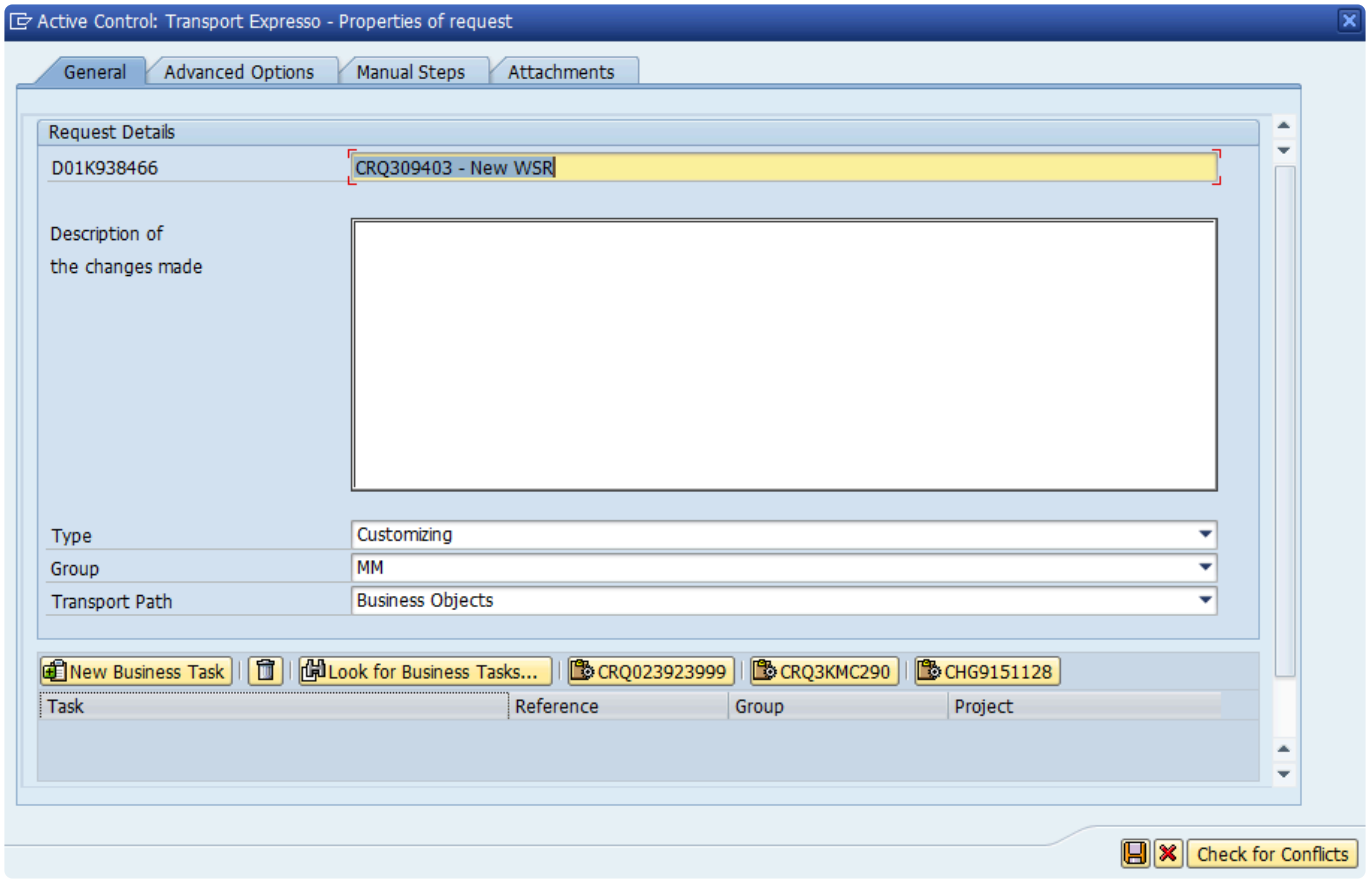


Figure: Transport Form screen in the SAP GUI

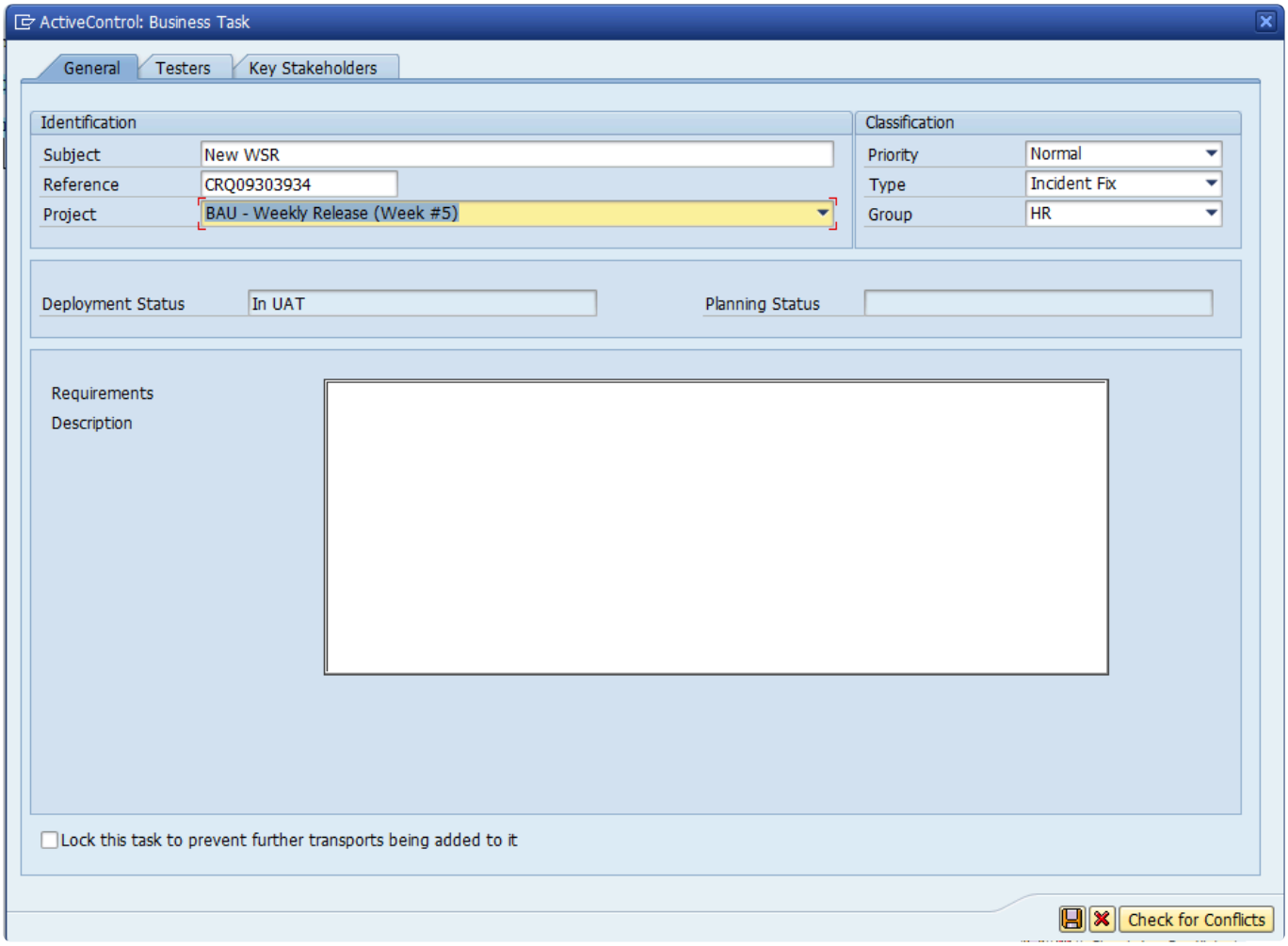


Figure: Business Task screen in the SAP GUI

> There are no ActiveControl screens in Java systems such as Portal or PI. For any non-ABAP systems managed by ActiveControl using the prerequisite CTS+, the Transport Form should be created via the ActiveControl Web UI (or Windows GUI).

4.1.1.1. Switching on SAP GUI Processing

As mentioned in previous section, ActiveControl has the ability to trigger a Transport Form ‘popup’ in the SAP GUI of the Development system, either when the transport is released or when objects are attached to a transport.

Other active functions are also available, depending on a customer’s exact requirements and process:

Active Function	Description
Transport is released (ON_RELEASE)	This uses /BTI/TE_BADI_TR_FORM BADI in the Development system, and comes active. No additional setup required.
Transport is first used (ON_CREATE)	This functionality uses an enhancement spot. No additional setup required.
In-Line Conflict Analysis	This functionality switches on Inline Conflict Analysis, to identify conflicts across parallel development systems. Please refer to Inline Conflict section for details on

(INLINECONF)	the other configuration required to use this.
Show object keys in In-Line Conflict Analysis (INLINEKEY)	This option results in Inline Conflict Analysis including object keys in the results. If this is activated, then In-Line Conflict Analysis active function above must also be active and configured.
Cache Transport Object details (OBJCACHE)	
Allow TF Creation by-pass (BYPASS)	This option can be used to bypass the Transport Form creation process in a particular system.
Unit test on release (UNIT_TEST)	This option switches on Automated Unit Testing within the SAP GUI. Note that there is some configuration that needs to be done as well, this is detailed in later 'Unit Test Automation' section.
Show Transport Form popup only when configured objects added (OBJ_DEPEND)	This option can be used by customers that only want a Transport Form to be required for certain objects. Other configuration is needed to activate this functionality, please refer to online forum for details.
Object Segregation (OBJ_SPLIT)	This option can be used by customers that want to ensure certain objects are not mixed in the same transport. Other configuration is needed to activate this functionality, please refer to online forum article for further details.
Prevent releasing the transport (PREVENTREL)	This option can be used to hide the "Do you want to release.." screen in the SAPGUI that is seen after populating the Transport Form using the ONRELEASE active function. It is useful for customers that want to prevent Developers prematurely releasing Transports before other approvals are performed.

The ActiveControl active functions are configured using table **/BTI/TE_ACTIVE** in the ActiveControl domain controller system. They are configured per satellite Development system in which they will be active.

System ID	ActiveFunction
Development SID	The active function(s) which is required to be switched on for that Development system)

There are numerous possible reasons for users not getting the Transport Form popup in the SAP GUI during an ActiveControl implementation. These are detailed in our online forum article here

To stop the SAP GUI screens from timing out when entering a Transport form or Business Task, it is recommended to increase the rdisp/max_hold_time parameter on all application servers of the Domain Controller controller and relevant SAP development

systems. The recommended value for this is **360**.

The ability to create a Business Task from the Transport Form is controlled by the authorization object Y_TE_TASK with the value CREATE. If you want the developer to assign the Transport Form to an existing Business Task, but not to be able to create a new one, then restrict the authorization by removing the value CREATE.

4.1.1.2. Configuring SAP GUI Processing

The activation of the SAP GUI processing for users is controlled via table **/BTI/TE_CONTROL** in each satellite Development system.

An ‘Active’ entry must exist for the relevant users for the processing to occur. It is possible to either switch on ActiveControl for only certain specific users by entering their username (this is typically done during the initial implementation phase so that certain users can test the product), or to switch on ActiveControl for ALL users by maintaining an entry with a blank username.

Do not add entries in /BTI/TE_CONTROL for users that do not yet have any ActiveControl authorisations assigned to them in the domain controller. Doing this will result in the users getting errors in the Development system when doing their work, and potentially being unable to save their work.

4.1.1.3. Transport Form Organizer

The ActiveControl **Transport Form Organizer** screen can used to access and maintain Transport Forms of any released or modifiable transports. This is accessed in the SAP GUI of the ABAP Development system via transaction **/n/BTI/TE_TR_FORM**. The Transport Form Organizer can be used to update a Transport Form after the transport is released, for example if a Developer wants to add a Manual Step or Dependency to a Transport Form they completed earlier, or to update the previously populated information.

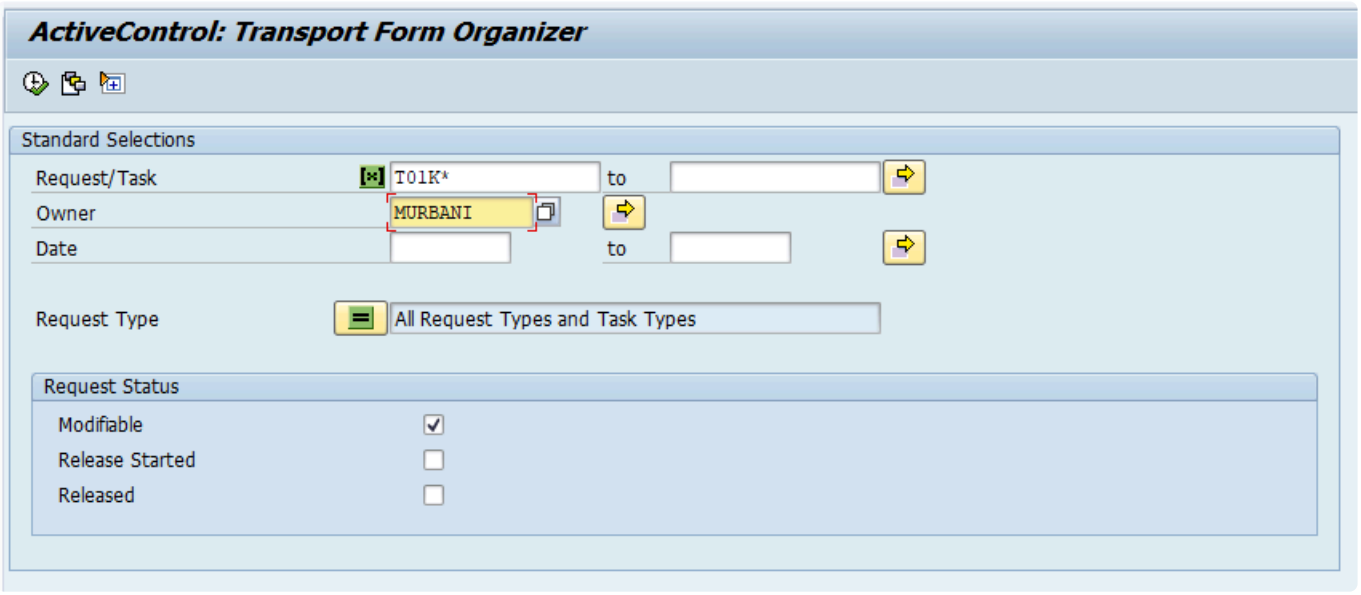


Figure: The Transport Form Organizer

Double clicking a transport in the result view of the Transport Form Organizer opens the existing Transport Form. If no Transport Form assignment already exists, an empty Transport Form can be maintained.

> Use of /n/BTI/TE_TR_FORM will require an ActiveControl authorization role in the Development system to be assigned to the person wanting to run it. The authorisation to access is included in the standard role /BTI/TE:CTS_USER.

4.1.1.4. Opening the Windows GUI from SAP GUI

It is possible to open the Windows GUI from within the SAP GUI backed of the ActiveControl Domain Controller by using transaction **/BTI/TE_STARTGUI**.

This can be useful for ActiveControl customers that use SSO and want to be able to trigger the Windows GUI via a central porta webpage.

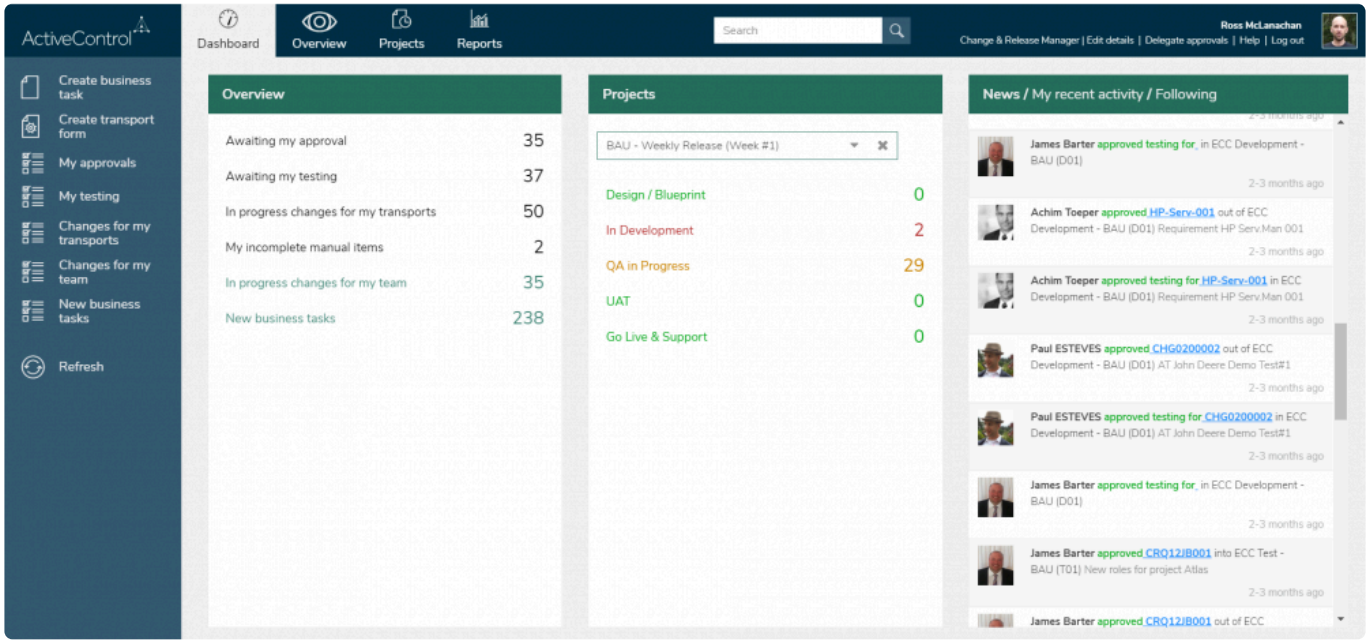
4.1.2. Web User Interface

ActiveControl includes a Web UI that allows access to key functionality via a web browser such as Internet Explorer and Chrome. The ActiveControl Web UI provides a dashboard based overview of projects and items requiring action, allowing individual users with a means of easily identifying their own current workload within the tool.

Users with the appropriate authorisations are able to access the following functionalities within the Web Interface:

- Ability to create SAP transports.
- Ability to create and maintain Transport Forms, Business Tasks and Projects.
- Ability to search for Business Tasks and Transports
- Ability to run ActiveControl Reports
- Visibility and analysis / approval / rejection of items that require your approval
- Visibility and test results entry of items that require your testing
- Visibility of tasks for your team / project
- Visibility of RAG status for Projects milestone phases
- Ability to follow specific Business Tasks, Transports and SAP Objects.
- A news feed to show recent events in ActiveControl – ie activity performed both the user, and by other people.

!{IMAGE-LINK+



}!

Figure: ActiveControl Web UI

4.1.2.1. BSP Services

To install the ActiveControl WebUI, the following services BSP need to be activated in the domain controller via transaction SICF:

- default_host -> BTI -> te_web_services
- default_host -> BTI -> tessocntl
- default_host -> sap -> bc -> bsp -> bti -> te_bsp_new

Other general BSP services may also need to be activated to enable the web interface to work.

- All subnodes in default_host -> sap -> public -> bsp -> sap

The following may also be required to enable the ActiveControl Reports to work in the Web UI:

- default host -> sap -> bc -> gui -> sap -> its -> webgui

By activating the services, this exposes the ActiveControl Web UI via the HTTP protocol to the ActiveControl domain controller server.

Once activated, you can test the ‘te_bsp_new’ service to determine the URL:

- http://[acdomaincontroller]:80[systemnumber]/sap/bc/bsp/bti/te_bsp_new/index.html

You will be prompted for a logon to the ActiveControl domain controller SAP system, and should use your SAP username and password. Once logged on, you will be able to access the WebUI functionality. This can be used simultaneously with the ActiveControl Windows GUI if you also have that installed.

> ! If the Domain Controller system is setup as a dual stack system (system/type = DS) then by default all /sap URLs get directed to the ABAP stack whereas everything else is sent to the Java stack. To mitigate this, the /bti URLs need to be added to the ICM URL prefix table so the ActiveControl web UI is directed to the ABAP stack. This is done in the ICM config transaction SMICM. This can be found in the following possible menu paths:
> Goto -> HTTP Server -> Load URL Prefixes
> Goto -> HTTP Plug-In -> Application Server -> Load URL Prefixes
> In both cases above use the Display Data option to view the current settings. /bti/ should be displayed in the URL Prefix Table.

> ✻ For Single Sign On (SSO) rather than using the index.html page above please use the main.html page instead as this will bypass the login screen and use the current user logon ticket. Please note that this obviously requires SSO to have been setup within your ActiveControl Domain Controller as a pre-requisite.

4.1.2.2. Web UI – Metrics

By default, all ActiveControl users are allocated with a*Default Web User Role* that provides access to the actions and metrics that are seen within the Web UI. The default metrics that are available within ActiveControl WebUI are:

Actions:

- View all changes awaiting my approval
- View all changes awaiting my testing
- View all my in progress changes
- View all my incomplete manual items
- View all in progress changes for my team

Metrics:

- Awaiting my approval
- Awaiting my testing
- My in progress changes
- My incomplete manual items
- In progress changes assigned to my role
- In progress changes for my team
- New business tasks

To maintain other roles to alter the actions and metrics that are available to users, the following tables must be configured. More detail is provided about these optional configuration tables later in this section

Table	Description
/BTI/TE_WEBUICFG	General web UI parameters / config table.
/BTI/TE_USERPREF	Web UI User preferences table.
/BTI/TE_ROLEACT	Role actions table to link user roles to the default actions.
/BTI/TE_ROLEMET	Role metrics table to link user roles to the default metrics.

To enable the “In progress changes for my team” each user will need to be allocated to the task groups that represent their team in table /BTI/TE_USERGRPS. This will then allocated all tasks for those groups to the team.

By default the primary tester of the task is designated as the task owner unless a user role is attached to the task to override this. See the details for the TASKOWNER_ROLEID field in the /BTI/TE_WEBUICFG table.

4.1.2.3. Web UI – Newsfeed

The ActiveControl Web UI includes a Newsfeed whereby recent activity in the tool is presented, both for the individual user (My Recent Activity) and for other all other users (News).

In order to generate the data for this newsfeed, program **/BTI/TE_RUNEWS_UPDATE** must be scheduled to run every 5-10 minutes in the domain controller. In the program variant, the “No. of days to retrieve events” parameter should specify the number of days that the News / My Recent Activity is required to be reported for.
The default is 60 days but in reality this could be much less for most organisations.

If user pictures are desired in the ActiveControl Web UI newsfeed (and other screens), these must be manually uploaded into the ActiveControl Domain Controller system. See the instructions for the USERPIC_PREFIX field in the **/BTI/TE_WEBUICFG** table.

4.1.2.4. Web UI – Following

Within the ActiveControl Web UI, is possible for users to follow specific Business Tasks, Transports and SAP Objects. This is aimed at making users and stakeholders more self-sufficient and helps to avoid the inevitable “Has this transport been imported”, “Has that change been tested” etc emails being sent during the delivery of SAP Change.

Users can create rules for the items they want to follow in the Web UI. As part of these rules, the User can define whether they want to receive an email notification on the items they have followed, and/or to see an update on the items they have followed in the News feed section of the Web UI.

There are two configuration activities that need to be performed by Administrators to enable Following capability within ActiveControl.

Step	Activity
1.	Program **/BTI/TE_RUFOLLOWITEMS_UPDATE** must be scheduled (after creating a variant) in the Domain Controller. The recommendation would be to add this program to the same scheduled job as the web UI news update program /BTI/TE_RUNEWS_UPDATE).
2.	The Following notification option must be switched on within program **/BTI/TE_RNOTIFICATION_ENGINE**.

4.1.2.5. Web UI – Project Milestones

The ActiveControl WebUI includes a Project view that shows RAG milestone status information for projects, based on where the underlying Transport Forms are in the workflow. To make use of this Project view, the following tables need to be configured by the Administrator in the domain controller.

Project Tables

/BTI/TE_PHASE	Project phases table
Field	**Description**
PHASEID	Unique Id for the project phase
PHASE_DESC	Description of the phase (e.g. Blueprint, Realisation, Final preparation, etc.)

/BTI/TE_PHASSTAT	Table to assign the task statuses to the project phases
Field	**Description**
PHASEID	Id for the project phase
STATID	Id for the relevant task deployment or planning status
STATTYPE	Selection of deployment or planning status

/BTI/ TE_PRJPHASE	Table to assign the project phases to a project along with the start and end dates
Field	**Description**
PROJID	Id for the relevant project
PHASEID	Id for the project phase
SEQUENCE	The sequence that the project phases should run in
START_DATE	The start date for the project phase
END_DATE	The end date for the project phase

If nothing is maintained in the above tables for an ActiveControl Project, the message “No phase data exists for the selected project” will be displayed.

4.1.2.6. Web UI – System messages

It is possible to issue system messages to all users in the ActiveControl Web UI News section via table **/BTI/TE_SYSMESS**. This could be used to inform users of a upcoming system outage or other significant event.

The start and end dates/times indicate the time period during which ActiveControl will present the configured message to users.

4.1.2.7. Web UI – Other Configuration Tables

Various optional configuration tables are also available for the Web UI. For most ActiveControl implementations, these do not need to be utilised.

/BTI/TE_WEBUICFG	General web UI parameters / config table
Field	**Description**
GENERIC_USERPIC	Name of generic user picture (/BTI/TE_WEBUI_USER_PIC_GENERIC)
TE_USERPIC	Name of user picture (/BTI/TE_WEBUI_USER_PIC_TE)
USERPIC_PREFIX	Name of user picture prefix. In order to show user pictures in the web UI the images need to be uploaded and stored on the Domain Controller via transaction SMW0. Choose option “Binary data for WebRFC applications” option. Each user will require an entry with an uploaded user picture and a naming convention should be used so they all have the same prefix ((e.g. ZTE_WEBUI_USER_PIC_). It is mandatory that the user ID follows the name prefix otherwise the user pictures cannot be located by ActiveControl E.g. for user TEUSER it would need to be ZTE_WEBUI_USER_PIC_TEUSER
USERPIC_URLPREFX	URL prefix for user picture (/bti/ te_web_services?action=GETUSERPICTURE&uname=)

DEFAULT_ROLEID	ID of the Role to be used for all users if one is not allocated individually in /BTI/TE_USERPREF Role ID 00000000000000000000 (Default Web User Role) is delivered as standard.
TASKOWNER_ROLEID	ID of the Role to be used to default the task owner. Based on the user roles defined in /BTI/TE_ROLEU in the ActiveControl config “User Roles” tab. A “Task Owner” role could be specified and attached to each ActiveControl Business Task. This can then be configured in this field for use in the web UI. If a role is specified here this will be automatically attached to all tasks when they are created and assigned with the user who is creating the task. The default tester of the task is designated as the task owner if not role is specified here to override this.
NUM_NEWSITEMS	Number of news items to display in the “News” feed. The default is 50
NUM_RECENCTACT	Number of items to display in the “My recent activity” feed. The default is 50
TL_GROUPBYID	Default task list group by option for all users
EXCLUDE_COMPLETE	Exclude tasks where the deployment status is “Deployment Complete” from the task metrics
INC_CURUSER_NEWS	Include the current user activities in the “News” feed. If not set the logged in user activities will not be shown in the news as they can be viewed in the “My recent activity” feed anyway

/BTI/TE_USERPREF	**Web UI User preferences table**
Field	**Description**
UNAME	User name
UIROLEID	ID of the default role to be used for the user in the web UI. Based on the user roles defined in /BTI/TE_ROLEU in the ActiveControl config “User Roles” tab If this is not specified, the role in /BTI/TE_WEBUICFG-DEFAULT_ROLEID will be used
DB_FILTERID	Last selected filter on the Dashboard Overview screen (auto populated by ActiveControl)
DB_PROJECTID	Last selected project on the Dashboard Overview screen (auto populated by ActiveControl)
PM_FILTERID	Last selected filter on the Dashboard Projects screen (auto populated by ActiveControl)
PM_PROJECTID	Last selected project on the Dashboard Projects screen (auto populated by ActiveControl)
TL_GROUPBYID	Last selected group by on the Task List screen (auto populated by ActiveControl)

/BTI/TE_USERGRPS	**User groups table to allow users to be assigned to teams based on the task group**
Field	**Description**
UNAME	User name
TASKGROUPID	Task Group ID of the task group that the user is to be assigned to. This will then associate all tasks that have this group to the user’s team

/BTI/ TE_ROLEACT	Role actions table to link the user roles to the default actions
Field	**Description**
ROLEID	ID of the role to be used. Based on the user roles defined in /BTI/TE_ROLEU in the ActiveControl config “User Roles” tab. Entries for Role ID 00000000000000000000 (Default Web User Role) are delivered as standard and can be copied to create further roles
ACTIONID	The Action Id of the actions to be assigned and displayed for the user role
SEQUENCE	The sequence the actions should be displayed in

/BTI/ TE_ROLEMET	Role metrics table to link the user roles to the default metrics
Field	**Description**
ROLEID	ID of the role to be used. Based on the user roles defined in /BTI/TE_ROLEU (via ActiveControl configuration screen “User Roles” tab Entries for Role ID 00000000000000000000 (Default Web User Role) are delivered as standard and can be copied to create further roles
METRICID	The Metric Id of the metrics to be assigned and displayed for the user role
SEQUENCE	The sequence the metrics should be displayed in

4.1.3. E-mail Notifications

4.1.3.1. Standard email notifications

ActiveControl can be configured to send automated e-mails notifications to relevant stakeholders in numerous scenarios, such as:

- Notify approvers when changes are waiting to be approved (provided the changes are associated with one or more tasks).
- Notify team members that their changes have been imported successfully or with errors. It is possible to additionally notify administrators when a transport is imported with errors.
- In the case of branched development systems, notify team members if their changes should not be imported because of conflicts with changes already made in the project development system.
- Notify team members when changes are waiting to be tested by them (provided that the changes are associated with one or more tasks).
- Notify administrators when a serious error occurs that prevents changes from being imported or an import error occurs. Non administrator users can also be informed of these issues for specific transport paths by maintaining entries in table **/BTI/TE_NOTIFUSR**.

A matrix of all of the Standard email notifications, along with a description of their purpose and audience is available in this online FAQ article.

In order to send e-mail notifications, report **/BTI/TE_RNOTIFICATION_ENGINE** must be scheduled to run periodically in the ActiveControl domain controller. This report allows you to filter those target SAP systems that various types of notifications should be sent for, as sometimes you may not want each type of notification to be sent for each target SAP system.

Standard RSCONN01 program must also be running within the ActiveControl Domain Controller for the emails to be sent.

ActiveControl provides out-of-the-box HTML templates for each of the standard email notifications. These are stored in the Web Repository (transaction **SMW0**), for each type of notification. These templates can be copied and adapted if desired.

Before sending a notification e-mail, ActiveControl attempts to determine the recipient's Internet e-mail address from their SAP user ID. If an Internet address is not specified then the e-mail is sent to the SAP user ID and must be viewed using the SAPoffice Business Workplace (transaction SBWP).

In order to send e-mails to Internet e-mail addresses, a connection to an external e-mail server must have been configured for the ActiveControl domain controller. In systems running on SAP's Web Application Server 6.20 and higher, this is easily done using SAPconnect's built-in SMTP connection.

✻ Email notifications where an action is required include an URL link to the Web UI. For Single Sign-On (SSO), please configure main.html instead of index.html at the end of the **Web UI HTML Path** parameter in /BTI/TE_RNOTIFICATION_ENGINE, this will bypass the login screen and use the current user logon ticket instead.

✻ To mark certain users to not receive any email notifications they can be maintained in table **/BTI/TE_EMAILIGN**. These users can then be excluded via user exit 0510 and removed from the list of recipients.

4.1.3.2. Custom email notifications

In addition to the many standard types of notifications that can be enabled, ActiveControl also allows the creation of additional 'Custom Notifications'.

For each custom notification you wish to send, an entry needs to be configured in table **/BTI/TE_NOTIF_CU**.

Field	Description	Notes
NOTIFID	A unique identifier for the custom notification	
NOTIF_NAME	A description of the custom notification	
NOTIF_HTML	The name of the HTML template for the custom notification	Use transaction SMW0 to

		copy and update an existing HTML template
TARGET LOCATION	If the custom notification is to be triggered when a Transport Form moves into a specific control point, the target and location should be entered	Both the Target and Location fields need to be filled out
DEPL_STAT	If the custom notification is to be triggered when a Business Task reaches a particular Deployment Status, then the status ID should be entered here	Task status IDs can be found in table /BTI/TE_TASKSTAT
PLAN_STAT	If the custom notification is to be triggered when a Business Task reached a particular Planning Status, then the status ID should be entered here	Task status IDs can be found in table /BTI/TE_TASKSTAT
RECIPIENTS	Determines the recipients for this custom notification	See the recipients table below for details
TASK_ROLEID	If the recipients of the custom notification are assigned to a role on the Business Task, the appropriate ROLEID must be entered here	The Task Role IDs can be found in table /BTI/TE_ROLEU

The recipients of the custom notification are determined by the value of the RECIPIENTS field in the custom notification table above. The possible values are:

Value	Recipient(s)
O	The Transport Form owner
T	All testers assigned to the Business Task
N	Users assigned in the system error notification table **/BTI/TE_NOTIFUSR** for the path which the Transport Form is in
R	Users assigned to a particular role on the Business Task. The role ID must be entered into the TASK_ROLEID field in the custom notification table
Y	Users configured in the recipients table **/BTI/TE_NOTIF_RE** for this custom notification

An example – with screenshots – of how to setup a ActiveControl custom notification is available in this online FAQ.

Whilst testing your custom notification, the selection screen of the /BTI/TE_RNOTIFICATION_ENGINE report allows for all generated e-mails to be sent to the current user, rather than to the actual recipients.

4.1.3.3. Reminder Notifications

Within ActiveControl it is possible to send automated **Reminder** notifications to Testers and Approvers when they do not perform their required action in a configurable timeframe, for example when a Business Task and/or Transport Form has been sitting in a particular Inbox, Outbox or Test Queue for a long time. These reminders can be configurable at a Path level, or down to an individual Target or Target

Location.

Two Reminders can be sent, each one potentially to different audiences. Eg the first reminder to the responsible person, the second reminder to a more senior manager.

Configuring Reminder Notifications

1) Configure /BTI/TE_RNTF_ALR with the required reminders.

Field	Description
TARGET	Target ID of the configurable target.
LOCATION	Location within the target (Inbox, Outbox, Test queue).
PATH	Path id for the configured target. Note, if using Path level reminders, you need to configure /BTI/TE_NOTIFUSR to specify the recipients.
ROLEID	Business role ID of a Business Task. Note, this comes from table /BTI/TE_ROLEU after configuring User Role via the Windows GUI.
ALERTDAY1	Number of days the first reminder e-mail should be sent.
ALERTDAY2	Number of days the second reminder e-mail should be sent.
GROUPID1	Groupid of the recipients of the first reminder e-mail. Note, you have to configure /BTI/TE_NOTIF_RE with any Groups you have specified.
GROUPID2	Groupid of the recipients of the second reminder e-mail. Note, you have to configure /BTI/TE_NOTIF_RE with any Groups you have specified.

2) Switch on the 'Remind Approvers' and/or 'Reminder Testers' notifications in your /BTI/TE_RNOTIFICATION_ENGINE program variant.

4.1.3.4. Analysis Result standard notifications

It is possible to send General Analysis or Shiftleft analysis results to defined users in an automated email. These Analysis Result notifications can be defined for specific Paths, Targets, and Locations, and optionally also based on the Type and Group of the Transport Form that had the analyser warnings.

Configuration Steps

Step	Details
1.	Create users groups and assign the relevant users in table **/BTI/TE_ANLNFGRU** table in Domain Controller.
2.	Define the conditions for sending out Analysis Results in table **/BTI/TE_ANLNFGRP** in the Domain Controller. Result notifications PATH: Path where the analysis is run (Mandatory)

	TARGET: Target where the analysis is run (Mandatory) LOCATION: Location (Inbox, Outbox, Import Queue) where the analysis is run (Mandatory) TYPEID: Transport form type (Optional). Only set to restrict the results for specific transports GROUPID: Transport form group (Optional) . Only set to restrict the results for specific transports REASON: Reason code for analysis issues (e.g. 21 is for the Risk Guard analysis). If not specified the email will show all analysis issues USRGROUP: User group (from /BTI/TE_ANLNFGRU) where you want to send the emails to.
3.	Switch on the 'Analysis results' notification in **/BTI/TE_RNOTIFICATION_ENGINE** program.

4.1.3.5. Analysis Result custom notifications

Within ActiveControl, it is also possible to email analysis results to key stakeholders using Custom Notifications functionality.

Configuration Steps

Step	Details
1.	Switch on Custom Notification as part of variant for program **/BTI/TE_RNOTIFICATION_ENGINE**.
2.	Add your required notification in table **/BTI/TE_NOTIF_CU**. Notes: NOTIF HTML: The standard /BTI/TE_SCHEDULED_ANALYSIS HTML template should be used. HANDLING CLASS: /BTI/TE_CL_CUST_NOT_SCHED_ANL
3	Schedule a variant of program **/BTI/TE_RANALYSE_REQUESTS_LOC** for the specific analyser/ path/location for which you want email to be sent. Note: Flag 'create notification entry' must be set.

4.1.3.6. Import Error (RC8) Notification – Additional Recipients.

A standard email notification available through /BTI/TE_RNOTIFICATION_ENGINE can be used to notify individual Transport Owners whenever their transport(s) import with RC8 or above.

It is also possible to notify additional stakeholders via the following configuration in the domain controller.

Step	Details
1.	Creating the required additional import error notifications via table **/BTI/TE_NOTIF_IE**
2.	Defining the recipients for the notification via table **/BTI/TE_NOTIF_RE**

4.1.4. Authorisations

4.1.4.1. User Authorisations

ActiveControl includes the ability to control which functions users are able to access and which objects are visible to the user when they log into the GUI.

Basic authorisations

Basic authorisation to the ActiveControl application for all users requires one of the following ActiveControl roles to be assigned:

Role Name	Description
/BTI/TE:CTS_USER	This role contains all of the basic authorisations that are required to log into ActiveControl application and needs to be assigned to every ActiveControl user.
/BTI/TE:CTS_ADMIN_USER	This role contains all of the basic authorisations that are required to use the application, plus some ActiveControl and SAP standard authorisations required for actions aimed more at Basis and Configuration users of ActiveControl.

Standard Single/Composite Roles

In addition the aforementioned basic authorisations, ActiveControl includes a set of standard out-of-the-box single and composite roles that can be assigned to users in the domain controller. A separate Roles Matrix describing each of these standard roles in detail is available on request, but in summary

Composite Role	Description
/BTI/TE:STD_VIEW_ROLE	The standard role for view only access to ActiveControl. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_VIEW_AUTHS
/BTI/TE:STD_DEVELOPER_ROLE	The standard developer role for creating and maintaining Transport Forms and tasks. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_DEVELOPER_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/TE:STD_PLANNER_ROLE	The standard role when using the planning functionality within ActiveControl and executing planning steps. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_PLANNER_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/TE:STD_TEAM_LEAD_ROLE	The standard role for team leaders and approvers in ActiveControl. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_DEVELOPER_AUTHS • /BTI/TE:STD_PLANNER_AUTHS • /BTI/TE:STD_TEAM_LEAD_AUTHS • /BTI/TE:STD_VIEW_AUTHS

/BTI/TE:STD_TESTER_ROLE	The standard role for testers in ActiveControl. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_TESTER_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/ TE:STD_CHANGE_TEAM_ROLE	The standard role for change teams. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_CHANGE_TEAM_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/ TE:STD_PROJECT_MGR_ROLE	The standard role for project managers. This contains: • /BTI/TE:CTS_USER • /BTI/TE:STD_PROJECT_MGR_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/TE:STD_BASIS_ROLE	The standard role for transport maintenance, import and error handling. This contains: • /BTI/TE:CTS_ADMIN_USER • /BTI/TE:STD_BASIS_AUTHS • /BTI/TE:STD_DEVELOPER_AUTHS • /BTI/TE:STD_TEAM_LEAD_AUTHS • /BTI/TE:STD_VIEW_AUTHS
/BTI/TE:STD_ADMIN_ROLE	The standard role for configuration and administration access to ActiveControl. This contains: • /BTI/TE:CTS_ADMIN_USER • /BTI/TE:STD_ADMIN_AUTHS • /BTI/TE:STD_BASIS_AUTHS • /BTI/TE:STD_DEVELOPER_AUTHS • /BTI/TE:STD_PLANNER_AUTHS • /BTI/TE:STD_TEAM_LEAD_AUTHS • /BTI/TE:STD_VIEW_AUTHS

! Most customers choose to use the standard ‘out-of-the-box’ ActiveControl roles, but if changes are required, it is important that the customer creates Z copies of the roles, as the standard roles may be updated during subsequent upgrades or other ActiveControl application updates.

Approvals and Test Results Entry Authorisations

To control approvals and test results entry / approval for non-administrators the following activities can be allocated to users.

Activity	Authorisation Object	Description
APPROVEALL	Y_TEUSER	Allows approvals to be processed in all locations in ActiveControl.

		Overrides the approvers configured in the target system inbox and outbox approvers
TESTALL	Y_TEUSER	Allows test results entry and “Save and Approve” to be processed in all locations in ActiveControl. Overrides the testers allocated to the tasks
TESTAPPROVE	Y_TE_TASK	Must be allocated to users needing to perform a “Save and Approve” during test results entry

Legacy Authorisation concept

In earlier versions of ActiveControl, the user authorisations was far less granular. For clients that wish to continue using the legacy authorisations concept, which broadly grouped users into Administrators and non-administrators, ActiveControl still comes with the following two compatibility roles:

/BTI/TE:COMP_ADMIN_ROLE	This role should be assigned to all users that currently have Administrator access
/BTI/ TE:COMP_STD_USER_ROLE	This role should be assigned to all other users

Once assigned to users, these roles will give people the same Activity Authorisations as they had in earlier versions of ActiveControl.

4.1.4.2. View Authorisations

View authorisations can be set up so that different groups of users only see the data objects relevant to them when using ActiveControl. For example, if you have completely different teams delivering projects and/or performing Production support activities, each team member only sees the transport paths, transports and configuration relevant to their area.

The main concept in view authorisations is the Authorisation Group. An Authorisation Group can be allocated to each ActiveControl object (Path, Project, Group etc.). Each user is then allocated a SAP Role giving them access to certain Authorisation Groups. A user will therefore only see the objects allocated to an Authorisation Group they have access to.

Authorisation Groups can be assigned to the following Objects in ActiveControl:

- Paths – If there are multiple teams, such as Production Support and Project teams, each team member only sees the transport paths relevant to them. So a Production support team member would only see the Production Support paths in ActiveControl and could only allocate a transport to those paths
- Projects – Different teams may only be able to allocate Tasks to certain projects
- Groups – The valid Groups can be restricted on Tasks and Transport forms
- Types – The valid Types can be restricted on the Tasks and Transport Forms
- Change Paths – The change paths can be restricted when creating Tasks
- User Roles – The user roles can be restricted when creating Tasks
- Custom Fields – Whether a custom field is visible to a user

- Import Methods – Import methods available can be restricted

Currently, all View Authorisations configuration is done in the SAPGUI.

Note: If an object (Path, Project, etc.) is not allocated to any Authorisation Group, all users will have view access to it by default so if this is not required please ensure that all object values are allocated to at least one authorisation group.

- Authorisation groups can be maintained in table /BTI/TE_AUTHGRPS.
- To allocate an Authorisation Group to a ActiveControl object an entry needs to be maintained in the table /BTI/TE_AUTHOBJS on the domain controller.

Table fields for /BTI/TE_AUTHOBJS

Field Name	Label	Description
OBJECT_TYPE	Object Type	This is the ActiveControl object type. Allowed types are in table /BTI/TE_AUTHOTYP. These table includes all of the objects that can be assigned to Authorisation Groups, including Paths, Projects, Groups, Types and custom fields
OBJECT_KEY	Internal ID	The internal ID of the ActiveControl Object. This will be the numeric ID used as the object key in ActiveControl: • Path: Field PATH from table /BTI/TE_PATH • Project: Field ID from table /BTI/TE_PROJ • Group: Field GROUPID from table /BTI/TE_GROUPS • Type: Field ID from table /BTI/TE_TYPE • Change Path: Field PATH from table /BTI/TE_CPATH • User Roles: Field ROLEID from table /BTI/TE_ROLEU • Custom Fields: Field FIELDID from table /BTI/TE_CUSTF • Import Methods: Field METHOD from table /BTI/TE_IMPMET
AUTH_GROUP	Authorisation Group	The authorisation group that is to be associated with this object. The valid list of authorisation groups should be maintained in table /BTI/TE_AUTHGRPS
CUSTOMFIELD_VAL	Custom Field Value	(Reserved for future use)

Once entries in the authorisation objects table have been maintained, roles will need to be created and allocated to users so that ActiveControl can determine which authorisation groups each user has access to. The authorisation object that controls this access is Y_TEAUTH_V.

Authorisation Object Y_TEAUTH_V fields.

Field Name	Label	Description
/BTI/TE_AG	Authorisation Group	This is the authorisation group (or groups) the user has access to. An * in this field indicates the user has access to ALL authorisation groups

/BTI/ TE_C1	Path Category 1	A category can be allocated to a path. This could be to indicate all paths that are BW or XI, for example. And these entries can be used to allow access only to certain categories. The categories a user has access to should be put in this field. An * indicates the user has access to ALL categories.
/BTI/ TE_C2	Path Category 2	(Reserved for future use)
/BTI/ TE_TA	Target Dev System	This is the development system (or systems) for which the user is able to see transports without a Transport Form. This access restriction should only be required if there are multiple 'source' development systems. The SID of the development system(s) should be entered here. An * in this field indicates the user has access to all configured source systems.

Once the required roles for View Authorisations have been created, they should be allocated to the appropriate users.

Important Restrictions Regarding View Authorisations

> ! It should be noted that although View Authorisations allow for a very granular way of restricting the objects that are displayed to users when they log into ActiveControl, the complexity and ongoing overhead of the role and user maintenance should be taken into account.

It is also important to understand that assigning Authorisation Groups can have unintended consequences if not all objects within the object hierarchy are allocated to the same set of groups. For example, the View Authorisations configuration may give a user access to see a particular path, but not access to see Tasks (or Forms) that are part of a particular Project. This means the user will not be aware of all of the transports waiting for approval in a control point, but only those they have authorisation to see. They could therefore inadvertently only approve the transports they have authorisation to see, without knowing the full picture.

In general it is only recommended to use View Authorisations to separate out ActiveControl objects at the highest level, for example, to give separate views to the project delivery and production support teams, or to give separate views to two completely separate project teams using the same ActiveControl Domain Controller.

4.1.4.3. Activity Authorisations

Activity Authorisations can be configured to further restrict user access in addition to the View Authorisations. These activity Authorisations allow the actions the user can perform on visible objects can see to be restricted. The activities are broken down into 6 categories, with each category being controlled by a separate authorisation object:

Authorisation Object	Description
Y_TECONF	This authorisation object controls access to the configuration options within ActiveControl,

	such as creating Target systems and Paths
Y_TEUSER	This authorisation object controls initial access to the ActiveControl GUI and certain activities allowable by user, such as the ability to delegate approval rights to another user.
Y_TE_TASK	This authorisation object controls all activities that can be performed on a Task in ActiveControl, including creating, changing and assigning users.
Y_TEFORM	This authorisation object controls all activities that can be performed on a Transport Form in ActiveControl, including creating, changing, releasing and assigning users.
Y_TEIMORT	This authorisation object controls all activities concerning the import of transports into systems, including marking transports as imported manually and re-importing transports as required.
Y_TEADMIN	This authorisation object controls access to the administrative functions within ActiveControl, including setting UModes on transports and manually changing the process flow of transports as required.

These authorisation objects should be used to create the appropriate roles for each user type.

Activity Authorisation Object Details

Each authorisation object related to Activity Authorisations is detailed below. Only a sub-set of activity codes are valid for each authorisation object, and these are also shown below.

Authorisation Object	Field	Description	Valid Entries
Y_TECONF	/BTI/ TE_OB	TE Configuration Object	TARGET, PATH, OTHERCONFIG, TARGETROLES, SCHEDULE, GROUP, TYPE, LABEL, PROJECT, FIELD, ADMIN, PAPPROVE, USERROLES, CHGSTEPTEMPLATE, CHGPATHSTEP, CHGPATH, USER, MANDATORYFIELD, TASKSTATUS
Y_TECONF	/BTI/ TE_AC	Activity Code	CREATE, CHANGE, DELETE
Y_TEUSER	/BTI/ TE_AC	Activity Code	DELEGATE, FORCEDELEGATE, CONFIGCHECK, CONFIG, APPROVEALL, TESTALL, VIEWCONIMPORTQUEUE, DEPLSTATUS, PLANSTATUS
Y_TE_TASK	/BTI/ TE_AC	Activity Code	CREATE, CHANGE, CHANGEASSIGNED, DELETE, LOCK, UNLOCK, ADDROLE, REMOVEROLE, ADDCHGPATH, REMOVECHGPATH, ADDTESTER, REMOVETESTER, CREATETR, CHANGETR, DELETETR, ADDTR, ADDUSER, REMOVEUSER, REMOVEFORM, CHGDEFAULTTESTER, CHGPATHSTEPUSER, TESTAPPROVE
Y_TE_TASK	/BTI/ TE_DS	Deployment Status	List the Deployment Statuses that the activity is permitted for or “*” for all
Y_TE_TASK	/BTI/ TE_PS	Planning Status	List the Planning Statuses that the activity is permitted for or “*” for all
Y_TE_TASK	/BTI/ TE_PR	Project	List the Projects that the activity is permitted for or “*” for all
Y_TE_TASK	/BTI/ TE_PT	Task Priority	List the Task Priorities that the activity is permitted for or “*” for all
Y_TE_TASK	/BTI/	Task Group	List the Task Group IDs that the activity is permitted for or “*” for all

	TE_GP	ID	
Y_TE_TASK	/BTI/ TE_TY	Task Type ID	List the Task Type IDs that the activity is permitted for or “*” for all
Y_TEFORM	/BTI/ TE_AC	Activity Code	CREATE, CHANGE, CHANGEOTHERS, DELETE, LOCK, UNLOCK, RELEASE, RELEASEOTHERS, FORWARD, FORWARDOTHERS, ADDTASK, REMOVETASK, DELETEOTHERS
Y_TEFORM	/BTI/ TE_GP	Form Group ID	List the Form Group IDs that the activity is permitted for or “*” for all
Y_TEFORM	/BTI/ TE_TY	Form Type ID	List the Form Type IDs that the activity is permitted for or “*” for all
Y_TEIMPORT	/BTI/ TE_AC	Activity Code	IMPORT, BACKOUT, MARKASIMPORTED, ADDTOIMPORTQUEUE, RESETIMPORTSTATUS, MARKASMANUALLYAPPLIED
Y_TEIMPORT	/BTI/ TE_TR	Target role	List the Target Roles that the activity is permitted for or “*” for all
Y_TEIMPORT	/BTI/ TE_TA	Target ID	List the Target System IDs that the activity is permitted for or “*” for all
Y_TEADMIN	/BTI/ TE_AC	Activity Code	APPROVEANYWAY , ADDTOCONTROLPOINT, REMOVEIMPORTLOCK, STOPONERROR, ASSIGNSCHEDULE, SETUMODES
Y_TEADMIN	/BTI/ TE_TR	Target Role	List the Target Roles that the activity is permitted for or “*” for all
Y_TEADMIN	/BTI/ TE_TA	Target ID	List the Target System IDs that the activity is permitted for or “*” for all
Y_TEADMIN	/BTI/ TE_CP	Location	List the Locations that the activity is permitted for or “*” for all

4.1.4.4. Legacy – Screen Variants

The current concept for ActiveControl is described earlier in this section, and should be configured by all customers using ActiveControl 7.3.

In older versions of the product, it was also possible to configure screen variants for the Business Task and Transport Form screens using the legacy authorisation configuration. The steps / tables which need to be populated within SAP and the values that are required in order for this functionality to work are as follows:

Step	Details
1.	Switch on legacy authorisations ‘active’ flag /BTI/TE_AUTHACTV so that the process will be allowed to run.
2.	Define default standard user profile assigned to users not defined in authorised user table /BTI/ TE_STDPRF.
3.	Define ActiveControl authorisation users in table /BTI/TE_ACCESS. If not specified all users will get the role defined in /BTI/TE_STDPRF.

4.	Define ActiveControl profile which an ‘object’ is to be assigned in /BTI/TE_USRPROF.
5.	Define ‘objects’ which activities can be assigned to in table /BTI/TE_AUTHTYPE.
6.	Define fields groups to be used for grouping fields in table /BTI/TE_FLDGRPT. See steps 8 & 9 for more detail
7.	Define screen groups to be used for group fields in table /BTI/TE_SCRNGRPT. See step 8 & 9 for more detail.
8.	Assign standard fields to field / screen groups in table /BTI/TE_STNDFDEF. The fields referred to in this table (Text ID column) are defined in table /BTI/TE_STNDF. The fields can be grouped together into sections, for example, to create a set of display only fields. E.g. Screen group S01 / Field Group S5 could be created and assigned with a group of fields to be made display only.
9.	Assign custom fields to field / screen groups in table / BTI/TE_CUSTFDEF. Note: To make a custom field editable there needs to be an entry in this table for it. The custom fields are defined in table /BTI/TE_CUSTF. The fields can be grouped together into sections, for example, to create a set of display only fields. E.g. Screen group 01 / Field Group A3 could be created and assigned with a group of custom fields to be made display only.
10.	Assign field / screen group to a form type (i.e. Task or Transport Form object) ActiveControl profile, via table /BTI/TE_CUSTGRD. Note: This only needs to be configured if certain standard or custom fields are to be made display only. Activity type 2 = Change and 3 = Display only. Entries need to be added for each combination of screen group and field group to be controlled for Tasks and Transport Forms.

For some screenshot examples of setting up Screen Variants, please refer to this online FAQ.

All references to authorisations and profiles in this Screen Variants section refer the legacy authorisation concept in ActiveControl and are no longer used for actual authorisation checks throughout the application.

4.2. Windows GUI Configuration

The configuration of ActiveControl in the Windows GUI can only be maintained by users who have been defined as a ActiveControl Administrator. Authorised ActiveControl Administrators can access the configuration screen after logging on with the ActiveControl Windows GUI, by selecting the menu item **Tools > Configuration**.

High level Summary of Windows GUI Configuration

Within ActiveControl, each system in your SAP system landscape is defined as a **target**. The target defines the SAP system and the group of clients within that system that changes should be imported into, along with the way in which transports should be imported to that system. The team members who can approve changes are defined for each target, providing the flexibility to have different approvers in your different SAP systems.

Targets are assigned to **Transport Paths** via a simple drag and drop user interface. The transport path

specifies the target systems that assigned transport requests will be imported into, the order of those imports and whether changes need to be approved and tested. The appropriate transport path must be selected when a Transport Form is completed for a transport request, however ActiveControl is usually configured to automatically select the correct transport path based on the Development system in which the transport has been created.

Transport Schedules can be defined to automatically import transport requests into target systems. A transport schedule defines the times when ActiveControl will import transport requests that are queued for import via a background job.

4.2.1. Defining Targets

ActiveControl is designed to import transport requests into all relevant clients of an SAP system in a single action. In ActiveControl, each group of clients within an SAP system is called a target; the properties of a target specify the following:

- The SAP system and clients that changes will be imported into.
- The behaviour of imports into the target.
- The team members who are authorised to approve changes into or out of the target.
- The properties of a target do not determine whether changes must be approved into, approved out of or tested in the target; those settings are made for each transport path. By enabling a target's inbox on a per transport path basis, you have the flexibility (for example) to require your implementation's changes to be approved before they are imported into a target, but to import changes from a central, global SAP development team automatically.
- The automated analysers that will run automatically (or manually) in the configured control points (ie Inboxes, Import Queues, Test Queues, Outboxes) in that target.

Targets are created and maintained by ActiveControl Administrators via the [Targets & Transport Paths] tab in the Windows GUI configuration screen. It is also possible for ActiveControl Administrators to open a Target directly from the Windows GUI main screen, by double-clicking on the Target.

The same actual SAP system can be defined as more than one ActiveControl target system. This can allow different processes to be controlled for the same systems. For example, if the same SAP systems are used for BAU and project changes it's a good idea to have a separate transport path for each which then uses a different set of target systems. This enables greater control and different approvers to be setup for each process.

Target Sequencing: By default, targets appear in the order that they were defined. This order can be easily changed by dragging the targets into the desired order in the main configuration window.

It is not possible to add the same Target to more than one Transport Path.

4.2.1.1. Target Properties – General

Target

Property	Description
SAP System ID	System ID of the target SAP system
Description	Short description of the target
Group Label	An optional label to group together common types of systems (e.g. “Business Warehouse”)
Role	This option allows you to group together separate SAP systems by their role, for the purpose of creating dependencies between transport requests on separate transport paths to support Release Orchestration. Please refer to section Defining Target Roles to learn how to create a target role.

Clients

Property	Description
Clients	The group of clients within the SAP system that transport requests will be imported into. Enter each client separated by a comma and a space (e.g. “100, 200, 300”).
Execute client copy SCC1 automatically on Transport Form creation	Check this option if you want SCC1 to be automatically done to other configured clients on Transport Form creation.
Execute client copy SCC1 automatically on transport release	Check this option if you want SCC1 to be automatically done to other configured clients on transport release.

Please note that S_USER_AGR needs to be assigned to the AC_RFC user role in development systems where SCC1 automation is used. S_USER_GRP and S_USER_PRO are also required for SCC1 of customizing transports that contain security roles.

Miscellaneous

Property	Description
Source	Check this option to allow team members to complete Transport Forms for transport

system for transport requests created in SAP	requests created in this system. This option must be enabled for each development system in your system landscape. Transport requests without a Transport Form appear in a team member's list of open transport requests, in the main window.
Hide this target within ActiveControl.	Check this option to prevent this target from appearing in the main ActiveControl window. You might enable this option when the SAP system no longer exists, or if the system does not exist yet. Hidden targets may still appear on a transport path.
Skip import queue for virtual targets	Virtual targets can be used to facilitate additional Approvals (Inboxes, Test Queues, Outboxes) as part of a complex workflow. A Virtual Target has no import queue; Transport Forms moving past that Virtual Target import queue will show as "Mark as Imported" for that location. The Virtual Target can be given the same SID as the Target in which you want the additional approval point(s) – or it can be given a dummy SID to easily identify that particular approval point(s). If the target system is unavailable or is a virtual target system use this option to skip the Import Queue, but still stop in the other control points for approvals.
Bypass all control points in this target (when system is offline)	This will skip all of the Import Queue, Test Queue and Inbox/Outbox for the system. In most cases, this is the option to use when a system is offline / unavailable.
After approval of items at the … on this target, tasks are automatically locked	Select one of the control points if Tasks are to be automatically locked after approval. This will prevent further allocation to Transport Forms.
After approval of items at the … on this target, automatically release transports	Select one of the control points if transports are to be automatically released after a specified control point approval.
Automatically run general analysis on testing approval	Check this option to run the General Analysis automatically as part of the 'Save & Approve Testing' action on a Test Queue
Allow test entry results for my transport on this target	Check this option to facilitate transport-level Test Queue signoff. Eg if you want a Developer to sign off their Unit Testing when there are other transports on the same BT that are not yet tested. Please refer to 'Partial Testing' section later in this Administration Guide for further information.
Automatically approve items where no critical analysis issues found in	Check this option if you want the configured Analysers to run automatically, and Transport Forms with no issues to be automatically approved.

Consolidated Import Queues

Property	Description
Consolidated Import Queue Options	Does not require consolidated import queue options: Use this option for targets where consolidated import queues are not required Provides consolidated import queue for related targets: Use this option for targets that are to be used as a consolidated target Contributes to a consolidated import queue provided by: Use this option to link a target to a consolidated import queue target so the import queues can be shared. • Target: In this case the consolidated target must be selected • Consolidated queue visible?: This should be switched on to make the queue visible in the normal ActiveControl paths underneath the main import queue. Please refer to the Consolidated Import Queues section for details about how to set this up.

4.2.1.2. Target Properties – Import Options

Import Options

Property	Description
Import Method	There are four standard Import Methods available with ActiveControl for importing transports i) **Import one request at a time**: this imports in the order that is displayed on screen. ii) **Fast Import (Import All)**: this imports using standard SAP Block import, again based on the order that is displayed on screen iii) **One request at a time – TE default sequence**: will import in the ActiveControl calculated sequence (ie the numerical order detailed in the Order column), regardless of the sequence of transports displayed on the screen. iv) **Fast Import (Import All) – TE default sequence**: will import using standard SAP Block Import in the AC calculated sequence (ie the numerical order detailed in the Order column), regardless of the sequence of transports displayed on the screen. Basis Technologies would generally recommend (iii) and (iv) import methods, with the latter particularly useful for project cutovers involving large numbers of transports. (i) and ii) work fine as long you have your Import History view on “By Transport”, and are sorting on the Order column. If you are doing a View by Project and Task, then you could run the risk of importing in an incorrect sequence. For Merge Targets where you are doing 1:1 Merge, it is better to set the Import Method to one request at a time.
Try to import transport requests in the order that they were imported into the predecessor target.	When determining the ideal import order of a selection of transport requests, ActiveControl normally considers any dependencies specified on a transport request’s Transport Form before sorting the transport requests chronologically according to release date and time. This option tells ActiveControl to consider the order that transport requests were imported into a particular “predecessor” target before applying the standard import order rules.
Force transport dependencies	If dependencies that are manually set on a Transport Form are to override the predecessor target import order then set this flag

when importing in the same order as predecessor system	
Scheduled Imports	Assign a transport schedule to a target to automatically import any transport requests in its import queue. Transport requests are imported in their ideal import order (as explained above). A transport schedule defines the times when an automatically scheduled background job will run to import waiting transport requests into the target. Please refer to section Defining Transport Schedules to learn how to create a transport schedule.
Optionally specify additional import schedules	If you assign a Import Schedule to a Target, you can use this option to add further Schedules to the Target
Suppress import analysis during scheduled imports	If you enable this option, ActiveControl will not perform its standard import analysis checks when the scheduled import background job runs. You should only enable this option if the checks have already been performed – for example, when the transport requests were approved into the target system.
Import Jobs Scheduled by ActiveControl	This option should be checked where Schedules have been created to automatically import transports
Orchestrated	This option should be selected where you want ActiveControl to automatically continue an import after i) a dependant transport in another system has been imported or ii) a dependant manual step has been performed in the target. Please refer to ‘Release Orchestration’ section later in this Administration Guide for more information on this functionality.
Continue importing transport requests when an import error occurs	By default ActiveControl stops importing subsequent transports when it detects that an import error has occurred (such as a syntax error). If this option is enabled, ActiveControl will continue importing the selection of transport requests when an import error occurs. Note that even when this option is enabled, ActiveControl will stop the import process if a system error occurs (such as not being able to connect to the target SAP system). For Merge Targets where you are doing 1:1 Merge, it is better to leave this option unchecked.
Automatically create backup transport requests	When this option is enabled, ActiveControl will create a backup transport request for the selection of transport requests that are about to be imported into a target SAP system. A backup transport request contains a copy of all things that are about to be updated as they were immediately before the import occurs. If an unexpected error occurs as the result of importing a selection of transport requests, the state of the target system can theoretically be restored by simply importing the backup transport request. However, this is not an automatic guarantee as there are exceptional situations, such as the deletion of content from an application table, from which only a database restore can recover. Use of this option when importing a large selection of transport requests is not recommended, as it can significantly slow the import process. In these situations, it is recommended to rely upon database recovery techniques. Please note that ActiveControl Backout is not relevant for BW or Java systems. It is currently only to be used in ECC/ERP type systems.
Timeout for delayed imports	Transport imports work in ActiveControl by calling the TP command to import the transport and waits for a response. In some cases (due to a very long running import)

(minutes)	the TP command will time out and therefore ActiveControl has no other option but to report a system error as it doesn't know what the actual status is. By setting a timeout for delayed imports ActiveControl will retry every minute for this many minutes and check if the import is now complete. Only after this time has elapsed will it then report a system error if the import is still not complete after that time.
Ignore System Id during import (CTS+ only)	For CTS+ systems the SAP system ID is different to the system that is the CTS+ domain controller. In this case the flag should be set to inform ActiveControl that the target is a CTS+ system.

Merge / Parallel Development Streams

Property	Description
Perform conflict analysis against this target	When this option is enabled, ActiveControl will perform an additional remote analysis check for transport requests that are to be imported into this target system. The remote import analysis checks whether the content that is about to be imported has already been changed in the target system. This option is particularly useful for branched SAP development systems, when changes from one development system must be applied to other development systems.
Client to be used during conflict analysis	Client in which to check for changes and in which merge transport requests will be created. For performance reasons, the remote import analysis checks are only performed in one client of the target SAP system. Similarly, merge requests are only created in this client. You should specify the client of this target SAP system in which client-dependent configuration changes are made.
Require that transports with changes to SAP objects be manually merged	When this option is enabled, ActiveControl will check if a transport to be merged contains changes to SAP standard development objects and if so, require that the transport be manually merged. This option is useful if you have branched development systems that are running different versions or patch levels.
Create a merge transport request in this SAP system after importing changes	When this option is enabled, ActiveControl creates merge transport requests in this system for each selection of transport requests imported. This option is particularly useful for branched SAP development systems. Creating merge transport requests provides an easy way to apply the same changes to downstream systems of the branched development stream (e.g. the project QA system).
Fix renamed objects in merge requests	During import into BW systems some BW objects are automatically renamed. The default behaviour of the merge process is to copy the objects from the original transport into the merge transport but this doesn't work for the affected BW objects. In this case ActiveControl can run a rename process either 'After Import', 'Before Release' or 'After Import & Before Release' to perform the renaming process. This will edit the merge transport and renaming the objects to the new names in the merged system. 'After Import' is the default renaming method.
Add dependant routines	Some BW objects should always be transported in their entirety. If, for example, a transformation was being transported this option would also add in all the dependent routines and formulas into the merge transport.

and formulae for BW merges	
Default package for merges objects	When BW objects are renamed sometimes they are not assigned with an object directory entry (TADIR). The package specified in here can be used to create this to prevent warnings during merge transport release
Inherit merge transport owner from original transport (CTS+ only)	Check for Java systems to ensure Merge transport owner is same person as the original transport (for 1:1 Merge scenario)

4.2.1.3. Target Properties – Import Options II

Merge / Parallel Development Streams (Continued)

Property	Description
Transport target for merge requests	This sets the transport target for merge requests created by ActiveControl, allowing merge transports to be handled via a different transport route if required. If not set the default value from TMS will be used.
Merge Size	Select whether you want one merge transport for every originating transport (1:1), a big one with everything (Consolidated) or one per import run (By Import Run)
Merge Path	Select which path the Transport Forms for the merges will be created. Leave it blank if you want to create them manually.
Merge Type	Select which Type you want Merge Transport Form to be allocated to. If not set, the merge will get the type of the first transport merged.
Merge Group	Select which Group you want for the Merge Transport Form to be allocated to. If not set, the merge will get the group of the first transport merged.
Stop on BW post-processing error	This option controls whether Merge process will stop or continue when BW postprocessing fails
Merge Task	Select which Task the Transport Form should be associated with. If not set, the merge will get the task of every transport merged.
On Error	Controls whether ActiveControl Merge should stop, continue or skip any other errors encountered during the Merge process.
For 1:1 merges automatically copy over Transport Form manual step details	Check if you want Manual Steps to be added in the Merge Transport Form in 1:1 merge scenario; *it not relevant to the Consolidated Merge scenario*.

Unconditional Modes

Property	Description

Unconditional Modes	These options allow you to apply various unconditional modes to all transport requests imported into this target system.

Custom Processing

Property	Description
Pre-import logical command	ActiveControl supports transporting between SAP systems that do not share common underlying transport directories. Before importing a transport request, ActiveControl checks whether the required transport files are available and, if not, automatically copies the files to the target transport directory. If further import pre-processing is required, you may use this option to invoke an SAP logical command (transaction SM69) to call your own script before the import is performed. You can choose whether the script is invoked on the domain controller or the target SAP system.
Post-import logical command	This option is similar to the previous option. It allows for a logical command to be called after a transport request has been imported into a target system. This option might be used for copying transport logs that were created by the import process back to the transport directories of the source SAP development system.

4.2.1.4. Target Properties – Inbox (Pending) Approvers

Property	Description
Inbox (Pending) Approvers	Here you specify those people who can approve changes into the target, from the target's inbox, or approve the transporting of changes from the target's pending queue. Approvers can be authorised to approve any transport request or only those assigned to particular groups. To add an approver, select either a specific group or All Groups and then click the Add button to choose an approver from the list of users in the domain controller SAP system.
Minimum number of approvals required	The number of approvers that must approve a change before the change moves to the next location on the transport path, such as from the target's inbox to its import queue.

4.2.1.5. Target Properties – Outbox Approvers

Property	Description
Outbox Approvers	Here you specify those people who can approve changes out of the target, from the target's outbox. For example, a test manager may wish to prevent the flow of changes into the production system until satisfied that appropriate levels of testing have been performed. Approvers can be authorised to approve any transport request or only those assigned to particular groups. To add an approver, select either a specific group or All Groups and then click the Add button to choose an approver from the list of users in the domain controller SAP system.
Minimum number of approvals	The number of approvers that must approve a change before the change moves from the target's outbox to the next targets, as per the transport path configuration.

required	

4.2.1.6. Target Properties – Analysis Types

Property	Description
Location	Here you can switch on the analysis types that are required for the control points of the relevant target system. This allows additional checks and validations to be performed at various points in the process. Select 'All' to switch on the analysis types for all the control points of the target. Select 'Inbox', 'Import Queue', 'Test Queue', 'Outbox' or any combination of those to switch on the relevant analysis type for the required target control point(s). Note: This is only available for the control points currently switched on for the target system in the transport path.
Analysis Type	Check the analysis types that are to be switched on for the selected control point(s). These are as defined in table /BTI/TE_ANLTYPE.
Settings (Attributes)	Highlight the relevant Analysis Type to see the attributes. Mandatory: Here the analysis type can be set as Mandatory or not. If set as mandatory then it will automatically be called by the General Analysis function and also during Import and Approval. Check Subsequent Target: In some cases for transports that are in a test queue or outbox the analysis might need to be performed on the subsequent system rather than the current system. If this is the case this option should be switched on. To set / unset the attributes, highlight the relevant attribute and choose Yes or No.
Settings (Parameters)	Highlight the relevant Analysis Type to see the attributes. Here the relevant parameters are displayed and can be set if applicable. This is based on what associated parameters have been set up in table /BTI/TE_ANLTYPEP for each analysis type. Table /BTI/TE_ANTYPEPT defines the parameter types. Table /BTI/TE_ANTYPEPV defines the parameter values. Set the parameters that are required for the specific analysis type. When the parameter is highlighted a caption will appear at the bottom of the screen to describe the function of that parameter. To set / unset the parameters, highlight the relevant parameter, right click and choose Add / Remove as required (or use the Add / remove buttons). When adding, select the appropriate value from the drop down or type the value directly if permitted.

4.2.2. Defining Transport Paths

A **transport path** is a hierarchy of target SAP systems, determining the systems that a transport request will be imported into and in what order. The transport path also determines whether changes need to be approved and/or tested.

Each transport request is associated with a single transport path when the Transport Form is completed. The transport path is usually selected automatically, according to the default settings made here, and ensures that transport requests are imported into the same systems, keeping the landscape in sync.

Usually only one transport path is necessary per development stream. For example, you might have one transport path for transporting between the development, test and production R/3 systems, and another transport path for transporting between the development, test and production BW systems.

Transport Paths are created and edited by ActiveControl Administrators on the [Targets & Transport Paths] tab of the Windows GUI configuration screen.

Property	Description
Transport Path	A short description of the transport path.
Group Label	An optional label to group common transport paths together.
Status	This field determines whether the transport path may be assigned to new Transport Forms. It also determines whether an inactive transport path (that can no longer be assigned to new Transport Forms) appear in the ActiveControl main screen
Enable a pending approval associated with target	If this option is enabled, transport requests must be approved before they can be approved into or imported into any target SAP systems. Transport requests waiting to be approved appear in the pending queue of the target system specified here.
Automatically release transports…	If this option is enabled, ActiveControl will automatically release a transport request when a Transport Form is completed. However, if a pending queue is enabled for the transport path, then the transport request is only released after transporting has been approved.
When completing a Transport Form, check whether the changes conflict with others that are still being transported	When this option is enabled, ActiveControl will automatically check whether there are other transports still being transported on the transport path with changes to the same content. The conflicting transports are displayed in a pop-up window.
Do not allow team members to complete a Transport Form if conflicting changes are found	This option can only be enabled when the previous option is enabled. When enabled, team members may not complete a new Transport Form if other transports with changes to the same content are still being transported on the transport path. This option does not apply to Administrators.

For additional control, ActiveControl allows combinations of valid and/or invalid source systems and clients to be defined for each transport path. When a Transport Form is completed for a transport request, ActiveControl will only allow the transport request to be assigned to a transport path if the source system and client of the transport request satisfies the transport path's source client restrictions

Property	Description
Valid Source Clients	In this field you may specify combinations of source systems and clients for the transport path. For example, if you specify combinations "DEV:100, DEV:200" then only transport requests created in either client 100 or client 200 of SAP system DEV may be assigned this transport path.
Invalid Source Clients	In this field you may specify combinations of source systems and clients that may not be assigned the transport path. For example, if you specify "DEV:300" then transport requests created in client 300 of system DEV cannot have this transport path assigned. The list of invalid source clients overrides the list of valid source clients, though it is not usually used in conjunction with a list of valid source clients.



If a target option is disabled, then transport requests that are currently in the inbox (for example) still need to be approved. The option is only disabled when existing transport

requests have been approved (or have had test results entered in the case of the test queue).

To prevent control points from being accidentally removed a warning message will be displayed if any transports currently reside in the removed inbox, test queue or outbox. This allows the user to cancel the action.

4.2.3. Assigning Targets to a Transport Path

Once a transport path has been created, target systems need to be assigned to the path. An Administrator does this by selecting the transport path in the configuration window – the targets that are currently assigned will appear in the transport path tree underneath – and then dragging each required target to the tree by either:

1. Drop the target system onto an empty space to add the target system as a root node of the transport path, or
2. Drop the target system onto another target system that already appears in the transport path tree to build a hierarchy of targets.

It is not possible to have the same Target on more than one Transport Path. If this is required, you need to create a second copy of the target to use in your second Transport Path.

When a Transport Form is completed (or a transport request pending transport is approved), the transport request automatically flows to the transport path's "root" targets. Once a transport request finishes processing within a particular target, it flows to the targets that follow on the transport path.

For example, if the Production target follows the Test target in the transport path tree, then a transport request that is assigned to the transport path will be imported into Production after it has been imported into the Test target – and potentially after the associated task has been tested and/or the changes approved out of the target (if these transport path options are enabled for the target).

The transport path also specifies whether transport requests need to be approved into, tested in and/or approved out of a target system (these are the Inbox, Test Queue and Outbox options respectively). Simply enable the desired options using the checkboxes that appear alongside each target system in the transport path tree.

4.2.4. Defining Target Roles

A **Target Role** is used to enforce transport request dependencies between systems in different transport paths. The target role essentially tells ActiveControl what systems are at the same level of a landscape

(eg that a ECC Dev and BW Dev system are at the same level).

When creating a dependency between transport requests, ActiveControl ensures that the dependent transport request will not be imported into a system until its dependencies have. With the configuration of target roles, it is possible to assign multiple systems in different transport paths the same role. This will ensure that the dependent transport request is not imported into a system with a specific role until all its dependencies are imported into another system with the same role.

A practical application of the use of roles can be explained with an example landscape that contains an ECC and Business Warehouse transport path, each with Development, Quality Assurance and Production systems. By grouping the two Quality Assurance systems with a role “Quality Assurance” and the two Production systems with another role “Production”, it is possible for transport request dependencies to be upheld across the entire landscape. This can ensure a necessary change is moved into the BW Quality Assurance system, before a dependent change is moved into the ECC Quality Assurance system.
Similarly the dependencies will be upheld when importing into the corresponding Production systems.

Target Roles are maintained by ActiveControl Administrators via the [Target Roles and Transport Schedules] tab in the Windows GUI configuration screen.

4.2.5. Defining Transport Schedules

A **Transport Schedule** defines a list of days and times when transport requests should be imported automatically.

When a transport schedule is assigned to a target, ActiveControl automatically creates a background job that will import transport requests that are currently within the target’s import queue. The name of the automated import job is **TE_SCHEDULED_IMPORT_SID_nnnn**, where SID is the SAP system ID and nnnn is an internal number for a target. The background jobs run in the domain controller. Do not maintain or delete these jobs from within SAP.

ActiveControl ensures that the background job always executes according to the current version of the transport schedule. This includes deleting the aforementioned background job if the transport schedule is no longer assigned to the target.

Transport Schedules are maintained by ActiveControl Administrators via the [Target Roles and Transport Schedules] tab in the Windows GUI configuration screen. Each transport schedule has a description and optionally the user ID that the import should be scheduled under.



Basis Technologies recommend to use a dedicated batch user in your Domain Controller for handing Transport Schedules. This batch user – which we typically call AC_BATCH – needs to have /BTI/TE:CTS_ADMIN_USER, /BTI/TE:CTS_RFC and /BTI/TE:COMP_ADMIN_ROLE roles assigned to it.

p(banner tip). As well as the AC_Batch user, it is also necessary for an RFC to be created in the Domain Controller, that points back at itself. (eg TRANSPORT EXPRESS [SID], where SID is the System ID of the Domain Controller. The AC_RFC service user in the Domain Controller needs to have the following roles. SAP_BC_TRANSPORT_ADMINISTRATOR, /BTI/TE:CTS_RFC , /BTI/TE:CTS_ADMIN_USER , /BTI/TE:CTS_ADMIN
Without all of these roles, the import jobs mentioned above will not be created.

Adding Times to a Transport Schedule

The transport schedule can be defined on the Days and Times tab of the transport schedule. Times are periodic. For example, if the time “Monday at 5:30pm” is added, then an import background job will be scheduled to run every Monday at 5:30pm.

To add times, select the required day of week checkboxes and the required time interval. A list of days and times will be listed. Next, select the required times and then drag and drop them onto the appropriate transport schedule (appearing on the left-hand side).

ActiveControl will automatically update the import schedule and associated background jobs for any targets that the transport schedule is currently assigned to.

Deleting Times from a Transport Schedule

To delete an individual time from a transport schedule, select the time in the transport schedule tree and press the Delete key.

On doing this, ActiveControl will automatically update the import schedule for any targets that the transport schedule is currently assigned to.

Condition-based Transport Schedules

It is possible to create and assign multiple condition-based Import Schedules to a target system. This can be helpful in scenarios where you want to deploy different types of transport/change to different release frequencies, for example if you want Emergency Fixes to be imported as soon as they are ready, but other Business Task ‘Types’ to be deployed as part of a less frequent release cycle.

Condition-based Schedules can be configured within the [Selection Criteria] tab of the Transport Schedule screen within the Windows GUI. Conditions can be created based on both standard and custom fields at both a Business Task and Transport Form Level.

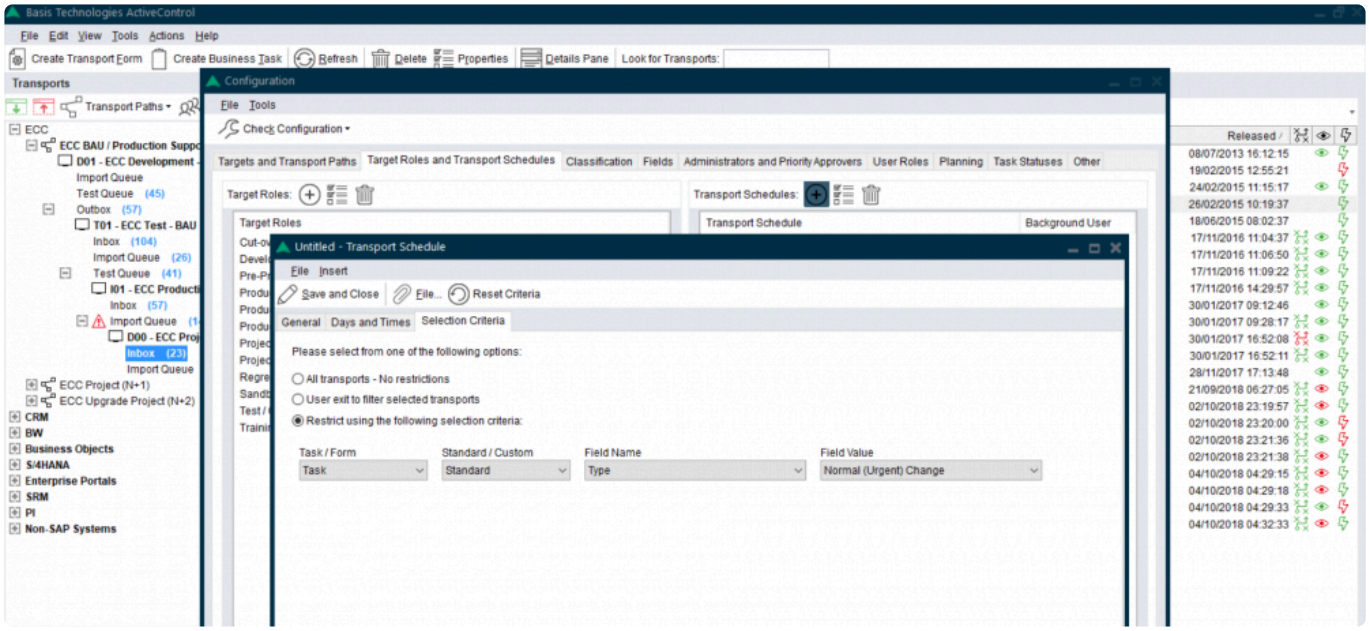


Figure: Creating a condition-based Transport Schedule

Configuration against Targets

The addition of additional schedules to SAP systems is done within the *Import Options" tab of an individual Target within the Windows GUI. It can also be done in the Advanced Import Options menu whilst on the main Windows GUI screen (on an Import Queue).

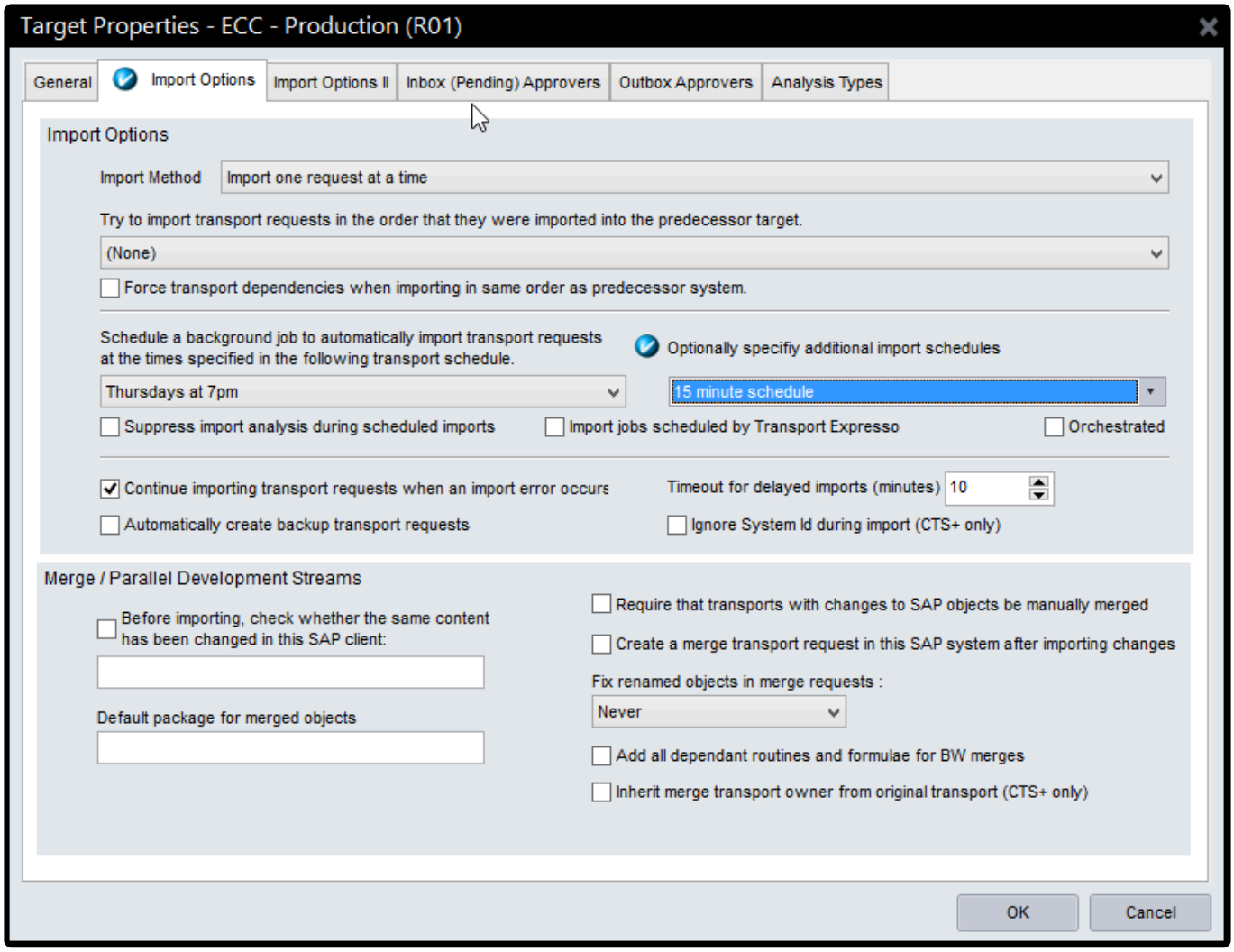


Figure: Assigning more than one Transport Schedule to a Target import queue.

Future scheduled imports

It is possible to forward schedule Transports into **Production**. The scenario this functionality aims to address is where an ActiveControl customer has a regular release cycle, but wants to delay specific changes until a specific date/time in the future after the next release. This can now be done via options available on the Business Task [Additional Data] tab.

To configure Future Scheduled Imports, a customers ‘Production’ Target Role must be assigned to the ‘Production’ Function within Windows GUI Configuration. It is this function that tells ActiveControl which is your Production system . No other Target Roles need to be assigned to a Function. (the Development, Test and Merge functions are for future ActiveControl development, and currently do not drive anything within the product)

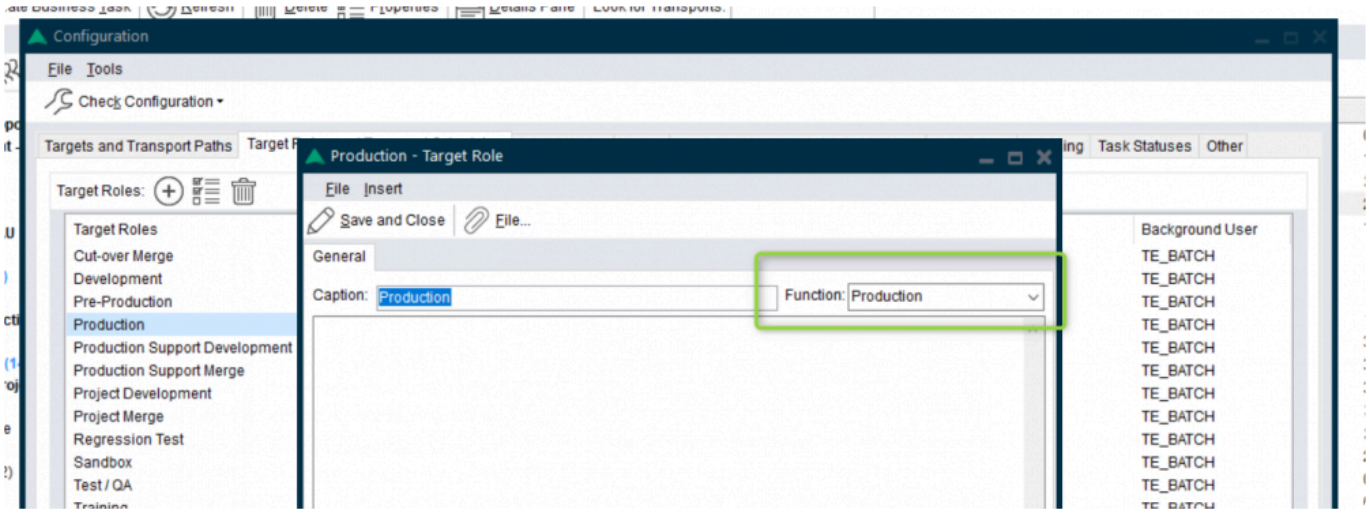


Figure: Assigning Production function to your Production target role.

> ! ActiveControl Future scheduled imports are only designed to work for Production imports. They do not work for non-Production systems.

> ✻ Forward Scheduling is completely separate to standard ActiveControl Import Schedules configured in the Windows GUI, and does not work in conjunction with any custom field based Selection Criteria rules that you may have setup. This means that if you setup a forward schedule, it will be imported at the specified future date/time, irrespective of any restrictions you have configured as part of ActiveControl standard Import Schedules.

4.2.6. Defining Projects

Projects within ActiveControl are used to define a logical grouping of Business Tasks. Projects are typically used at most ActiveControl customers to group together Releases, Upgrade Projects, SAP Projects. Defining and using projects give you further control of change management reporting through ActiveControl

Projects are maintained by authorised users in the Windows GUI or Web UI.

Projects that are no longer required or finished can be marked as inactive so they no longer appear in the Business Task creation process.

4.2.7. Defining Groups

Groups are used to arbitrarily group transport requests and Business Tasks (units of work). Typically, groups are used to classify the transport request or task by functional area or team. For example, groups can be used to differentiate Finance changes from Procurement changes.

Groups are created and edited by ActiveControl Administrators on the [Classification tab] of the Windows GUI configuration screen.

A group may be assigned to a transport request or assigned to a business task, as indicated by the Type property. Groups that have been used in the past but have now become obsolete may be hidden so that they can no longer be assigned.

> Transport Form ‘Groups’ are the basis for assigning approvers for Inboxes and Outboxes. For example, an approver may be authorised to approve Finance changes into a target system, but not Logistics changes.

4.2.8. Defining Types

Types are used as another way of classifying Business Tasks or Transport requests. For example, a Business Task ‘Type’ can differentiate enhancements from fixes.

Types are created and edited by ActiveControl Administrators on the [Classification tab] of the Windows GUI configuration screen.

A type may either be assigned to a task or assigned to a transport request, as indicated by the Applies to property. Once a type has been created, this field cannot be changed. Types that have been used in the past but have now become obsolete may be hidden so that they can no longer be assigned by selecting the Hide this Item toolbar option.

> Types are typically used only for Reporting purposes, or in some customers for driving skipping rules so that certain types of Business Tasks or Transports bypass certain parts of the workflow.

> Examples of commonly configured Business Task types include: Business Change, Incident Fix, Standard Change and Project Change.
> Examples of common Transport Form types are: Customization, OSS Note, Merge, Authorizations, External Transport and Workbench .

4.2.9. Defining Text Fields

It is possible to define additional free text fields to appear on the [General] tab of both the Transport Form and Business Task.

Text fields are configured by ActiveControl Administrators on the [Fields] tab of the ActiveControl Windows GUI configuration screen.

When configuring a text field, you define the name of the text field, whether it is for tasks or Transport Forms, and its maintenance status. The name of the text field is the text that is displayed beside the new field on the Transport Form. The maintenance status defines whether the field contents are optional or required when creating the Transport Form. It is also possible to configure the maintenance status to be

hidden in the case that the field is no longer wanted on the Transport Form.

If the maintenance status of the field is set to hidden, then any Transport Forms that were previously created that still contain text for that field, still display this information. If the Transport Form is then changed, and the contents of the field deleted, then the documentation field is no longer displayed.

> ! Text Fields have been largely deprecated by the newer and more functionally custom field functionality within ActiveControl. Basis Technologies would recommend to use custom fields instead of text fields, as the latter might be removed from future versions of ActiveControl.

4.2.10. Defining Custom Fields

It is possible to enhance the standard Transport Form, Business Task and Project screens with **Custom fields**. Custom Fields are a popular way for ActiveControl customers to tailor the out-of-the-box product to their individual processes. Custom Fields can be used to drive ‘Skipping’ rules within an ActiveControl workflow, and are also available in most ActiveControl Reports.

Custom Fields are created and edited by ActiveControl Administrators on the [Fields] tab of the Windows GUI configuration screen.

The following custom field types are currently supported:

Field Type	Transport Form	Business Task	Project
Text fields	Supported	Supported	Supported
Drop-down List	Supported	Supported	Supported
Selection List	Supported	Supported	Not Supported
Option/Radio-Button	Supported	Supported	Not Supported
Check-box	Supported	Supported	Not Supported
Date	Supported	Supported	Supported
Time	Supported	Supported	Not Supported
User Selection	Supported	Supported	Supported
Section Header	Supported	Supported	Not Supported

All custom fields can be marked as optional, mandatory or hidden. Other options are also available for specific custom fields, such as to limit the number of characters of a text field.

> ✻ When you create a Dropdown custom field, the order that you see them in the configuration screen is the sequence that you will see them in the Dropdown on the Business Task or Transport Form.

Section Header field type can be used to create logical groupings of custom fields within the same screen. It creates a line across the screen.

Note that it is not currently possible to select multiple values in a Select List custom field.

Make custom fields read-only after a certain Deployment or Planning status

It is possible to make a custom field (on either a Business Task or Transport Form) become read-only after a certain Deployment or Planning status has been reached. This configuration is done via backend table **/BTI/TE_CF_STYLE** in the ActiveControl domain controller.

Field	Description
FIELD_ID	This is the number of the custom field that you want to make read-only, you can get this number from the Windows GUI Configuration screen.
DSID	This is the Deployment Status (you can select the status from drop-down during configuration, you do not need to get the long GUID from another table).
PSID	This is the Planning Status (you can select the status from drop-down during configuration, you do not need to get the long GUID from another table). Note, most customers do not use Planning workflow so this field will be blank.
READONLY	You need to tick this box to 'activate' your rule.

Notes on read-only Custom Fields

(1) Read-only custom fields only works on custom fields. It does not work on standard fields.
(2) You must configure all statuses where you want the field to be read-only. You cannot just enter the first status in the sequence where it should become read-only.
(3) The sequence of statuses is based on what you have defined in [Task Statuses] configuration screen.
(4) User with authorization object Y_TEUSER for field '/BTI/TE_AC' should have the value 'EDITCUSTFIELD' to edit the configured field to edit the custom fields at any point, regardless if they are read-only due to the configuration.
(5) This functionality relies on a Deployment Status being defined for every target/location in your workflow. This should be done any for effective Reporting within ActiveControl, but it becomes even more important with this functionality. If you have not set Deployment statuses for some of your path/target/locations – you will see spurious results. Please refer to Defining Task Statuses section of this Administration Guide for more information on how to create these.
(6) This functionality uses the Task Statuses calculations ("Use latest step in sequence" or "Use earliest step in sequence") defined in Windows GUI Configuration [Other] tab as part of its logic. This means that if you have more than one transport against a Business Task, you may see unexpected results depending on how you are calculating the status. Similarly, you may also see unanticipated results in the

event you link a Transport Form to more than one Business Task as part of your process. It should be noted that this activity is not generally recommended by Basis Technologies for most customers.

4.2.11. Defining Custom Tabs

ActiveControl includes a capability to add **Custom Tabs** on a Business Task or Transport Form level.

This can be useful for customers that want to add a large number of custom fields within ActiveControl to track additional information, but don't want them all to appear on the General tab as it would result in lots of vertical scrolling.

Custom tabs are visible on the Business Task and Transport Form screens in the ActiveControl Windows GUI and Web UI and also in the SAP GUI.

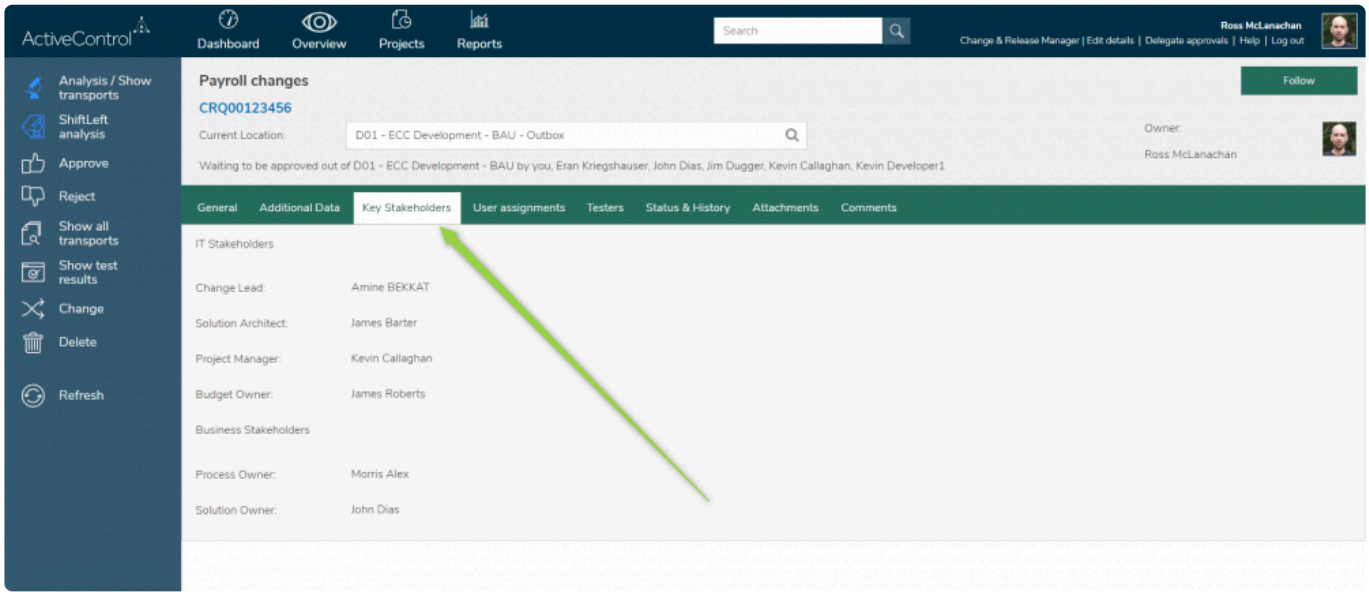


Figure: Example of a Custom Tab on a Business Task in the Web UI.

Configuration Steps

Custom Tabs can be created via the [Custom Tab] section of the Fields tab in the ActiveControl Windows GUI configuration screen. After creating the new custom tab, custom fields can then be assigned to the custom tab via new tab dropdown. If left blank, the custom field will appear on the 'General' tab by default.

If you decide that a custom tab is no longer needed as part of your process, you can only delete the tab and/or the custom fields. It is not possible to hide the custom tab.

4.2.12. Defining Mandatory Fields

It is possible to configure standard fields on Business Tasks and Transport Forms as **Mandatory fields**, to force users to populate them at the point of creating the Transport Form or Business Task. If a mandatory field is not populated, the Transport Form or Task cannot be saved.

Mandatory fields are defined by ActiveControl Administrators on the [Fields] tab of the Windows GUI configuration screen.

✻ Some fields must be mandatory by default for ActiveControl to function correctly. For example, if the Transport Form Group is not populated, it would impact the Approvals workflow. Similarly, if you are using some of the standard fields to drive your workflow or skipping rules, it would be advisable to make all such fields mandatory.

4.2.13. Defining Administrators

An **Administrator** is a resource within the customer organisation who can maintain the ActiveControl configuration and is also able to perform all activities within ActiveControl including approvals, imports and test result entry / approval. Within most organisations, the Administrators are typically either Basis resources or Change & Release Managers.

ActiveControl Administrators are maintained on the [Administrators and Priority Approvers] tab of the Windows GUI configuration screen.

✻ When the first user logs into the Windows GUI after ActiveControl has been installed and a license key uploaded, they are automatically assigned as an Administrator. Basis Technologies highly recommend adding another couple of other Administrators even at this early stage of the implementation, as it can cause technical issues within the tool if no Administrators are defined in the Windows GUI.

4.2.14. Defining Priority Approvers

Priority Approvers may approve changes into or out of any target. Additionally, when approving a selection of changes, they may choose to approve with priority, which means that the changes will not need to be approved into or out of any other targets, once the priority approval is given.

Priority Approvers are maintained by ActiveControl Administrators on the [Administrators and Priority Approvers] tab of the Windows GUI configuration screen.

! Priority Approvers is quite powerful functionality within ActiveControl, as it in effect

means that the automated analysis checks are bypassed and issues are potentially not highlighted. For this reason along with wider audit/compliance concerns, many Basis Technologies' customers choose not to enable Priority Approvers as part of their ActiveControl workflow.

4.2.15. Defining Custom Labels

Custom Labels allow customers to rename the standard Business Task field labels (e.g. Description, Subject etc.) to align with their own internal organizational terminology. For example, it is possible to rename the Business Task 'Group' field to 'Work Package', or the 'Reference' field to say 'JIRA ticket' instead.

The Business Task fields whose labels can be amended with custom labels are:

1. Description
2. Subject
3. Group
4. Reference
5. Project
6. Type
7. Priority

Note that the current custom labels functionality only supports Business Task fields and not Transport Form fields.

Labels are added/removed by ActiveControl Administrators via the Windows GUI configuration screen (on the [Classification] tab).

4.2.16. Defining Business Task Statuses

A Business Task can have two different kinds of status configured.

1. Deployment status
2. Planning status

Business Task statuses are defined by ActiveControl Administrators via the [Task Statuses] tab on the Windows GUI configuration screen.

For both Deployment and Planning status, a sequence must be defined to indicate the order in which the status is to be applied.

Deployment status

The **Deployment Status** of a Business Task reflects where the underlying Transports are in the workflow process. Each control point within a path can be configured with a Business Task Deployment status. Whenever a Business Task, or any of the Transports attached to it are changed or move along a transport path, the Business Task Deployment status is updated. The Deployment Status assigned to each control point in a transport path is configured as the target is associated to a Path during path configuration.

A Deployment Status can be configured for each control point enabled (Inbox, Test Queue and Outbox). This is done by ActiveControl Administrators in the Deployment Status section of the [Targets & Transport Paths] tab in the Windows GUI configuration screen.

When ActiveControl calculates the Deployment status of a Task, it takes into account all of the Transport Forms assigned to the Task and the Deployment Status Calculation configuration (this is detailed in the Other Configuration Options section of this document). If transports are in several control points, ActiveControl will use the status with either the lowest or highest sequence number depending on this configuration option. If all transports assigned to the Business Task are finished in all paths, ActiveControl will set the system Deployment Status as ‘Deployment Complete’.

Planning Status

> ! Planning status’ are only required for customers using the Task Planning capability within ActiveControl. Please note that this Task Planning workflow is legacy functionality within ActiveControl, and may be retired completely from future versions of the product. For this reason, Basis Technologies would not recommend new customers to implement the Planning workflow functionality; it is no longer supported by Basis Technologies as of ActiveControl 8.0.

The Planning Status indicates where a Business Task is in the Planning workflow. Each planning step in a Change Path can have a status associated with it; every time a Business Task is changed, its planning status will be automatically updated.

The Planning Status assigned to each change path step is configured in the Options tab of the step (in the [Planning] tab of the Windows GUI configuration screen.

When ActiveControl calculates the Planning status of a Business Task, it takes into account all of the active Change Path Steps for the Business Task and the Planning Status Calculation configuration (this is detailed on the Other Configuration Options section of this document). If more than one step is active, ActiveControl will use the status with either the lowest or highest sequence number depending on this configuration option. If all steps in all Panning Change Paths for a Business Task are complete, ActiveControl will set the system Planning Status of “Complete”.

> ✻ Changing the configuration of new or existing Deployment or Planning statuses will not

trigger updates to existing Business Tasks within ActiveControl. Once changes to the Planning Status configuration have been completed, execute the report /BTI/TE_RUTASK_STATUS_UPDATE in the Domain Controller. This will re-calculate the planning and deployment statuses for each selected Business Task.

4.2.17. Other Configuration Options

Numerous other important ActiveControl configuration options are available on the [Other] tab in the Windows GUI configuration screen. These are all global configuration options that impact all usage of the product.

Option	Description
Require Transport Forms to be assigned to related tasks	When a Transport Form is completed for a transport request, it is always possible to assign the transport request to one or more Business Tasks, such as a request for a new report or a logged production support problem. Business Tasks provide greater visibility of related transport requests and associate technical changes with business requirements. Business Tasks are necessary if test results are to be entered. Enable this option to prevent a Transport Form from being saved if the related tasks section has not been maintained.
Allow team members to delete Transport Forms of transport requests that have been imported	By default, in order to avoid losing transport history, only an administrator may delete the Transport Form of a transport request if the transport request has already been imported into some target systems. Enable this option to override this behaviour.
Configured testers only to complete testing	When this option is enabled ActiveControl will only allow users allocated as testers in the Business Task to “Save and Approve” testing. This will also validate testers assigned to specific systems and system roles. If not set anyone can complete the testing for a Business Task. Note: In all cases users must be allocated with the TESTAPPROVE activity (Authorisation Object: Y_TE_TASK – Task Activities) to allow them to “Save and Approve” testing.
Enable “Add to Control Point” function	This option switches on the “Add to Control Point” functionality that allows Transport Forms to be assigned to control points in the transport path. This can be an inbox, outbox or test queue as required. This is used when a re-approval is required for the relevant transports.
Cache remote transport data	This option enables the caching function that stores details of remote transports on the domain controller. This means that ActiveControl does not have to make RFC calls to the remote systems to obtain the transport data. Performance is therefore improved when accessing transport requests without a Transport Form and also when viewing transports in control points. Note: Once switched on the ActiveControl GUI will need to be closed and re-started for the caching to take effect.
Hide Transport originating in ‘hidden’ systems	This option will not show transports if the target is set as hidden
Request password re-entry on Approval	This option allows ActiveControl customers to specify whether password must be re-entered during any approval, for increased security. This is of particular interest in an FDA or Sarbanes-Oxley compliance perspective at some customers. When switched

	on, every Inbox and Outbox approval will requires a password confirmation.
Automatically delete SAP transport buffer after transport import	This option can be used by customers that do not want to leave the SAP transports in the STMS import buffer after they have been imported via ActiveControl.
Auto-generate Task Reference with prefix:	This is useful for customers that do not use another Change Management ticketing tool, and are looking for a way to generate a unique identifier for each Business Task within ActiveControl. It is possible to configure a 3-character prefix for the sequential number.
Require reason on transport Approve/ Import Anyway	This option can be used to force approvers to enter a reason when the 'Approve Anyway' button is clicked on the Analysis Results warning screen.
Default "Allow Progress when incomplete" for new manual steps	This option can be used to ensure that the corresponding option on a new Manual Step is always ticked.
Warn if "Allow Progress" unchanged	This option can be used to warn the Manual Step creator if they have not updated the flag.

Analysis Mode

Option	Description
Analysis Mode	Select whether you want to: (i) Disable cross path conflict checks. (ii) Enable cross path conflict checks for non-hidden Targets. (iii) Enable cross path conflict check for all Targets.

This is where you specify whether you want the Conflict Analysis Checks to only look down the path in which you are doing the analysis, or alternatively analyse all paths (based on the system you are in and its allocated target role etc). The default value (and generally what it should be set-to for most customers) is: "*Enable cross-path conflict checks for all targets*". This ensures that any conflicts are picked up in the situation where multiple transport paths are being used by an ActiveControl customer for the same SAP system, and also when a user has manually set cross-system/path dependencies (e.g. a BW transport on an ECC transport etc.).

Customers that have a lot of hidden/legacy systems/paths set up in ActiveControl might want to exclude the hidden systems to help the performance of the analysis checks, but as a general rule, no customer should ever really need to disable the cross path conflict checks completely.

Task Statuses

Option	Description
Planning Status	If more than one planning status could be applied to a task, this configures which one the system should assign. Either the one with the earliest sequence or the one with the latest

Calculation	sequence.
Deployment Status Calculation	If more than one deployment status could be applied to a task, this configures which one the system should assign. Either the one with the earliest sequence or the one with the latest sequence.

Non-SAP Deployment and Manual Activities

Option	Description
Non-SAP Deployment and Manual Activities	Specify what system where empty 'dummy' transports for Manual Activities are created. These transports will not be imported anywhere.

Currency

Option	Description
Currency	Allows ActiveControl customers to specify the primary reporting currency within the organisation, for some optional meta-data / reporting.

4.2.18. Check Configuration

When an Administrator has finished making changes to the ActiveControl configuration, they should run **Check Configuration** via the Windows GUI. This is available via the Tools dropdown in the main screen of the Windows GUI, and also via a button within the configuration screen itself.

The configuration check window will appear and automatically perform a number of connection and validation checks in each of the SAP systems defined as an active target. Targets that are hidden are not checked. Most significantly, the configuration check will verify that the RFC destinations that have been maintained in SAP transaction SM59 are correct and connect through to the matching target SAP system.

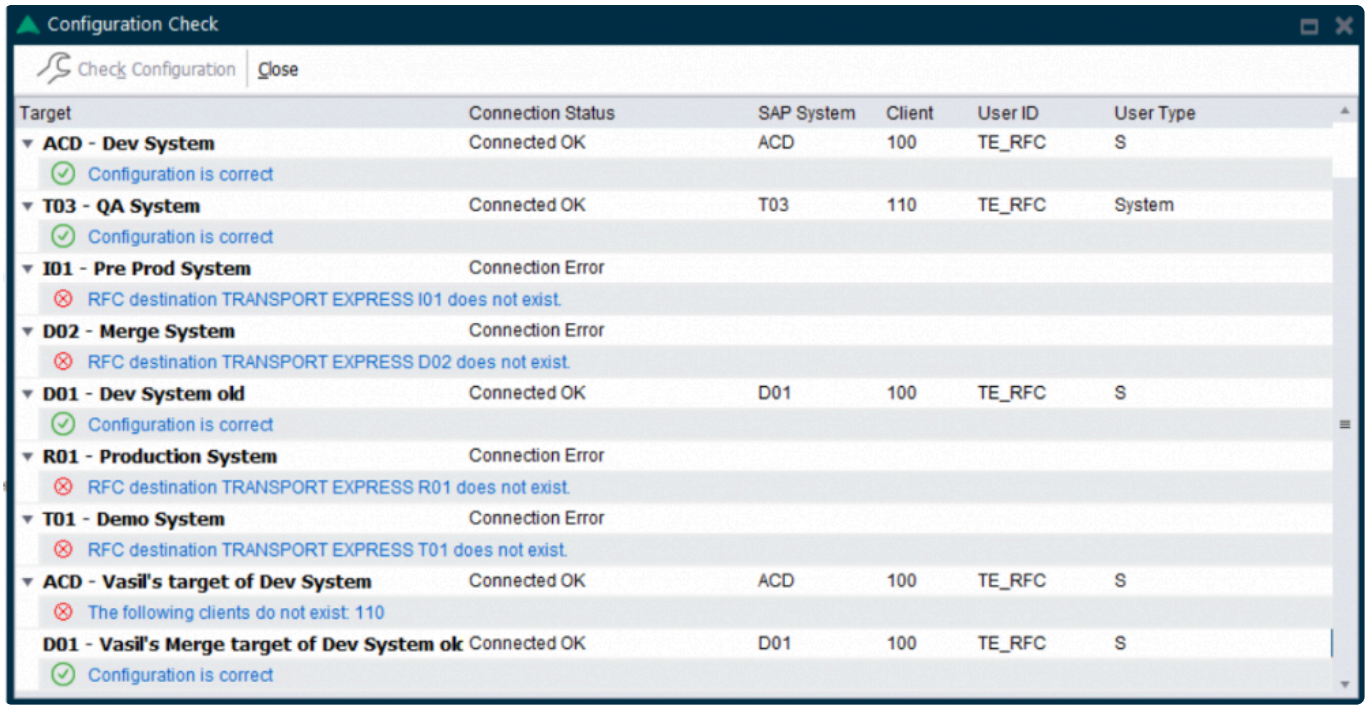


Figure: Check Configuration result list.

5. Advanced Configuration Topics

5.1. Automated Analysis Checks

ActiveControl includes several analysis functions that run automatically as part of every Inbox or Outbox approval and Import Queue deployment – and also as part of Test Queue signoff (if configured).

These core analysis functions are called the General Analysis, and consist of:

1. Check Dependencies
2. Overtake and Regression Checks
3. Check Locked Transport Forms
4. Check Authorisations
5. Check Transport Release
6. Conflict Analysis
7. Check Merge Origin
8. Check Manual Steps
9. Check Manual Activities

In addition to the General Analysis, the ShiftLeft component of ActiveControl offers more than 60+ more configurable analysis that can also be used to ensure any potential issues are indicated as early on in your SAP Change Management process as possible. These ShiftLeft analysis functions can be turned on or off individually within ActiveControl depending on a customers specific requirements.

The rest of this section describes each of the General and ShiftLeft analysis checks in more detail.

5.1.1. General Analysis

ActiveControl performs a General Analysis function automatically during approval and import, this includes the following checks.

(0030) Check Dependencies
(0031) Overtake and Regression Checks
(0032) Check Locked Transport Forms
(0033) Check Authorisations
(0034) Check Transport Release
(0035) Conflict Analysis
(0036) Check Merge Origin
(0037) Check Manual Steps
(0039) Check Manual Activities

The rest of this section details of the above analysis functions in turn.

> ! Analysis types 0030 to 0039 represent the General Analysis functions, they do not need to be switched on via configuration. The General Analysis will also perform any other configured analysis types that have been marked as *mandatory* for that control point.

> ✻ It is possible to suppress the running of an analysis type for a specific target and location by adding entries to table **/BTI/TE_ANALEXCL**. This can be useful to stop the overtake/ regression/conflict analysis types from being executed in target system import queues when there are large numbers of transports to be imported. If the analysis has already been executed and verified during the inbox approval it might be useful to disable these for the import queue to save time when importing transports.

5.1.1.1. Check Dependencies (0030)

- Is there a dependency on another transport that has yet to be imported?
- Is the Transport Form attached to a Business Task but not all the requests in that Business Task have been imported yet or are not being approved now? Normally all transports attached to a task fulfil a business requirement so they should be approved / imported together.

Parameters

None.

> ✻ Check Dependencies (0030) relies on Target Roles set in Windows GUI configuration against the Target. If two targets in the same path are set with same target role, or if one of the targets has no target role assigned, then you may see unexpected results.

5.1.1.2. Overtake and Regression Checks (0031)

The Overtake and Regression checks are used to ensure that transports and changes are approved and imported in the correct sequence.

- **Overtake**: This is where newer changes, for example an emergency bug fix, are being moved before existing in-progress changes to the same objects. In this case there is a risk of deploying a partial change with the big fix that is incomplete and not yet ready.
  A check is made to see if the transport(s) contains a newer version of any objects that also appear on older transports that have not yet been imported into the target system?
  If so, when the older request is imported in future you risk overwriting the target system objects with old out of date versions.
- **Regression**: This is where older changes overwrite newer ones that have been deployed out of

order. Here there is the risk of losing the latest changes and unexpectedly regressing the system to a previous version
A check is made to see if the transport(s) contains an older version of any objects that have been previously imported into the target system?
If so you risk overwriting the target system objects with old out of date versions.

In both cases **all** objects are checked across SAP transports including, but not limited to, development and configuration objects.

In the case of configuration changes checks are done for:

- Table configuration entries (TABU). Includes the identification of overlapping table key entries so that, for example, a conflict would only be identified if the actual table key contents overlap (including wildcards)
- View Cluster Maintenance: Data (CDAT). Includes the expansion of the associated tables for conflict checking
- Customizing: Table Contents (TDAT). Includes the expansion of the associated tables for conflict checking
- View Maintenance: Data (VDAT). Includes the expansion of the associated tables for conflict checking

Parameters

Parameter	Description
KEY_NO_RELATED_VIEWS	By default, when transporting table entries, views that use the same table in key fields will be considered a conflict. This behaviour can be turned off with this flag.
KEY_NO_WILDCARDS	If this flag is set, wildcards are not considered when searching for conflicting table entries.
NO_CACHE	Turn off conflict cache (always recalculate the conflicts from the actual object lists).
SHOW_CONFLICTING_OBJ_KEYS	Show the actual table entries that are in conflict. (This is not purely display but can have performance implications since when you don't show the keys, you can stop at the first conflict, but otherwise it needs to go on.)

5.1.1.3. Check Locked Transport Forms (0032)

Check Locked Transport Forms (0032) checks if the Transport Form is locked, as this would prevent approval or import.

Parameters

None.

5.1.1.4. Check Authorisations (0033)

Check Authorisations (0033) checks if the user is authorised to perform the approval based on the settings in the target system approvers.

If the user running the Analyser is an ActiveControl Administrator, Priority Approver or has the APPROVEALL authorisation then they will always be allowed. If not, they must be a designated approver for the relevant group or must have been delegated to allow approval.

Parameters

None.

5.1.1.5. Check Transport Release (0034)

Check Transport Release (0034) checks if the transport is released (as otherwise it cannot be imported).

Parameters

None.

5.1.1.6. Conflict Analysis (0035)

Conflict Analysis (0035) checks if the transport contain objects that have also been changed on the target system. This is used for conflict analysis when moving from one system to another (e.g. when merging from a production support development path to a project development path).

Parameters

Parameter	Description
COMPREHENSIVE	If set, the analyser will report on all conflicts and objects.
EXCLUDE_MERGED	if set, the analyser will ignore any conflicts before last merge.
EXCLUDE_TOC	if set, the analyser will ignore any conflicts with Transport of Copies.
GLOBALMERGECHECK	By default, only transports that originate in the target system are checked for conflicts. If this flag is switched on, then all transports in the target are checked, regardless of their original system. There is a potential performance downside to having this functionality active.
SHOW_CONFLICTING_OBJ_KEYS	If set, the analyser will show the actual table entries that are in conflict. This is purely display but can still have performance implications since when you don’t show the keys, you can stop at the first conflict, but otherwise it needs to go on.
UNRELEASED_TRANSPORT_CHECK	If set, the analyser will also indicate in the results whether the

	conflicting transport is released or not.
USE_CACHE	If set, the analyser uses cached results, otherwise it is always recalculated.

5.1.1.7. Check Merge Origin (0036)

Check Merge Origin (0036) checks that the origin of the transport(s) that make up the merge request is the same system. This is relevant in case of consolidated merges and it checks if the transports you are merging are from the same source system as all the other ones that are in the current open merge.

Parameters

None.

5.1.1.8. Check Manual Steps (0037)

Check Manual Steps (0037) identifies if there is a dependency on any manual steps that have not been marked as complete.

Parameters

None.

5.1.1.9. Check Manual Activities (0039)

Check Manual Activities (0039) identifies if there is a dependency on any manual activities that have not been marked as complete.

Parameters

None.

5.1.1.10. Check future import date/time (0056)

Check future import date/time (0056) can be used to identify if any analysed Business Tasks have had future import scheduling defined.

Parameters

None.

5.1.2. ShiftLeft Analysers

5.1.2.1. Analysis Type Configuration

The analysis types are configured against the relevant target systems and control points in ActiveControl.

Access to this can be found under menu options Tools -> Configuration menu in the **Targets and Transport Paths tab**.

To setup the analysis types for a specific target system open the target system properties and choose the **Analysis Types** tab.

The screen is divided in to three segments:

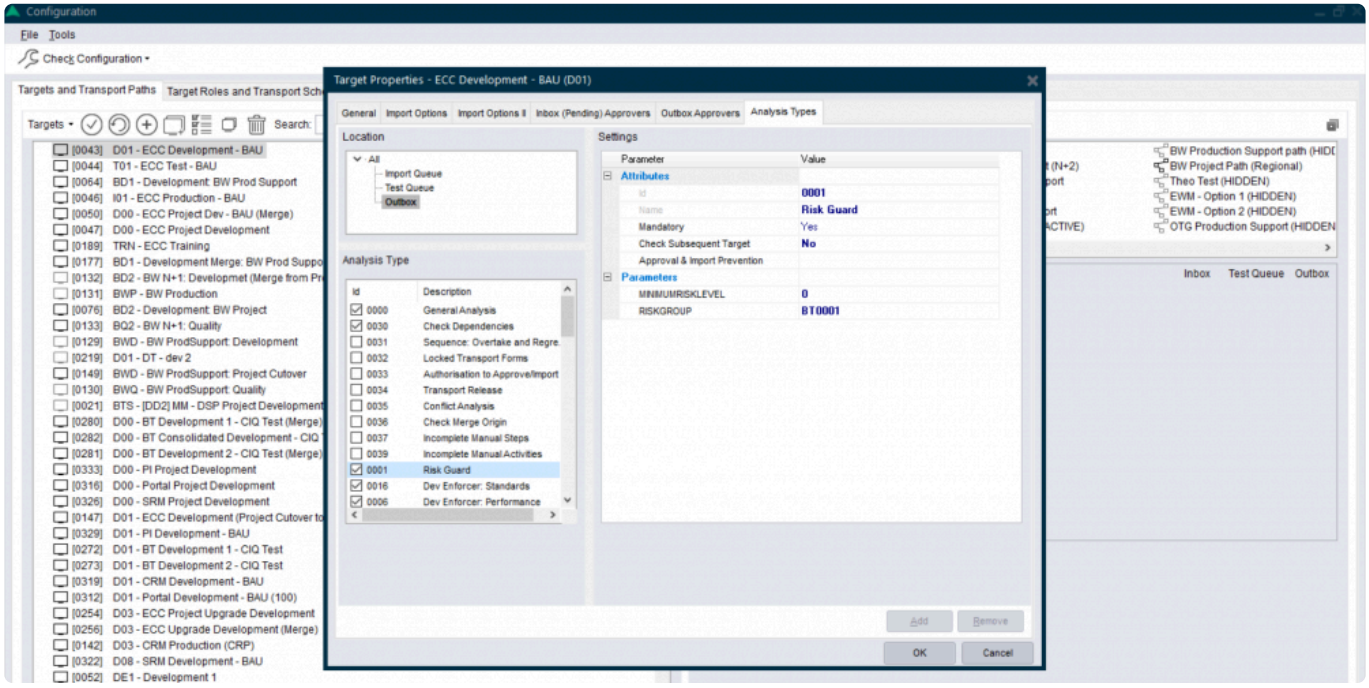


Figure: Analyser configuration screen

1. Location:

- The Analysis Types can be set differently for each control point within the target. Only the control points that are switched on for the selected target in the transport path it exists in are selectable.
- Either select “All” to switch on the analysis types for all control points or select the required control point.

2. Analysis Type:

- These are the different Analysis Types that can be selected (based on /BTI/TE_ANLTYPE).
- The General Analysis will always be switched on as this is mandatory.

- To switch on an analysis type, use the relevant checkbox.

3. Settings:

- This is where the analysis attributes and parameters are selected.
- Click on the analysis number or name to highlight it to allow these values to be maintained.

Attributes:

- Mandatory: To make the analysis mandatory so it automatically runs during the General Analysis choose “Yes”
- Check Subsequent Target: To make the analysis run on the next target in the path rather than the current target choose “Yes”
- Approval & Import Prevention: To prevent approval or import if this analysis returns an error use the button. This will allow the “Prevent Approval”, “Import Approval” and “Admin Override” options to be set for each relevant reason code (based on /BTI/TE_ANRELINK).
- The “Admin Override” option allows ActiveControl Administrators (or anybody else that has role including authorisation activity APPROVEANYWAY (within Y_TEADMIN)) the ability to override the warnings using the ‘Import Anyway’ button
- For the General Analysis this can be used to control this for each of the attached reason codes.

Parameters:

- If parameters are available for the selected analysis types these will be listed (based on /BTI/TE_ANLTYPEP).
- To enter a value in the parameter, highlight the required entry and choose the “Add” button. This will allow an entry to be made in the parameter value field.
- To remove a parameter value, highlight the required entry and choose the “Remove” button.

5.1.2.2. Risk Guard (0001)

Risk Guard (0001) checks all objects on the selected transport requests and highlights any that contain objects identified as critical or risky, for example Company Code changes, Plant changes or changes to Number Ranges. If there are any changes to these types of objects then a message will be issued notifying the user of the level of risk for that request.

Parameters

Parameter	Description
MINIMUMRISKLEVEL	Minimum Risk Level from table/BTI/TE_RISKL
RISKGROUP	Risk Group from table /BTI/TE_RISKG

Prerequisite Configuration

There are 4 tables that must first be configured in the ActiveControl domain controller before Risk Guard will work:

Table	Description
/BTI/TE_RISKL	Risk Levels
/BTI/TE_RISKG	Risk Groups
/BTI/TE_RISKGOB	Risk Group Objects
/BTI/TE_RISKGOBT	Risk Group Object Texts

Some sample entries are delivered in these tables but it recommended to create local customer specific entries so they won't be overwritten during any upgrades. The tables can be configured using via SE16 or SM31. It is also possible to download whatever has already been maintained in Risk Group Objects and texts using /BTI/TE_RUDOWNLOAD_RISK_DATA, and also to do an enmasse upload of this data using /BTI/TE_RUUPLOAD_RISK_DATA.

Risk Levels

Risk levels are configured in backend table **/BTI/TE_RISKL**.

Field	Details
RISKLEVELID	10 character field for the Risk Level Identifier.
RISKLEVEL	10 digit numerical field used to indicate the risk level. The higher the value of this field, the higher the risk.
TITLE	50 character comment field. This can be used to indicate the risk level in words. For example, "Low risk", "Moderate risk", "High risk" etc.

Below are some example entries for this table:

Risk Level Identifier	Risk Level	Comment
BT0001	1	Risk level 1
BT0002	2	Risk level 2
BT0003	3	Risk level 3
BT0004	4	Risk level 4
BT0005	5	Risk level 5
BT0006	6	Risk level 6
BT0007	7	Risk level 7
BT0008	8	Risk level 8
BT0009	9	Risk level 9

BT0010	10	Risk level 10

When setting up Risk Guard, you can specify a risk level value. If that is set, the Risk Analysis will only report on objects that have a risk level greater than the specified value.

Risk Groups

Risk groups provide ActiveControl with a way of varying how risks are identified for different systems or control points. For example, a program called ZZ_TEST may be a moderate risk when going into a QA system but a high risk when moving into a Production system. Risk Groups are configured in backend table **/BTI/TE_RISKG**.

There are 2 fields in this table. Field **GROUPID** is the group identifier and the field **TITLE** is the description of the group.

Below are some example entries for this table:

GROUPID	TITLE
BT0001	Basis Technologies Risk Group 0001

Risk Group Objects

Risk Groups Objects are configured in backend table **/BTI/TE_RISKGOB**.

Field	Description
GROUPID	Links the object to the group specified in table /BTI/TE_RISKG.
PGMID , OBJECT, OBJ_NAME	Relate to the same fields in the object list for a transport. For example, to highlight the transports of any number range objects as risks, there needs to be 2 entries in this table, one for R3TR NROB and one for R3TR TABU NRIV. Note that * can be used as a wildcard value. E.g. T0*
RISKLEVELID	References the risk level table /BTI/TE_RISKL.
APPENDFLAG	Checking this will include direct includes (eg. adding MARA with this flag set also adds EMARA and IST_MAT_FIELDS)
INVERSEFLAG	Currently does not do anything.
DELETEFLAG	Can be used if you want the Risk Guard analysis to indicate if a transport contains a deletion to the object)
URL	Can be used if you want the Risk Guard analyser output to include a URL to further information.

Below are some example entries for the **/BTI/TE_RISKGOB** table:

GROUPID	PGMID	OBJECT	OBJ_NAME	RISKLEVELID	TEXT	APPENDFLAG	INVERSEFLAG	DELE
BT0001	LIMU	INDX	*	BT0009	Database indexes require rebuilding – take care importing into production			
BT0001	LIMU	TABD	*	BT0007	Table definition changes may have an effect on production data			
BT0001	R3TR	NROB	*	BT0008	Be careful when transporting number range objects			
BT0001	R3TR	TABL	*	BT0008	Table definition changes may have an effect on production data			
BT0001	R3TR	TABU	NRIV	BT0010	Number range intervals should not be transported unless during setup			
BT0001	R3TR	TABU	PATH	BT0007	Be careful when transporting logical filenames and directories			
BT0001	R3TR	TABU	T000	BT0008	The client table should never be transported			
BT0001	R3TR	TABU	T001	BT0008	Be careful when changing the critical			

					Company Code table			
BT0001	R3TR	TABU	T012	BT0007	Be careful when transporting the House Banks table			
BT0001	R3TR	TABU	T030	BT0007	Be careful when transporting the critical General Ledger table			
BT0001	R3TR	TABU	TBDLS	BT0008	Be careful when transporting Logical Systems			
BT0001	R3TR	XINX	*	BT0009	Database indexes changes should be carefully applied to production			
BT0001	R3TR	XPRA	*	BT0006	Take care as this change will be executing program(s) after import			

Risk Groups Object Texts

Risk Groups Object Texts are configured in backend table **/BTI/TE_RISKGOBT**. This table contains the text to be displayed when the Risk Guard finds any risks.

5.1.2.3. Dev Enforcer: Security (0004)

In transaction SCI it is possible to setup Code Inspector variants to perform security checks on SAP programs. These can be setup as General variants for all users or user specific ones. Dev Enforcer: Security (0004) hooks into these variants to perform the check as part of the Analysis/Approval process within ActiveControl.

Parameters:

Parameter	Description
CIEXCL_EXISTING_ISSUES	Can be used to display the Code Inspector results in following modes: LIST_ALL – All the CI issues that have been existing will be listed LIST_BOTH_SEP – issues that have been existing and issues that are introduced with the current transport will be displayed with separate section. LIST_NEW – Only issues that are introduced with the current transport will be displayed.
IGNORE_WARNING_INFO	EXCLUDE_BOTH: Exclude both Informational and Warning messages from DevEnforcer results. Only Errors will be reported. EXCLUDE_INFO: Exclude only Informational messages. Warning and Error messages will be reported. EXCLUDE_WARNING: Exclude only Warning messages. Information and Errors messages ill be reported.
CISID	SAP System ID for Execution. Specify the system ID (e.g. D01) where the Code Inspector variant is to be run from.
CITARGET	Target System for Execution if CISID is not specified. Specify the target system ID (e.g. 0001) where the Code Inspector variant is to be run from.
CIUSER	Code Inspector Execution User. Leave blank for global variants.
CIVARIANT	Code Inspector Variant name.

More information – including screenshots – of SAP Code Inspector can be found in this online FAQ.

5.1.2.4. Dev Enforcer: Performance (0006)

The Dev Enforcer: Performance analysis is achieved by making a call to SAP Code Inspector in the relevant target system.

In transaction SCI it is possible to setup Code Inspector variants to perform the required performance checks on SAP programs. These can be setup as General variants for all users or user specific ones.

Parameters

Parameter	Description
CIEXCL_EXISTING_ISSUES	Can be used to display the Code Inspector results in following modes: LIST_ALL – All the CI issues that have been existing will be listed LIST_BOTH_SEP – issues that have been existing and issues that are introduced with the current transport will be displayed with separate section. LIST_NEW – Only issues that are introduced with the current transport will be displayed.
IGNORE_WARNING_INFO	EXCLUDE_BOTH: Exclude both Informational and Warning messages from DevEnforcer results. Only Errors will be reported. EXCLUDE_INFO: Exclude only Informational messages. Warning and Error messages will be reported. EXCLUDE_WARNING: Exclude only Warning messages. Information and Errors messages ill be reported.

CISID	SAP System ID for Execution. Specify the system ID (e.g. D01) where the Code Inspector variant is to be run from.
CITARGET	Target System for Execution if CISID is not specified. Specify the target system ID (e.g. 0001) where the Code Inspector variant is to be run from.
CIUSER	Code Inspector Execution User. Leave blank for global variants.
CIVARIANT	Code Inspector Variant name.

More information – including screenshots – of SAP Code Inspector can be found in this online FAQ.

5.1.2.5. Dev Enforcer: Standards (0016)

In transaction SCI it is possible to setup Code Inspector variants to perform the required programming conventions checks on SAP programs. These can be setup as General variants for all users or user specific ones. Dev Enforcer: Standards (0016) analysis check is achieved by making a call to SAP Code Inspector in the relevant target system.

Parameters

Parameter	Description
CIEXCL_EXISTING_ISSUES	Can be used to display the Code Inspector results in following modes: LIST_ALL – All the CI issues that have been existing will be listed LIST_BOTH_SEP – issues that have been existing and issues that are introduced with the current transport will be displayed with separate section. LIST_NEW – Only issues that are introduced with the current transport will be displayed.
IGNORE_WARNING_INFO	EXCLUDE_BOTH: Exclude both Informational and Warning messages from DevEnforcer results. Only Errors will be reported. EXCLUDE_INFO: Exclude only Informational messages. Warning and Error messages will be reported. EXCLUDE_WARNING: Exclude only Warning messages. Information and Errors messages ill be reported.
CISID	SAP System ID for Execution. Specify the system ID (e.g. D01) where the Code Inspector variant is to be run from.
CITARGET	Target System for Execution if CISID is not specified. Specify the target system ID (e.g. 0001) where the Code Inspector variant is to be run from.
CIUSER	Code Inspector Execution User. Leave blank for global variants.
CIVARIANT	Code Inspector Variant name.

More information – including screenshots – of SAP Code Inspector can be found in this online FAQ.

5.1.2.6. Dev Enforcer: Naming Conventions

(0003)

Naming Conventions (0003) can be used to check that the transport description of all analysed transport (s) meet a specific naming convention. As transport naming conventions for this differ from customer to customer, this is generally delivered by Basis Technologies.

> Nowadays most Basis Technologies' customers use ActiveControl's newer 'Automated Naming Convention' functionality to ensure a consistent naming convention of their SAP transports. Using that functionality makes this analyser check redundant.

5.1.2.7. Check Valid To Date (0007)

Check Valid To Date (0007) gets a unique list of tasks relating to the analysed transport requests and then checks the validity date of these tasks against the system date. This is based on the date contained in the specified custom field. If the system date is greater than that of the custom field date then a message is issued stating that the change is no longer valid.

Parameters

Parameter	Description
ANALYSISCAPTION	Allows the analysis caption to be overridden. Enter the required text up to 100 characters. This text will appear in the analysis results.
ERRORIFNOTSET	If the valid to date is not set then display an error.
IGNOREURGENT	Ignore check if task priority is set to Urgent.
TASKFIELDTEXTID	Custom field text ID for Valid To date (from table /BTI/TE_CUSTF).

5.1.2.8. Check Don't Approve Before Date (0008)

Check Don't Approve Before Date (0008) gets a unique list of tasks relating to the analysed transport requests and then checks the approval date of these tasks against the system date. This is based on the date contained in a specified custom field. If the system date is less than that of the custom field date then a message is issued stating that the change is not ready to be approved yet.

Parameters

Parameter	Description
ANALYSISCAPTION	Allows the analysis caption to be overridden. Enter the required text up to 100 characters. This text will appear in the analysis results.
ERRORIFNOTSET	If the valid to date is not set then display an error.
IGNOREURGENT	Ignore check if task priority is set to Urgent.

TASKFIELDTEXTID	Custom field text ID for Valid From date (from table /BTI/TE_CUSTF).

5.1.2.9. Check Releasability (0009)

Check Releasability (0009) looks at each unreleased transport request and checks to see if there are any problems that would cause the release to fail, such as if a program is still inactive in SE38. An SAP function is used to perform this check and if any issues are found a message will be output.

Parameters

CHECK_CUSTOM_VALIDATION: Flag can be used to show releasability problems with BW/PBC objects specifically (as those sometimes have a seperate screen in the SAPGUI that highlight issues at the point of releasing the transport via SE09/SE10.

Notes

Please note that this Analyser currently does not work on non-ABAP systems being managed by CTS+.

5.1.2.10. Check Date (0012)

Check Date (0012) checks gets a unique list of tasks relating to the analysed transport requests and then checks the date contained in the specified custom field against the system date. The custom field date is compared to the system date based on the specified operator and if the comparison is true a message is issued stating that the change is not yet ready for deployment.

Parameters

Parameter	Description
ANALYSISCAPTION	Allows the analysis caption to be overridden. Enter the required text up to 100 characters. This text will appear in the analysis results.
ERRORIFNOTSET	If the valid to date is not set then display an error.
IGNOREURGENT	Ignore check if task priority is set to Urgent.
OPERATOR	Date comparison operator. Valid values: >, <, >=, <=, =, <>
TASKFIELDTEXTID	Custom field text ID for date (from /BTI/TE_CUSTF)

5.1.2.11. Conflict Analysis (0005)

Conflict Analysis check (0005) analyses a transport request against the target system and checks for conflicts before importing changes into that system. This is mainly used in the merge processing to identify objects changes in the merge target system that are also contained on the transport request being analysed. If any conflicts are found the details of the source system and merge target system transports are displayed along with the list of conflicting objects.

Parameters

Parameter	Description
CHECK_CONF_FLAG	If set, the analyser will only consider targets in the analysis that have the “remote import analysis” flag set.
EXCLUDE_MERGED	if set, the analyser will ignore any conflicts before last merge.
EXCLUDE_TOC	if set, the analyser will ignore any conflicts with Transport of Copies.
GLOBALMERGECHECK	By default, only transports that originate in the target system are checked for conflicts. If this flag is switched on, then all transports in the target are checked, regardless of their original system. There is a potential performance downside to having this functionality active.
SHOW_CONFLICTING_OBJ_KEYS	If set, the analyser will show the actual table entries that are in conflict. This is purely display but can still have performance implications since when you don’t show the keys, you can stop at the first conflict, but otherwise it needs to go on.
UNRELEASED_TRANSPORT_CHECK	If set, the analyser will also indicate in the results whether the conflicting transport is released or not.
USE_CACHE	If set, the analyser uses cached results, otherwise it is always recalculated.
USE_CURRENT	By default, if the transport is past the import queue, the conflict analysis will be done in the successor target. If this is set, we can force it to run it on the current one even in this case.

5.1.2.12. BW Conflict Analysis (0013)

BW Conflict Analysis (0013) looks for specific BI/BW objects that are renamed on import. As the names differ between systems the conflict analysis is not reliable so if any are found in they are reported. This specifically refers to Routines (ROUT) and formulas (RSFO).

Parameters

None

5.1.2.13. CTS+ Conflict Analysis (0011)

CTS+ Conflict Analysis (0011) performs the conflict checks for non-ABAP objects during the merge process to ensure that the transport being imported does not overwrite local changes in the merge target system. The transport attachments are exploded and the contents are analysed to determine if any conflicts are present.

For conflicting objects, the last changed timestamps are compared and a message is output if it has been found that the transport to be imported has an earlier timestamp than the one in the merge target system.

Parameters

None

5.1.2.14. MDM Conflict Analysis (0025)

MDM Conflict Analysis (0025) is used to identify MDM overtakes and conflicts. It will highlight:

- Other transports with full MDM exports that potentially conflict with the transports being analysed
- Transports that contain full MDM exports which potentially could conflict with everything else

Parameters

None

5.1.2.15. Check Transport Release (0014)

Check Transport Release (0014) checks whether the selected transport requests are released or not, and if not it issues a message to indicate this.

Parameters

None

5.1.2.16. Check Unicode (0015)

Check Unicode (0015) checks all transport request objects to see if they are Unicode compliant.

Parameters:

Parameter	Description
CHECKFLAG	A flag to indicate if the analysis should report programs that do not have the Unicode Check flag set. If the CHECKFLAG attribute is set (‘X’), any programs without the Unicode Check flag set will be reported.
CHECKCODE	Indicates if a Unicode Syntax check should be carried out on programs and the scope of the syntax checking. If the CHECKCODE attribute is set, a Unicode syntax check will also be carried out on the object. Setting this attribute to ‘1’ means the syntax check is carried out on all programs, setting it to ‘2’ will mean only programs without the Unicode Check flag set will be syntax checked.
CHECKSID	SAP System ID where the Unicode check is to be run

5.1.2.17. Check Unconditional Modes (0021)

Check Unconditional Modes (0021) will display the details of any Unconditional Modes set on the

selected transports. Unconditional Modes are added on Transport Forms in the Advanced Options tab.

Transports with the following modes set will be highlighted by this analysis:

- 2: Overwrite original objects
- 6: Overwrite objects in unconfirmed repairs
- 8: Import customer table entries

Parameters

None

5.1.2.18. Show Future Manual Steps (0022)

This analysis will display the details of any manual steps that will need to be completed during import of the selected transports into targets in the current path. If the parameter is set, any Manual Steps that have already been completed for future targets are also included in the output.

Parameters

Parameter	Description
DISPLAY_COMPLETED	Show Manual Steps that have already been completed.

5.1.2.19. Deep Impact Analysis (0023)

> ! Please note that this is a legacy analyser. It is replaced by Deep Impact Analysis (New) (0060).

Deep Impact Analysis (0023) can be used to highlight the lower level dependencies between the repository objects contained within the transport(s) being analysed and other transports existing in the landscape. This can be used to avoid deployment errors where transports are imported without the necessary dependent objects being present in the target SAP system.

The process checks the objects included in the analysed transports and determines whether any of the objects referenced within those are either part of the group of changes being approved/imported or have already been imported beforehand. For example, if a database table is being transported, a check would be made to ensure that all referenced data elements are not missing. The same would apply to a program being transported to ensure that any referenced data dictionary objects, function modules, classes, methods, etc. are not missing. This can then reduce the number of import failures due to hidden dependencies between changes and transports.

Deep Impact Analysis is used to perform technical / repository checks so therefore doesn't include configuration objects in its analysis.

Note that there is some pre-requisite Diffuser configuration setup that needs to be performed for Deep Impact Analysis (0023) to run correctly. This is detailed in this FAQ.

Parameters

Parameter	Description
GROUP_IMPORT_METHODS	Fast import/Import All methods to ignore transport sequence. Default is methods 2 and 5.
IGNORE_SEQUENCE	Ignore the transport sequence when calculating missing dependencies (transports with missing dependent objects that are contained in transports later in the import sequence will not be reported) . If not set then Fast import/Import All methods specified in GROUP_IMPORT_METHODS will ignore the transport sequence anyway. It's good practice to set IGNORE_SEQUENCE = 'X' on non-Import Queue.
LEVELS	Specify the number of levels to go down in object hierarchy.
MAX_OBJ_TIME	Maximum time to analyse an object. Default = 60 seconds.

5.1.2.20. Previous Import Errors (0024)

Previous Import Errors (0024) can be used to highlight any transports that are likely to fail when being imported into the next target, as they have failed during import to previous systems. If any of the selected transports were imported with errors into previous systems in the path, they will be highlighted by this analysis. The results of all imports into the same target will be displayed. Setting the below input parameter will suppress the display of any transports which had previous import errors, but for which the latest import was successful.

Parameters

Parameter	Description
IGNORE_LATEST_IMPORT_OK	Flag to indicate whether or not to ignore transports if the latest import was successful.

5.1.2.21. Check Import Order (0026)

Check Import Order (0026) can be used in import queues to check whether the selected transports are sorted into the correct ActiveControl calculated sequence prior to import. If the user has not sorted the transports into the correct order the analysis will highlight this fact along with the correct sequence that should be used.

Parameters

Parameter	Description
EXCLUDE_UNRELEASED	Flag to indicate whether or not to include un-released transports in the analysis.

IGNORE_COMPLETENESS	Flag to indicate whether the analysis should return an error if the sequence is correct but there are some missing transports in the selection.

5.1.2.22. Lock Control Point / Import Queue (0027)

Lock Control Point / Import Queue (0027) can be used to lock a control point or import queue to prevent approvals or imports. This can be useful when a system is taken off line for a refresh, to prevent any imports from being processed.

Parameters

None

> ! This particular analysis needs to be made mandatory so that it is always run when approvals or imports are processed. Otherwise it won't have the desired outcome of preventing Approvals and/or Imports.

5.1.2.23. Check Own Changes (0028)

Check Own Changes (0028) can be used to check the ownership of a Transport Form during approval / import. If the current user *is* the owner of any of the transports being analysed and they are not allowed to approve their own changes a message can be output to report this to the user and prevent approval / import of their own changes. If the current user *is not* the owner of any of the transports being analysed and they are only allowed to approve their own changes a message can be output to report this to the user and prevent approval / import of their own changes.

Parameters

Parameter	Description
PERMITOWNCHANGES	Flag to indicate whether the current user is allowed to approve their own Transport Forms or not. Blank = Not allowed to approve your own transports X = Only allowed to approve your own transports
OWNERFIELD	Field to be checked to determine the ownership of the Transport Form. If nothing is specified the ORIG_OWNER field will be checked. If you specify REQUESTOR, then it will check the current Transport Form owner.

5.1.2.24. Check SAP Objects and OSS Notes (0038)

Check SAP Objects and OSS Notes (0038) can be used to check whether any of the selected transport

requests contain changes to SAP objects or OSS notes.

If so, the relevant objects will be listed along with any OSS note numbers included in the transports.

Parameters

None

5.1.2.25. Check Calendar (0040)

Check Calendar (0040) analysis can be used to check a factory calendar (in any system) and be configured to give a warning message if today is not a ‘work day’ in the factory calendar.

Parameters

Parameter	Description
SYSTEM	The System ID of the system the factory calendar resides in.
CALENDAR	The ID of the factory calendar.
TEXT_NAME	The name of the standard SAP text (created using SO10) that is displayed if today is not a ‘workday’ in the factory calendar.

5.1.2.26. Check for Local Non-Transportable Requests (0041)

Check for Local Non-Transportable Requests (0041) can be used to check and report local non-transportable transport requests where the SAP transport target is blank.

Parameters

None

5.1.2.27. Dual Domain Controller Overview (0042)

Dual Domain Controller Overview (0042) can be used to check the status of any Dual Domain controller integrations. The transports are analysed to check whether the transfer of them to another ActiveControl Domain Controller can be carried out or not.

It will highlight the following:

- Requests that have already been interfaced
- Requests that have been filtered out based on the configuration (so will not be transferred)

- Requests that are merge requests so will not be transferred
- Requests that are valid and will be transferred

Parameters

None

5.1.2.28. Test Impact Radar (0043)

Test Impact Radar (0043) analysis can be used to highlight what top level SAP objects (e.g. Transactions, programs, etc.) are impacted by the selected transport(s). For example, if a database table structure was changed it could be used to identify all transactions and programs that use that table to help target testing effort.

It can also indicate what Test Scripts are impacted by the changes in the transports if these have been uploaded into ActiveControl. I.e. identify what test scripts should be executed at a minimum as a result of the changes contained in the transports.

Note that there is some pre-requisite setup that needs to be performed for the Test Impact Radar analysis check to run correctly. Please refer to ‘Test Impact Radar’ section of this Administration Guide for details of this setup.

Parameters

Parameter	Description
CHECK_LINKS	Check if the link creation is up to date.
MAX_LEVEL	Maximum recursion level for analyser (Default = 10).
MAX_OBJ_TIME	Maximum time to analyse an object (Default = 60 seconds).
TARGET	Target ID where the dependencies will be checked.

5.1.2.29. Check Unreleased Tasks in Requests (0044)

Check Unreleased Tasks in Requests (0044) can be used to check and report any unreleased sub-tasks inside transport requests.

Parameters

None

5.1.2.30. Check Request Tasks not yet in a TOC (0045)

Check Request Tasks not yet in a TOC (0045) can be used to ensure that all transport sub-tasks have been included in a Transport of Copies (ToC), for example, if ToC's are being used to deploy development changes to QA.

Parameters

None

5.1.2.31. TOC Analysis (0046)

TOC Analysis (0046) can be used to check if a transport is a Transport of Copes (ToC) and if so report it.

If, for example, ToC's have been used to deploy development changes to QA it could be that the ToC's should not move past QA to production in the transport path as the original development transports should be used in this case. The analysis can be used to prevent the ToC's from moving past QA in this case.

Parameters

None

5.1.2.32. BPCA (0047)

SAP Business Process Change Analyzer compares the objects included in a transport to the objects contained in the TBOMs of the target system.

BPCA (0047) hooks into that standard functionality to present the results information within ActiveControl.

Parameters

Parameter	Description
DESTINATION	System ID of the Solution Manager system where SAP Business Process Change Analyzer has been setup. (mandatory field).
EXCLUDE_GROUP	Can be used to exclude specified ActiveControl [Group](s) from the analysis.
EXCLUDE_TYPE	Can be used to exclude specified ActiveControl [Type](s) from the analysis.
FORCENEW	Can be used to force a new analysis, even if BPCA has run previously on the same transport(s).

MULTIPLE	Specify if you want to analyse multiple transports during a single analysis run.
VARIANT	Can be used to force a new analysis, even if BPCA has run previously on the same transport(s).

5.1.2.33. Changes to Same Objects (0048)

Changes to Same Objects (0048) can be used to identify other transports containing the same objects in the *same control point* as the ones being analysed. The original use case of this analyser was for a customer wanting to identify other transports containing same objects in a Test Queue, so that they were aware that other changes might be positively/negatively impacting the testing results.

Parameters

None

> To identify transports containing same objects in *other control points of the transport path*, you should use the newer Changes to Same Objects (Path) (0055) analyser.

5.1.2.34. Allowed Objects Check (0049)

Allowed Objects Check (0049) can be used to stop Business Tasks containing Transport Forms that contain anything other than a pre-defined set of objects being approved. The original user case for this analyser was for a customer that wanted to prevent Standard Changes containing anything other than specific objects.

Parameters

None

Configuration

Allowed Objects Check (0049) uses the same configuration tables as Risk Guard (0001) – however since they are different analysers, it is possible to configure both analysis to run at the same control point, but with different Risk Groups.

1) Create your Risk Group in table **/BTI/TE_RISKG** in the Domain Controller e.g. STANDARD
2) Define the SAP objects that you want to be part of the Risk Groups that you have created in table **/BTI/TE_RISKGOB** in the Domain Controller (e.g. your agreed list of objects that can be delivered in 'Standard' changes)
3) Populate table **/BTI/TE_ANRGTYPE** with the Task Types that you want to check (taking the serial ID from table /BTI/TE_TYPE) and the associated Risk Groups you created in (1) above.

5.1.2.35. Disallowed and Critical Objects Check (0050)

Disallowed and Critical Objects (0050) can be used to check if Transports contain particular objects. This Analyser is ostensibly the same as the ShiftLeft: Risk Guard analysis, but since it is possible to have both Analysers running in the same control point, you can set the Disallowed and Critical Objects Check analyser to not allow ‘Approve Anyway’ but set the standard Risk Guard to allow ‘Approve Anyway’.

Parameters

Parameter	Description
MINIMUMRISKLEVEL	Minimum Risk Level from table/BTI/TE_RISKL.
RISKGROUP	Risk Group from table /BTI/TE_RISKG.

5.1.2.36. Check Documentation (0051)

Check Documentation (0051) can be used to check if documents of particular categories have been attached/linked to a Business Task as part of the approval in a particular ActiveControl Control Point. This can help to automatically check if, for example, a functional specification has been attached to a Business Task at the point a Dev Outbox approval is being performed, or if UAT Test Evidence has been attached at the point a Production Inbox approval is being performed.

The Check Documentation analyser does not check the contents of the documents, it purely validates that the defined documents have been attached.
It is possible to check for a maximum of 5 different document categories in a particular control point.

Parameters

Parameter	Description
CATEGORY1	Populate with the document category that you want the Analyser to check for in this control point.
CATEGORY2	Populate with the document category that you want the Analyser to check for in this control point.
CATEGORY3	Populate with the document category that you want the Analyser to check for in this control point.
CATEGORY4	Populate with the document category that you want the Analyser to check for in this control point.
CATEGORY5	Populate with the document category that you want the Analyser to check for in this control point.

The Document Category’s used in this Analyser are configured in table /BTI/TE_ATT_CAT. Please refer to [Document Categories] section of this Administration

Guide for more information on this functionality.

5.1.2.37. Unit Test automation (0052)

Unit Test Automation (0052) can be used to automate unit testing in your SAP Development system. This analyser hooks into standard SAP functionality available within SAP Code Inspector, and ensures that Approvers at the configured control points (typically Dev Test Queue or Dev Outbox) have visibility of any development issues prior to the transports being released and deployed to QA systems.

Parameters

Parameter	Description
CIUSER	Username that the SAP Code Inspector variant has been created against.
CIVARIANT	Name of the SAP Code Inspector variant.

5.1.2.38. Check Component Version (0053)

Component Version differences (0053) is used to report if a transport has a different component version than the next system that it is to be imported into.

Parameters

None

5.1.2.39. Changes to same objects in path (0055)

Changes to same objects in path (0055) identifies what other transports in a transport path contain the same objects as those contained within the transport(s) being analysed. It was requested by a few existing customers that want to stop new versions being imported into a system until an older version that has already been imported into that system had been tested and deployed to Production

The analyser can look for the object in other transports i) Forward, ii) Backward or iii) Forward and Backward in the Path, depending on how the exact requirement

Parameters

Parameter	Description
CHECK_BACK_AND_FORWARD	Define whether you want to look only Forward, only Backward or both Backward and Forward in the Path.
IGNORE_SAME_TASK	Can be used if you want to ignore other transports assigned to same

	Business Task that contain the same object(s).
KEY_NO_RELATED_VIEWS	By default, when transporting table entries, views that use the same table in key fields will be considered a conflict. This behaviour can be turned off with this flag.
KEY_NO_WILDCARDS	If this flag is set, wildcards are not considered when searching for conflicting table entries.

Note that separate Changes to same objects (0048) analyser can be used to identify other transports containing the same object(s) in the same control point.

5.1.2.40. Check Custom Field entered (0057)

Check Custom Field Entered (0057) analyser can be used to check if a custom field has been populated or not. This is useful for customers that do not want a particular custom field to be mandatory at the point the Transport Form or Business Task is created, but want to ensure that it has been populated by a certain point in the process, e.g. for customers that have a separate Change Ticket raised as part of their process for moving SAP Change to Production.

Parameters

Parameter	Description
CHECK_TASK_AND_FORM	TASK or FORM or TASK & FORM – this is the level of the ActiveControl Hierarchy that you want the Analyser to check.
CUSTF1	This is the number of the custom field that you want to check. You can get this number from the [Fields] tab in the Windows GUI configuration screen.
CUST1_MSG	This is the message you want to appear in the analyser warning if the field is blank.

5.1.2.41. SCC1 Client Copy Check (0058)

SCC1 Client Copy Check (0058) can be used to check that a client dependant transport was copied to the other clients in that Development system.

Parameters

None

5.1.2.42. Check sibling system imports (0059)

Check sibling system imports (0059) can be used to analyse if a transport has been imported into a sibling system in a path. An example of its use is where an SAP customer has parallel regional SAP QA systems. Using this analyser can ensure that the transport does not progress onwards to Production until it has been approved and imported into all of the QA systems.

Parameters

Parameter	Description
TARGETROLEID	Enter the appropriate Role ID (from backend table /BTI/TE_TARGROLE) if you only want to check the sibling systems that are assigned to the specified target role. If it is not set, then the analyser will check all sibling systems.

5.1.2.43. Deep Impact Analysis (New) (0060)

Please note that Deep Impact Analysis (New) (0060) replaces the original Deep Impact Analysis (0023) analyser, from ActiveControl 7.0 onwards.

Deep Impact Analysis (0060) can be used to highlight the lower level dependencies between the repository objects contained within the transport(s) being analysed and other transports existing in the landscape. This can be used to avoid deployment errors where transports are imported without the necessary dependent objects being present in the target SAP system.

The process checks the objects included in the analysed transports and determines whether any of the objects referenced within those are either part of the group of changes being approved/imported or have already been imported beforehand.

For example, if a database table is being transported, a check would be made to ensure that all referenced data elements are not missing. The same would apply to a program being transported to ensure that any referenced data dictionary objects, function modules, classes, methods, etc. are not missing. This can then reduce the number of import failures due to hidden dependencies between changes and transports.

Parameters

Parameter	Description
GROUP_IMPORT_METHODS	(only relevant in import queues). The analyser already knows how to behave with standard AC import methods, assumes any unknown one to be sequential. Only set this if you use a non-standard fast import method.
INCLUDE_DUPLICATES	By default, when the Analyser find an object to be a missing dependency at the 1st level, it doesn't check it at the 2nd level to save time. When this

	is set, Analyser keep looking at every level. Gives better results but slows down the analyser, particularly when levels is high.
LEVELS	Determines how many levels the Analyser should travel the dependency tree. High value are slow and give many false positives. Recommend to stay between 1 and 3.
MAX_LINKAGE_AGE	Number of hours after which transports for unreleased transports are considered stale. We recommend leaving this to 1 hour.
MAX_LINK_TIME	Maximum total time in seconds to wait for the links to be created on the fly. Recommendation is 60 seconds.
MAX_OBJ_TIME	Maximum time in seconds to be spent analysing an object. Recommendation is 2-3 seconds.
RELEASE_TASKS	Release tasks of unreleased transports during link creation.

! A step by step setup – including configuration screenshots – on how to setup Deep Impact Analysis can be found in this online FAQ.

* Deep Impact Analysis is used to perform technical / repository checks so therefore doesn't include configuration objects in its analysis. It currently covers objects on ERP-type systems (ie ECC, CRM etc) and also BW and Gateway. It will not work on systems like Fiori or Java systems.

5.1.2.44. Critical Impact Analysis (0061)

Critical Impact Analysis (0061) can be used to check to identify critical impacts by using a combination of existing Risk Guard and Test Impact Radar functionality.

For an analysed transport, the check will:

1. Traverse up the dependency hierarchy to find all objects above that are possibly affected.
2. Check this set of objects against a set of objects configured in a Risk Group
3. The intersection of the two sets of objects creates the output

The output of the analyser shows:

i) a 'summary' the list of the objects in the risk group that are affected by the transport(s) being analysed.
ii) a 'detail' list for each of the objects in the 'summary' list, to show all of the objects from the original transport(s) that affect this particular risk group object.

Parameters

Parameter	Description
INCLUDE_DUPLICATES	if there are duplicates in the results, whether you want to see them or not. (the default is No).

LEVELS	How many levels up to check.
MINIMUM RISKLEVEL	Minimum Risk Level from table /BTI/TE_RISKL
RISKGROUP	Risk Group from table /BTI/TE_RISKG

5.1.2.45. Check Interdependencies in Path (0062)

Check Interdependencies in Path (0062) analysis uses a combination of the existing Test Impact Radar, Deep Impact Analysis and Overtake sequencing analysis checks to do the following:

1. Explodes out all dependent objects for a transport/change (to configurable number of levels)
2. Adds in all objects above the transport objects in the dependency hierarchy
2. Checks to see if any of the exploded out objects are in transports in the path, in the same target as the transport being analysed or in front of the transport being analysed (i.e. not in production yet) – to provide an intersection of the transports/objects between the transport(s) being analysed and other transports in the path.

It is possible to run the Check Interdependencies in Path analyser in three ways:

i) to report on the intersection of transports/objects if the intersecting transports do not have the same value as a Custom Field on the Business Task(s) being analysed
ii) to report on the intersection of transports/objects if the intersecting transports are not in the same Project(s) as the Business Tasks I am analysing
iii) to report on the intersection of transports/objects if the intersecting transports are not in the Business Tasks I am analysing.

Parameters

Parameter	Description
ANALYSIS_TYPE	Whether you want to run the Analysis by i) Custom Field, ii) Project or iii) Task.
CUSTOM_FIELD	The Custom Field number if you have selected that Analysis type.
LEVELS	Specify the number of levels to go down in object hierarchy.

5.1.2.46. Approver SOD Check (0063)

Approver SOD Check (0063) analysis can be used to ensure that the same Approver does not do two particular (configurable) control point approvals within ActiveControl. The analyser should be switched on in the second approval point, ie where you want to ensure the approver cannot be the same person that did the approval of the same transport(s) in an earlier approval point. These two approval points can be any locations in the transport path, they do not need to be adjacent approval points in the workflow.

Parameters

Parameter	Description
CONTROL_POINT	This is the prior Inbox/Test Queue/Outbox that you want to check against.
TARGET	This is the number of the prior Target that you want to check against.
MESSAGE	This is the custom message that you want to be presented to the Approver when they try to perform an approval when it was them that did the approval in the defined prior Target/Control Point.

5.1.2.47. Check SAP Import (0064)

Check SAP Import (0064) analysis can be used to perform an SAP-level check if a transport has been imported. If the transport has NOT already been imported, the analyser will return a warning.

Parameters

Parameter	Description
MSG	This is the message that will be presented to the Approver if the analysed transport(s) have not been imported to the defined SAP system.
SID	This is the SAP system that you want to check if the analysed transport has been imported.

5.1.2.48. Integration Status Check (0065)

Integration Status Check (0065) analysis can be used to perform an automated check to ensure that a corresponding ticket in an integrated ITSM or other external system is in the correct/expected status. The analyser will warn if the other tool ticket is not in the expected (configurable) status (as can potentially happen depending on the setup of the Integration / the external ticket being transitioned back in its workflow). For example – if you are about to approve some Transports to move to Production in ActiveControl, but the corresponding external ticket for that Business Task is somehow sitting in status “Awaiting UAT”.

Parameters

Parameter	Description
EXTERNAL_SYSTEM_ID	Needs to match what was configured for your external system in table /BTI/TE_INT_SYST.
MSG	The message you want to appear in the message output if the expected/actual status’ do not match.
STATUS_FIELD_NAME	This is the field name from the external System. E.g. status-name in the case of JIRA Status field.
STATUS_IN_EXTERNAL_SYSTEM	This is the Status field value from the external System that you want to check for (any other value will make the Analyser present a warning). Please note that [STATUS_IN_EXTERNAL_SYSTEM] is the technical value – not the Label that is seen by the user (in the

	case of JIRA). Also please note that the status parameter is case sensitive. It needs to match the technical field value in your external system exactly (eg “To Do” and not “TO DO” in the above screenshots).

Notes
(i) The analyser requires some of the standard Integration framework to already be setup for your external system. (i.e /BTI/TE_INT_SYST plus associated configuration tables).

(ii) The analyser has been build to satisfy a specific customer requirement for their existing ActiveControl integration with JIRA, but the reality is the Analyser can potentially be used with any ITSM / external tool that is supported by the ActiveControl Integration Framework.

5.1.2.49. Version Comparison (0066)

Version Comparison (0066) enables Customers to easily indicate any other inflight versions of an object being analysed, that have not yet reached Production – and also Production versions that are later versions of the same objects in the transport being analysed.

Unlike most of the other ActiveControl analysers, Version Comparison analyses at an SAP level, and therefore also reports on transports that do not have Transport Forms.

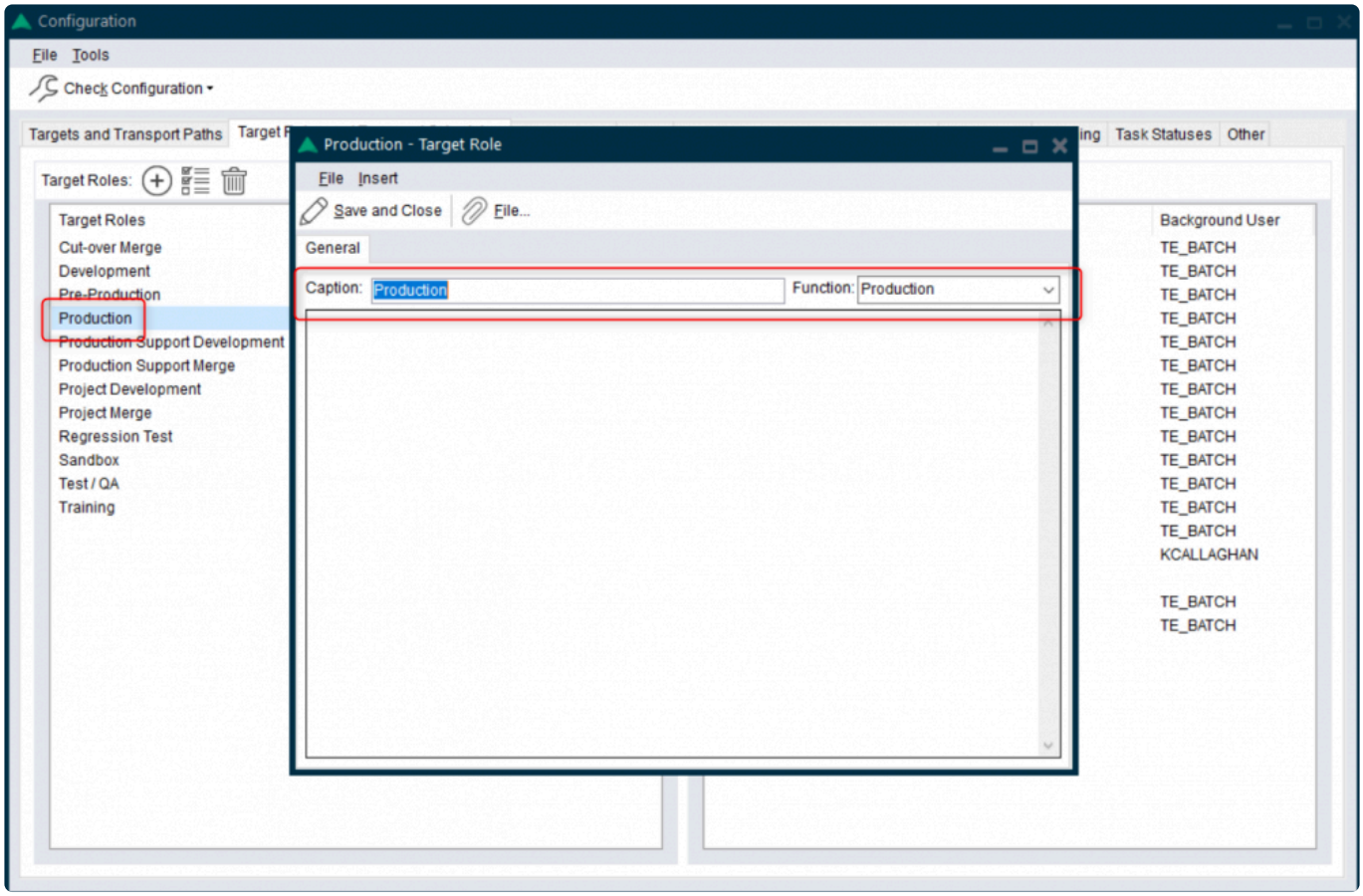
Parameters

None.

Additional Notes

This Analyser makes use of Target Roles and the Production function – in order to be able to determine what system is the Production system in the path.

As such, for the Analyser to work, you need to (1) link your Target Role to the “Production Function” as per below screenshot – and also (2) ensure that all your Production targets are linked to your Production target role



> Because Version Comparison (0066) also looks at transports that do not have Transport Forms, this analyser could be particularly helpful within organisations that have lots of historical orphaned transports that have not been deployed right through the SAP landscape, and that are not migrated into ActiveControl.

5.1.2.50. Impacted Batch Jobs (0067)

Impacted Batch Jobs (0067) allows customers to easily identify which scheduled Batch Jobs are potentially impacted by the objects being delivered in SAP transports. It was added in response to several customers requesting the ability to be able to easily avoid changes being moved to Production at the same time as critical batch jobs were scheduled to run.

This Analyser uses the same Diffuser-based object linkaging technology as that employed by Deep Impact Analysis (0060) and several other Analysers. As a result, setting up Deep Impact Analysis should be seen as a pre-requisite for using this analyser.

Parameters

Parameter	Description
INCLUDE_DUPLICATES	If there are duplicates in the results, whether you want to see them or not. (the default is No).

LEVELS	How many levels up to check.

5.1.3. Custom Analysis Types

Custom analysis types can be created by customers to complement the standard-out-of-the-box ShiftLeft analyzers. In order to do this, entries must be maintained in the following tables:

Table	Usage
/BTI/ TE_ANLTYPE	Lists all the analysis types currently available within ActiveControl
/BTI/ TE_ANLTYPEP	Contains all the analysis type parameters. These are only required for the types that need them
/BTI/ TE_ANTYPEPT	Analysis parameter types: 01 BOOL TRUE or FALSE 02 COMBO Combination 03 COMBOLIST Combination list 04 DATE Date value 05 STRING String value 06 MEMO Multi-line 07 NUMERIC Numeric value
/BTI/ TE_ANREASON	Lists all the analysis error reason codes
/BTI/ TE_ANREAS_P	Configuration used to control the approval / import anyway functionality. This is stored by Reason, Analysis type, Target and Location. • If the Analysis type is blank the config applies to all analysis types • If the Target is blank or 9999 the config applies to all targets • If the Location type is blank the config applies to all locations
/BTI/ TE_ANRELINK	Used to link the error reason codes to the relevant analysis types

To implement an analysis type one of the standard ShiftLeft analysis function modules should be copied to a Z version so the framework and parameters can be copied and utilised. The recommendation is to copy function module /BTI/TE_ANALYSIS_CHECKDATE as this FM utilises analysis parameters and custom field processing both of which are likely to be required in most custom analysis types.

It is recommended to number any custom analysis types and reason codes starting with a ‘9’ to distinguish them from standard ones and to prevent them from being overwritten during upgrades. E.g. 9001

Once the table entries and function module are available they will appear for use within the ActiveControl configuration alongside the standard ShiftLeft analysers.

5.2. Rules Engine

5.2.1. Introduction

Background

Skipping and Approvals are features used by almost all ActiveControl customers. Despite their popularity, Basis Technologies have occasionally received feedback from some users that both parts of the product could benefit from additional flexibility and versatility such as:

1) To be able to define Approvers based on a combination of criteria, and not only the Transport Form [Group] field.
2) To be able to define Peer Reviewers for a Dev Outbox, based on Business Task assignment instead of Windows GUI configuration.
3) To be able to create AND, OR and NOT logic-based skipping rules.
4) To be able to create rules whereby transports created by specific external resources stopped for additional approvals.
5) To be able to create rules whereby particularly old transports stopped for additional approvals.
6) To be able to create ‘template’ skipping rules, that can be reused to avoid having to configure duplicated rules.

During the last few months, Basis Technologies have worked on a co-innovation project with a large SAP user to enhance ActiveControl to support their complex landscape and process requirements. To achieve their requirements – and to deliver on the feedback received from them and other ActiveControl users – a new Rules Engine has been developed which can be used to create more complex rules for both Skipping and Approvals. It is also expected that this Rules Engine concept will be extended in future releases of ActiveControl to enable rules to also be created in other areas of the product such as Testing, Import Schedules and Email Notifications.

Key Terminology

Consumers: Consumers are the ActiveControl functionalities that make use of the Rules Engine. For ActiveControl 8.0, a Skipping consumer and an Approver consumer are available.

Step: A Step is a condition that will return True or False depending on the particular logic it is coded to perform. (eg if a field = a certain value). Steps can be created for Transport Form, Business Task or Project level conditions. Some Steps will call on other child steps as part of their evaluation (eg AND and OR) and are essentially Parent Steps, whereas other Steps might perform the entire evaluation themselves.

Rule: A Rule is essentially the top-level ‘parent’ Step which is then called by the Consumer.

Compatibility

The Rules Engine solves lots of complex requirements, but the reality is that many ActiveControl customers do not have such complex requirements. As such, the Rules Engine has been added to the product in such a way that it is essentially optional functionality. Individual ActiveControl customers can therefore choose whether they want to use the Rules Engine, or continue to use the simpler (but less functionally rich) Approvals and/or Skipping capabilities that were already available prior to ActiveControl 8.0.

For existing ActiveControl customers that want to use the Rules Engine as part of an Upgrade to ActiveControl 8.0 or later versions, a conversion exercise will need to be performed to convert their existing Skipping rules in /BTI/TE_SKIPCP to the new structure used within the Rules Engine. No functionality will be provided by Basis Technologies to automate this, but our Solution Specialists can of course support customers performing this conversion as part of a formalised Upgrade project.

Rules Engine – Technical Summary

1.	Summary	Technical Details	Notes
1.	RFCs	No additional RFCs required.	–
2.	Users	No additional Users required.	–
3	Authorisations	No additional Authorisations required.	–
4.	Configuration Tables	Consumer activation	/BTI/TE_RE_CONSE
		Steps	/BTI/TE_RE_STEP
		Rules	/BTI/TE_RE_RULES
		Step Conditions	/BTI/TE_RE_STEPC
		System Types	/BTI/TE_RE_SYSTP
		Variables	/BTI/TE_RE_TVARV
		Resusable Steps	/BTI/TE_RE_STEPR
5.	Consumer Tables	Skipping	/BTI/TE_RE_SKPCP
		Approvals	/BTI/TE_RE_APPRV
6.	Programs	Rules Harness	/BTI/TE_RURE_TEST_HARNESS
7.	Jobs	None required	–
8.	User Exit	None required	–

The rest of this section of the Administration details the configuration of the Rules Engine.

5.2.2. Core Configuration

This section details the core Rules Engine configuration tables.

Table	Description
/BTI/TE_RE_CONSE	Table where Rules Engine Consumers are switched on. Otherwise, the legacy Skipping and/or Approvals functionalities are used.
/BTI/TE_RE_STEP	Table where all parent and child Steps are defined.
/BTI/TE_RE_RULES	Table where the overarching parent step Rules are defined (and also briefly documented if desired).
/BTI/TE_RE_STEPC	Table where each child Step Conditions are defined.
/BTI/TE_RE_SYSTP	Table where Systems are associated as Development, QA, Production systems. This is required if you want to create System-level rules as part of the Skipping and/or Approver consumer configuration tables.

Tips for Rules Engine Configuration

1. Map out your intended rules diagrammatically before starting any configuration. This will not only aid your thought process in determining what child / parent steps you need to make up your overall Rule, it will also help Basis Technologies assist you more easily in the event you need to raise a support ticket.

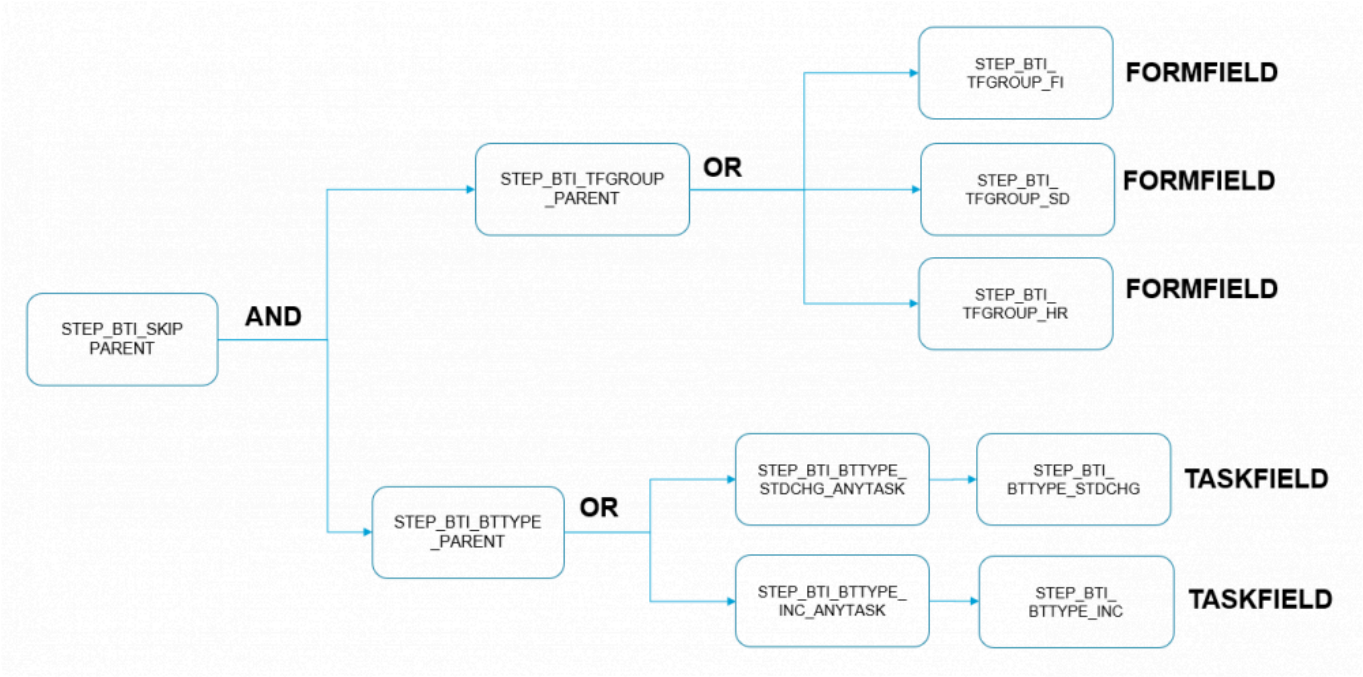


Figure: Example of planning out an intended Rule diagrammatically

2. Give all your steps meaningful names, and use a consistent naming convention. This will make it a lot easier to audit and maintain your Rules Engine configuration later.

3. When configuring, create the Steps from left to right. This is required because you cannot define a child step to a parent step unless the parent step has already been configured.

4. If you make any changes to the Rules Engine configuration tables, you must log out of the Windows GUI and back in before they become active. This is not a new phenomenon, it was also the case in previous Skipping and Approvals configuration.

5.2.2.1. /BTI/TE_RE_CONSE

Table /BTI/TE_RE_CONSE is used to switch on/off individual consumers within the Rules Engine. If this table is not configured, then the Skipping and/or Approvals functionalities that were available prior to ActiveControl 8.0 will be used, using /BTI/TE_SKIPCP and Windows GUI Inbox/Outbox Approver target configuration respectively.

Field	Description
Consumer	This is the Consumer being switched on, the variables comes from preconfigured /BTI/TE_RE_CONSU table: APPROVAL_RULE_ENGINE : Used to enable/disable the Rules Engine Approvals consumer. SKIP_RULE_ENGINE : Used to enable/disable the Rules Engine Skipping consumer.
Enabled	This is used to switch on/off the Consumer.
Filter_Field	(optional) if you want to use Filters, the Field name is entered here. See Filters section for more information.

5.2.2.2. /BTI/TE_RE_STEP

Table /BTI/TE_RE_STEP is where all the simple/multilevel steps used by the Rules Engine are configured.

Field	Description
Step ID	Unique name of the Step.
Parent Step ID	Used when you are creating a child step that forms part of a parent step (eg when you have an AND logic parent step consisting of two child steps. Note that the Parent Step also needs to be defined as a seperate Step in this table, and you must configure this first.
Step Type	See list of available Step Types below.

Step Types

The following details the available Step Types that can be configured in /BTI/TE_RE_STEP (the most common types are detailed at the top).

Step Type	Description
FORMFIELD	This can be used to evaluate a value in a standard or custom field on a Transport Form. Note: Step type FORMFIELD cannot be a child node to ALLTASK/ANYTASK.

TASKFIELD	This can be used to evaluate a value in a standard or custom field on a Business Task. Note: Step type TASKFIELD can only be a child node to either ALLTASK or ANYTASK
PROJFIELD	This can be used to evaluate a value in a standard Project field.
ALLTASKS	This can be used to evaluate whether ALL of the Business Tasks assigned to a Transport Form satisfy the child condition. This has been added for the benefit of a few longterm ActiveControl customers that associate a Transport Form with more than 1 Business Task (although this is generally not something Basis Technologies recommend in newer versions of the product due to the concept not working well with newer capabilities such as Partial Testing).
ANYTASKS	This can be used to evaluate whether ANY of the Business Tasks assigned to a Transport Form satisfy the child condition. Similar to ALLTASKS, this has been added for the benefit of a few longterm ActiveControl customers that associate a Transport Form with more than 1 Business Task.
AND	This can be used to evaluate several steps as an AND scenario, whereby all the steps must be TRUE for the result to be TRUE. In this instance, the parent step would have the AND Step Type, and the child steps would each have their own condition. Note: Step type AND must have at least 2 child nodes
OR	This can be used to evaluate several steps as an OR scenario, whereby if any of the steps is TRUE, then the result is TRUE. In this instance, the parent step would have the OR Step Type, and the child steps would each have their own condition. Note: Step type OR must have at least 1 child nodes
TASKPROJECT	This can be used to evaluate an ActiveControl Project against the Business Task.
ALWAYSTRUE	This can be used when you always want the result to be TRUE. A common use case for this is in ALLGROUPS approval process. Please refer to this FAQ for an example configuration of this scenario.
CHECKSTEP	This is used in conjunction with Reusable Steps.
RISKCHECK	This can be used to evaluate whether a transport contains risk objects, similar to what was possible in /BTI/TE_SKIPCP previously. Note that only the relational operators (eg =, <>) can be used as conditions against this step type.
CONTEXT	This Step Type should be used for Fields which are neither on FORM or TASK. For example, it can be used in conjunction with TARGET or LOCATION conditions in /BTI/TE_RE_STEPC.
CUSTOM_STEP_TYPE	This can be used by customers wanting to create Custom Step Types – see here for further information.

ALLTASK/ANYTASK can only have one child node which is either AND or OR(followed by multiple TASKFIELD) or 1 TASKFIELD or or 1 TASKPROJECT.

5.2.2.3. /BTI/TE_RE_RULES

Table /BTI/TE_RE_RULES is where the overarching parent step to be used in the Consumers are defined. This mandatory table is also useful for summarising textually what each Rule is for, making it easier to audit and maintain Rules later. (this is something that customers of /BTI/TE_SKIPCP concept have often asked for in the past).

Field	Description
Step ID	Name of the main Step.
Consumer	(Optional) Can be used to refer to the Consumer (eg Skipping, Approvals) that the Rule will be used for.
Step Description	(Optional) Can be used to summarise the purpose of the Rule.

5.2.2.4. /BTI/TE_RE_STEPC

Table /BTI/TE_RE_STEPC is where Step Conditions are defined.

Field	Description
Step ID	This is the Step ID as defined in table /BTI/TE_RE_STEP.
Field	The current available Fields supported are detailed in table below.
Operator	The current Operators supported are as follows: = <> < <= > >= IN REG_EXP INITIAL NOTINITIAL
Value	The Value format will depend on the Field. See below table for examples.

Available Fields

The following table details the Field variables that are typically configured in /BTI/TE_RE_STEPC. The Fields at the top of the list are the ones that are more likely to be widely used.

Available Field	Value	Other Information
GROUPID	Value will be the Group GUID from /BTI/TE_GROUPS	Refer to this online FAQ for a configuration example

TYPEID	Value will be the Type GUID from /BTI/TE_TYPE	Refer to this online FAQ for a configuration example
[CUSTOM FIELD NUMBER] eg 510	(custom fields at Transport Form, Business Task and Project level are all supported)	Refer to this online FAQ for a configuration example
PROJECTID	Value will be the Project GUID from /BTI/TE_PROJ	
PHASEID	Value will the nnnn Phase ID, from /BTI/TE_PHASE	
PHASEID	Value will the nnnn Phase ID, from /BTI/TE_PHASE	
START_DATE	Value will be in YYYYMMDD date format	
CLIDEP	Value will be X for client dependent	Typically used to avoid the symptom described in this online FAQ
[RISK GROUP] eg BT0001	Value will be 1-digit number – for the Risk Level	Refer to this online FAQ for a configuration example
PRIORITY	Value will be 1, 2, 3 or 4 (Low, Normal, High, Critical respectively)	Typical use case is for high priority changes to skip certain approval points, or stop for additional approvals, depending on the Customer's process
TESTERID	Value would be something like JROBERTS.	Refer to this online FAQ article for a configuration example.
REFERENCE	Value will be the Business Task Reference field value, eg CRQ0001293	
CAPTION	Value would be a text string.	
NUM_OBJECTS		Possible use case would be to stop transports containing high numbers of objects for additional approval
NUM_KEYS		
PATH	Value would be something like 02 (ie the 2-digit number of the path, from /BTI/TE_PATH	
TARGET	Value would be something like 0005 (ie the 4-digit number of the path, from /BTI/TE_TARG	
STAT_DEPL		
STAT_PLAN		
TRKORR	Value would be something like ACDK904698.	
CREATION_DATE		Possible user case could be used to drive an additional approval on transports older than X years
CREATION_TIME		CREATION_DATE is probably more likely to be used.
CREATED_BY	This is the username of the	

	person that created the AC Transport Form.Value would be something like RROBB.	
REQUESTOR	This is the username of the person that created the SAP transport. Value would be something like JBARTER.	Possible use case would be to force additional approval on transports created by certain users (eg external consultants). Refer to this online FAQ article for a configuration example.

Available Operators

Operator	Description	Other Information
=	Equals	Note, if you want to create a condition based on a field being blank, you would use the Operator = and the Value “”
<>	Not Equals	Note, if you want to create a condition based on a field not being blank, you would use the Operator <> and the Value “”
<	Less than	
<=	Less than or Equal To	
>	Greater than	
>=	Greater than or Equal To	
IN	Within a range of Values	Please refer to this online FAQ article for example configuration of a Range based IN operation.
INITIAL	Checking if a field is blank	
NOTINITIAL	Checking if a field is not blank	

Custom Field’s supported

Field Type \ Supported ?	Transport Form	Business Task	Project
Text	Supported	Supported	Supported
Dropdown	Supported	Supported	Supported
Checkbox	Not Supported	Not Supported	Not Supported
Date	Supported	Supported	Supported
Time	Supported	Supported	Supported
User Selection	Supported	Supported	Supported
Selection List	Supported	Supported	Supported
Option (Radio Button)	Not Supported	Not Supported	Not Supported
Calculated	Supported	Supported	Supported

5.2.2.5. /BTI/TE_RE_SYSTP

Table /BTI/TE_RE_SYSTP is used to map Targets configured in the Windows GUI to **System Types** used in the Rules Engine.

This configuration table is only required if you want to be able to configure the Consumer tables based on System Type / Location level, instead of Target/Location.
Some more information on why you might want to use System Types is detailed in this online FAQ article.

Field	Description
System Type	eg Development, Quality Assurance, Production.
Target	{Target Number} per the target configured in the Windows GUI.

> ! The Rules Engine does not use Target Roles defined in the Windows GUI to group systems together.

5.2.2.6. /BTI/TE_TVARV

RE_CACHE_ON variable should be added to table /BTI/TE_TVARV, with Low value 90.

This is required to optimise the performance of the Rules Engine, particularly around the Approvals Consumer in the Web UI.

5.2.3. Consumers

5.2.3.1. Skipping (/BTI/TE_RE_SKPCP)

Skipping consumer is used in conjunction with the Rules Engine to enable customers to allow Transport Forms to bypass specified locations (Inboxes, Import Queues, Test Queues, Outboxes) based on configurable logic. The skipping consumer replaces table /BTI/TE_SKIPCP configuration that was available in earlier versions of ActiveControl.

/BTI/TE_RE_SKPCP

Table /BTI/TE_RE_SKPCP is the consumer table for Skipping rules.

Field	Description
Filter	(optional) Please refer to Filter section.
System Type	System Type as defined in /BTI/TE_RE_SYSTP. You can create rules based on either System Type/Location combination OR Target/Location. You cannot create a rule with both System

	Type and Target.
Target	Target number, as defined in Windows GUI configuration. You can create rules based on either System Type/Location combination OR Target/Location. You cannot create a rule with both System Type and Target.
Location	I (Inbox), Q (Import Queue), T (Test Queue) or O (Outbox).
Step	The overarching Rule, as configured in /BTI/TE_RE_RULES.
Priority	Number sequence (1,2,3 etc). This is the order in which the rules will be processed. See additional Priority note below.
Skip Action	'Skip' or 'Do Not Skip' or 'Delete'
Def on Exception	"Defer on Exception" is an optional setting. If switched on, then if there is an exception, then the action (Delete/Skip/NoSkip set for the rule will be carried out, but a system error will also be logged.

As mentioned above, you must configure EITHER a System Type or a Target/Location combination. You cannot configure both within the same entry. If you configure an entry against target/location and another entry against system type which includes that target, then both records will be considered during the rule execution.

Priority is used to create a sequence in which the rules should be processed. In the old /BTI/TE_SKIPCP table, 'Do Not Skip' always took precedence over a skip rule in the same location. The priority field in the Rules Engine allows the ActiveControl Administrator to define which rule should be treated with the greatest importance (ie Priority = 1). You should not configure two rules with the same numerical priority. Although Priority is not a mandatory field, it is recommended to maintain it, as otherwise the Rules Engine will just run the rules in the order in which they are maintained in the table.

5.2.3.2. Approvals (/BTI/TE_RE_APPRV)

Prior to ActiveControl 8.0, it was only possible to define approvers within ActiveControl based around the Transport Form [Group] field. Some customers – particularly larger enterprises with more complex organisational structures – told us that this Approval structure could sometimes be limiting, and that they would like the flexibility to be able to assign Approvers more granularly, such as

1) Different Approvers for CRs and Incidents (ie Business Task [Types].
2) Different Approvers for transports relating to different Projects.

ActiveControl 8.0 enables this increased flexibility, via the Approvals consumer that forms part of the core Rules Engine. It should be noted that this increased flexibility comes at the cost of increased configuration – as it requires several backend tables to be configured. For customers that do not want to use the Approvals consumer within the Rules Engine, they can simply continue to use the existing Approvals configuration done via the Windows GUI Target screens. It is not possible to use a mixture of the Rules Engine approvers and WIndows GUI approver structure.

/BTI/TE_RE_APPRV

Table /BTI/TE_RE_APPRV is the consumer table for Approver rules. If you want to have Inbox and Outbox Approvers in your workflow based on fields other than the Transport Form [Group] field, then this is where you define them.

Field	Description
Filter	(optional) Please refer to Filter section.
System Type	System Type as defined in /BTI/TE_RE_SYSTP. You can create rules based on either System Type/Location combination OR Target/Location. You cannot create a rule with both System Type and Target.
Target	Target number, as defined in Windows GUI configuration. You can create rules based on either System Type/Location combination OR Target/Location. You cannot create a rule with both System Type and Target.
Location	I (Inbox), Q (Import Queue), T (Test Queue) or O (Outbox)
User Group	User Group, as per configuration in /BTI/TE_ROLEU table.
Group Type	The current available Group Types supported are detailed in table below.
Step	The overarching Rule (Step) configured in /BTI/TE_RE_RULES.

Group Types

Group Type	Description
ROLE	If you want your Approvals to be based on the Users defined against the User Role (as stored in /BTI/TE_ROLEU / /BTI/TE_ROLEUX)
PROJECT	If you want your Approvals to be based on the Users maintained against a Role on a particular Project (as stored in /BTI/TE_PROJU)
TASK	If you want your Approvals to be based on the Users maintained against a Role on a particular Business Task (as stored in /BTI/TE_TASKU)

The Approvals consumer uses pre-existing tables /BTI/TE_ROLEU and /BTI/TE_ROLEUX for the definition of User Roles and User assignments to those roles. These are configured in the Windows GUI, the SAPGUI tables do not need to be maintained directly in the backend.

5.2.3.3. Consumer Configuration Examples

The ActiveControl Rules Engine offers a lot more flexibility for Skipping and Approvals than was

available prior to ActiveControl 8.0, however does require a greater level of configuration to achieve these more powerful scenarios.

To help our customers get up to speed quickly with the configuration, we have created some online Knowledge Articles of how to setup common scenarios:

Skipping FAQs

Scenario	Knowledge Article Link
HOW DO I: create a single condition Transport Form standard field skipping rule?	Click here
HOW DO I: create a single condition Business Task standard field skipping rule?	Click here
HOW DO I: create a skipping rule based on a Custom Field?	Click here
HOW DO I: create a skipping rule of client dependent transports?	Click here
HOW DO I: create a skipping rule based on OR condition?	Click here
HOW DO I: create a skipping rule based on Project Phase?	Click here
HOW DO I: create a skipping rule based on objects contained in rule?	Click here
HOW DO I: create a skipping rule based on AND condition?	Click here
HOW DO I: create a skipping rule in the Rules Engine based on Tester?	Click here
HOW DO I: create a skipping rule IN a range of values?	Click here
HOW DO I: create a skipping rule based on NOT IN a range of values	Click here
HOW DO I: create a skipping rule based on a Calculated Custom Field?	Click here

Approvals: Configuration Examples

Scenario	Knowledge Article Link
HOW DO I: create Approver structure based on Transport Form [Group], with an ALLGROUPS approver?	Click here
HOW DO I: create Approver structure based on Transport Form [Group] AND Business Task [Type]	Click here
HOW DO I: create Approval structure based on Transport Form [Group] AND [Project]	Click here
HOW DO I: configure a Peer Review process for an Inbox or Outbox ?	Click here
HOW DO I: configure Project Managers to perform approval of RICEFs in their Project?	Click here

Other Rules Engine FAQs

Scenario	Knowledge Article Link
HOW DO I: create a custom node type?	Click here
Why do I need a ANYTASK or ALLTASK parent step when creating a TASKFIELD step in Rules Engine?	Click here

5.2.4. Rules Test Harness

A **Rules Test Harness** has been built as part of the Rules Engine solution. This was initially built for Basis Technologies internal use to help our Development and QA team test the Rules Engine, however it is also an extremely useful mechanism for our customers to validate how a particular transport will behave within the ActiveControl workflow as a result of their Rules Engine configuration, or indeed to troubleshoot any unexpected behaviour resulting from their Rules Engine setup.

The Rules Harness is accessed via program **/BTI/TE_RURE_TEST_HARNESS** in the Domain Controller.

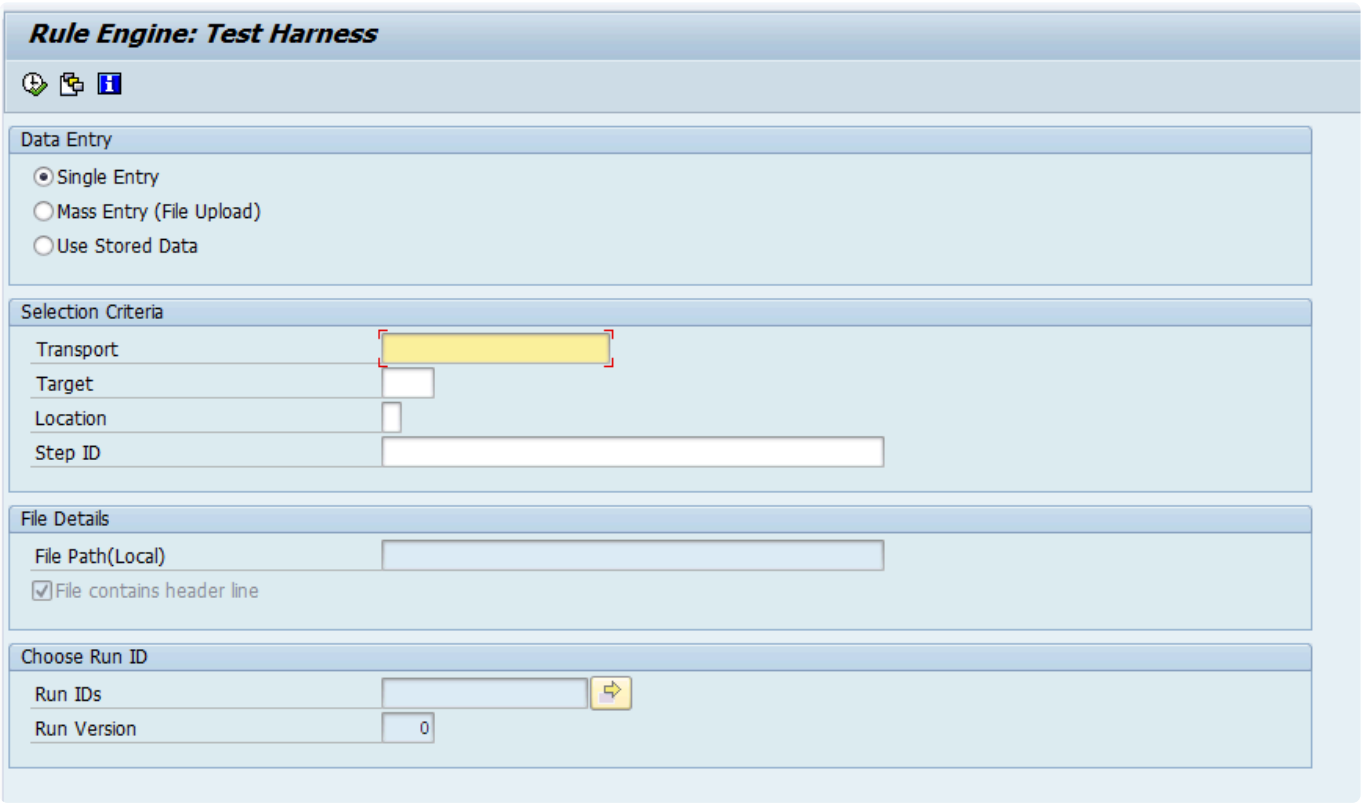


Figure: The Rules Harness

Selection Screen

Field	Description
Data Entry	Single Entry : Should be used when you want to evaluate a specific transport against a Rule. Mass Entry : Should be used if you want to evaluate a spreadsheet list of transports against

	a Rule Used Stored Data : This can be used if you want to analyse a historical Validation run
Transport	The Transport against which you want to validate against a Rule
Target	The target against which you want to validate a Rule
Location	The location of the target defined above, against which you want to validate a Rule
Step ID	The Rule against which you want to validate a Transport
File Path (location	Used in conjunction with the 'Mass Entry' mode. Please refer to this FAQ for instructions on how to use this data entry mode.
Run IDs	Used in conjunction with the 'Use Stored Data' mode. This mode is primarily for Basis Technologies internal usage.
Run Version	Used in conjunction with the 'Use Stored Data' mode. This mode is primarily for Basis Technologies internal usage.

5.2.5. Troubleshooting Issues

Validating Rules

The Rules Test Harness is a really useful tool for testing configuration and troubleshooting issues. When creating new rules or maintaining existing ones, it is strongly recommended to evaluate some transports with the Rules Harness, to check that the configuration is working as expected.

Error Logging

Unlike the historical /BTI/TE_SKIPCP, a new error logging capability is included as part of the Rules Engine. This error logging is aimed at helping customers quickly identify and troubleshoot configuration errors in their setup during the implementation or ongoing maintenance of the Rules Engine.

Errors are logged within the "System and RFC Errors" section of the Windows GUI, and are also sent to ActiveControl Administrators via email notification, if the corresponding 'System Errors' notification is switched on within /BTI/TE_RNOTIFICATION_ENGINE variant.

Common Configuration Errors

The following FAQ articles summarise some of the most common configuration issues on the Rules Engine.

Error	Knowledge Article Link
"StepID [XYZ] is not maintained as Skip Rule in table /BTI/TE_RE_RULES"	Click here
"StepID [ABC] cannot evaluate a subject of type FORM for the step type	Click here

TASKFIELD”	

5.2.6. Advanced Topics

5.2.6.1. Ranges

Ranges can be used in conjunction with the IN operator, to create a step condition based on one of a range of values. This can be useful in certain scenarios to help significantly cutdown on the number of steps/rules that you might otherwise need.

For details of how to configure a Range-based Rule, please refer to this online FAQ article.

/BTI/TE_RE_TVARV

Table /BTI/TE_RE_TVARV is used to as part of the Contains / IN operator within the Rules Engine.

Field	Description
Variable Name	Please see underneath for the variables relating to the Rules Engine.
Number	This can be 1 (unless several line items are required, in which case they should be in numerical sequence, 1,2,3 etc.
INCL / EXCL	Optional, will depend on the variable.
Option	Optional, will depend on the variable.
Selection Value	Optional, will depend on the variable.

Variable Name	Description
[RANGE NAME]	Refer to FAQ article linked earlier in this section.
[CONTAINS]	Refer to Contains section.

! Please note that the Rules Engine variable table is different to the /BTI/TE_TVARV table used by other functionalities within ActiveControl.

5.2.6.2. Contains

CONTAINS can be used in conjunction with the IN operator, to create a step condition based on the value containing a string within that value. This can be useful in certain scenarios to help significantly cutdown on the number of steps/rules that you might otherwise need. For example, if you have a dropdown with values such as:

EU – France
EU – Germany
EU – Italy
EU – Greece
APAC – Australia
APAC – New Zealand
APAC – Malaysia
AFRICA – South Africa
AFRICA – Zimbabwe
AFRICA – Ethopia
LATAM – Colombia
LATAM – Brazil
LATAM – Argentina

…and want to create a skipping rule based on the selected value being a European country – you can now achieve this with one single CONTAINS step/condition (EU*), instead of having to create a step/condition for each of the individual EU country values.

Please refer to this online FAQ for a configuration example of a CONTAINS logic step.

5.2.6.3. Reusable Steps

Table /BTI/TE_RE_STEPR can be used to define child steps / conditions for reuse – to avoid duplicated creation of complex rules.

Field	Description
Step ID	Text description of the step.
Reusable Step	Text name of the reusable step that will then be configured in /BTI/TE_RE_STEP.

Please refer to this online FAQ for a Reusable Step configuration example.

Note that you do not to create a reusable Step in the scenario where you want to reuse the same complex step / rule within the Consumers. The same Step/Rules can be used against different Consumer locations.

Resusable steps are really only beneficial for complex rules involving multiple child steps

(eg AND or OR). There is no real benefit of having a resusable step for a single condition step, you are better just to duplicate that in /BTI/TE_RE_STEP and not use /BTI/TE_RE_STEPR.

5.2.6.4. Custom Node Types

Custom Node Types can be used by customers wishing to create their own custom node types that run within the Rules Engine architecture. This is an advanced capability of the Rules Engine which Basis Technologies do not envisage many customers needing to use, given the standard /BTI/TE_RE_STEP and /BTI/TE_RE_STEPC tables should cover the vast majority of desired condition use cases.

/BTI/TE_RE_STEPT

Table /BTI/TE_RE_STEPT is an application table used by the Rules Engine. It is pre-configured with the standard classes used by the Rules Engine to perform the various functionality, and should not need to be maintained by the majority of customers. The only exception to this is for customers creating custom node types. In this instance, the custom step type and implementing class must be defined in /BTI/TE_RE_STEPT

Please refer to this online Knowledge Article for a configuration example of a Custom Node Type.

5.2.6.5. Filters

Filters can be used on the Consumer configuration tables in conjunction with Calculated Custom Fields (or indeed any other fields), to limit (filter) the number of Skipping or Approval rules to be evaluated.

This is optional functionality that Basis Technologies does not anticipate needing to be widely used in most ActiveControl-managed SAP environments.

5.2.6.6. Calculated Custom Fields (CalcCF)

Calculated Custom Fields enable ActiveControl customers to write their own custom calculations and logic upon which Rules Engine rules can be based. Essentially this custom calculation is written to a custom field on the Transport Form or Business Task, which then drives behaviour of the Rules Engine consumers.

A new calculated custom field type has been created to support this new functionality. This is configured via the Windows GUI. A new implementation method/class is available for customers to write their own complex custom calculations, which will result in a value being populated in the Calculated Custom

Field.

Please refer to this online FAQ for a Calculated Custom Field configuration example.

/BTI/TE_CCF_IMPL

It is possible to have more than one calculated custom field (CCF). Table /BTI/TE_CCF_IMPL is used to associate the CCF to the specific implementor.

Field	Description
Field ID	This is the field number of the calculated custom field (as seen in the Windows GUI).
Implementer	Name of the custom Implementing class.

Calculated Customer Fields is an advanced topic of the Rules Engine, and will not be required in most customer implementations of ActiveControl. The core Rules Engine capabilities should fulfil the vast majority of customer process / workflow requirements.

5.2.6.7. Application Tables

Various application tables are introduced with the Rules Engine. These tables do NOT need to be maintained by customers. This section is purely informational, and to hopefully avoid customers from updating the tables.

/BTI/TE_RE_STEPT

Table /BTI/TE_RE_STEPT is an application table used by the Rules Engine. It is pre-configured with the standard classes used by the Rules Engine to perform the various functionality, and should not need to be maintained by the majority of customers. The only exception to this is Custom Node Types, in this instance the custom step type and implementing class must be added to /BTI/TE_RE_STEPT.

/BTI/TE_RE_CONSU

Table /BTI/TE_RE_CONSU is an application data table used by the Rules Engine. It is pre-configured with the available Consumers, which can then be enabled/disabled via /BTI/TE_RE_CONSE configuration table.

/BTI/TE_RE_RUNS

Table /BTI/TE_RE_RUNS is an application data table where data associated with running the Test Harness is stored.

/BTI/TE_RE_GTYPE

Table /BTI/TE_RE_GTYPE is an application table where the possible Approval types are defined. These are available for selection within configuration table /BTI/TE_RE_APPRV.

5.3. Skipping

Please note that the Skipping functionality detailed in this section is superseded by the Rules Engine Skipping consumer.

It is possible to configure **Skipping Rules** so that Transport Forms bypass certain control points in the ActiveControl path.

This can either be done using the Rules Engine Skipping consumer (from ActiveControl 8.0 onwards) detailed earlier in this Administration Guide, or using the legacy Skipping capability described in this section of the Administration Guide.

These rules can be created based on the following fields.

- Transport form group
- Transport form type
- Business Task group
- Business Task type
- Business Task priority
- Any custom fields
- Project Phases

Example scenarios of where Skipping might be useful include where you do not require Merge TOC transports to be approved through every step of your Project landscape process, or similarly, where you do not want Maintenance changes to skip certain control points because they do not need the same level of approval as other changes being delivered.

Configuration of Control Point Skipping

Skipping is setup via configuration table /BTI/TE_SKIPCP in the Domain Controller, by adding rows for the relevant task/form fields that you want to skip CPs for along with the targets/locations.

Field	Description
CLASS	Either REQUEST or TASK or METADATA
FIELD	Either GROUPID or TYPEID or PRIORITY or [Custom Field Number] or PHASEID
ID	If skipping on Group, should be long serial number from /BTI/TE_GROUPS If skipping on Type, should be long serial number from /BTI/TE_TYPE If skipping on Priority, should be 1,2,3,4

	If skipping on Custom Field, should be the custom field value.
TARGET	Target ID from within ActiveControl Windows GUI.
LOCATION	(I = Inbox , Q = Import Queue , T = Test Queue , O = Outbox)
RISKGROUP	(optional) Only required for the (rare) scenario where you want objects defined in a certain Risk Group to stop or otherwise skip an approval point.
SKIP	“Skip”, or “Do not skip” or “Delete from branch”, depending on the requirement.
CUSTOM_FIELD	Check this box if defining skipping rules based on custom fields. If doing so, the custom field number should be entered in [FIELD]
RISKLEVEL	Again, only required for the (rare) scenario where you want objects defined in a certain Risk Group to stop or otherwise skip an approval point.

Further information – as well as configuration screenshots – of ActiveControl skipping is available in this online FAQ.

Information on Project Phase based skipping is available in this online FAQ

Information on Critical Objects based skipping is available in this online FAQ

5.4. Merge

ActiveControl offers support for branched SAP development streams, such as when a Production Support development stream is copied to a project development stream to allow project changes to be made in parallel to and independent of continuing Production Support changes.

The problem to be overcome with branched development systems is ensuring that Production Support changes are applied completely, correctly and at the right time to the project development stream. ActiveControl provides various functionality to help manage and control branched development streams as an integrated whole by:

1. Tracking when changes must be applied to the project development system.
2. Maximising productivity by analysing changes to determine those that can be automatically applied and those which must be manually merged because of conflicts with project work already undertaken.
3. Providing a reconciliation capability to give confidence that all changes have been applied completely and without regression.

It is important to note that whilst the methodology for managing branched development systems may result in production support transport requests being imported into the project development system, such transport requests are not imported into the downstream systems of the project development stream, such as the project Quality Assurance system.

Instead, ActiveControl creates special transport requests called **Merge** requests (similar in nature to backup requests) in the project development system and it is these requests that are transported throughout the project development stream. This differentiation is made to reduce complexity and confusion caused by transports being source from multiple systems.

How does Merge work?

ActiveControl Merge works according to the following high-level process:

1) The original transport(s) are imported to the target development system.

2) A Merge transport (Transport of Copies) is created (not imported) in the target development system.

3) All of the objects in the original transports are put into the merge transport.

4) That new merge transport is then released and moved down the target landscape.

When should we Merge in the process?

Production support changes should only be applied to the project development stream after they have been imported into the Production system, otherwise there is a risk that the project will introduce changes that the business is either not ready for or do not want, when it goes live.

To this end, it is necessary to add the target for the project development system to the transport path for the Production Support development stream. The project development system must appear as a child of the Production system. Once changes are applied to the Production system, they will flow to the import queue of the project development system and will be automatically analysed to determine whether the changes can be applied automatically (as described further on the next page).

Conflict Analysis.

When production support changes reach the import queue of the project development system, ActiveControl can analyse the changes to determine if there are any conflicts with changes already made in the project development system. This is done using the ShiftLeft: Conflict Analysis (0005) analysis check

For performance reasons, ActiveControl only checks whether the changes conflict with changes already made in the target SAP system (that is, the project development system) if the target's import option "Check whether the same content has been changed in this system before importing" is enabled. When this option is enabled, you must also specify the client in which client dependent changes are analysed. Specify the client in which you make client dependent configuration changes.

If the changes are found to conflict, then unfortunately the production support changes must be manually merged with the changes already made in the project development system. Such transport requests appear in the import queue with a blue icon to indicate that they should be manually applied.
Changes that do not conflict can be imported automatically.

Importing changes that do not conflict

Transport requests that do not conflict may be imported automatically, by selecting them in the project development system's import queue and clicking the Import toolbar button.

If the target option "Create a merge transport request in this system after importing changes" is enabled, ActiveControl will create a special transport request, called a merge request, in the project development system after the changes of the import run have been imported. The merge request contains all the items that were on the production support transport requests and is used to transport the same changes to the downstream systems of the project development stream.

ActiveControl automatically associates the merge request with the same tasks (that is, business issues and requirements) as the production support changes. This supports later reconciliation that those business issues and requirements have been applied to the project development stream before the project goes live.

Indicating that changes have been manually applied

If ActiveControl determines that production support changes conflict with changes already made in the project development stream, then those changes should be applied manually, merging the production support changes with the project changes.

Once this activity is complete, the responsible team member must 'inform' ActiveControl that the changes have been manually applied and on which transport request of the project development system the changes were applied (also referred to as a merge request, however this merge request is not created automatically by ActiveControl). This is done by selecting the applicable transport requests in the project development system's import queue and clicking on the **Mark as Applied Manually** option via the Actions toolbar in the Windows GUI.

By performing this Mark as Manually Applied action, ActiveControl will automatically associate the manually dual maintained transport request with the same Business Task as the original source transport.

Transporting the Merge requests

When demanded by the project, merge transport requests (either manually or automatically created)

should be transported to the downstream systems of the project development stream (such as QA), most likely along with other changes made by the project. No special action is required to do this, other than to approve the Transport Forms for each of the merge requests – as must be done for all transport requests being transported via ActiveControl.

Detailed information on how to configure Merge can be found in this online FAQ article

5.5. Inline Conflict Analysis

ActiveControl includes an 'Inline Conflict' capability to notify Transport creators – at the point of assigning their object change to a transport – if the change conflicts with another change already made to the same content in another development system. When a conflict is detected, the conflicting transports are displayed and the team member is given the opportunity not to save their changes.

When this Inline Conflict Analysis is performed, ActiveControl connects from the current SAP development system to the ActiveControl domain controller, which in turn connects to the other configured SAP development systems to check for conflicting changes. A target SAP system is deemed to be a development system if the target attribute "Show transport requests created in this SAP system within ActiveControl" is enabled.

Inline Conflict Analysis is switched on for the relevant Development systems within table **/BTI/TE_ACTIVE** in the ActiveControl domain controller.

In order to restrict the systems that ActiveControl checks during the analysis, the relevant entries need to be maintained in table **/BTI/TE_INLINE**. For example, this could be used to check from the ECC project development system (field SYSID) to the ECC BAU development system only (field CHECK_SYSID). The INACTIVE flag allows specific systems checks to be switched off if required.

In addition to the /BTI/TE_ACTIVE and /BTI/TE_INLINE table configuration, the client to check when connecting to the relevant development system must be specified in the Windows GUI target configuration Import Options tab in field "Before Importing, check whether the same content has been changed in this SAP client". This must be done in the target system that is being connected to for the conflict analysis.

Further information on Inline Conflict Analysis – including screenshot – is available in this online FAQ.

5.6. Inline Risk Analysis

ActiveControl includes a Risk Guard analysis to warn users when doing an Approval or Test signoff in

the Windows GUI or Web UI if the transports contain ‘risky’ objects. This functionality also exists in the SAP GUI in the form of an **Inline Risk Analysis** that can be presented to users at the point they are adding objects to transports. This Inline Risk functionality has several main use cases:

1) to warn Developers if the objects they are changing have been deemed risky, earlier in the process than the Approval/Test sign-off process.
2) to facilitate a whitelisting/blacklisting scenario – whereby certain objects cannot be touched in certain Development systems.
3) to facilitate transport splitting scenario – where certain objects cannot be mixed in the same transport.

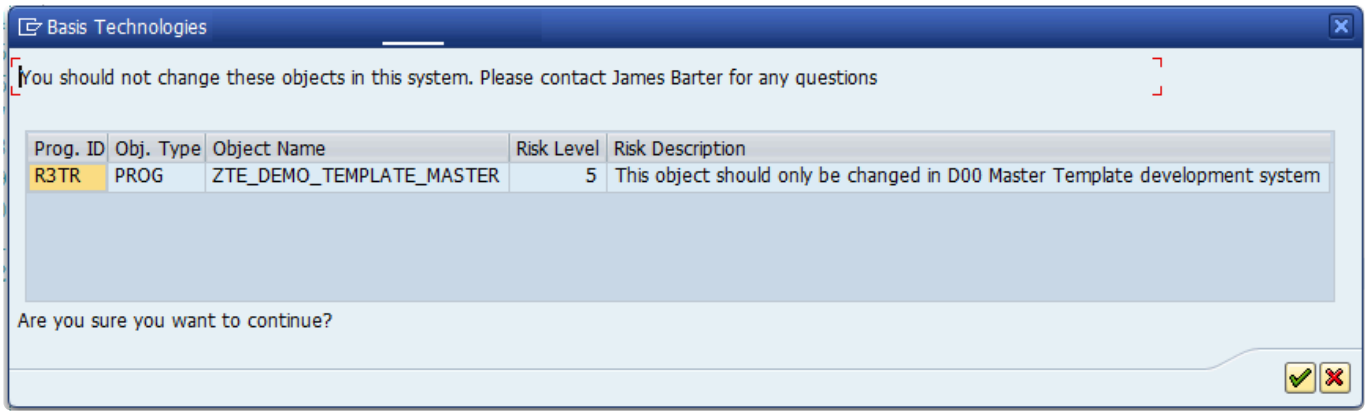


Figure: Example of Inline Risk screen presented to Developers.

Inline Risk

To use Inline Risk Analysis in its simplest form of warning Developers/Functional teams that the objects they are changing have been deemed risky objects, the following configuration is required in the ActiveControl Domain Controller. Most of the configuration steps are similar to that used by the ShiftLeft: Risk Guard analyser.

Step	Description
1	Risk Group should be configured in **/BTI/TE_RISKG**.
2	The set of objects should be added to **/BTI/TE_RISKGOB**.
3	Each Development system is then configured for the appropriate risk object groups along with a risk level that if exceeded with trigger the error/warning, in table **/BTI/TE_RISK_INL**. **System ID**: Enter the SID of the development system for which you are switching on Inline Risk Analysis **Risk Group**: Enter the Risk Group (from /BTI/TE_RISKG) that contains the risky objects that you want to highlight to the developer **Risk Level**: Enter the minimum risk level that you want to report on **Error Mode**: Choose here whether you want to present a Warning to the developer (allowing them to continue) or an Error (which will stop them from adding the object)

Object Allowed Lists / Blocked Lists

Within ActiveControl it is possible to setup object allowed lists / blocked lists to support Template protection, where certain objects can only be changed in a Master Template development (and certain objects cannot).

ActiveControl can be used to manage a Master Template / Child Developments based SAP landscape, and help ensure the Template system is protected.

Template Protection takes the form of object blocked lists – to prevent objects from being created in either the master template or child development systems. When an object contained within the blocked list is then added to a transport, the developer will receive message when saving to a transport. Depending on the configuration, this message can be either an error or a warning.

The Template Protection capability runs in the SAP GUI using the existing Inline Risk functionality. Please refer to online FAQ for more details of how to configure this topic.

Object Mixing (Transport Splitting)

Within ActiveControl it is possible to force certain objects to be split into separate transports and by the same token prevent certain objects from being mixed within the same transport. This functionality uses the standard Risk Guard functionality. Objects are defined in Risk Groups, and these objects in these Risk Groups are then prevented from being mixed with objects in other Risk Groups.

Again, the functionality runs in the SAP GUI using the existing Inline Risk functionality. The message, “The object you are adding cannot be mixed with other object(s) that are already on this transport.” will be presented when the Developer tries to mix objects from different Risk Groups that are not supposed to be mixed.

Please refer to online FAQ for more details of how to configure this topic.

5.7. Backout

Sometimes, despite best intentions, unwanted or adverse changes might be applied and you need a way to quickly and safely restore the affected SAP system to its pre-import state. ActiveControl Backout supports this requirement through the creation of **Backup** requests.

A backup request is a special transport request that ActiveControl can be configured to create in each client of the target SAP system before a selection of transport requests is imported. The purpose of a backup request is to export (and thereby save) an image of the content that is about to be changed by the import process, which can then be applied to revert out the changes in the event of an issue.

Each attempt to import a selection of transport requests is logged and is referred to as an import run. ActiveControl associates each backup request with its corresponding import run. If necessary, it is possible for an administrator to back out the changes applied by an import run by clicking on the Backout Import Run toolbar button in the History pane when displaying the target SAP system's import queue. When this is done, ActiveControl restores the pre-import state of the SAP system by simply importing the backup request back into the SAP client from which it was exported. If the target SAP system is configured with multiple clients, then the backup request created for each client is imported into each of the corresponding clients.

Creating backup requests slows the import process and so ActiveControl will only create backup requests for a target SAP system if the import option "Automatically create backup transport requests" is enabled. In addition, the following technical requirements must be observed:

1. A virtual system called **BAK** must be defined in TMS. This virtual system is necessary because backup requests are transport requests of type "transport of copies" and the virtual system BAK is specified as the target system in the transport request header (as displayed in transaction SE10).

1. The logon details specified on the target system's RFC destination (see RFC Destinations) must be valid for each client specified in the target's properties. This is required so that ActiveControl can log into the correct client when creating and applying backup requests, so that client dependent content can be correctly backed up and restored.

1. If using Backout, you also need to set the STMS parameter 'tadirdeletions' = "True" in each of the systems where Backout transports will be created.

1. Import runs must be backed out in reverse chronological order in order to guarantee the integrity of the SAP system. Of course, changes can only be backed out if backup requests were created when the import run was applied.

✻ Additional information on how to configure ActiveControl Backout – along with some configuration screenshots – is available in this online FAQ.

! ActiveControl Backout is currently only intended for use on ECC-type SAP systems. It will not work fully on BW or Java type systems.

5.8. Release Orchestration

ActiveControl Orchestration provides a seamless and automated approach to Cross System Dependencies and Manual Step scenarios.

In earlier versions of ActiveControl, a System Error would have been generated in the event of an unmet dependency or incomplete pre-import Manual Step when that Transport reached the Import Queue and an automated import ran. This resulted in Basis Team or Administrators having to get involved, which

resulted in delays as well as a rather inefficient manual intervention.

With Orchestration, a Transport will sit in ‘Pending Import’ status where there is an unmet transport dependency or incomplete pre-import Manual Step. As soon as the pre-requisite transport is imported, or the Manual Step is completed, then the pending transport will be automatically imported. This happens immediately, not when the next scheduled import runs.

Configuration Steps

Orchestration is switched on via the ‘Orchestrated’ option on the [Import Options] tab of a Target. Basis Technologies would generally recommend switching it on for all configured Targets.

5.9. Partial Testing

In earlier versions of ActiveControl, it was only been possible to sign off testing in Test Queues at a Business Task level. The guiding principle behind this has always been that testing is being signed off on an entire change, and so all Transport Forms relating to the Business Task should be moved at the one time.

There are scenarios where customers want to move individual transports forward from Test Queues, (for example when unit testing of one transport is complete, and the Developer wants to move it into subsequent test system for further testing in advance of the other transports relating to the same change being developed/unit tested).

As such, a **Partial Testing** capability is now available in ActiveControl to allow individual transports to be signed off within Test Queues. This can be performed in both the Windows GUI and in the Web UI. The option must be switched on explicitly for each of the target development system.

Configuration Steps

To enable transport level testing signoff, tick the ‘Allow test result entry for my transports on this target’ checkbox on the target.

> If the new configuration option is left unchecked, then Test Queue sign-offs will operate on Business Task level. Only users assigned to the Business Task will be able to do a “Save and Approve” in the Test Queue.

> Partial testing can only be done on transports that you own, not on other people’s transports. It can also be done across Business Tasks, ie by selecting Transport Forms on different business tasks.

5.10. Automated Transport Naming

Convention

ActiveControl can be used to automatically populate the Transport Description in an SAP transport at the point the Developer is saving a Transport Form in the SAP GUI of the ABAP Development system. This can help with the consistency of descriptions within SAP, and also the auditability of transports later.

The following information can be used to build an automated Transport naming convention:

1) Standard/Custom fields from a Business Task (or a specified substring from that value)
2) Standard/Custom fields from a Transport Form (or a specified substring from that value)
3) Hardcoded Text strings
4) The Transport Description that the Developer writes prior to saving the Transport Form.

Configuration of Transport Description naming automation

To setup this functionality, you need to configure the required Transport Description naming convention in **/BTI/TE_TR_DESC** table:

Field	Description
PATH	The path that this configuration is valid for.
SEQUENCE	The sequential number to put the records in the correct order.
TYPE	This is the type of the record. T = Task Field, F = Form field, X = constant/delimiter, D = current description of transport.
CUSTOM_FIELDNUM	If you want to use the contents of a custom field, this is the number of it.
TE_FIELD	This is the fieldname of the Task or TF. Note, for Group and Type fields, the field names are actually GROUP_NAME and TYPE_NAME. If you use GROUPID, it will actually use the internal ID of the group. Task Reference = REFERENCE, not TASK_REFERENCE Task Subject = CAPTION, not SUBJECT Project = PROJECT_NAME, not PROJECT.
JUSTIFICATION	Use this if you only want a sub-set of what is in the field. L = left x characters, R = right x characters, M = Middle of value, D = characters after a delimiter.
LENGTH	The number of characters from the field you want to use (notes this is used in conjunction with the JUSTIFICATION above). If you leave this blank, the whole value is used.
MID_START	If you select M for the JUSTIFICATION, this is the starting point in the value.
TEXT	If this is a constant/delimiter, then the text value should be entered here. Also, if you use the JUSTIFICATION = D, this is the delimiter in the text that the api looks for.

Notes:
i) you should enter the **&** character before a "*" or "!" if you want to use either of those two characters in your text element. ie* **&** or **&!**
ii) you should enter **& &** if you want to have a leading or trailing space in your text element

iii) the transport naming automation only works when the Transport Form is created, not on later updates to existing Transport Form. So if you create the Transport Form and then later update fields on the Transport Form or Business Task, the Transport Description will NOT get automatically updated. In that scenario, you will need to manually amend the Transport Description in SAP GUI or in the Transport Form (if the transport is still modifiable).

> Further information on how to configure Automated Transport Naming Convention – plus a screenshot example of some configuration – is available in this online FAQ.

5.11. Automated 'Exception-based' Approvals

ActiveControl offers an exception-based auto approval capability, to allow customers to have certain (configurable) analysers running automatically in a particular Inbox or Outbox, and only stop the transports for a manual approval if those analysers identify issues.

Three outcomes are possible with the analyser results:

- BLOCK: if a particular analyser highlights an issue, ActiveControl will stop the Transport Form for a manual approval by the configured approver(s)
- WARN: if a particular analyser highlights a issue, ActiveControl will notify the configured approver(s) but still move the Transport Form forward to the next location.
- IGNORE: if a particular analyser highlights an issue, ActiveControl will move the Transport Form forward to the next location and not notify the configured approver(s).

Configuration Steps

Detailed information on how to configure automated exception-based approvals is detailed in this online FAQ.

Notes

(i) Auto-Approval runs at Transport Form level. This is important as it means you should probably set the General: Dependency Check to BLOCK to ensure that individual transports do not move forward.

(ii) Auto-Approval is only designed to work on Inboxes and Outboxes, NOT on Test Queues

(iii) Priorisation of BLOCK / WARNING / IGNORE: If a Transport has a BLOCK on one analyser and a WARNING or IGNORE on the other analyser, the BLOCK will always take priority and the transport will stop for a manual approval.

5.12. Additional Approvals for specific

Objects

It is possible to force transports containing specific critical standard or custom objects to stop for an additional approval in the ActiveControl workflow. This feature sits on top of the existing Skipping and Risk Guard functionalities.

Configuration Steps

Step	Details
1	Add the following four entries to **/BTI/TE_EXITC** table: /BTI/TE_EXIT_SAMPLE_0740 /BTI/TE_EXIT_SAMPLE_0500 /BTI/TE_EXIT_SAMPLE_0505
2	Create Risk Groups in table **/BTI/TE_RISKG** for each grouping of approvals that you want. For example: BT9001 – Security Objects BT9002 – Basis Objects BT9003 – Finance Objects
3	Create a new Risk Level for these objects in **/BTI/TE_RISKL**.
4	Add entries in table **/BTI/TE_RISKGOB** for each object you want to stop for additional approval. Use RISK LEVEL defined in previous step.
5	Add a new Control Point (ie Inbox or Outbox) where you want the additional approval to occur if the transport contains the objects defined in previous step.
6	Add entries in table **/BTI/TE_RISKUSER** for who you want to be able to approve each Risk Group.
7	Add the Approvers in the ActiveControl Windows GUI configuration against the required target/ control point.
8	Create new Skipping Rules in **/BTI/TE_SKIPCP**. (i) Add Skipping Rules for all Transport Form ‘Types’ to skip new Control Point. (ii) Add Skipping Rules for all Transport Form ‘Types’ to NOT SKIP when they contain the Risk Groups created above – by adding in RISK_LEVEL that you defined in earlier step.

5.13. Document Categories

It is possible to add documents into ActiveControl via the SAP GUI, Windows GUI and Web UI. Documents can be added at Business Task or Transport Form level, and also in Test Queues. These documents can be uploaded directly (in which case a copy of the document is stored in the Domain Controller, or alternatively a URL link can be added to reference where the document is stored elsewhere.

As part of the process for adding documents (either uploading or linking), a **Document Category** can be specified – for example Functional Spec, Unit Test Evidence, UAT Test Script, Email, etc). These Document Categories can be used for Reporting, and also by ShiftLeft: Check Documentation (0051) to perform an automated check for the presence of a uploaded document at a particular Inbox, Test Queue, Outbox control point in the workflow.

Configuring Document Categories

Document categories are defined in table **/BTI/TE_ATT_CAT** in the Domain Controller. These Document Categories can be restricted so that they can only be used for Business Tasks, Transport Forms or Test Results. This is controlled via the CLASS field. The valid entries for this field are:

- TASK
- REQUEST
- TESTRESULT

If the CLASS field is left blank, the document category is valid for all object types.

Default Document Categories

It is possible to set default document categories so that a configured Category is pre-populated during the process of uploading a Document in both the ActiveControl Web UI and Windows GUI; this is useful for customers that always upload the same types of documents or alternatively do not want to benefit from the Document Category functionality, but do want to upload or link documents.

Defaults can be specified (for Task, Request and Test Queue level) via the DEFAULT_CAT field in /BTI/TE_ATT_CAT table, by entering an “X” against the row that you want to be defaulted.

> Further information on Document Categories – along with a screenshot example of configuration – is detailed in this online FAQ.

5.14. Documentation Links

It is possible to add **Documentation Links** via the web UI and Windows GUI Help menus to point at standard BTI or customer specific documentation relating to ActiveControl (or anything else). These document links will then be visible via the Help menu in both the ActiveControl Windows GUI and the Web UI.

Document links can be maintained in the Domain Controller via table **/BTI/TE_HLP_LINK**.

5.15. Manual Activities and Non-Transport Deployments

It is possible to manage **Manual Activities** and **Non-SAP changes** within ActiveControl. These can be used to manage non-transportable changes and activities and still enforce an approval and audit process. These types of changes will still use the concept of transports and paths and so will follow the existing process and logic in ActiveControl. They can either be sent down an existing SAP path or a new path created specifically for each non-SAP system.

When a Transport Form is created there are 3 options:

1. Existing Transport Request
2. Manual Activity (SAP or non-SAP system) – For example, creation of a SAP number range or RFC destination or for non-SAP systems, this could be the installation of a printer or any other manual action.
3. Non-Transport Deployment (SAP or non-SAP system) – For example, this could be used to deploy an script or executable to any system

For the Manual Activity and Non-Transport Deployment types, a description will need to be entered and a dummy SAP transport will be created. This transport is just a holder for the change to allow it to be sent down any transport path which can either be an existing SAP systems path or a path that’s created for non-SAP systems as well.

Manual Activity type changes automatically create a manual step in the Transport Form. This can be updated to choose the systems the activity is relevant for and then be marked as complete for each system rather than performing an import. Once completed the import queue will be automatically skipped for the relevant system.

Non-Transport Deployment type changes are handled in the same way as normal SAP transports but instead of a transport import a custom import method can be called via a user exit to perform the deployment. The required deployment can be attached to the Transport Form and used by the deployment method to execute the change.

In order to support these new change types a dummy transport will be created in a nominated SAP system, as configured on the [Other] tab of the Windows GUI confguration screen. It is recommended to use the Domain Controller system for this but not essential. A SAP transport target can also be specified if required but using “SAP Default” will allow SAP to determine this automatically.

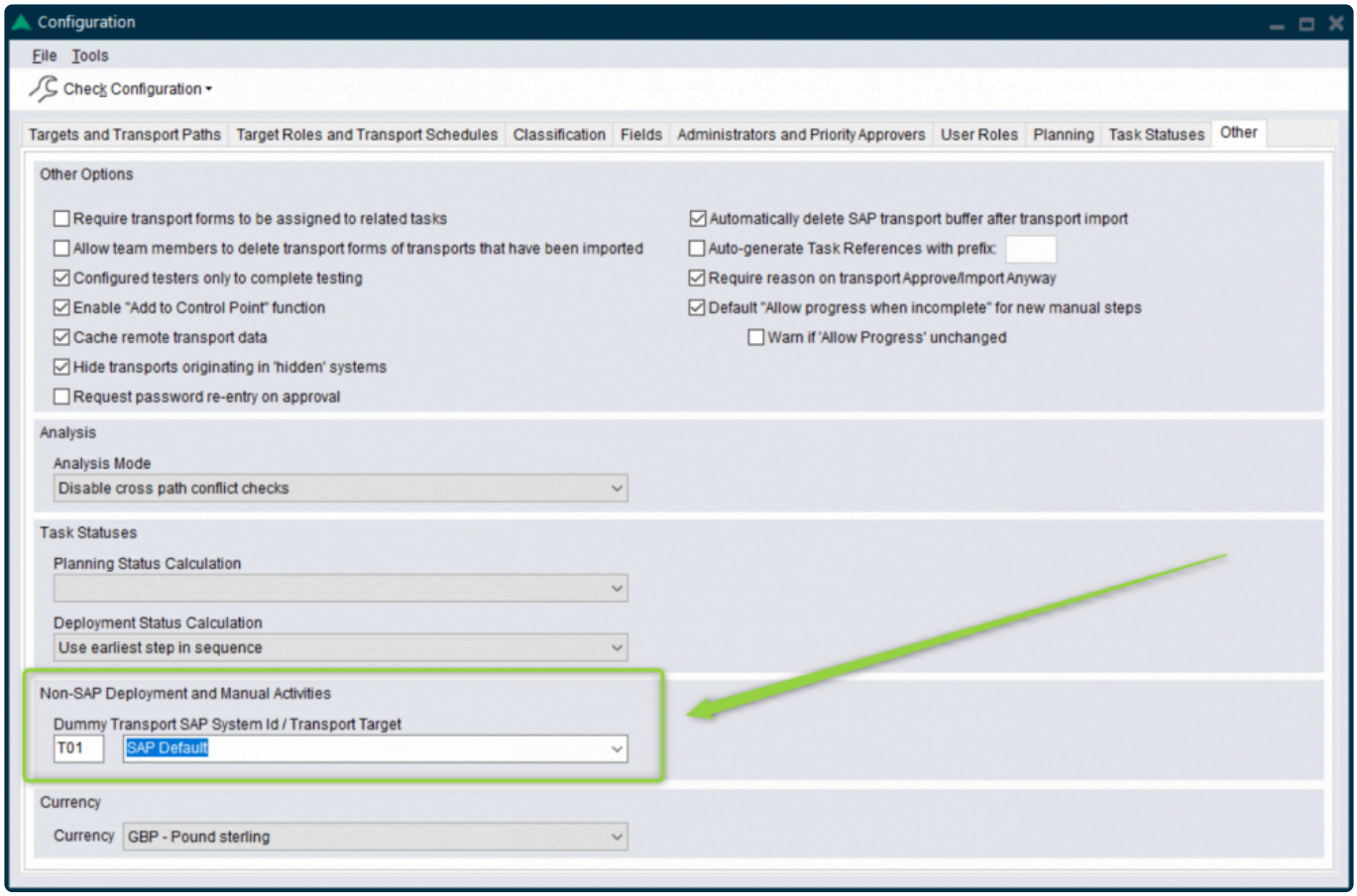


Figure: System definition for Manual Activities and Non-SAP Deployments

The transports do not need to be released as there will be no import process for them so if required they could be deleted afterwards as part of a clean-up. In addition, in order for the ActiveControl caching to function correctly the SAP system configured here also needs to be created as a Target system and marked as a source system for transports. (This doesn't need to be added to any transport paths):

> ✻ Note that using this functionality might require additional authorisations, depending on what access the user already has in the system in which the dummy transport is being created. Authorization object S_TRANSPRT with activity 01 (Create) and S_SYS_RWBO may be required.

5.16. External Transports

External third party transports can be uploaded into ActiveControl via the Windows GUI or the Web UI. This can be done by any user, it is not solely a Administrator or Basis function within the tool.

Adding External Transports in the Windows GUI

External Transports are added in the Windows GUI via the **External Transport Request** option on the [New Transport Form] screen. From this screen, the cofile and data file for the external transport is

uploaded to either the Domain Controller or a specific local development system.

When you first add an external transport, an entry is created to link the SAP transport prefix (SID) to the Domain Controller or the local Development system that you selected. From that point onwards, the system that you chose will always be where external transports of that SID are uploaded. (this information is stored in /BTI/TE_TARG_EXT table in the Domain Controller). A Transport Form can then be populated for the transport, and moved through the approvals workflow in the usual manner.

Adding External Transports in the Web UI

External Transports are added in the Web GUI via the **Create Transport Form** option on the main home screen. Thereafter the same process applies for adding your External Transport as described above for the Windows GUI.

Adding External Transports in the SAP GUI

For any ActiveControl users that prefer to upload external transports via the SAP GUI instead of the Windows GUI or Web UI, this can be done using program **/BTI/TE_RUPLOAD_TR_REQUEST** in the Domain Controller.

> Unlike earlier versions of ActiveControl, there is now no pre-requisite requirement to manually create RFCs or ActiveControl Targets to upload external transports into ActiveControl.

5.17. Delegation

Delegation enables Inbox and Outbox approvers to delegate their approvals within ActiveControl. This functionality is possible within both the Windows GUI and the Web UI.

It is also possible for authorised users to ‘Force Delegation’ to set delegations for other users if they are off sick or on other unexpected leave. To be able to force delegations, a user needs to have authorization object Y_TECONF with value FORCEDELEGATE.

> It is not possible to define Delegates on a Test Queue. To ‘delegate’ Testing, you should simply assign other Testers to the Business Task.

> When you delegate somebody, they do not need to be added as an Approver in

> Windows GUI configuration. However they will need to have some sort of ActiveControl role assignment in the Domain Controller, to be able to access ActiveControl to perform the approval.

5.18. Non-ABAP / Java transports and CTS+

In order to deploy non-ABAP / Java transports, ActiveControl utilises CTS+ so this must be setup and configured beforehand.

Configuring ActiveControl to manage non-ABAP systems

For each non-ABAP / Java system to be managed by ActiveControl, a new target and RFC destination must be setup as described in the 'Configuring ActiveControl' section of this Administration Guide. In this case there are some small differences in the configuration.

For example, you might have the following non-ABAP / Java systems:

- JD1 – Development
- JT1 – Test
- JP1 – Production

The CTS+ controller system in this example is CT1

RFC Destinations

An RFC destination is required for each system but for non-ABAP / Java systems this must connect to the relevant CTS+ controller system. In the example above these would be setup as follows:

- ACTIVECONTROL JD1 – connects to CTS+ controller system CT1
- ACTIVECONTROL JT1 – connects to CTS+ controller system CT1
- ACTIVECONTROL JP1 – connects to CTS+ controller system CT1

Target Systems

Each non-ABAP / Java system needs to be created as a Target System in the same way as ABAP systems, but with some subtle configuration differences:

Clients	For the target client specify the client of the CTS+ controller system.
Ignore System Id during import (CTS+ only)	For CTS+ systems the SAP system ID is different to the system that is the CTS+ domain controller. In this case the flag should be set to inform ActiveControl that the target is a CTS+ system.

Transport Paths

When creating the transport path the "Valid Source Clients" should be setup for the relevant CTS+

development system and referencing the client for the CTS+ controller. E.g. JD1:100

Once the above setup has been done, ActiveControl will then automatically use CTS+ to process and import the non-ABAP / Java transports.

> Further information – including configuration screenshots – is available in this online FAQ.

5.19. Transport of Copies (Testing TOCs)

ActiveControl includes the ability to create Transport of Copies (TOC) of existing transport(s) to push to their test systems. This avoids the need to release the original transport until testing is complete, and can significantly reduce the number of transports moving beyond your test system to Production.

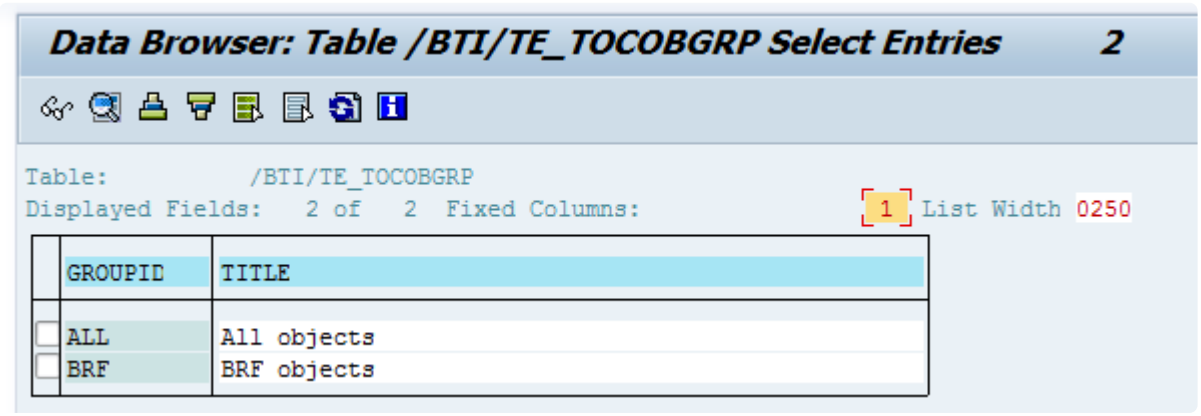
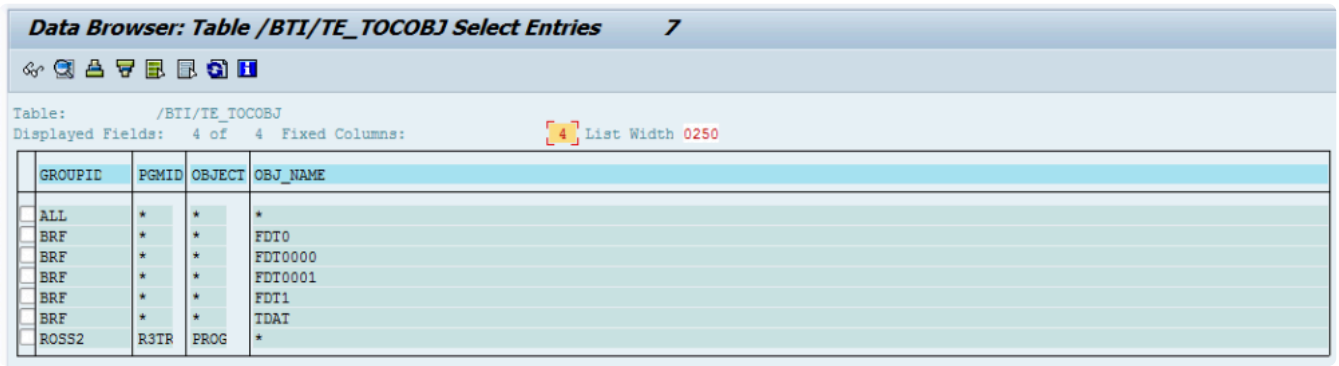
To enable the TOC functionality within ActiveControl requires the following configuration in the Domain Controller:

Step	Details
1.	Use SAP transaction SE16 within table **/BTI/TE_TOCONFIG** – add a configuration entry depending on your requirements. The configuration flags are as follows: • *CREATELEVEL* allows you to dictate at what level the TOCs are created at. E.g. if you have two Transport Forms that are part of two different Projects, configuring CREATELEVEL = P will result in two separate TOCs being created. • *SELECTLEVEL* • *REMOVEONOVERTAKE* • *DELETE_EMPTY* will remove any SAP Tasks which contain no objects. • *AUTO_RELEASE* will automatically release any unreleased SAP Tasks that are included in the Transport(s) that you are creating a TOC for. • *OPEN_NEW_TASKS* will automatically create new SAP Tasks assigned to the same consultant as those included in the Transport(s) that you are creating a TOC for.
2.	(optional) Use SAP transaction SE16 within table **/BTI/TE_EXITC** – add an entry for user exit 0855 Note this is an optional step that will auto-create/populate Transport Form for your TOC. Without the user exit switched on, the TOC will be created, but no Transport Form created for it.

> Further information – including configuration screenshot – on setting up ActiveControl TOCs can be found in this online FAQ.

TOC object splitting

As part of the ActiveControl 'Transport of Copies' functionality, it is possible to split out the original transport into several separate TOCs based on the objects contained within the original transport. Setting up this object splitting is a two step process, as follows:

Step	Activity
1	Create **Groups** for each of the TOCs that you want to create by populating table **/BTI/TE_TOCOBGRP**  Data Browser: Table /BTI/TE_TOCOBGRP Select Entries 2 Table: /BTI/TE_TOCOBGRP Displayed Fields: 2 of 2 Fixed Columns: 1 List Width 0250 GROUPID TITLE ALL All objects BRF BRF objects In above screenshot, we intend to split out each transport into two TOCs, one for BRF objects, and one for everything else
2	Allocate what objects you want to allocate to each groups by populating table **/BTI/TE_TOCOBJ**  Data Browser: Table /BTI/TE_TOCOBJ Select Entries 7 Table: /BTI/TE_TOCOBJ Displayed Fields: 4 of 4 Fixed Columns: 4 List Width 0250 GROUPID PGMID OBJECT OBJ_NAME ALL * * * BRF * * FDT0 BRF * * FDT0000 BRF * * FDT0001 BRF * * FDT1 BRF * * TDAT ROSS2 R3TR PROG * In above screenshot, we intend to create one TOC for the various BRF related objects, and one for everything else

Note that wildcard * can be used when defining the objects for TOC object splitting, to reduce the configuration required.

5.20. Transport of Copies (Production TOCs)

As well as the functionality to push TOCs from Dev to QA, ActiveControl also includes the ability to

create Transport Of Copies (TOC) for deployment to Production. This can be used to reduce the number of transports being deployed to Production, and by extension, reduce the time it takes to deploy them.

Production TOCs are created in the ActiveControl Windows GUI. This is done by right-mouse clicking on the required transport(s) and selecting “New Transport of Copies” > “For Production”.

The new Production TOC behaves quite differently from the ActiveControl ‘Testing TOC’ capability. Key points of note:

(1) the TOC will be put in the same location as the original transports.
(2) the original transports are deleted from the location.
(3) there is no user exit, a Transport Form is created automatically, which (i) ignores the task selections made in the GUI (ii) assigns the tasks of the original transports (4) takes the ActiveControl [Group] and [Type] coming from /BTI/TE_TOCONFIG DEPLOYTYPEID and DEPLOYGROUPID.
(5) If there is a user exit, then creating and moving the TF will be handled by the exit.
(6) When a TOC Transport Form is deleted completely without being completed, the original transports are added back to the original location.

This functionality is intended only for Production deployments, and will only work if all the transports imported in QA but not in Production are merged into a single TOC. Side effects could happen if the TOC is created earlier in the path or some transports are excluded from the TOC (and also many of the ShiftLeft analysers will not work correctly)

Configuration Steps

To configure Production TOCs, two configurations are required:

1) a couple entries must be added in table /BTI/TE_TOCONFIG in the Domain Controller.

DEPLOYTYPEID: The “Type” to be used for the TOC Transport Form (from /BTI/TE_TYPE)
DEPLOYGROUPID: The “Group” to be allocated to to the TOC Transport Form (from /BTI/TE_GROUPS

2) the system in which the TOC will be created should be configured as a valid source system in the path (in the ActiveControl Windows GUI)

3) User creating the Production TOC in the Windows GUI needs to have CREATETOCDEPLOY authorisation activity.

5.21. Unit Test Automation

ActiveControl **Unit Test Automation** capability hooks into standard SAP functionality available within SAP Code Inspector, and can be used to:

1) run automatically at the point of transport release in the SAPGUI – ensuring that Developers are aware of unit testing issues.

2) run automatically as a ShiftLeft Analyser in the Web UI and Windows GUI, to ensure that Approvers also get visibility of non-adherent transports.

Configuring Unit Test Automation

Step	Configuration
1.	The unit test automation on transport release is switched on in table **/BTI/TE_ACTIVE** in the domain controller by adding the *Unit test on release* active function for the relevant development system.
2.	The SAP Code Inspector variant and user parameter also needs to be configured in the Domain Controller in backend table **/BTI/TE_ACTPARMS**.
3.	Switching on ShiftLeft: Unit Test Automation.

5.22. Diffuser / Object Linkaging

Several parts of ActiveControl process significant amounts of SAP object level data, and use another SAP add-on developed by Basis Technologies called **Diffuser** (previously known as Mass Data Runtime (MDR)) to enable ABAP programs and transactions to run in a fraction of their normal run-time without the need for additional hardware.

Diffuser is used to process the object linkaging which is used in in Deep Impact Analysis, Test Impact Radar and several other ActiveControl functionalities.

Setting up Diffuser for ActiveControl

Step	Setup required
1.	A Diffuser license key needs to be uploaded to the Development system using transaction /N/BTR/LICENSE.
2.	Define the following programs via /n/BTR/MDR in the Development systems: /BTI/TE_REP_MDR_OBJECT_DI_LINK /BTI/TE_REP_MDR_CR_TRAN_LINKS
3.	Create number range for /BTI/TE_OI via transaction SNRO.

Initial Object Linkaging

There are two main ways of setting up the object linkaging, and it really depends on what ActiveControl functionality you want to use it for.

1) Object Linkaging for use with Test Impact Radar

Use /BTI/TE_REP_MDR_OBJECT_DI_LINK in the Development system. This will typically run for days and use a lot of data. Function module flag should be left unticked. “Save links determined” should be ticked to avoid errors

2) Object Linkaging for use with Deep Impact Analysis

Use /BTI/TE_DI_LINKS_UPDATE_JOB in the Domain Controller, on every transport form in the relevant paths, a dev target for each path and * for the location. This will schedule /BTI/TE_REP_MDR_CR_TRAN_LINKS on each dev system, creating links for every transport with a TF.

> Please refer to this online FAQ for a lot more detailed information – and screenshots – on how to setup Object Linkaging for use with Deep Impact Analysis.

5.23. Test Impact Radar

ActiveControl includes a **Test Impact Radar** functionality that can be used to assess which customer test scripts are impacted by the delivery of a particular SAP change. This was a precursor to the fully automated regression testing capability of Basis Technologies more recent product Testimony.

Diffuser setup

Basis Technologies separate product Diffuser is a pre-requisite for using Test Impact Radar. Information on setting up Diffuser for use with ActiveControl is found here.

Test Script Upload

To use Test Impact Radar, you must upload your existing suite of test scripts into the ActiveControl domain controller using transaction **/n/BTI/TE_TSCRT_IMPORT**.
These test scripts files can be in either Microsoft Word or Excel format.

Field	Description
Local source directory	Specify the local directory in which the test scripts are stored. These should have been exported out of the test script repository (e.g. HP QC) via the bulk export function. Effectively you’ll end up with N number of Excel/Word files within this local directory.
Filename filter	Specify the filename filter to restrict which files in the local source directory are selected (e.g. *.xls or *.xlsx).
File type	Allows you to select whether the test script files are Excel or Word. Alternatively, just leave as “Auto detect
Remote	Since this is running on the Solution Manager system (Domain Controller controller) then there

Target System	are no SAP applications installed there (e.g. HR, FI, SD). Hence, specify a target system upon which the SAP transactions, programs, BAPI's, BSP applications and Web-dynpro's will be retrieved from.
Update existing scripts	Updates existing test scripts that have already been loaded. This may be required if test scripts have been changed in the external test management system (e.g. HP QC).
Output progress indicator	Outputs the progress
Test mode	Run the upload in test mode only. No database changes will be made. Used for testing purposes.
Output token list	Output the list of "tokens" that are found within the file (for testing purposes only).

Initial Linkage Creation

The initial linkage creation for Test Impact Radar is performed as follows:

Step	Activity
1.	Run /BTI/TE_REP_MDR_OBJECT_DI_LINK program in the Domain Controller. Function module flag should be left unticked. "Save links determined" should be ticked to avoid errors. This will typically run for several days and use a lot of data.

Ongoing Linkage Creation

The ongoing linkage creation for Test Impact Radar can be setup as follows:

Step	Activity
1.	Create a variant /BTI/TE_DI_LINKS_UPDATE_JOB program in the Domain Controller as follows Source System IDs = SID of the Development System. Newly Released Transports. Maximum number of retries = 30 Update reference date (to now) (plus a date and time from a couple weeks ago)
2.	Schedule above program/variant to run every 10 minutes.

Analysis Check setup

Test Impact Radar (0043) analysis check is used to highlight what top level SAP objects are included in the selected transport(s) for which it is run, and also to indicate what Test Scripts are impacted by the changes in the transports – i.e what test scripts should be executed at a minimum. This Analysis Check

needs to be switched on at each of the Control Points in your Transport Path that you wish to be told this information.

Please refer to (link this) section of this Administration Guide for details of how to setup the Test Impact Radar analysis check.

Test Script Management

After uploading test scripts, the Test Script Management console can be used to maintain existing scripts in your repository and also to upload additional individual scripts in the future. This is done via transaction **/BTI/TE_TSCRT_MANAG** (technical program: /BTI/TE_RTEST_SCRIPT_MANAGE)

The left hand pane outlines the scripts that have been uploaded. Double clicking on an individual script (on the folder icon) will bring up details of the script in the right hand pane. From this screen, it is possible to deactivate test scripts so they are not included in the Test Impact Assessment. It is also possible to add additional meta-data information about the script (for example the business process the script relates to), and also view the top level SAP objects impacted by the test script.

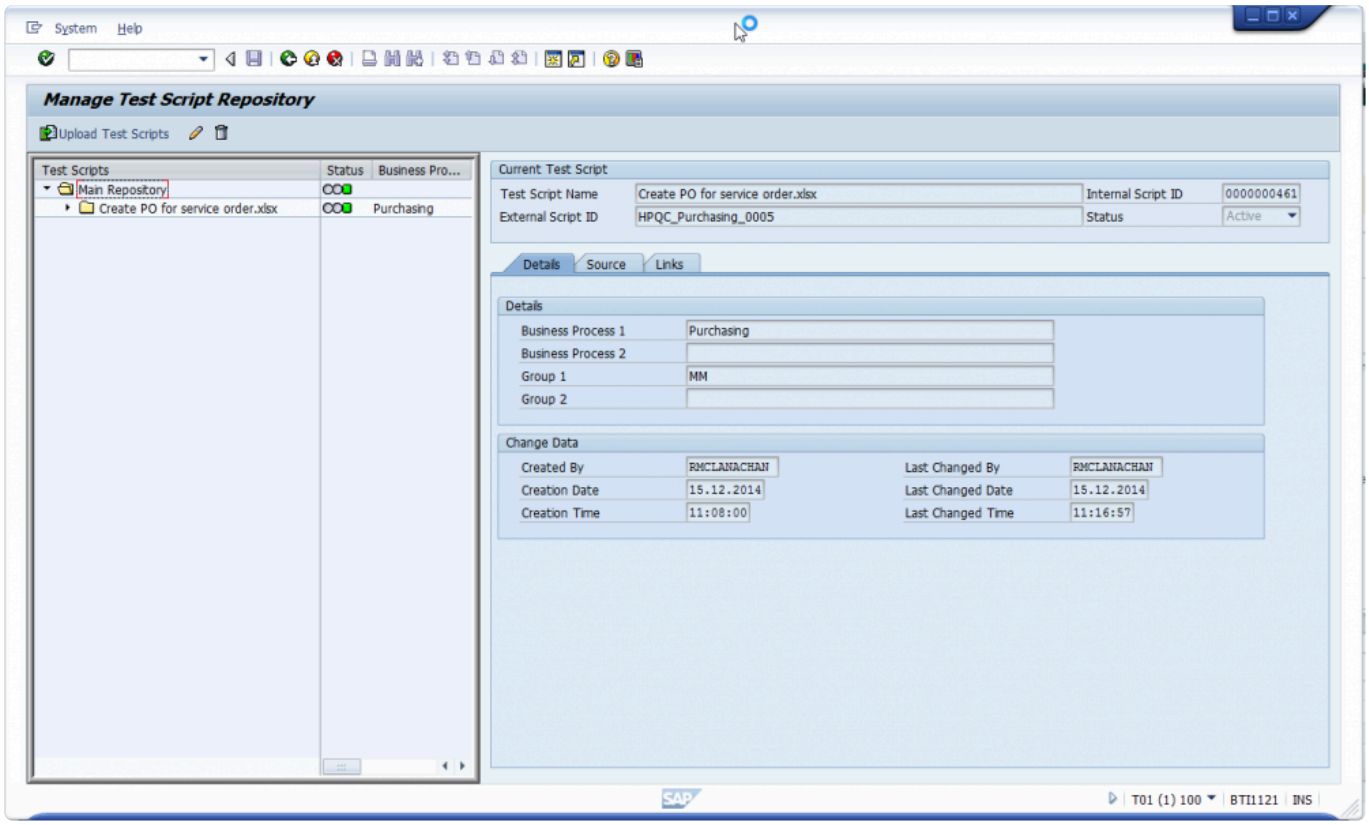


Figure: Test Script Management console.

5.24. Integration Framework

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 ActiveControl – by exposing several fully documented

API's and web-services that allow manipulation of ActiveControl objects by these systems.

In addition, as ActiveControl 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.

ActiveControl integrations have already been setup with existing Basis Technologies' customers on the following tools:

- ServiceNow
- JIRA
- Gitlab
- BMC Remedy
- Solution Manager
- Cherwell
- Footprints

5.24.1. Integration Scenarios

The standard integration scenario is to combine ActiveControl 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:



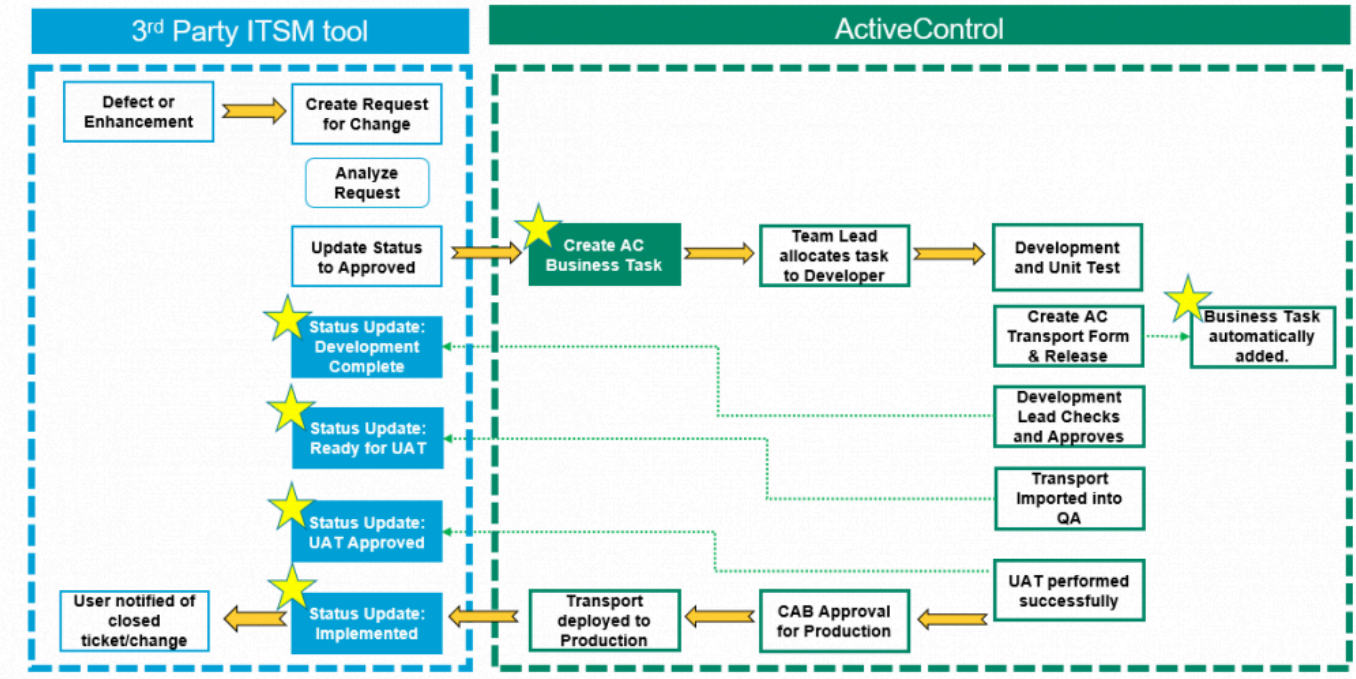
1. Change created in third party ITSM system
2. Change approved for development in ITSM system
3. Change interfaced to ActiveControl (inbound integration)
4. Change managed through ActiveControl for deployment to Test and Pre-Prod with updates sent to ITSM system to reflect progress (outbound integration)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

Note: This section of the Administration Guide will describe in detail 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.

5.24.2. Integration Process Flow

The 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:



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 ActiveControl, 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 ActiveControl representing the change. The ticket in the ITSM system and the Business Task are then tied together for the remainder of the process
2. The creation of the Business task in ActiveControl marks the start of the development process. The Business 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. ActiveControl 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. ActiveControl 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 ActiveControl and the ITSM system automatically updated.
7. CAB approval is sought and General/ShiftLeft analysers are run 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.

5.24.2.1. Inbound Integration Process

There are two inbound calls in the above scenario:

1. Creation of the Business Task in ActiveControl
2. Approval of Testing/Entry of test results once testing complete

Both of these calls would be web service calls to standard ActiveControl 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 ActiveControl for classification of the change.

5.24.2.2. Outbound Integration Process

The outbound calls from ActiveControl to the external ticketing system can all be based on the Deployment Status of a change within ActiveControl.

Integration scenarios based on ActiveControl status changes are delivered as standard with the Integration Engine and therefore require no development.

The steps to set up this type of status based integration are:

1. Complete base 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 user exits and how they are implemented, please refer to later section of this 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 ActiveControl 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 system external system the trigger program is to be run against	
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)

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.

Program Name: **/BTI/TE_INTEG_SEND**

Selection Option	Description
External System	The system external system that 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

4. The outcome of the send process is recorded for audit purposes. If successful, any updates configured are made to the ActiveControl 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.

Program Name: **/BTI/TE_RINTEG_AUDIT**

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

5.24.3. Outbound Integration

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.

5.24.3.1. Integration – Configuration Overview

The table below shows a list of database tables with descriptions that need to be maintained followed by a more in depth description of how to configure the tables.

Name	Description
/BTI/TE_INT_CLAS	Integration Object Class List
/BTI/TE_INT_PC	Process Codes
/BTI/TE_INT_PROC	Process Identifier
/BTI/TE_INT_CONV	Conversions
/BTI/TE_INT_USR	Notification Users
/BTI/TE_INT_MAPP	Mapping Table
/BTI/TE_INT_FLDE	Complex Mapping (user exits)
/BTI/TE_INT_SYST	Integration System List Table

5.24.3.2. Detailed Configuration Table Information

5.24.3.2.1. External System(s)

The 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.

/BTI/TE_INT_SYST – 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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
EXTSYS_ID	Single word identifier for external system. E.g. REMEDY or 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.

/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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
CLASSNAME	Held here is the class name where the bulk of the integration processing is done. 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.

5.24.3.2.2. 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.

/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.

5.24.3.2.3. Integration Points

The standard out of the box ActiveControl 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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
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.

5.24.3.2.4. 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 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 – 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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
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 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.

5.24.3.2.5. Notification Users

It is also 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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
EXTSYS_NAME	Full description of external system
USERID	SAP Logon ID of person that needs to be notified of failed integrations.

5.24.3.2.6. Mapping

An essential part of the Integration Framework is mapping relevant ActiveControl fields to the equivalent fields on any external ITSM 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 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 ActiveControl 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. An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
EXTSYS_NAME	Full description of external system
TEFIELDREF	This is the 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 ActiveControl 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 Custom field to be mapped.
DEFAULT_VAL	Defaulted Value to be mapped over to the integrated system field.

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.

	An example of this could be: 1 – Remedy 2 – Solution Manager 3 – Service Desk
EXITFIELDNAME	External field that this refers to
EXITFUNCNAME	The function module to be executed to perform this exit

5.24.4. Inbound Integration

For inbound integration scenarios ActiveControl 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

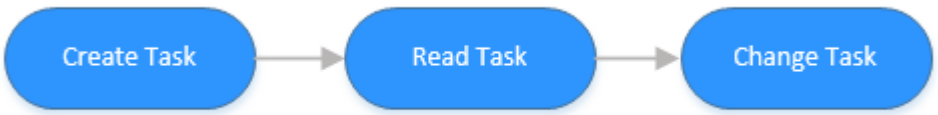
Each web service is detailed in the following sections.

5.24.4.1. Inbound Process Flow

The standard inbound integration process flows would be:

- **Create/Change a Business Task in ActiveControl**

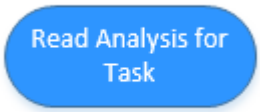
Creating or changing a Business Task requires simple calls to the appropriate web service. When changing a Business Task, the current field values should be read first to ensure changed data is not overwritten. The process flow should therefore be:



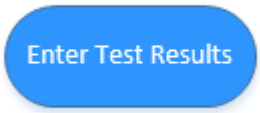
- **Approve a Business Task**
  When approving a Business Task it is important that the Task Analysis is completed to first to ensure that the approval can take place safely. The approval web service will not stop the approval if analysis results are ignored. The process flow for approving a Business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



- **The analysis results for a Business 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.



5.24.4.2. Web Services

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

The following documentation, which details every action available, has been generated from the WSDL above.

Port type _-BTI_-TE_TASK_WS

5.24.4.2.1. 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)

5.24.4.2.2. 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)

5.24.4.2.3. 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)

5.24.4.2.4. 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)

5.24.4.2.5. 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*)

5.24.4.2.6. 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)

5.24.4.2.7. 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)

5.24.4.2.8. 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

5.24.4.2.9. 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.

5.24.5. ServiceNow Integration

ActiveControl includes an out of the box solution add-on to ServiceNow, which was built using the ServiceNow Certified Integration Program.

This integration provides bi-directional data flows between ActiveControl and ServiceNow including:

- Integration triggers
- Data mapping
- Error handling and notifications

Please refer to the separate ServiceNow Integration Administration Guide for details of how to configure this Integration.

5.24.6. JIRA Integration

ActiveControl includes an out of the box JIRA integration that makes use of the standard Integration Framework.

This was added in 2017 in response to a customer that wanted their business users to continue doing approvals in JIRA, whilst the rest of the workflow (including technical approvals, automated analysis, automated deployment) would be done via ActiveControl.

Configuration Steps

The JIRA Integration is detailed in a separate Administration Guide, which is available from Basis Technologies on request.

5.24.7. BMC Remedy Integration

The information contained in this section is aimed at experienced Remedy administrators and developers and does not cover in detail the setup and installation of Remedy workflow or foundation data as sufficient knowledge of such topics is assumed.

The integration has been developed against a Remedy environment running version 7.6.3 of the application server and ITSM applications, but code should work without issue on any 7.6.x environments. However, future implementations may be subject to change should BMC alter the way web services are implemented in Remedy.

Setup

There are 5 components that must be created and configured in order to complete the Web Service integration between ActiveControl and Remedy. These are:

1. The Basis Technologies Remedy Web Service User (ActiveControl to Remedy Web Service calls)
2. Remedy foundation components that are needed to underpin web service trigger code on the Remedy application server.
3. Remedy code running in the OOTB Change Management module to trigger a web service call.
4. The Basis Technologies Remedy Web Service Interface form (Remedy to Service calls)
5. The Basis Technologies Remedy Web Service Interface form workflow (Remedy to Service calls)

5.24.7.1. Web Service User

In order that ActiveControl can make use of the Out-Of-The-Box (OOTB) Remedy Web Service for creating Change Work Detail entries, a named account will need to be created in order to allow connection to the designated Remedy application server that is hosting the Web Service interface.

Using a specific account obviates the need for the use of a nominated account to be configured in the Remedy Mid-tier to handle anonymous Web Service invocation.

Using a specific account to allow ActiveControl to consume a Remedy Web Service is recommended for the following reasons:

- The use of an anonymous Web Service call may run counter to the customer’s application security policy
- A specific account increases the auditability of Web Service invocation and traceability of processing during any debugging activity.

The default account used to allow ActiveControl to consume Remedy Web Services is btiwebservice. If the customer wishes to use another account to comply with security or corporate naming standards, then this would not be an issue, but it would require modification to the ActiveControl code that consumes the Remedy Web Service as this has been built to use the btiwebservice account.
The account must be set up via a profile in the CTM:People form in order to correctly allocate the desired licenses and application permissions. OOTB, the account only requires permissions to the Change Management application. In order that it can guarantee application access, the account must be allocated a fixed AR user license unless the customer operates a sufficient floating license ratio.

Furthermore, it is recommended that the btiwebservice account uses Master permissions for the ITSM modules in which it will be creating entries. This will avoid any application workflow exceptions that might

block such entries being created. However, the allocation of a floating application license for the module in question would be sufficient.

If the customer has made modifications to Change Management application that have altered the behaviour of Change Master rights in terms of business rules, then these will need to be taken into consideration when creating the account.

In order for the account profile to be created, it will need to be associated with a company defined in the customer's Remedy application. The appropriate company will need to be decided via agreement with the customer.

The following screenshots outline the field values for the btiwebservice account when set up against the Calbro Services OOTB ITSM application sample company that is created during Remedy installation.

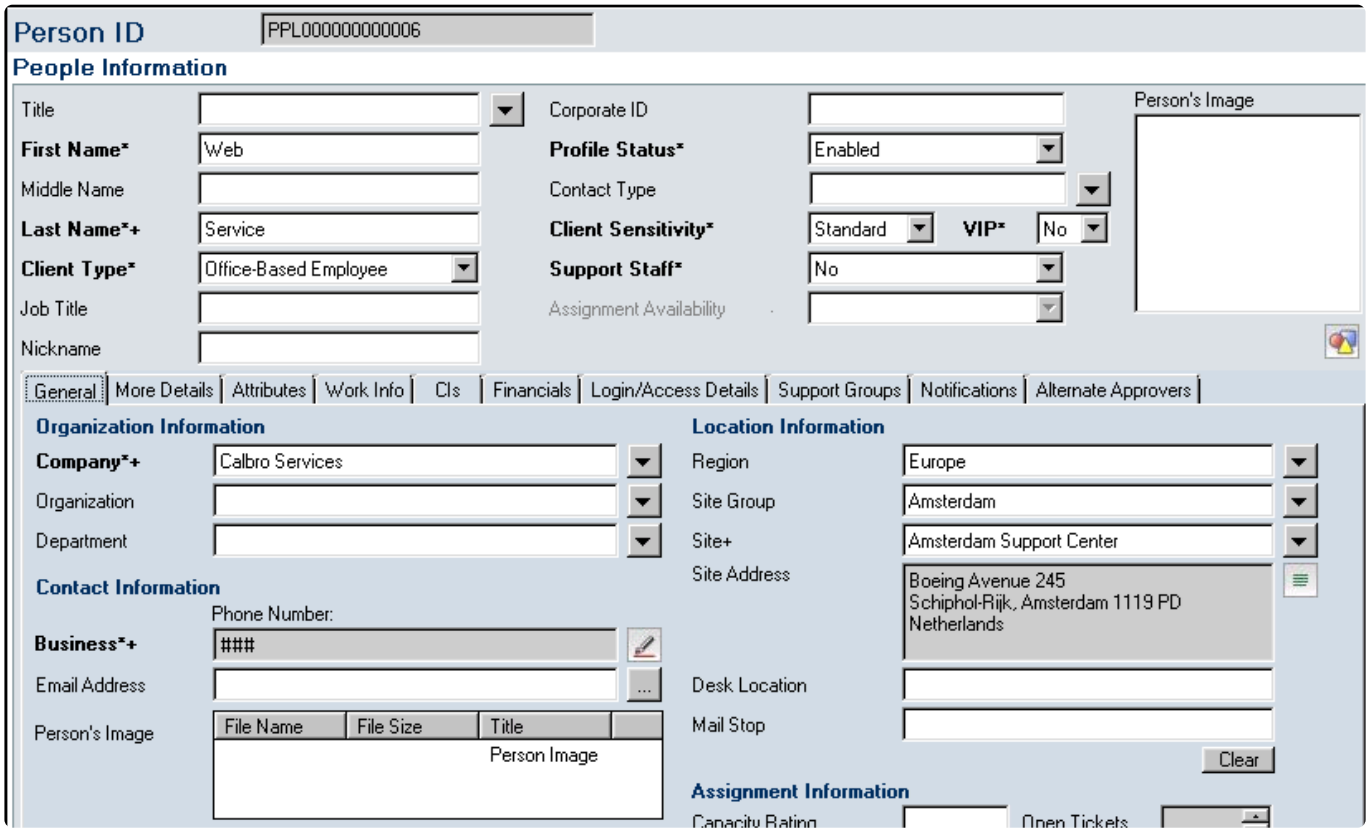


Figure 1: The General tab on CTM:People for the btiwebservice integration account

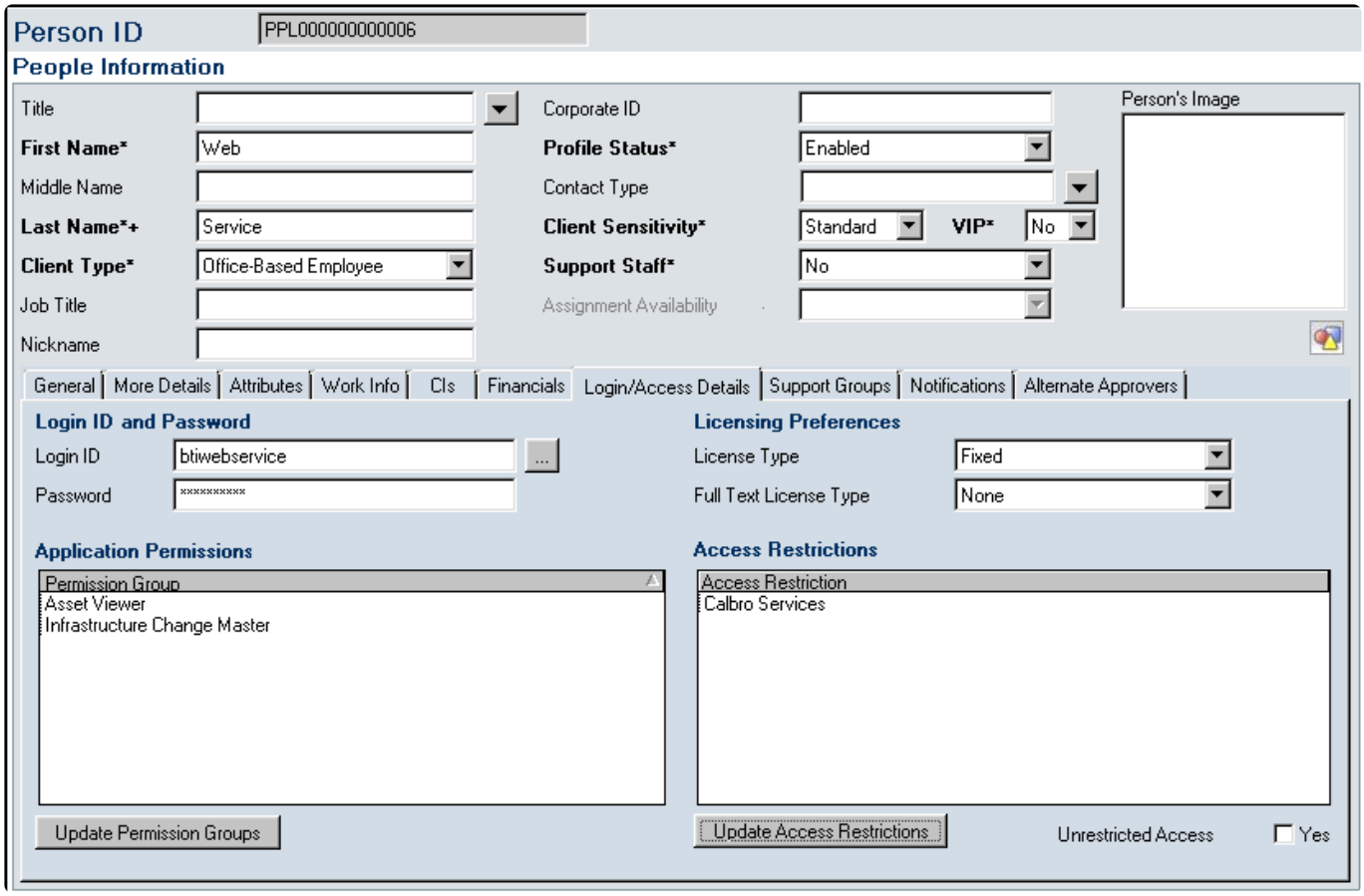


Figure 2: The Login/Account tab on CTM:People for the btiwebservice integration account

5.24.7.2. Remedy foundation data

Trigger Conditions

In order to provide a degree of flexibility as to when a web service might be called as part of the Remedy Change Management workflow, a new foundation form BTI:TE:WS_TriggerConditions (see Figure 3), has been created. This form can contain entries that are used to determine if an instance of a Change Request should call a web service in order to create a related Business Task in ActiveControl.

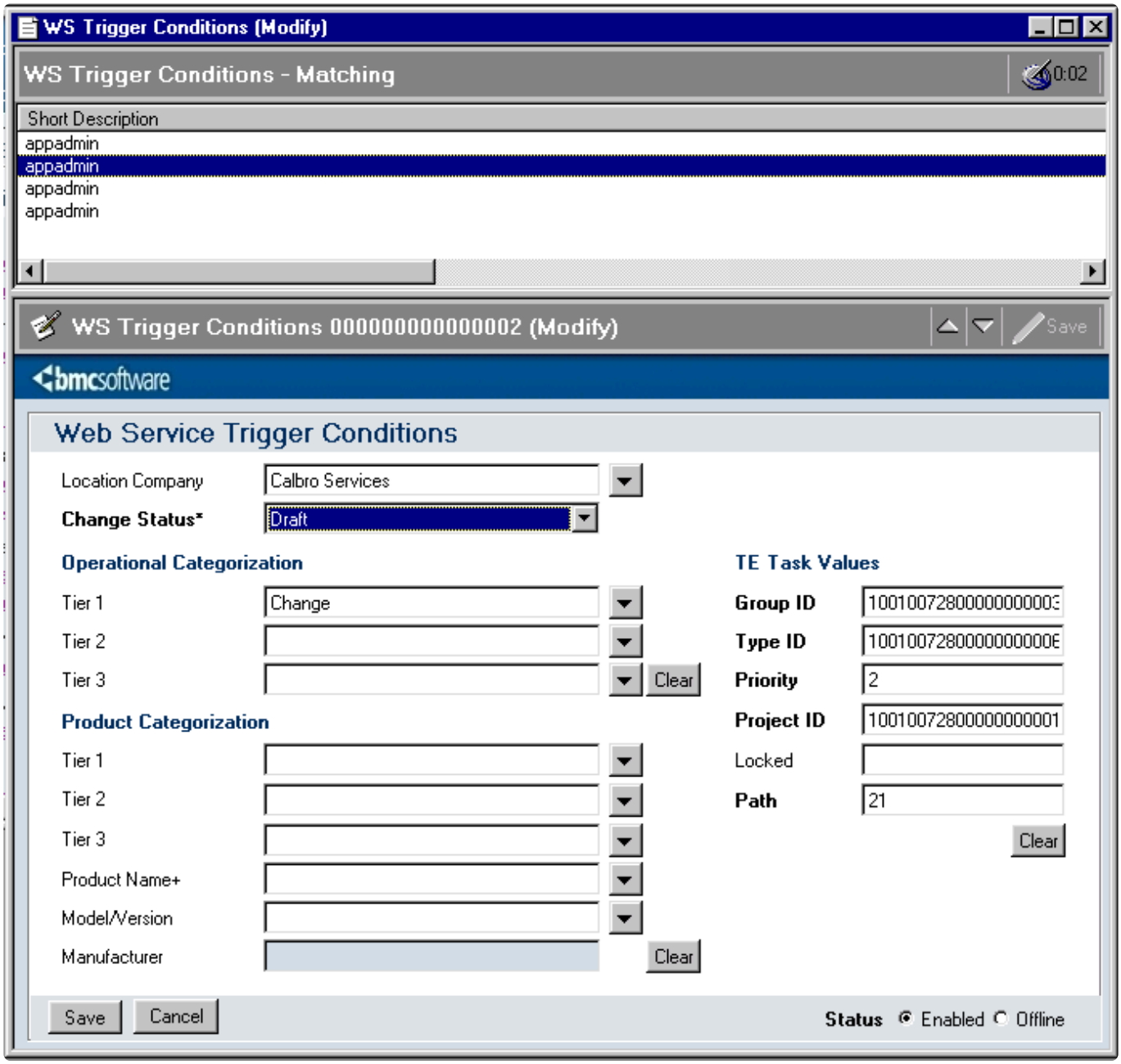


Figure 3: BTI:TE:WS_TriggerConditions

Entries in BTI:TE:WS_TriggerConditions define various criteria that a Change Request is compared against to determine if that request should create a Business Task in ActiveControl. The criteria entries must contain a minimum of a Location Company (matched against the Location Company field on the Change Request form or any) and a Change Request Status (matched against the Change Request Status field on the Change Request form), but can also include a combination of Operational and Product Category tier fields that if defined are matched against the corresponding field combinations in an instance of a Change record. Note that entries in this form can use the value of – Global – for the Location Company field, in which case that entry will match any Location Company value found in a Change Request.

In addition to the trigger criteria fields referred to above, entries in this form also contain Web Service input values that are applicable to that triggering condition. This allows flexibility to create Business Tasks in ActiveControl with different input values for different Change states, companies and Change categorisation combinations. It also allows for flexible maintenance of this data should it be required due

to configuration changes on the target environment.

To reach BTI:TE:WS_TriggerConditions in order to set up web service trigger condition records, it is necessary to create entries in the following forms:
• RAC:Config Options
• RAC:Config Tasks

These entries will ensure that a menu option appears in the Application Administration (Custom tab) under the Change Management -> Basis Technologies Settings menu structure.
The values for these entries are as follows:

RAC:Config Options

Field Label	Field Value
Application	Change Management
Object Name	BTIWSTrigger
Console Text	Basis Technologies Settings
Description	Manage Basis Technologies Web Service Triggers
Status	Enabled
Sorting Order	8
Display on Console	Yes
DataTags	Config-CHG

RAC:Config Tasks

Field Label	Field Value
Option Name	BTIWSTrigger
Task Name	Web Service Trigger
Console Text	Web Service Trigger
Task Type	Standard
Description	Trigger condition combinations (Location Company / Change Status / Op Cat / Prod Cat) can be created or modified using this option.
Accessible Groups	Infrastructure Change Config
Display on Console	Yes
Status	Enabled
DataTags	Config-CHG
Config Form	BTI:TE:WS_TriggerConditions

WUT View Name	Default Administrator View
Web View Name	Default Administrator View
Form Open Mode	Submit

Selecting the menu item for Web Service Trigger brings the form up in new mode. If you wish to modify existing entries in the form, simply change the mode to search and perform either a blank or QBE search. Pressing the Cancel button will close the form without saving changes or creating a new entry. Pressing the Save button will either create a new entry, or modify an existing one depending on upon the mode the form is currently in.

5.24.7.3. Custom Mappings

In addition to data defined trigger events, the Remedy integration allows users to define custom field mappings. These mappings allow administrators to specify any field available in the Remedy Change Infrastructure form and map it to the ActiveControl custom field of their choice.

Note: The example custom field mappings should be deleted once the package has been imported. These are for example purposes only.

As with the trigger conditions, the custom mappings administration form is accessible via the application administration console. To enable this, entries must be created in the following forms:

- RAC:Config Options
- RAC:Config Tasks

These entries will ensure that a menu option appears in the Application Administration (Custom tab) under the Change Management -> Basis Technologies Settings menu structure.

The values for these entries are as follows:

RAC:Config Option

Field Label	Field Value
Application	Change Management
Object Name	BTIWSMapping
Console Text	Basis Technologies Settings
Description	Manage Basis Technologies Web Service Mappings
Status	Enabled
Sorting Order	8
Display on Console	Yes

DataTags	Config-CHG

RAC:Config Tasks

Field Label	Field Value
Option Name	BTIWSMapping
Task Name	Web Service Mapping
Console Text	Web Service Mapping
Task Type	Standard
Description	Custom field mappings for ActiveControl Task web service
Accessible Groups	Infrastructure Change Config
Display on Console	Yes
Status	Enabled
DataTags	Config-CHG
Config Form	BTI:TE:WS_CustomMappings
WUT View Name	Default Administrator View
Web View Name	Default Administrator View
Form Open Mode	Submit

Selecting the menu item for Web Service Mapping brings the form up in new mode. From this screen you can create, modify or delete mappings entries as desired. The source fields are drawn from the Infrastructure Change form and the target fields are those that data will be pushed to on the BTI:TE:WS_Interface form when a web service call is set up.

5.24.7.4. Remedy Web Service trigger code

This code is attached to the CHG:Infrastructure Change form and compares field values in a Change Request with values in any entries found in BTI:TE:WS_TriggerConditions . If any match is found then code is triggered to push an entry into the form BTI:TE:WS_Interface which in turn will commence consumption of the ActiveControl task create web service.

The code has been written to run against an OOTB Change implementation. The execution order has been set to a range to try and ensure that it will also work in as many modified implementations as possible. However, if modifications have been made to the OOTB code, compatibility cannot be guaranteed and some modification may be necessary on a per customer basis.

The value in the Change Request number field (field ID 1000000182) of a given Change Request is used as the reference for the task that will be created in ActiveControl. Because that reference must be unique in it is only possible to create a single task per Change Request in Remedy.

The tester id of the task is set to the Remedy user ID (field 1000003231) of the assignee in the Change

Request record.

5.24.7.5. Web Service Interface form

The form BTI:TE:WS_Interface is used to consume Web Services on the ActiveControl side. It is recommended that any additional Web Service calls made to ActiveControl over and above task creation do so using code associated with this form.

At the time of writing, workflow has been developed and tested for creating tasks in ActiveControl, but other workflow along similar lines could be created in Remedy to consume further Services. Such calls may necessitate the creation of additional input (i_) and output (o_) fields to accommodate the specific interfaces of other Services.

An overview of processing carried out by this form follows:

1. A record is created from workflow running on the CHG:Change Infrastructure form if values from a triggering condition record in the BTI:TE:WS_TriggerConditions form matches the field values in the Change Request record. The fields set at this point are:

Field	Value
z1D Action	"CREATETETASK"
SourceForm	$SCHEMA$
z1D_TriggerId	$z1D_RecordExists?$
sysRequestId	$Request ID$
i_Testerid	$ASLOGID$
i_Reference	$Infrastructure Change ID$
i_Caption	$Description$
i_ChgReqStatus	$Change Request Status$

2. An instance ID GUID is generated.

3. The Web Service input fields from the trigger condition entry referred to in step 1 are pulled into the i_ fields in this record along with some additional fields in the change record that is keyed off the sysRequestId field.

4. A value of CREATETASK is pushed to the z1D Action field in this record in order to trigger modify related code.

5. The first action on modify is loop through any custom field mappings defined in the BTI:TE:WS_CustomMappings form and pull the relevant source field values from the change request keyed off the sysRequestId field. Each value is drawn into the specific zi_ field in the interface record that is specified in the target reference in the mapping entry.

6. Once all mapping entries have been processed, the next step is to call the ActiveControl create task web service using the values of the input (i_ and zi_) fields.

7. If any errors result in the calling of this Web Service, the interface record is placed into an error state and a Work Log entry associated with the initiating Change Request is created. This entry contains details of the error encountered when calling the Web Service.

8. If there was an error on the ActiveControl end of the call that occurred after successful invocation of the Web Service, then the details will be contained in the output Web Service Message fields. An error is denoted by the code E in the o_Msgtyp field. In such an event, code will trigger to place the interface record into an error state and to create a Work Log entry associated with the initiating Change Request. This entry contains details of the error encountered after successful invocation of the Web Service.

9. Alternatively, if the Web Service was called successfully, it will return output values to the Web Service output (o_) fields.

10. If there was no form of error resulting from the Web Service call, it means a task now exists in ActiveControl that has as a reference equivalent to the Change Request number of the triggering Change. This outcome is indicated by the o_Msgtyp return field containing a null value. A successful call and return to the ActiveControl create task Web Service results in the interface record being marked as Processed and a push being performed to create a Work Log entry associated with the initiating Change Request. This entry contains details of the newly createdActiveControl Business Task.

5.24.7.6. Setup of Web Service user

If implementation of the Web Service integration at a customer’s site requires consumption of the Task creation Web Service, then if an application account specifically for Web Service consumption does not already exist, one must be created.

Whether using an existing or newly created account, the details of the account and its password must be supplied to the person(s) responsible for setting up the Remedy side of the integration.

This is so that these details can be captured when the filter consuming the Web Service is being amended for site specific operation.

In the Basis Technologies environment, this account is named remedy and the password for it is remedyrfc.

5.24.7.7. Web Service Workflow setup

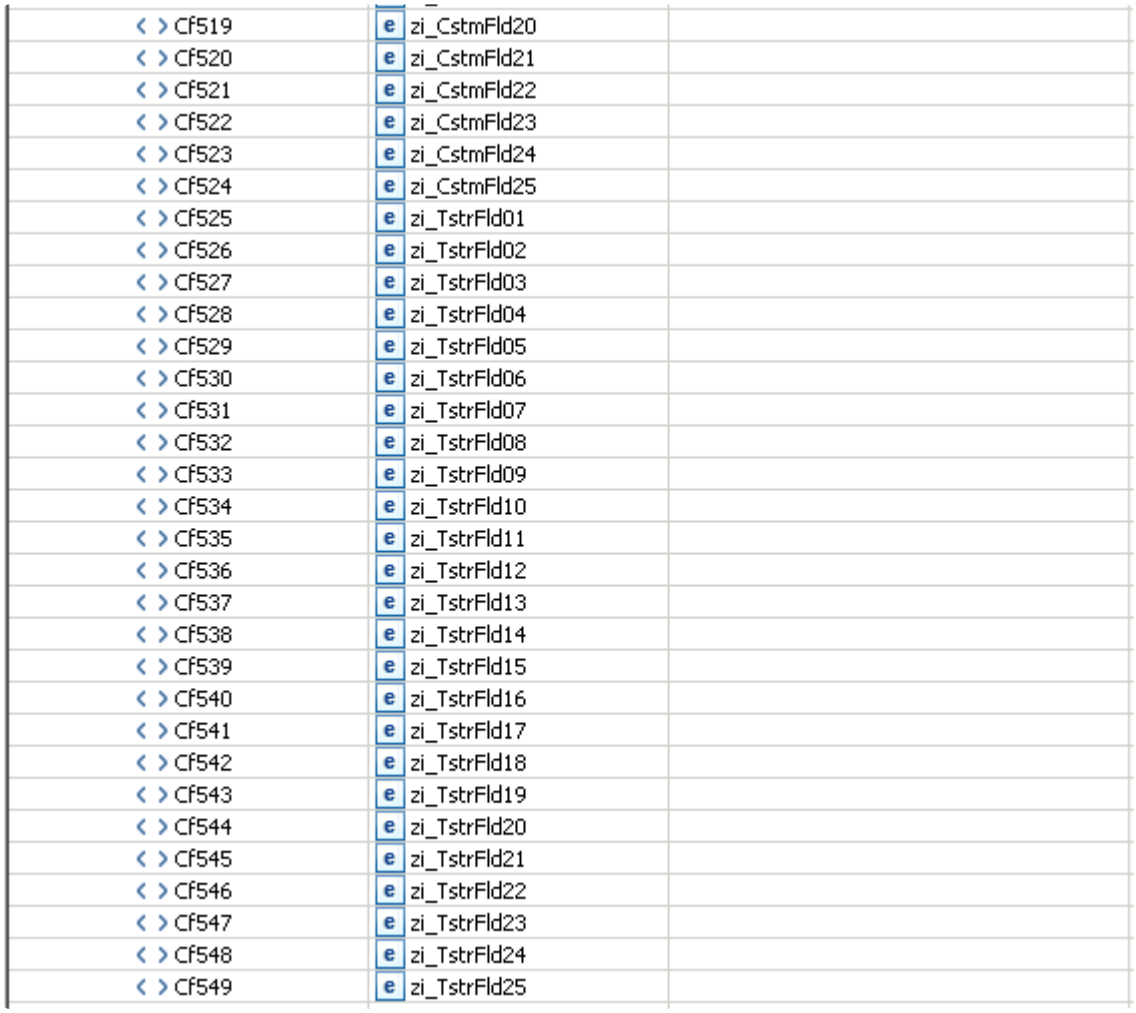
The filter BTI:TE:CreateTETask_M_500_SF(WS:_-bti_-teTaskCreateWs-v2, has been developed as a template for consuming the ActiveControl task creation Web Service. When it is imported at a customer site, the server reference in the WSDL File line must be updated with specific values for the customer’s location. There should not be the need to reload the WSDL once these changes have been made, it should simply be a case of pressing the login button again to apply the user name and password referred to in section 3.3. Once this is done, the filter can be tested for connectivity.

However, should it be necessary to remap inputs and outputs for the Web Service call, then the following

mappings need to be applied.

XML Data Type	Form/Field	Mapping Info
ROOT	BTI:TE:WS_Interface	Primary Key = Request ID
XDescription	zi_DetailedDescription	
XSystemNumber	01	
XTask		
Id	i_TaskId	
Caption	i_Caption	
Reference	i_Reference	
Groupid	i_Groupid	
Typeid	i_Typeid	
Testerid	i_Testerid	
Priority	i_Priority	
Projectid	i_Projectid	
Locked	i_Locked	
Path	i_Path	
StatDepl		
StatPlan	i_ChgReqStatus	
StatDeplMan		
StatPlanMan		
Owner	zi_ChangeCoordinator	
Text		
Cf500	zi_CstmFld01	
Cf501	zi_CstmFld02	
Cf502	zi_CstmFld03	
Cf503	zi_CstmFld04	
Cf504	zi_CstmFld05	
Cf505	zi_CstmFld06	
Cf506	zi_CstmFld07	
Cf507	zi_CstmFld08	
Cf508	zi_CstmFld09	
Cf509	zi_CstmFld10	
Cf510	zi_CstmFld11	
Cf511	zi_CstmFld12	
Cf512	zi_CstmFld13	
Cf513	zi_CstmFld14	
Cf514	zi_CstmFld15	
Cf515	zi_CstmFld16	
Cf516	zi_CstmFld17	
Cf517	zi_CstmFld18	
Cf518	zi_CstmFld19	

Figure 4: Input mappings for the ActiveControl create task Web Service call



Cf519	zi_CstmFld20
Cf520	zi_CstmFld21
Cf521	zi_CstmFld22
Cf522	zi_CstmFld23
Cf523	zi_CstmFld24
Cf524	zi_CstmFld25
Cf525	zi_TstrFld01
Cf526	zi_TstrFld02
Cf527	zi_TstrFld03
Cf528	zi_TstrFld04
Cf529	zi_TstrFld05
Cf530	zi_TstrFld06
Cf531	zi_TstrFld07
Cf532	zi_TstrFld08
Cf533	zi_TstrFld09
Cf534	zi_TstrFld10
Cf535	zi_TstrFld11
Cf536	zi_TstrFld12
Cf537	zi_TstrFld13
Cf538	zi_TstrFld14
Cf539	zi_TstrFld15
Cf540	zi_TstrFld16
Cf541	zi_TstrFld17
Cf542	zi_TstrFld18
Cf543	zi_TstrFld19
Cf544	zi_TstrFld20
Cf545	zi_TstrFld21
Cf546	zi_TstrFld22
Cf547	zi_TstrFld23
Cf548	zi_TstrFld24
Cf549	zi_TstrFld25

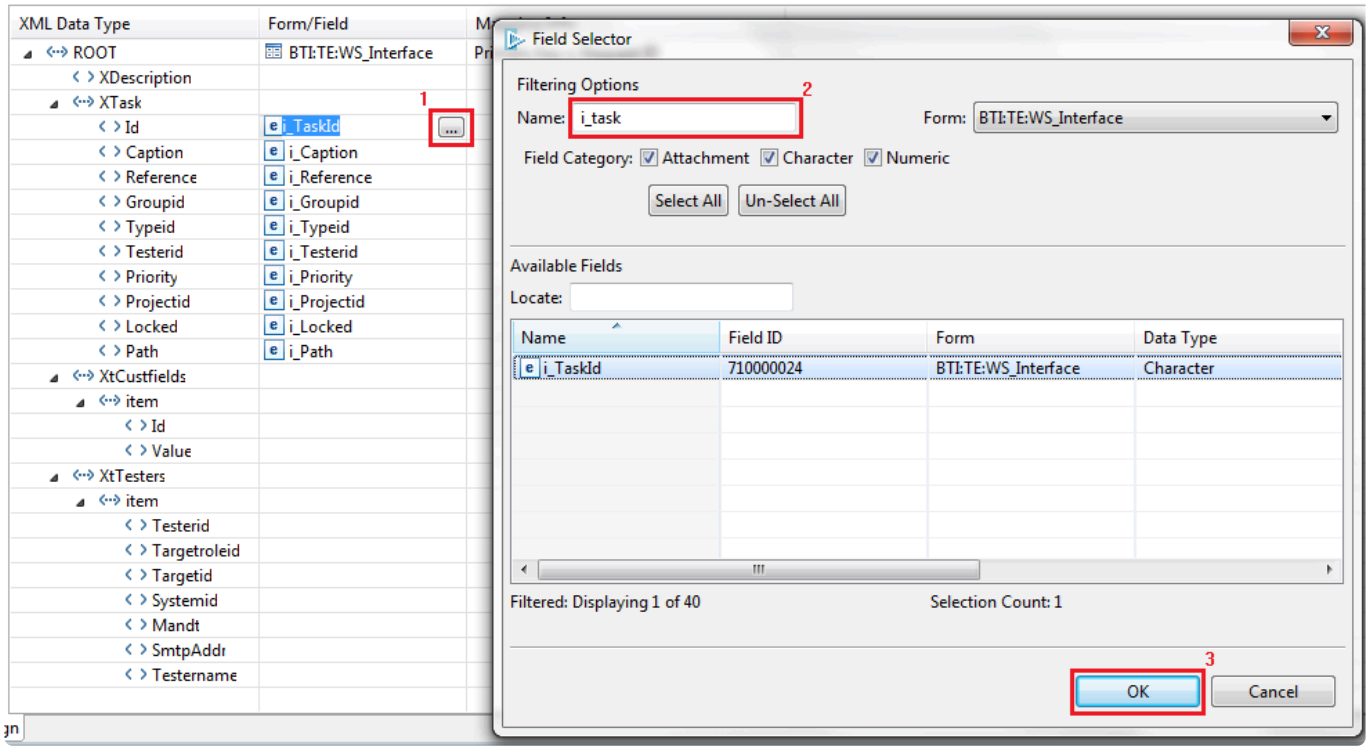
Figure 5: Output mappings for the ActiveControl create task Web Service call

NOTE: These mappings relate only to the create task Web Service and input/output mappings to other Services are expected to be different.

NOTE: The filter BTI:TE:CreateTETask_M_500_SF(WS:_-bti_-teTaskCreateWs is from the previous implementation of the integration that did not include custom fields. It is included in the package for reference purposes only and would not normally be enabled for current ActiveControl implementations.

To reapply mappings, follow this process:

1. Click in the Form/Field cell for which you wish to map the Remedy field to/from. At this point a dialogue pop up button will appear (red square 1).
2. Click on this button to bring up the Field Selector dialogue.
3. Via this dialogue, use the field name box (red square 2) to type the name of the field on the BTI:TE:WS_Interface form that will either be used as an input to the ActiveControl Business Task creation web service, or receive an output from that web service. This is an auto-complete field so as you type the list of matching available fields will diminish.
4. From the list of available fields, select the correct one and press the OK button (red square 3).
5. Repeat this process for each Remedy field on the BTI:TE:WS_Interface form that will be used as an input (i_ prefix) to the ActiveControl create task web service, or receive an output (o_ prefix) from consumption of that service. Refer to figures 4 and 5 for the complete list of field mappings.



5.24.7.8. Use of values returned from a Web Service call

Once values have been returned from a Web Service call to ActiveControl, what is done with those values will most likely be customer specific. Additional customer specific workflow would need to be written on the Web Service interface form. This workflow would execute on modify at execution orders higher than 500 in order to pass the returned values out to customer specific forms, or to OOTB ITSM forms other than CHG:WorkLog.

For example, if the task id were to be written to a custom third party reference field in the Change Request that had issued the call to the Web Service in the first place, then a filter would need to be written to handle this requirement.

5.24.7.9. Consumption of the Task Create Web Service

This section outlines the processing that takes place in order for a Business Task to be created when triggered by changes to the status of a Remedy Change Request

5.24.7.10. Consumption of the Remedy Create Change Work Info Web Service

This section outlines the processing that takes place in order for ActiveControl to create a Work Info entry linked to a specific Remedy Change Request entry.

Remedy Service trigger code

The inbound integration between ActiveControl and Remedy utilizes OOTB Remedy Web Services. In particular, the CHG_ChangeInterface_WS Web Service defined against the CHG:ChangeInterface form.

In most implementations, in order to access this Web Service, ActiveControl will need to use a specified Remedy account. Details of default account ActiveControl uses to connect to Remedy can be found in earlier section. If an alternative account is to be used, details will need to be supplied to the customer's staff with responsibility for ActiveControl in order to allow them to configure and test the Service consumption code.

Consumption of the CHG_ChangeInterface_WS Web Service requires a two part operation. The first operation (Change_Query_Service) is used to retrieve details of the Change Request for which a Work Info entry is to be created. In order to do this, it will be necessary to supply an Infrastructure Change ID (field 1000000182 on CHG:Infrastructure Change). This ID will be available to ActiveControl from a previous task create Web Service operation initiated from the Change Request in question.

Once details of the Change Request have been retrieved, the second operation (Change_Modify_Service), is then used to pass in details of the new Work Info entry.

As previously stated, the default implementation of the ActiveControl to Remedy integration uses OOTB Remedy Web Services. This means that once connection details are configured in the Service consumption code, then no further action is required on the Remedy side once connection and functional testing have been satisfactorily concluded.

If however, changes have been made to either the OOTB Web Services referred to above, or code that fires as a result of the consumption of these Web Services, then depending upon the nature of these changes, it may be necessary for Basis Technologies Development team to alter the Web Service consumption code in order for the integration to work at a given customer site. For example, such customisations might include the addition of custom required fields, or fields that are enforced via the use of custom business rules.

5.24.8. ChaRM integration

ActiveControl includes a ChaRM integration that was developed for an existing customer that wanted to continue using their productive ChaRM approvals workflow – whilst also benefiting from various ActiveControl functionalities such as Merge (because they had some challenges with ChaRM Retrofit in the past) and the extensive suite of ShiftLeft analysis checks to help improve the quality of change

moving through the SAP landscape.

The ChaRM integration within ActiveControl covers five main requirements – essentially delivering an integration of both Business Task and Transport Form level items.

(1) Business Task Creation
ActiveControl will automatically create a Business Task at the point a ChaRM Change Request reaches a certain (configurable) status. This will involve values in both standard and customer Business Task fields.

(2) Transport Form Creation
ActiveControl will automatically create a Transport Form for every new SAP transport created in certain (configurable) Development systems. The Transport Form will be created as soon as SAP transport first created by ChaRM. Only Workbench and Customising transports should have a Transport Form created for them. All Transport of Copies created by ChaRM, and all Merge TOCs created by ActiveControl should not have a Transport Form created for them by this new automation. The Transport Form will be automatically linked to the appropriate Business Task.

(3) Business Task Updates
Updates in Solman/ChaRM to certain fields of a particular CR will be reflected in the corresponding Business Task.

(4) Transport From Updates
Updates in Solman/ChaRM to certain fields of a Change Document will be reflected in the corresponding Business Task.

(5) Transport Decoupling
If a Transport is “decoupled” from its ChaRM CR – it will be assigned to a single (configurable) Business Task within ActiveControl.

Configuration Steps

The ChaRM Integration is detailed in a separate Integration Administration Guide, which is available here.

5.24.9. Integration based on Custom Field

The Integration Framework typically integrates based on the Business Task [Reference] field containing the 3rd Party ticket number.

Since some customers prefer to use ActiveControl auto-generation of Task References – it is now possible to integrate based on the 3rd Party ticket number being stored in a separate ‘custom’ field.

Please note that this capability is currently only available as part of JIRA integration.

Configuration Steps

The Configuration Steps are detailed on Basis Technologies online forum – however the reality is that this capability would typically be configured by Basis Technologies as part of your Integration setup.

5.25. Airgap

Airgap is a requirement within some organisations with secure networks whereby they cannot allow changes to be electronically transmitted from development to their production systems. It is typically found in industries such as the public sector, aerospace and defence, exploration and energy. In an SAP context, an Airgap typically means that following Development or QA or Regression testing, the SAP transport files need to be downloaded to some external media (disk, USB drive, DMZ) and then physically moved and uploaded to the production system side of the Airgap for import there.

High Side / Low Side

The key principle of the ActiveControl Airgap solution is the presence of two Domain Controllers, one on the Low side and one on the High side.

- The Low side is the Dev instance
- The High side is the (main) Production instance.
- Information is passed from one AC instance to the other in order to manage the complete flow of change.

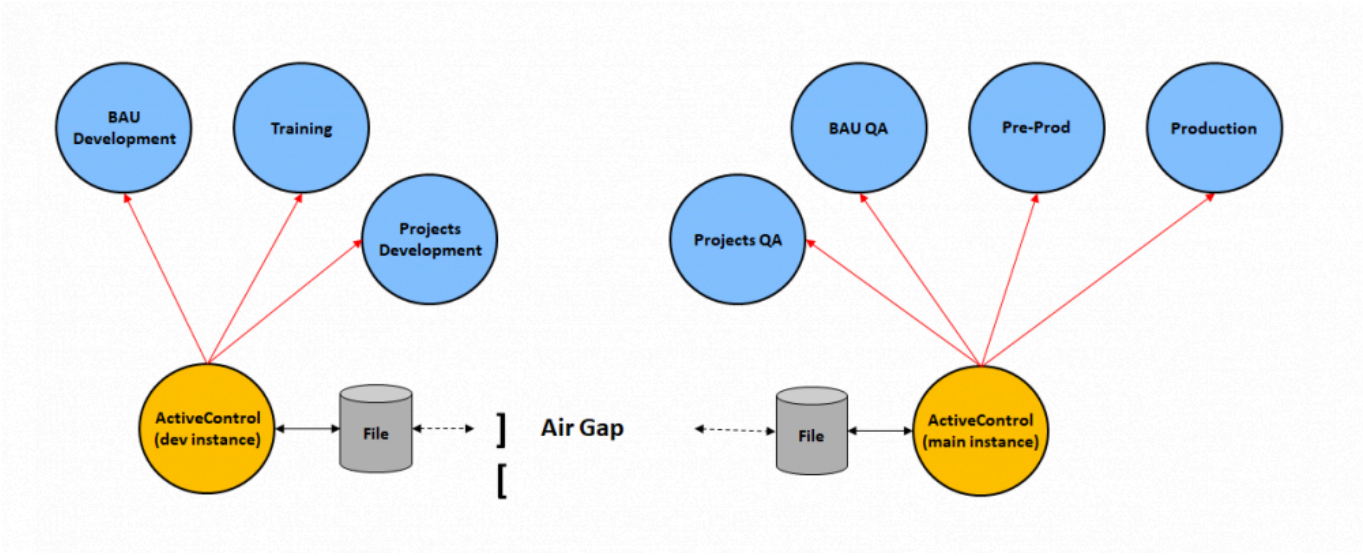


Figure: Airgap within ActiveControl

Configuring Airgap

Airgap usage requires the configuration of various ActiveControl tables, and the scheduling of several

programs.

> Further information on the Airgap configuration – plus screenshot examples – can be found in this online FAQ.

Table /BTI/TE_TVARV

Standard configuration table /BTI/TE_TVARV is used as part of Airgap configuration, with various new variables used to achieve the required scenarios.

1) Path location of files
Configuration, Business Task and Transport Form related files will be written to a (user configurable) location as part of the solution. This is done via the TE_AIRGAP_OUTBOX variable on both the Low and High side.

2) Send Business Task and Configuration from High Side to Low Side
Most Airgap customers will want to create the Business Task on the High Side, and then have this sent over the Airgap to the Low side. This is done via the following configuration on the Low and High side:

High Side: Maintain TE_AIRGAP_ALLOWED_RECIDS_OUT in /BTI/TE_TVARV with value TASK, CONFIGURATION and ACKNOWLEDGE.
Low Side: Maintain TE_AIRGAP_ALLOWED_RECIDS_IN in TVARV with value TASK, CONFIGURATION and ACKNOWLEDGE

3) Create Transport Form on Low Side, send to High Side
Most Airgap customers will want to create the Transport Form on the Low Side, and then send this over the Airgap to the High Side. This is achieved with the following configuration.

Low Side: Maintain TE_AIRGAP_ALLOWED_RECIDS_OUT in /BTI/TE_TVARV with value TRANSPORT and ACKNOWLEDGE in the field low.
High Side: Maintain TE_AIRGAP_ALLOWED_RECIDS_IN in /BTI/TE_TVARV with value TRANSPORT and ACKNOWLEDGE in the field low.

4) Send programs
In order for the value entered in field external system in the AirGap SEND programs to be validated, values must be maintained in /BTI/TE_TVARV.

High Side Configuration:
AIRGAP_ALLOWED_SYS_IN: the systems valid for incoming messages (usually only one). Used to validate the system ID for inbound messages (used to receive tasks on low side and forms on high side)
AIRGAP_ALLOWED_SYS_OUT: the systems valid for outbound messages (usually only one). Used to validate the system for outbound messages (used to send tasks to low side and forms to high side)

Low Side Configuration:
AIRGAP_ALLOWED_SYS_IN: as per above description.

AIRGAP_ALLOWED_SYS_OUT: as per above description.
AIRGAP_ALLOWED_SYS_DECODE: used to validate the system used to decode task IDs (currently only used when sending forms, low side only)

Table: /BTI/TE_AGMSGTYP

For the framework to pick the correct class for every message type, table /BTI/TE_AGMSGTYP will have require entries to be configured. These table entries are shipped as part of the standard ActiveControl software transports. You do NOT need to maintain or change this table manually.

Table: /BTI/TE_INT_SYST

To send Business Tasks or Transport Forms over the Airgap, an integration ID must be defined for both systems in standard Integration table /BTI/TE_INT_SYST.

Table: /BTI/TE_INT_MAPP

As part of Airgap Solution, integration mapping is only required for field PATH for creating a transport form on the low side. All other values such as Groups, Types are same in both system.

Table /BTI/TE_INT_MAPP should be configured on the High side as follows:

Field	Description		
EXTSYS_NO	Value configured in table /BTI/TE_INT_SYST		
EXTSYS_NAME	Name: configured in table /BTI/TE_INT_SYST Direction: Inbound (I) Sequence No: Doesn't have to be in sequence but just to have part of key	TEFIELDREF & EXTFLD_VAL	Field that needs conversion. Will be same. Refer structure /BTI/TE_ST_IFORM for other fields for transport forms.

Table: /BTI/TE_INT_CONV

Table /BTI/TE_INT_CONV should be configured on the High side as follows:

Field	Description
EXTFLD_ID	Value of path from other side(LOW)
EXTFLD_VAL	Value of path in the same system(HIGH)

Airgap: Programs

After maintaining the aforementioned Airgap configuration tables, several programs also need to be scheduled as part of the Airgap solution.

Program: /BTI/TE_RUAIRGAP_SEND_TASKS

/BTI/TE_RUAIRGAP_SEND_TASKS is used on the Low side to send Business Task information to the High side.

Option	Description
Receiver	Value must be the number configured in table /BTI/TE_INT_SYST on High side.
Sender	value must be the one configured in table /BTI/TE_INT_SYST on Low side.
Specify Target and Location for when the transports are ready to send across Airgap.	

Program: /BTI/TE_RUAIRGAP_SEND_CONFIG

Program /BTI/TE_RUAIRGAP_SEND_CONFIG is used to send Groups, Types, Projects and Custom Fields from High Side to Low Side.

It is recommended to check the flag “Only send if changed”. Otherwise, same data will be sent on each run and generates spool.

Program: /BTI/TE_RUAIRGAP_PROCESS_INBOX

Program /BTI/TE_RUAIRGAP_PROCESS_INBOX is used on both the Low side and the High side to create what is required.

Option	Description
SAP Directory name	Directory where all the messages are stored to process.
No. Retries before error	The number of times the program will try to process the message before changing the status to failed.
Archive processed & failed msg	(Default) Choosing this will move down the file to the specified path if the message is processed or failed.
Delete processed & failed msg	Choosing this will delete the file if the message is processed or failed.

Keep the files	Select this if you want files to remain in the inbox. If the option Keep the files is chosen, it is better to flag the Checkbox ‘Display at least one new file is processed’ when running in background to avoid generating the spool unnecessarily for an already processed/failed messages.

5.26. Consolidated Import Queues

If you have the same SAP systems configured on more than one path in ActiveControl (e.g. to separate BAU and project changes into different paths through the same SAP systems), it is possible to set up a Consolidated Import Queue (CIQ).

This gives a complete view of all transports waiting to be imported into that system and can be useful during cutovers and for system refreshes.

Even if a target is part of a CIQ, transports can still be imported using the individual Target Import Queues, as well as using the consolidated queue. A CIQ can be made up of any number of targets, all of which must point to the same physical system. Configuration can be used to show the CIQ within the respective paths or be shown as a separate entity in the path window of the GUI.

An example of when you may want to use a CIQ is if you configure ActiveControl to have two paths using the same systems, one for Production Support / BAU and one for Minor Projects. In ActiveControl the paths could look something like this:

Production Support / BAU:	D01 -> T01 -> PRD -> TRN
Projects:	D01 -> T01 -> PRD -> TRN

If you want to see all of the transports waiting to be imported into PRD, you could look at the Import Queues of each of the PRD Targets in each path, but you could also configure a CIQ that will show you the transports waiting to be imported into PRD in both paths.

- Note: It is important to remember that when importing using a CIQ, all import options are taken from the CIQ Target configuration and NOT the individual Targets that make up the CIQ. For example, if the individual Targets are set up to create backup / merge transports, but the CIQ Target is not configured to do so, when importing using the CIQ a backup / merge transport will not be created, whereas if you import transports using the individual Import Queues, then backup / merge transports would be created.
- To be able to see Consolidated Import Queues within the GUI, users must have the VIEWCONIMPORTQUEUE activity in the TE_USER authorisation object. This is delivered in the ActiveControl Basis and Admin roles, as detailed in the ActiveControl Roles Matrix.

Configuring a CIQ

To configure a Consolidate Import Queue, the following steps should be performed:

Step	Activity
1	Create a Target in ActiveControl for the SAP system that you wish to show a consolidated view for. This Target cannot have any control points configured. To act as a CIQ, this Target must have the configuration option set
2	Configure the Import Options for the CIQ target.
3	In each of the individual Targets that make up this CIQ, the configuration options below must be entered. I.e. you need to point each individual target for the SAP system to the relevant CIQ target.
4	If you want the CIQ to be displayed within the individual paths, select the consolidated queue visible check box. When looking at the path in the GUI, you will see the CIQ as a child object to the individual Import Queue.
5	If you want to display the CIQ as part of a path (i.e. separately and not visible within the individual paths), create a path as normal, but set the path status to “Cannot be assigned to new Transports, but visible within….”

This will allow you to show this and any other CIQ’s as separate entities in the GUI Path/Target view but not allow Transport Forms to be created for them as these should still use the individual paths

Once a CIQ has been configured in ActiveControl, it will automatically show all transports waiting in the Import Queue of each related individual Target. In the screen shot above, the T01 CIQ has been configured to show all the transports in both the ‘BT QA Test’ T01 Target and the ‘BT Project Test’ T01 Target. It shows that there are 16 transports and 2 transports waiting in each of the respective Import Queues. Therefore the CIQ is showing 18 transports. The order in which ActiveControl imports the transports in a CIQ is dependent on the Import Options configuration of the CIQ Target.

! To import the transports in release order, the import options for the CIQ Target and all the related Targets must have the same Import configuration.

* To import the transports in the same order as a previous system, the CIQ Target should have something selected for the configuration option “Try to import transport requests in the order….”. It does not matter which Target is selected here, as the actual ‘previous system’ that ActiveControl will check is the one configured in each individual Target that makes up the CIQ.

5.27. Client-based Transport distribution

Some SAP customers have a SAP client architecture requirement to distribute transports to particular system clients in the landscape, based on rules and the source client of the transport. ActiveControl includes a capability to automate this client-based transport distribution, which can be setup via the

following steps.

1) Configure table /BTI/TE_IMP_CLI in the Domain Controller with the required transport distribution rules

Field	Description
PATH	the ActiveControl path for which the rule is being defined.
SOURCE SYSTE	the source system (SID).
SOURCE CLIENT	the source system client.
TARGET	the target system in the path where the transport has/not to be imported into particular client.
SORT ORDER	This is used for sorting the clients in a particular target.
TARG CLIENT	The client of the target system.
SKIP	An X should be entered in this field for the rules where you want the transport to not be imported into the defined client defined in the rest of the rule.

2) Switch on TE_EXIT_SAMPLE_0065 in /BTI/TE_EXITC table in the Domain Controller

Notes:
i) the import queue is not skipped, it is during the import that the skipping occurs based on the rules defined in the configuration table
ii) the clients will be sorted based on SORTORDER
iii) if a client is not defined in the configuration it will be ignored
iiii) if no entries are found for a transport it will be imported in all the clients defined in the config
iv) the transports can still be imported using both single one-by-one or block import.
v) It is NOT possible to use this functionality during the standard Merge process into a multi-client Dev system (as Merge process will import the source transport into all configured clients of the Development target and then create the Merge TOC in one of those clients). If you really want to achieve Merge to specific client, the solution would be to configure multiple Merge targets, each with one client.

5.28. Additional Metadata

Various optional ‘Business’ and ‘User’ metadata is available within ActiveControl for reporting purposes.

Business metadata

Additional metadata can be recorded against Business Tasks via the [Additional Data] tab of the Business Task screen on both the Windows GUI and Web UI. The metadata seen within the various fields on this tab is configurable, it is maintained via the following tables in the ActiveControl domain controller:

Table	Description
/BTI/TE_BUS_UNIT	Business Unit.
/BTI/TE_STTHRUST	Strategic Thrust.
/BTI/TE_COMP	Complexity.
/BTI/TE_AREA	Area.
/BTI/TE_TSK_PRTY	Task priority.
/BTI/TE_CONFIG	Set the currency to be used.

User metadata

Additional metadata can be recorded against individual users on the [Details] screen within the Web UI. The metadata seen within the various fields on screen is configurable, it is maintained via the following tables in the ActiveControl domain controller:

Type	Table	Description	Notes
User	/BTI/ TE_EMPL_TYP	Employee Type	
User	/BTI/ TE_SUPPLIER	Supplier	
User	/BTI/ TE_GEOLOC	Location (With Geo coordinates)	
User	/BTI/ TE_COSTBAND	Cost band	
User	/BTI/ TE_USERPREF	User preference data	
User	/BTI/ TE_WEBUICFG	Web UI config	SUPLPIC_PREFIX and SUPLPIC_URLPREFX need to be maintained

5.29. Reports

ActiveControl includes a number of out-of-the-box Reports aimed at cutting out the need for spreadsheets and manual analysis performed within many organisations as part of the delivery of SAP change.

These Reports are available in the Windows GUI and Web UI, and also via transactions in the SAP GUI of the Domain Controller. A Reports and Utilities menu is also available in the SAP GUI.

Report Name / Transaction	Description
/BTI/ TE_RTASK_STATUS_REPORT /	This operational report gives detailed project and task level status information. It provide a snap shot of the status of projects giving both

/BTI/TE_RTASK_STATUS	detailed and summary information about the status of all items relating to the projects. Some of the key data that can be reported upon is: • Project/Task/Transport reporting showing the current deployment status (e.g. In development, In test, Awaiting production, etc.) • Project/Task reporting showing the current planning status • The full details of all items returned (Projects/Tasks/Transports) for the selected criteria are available along with a high level summary of the number of items in each status
/BTI/TE_RACTIVITY_EVENTS / /BTI/TE_RACT_EVENTS	This report can be used to show an audit of ActiveControl activities and events. This includes Imports (including import errors), Approvals, Rejections, Mark as Imported, Mark as Manually Applied, Added to Import Queue, Added to Control Point, etc. The data can be reported at transport level or task and project level as required
/BTI/TE_RIMPORT_AUDIT / /BTI/TE_RIMP_AUDIT	This report provides an audit of transport imports allowing imports carried out in ActiveControl to be compared to those performed in SAP / STMS so differences can be highlighted. It can be used find transports imported via SMTS that need to be updated in ActiveControl.
/BTI/TE_RANALYSIS_RUN_HISTORY / /BTI/TE_RANAL_HIST	When an analysis is performed the results of the analysis run are stored in the ActiveControl database. This provides a history of analysis issues along with the action taken by the user. The change analysis history report allows this history to be reported upon. It gives the full details of the analysis issues along with the action taken by the user (e.g. Approve anyway)
/BTI/TE_RTRANSPORT_LOCATIONS /BTI/TE_RTRAN_LOCN	This report provides transport and location information as follows: • Retention: For a selected period, the transports with import date/time and duration of “retention” in all relevant systems in the selected transport path • Object: For one or more objects, the transports imported in selected systems • Project: For a project, where the relevant transports are located • Owner: For one or more transport owners, where the relevant transports are located
/BTI/TE_RU001 / /BTI/TE_RCONFLCT_ANL	The conflict analysis report is used to report on conflict analysis issues during the merge process. It can be executed for a specific merge target and all conflicting transports are reported. Each transport can be double-clicked to view the object details
/BTI/TE_RTRANSPORT_IMPSTATUS / /BTI/TE_RTR_IMPSTAT	The Import Status report displays all transport requests for a given transport path and into which systems the transport has been imported. The report allows for you to specify whether you want the ActiveControl view of the transport request (i.e. whether ActiveControl believes the transport was imported) or via the TMS view which examines the transport logs to determine whether the transport was imported. Alternatively, both views can be combined. Each system in the transport path is displayed dynamically, and whether the transport has or hasn’t been imported into that given target is displayed.
/BTI/	The Transport Location report displays all Transport Forms belonging

TE_RTRANSPORT_LOCATION / /BTI/TE_RTR_LOCATION	to a particular Project or Transport path and where they current “sit” within the landscape.
/BTI/ TE_RMERGE_RECONCILIATION / /BTI/TE_RMERGE_RECON	The transport and merge reconciliation report is used to provide reconciliation from the original transport to all related merge transports along with an import status to show which systems the transport has been imported into. It provides a linkage into all merged systems so the progress of the original transport can be fully tracked.
/BTI/TE_RTRANSPORT_MERGE / /BTI/TE_RTR_MERGE	The Transport Merge report should be run during and just prior to the release of a project (or upgrade project) that is currently using the “Merge” process. During the merge process, Production Support transport requests are merged into the future release / project and this is either done manually or automatically. In either case, this report displays all Production Support transport requests and the corresponding merge request that they were merged into the project/release on. This report should be run during and just prior to the project/release to ensure that the Production Support transports have all been fully merged and that no regression will occur when the project is set live.
/BTI/ TE_RMERGE_CONFLICT_MGR	Used to manage merges and conflicts in dual track landscapes.
/BTI/ TE_RSYSTEM_COMPARISON / /BTI/TE_RSYSTEM_COMP	The System Comparison report allows you to compare two target systems to determine what differences exist between what has been imported into each. A break-down of the missing transport requests (and Tasks) is provided enabling you to take appropriate steps.
/BTI/TE_RANALYSIS_CONFIG / /BTI/TE_RANAL_CONFIG	The analysis configuration report provides a view of which analysis types are switched on for specific targets / locations in ActiveControl. The yellow icon shows where the analysis is switched on for the location and the green icon indicates where it’s mandatory. Clicking on the items will show a popup if there are any parameters entered.
/BTI/ TE_RTASK_ROLES_STATS_DETS /BTI/TE_RTSK_STAT	This report provides an overview of the roles and users assigned to the selected tasks and an overview of the changes to both deployment and planning statuses
/BTI/TE_ROPEN_TRANSPORTS /BTI/TE_ROPEN_TRANS	This report can be used to identify old/orphan unreleased transports that are present in development systems and haven’t been changed since a specified date. This is useful to aid in the clean-up activity of unrequired transports
/BTI/TE_RCHANGE_METRICS /BTI/TE_RCHG_METRICS	This report will give total number of transport imports (and associated tasks) into selected systems for a date / time range. It can be used to report metrics for systems and projects by priority, group, type, etc.
/BTI/ TE_RTEST_RESULTS_ENTRY /BTI/TE_RTEST_RESULT	This report can be used to give visibility to all test results entered for multiple Business Tasks or even a whole project. It will show negative and positive entries for each Business Task and any comments entered.
/BTI/TE_RDELEGATION_REPORT /BTI/TE_RDELEGATIONS	This report shows the details of all active delegations in ActiveControl.
/BTI/ TE_RTRANS_IMPSTATUS_ALL /BTI/TE_RTR_IMPPATHS	This report is an extension of the Import Status Report, but allows you to view transports over multiple paths. For the report to format the data properly, the Roles assigned to each Target must be consistent across the paths
/BTI/	This new utility can be run to remove all of the Transport Forms over

TE_RLOCATION_CLEARDOWN /BTI/TE_RTR_CLEARLOC	a certain age from any particular Control Point. Any Transport Forms that have been in the selected Control Point for more than the configured number of days are ‘completed’ and removed from the Path
/BTI/TE_RBUILD_LIST /BTI/ TE_BUILD_LIST	This report allows you to sequence all Transports and manual activities for a group of transports in a control point and to give estimated timings using previous import history. This report allows you to create an always up-to-date build list for project cut-overs
/BTI/TE_RISK_ANALYSIS_TR /BTI/ TE_RRISK_ANLSIS	This report is used to identify transports containing objects contained in a specific set of risk groups. This can be useful for enabling a retrospective analysis on changes made to critical objects / blacklists etc. It is based on current Risk Guard group / template locking functionality.

5.30. Domain Controller Utility Programs

ActiveControl includes a number of utility programs in the Domain Controller to help administer and operate the system.

These utilities are available only via the SAPGUI.

Report Name	Description
/BTI/TE_RUTASK_STATUS_SET	This program can be used to do a mass update of existing Business Tasks to specific Deployment and/or Planning statuses
/BTI/TE_RURL_STRING_REPL	This program can be used to do an en-masse update of string in URL links for AC attachments
/BTI/ TE_RUADD_APPROVERS_TO_CP	This program can be used to perform en-masse updates to Approvers assigned to ActiveControl Points.
/BTI/TE_MASSUPD_FLDS	This program can be used to perform en-masse updates to a custom field at Business Task or Transport Form level.
/BTI/TE_RMASS_ADD_TO_CP	This program can be used to perform an en-masse addition of existing Transport Forms to a specific Control Point.
/BTI/TE_RARCHIVE_ENGINE	This program can be used to perform Archiving of ActiveControl data. More details on this program can be found here.
/BTI/TE_RTROBJ_REPORT	Report that be used to generate a list of objects contained within a set list of transports.
/BTI/ TE_RTARGET_SET_SCHED_FLAG	This program can be used to do an en-masse switch on/off of Schedules in defined Targets.
/BTI/TE_AUTOCREATE_BTTF	This program can be used to automatically create Business Tasks and Transport Forms for transports imported to specific SAP systems. There is other configuration required to use this functionality – please refer to online Forum for details.
/BTI/ TE_REVENTS_COFILE_RECON	This program can be used to revert transport sequencing to be based on initial date/time of import, in the event of accidental reimport
/BTI/TE_RMASS_TESTERS	This program can be used to perform an enmasse bulk upload of

	Tester assignments on Business Tasks.
/BTI/TE_RMANSTEPS_CHANGE	This program can be used to perform an en-masse bulk change of Manual Step user assignments.
/BTI/ TE_RBACKUP_DATA_EXP_NEW / /BTI/TE_UBACKUP_DATA	This utility program enables you to export all ActiveControl data to the file system. These files can then be imported into another Domain Controller system (via program /BTI/ TE_RBACKUP_DATA_IMP). This utility allows you to move your Domain Controller to another system. It can also be used to backup the ActiveControl data if desired or simply rebuild an ActiveControl playpen system.
BTI/TE_RBACKUP_DATA_IMP_NEW / /BTI/TE_UIMPORT_DATA	This utility program allows you to take the data exported from program /BTI/TE_RBACKUP_DATA_EXP and load these files into a new Domain Controller. This can be used for moving the Domain Controller to a new system or for restoring the Domain Controller from a backup copy.
/BTI/TE_RDATA_SYNCHRONISE	This program should only be used when two Domain Controller are running in parallel and need to be synchronised. Please consult Basis Technologies support for further information on the use of this program.
/BTI/TE_RNOTIFICATION_ENGINE / /BTI/TE_UEMAIL_NOTIF	The notification engine program sends email notifications for Approvals, Imported transports, etc. Further details about running this report can be found in earlier sections of the Administration Guide.
/BTI/TE_RSYSTEM_REBUILD / /BTI/ TE_USYS_REBUILD	If you ever rebuild a system (e.g. rebuild your test system from your production system), then this report will ensure that ActiveControl is brought back into synch. For example, if transport X was applied to the Test system but not Production and then the Test system was rebuilt from production, ActiveControl needs to be put back in synch so that transport X is added back to the import queue of the Test system). Important Note: Whenever a system is rebuilt, this utility program should always be run to get ActiveControl back in synch.
/BTI/TE_RTASK_UPLOAD / /BTI/ TE_UTASK_UPLOAD	This utility program can be run to upload one or more Tasks into the Domain Controller. The format is relatively straight forward and is described in the selection screen of the program. Data is uploaded from the front-end PC from a flat character-separated file (the character is a parameter). The program should be used during implementations of ActiveControl (during migration) or to take a feed of Task data from an external fault management software.
/BTI/TE_RTRANSPORT_FORM_EXP	Export Transport Form data to XML format to the file system (this program is superseded by /BTI/TE_RBACKUP_DATA_EXP).
/BTI/TE_RTRANSPORT_FORM_IMP	Imports Transport Form data from XML data into the Domain Controller (this program is superseded by /BTI/ TE_RBACKUP_DATA_IMP).
/BTI/ TE_RTRANSPORT_FORM_UPLOAD /BTI/TE_UTF_UPLOAD	This program is used during the migration phase of ActiveControl implementation. It allows “in-flight” transport requests (prior to ActiveControl implementation) to be added to the correct control point (i.e. normally an import queue) in the correct transport path. Once migrated to the correct control point, the ActiveControl change process can then proceed. Please refer to the program selection-screen for further details of the operation of this report.

/BTI/ TE_RERUN_BW_MERGE_NAMEFIX /BTI/TE_UBW_MERGE_PP	During BW/BI merges some objects are renamed based on the system they are being imported into. This means that the objects added to merge transport of copies also need to be renamed. If there are issues with the merge process this renaming can fail. Use this program to re-run the renaming process for the relevant merge transports
/BTI/ TE_REVENTS_COFILE_RECON / /BTI/TE_UEVENT_RECON	If transports are manually “Marked as imported” in ActiveControl the import date and time of the transport may not be based on the actual time and date in the transport logs. This can cause sequencing issues as ActiveControl uses this information when calculating the import sequence. This report can be used to update the import date and time to the actual values based on the SAP transport logs.
/BTI/TE_RU002 / /BTI/ TE_UASSIGN_ATTR	This report is used to update the transport attributes for the selected transports. This needs to be run in the relevant development system where the transports reside. This can be used after system refreshes / copies and when new projects are started to mark older transports as not relevant for the Conflict Analysis (see Housekeeping section of Administration Guide)
/BTI/TE_RU403 / /BTI/ TE_UUP_ORIG_SYS	This program is used to update the original system TADIR entries for all objects in a system to the current system. It can be used after project go-lives when objects may be still marked as owned by a different system. Only DDIC users are permitted to run this report
/BTI/TE_RU404	This program is used for caching of open transport requests for performance reasons. Please contact Basis Technologies support for further information on this program.
/BTI/ TE_RUTASK_STATUS_UPDATE	This program can be used to update the Planning and Deployment statuses of Business Tasks currently in ActiveControl. This should be run when configuration changes to the statuses have been made to set the statuses correctly for all existing Tasks

6. Deployment Considerations

In order to ensure a smooth deployment and adoption of ActiveControl, it may be helpful to give the topics detailed in this section particular attention during the implementation of the tool.

6.1. Windows Client and RFC

The ActiveControl Windows GUI client software communicates with the ActiveControl domain controller via SAP's remote function call (RFC) protocol.

RFC is a TCP/IP sockets-based protocol that uses TCP ports in the range 3300-3399.

The specific port number that is used will depend on the application server configuration of the SAP system that you select as the ActiveControl domain controller. If we assume that the chosen SAP system has only one application server with system number 10, then the TCP port number 3310 will be used. In this manner, the RFC protocol works in the same manner as the SAP GUI, except that it utilises the port number range 3300-3399 instead of the port number range 3200-3299. It follows that the network configuration must allow the necessary port numbers to be used.

6.2. E-mail Integration

ActiveControl relies upon SAP's support for SMTP integration with e-mail servers to deliver e-mail notifications, such as changes requiring approval, direct to e-mail clients such as Microsoft Outlook.

As e-mail notifications are sent from the ActiveControl domain controller, ActiveControl only requires this SAP system to have a working SMTP mail server connection. SAP systems based on Web Application 6.20 or later will find that the SAPconnect functionality of SAP comes with built-in support for SMTP mail integration. SAPconnect functionality is administered using SAP transaction SCOT.

In addition, the address data for each SAP user ID within the domain controller should be associated with an Internet e-mail address. If this is not the case, then the e-mail notification for that user will be delivered via the SAPoffice Business Workplace inside the ActiveControl domain controller.

6.3. TMS Configuration

ActiveControl does not require changes to existing TMS configuration; existing transport layers, transport groups and transport routes do not need to be changed as part of implementing ActiveControl.

Also please note that there is absolutely no requirement for the ActiveControl domain controller to be the same SAP system as the TMS domain controller – in fact a single ActiveControl domain controller can manage transports for multiple TMS domains.

In order to maintain tidy import queues within TMS, it can be useful to change the transport routes for

transportable changes to direct changes to a dummy virtual system. This is not a requirement, however it can prevent confusion caused by standard TMS functionality adding transport requests to the TMS import queue of recipient SAP systems in advance of when ActiveControl determines the changes ready to be imported.

If you wish to use ActiveControl Backout functionality, then you must define an additional virtual system called BAK within TMS. This is described in more detail in the Backout section of this Administration Guide.

Finally, if you wish to import changes into other clients of the same (development) system in which the changes were originally made, then you should enable TMS' extended transport control. This is not a requirement so much of ActiveControl, but is required to avoid the unnecessary attempt to re-import the client independent content (which would usually result in an import error).

6.4. User Exits

Numerous user exits points are available to enhance ActiveControl for specific customer requirements.

User Exit	Description
Validation during saving of a Transport Form **(Exit Identifier 0010)**	When a Transport Form is being saved by the user, this user-exit enables one to do whatever is necessary (for example validation of the input data).
Validation of a locked/unlocked Transport Form **(Exit Identifier 0011)**	When a Transport Form is locked or unlocked by the user, this user-exit enables one to perform any required function.
Transport Form creation (Map User) **(Exit Identifier 0012)**	Can be used to map users during Transport Form creation process (incase the userid is different in Development and Domain Controller)
Get extended task list item **(Exit Identifier 0020)**	When searching for a Task in the "Task Search" dialog, it is possible to retrieve other tasks from, for example, external systems in this user-exit.
Action Confirmation **(Exit Identifier 0030)**	Implement this exit to override whether a particular action within ActiveControl requires confirmation, as well as the behaviour of the confirmation dialog. Possible actions are: A = Approve a task (or set of tasks) I = Import a transport request (or set of transport requests) T = Save test results and finish testing Errors can be issued by setting the return structure Y_RETURN by calling function module /BTI/TE_RETURN_GET, passing in the message details: message type, message ID, message number and variables. Set parameter XY_CONFIRMATION_REQUIRED=X to force ActiveControl to ask the user to confirm their actions.
Validation during saving of a Task (Create or	This user-exit is called during the creation or changing of a Task (after the save button is selected). Refer to User Exit 0620 for a user-exit prior to display of the Task. The mode is passed through (Create or Change) as well as the values that have been entered.

Change) **(Exit Identifier 0040)**	Based upon this, the user-exit can issue a message to the user to prevent the creation/ change of the Task or add additional data behind the scenes. It should be noted that this user exit cannot be used to update custom fields.
Validate transport creation from task **(Exit Identifier 0041)**	This user exit can be used to validate the populated Transport Form from Business Task information.
Check for duplicate task reference during saving of a Task (Create or Change) **(Exit Identifier 0045)**	This user-exit is called during the creation or changing of a Task (after the save button is selected). A flag can be set to indicate whether the task reference field can accept duplicates values or whether it must be unique.
Dependency Check **(Exit Identifier 0050)**	This user exit is called during dependency checking.
Transport import **(Exit Identifier 0060)**	This user-exit is called during the transport import process. It allows specific actions to be carried out during the import process.
Change clients during import process **(Exit Identifier 0065)**	This user exit is called during the import process to override the clients used for client dependent imports. By default ActiveControl will import into all configured clients for the relevant target but this user exit allows the list to be adjusted if there are any special rules. E.g. only import transports originating in client 200 into client 210 and ignore clients 100 and 110
Non SAP Change Import **(Exit Identifier 0068)**	Can be used to deploy non-SAP changes from within ActiveControl
Merge request creation in participating systems **(Exit Identifier 0070)**	This user exit is called at the point where either a merge request or a backup request is to be created in a participating system. It is possible to override the standard functionality for creating these requests within this user exit.
Merge request processing **(Exit Identifier 0071)**	This user-exit is called during the processing of merge transports. This can be used to influence how the merge requests are created. For example, the standard implementation can be used to create a new merge transport for every source transport being merged.
Set form fields for merge **(Exit Identifier 0072)**	This user-exit can be used to set fields for auto merge form creation. It has been largely replaced by configuration options within the Windows GUI in later versions of ActiveControl
Task creation webservice preprocess **(Exit Identifier 0080)**	This user-exit can be used to perform webservice pre-processing during Business Task Creation. This needs to be activated to enable custom fields (e.g. CF_500) to be saved in the in the header structure for task creation
Task read webservice postprocess **(Exit Identifier 0081)**	This user-exit can be used to perform webservice post-processing during Business Task read

Task change webservice preprocess **(Exit Identifier 0082)**	This user-exit can be used to perform webservice post-processing during Business Task change. This needs to be activated to enable custom fields (e.g. CF_500) to be saved in the in the header structure for task change
Task test results webservice preprocess **(Exit Identifier 0083)**	This user-exit can be used to perform webservice pre-processing during Test Results entry
Task approval webservice preprocess **(Exit Identifier 0084)**	This user-exit can be used to perform webservice pre-processing during Business Task approval
Task get transports WS postprocess **(Exit Identifier 0085)**	This user-exit can be used to perform webservice pre-processing to get Transport list from Business Task
Integration conversions **(Exit Identifier 0090)**	This user-exit can be used to perform conversions as part of Integration
Integration notifications **(Exit Identifier 0091)**	This user-exit can be used to perform alternative notifications as part of Integration
Load User List **(Exit Identifier 0100)**	Implement this exit to override the list of users that will appear in ActiveControl when choosing administrators, approvers and managing the list of users whose transports are being displayed. The main user details like user ID, first / last name, email address, etc. can be returned.
Load Test User List **(Exit Identifier 0110)**	Implement this exit to override the list of users that will appear in ActiveControl when choosing test users within tasks. The main user details like user ID, first / last name, email address, etc. can be returned.
Before Transport Release: Runs in target **(Exit Identifier 0120)**	This exit can be used to perform target activity prior to transport release.
Extra validation in Inline Risk Analysis **(Exit Identifier 0130)**	This exit can be used to perform additional validation during the conflict analysis process.
Integration exit: Mapping before sending message **(Exit Identifier 0200)**	This exit can be used to perform mapping activity prior to sending Integration message.
RFC destinations **(Exit Identifier 0310)**	This exit can be used to manipulate the name of the RFC destination that is used when connecting to remote systems. For example, it can be used to override the default destination based on the system ID to point to an alternative system.
Modify list of approvers **(Exit**	This exit can be used to modify list of approvers for web gui and notification engine, prior to Delegation.

Identifier 0500)	
Filter transports for approval **(Exit Identifier 0505)**	This exit can be used to filter list of transports awaiting approval.
Pre-population of emails during notification **(Exit Identifier 0510)**	This exit is called during the notification engine just prior to any emails being sent. This means that all email details that have been derived, the subject, message body and recipients are provided to the user-exit for alteration if desired.
Post-population of emails during notification **(Exit Identifier 0520)**	This exit is called during the notification engine but after each email has been sent. During this exit you can send further emails if desired.
Pre-population of a Transport Form **(Exit Identifier 0610)**	When a Transport Form is initially created, this exit is called prior to the Transport Form appearing. This enables the default values of the Transport Form to be pre-populated. It also enables attachments to be automatically added as well as dependencies to other transport requests.
Pre-population of a Task **(Exit Identifier 0620)**	When a Task is either initially created or changed, this exit is called prior to the Task form appearing. All fields of the Task can be defaulted or changed prior to display of the form. More importantly, both the default set of Testers and the default set of attachments can be pre-populated.
Pre-population of a Test Result **(Exit Identifier 0630)**	When a Test Result (via a Task and a Test Queue) is initially created, this exit is called prior to the Test Result appearing. The Test result has only 2 fields that can be pre-populated (Test Result and Test Description). More importantly, however, you are also able to attach file templates within this exit. Hence, attachments such as Test Documentation can be added to the Test Result automatically upon entry of the Test Result. The user then has the option to either populate the template or to remove it (if required). Please note that the context of the Test Result (i.e. the Test Queue control point) is also passed to this user-exit along with the Task details so that the behaviour can be altered dependent upon this context (e.g. one template for Integration Testing and another for User Acceptance Testing).
Test Result validation when saving **(Exit Identifier 0635)**	This user exit can be used to validate the entered Test Results during save process.
Task Search Results Filtering **(Exit Identifier 0640)**	This user exit is processed during the task search process. It can be used to further filter the list of tasks returned by the search to build in any customer specific checks. For example, it could be used to additionally search for the entered text within custom fields.
Transport Request Travel **(Exit Identifier 0710)**	When a transport request moves from one location (e.g. Inbox) to another location (e.g. Import Queue) it is possible to manipulate the movement of the transport request within this user-exit. In particular, one is able to “skip” control points and even entire systems if desired (e.g. if the transport request is of a particular type). It is also possible to release the transport request automatically after movement into a particular control point (e.g. after approval is given and the transport request moves into an import queue of a given system then perform the auto-release).
End of Transport Request Travel **(Exit Identifier 0720)**	This user exit is called at the end of the transport path travel function module. By this stage the location of the Transport Form will have been updated and this can be used, for example, to trigger the creation / sending of ActiveControl notifications.
SAP Transport	This user exit is called during SAP transport request creation. It can be used to set

Request Creation **(Exit Identifier 0730)**	defaults for the transport request details like description, owner, target and project.
SAP Transport Request Approval, before Travel **(Exit Identifier 0740)**	This user exit is called during approval, before the actual travel takes place.
Merge Request Creation **(Exit Identifier 0800)**	This user exit is called during the creation of merge request. Note that this user exit is deprecated in more recent versions of ActiveControl.
After Merge Request Creation **(Exit Identifier 0810)**	This user exit is called after the creation of merge request. It can be used to automatically create a Transport Form for the new merge transport.
Merge CTS+ Processing **(Exit Identifier 0820)**	This user exit is called during the processing of CTS+ transports. It can be used to obtain the timestamp for the CTS+ transport objects for use in overtake and conflict checks.
Post Import Processing **(Exit Identifier 0850)**	This user exit is called after transport import is complete. It can be used to carry out any required post processing activities on the imported transports or the created merge requests. For example, it could be used to perform the BW renaming on created BW merge requests.
Post Import Processing (single tp/client) **(Exit Identifier 0851)**	This user exit can be used for post processing activities on the imported transports, per single transport/client deployment.
Post Import Processing (unconditional) **(Exit Identifier 0852)**	This user exit can be used for post processing activities on the imported transports.
TOC Creation **(Exit Identifier 0855)**	This user exit can be used to create Transport Form for Transport of Copies.
Add to Import Queue (Validation) **(Exit Identifier 0860)**	This user exit can be used to validate the allowed Import Queue locations whilst trying to perform an Add to Import Queue action in the Windows GUI or Web UI.
Add to Control Point (Validation) **(Exit Identifier 0861)**	This user exit can be used to validate the allowed Add to Control Point locations whilst trying to perform an Add to Control Point action in the Windows GUI or Web UI.
Add to Import Queue (Change Locations) **(Exit Identifier 0862)**	This user exit can be used to alter the list of locations you can choose whilst trying to perform an Add to Import Queue in the Windows GUI or Web UI.
Add to Control Point (Change Locations) **(Exit Identifier 0863)**	This user exit can be used to alter the list of locations you can choose whilst trying to perform an Add to Control Point Queue in the Windows GUI or Web UI.

User Exit for global check in General Analysis **(Exit Identifier 0900)**	Used to set the global check flag during the general analysis. Setting the global check flag will mean that the general analysis will look at all transports in the path whether created locally or imported from other systems.
RFC client destination for conflict analysis **(Exit Identifier 0910)**	This user exit is called during the conflict analysis process to override the RFC destination and therefore the client used. By default the client in the merge target will be used for the conflict analysis but it can be overridden to check a different client if required but using a different RFC destination. E.g. check client 100 for conflicts with transports originating from this client but check client 200 if the transport originates from there
Web UI News feed filtering **(Exit Identifier 0920)**	This user exit can be used to restrict and filter the news and recent events items returned for the logged in user so they don't see news relating to all users.
Link creation validation **(Exit Identifier 0930)**	This user exit can be used to validate link creation.
Priority Schedule Filter **(Exit Identifier 0940)**	This user exit can be used to perform a filter on Priority Schedule.
Overtake Analysis Filter **(Exit Identifier 0950)**	This user exit can be used to perform a filter during Overtake analysis process.
Custom Field updates on Transport Form save **(Exit Identifier 0960)**	This user exit can be used to update custom fields on Transport Form when saving the Transport Form.
Custom Field updates on Transport Form save **(Exit Identifier 0961)**	This user exit can be used to set custom fields to read/write based on any conditions.
Custom Field updates on Transport release **(Exit Identifier 0970)**	This user exit can be used to update custom field on transport release (if Transport Form already exists). This is called after is called after transport release (class /BTI/CL_IM_TE_BADI_TR_REL).
Exclude objects from analysis **(Exit Identifier 0980)**	This user exit can be used to exclude unnecessary objects from an analysis

User Exit Implementation

If a user exit is to be implemented, simply copy the sample function module which has a naming standard of **/BTI/TE_EXIT_SAMPLE_XXXX** (where XXXX = 4 digit user exit number) into the customer name space. This copies across the signature of the function module (Importing, Exporting, Changing,

Tables and Exception parameters). The copied function module should then be amended to the client's specific requirement.

Once implemented, an entry must be added to table **/BTI/TE_EXITC** in the ActiveControl Domain Controller. A corresponding entry may also need to be added to /BTI/TE_EXIT if it does not already exist and has been shipped.

The entry in /BTI/TE_EXITC requires:

Field	Description
EXITID	This is the User Exit identifier such as 0010, 0020, etc.
COUNTER	This is just the sequence number, starting from 01. However, if the user-exit allows multiple exits to be defined, then this can be incremented for each implemented user-exit (i.e. 01, 02, 03, etc).
EXITFUNC	The function module name defined in the customer namespace (e.g. ZABC_TE_EXIT_0010).

! Please note the that the following can have detrimental impacts to wider ActiveControl and should be avoided:
(i) A non active exit added to table /BTI/TE_EXITC.
(ii) A non active exit created in a Function Group which has an existing FM in use.

6.5. Windows GUI Command Line Parameters

The following command line parameters are currently available with the Windows GUI:

1. Start the ActiveControl GUI at a particular Target and Location

- ActiveControl.exe /TARGET: /LOCATION: * TARGETID = 4 digit target identifier as per entry in /BTI/TE_TARG * LOCATION = 1 character control point (Q=Import Queue, T=Test Queue, I=Inbox, O=Outbox)

2. Open a specified Task upon startup of the Windows GUI

- ActiveControl.exe /TASK: * TASKID = Internal Task Identifier as per entry in /BTI/TE_TASK

6.6. Language Support

ActiveControl default language is English, however it also includes an out of the box German translation and also offers the capability to setup the tool for other languages if required.

The language is determined by the language settings in the user's SAP system.

Setting up translation within ActiveControl to a language other than English or German is a technical exercise that will require Basis Technologies involvement. If this is something you are interested in, please contact your Basis Technologies Account Manager.

7. Housekeeping Activities

7.1. Recommended Housekeeping Activities

The performance of some functionality within ActiveControl can inevitably slow down over time, particularly in organisations delivering large volumes of transports through their SAP landscape. To avoid performance degradation over time, Basis Technologies recommend some simple housekeeping activities be performed periodically within ActiveControl.

1. Perform periodic Data Archiving

Data archiving can be performed to remove old data from the system. See proceeding “Data Archiving” section for more details on this process.

2. Review / remove non-required ‘in-flight’ transports

SAP best practice is always to move every transport through the entire SAP landscape to keep every system in sync, however the reality is that almost all SAP customers do not always do this. Basis Technologies therefore recommend a regular review of transports that have been stopped half-way through your landscape, and if it certain that they will never be moved further through the SAP landscape, to consider deleting them from ActiveControl.

Step	Activity
1	Run the Transport Locn by Date/Obj/Project/Ownr report. On the report selection criteria screen, enter wide start/end dates to ensure you capture all transports.
2	In the report output screen, sort the results by the [Time In (SID)] column.
3	Review the oldest transports sitting in each System, and verify if 1) whether they are valid changes that still need to be moved through landscape in the future or 2) whether they are never going to be moved.
4	For all transports that fall into the latter category and are never going to be moved, do a Look For / Locate for each transport within the Windows GUI and delete them from ActiveControl (using File > Delete (Just Remove The Items From the {Control Point}.

3. Set Ignore attributes on Transports

Certain ActiveControl analysis checks analyse all transports logged within ActiveControl, including those that have already reached your Production system. To avoid this becoming a performance issue over time, it is possible to set a couple of “IGNORE” flag attributes on transports so that they are excluded from specific ActiveControl Analysis Checks.
Basis Technologies recommend doing this activity periodically on transports that have been moved through your entire landscape, to avoid the analysis checks slowing down.

3.1 Attribute: YBT_TE_IGNORE

Can be used to remove transports from the Conflict Analysis check.

Scenario	Recommended Activity
Single Track Landscape	Whenever a transport reaches Production, run Program /BTI/TE_RU002 in each of your development systems to assign attribute YBT_TE_IGNORE to all the transports included in that Project release so they are no longer considered during conflict analysis.
Multi-Track Landscape	Whenever a Project Release reaches Production, run Program /BTI/TE_RU002 in each of your development systems to assign attribute YBT_TE_IGNORE to all the transports included in that Project release so they are no longer considered during conflict analysis.

3.2 Attribute: YBT_TE_OVERTAKE_IGNORE

Can be used to remove transports from the Overtake/Regression checks.

Step	Activity
1	Within the Windows GUI, locate the oldest ‘in-flight’ transport (i.e. the one with the earliest release date/time) in a system landscape, by manually checking each Control Point in in the path.
2	Any transport that has reached Production that has a release date/time older than this in-flight transport can have the OVERTAKE_IGNORE attributes set since there is no risk of an overtake/ regression issue.
3	Run Transport Activity Report in the Windows GUI Dashboard to identify all transports that have been imported to Production already. Sort by Release Date to identify all the transports that were released before the oldest in-flight transport that you identified in previous Step.
4	Run program /BTI/TE_RU002 in each of your Development systems to assign attribute YBT_TE_OVERTAKE_IGNORE to all the transports identified in previous step, so they are no longer included in overtake/regression analysis check calculations.
5	Repeat for each of your SAP system landscapes

Note: Attribute EXPTIMESTAMP must exist in the development system in order for the attribute update to be successful. If this doesn’t exist, run RSWBO006 and copy the settings for EXPORT_TIMESTAMP to EXPTIMESTAMP.

4. Review / remove unnecessary Open transports

Unless ActiveControl has been specifically configured to prevent it, your organisation may have a gradual build-up of open transports in your Development systems which have either been forgotten about or deliberately not been released and moved through your SAP landscape.

Step	Activity
1	Run the Open Transports by System & Change Date report to capture a list of all historical open transports.
2	Extract this list and review with the Transport Owners, to validate if the transports are likely to be progressed in the future.
3	For the transports that are not expected to be progressed, request that the Transport Owners delete them from SAP. For the ones that already have a corresponding Transport Form, also delete them from ActiveControl (using File > Delete (Permanently Delete…) in the Windows GUI.

5. Do not configure unnecessary Production Outbox

Some ActiveControl customers set-up a Production outbox to act as an easy way of viewing all transports that have gone to Production. The downside of this is that none of the Tasks to which these Transports are assigned are ever marked as “Completed” within ActiveControl. Basis Technologies would recommend not to have a Production outbox unnecessarily, and if one already exists and is not being used as a control point to a subsequent target system, to approve everything out of it.

6. Hide all inactive SAP Paths

If a SAP landscape path is no longer actively being used, deactivate it within ActiveControl so that any transports that only exist in that path are not included in the ActiveControl Analysis Checks.

Step	Activity*
1	Within Windows GUI Configuration screen, change the status of any transport paths not being used to “Cannot be assigned to new transports, not visible within ActiveControl”

7.2. ActiveControl Data Archiving

ActiveControl includes a data archiving capability to ensure that the performance of the tool does not deteriorate over time due to data constraints.

This can be used to archive historical Business Tasks, Transport Forms, import runs and analysis history data.

Archiving is performed by ActiveControl Administrators in the backend of the ActiveControl Domain Controller, via program **/BTI/TE_RARCHIVE_ENGINE**

This program can be used to archive as well as restore data. It can also be run in test mode where required.

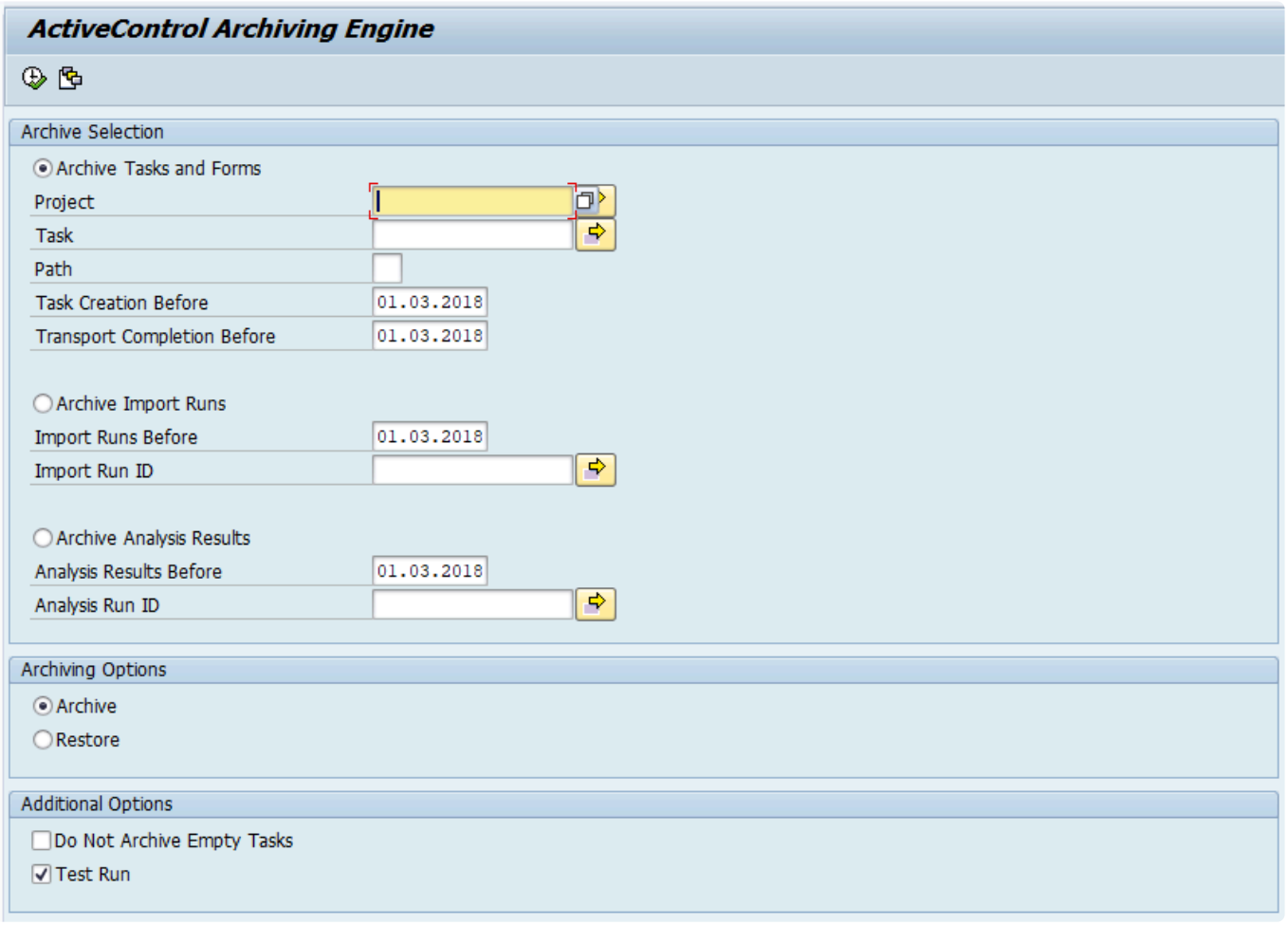


Figure: ActiveControl's Archive Engine

If there are events found for a particular import run, in table /BTI/TE_EVENTS then the import run can not be archived.
Message :- Import Run 00119022600000001861 is not achievable: has existing events found in database.

If there is a form available for an import run, in table /BTI/TE_FORM then import run can not be archived.
Message :- Import Run 00119022600000001861 is not achievable: DEVK971882 has been found in database.

If no data form, merger form, no events, no links in /bti/te_link are found for an import run, then we can Archive the import run.

Reporting on archived data

It is possible to run relevant reports relating to Activities, Events and Test Results on archived data.

This is done via the aforementioned reports, by making use of the options on the report selection screen.

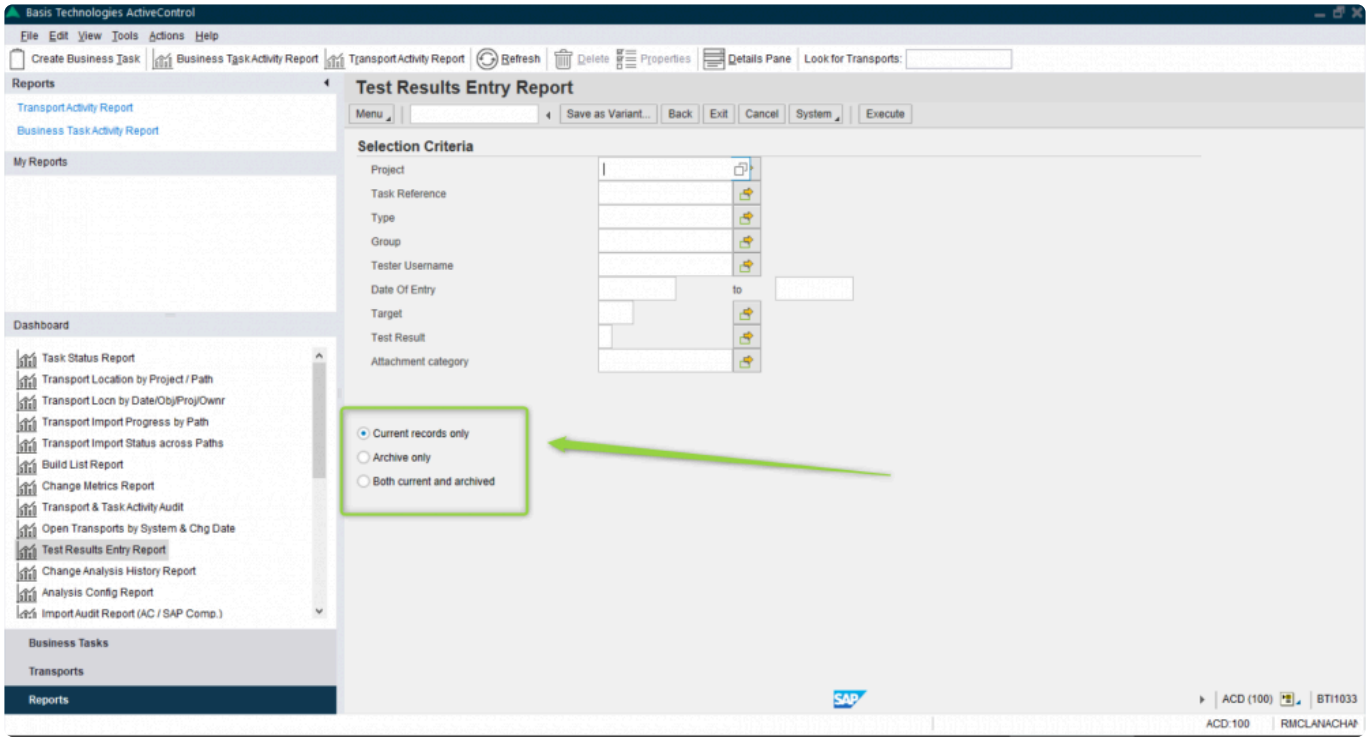


Figure: Running ActiveControl Reports on archived data

8. Upgrading ActiveControl

Upgrading to the latest version of ActiveControl is a relatively straightforward activity. This section summarises the key considerations and the steps involved in performing such an upgrade. The information contained herein is version-agnostic and aims to act as a record of the high-level steps of upgrading ActiveControl, not in configuring new functionality contained within the latest release of the product.

Basis Technologies strongly recommend that an ActiveControl upgrade is performed with assistance from one of our Solution Specialists. This not only ensures that the benefits of the newer functionality can be more efficiently reaped, it also helps ensure that any teething issues and questions encountered during the Upgrade process can be addressed in a timely fashion. If you are interested in upgrading to latest ActiveControl, please contact your Basis Technologies' Account Manager to discuss next steps.

8.1. Upgrade Considerations

The following should be considered as part of an upgrade to latest version of ActiveControl:

1. Latest software transports will need to be applied in the Domain Controller plus all satellite SAP systems.

2. ActiveControl is NOT backwards compatible because of the nature of some of the new functionalities. This means you have to apply the new Transports in the Domain Controller and all the satellite systems (and deploy the new version of the Windows GUI and Web UI) at the same time.

3. New version of the Windows GUI will need to be rolled out to all users. Again, this is not backwards/ forwards compatible. Users will not be able to use the old version of the GUI after the SAP systems have been upgraded.

4. New version of the Web UI will need to be rolled out to all users

5. A new License Key will be required, this needs to be uploaded into the new version of the Windows GUI after the Upgrade.

6. New Authorisations roles will need to be applied to all users. If copies of roles were taken previously, these will need to be updated as part of the Upgrade.

7. Performing a AC Upgrade has no impact on in-flight customer transports. You won't have to re-add any in-flight transports into ActiveControl after the upgrade, they will all still be sitting in the same Inbox, Test Queue etc that they were sitting in before the Upgrade.

8. Similarly, existing ActiveControl configuration will be similarly unaffected by performing an Upgrade.

9. Any existing custom user exits will need to be analysed for potential upgrade impacts, prior to the Upgrade.

10. Please note that if you are an existing Mass Data Runtime (MDR) / Diffuser user, this will need to be carefully managed during the upgrade. Productive use of MDR / Diffuser can be negatively impacted if an Upgrade to ActiveControl is done in isolation of it.

8.2. Upgrade Steps

The key steps to perform an upgrade of ActiveControl are summarised below:

Step	Activity	Other Information
01	Provision of ActiveControl Software + Software Links	To request the latest version of ActiveControl, please email your Basis Technologies Account Manager
02	Delivery of License Key by Basis Technologies	Once you have received the software, you can email support@basistechnologies.com to request a license key. You should provide the following information in your request : 1) Customer name 2) Domain Controller System ID 3) Domain Controller Installation Number. 4) ActiveControl version. To make things easier, please include a screenshot of the License screen from the Windows GUI (via Help>License Key dropdown option)
03	Verify if existing MDR/Diffuser Customer	If you are an existing MDR/Diffuser customer, please ensure you inform Basis Technologies – as upgrading to the latest version of ActiveControl can have detrimental impact on MDR/Diffuser if not carefully planned.
04	Delivery of Health Check by Basis Technologies	
05	Review of current user exits and enhancements, and impacts of upgrade	Please review table /BTI/TE_EXITC in the Domain Controller. If you have any non-sample User Exits defined in this table, please copy the contents of the Function Module code from SE37 and email it to Support.
06	Review of current ActiveControl roles usage – and impacts of new ActiveControl version on existing roles	Refer to Roles Matrix that should have been provided to you with the new ActiveControl software.
07	Development DC: Take Backup of AC Config and AC Data in Dev DC	Refer to online FAQ for process
08	Development DC: Perform ActiveControl Upgrade	
08a	Development DC: Apply Transports – Domain Controller	Nb: this cannot be done via ActiveControl.
08b	Development DC: Apply Transports – Satellite Systems	Nb: this cannot be done via ActiveControl. Also note, you may wish to only update certain satellite system path for testing, or even use a Sandbox path for initial testing.

09	Install new Windows GUI	Note that it is not backwards/forwards compatible with the previous GUI
10	Install new License Key	The first person logging into Windows GUI after the Upgrade will need to upload the new License Key
11	Development DC: Test ActiveControl Upgrade	
12	Training (if required, depending on process changes)	
13	Communicate Change Freeze	
14	Production DC: Perform ActiveControl Upgrade	
14a	Production DC: Take Backup of Config and Data	Refer to online FAQ for process
14b	Production DC: Switch off any existing Import Schedules	This is done via the Windows GUI, via the Advanced Import toolbar to remove the existing Schedule, or via Configuration screen (on the Import Options tab within the Target). After doing this step, check via SM37 that the import jobs are not still running (they will all be called TE_SCHED*
14c	Production DC: Switch off Notification Jobs	You can delete the existing job via SAP GUI (SM37). The program is /BTI/TE_RNOTIFICATION_ENGINE.
14d	Production DC: Apply Transports – Domain Controller	Nb: this cannot be done via ActiveControl itself.
14e	Production DC: Apply Transports – Satellite Systems	Nb: this cannot be done via ActiveControl itself.
14f	Production DC: Deploy new ActiveControl Windows GUI	
14g	Install new License Key	The first person logging into Windows GUI after the Upgrade will need to upload the new License Key
14h	Production DC: Perform Roles Updates	If you took copies of the out-of-the-box ActiveControl roles during original implementation, you need to overwrite those roles with the latest ActiveControl roles
14i	Production DC: Configure all new functionality required	Refer to Release Notes for configuration steps
14j	Production DC: Switch on Notification Jobs	Reschedule program /BTI/TE_RNOTIFICATION_ENGINE via SM36, using your existing variant.
14k	Production DC: Switch on Import Schedules	Switch on the Import Schedules via the Windows GUI. Check in SM37 that they are released after doing so.
15	Production DC: Initial validation of the Upgraded system	
16	Production DC: Perform Archiving (if appropriate)	Refer to online FAQ for more details on the process

17	Communicate End of Change Freeze	

9. Software Support

9.1. Online FAQ Forum

Basis Technologies offer an online forum containing over 700 searchable **Frequently Asked Questions** relating to ActiveControl. These FAQs cover many of the common error / warning messages that might be experienced during usage, and also useful HOW TO guides to perform many of the common operations within the tool.

The online forum can be accessed via the following URL:

https://basistechnologies.zendesk.com/hc/en-us

We strongly encourage all our customers (in particular ActiveControl Administrators and Basis team) to register for accounts on our website, and actively make use of this forum. It not only helps our customers be more self-sufficient and resolve common issues quickly and without the need for Basis Technologies involvement, but also helps us understand the common challenges our customers are facing.

9.2. Support from Basis Technologies

Raising Support Tickets

In the event that the online ActiveControl documentation and FAQ forum is not able to address a question or issue that you have, ActiveControl Administrators can request support from Basis Technologies by sending an email to the following email address:

support@basistechnologies.com

Sending an email to this address will automatically create a ticket in Zendesk, the ticketing tool used by Basis Technologies.

To help our support team provide you with the best service with your issue, please include as much information as possible, with particular attention to the following:

- **Customer:** Include the name of the customer you are representing, it may not always be obvious from your email address
- **Product and Version:** Include the Basis Technologies product and version that you are operating that has the issue
- **System & Client:** The system and client where the issue/fault occurred and if it's a license key issue provide the SAP system installation number (it is always ten digits long)
- **Description:** A clear description of the problem and the steps to replicate the issue, with screen shots
- **Data:** Any master or transactional data objects associated with the issue.
- **Error Messages:** Details of any error or warning messages given including where applicable run

time errors, short dumps and error logs
- **User ID:** The User ID being used when the issue occurred
- **Authorisations:** Ensure transaction SU53 is run and results shared to help with authorisation issues
- **Contact Details:** Please include your own contact details in your email
- **Priority:** Reflect any high priority issues by including URGENT or HIGH PRIORITY at the start of the email subject

Support Escalation

If you have any concerns with the service you are getting from Basis Technologies support, or wish to escalate any high priority issues please email **supportescalation@basistechnologies.com**

Require additional Information or Services?

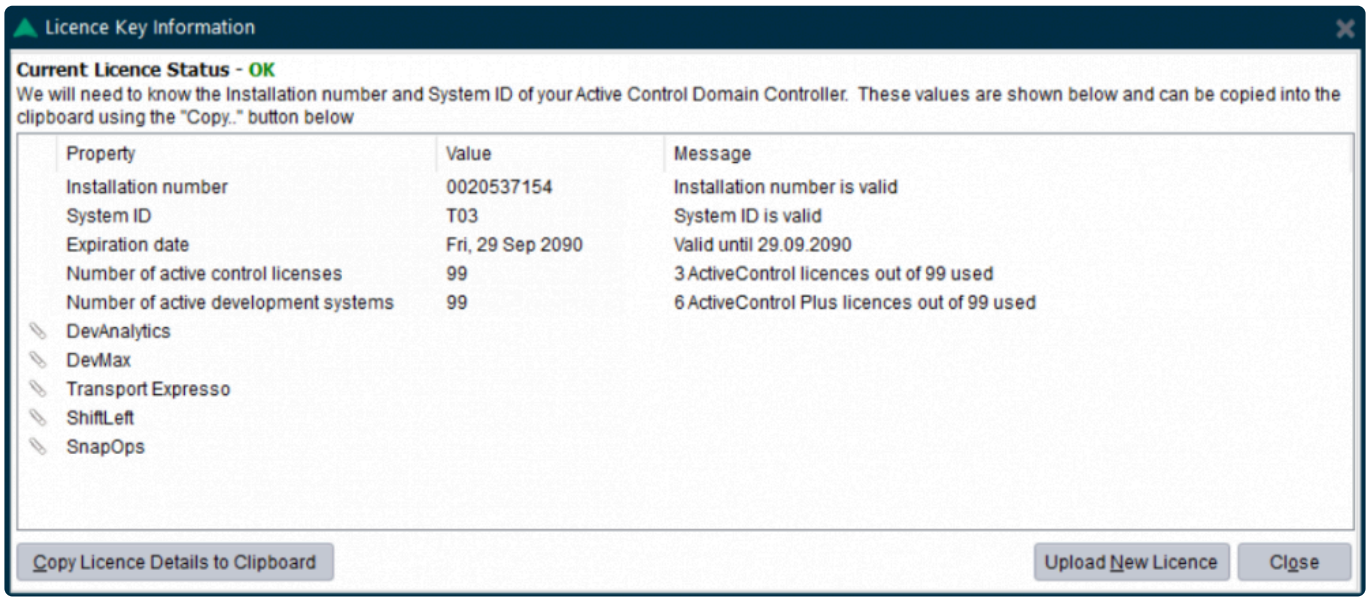
If additional information or services relating to any of Basis Technologies product sets is required, you can contact us via the above support@basistechnologies.com address, or alternatively by contacting your assigned Basis Technologies Account Director.

9.3. Updating ActiveControl License Keys

Basis Technologies typically issue a new ActiveControl license key to our customers as part of the annual renewals process.

This license key can be uploaded by ActiveControl Administrators via the Windows GUI. This is done via the [Help] dropdown > [License Key…] option.

Click on the ‘Upload New License’ button to upload the new key you have been sent.



9.4. Uninstalling ActiveControl

ActiveControl is installed in the central Domain Controller and all participating satellite SAP systems.

This section details the steps to remove the installation of ActiveControl from your system.

Deletion Transports

Deletion transports to uninstall ActiveControl in both the Domain Controller and participating satellite SAP systems can be provided by Basis Technologies upon request. Via the Add-On Manager, you can apply the deletion transports to remove all the objects associated with ActiveControl. All workbench objects will be removed via this approach and none are required to be manually handled. All ActiveControl related data and configuration will also be removed when uninstalling.

The only thing that will remain after applying the deletion transports will be the RFC connections and Users. These should be manually deactivated and removed, or retained if still operational for other purposes.

It is recommended that the uninstallation of any Production system objects is carried out according to your system maintenance schedule.

Uninstallation Timing & Sequence

There is no specific order in which to uninstall the ActiveControl add-on as there are no inter-system dependencies between the Domain Controller and all participating satellite SAP systems.

License Keys

Any license keys associated with your ActiveControl tool will become inactive at the time of uninstallation.

Dependencies

If you are operating other products from Basis Technologies, you should check with support@basistechnologies.com before uninstalling the Diffuser Framework.
Diffuser is separate from ActiveControl and could impact the usage of the other Products you have installed.

10. Legal Statement

All product names, logos, and brands are property of their respective owners. All company, product and service names mentioned in this ActiveControl Administration Guide are for identification purposes only. Use of these names, logos, and brands does not imply endorsement.

- BMC, BMC Remedy and Remedy are trademarks of BMC Software or its subsidiaries, registered or used in many jurisdictions worldwide.
- HP, HP Service Manager and HPSM are trademarks of Hewlett Packard Enterprise or its subsidiaries, registered or used in many jurisdictions worldwide.
- IBM and Virtualforge are trademarks of International Business Machines Corporation, registered or used in many jurisdictions worldwide.
- SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, Duet, Business ByDesign, ByDesign, PartnerEdge, HANA, S/4HANA, Fiori, SAP Business Suite and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world.
- ServiceNow is a trademark of ServiceNow, Inc, and is registered or used in many jurisdictions worldwide.

Other trademarks and trade names may be used in this document to refer to either the entities claiming the marks and/or names or their products and are the property of their respective owners. Basis Technologies disclaim proprietary interest in the marks and names of others.