RLINK

SAP/R3 PP-PI Interface

version 1.6

Build 1

most recent printing 6/07/2004

OSI SOFTWARE, INC.
777 Davis Street, Suite 250, San Leandro, CA 94577

How to Contact Us

Phone (510) 297-5800 (main number)
(510) 297-5828 (technical support)

Fax (510) 357-8136

Internet techsupport@osisoft.com

World Wide Web http://www.osisoft.com

Bulletin Board (510) 895-9423
Telebit WorldBlazer modem (Hayes, MNP, or PEP compatible)
8 data bits, 1 stop bit, no parity, up to 14400 baud
download protocols: Xmodem, Ymodem, Zmodem, Kermit

Mail OSI Software, Inc.
P.O. Box 727
San Leandro, CA 94577-0427
USA

OSI Software GmbH
Hauptstraβe 30
D-63674 Altenstadt 1
Deutschland

OSI Software, Ltd
P. O. Box 8256
Level One, 6-8 Nugent Street
Auckland 3, New Zealand

Table of Contents

Chapter 1
Overview

RLINK

RLINK provides a certified method of interfacing with the SAP R/3 Enterprise system. The drivers behind the RLINK product are

- Provide Integration of the Enterprise wide planning applications with the Industrial Desktop
- Reduce magnitude and complexity of production management and reporting
- Enable process engineers and operators to support core business functions
- Enhance the level of coordination between manufacturing, maintenance and logistics functions
- Provide understanding of integration issues and solutions from the point of view of the Industrial desktop.
- Integrate the enterprise portal with the manufacturing portal

Some of the benefits and services provided by OSIsoft's RLINK gateway to SAP's PP-PI module are listed below.

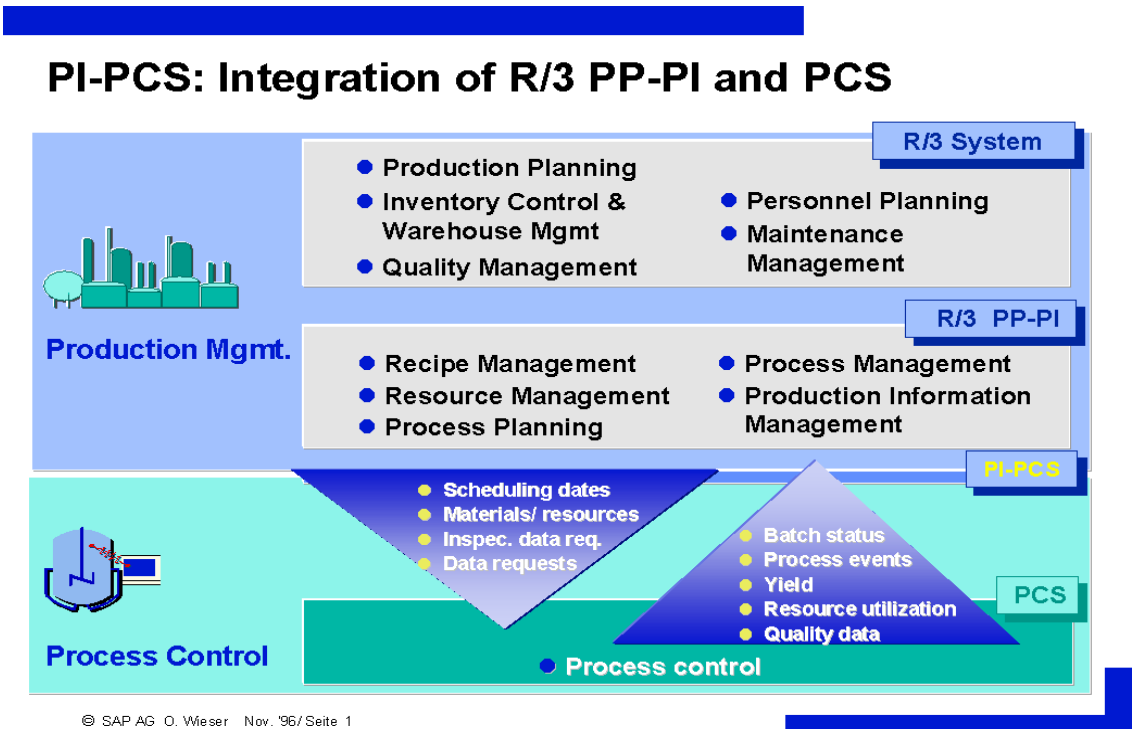
- Although SAP understands process orders, RLINK taps into the wealth of information that operational personnel understand about production
- RLINK correlates SAP process orders to plant floor orders and provides cross-references between them. This correlation is important for technical staff to support their customers after delivery.
- RLINK correlates quality and process measurements needed for analysis of a customer complaint or an order variance by relating the time stamp back into the process data.
- With RLINK there is no duplication of data entry because the automatic data transfer eliminates the manual entry errors.
- By using standard OSIsoft client tools as the user interface to SAP, training requirements are greatly reduced
- All production lines within a corporation can be compared and analyzed together to gain a better understanding of the grades or products that perform best with each resource. Process data from the PI System data archive can add

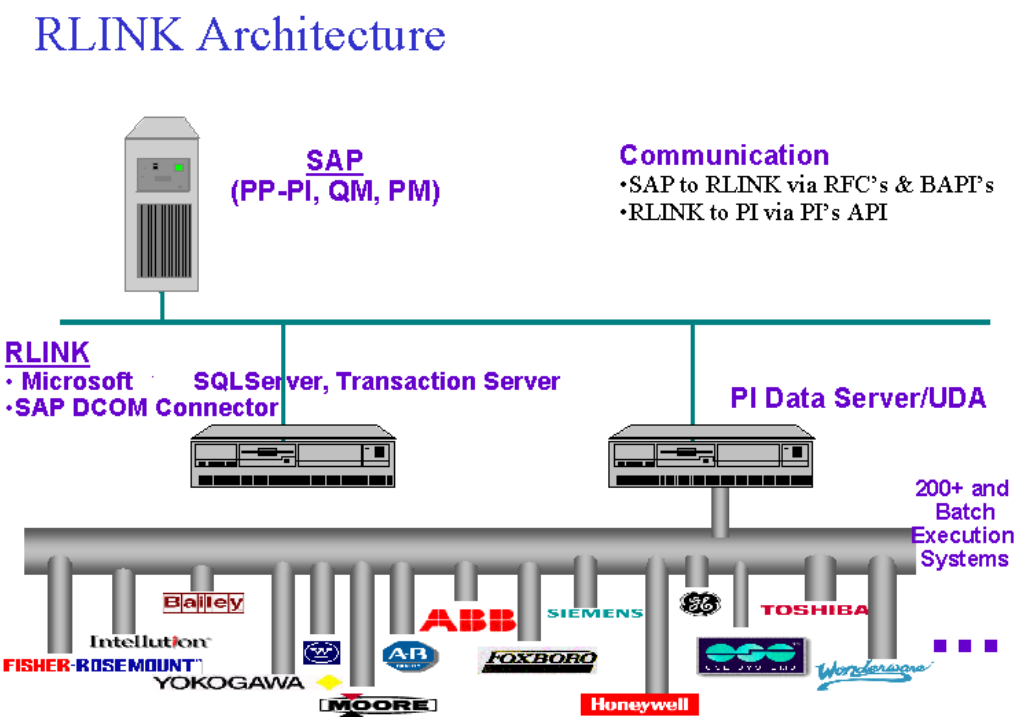
to that comparison to help engineers effectively perform analysis of the variances.

- Cost data is available in real-time (rather than month-end) to facilitate timely business decisions, such as operational efficiency analysis based on specific grade runs on various machines as well as sales decision support for future pricing.
- OSIsoft's RtPM Platform and RLINK enable customers to satisfy the business needs that will take the optimum advantage of SAP R/3 (e.g., material consumption, energy usage, asset utilization, etc.). This software also provides supportable, maintainable tools for management to analyze why a product was made to a given quality by correlating production data with business data.
- Using RLINK enables corporations to complete ambitious plant integration projects more rapidly with costs below projections. The RLINK/RtPM infrastructure creates new opportunities for process improvements and operational efficiencies that cannot be easily achieved with traditional cost accounting systems.

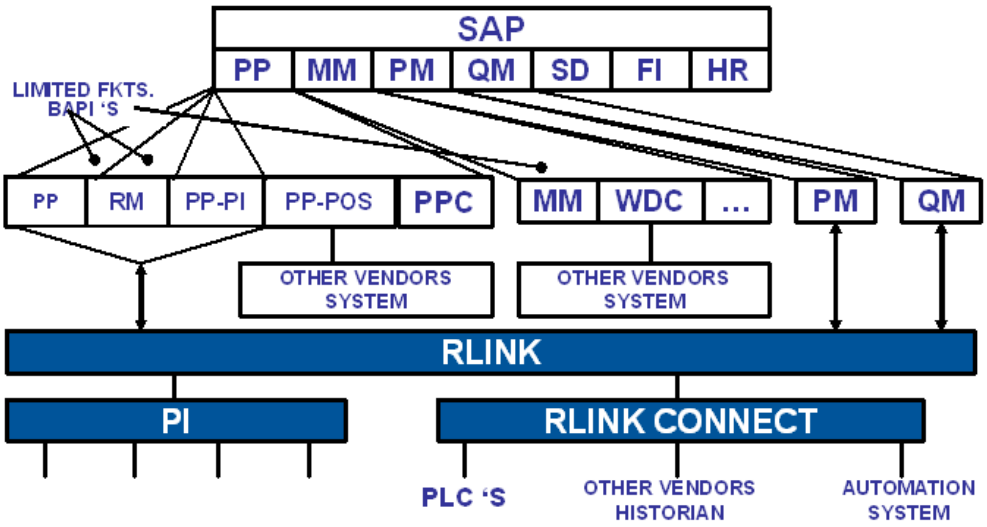
Accurate Inventory	Inventory Reorder Minimize Inventory Meet order requirements
Timely Data	Business Status in Real Time React to Business Issues Analysis in time to react or take advantage of opportunities
Automated Asset Availability	Focused Maintenance Expenditure Asset Availability Capital Utilization Reduced Costs
Quality Integration (on-line sensor integrate lab and process)	Compliance Reporting Process Improvements Timely Data
Visibility	Collaborative Enterprise Sarbanes-Oxley Customer Responsive
Traceability	Compliance Reporting
Product production variables	Cost of production, increased margins
Performance Improvement	Performance Improvements over all manufacturing

The RLINK product has three modules that correspond to the SAP R/3 modules PP-PI (Production Planning in Process Industries), PM (Plant Maintenance) and QM (Quality Management). This manual will cover the interface to PP-PI. The SAP certified RLINK gateway reduces enterprise integration costs. The result is a standard R/3 configuration that allows operations and management to leverage production information.

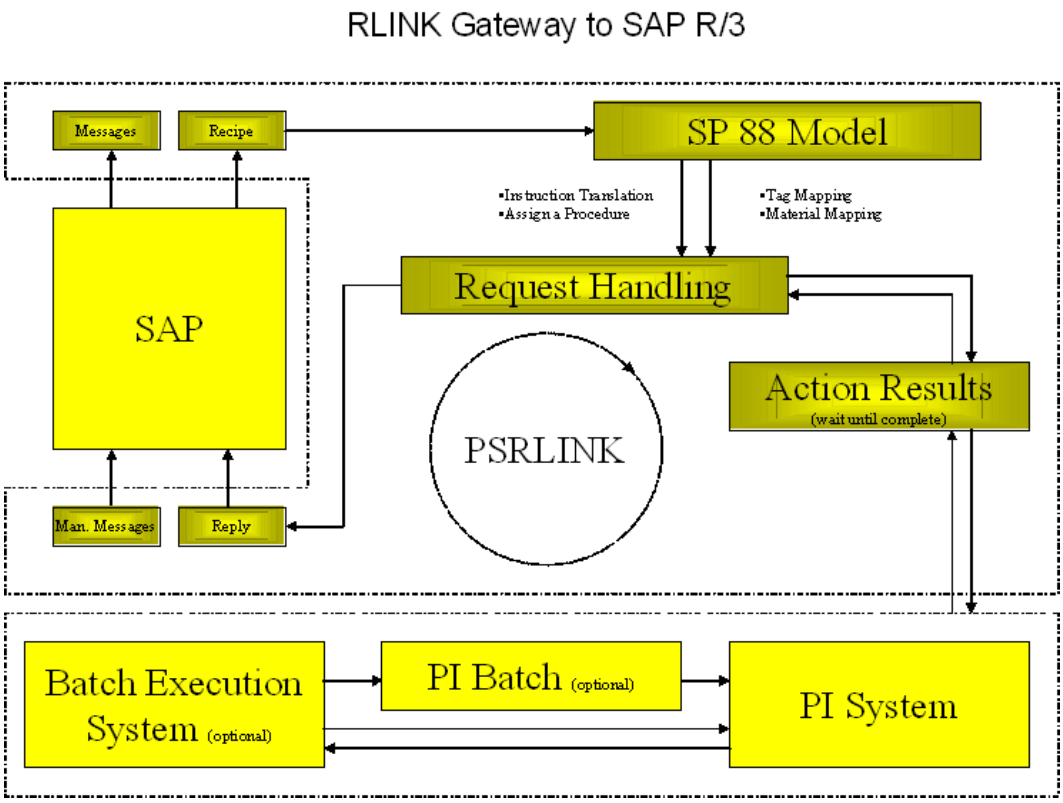




SAP Modules and RLINK



* INCLUDED IN PP- PI



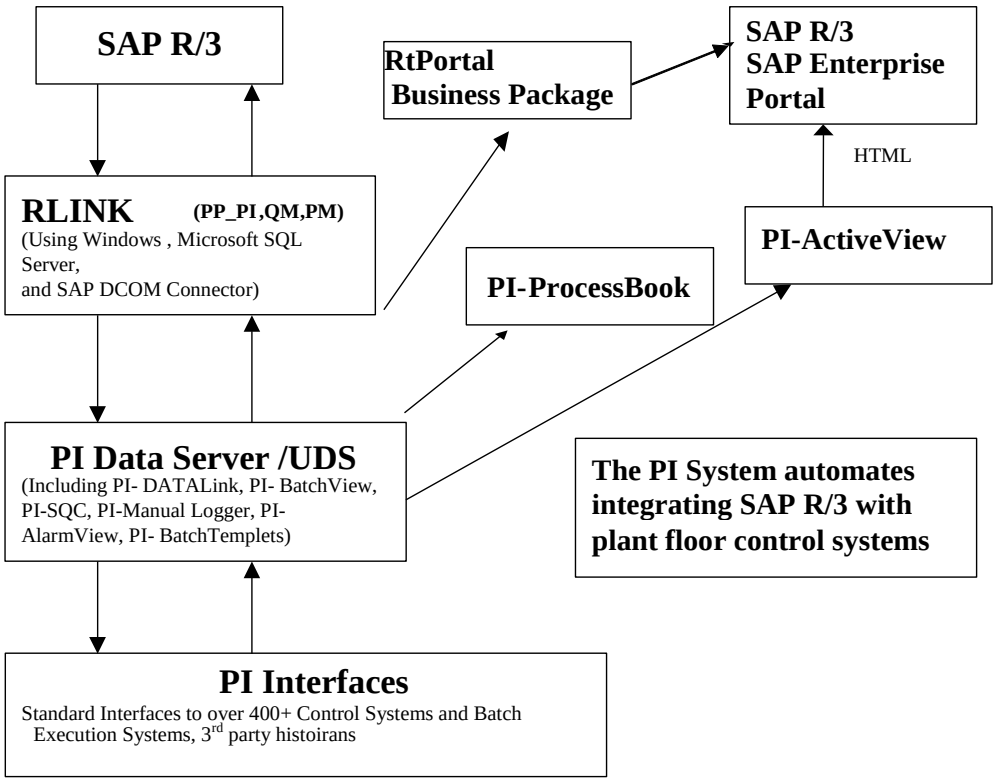
Getting Started Questions

- Number of plants to be handled and do they have PI systems and what version of PI.
- Is PP-PI installed or are you doing goods receipt and issue against PP?
- Is there a Batch Execution system and if so what product? Does it have any API or interface language?
- Is the plant continuous or batch?
- Do you want to store the recipe information in PI this could include the Process Order and for each material the quantity, batch_id, reservation and reservation_item?
- What are the number of materials that you make?
- Will you be doing resource changes after the recipe has been down loaded?
- Do need to store additional information with the batch in SAP, will the batch characteristic instruction be sufficient?
- Will you be passing quality information up through the recipe?
- If you need material batch_id's who will determine the product batch number the plant floor or SAP?
- Will backflushing be used to automatically post consumption data based on production?
- Will you use PP-PI to retrieve other activities for costing?

- If continuous what will be the time period of the recipe?
- Do you have both continuous and batch production in the same plant?
- If the plant is a batch plant and you do not have a batch execution system then what is the workflow for starting a recipe?
- Do you have co-products and by-products?
- What is the number of recipes?
- What is the volume of recipes per day?
- What is the average number of materials per recipe?
- What version of SAP are you using? Is there one SAP instance for the corporation?
- Are the plants in the same time zone?
- Does instrumentation exist to measure the quantities requested?
- Have recipes been drafted, what is the starting point?
- If the plant is a batch execution plant has the naming of materials and resources been coordinated with the naming in SAP or will you have to use the alias feature?
- Where will calculations be done, performance equations, executables, DCS or in RLINK?
- When reading a tag do you want interpolated, last value, will it be totalized, will you wait for the value to appear past the timestamp of the machine?
- How are material movements to be handled?
- What is your client platform Windows 95, 98, NT or 2000?
- What is the length of data to be kept in RLINK?
- Will you be implementing PM and QM modules in SAP, if so do you want to look at the RLINK modules to interface to these products?
- What training do you have in SAP? Do you have someone knowledgeable in PP-PI and the customization?
- What training do you have in PI? Do you have someone knowledgeable in tags configuration, PI-Batch, ProcessBook, VBA and DataLink?
- What training do you have in SQL databases? Do you have knowledge of Microsoft SQLServer?
- Does you control system handle string tags?
- Are you doing made to stock or made to order in SAP?
- Will you need to send batch characteristics to SAP?
- Are you doing time ticket or time event in SAP?

OSIsoft Products

RLINK-integrates the SAP R/3 enterprise system with the plant floor. Interfaces are provided for the PP-PI, QM and PM modules. RLINK provides a standard interface between all plant systems across all sites by leveraging the PI architecture that interfaces to 400+ control systems. RLINK runs on Microsoft Windows and uses Microsoft SQLServer. It is developed using RFC's (Remote Functions Calls) and BAPI's (Business Application Programming Interfaces) from SAP. The system includes extensive error handling and other functions designed for robustness and reliability.



Other OSIsoft Products which can be used in conjunction with this interface to provide greater functionality are as follows:

RtPortal Business Package – collection of iViews that bring manufacturing floor data into the SAP Enterprise Portal. This is a certified Business Package for the SAP Portal that includes schematics, trends, gauges, and iViews for alarms, functional locations, recipes, and quality data.

PI Data Server - is a time series database designed and optimized to quickly receive, store and retrieve time oriented manufacturing data. The database stores numerical and strings data in large quantities for extended periods. Support for Binary Large Objects (BLOBs) is also included. Data can be stored to a resolution of sub-second. A "swinging door compression" method allows PI to keep orders of magnitude more data on-line than conventional scanned systems. The archive also includes the ability to do performance equations, totalizer and alarming. The batch subsystem, **PI-BATCH** provides for the storage of records associated with batches. Interfaces are provided to 400+ control systems. There are also available **PI-ODBC** and **PI-OLEDB** interfaces.

ProcessBook - is the premier graphical user interface for the Plant floor. It provides a schematic and trending view into data that is enabled with VBA and event based

processing. ProcessBook is an ActiveX Control Container, thus allowing for the embedding of controls including live video of the process.

PI-Datalink - is used to generate and publish reports using Microsoft Excel or Lotus 1-2-3

PI-ActiveView - is used to create and view HTML pages that contain PI-ProcessBook data and displays. This allows for central management of web applications. This is the mechanism used to display the PI data is displayed in the SAP Process Cockpit.

PI-Batch View - allows for easy search and reporting on the batch database. The batch trend Add-In allows for analysis of batches. All SAP recipes and lots are stored as PI-Batches.

PI-SQC – automates the task of SQC calculations and charting of process history. Analysis of the best batch can be done in comparison to other production.

PI-Manual Logger - is used to log data that is not collected automatically from instruments and control systems. Data capture can be done via hand held terminals (HHT) or from terminal data entry. This provides a mechanism for combining this data with other process data for analysis and reporting.

PI-AlarmView - provides a view into the alarm log of displaying current alarms. Combined with the alarm server , that keeps track of the alarm history the PIAlarmView facilitates the user in detecting alarm patterns.

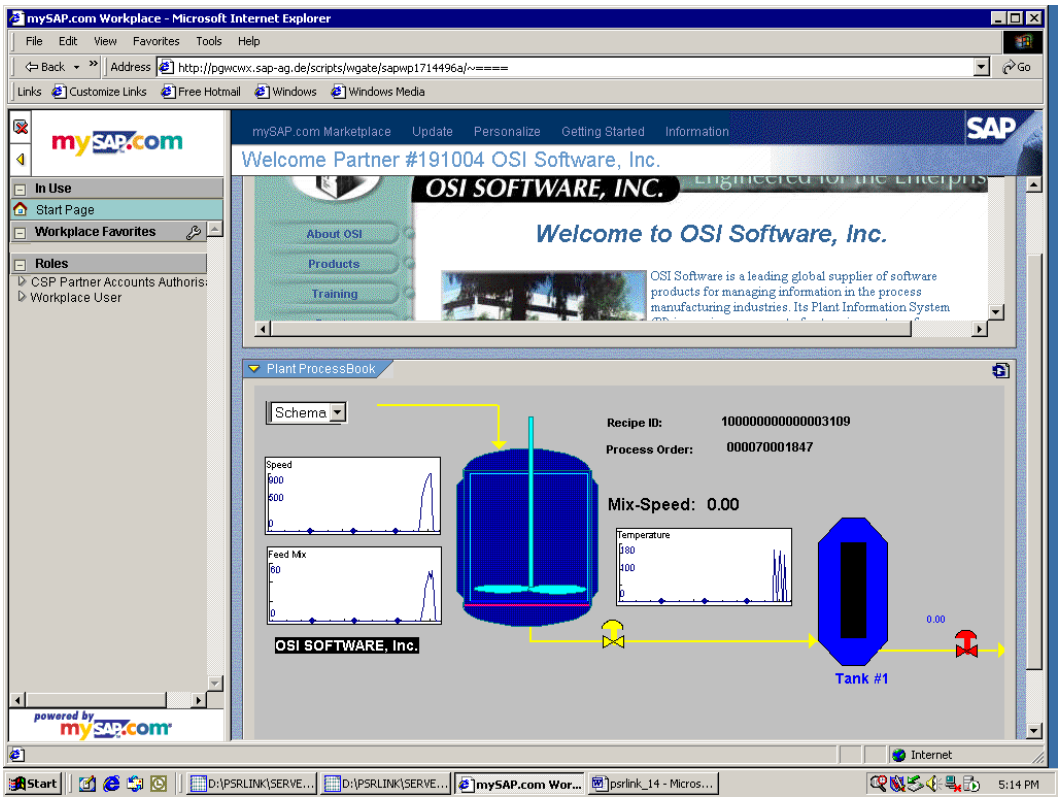
PI-UDS - Universal Data Server enables PI clients to access other data historians.

Interfaces – The Pi data archive has interface to 400+ plant floor systems (e.q. Allen-Bradley, Bailey, Foxboro, Fisher-Rosemount, Honeywell, Siemens, Intellution, Yokogawa, etc), lab systems, tank gauging systems and others.

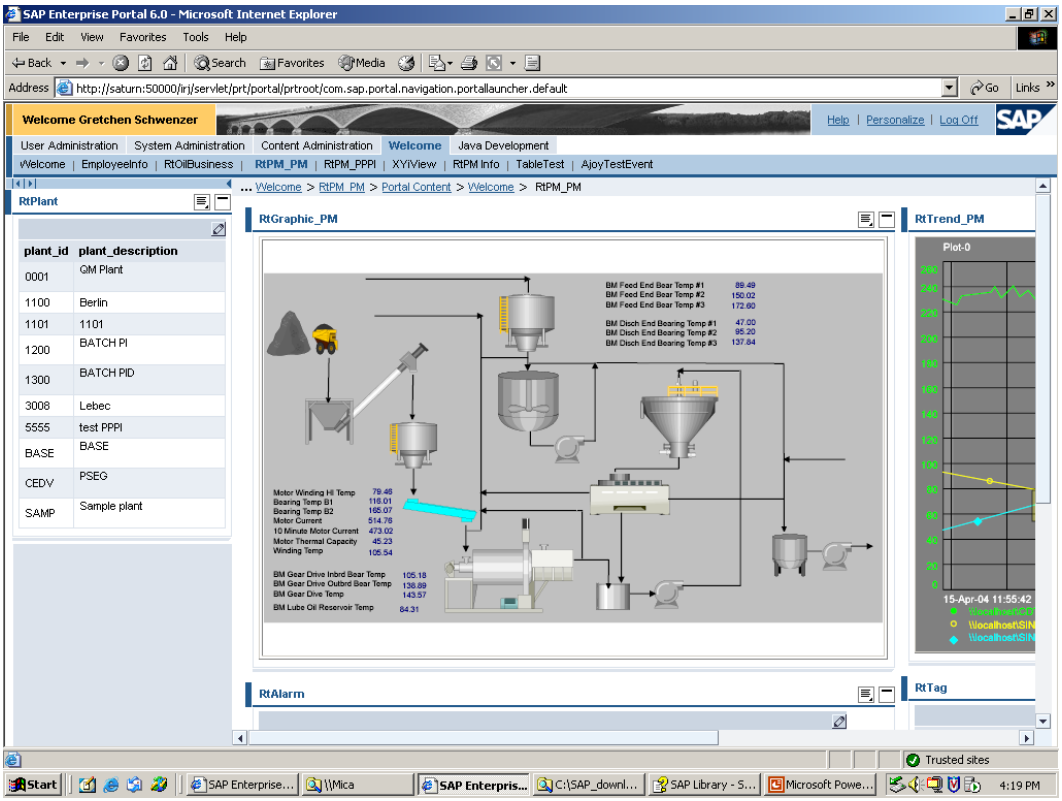
PI-ACE – Calculation engine that allows calculations to written combining PI data with any other data by programming in Visual Basic.

SAP R/3 mysap.com

ProcessBook and PI-ActiveView provide flexible viewing of plant floor data and easily integrate with the mysap.com strategy. ProcessBook displays that incorporate process, quality and equipment displays can be incorporated in the SAP mySAP.com by using the PI-ActiveView product. ProcessBook and PI-ActiveView are very useful for monitoring plant and product conditions and for displaying this data in a consistent, understandable, and visual environment in the SAP mySAP.com technology. This methodology can be used to give a looking glass into the plant from the Enterprise level.



The above graphic shows SAP R/3 recipe data and actual process data integrated in an easy to use ProcessBook graphic. Using PI-ActiveView live ProcessBook graphics can also be viewed in the SAP Process Cockpit.



The above graphic shows actual process data integrated in an the SAP Enterprise Portal using the RtPortal Business Package. This collection of iViews bridges the gap between the manufacturing and business parts of the enterprise.

SAP/R3 PP-PI Interface

PP-PI is Production Planning for Process Industries and is intended to be the link to the process information system. While PP-PI feeds data to the SAP/R3 modules for Production Orders, Costing based on material consumption and production and activity usage it does not handle all of the information available from the industrial desktop. For example finished quality data is handled through QM-IDI, Plant Maintenance and Inventory Transactions are yet other modules.

SAP/R3 provides a mechanism to communicate with their product thru RFC's (Remote Function Calls). By using this SAP/R3 supported methodology one is guaranteed compatibility with SAP/R3 future product enhancements and support of their strategic direction of support of Microsoft Technology.

Feature	***PlantSuite RLINK***
Certification	3.0-4.7 certified
Interface to control Vendors	Provided to all supported via PI
Language	C++, SQL
Platform	Windows NT, Supports SAP/R3 on any platform via SAP/R3 RFC's, Supports PI on any platform via PI-API
SAP/R3 Technology Used	RFC (Remote Function Call) and BAPI's
Microsoft Compatibility	Windows 2000, 2003 and Microsoft SQL server Technology
Translation Method Used	Stored procedures
Data Store	Microsoft SQL Server
Point to Point vs Multipoint	Multipoint Data Server
Push and/or Pull Recipes	Push and Pull of data from SAP/R3 supported
Customizing	User based customizing supported through stored procedures
Interfacing to other Products	If third party product has C callable interface data can be exchanged thru the SQLServer and the third party product.
Knowledge and Services	In-house Knowledge in use of SAP/R3 integration tool set and applications which impact the Industrial Desktop
Future Plans	Support of Additional SAP/R3 modules. Support for SAP XI. Support of BAPI additional BAPI's.

Features

- SAP/R3 RFC Function library for

Control Recipe Download (Transactional)

Control Recipe Pull (Transactional)

Control Recipe Available (Transactional)

Control Recipe Pull Single (Transactional)

Process Message Download (Transactional)

Process Message Upload (Synchronous and Transactional)

Download of the return code for message processing with tRFC (Transactional)

Download of detail data on characteristics (Synchronous)

Download of allowed characteristic values (Synchronous)

- SAP BAPI's

  Create Process messages

  Check process message existence

  Read process characteristics, include detail data

  Read allowed values for process characteristic

  Read control recipe list

  Request and receive control recipe

- Batch Program Execution Environment which can handle Executables and stored Procedures.
- SAP Material Movement BAPI's

  BAPI_GOODSMVT_CREATE

  BAPI_GOODSMVT_CANCEL

- SAP Repetitive Manufacturing

  RepManConfirmation1

  Cancel

  CreateMTO

  CreateMTP

  CreateMTS

  ExistenceCheck

- Data store based on SQL Server for Recipe and data requests thus maintaining history of request and response for auditing and data retrieval
- Database Purge Utility
- Translation of SAP/R3 download into data structure which represents the SP88 data model
- Generic method to handle data requests from Sources
- Library of routines to request data from PI
- Library of methods to translate SAP/R3 data requests into specifics required for data query of PI or other system
- User added translation routines and data collection routines feasible for interface to third party products and other PI functions.

- Can service multiple Industrial Desktop Programs requesting the same data of SAP/R3
- Message Request Triggers for Satisfaction of data requests
- Can support continuous and batch processes
- Checks for instruction completeness in recipe request before processing the recipe
- Uses standard Microsoft technology
- Recipe and Messages can be view via ProcessBook thus making one window into the process
- Timezone independence for PI Server

System Requirements

Item	*Version*
Server Machine	Windows 2000, 2003. NT 4.0 clusters tested, Windows 2000 clusters not tested.
Client Machine	Windows 2000, 2003 or XP
Microsoft SQL Server	2000
Microsoft Access	Windows 2000
SAPGUI	4.6D or 6.2 compliation 3
SAP	3.0D or greater
PI API	
PI SDK	
Ram > 64 megabytes	
Intel NT Server Machine	
DISK 2-3 GB	
PI	3.x if you want to store SAP/R3 recipe no in PI otherwise 2.x is sufficient
Webex accessible	Required for on-line support

Prerequisites

One member of the company team working on this project should have attended the following SAP/R3 courses: LO315

One person should be familiar with SAP/R3 Customizing.

One member of the company team should be proficient in working with PI and point configuration.

One member of the team should have general knowledge of databases and use of Microsoft tools.

Chapter 2
Installation

Preparation Prior to Install

The following information and material is required prior to an installation:

- An account on SAP must be provided which has privilege to do the SM59 transaction.
- The SAP account that is going to be used by RLINK should have the decimal delimiter set to "." .
- SAP/R3 version number which should be at least 3.0C
- A copy of the SAPGUI CD
- The IP address of the SAP/R3 application server, router and the gateway name
- In order to do a demonstration recipe at least 2 materials should be configured in SAP/R/3
- A licensed copy of Microsoft SQLServer, Microsoft Access and Microsoft Excel
- Either OSI's Datalink or Manual Logger for input of data values to PI for testing

PI Issues

- PI Login set to the correct location for the default PI system
- Security set on points that are to be written to so that the account chosen has access
- If you are installing on the same machine with PI then you must shutdown the PI services during in the install process. It can be restarted after the installation script has finished.
- PI-SDK version 1.1.0.142 or greater must be installed on the server and client machines. This is done automatically with the setup program for version 1.6 of RLINK PPPI.

BAPI's vs RFC calls

We support both the BAPI and RFC calls to SAP however when choosing which method that you want to implement you should consider the following.

- Ad-hoc messaged from SAP are downloaded only if you use the RFC function. SAP has not provided an equivalent BAPI.
- RFC establish continuous communications there is no logon and logoff happening continually and there is no polling if there is a recipe available.
- RFC recipe are pushed down with the BAPI you can only configure a recipe destination to be pulled.
- The BAPI for message upload gives you a mechanism to confirm that a message has been received by SAP.

Therefore the best recommended configuration is a combination of both. Use the BAPI for the message upload and use the RFC for the recipe and ad-hoc messages.

Set-up for SAP/R3

The following section gives menu paths in SAP. Since SAP changes these paths with each release you might find descrepencies between what is given in this document and the actual paths.

- Install the SAPGUI 4.D or greater. If you are running on a Windows 2000 there is a patch from SAP for the librfc32.dll that is referenced in SAP note 0370107. We have included this version of the librfc32.dll on the RLINKCD. When you replace the DLL it must be un-registered and re-registered. If your are using version 6.2 of the SAPGUI you must have at least compliation 3 (6203.3.22.953) the version of librfc32.dll (6203.3.480.3788).
- SAP/R3 SM59 transaction for destination

For the destination name chosen and which will be setup in the customizing section and entry will be made for an external TCP/IP system with this transaction. Note the destination name must not have any embedded spaces.

Select TCP/IP and Create. Enter the RFC Destination name, Set connection type to T, Trace to on, Enter Description which will appear in the menu list, Enter, Select the Register button, Enter the program name starting with the name of machine followed by tcrd.exe as follows:

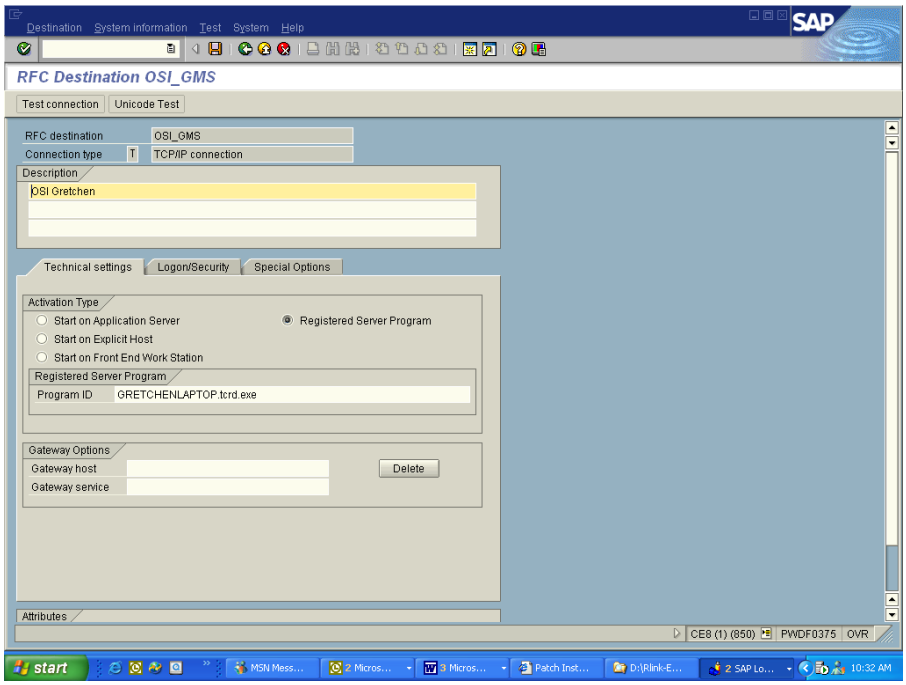
Machine_name.tcrd.exe

Save the configuration. This will be tested later.

Note: Depending on your installation of SAP/R3 it might be required to setup a Gateway on this transaction. You should try it first without the Gateway.

Note: If the SAP/R3 machine you are using supports multiple Gateways you will have to configure the Gateway option on the SM59 screen.

Note: With release 3.1G of the SAPGUI if you have turned trace on the SAP/R3 functions write to the DOS box of PSRLINK all information about the transfer. Therefore this should not be checked if you are using 3.1G.



- Host files must have been configured on SAP/R3 and Server machine

The TCP/IP address of the plant_suite Server must be setup on the SAP/R3 server. Likewise the plant_suite server machine must have the host address of the SAP/R3 machine in its host file.

- Destination in SAP/R3

Select Tools, Business Eng, Customizing, Implement, projects, SAP/R3 Ref.IMG, Production Planning for the Process Industries, process management, control recipes / PI sheets, set up control recipe destinations. Enter Plant name and then the dialog is given for creating a destination.

The following information will be requested when you select new entry.

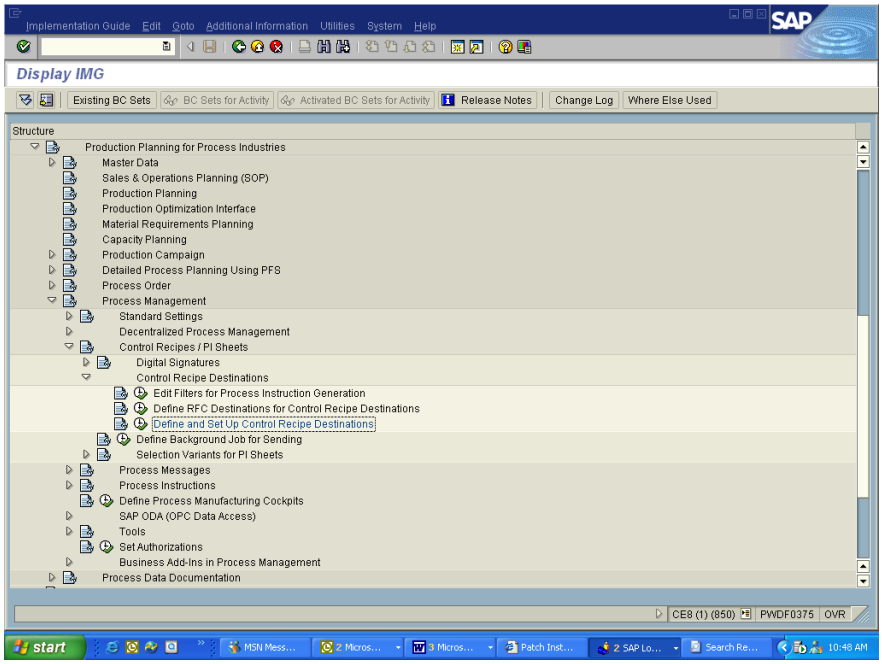
CRD This is the number that will be entered in the recipe instructions for the destination

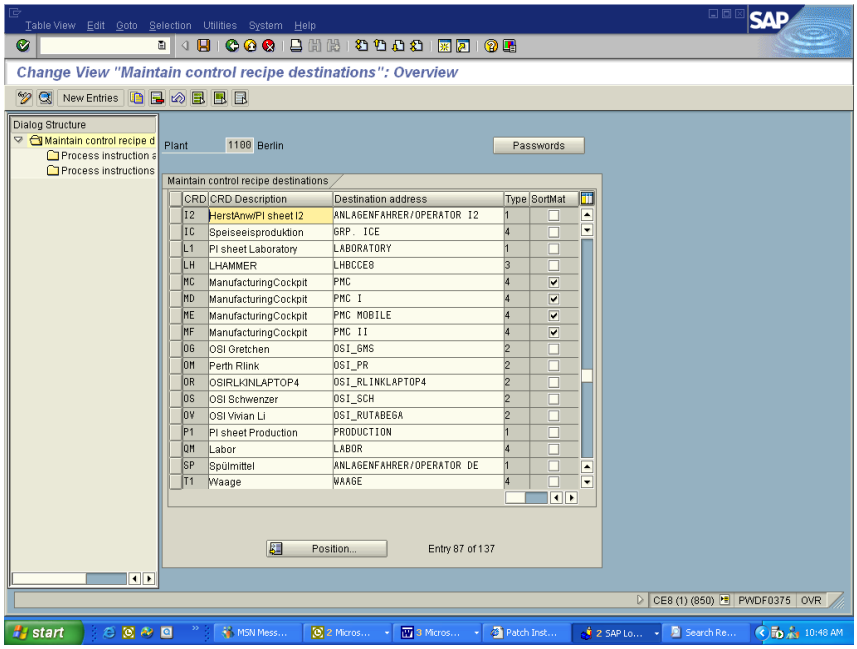
Description Anything can be entered here

Destination Address This must agree with the name used in SM59 transaction

Type 2=Recipe will be pushed down from SAP/R3 (This is done by scheduling a program to run on the SAP side. The correct program is RCOCB004)

3=Recipe will be pulled from the control system





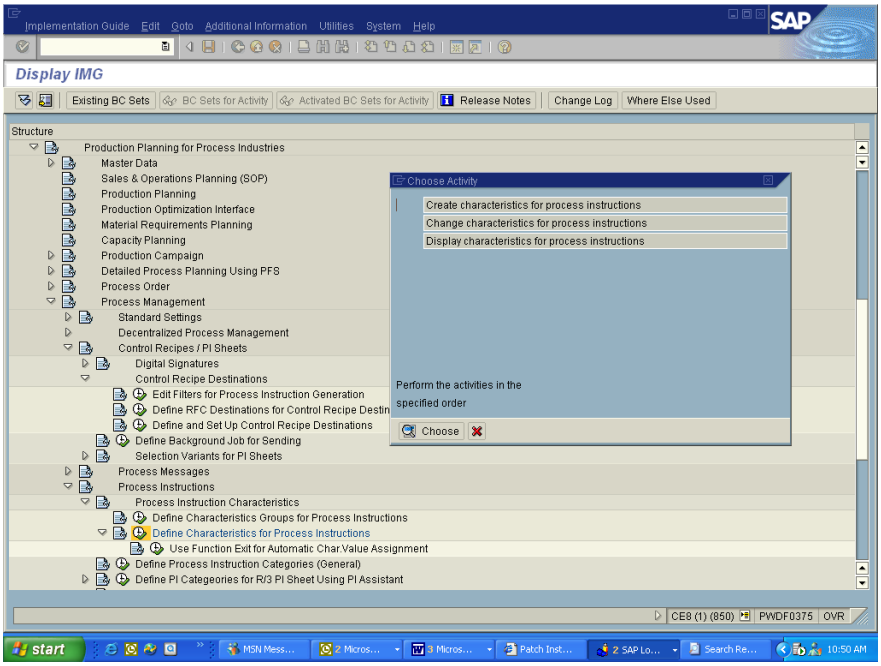
Transaction O10C

- Make modifications to SAP/R3 instructions. A few additional characteristics are required in the SAP/R3 instructions. The user can choose to make these modifications in the existing SAP/R3 instructions or duplicate the SAP/R3 instructions under a new name. The instructions that follow will be as if you modify the SAP/R3 instructions. The addition of the characteristics required can be made in the master recipe or can be permanently made in the SAP/R3 instruction or can be made in a copy of the SAPR/3 instruction with a name chosen by the user. Alias values for the characteristic names can be configured in the interface. The following shows using the OSI prefix if you require some other naming convention alias values need to be configured during the install.

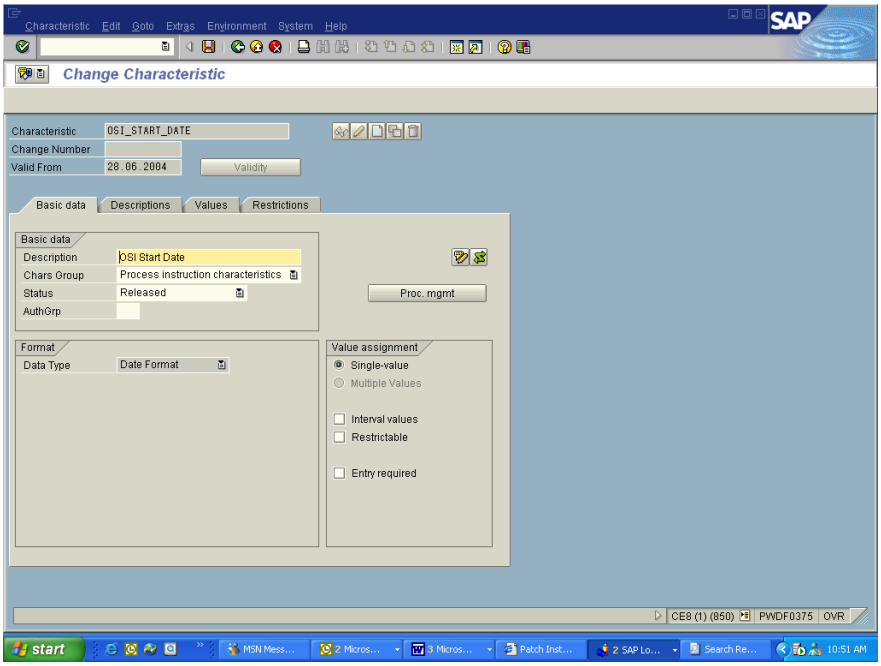
  Add the following characteristics.

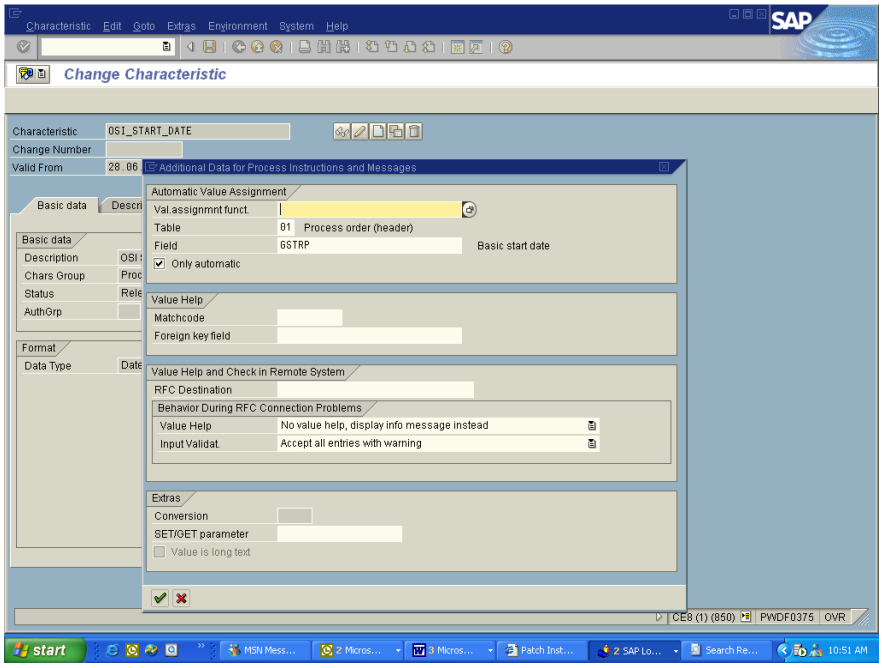
AORD OSI_START_DATE DATE

This should be set up for automatic assignment from SAP/R3 table CAUFVD or the Order header and field GSTRP.

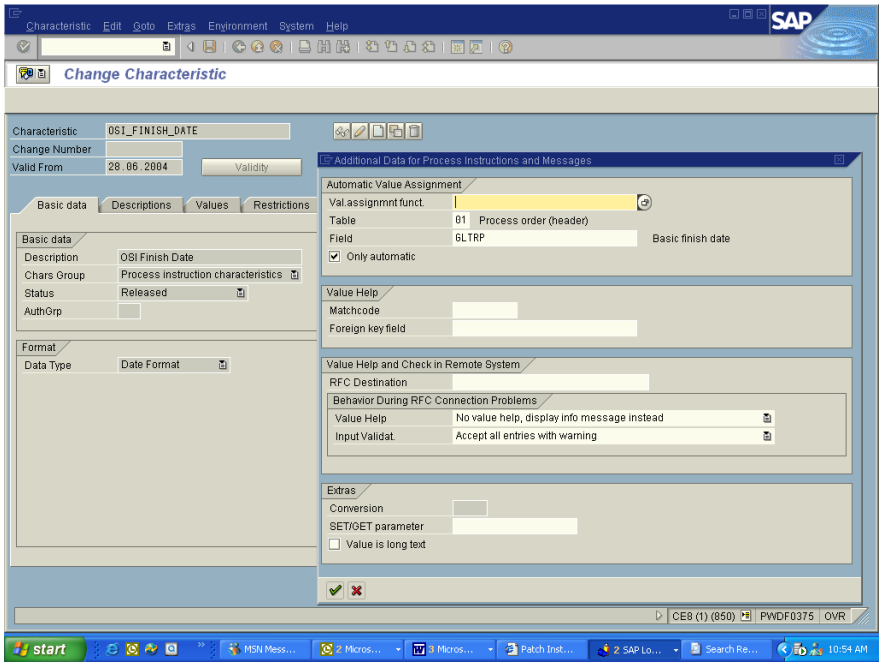


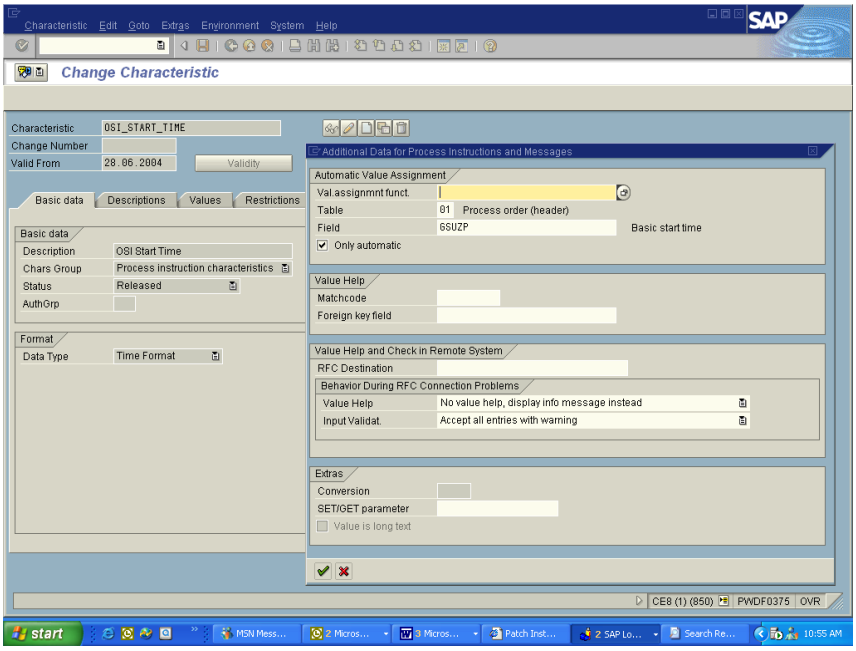
Select the Proc.mgmt button to get the automatic assignment of value screen.





You can also create OSI_FINISH_DATE. This is used in a continuous recipe for the finish date. In BES or BPI plants it is moved to the recipe table for information only.

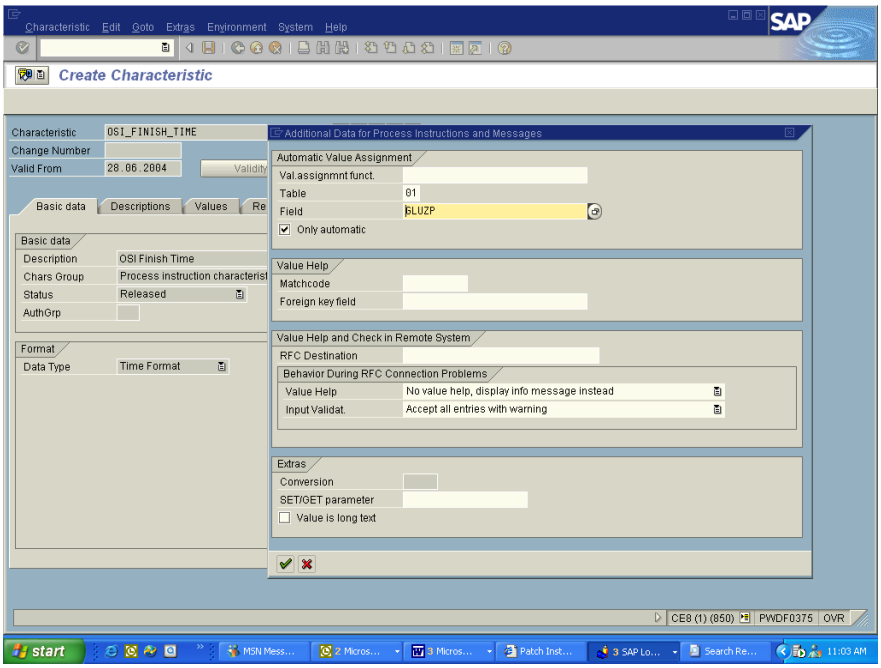




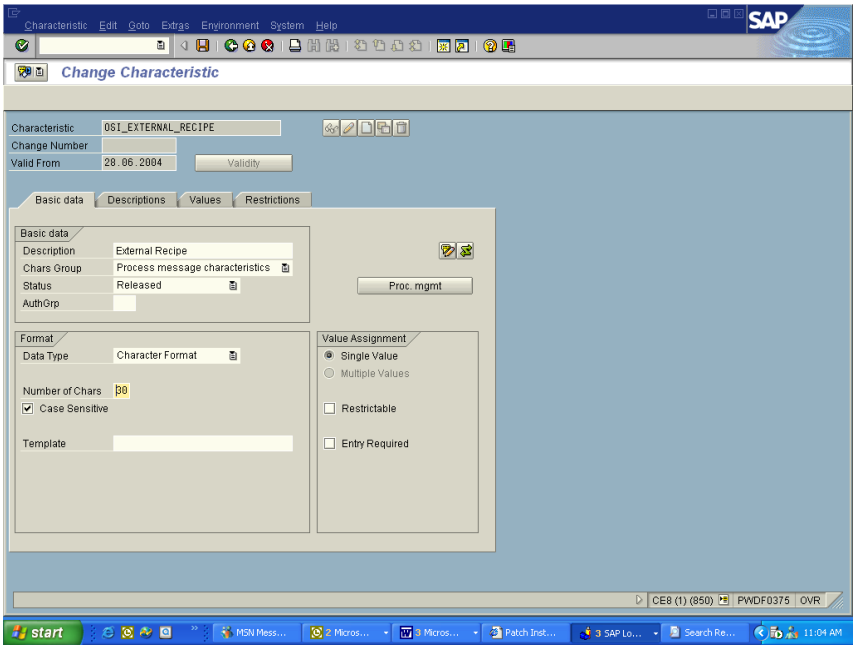
AORD OSI_START_TIME TIME

This should be set up for automatic assignment from SAP/R3 table CAUFVD or the Order header and field GSUZP.

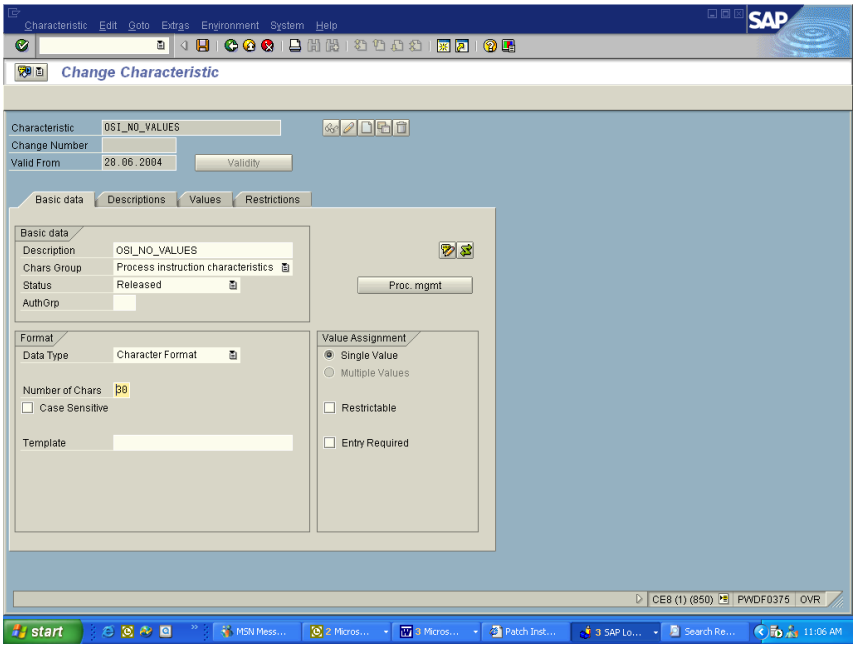
You can also create OSI_FINISH_TIME in a continuous recipe. This is used as the end time of the recipe. In BPI and BES plants this is transferred to the recipe table only for information.



AORD OSI_EXTERNAL_RECIPE CHAR 30 Case Sensitive

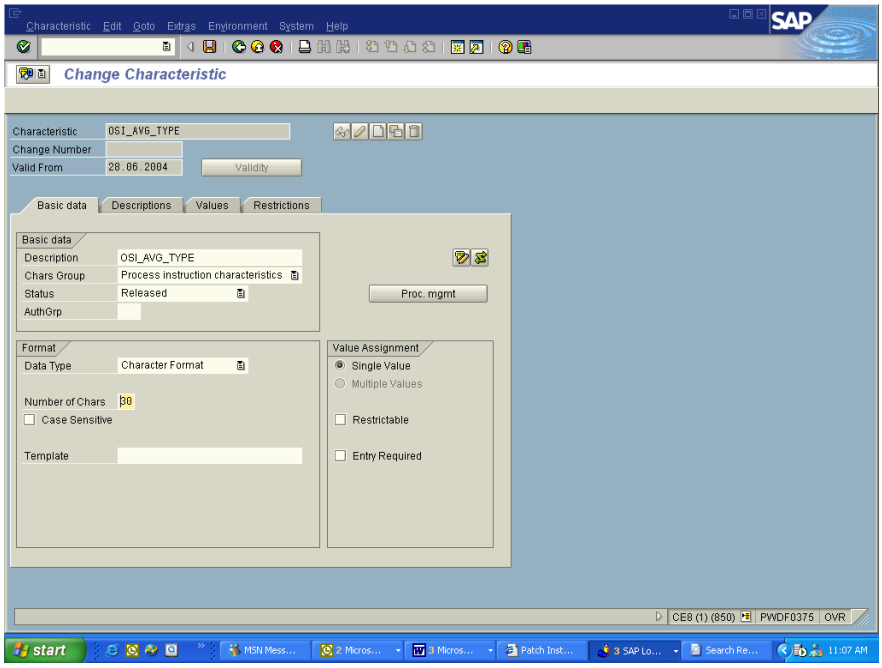


AREAD2 OSI_NO_VALUES CHAR 30



Defines the no of values to be returned in the multiple value read. If no value is given then 10 will be returned by default.

AREAD1 OSI_AVG_TYPE CHAR 30



Define new characteristics by selecting Tools, Customizing, Implement.projects, Display SAP/R3 Ref.IMG, Production Planning for the Process Industries, process management, process instructions, process instruction characteristics, define characteristics for process instructions, create characteristics for process instructions.

Enter the characteristic name given above and the description. For the date and time these can be assigned to SAP/R3 table names and fields for assigning the start time of the Process order. These assignments are as follows (check automatic only)

Date	Process order header 01	GSTRP
Time	Process order header 01	GSUZP

Note: For release 3.0C-3.0E of SAP/R3 release note 51371 must be applied for the automatic assignment of date and time to work correctly.

Special word of caution. When sending the time down from SAP/R3 we have noticed that if you set the recipe start time and then copy in the master recipe the start time is changed back to 00:00:00 no matter what time you had entered. In this case you will get a start time of 00:00:00 sent down. You must reset the start time after you copy in the master recipe and before you create the control recipe to the correct time. If the time is set to 24:00:00 you need to set the system_parameter DTTM to determine if you want this to move ahead one day or back one second.

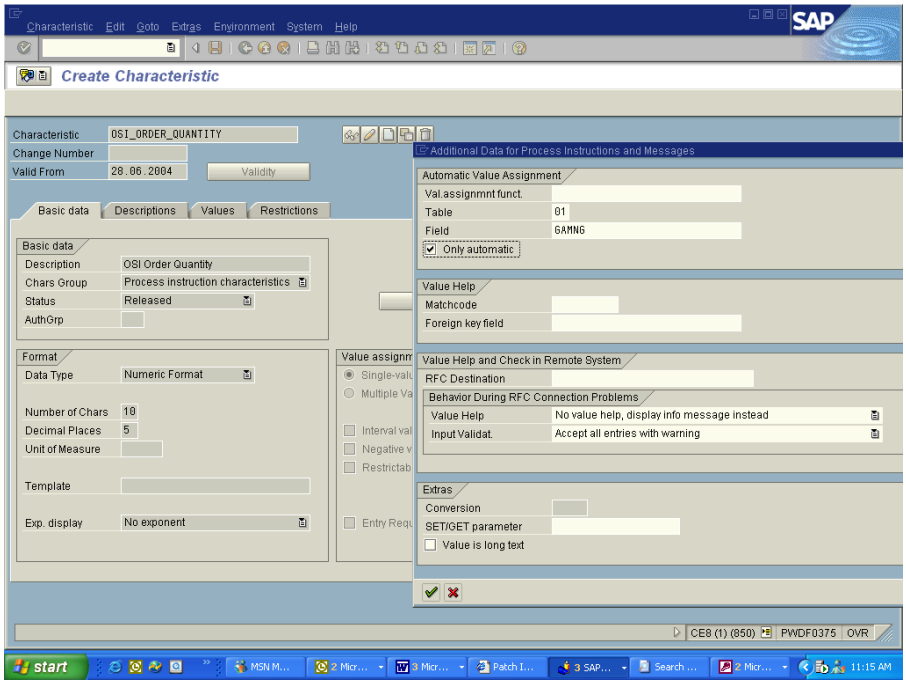
Assign the data type as shown above for each instruction characteristic. Assign the characteristic group PPPI_01.

For the OSI_AVG_TYPE characteristic the acceptable values must be given as follows

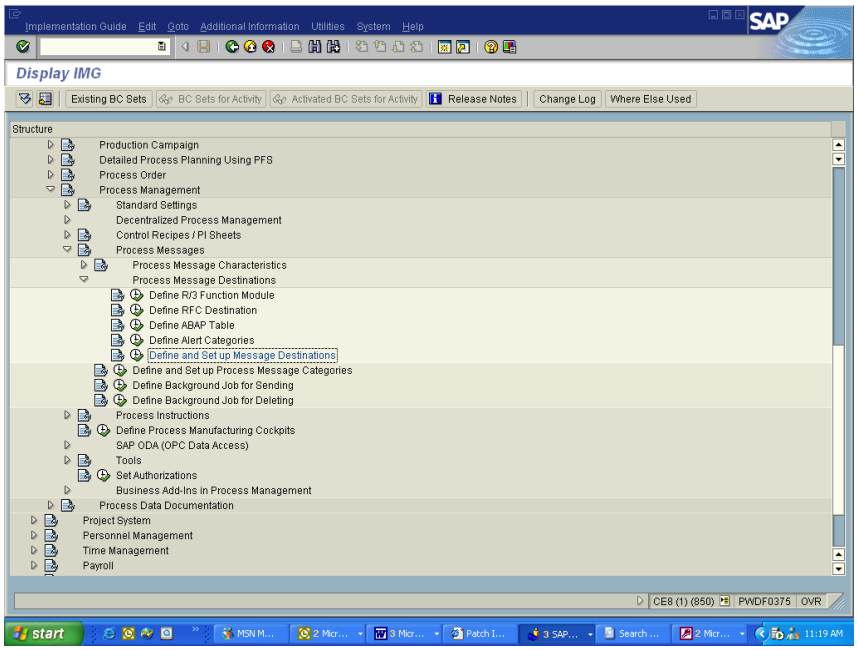
ARCTOTAL	AREAD1
ARCMINIMUM	AREAD1
ARCMAXIMUM	AREAD1
ARCSTDEV	AREAD1
ARCRANGE	AREAD1
ARCAVERAGE	AREAD1

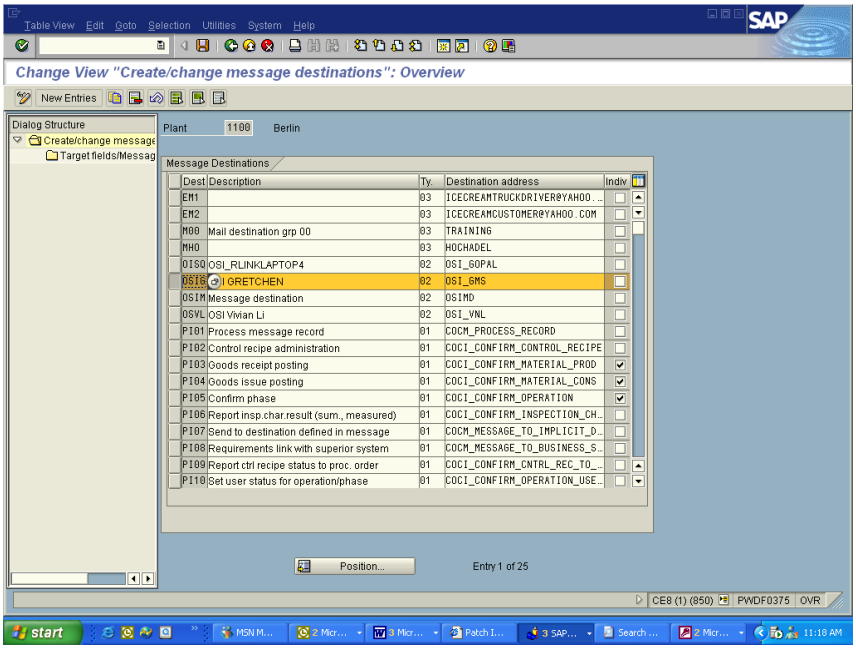
ARCMEAN AREAD1

If you want the order quantity in your recipe to be assigned automatically from the first screen of the SAP/R3 Process Order you need to change the Characteristic PPPI_ORDER_QUANTITY to obtain the value automatically from the SAP header. If you cannot change this instruction you can use a new characteristic called OSI_ORDER_QUANTITY that is setup as shown in the following displays. However you must then change the AORD_1 instruction to refer to OSI_ORDER_QUANTITY instead of PPPI_ORDER_QUANTITY.



- Setup comment message destination in SAP/R3





If comment message text is to be sent down to the control system then a destination must be setup for these ad-hoc messages.

Tools, Business Eng, Customizing, Implement.projects, SAP/R3 Ref.IMG, Production Planning for the Process Industries, process management, process messages, setup message destination. The other properties of the message category are assigned to the destination in the transaction /nO13C.

Dest	OSI
Description	Any description
Type	=02 for SAP/R3 to push down
Address	The address must match the SM59

Use the customizing menu option “set up process message category” from customizing menu to set up the message category OSI_COMM

Dest	OSI
Description	Any description
Type	=02 for SAP/R3 to push down
Destination Address	The address must match the one setup in SM59

The characteristics for this message category would include PPPI_MESSAGE_TEXT along with order, phase, source, event time and date. The specific characteristics are as follows:

PPPI_EVENT_DATE

PPPI_EVENT_TIME

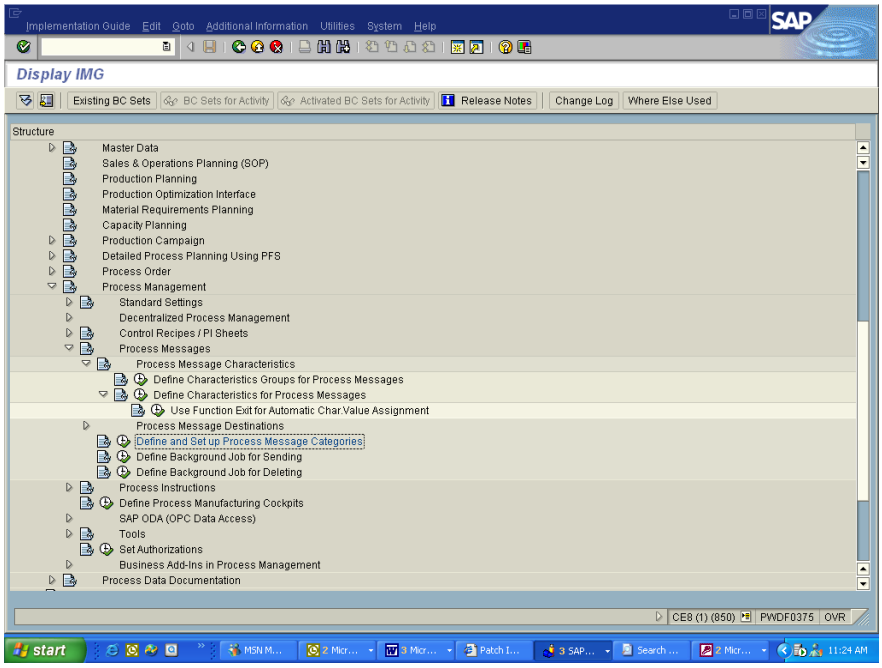
PPPI_MESSAGE_TEXT

PPPI_OPERATION

PPPI_PHASE

PPPI_PROCESS_ORDER

PPPI_SOURCE


Implementation Guide Edit Goto Additional Information Utilities System Help
Display IMG
Existing BC Sets
BC Sets for Activity
Activated BC Sets for Activity
Release Notes
Change Log
Where Else Used
Structure
Master Data
Sales & Operations Planning (SOP)
Production Planning
Production Optimization Interface
Material Requirements Planning
Capacity Planning
Production Campaign
Detailed Process Planning Using PFS
Process Order
Process Management
Standard Settings
Decentralized Process Management
Control Recipes / PI Sheets
Process Messages
Process Message Characteristics
Define Characteristics Groups for Process Messages
Define Characteristics for Process Messages
Use Function Exit for Automatic Char.Value Assignment
Process Message Destinations
Define and Set up Process Message Categories
Define Background Job for Sending
Define Background Job for Deleting
Process Instructions
Define Process Manufacturing Cockpits
SAP ODA (OPC Data Access)
Tools
Set Authorizations
Business Add-Ins in Process Management
Process Data Documentation
CE8 (1) (850)
PWDF0375
OVR
start
11:24 AM

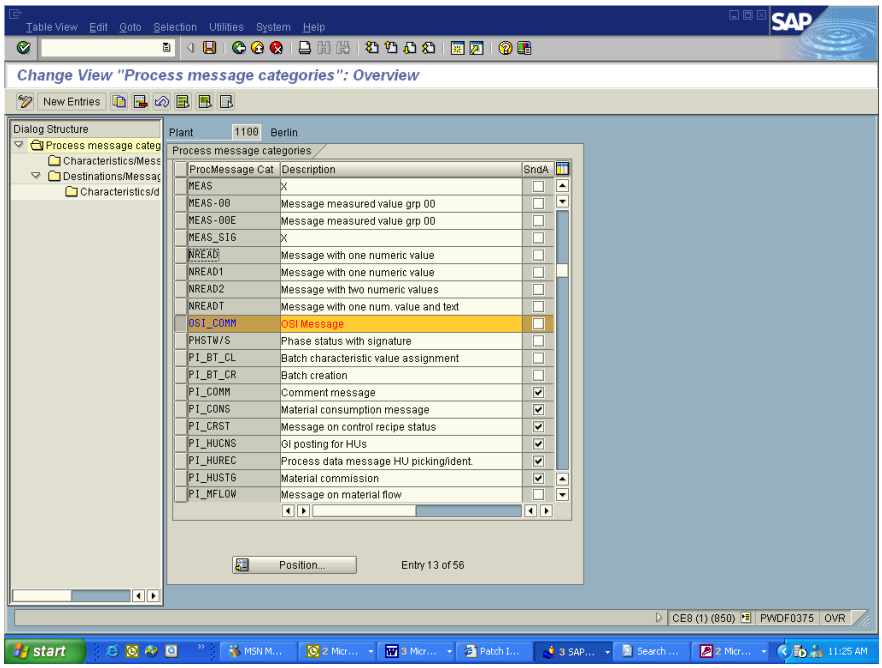


Table View Edit Goto Selection Utilities System Help
Change View "Process message categories": Overview
New Entries
Dialog Structure
Process message categ
Characteristics/Mess
Destinations/Messag
Characteristics/d
Plant 1100 Berlin
Process message categories
ProcMessage Cat Description SndA
MEAS X
MEAS-00 Message measured value grp 00
MEAS-00E Message measured value grp 00
MEAS_SIG X
NREAD Message with one numeric value
NREAD1 Message with one numeric value
NREAD2 Message with two numeric values
NREADT Message with one num. value and text
OSI_COMM OSI Message
PHSTW/S Phase status with signature
PI_BT_CL Batch characteristic value assignment
PI_BT_CR Batch creation
PI_COMM Comment message
PI_CONS Material consumption message
PI_CRST Message on control recipe status
PI_HUCNS GI posting for HUs
PI_HUREC Process data message HU picking/ident.
PI_HUSTG Material commission
PI_MFLOW Message on material flow
Position...
Entry 13 of 56
CE8 (1) (850)
PWDF0375
OVR
start
11:25 AM

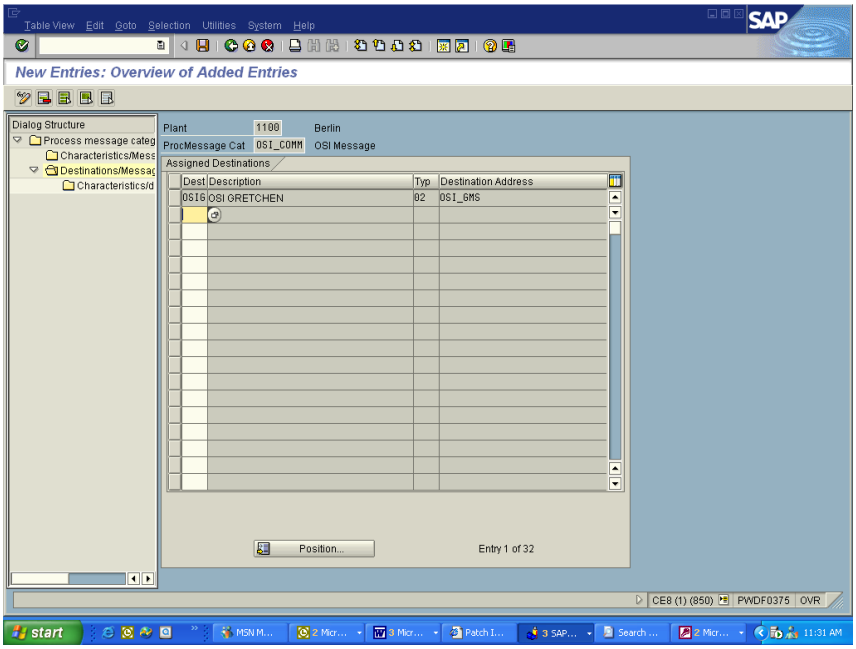


Table View Edit Goto Selection Utilities System Help
New Entries: Overview of Added Entries
Dialog Structure
Process message categ
Characteristics/Mess
Destinations/Messag
Characteristics/d
Plant 1100 Berlin
ProcMessage Cat OSI_COMM OSI Message
Assigned Destinations
Dest Description Typ Destination Address
OSIG OSI GRETCHEN 02 OSI_GMS
Position...
Entry 1 of 32
CE8 (1) (850) PWDF0375 OVR
start
11:31 AM

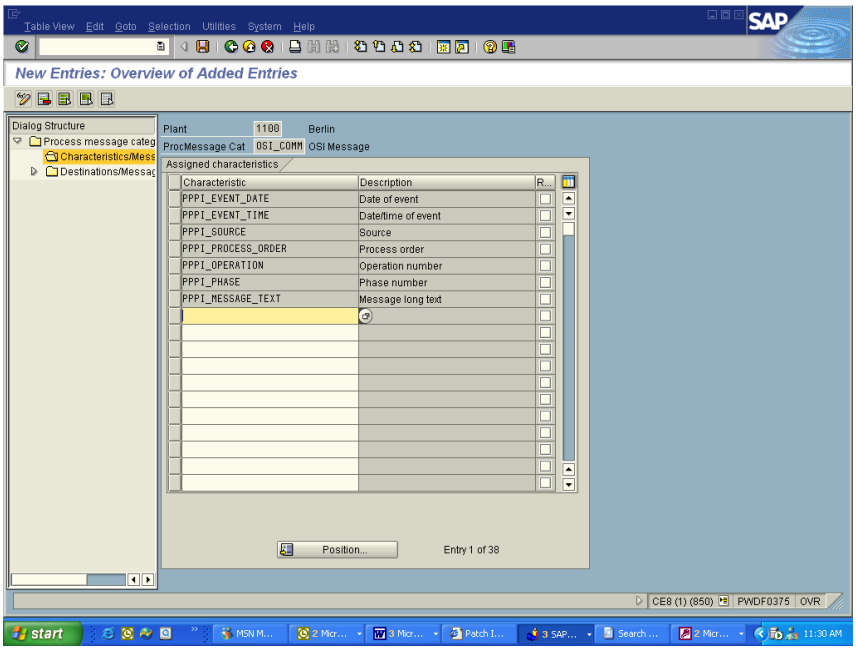
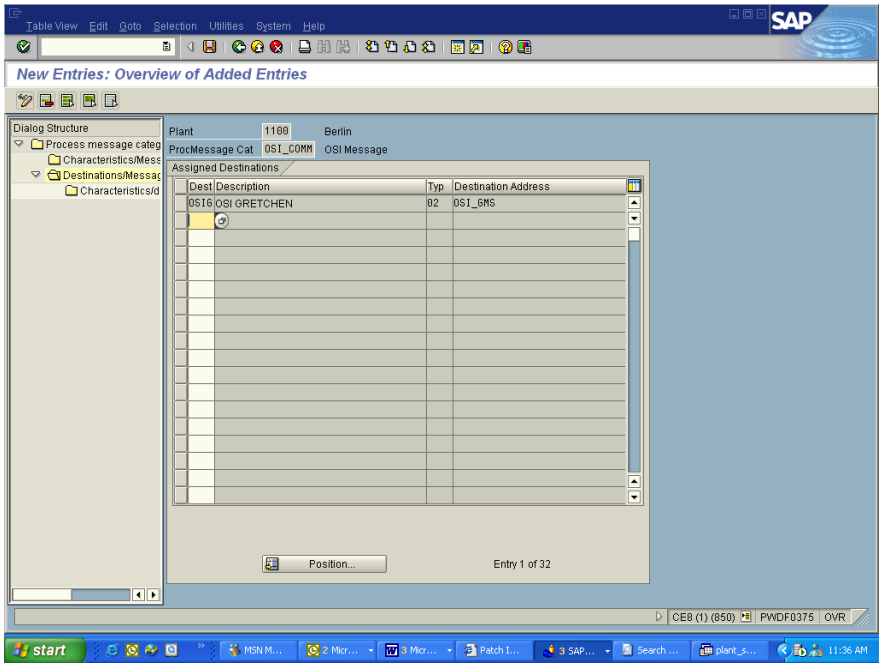
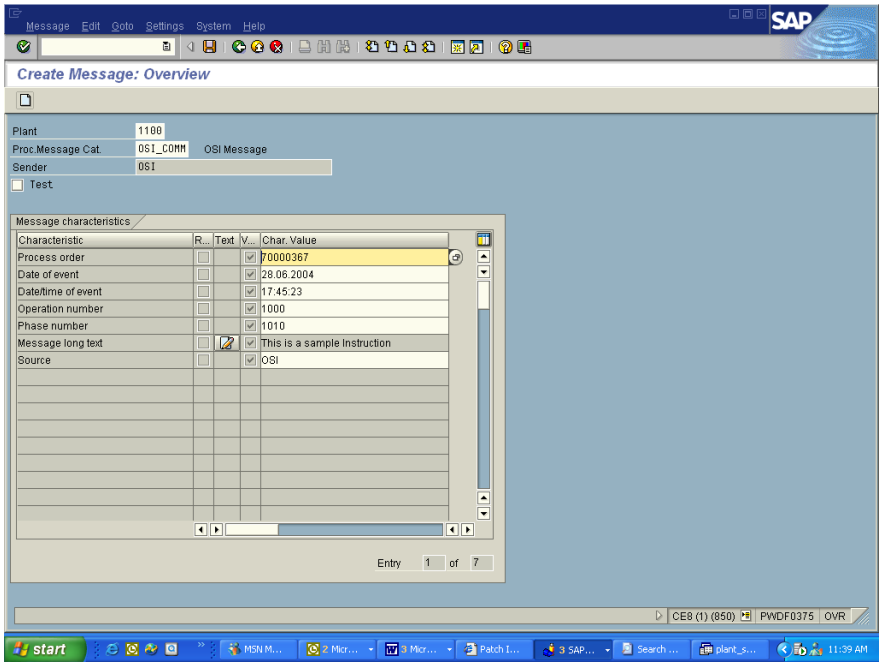
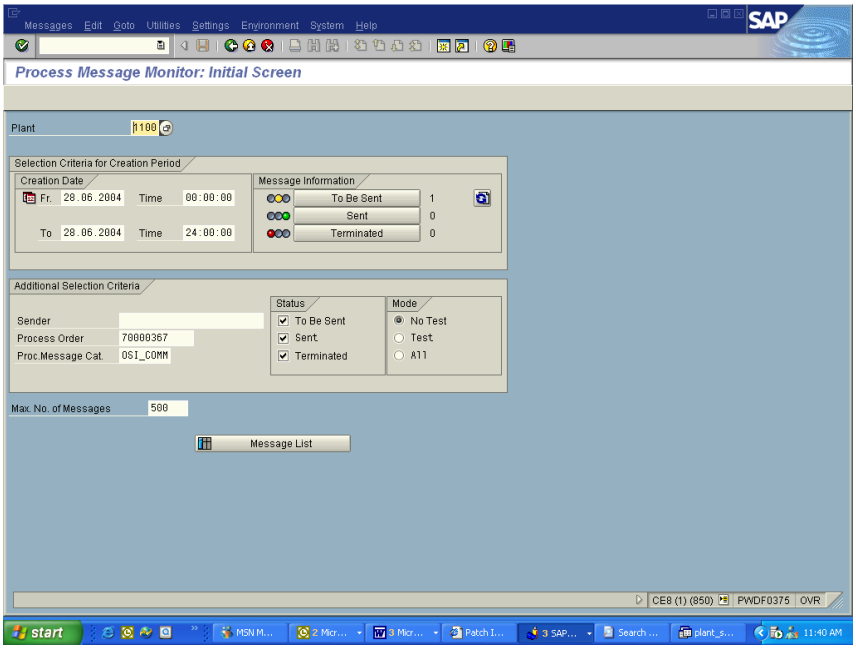


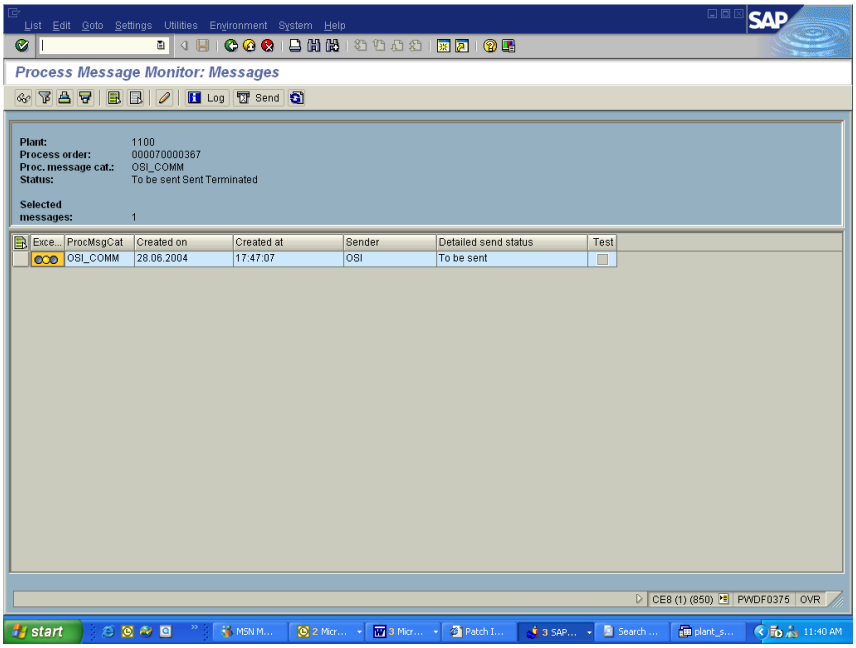
Table View Edit Goto Selection Utilities System Help
New Entries: Overview of Added Entries
Dialog Structure
Process message categ
Characteristics/Mess
Destinations/Messag
Plant 1100 Berlin
ProcMessage Cat OSI_COMM OSI Message
Assigned characteristics
Characteristic Description R...
PPPI_EVENT_DATE Date of event
PPPI_EVENT_TIME Date/time of event
PPPI_SOURCE Source
PPPI_PROCESS_ORDER Process order
PPPI_OPERATION Operation number
PPPI_PHASE Phase number
PPPI_MESSAGE_TEXT Message long text
Position...
Entry 1 of 38
CE8 (1) (850) PWDF0375 OVR
start
11:30 AM



A message is actually sent by using transaction CO57 message create and monitor using transaction CO54.







- Adding the INSTR instruction to recipe. If you are going to use this instruction to send down text messages in a recipe then use the characteristic PPPI_MESSAGE_TEXT with the instruction for the text.
- Ad-hoc messages have been used to send down information after the recipe has been sent down. Examples of this are changing the batch number, changing properties of materials or gneric values you want to send to the PI System.

Install Microsoft SQL Server

- Install as given in the Microsoft SQL Server Documentation. In the US version of SQLServer the default option is case insensitive. The product is fully tested under this version. We have found that in some international versions of Microsoft SQLServer this option does not exist. The PS-RLINK product has been fully tested with dictionary case insensitive and character set 850 for International character

sets. If your version of SQLServer does not offer either of these options you should contact technical support.

- If you are using a language other than English then the install should be done using multi lingual 850 during the SQLServer install. Configure the language as English on NT for the user in SAP fand for the ODBC configuration slect perform translation.
- You must use Enterprise manager to register the server. Use the control panel to set MSSQL for automatic starting upon system boot. If you are unable to register your server check the client configuration, net library that the named pipes option is selected.
- In Query Analyzer or ISQL un-check the Nulls padding for the configuration.
- If you are running Microsoft SQL2000 then the set_quoted_identifer must also be unchecked.
- If you are using any special characters such as the degree mark then you must slect the SQLServer, Client network Utility. Then select DB-Library Option and uncheck the box tha is Automatic ANSI to OEM.

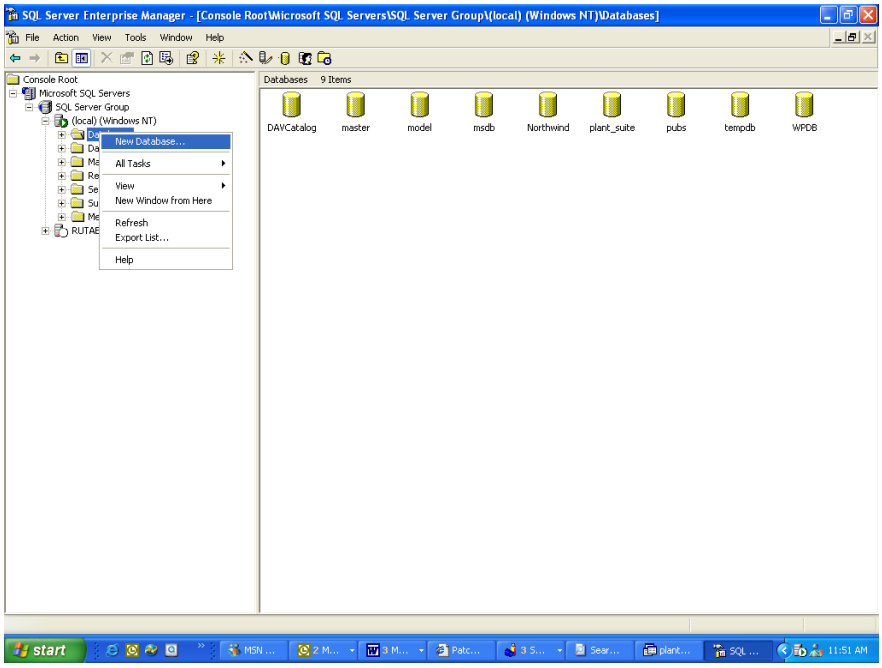
Creation of DB Devices and Sizing

The sizing of the database should be done by the following rule that was derived from a recipe of 14 phases and 17 materials. Assume 400KB per Recipe for the data and indexes and 600KB for logspace processing. Thus for 100 recipes/ day and 30 days of storage this would be 1.2GB for Data. Using a weekly backup schedule of the database and daily backup schedule for transaction log the logspace required is 60MB or to be safe 100MB and 600MB of filespace for dumping transaction logs. To backup the database you need 1.2GB

Two devices should be created with the following:

Name	Size
PLANT_SUITE_data	1200MB
PLANT_SUITE_log	100MB

Right clicking on Database or selecting new database from the Action menu will give you the option to create a database.



For the new database enter the name “plant_suite” on the general tab of the dialog and type the initial size of the database in this example it is 1200. Click on the Transaction Log tab and type the initial size of the log is100 in this example.

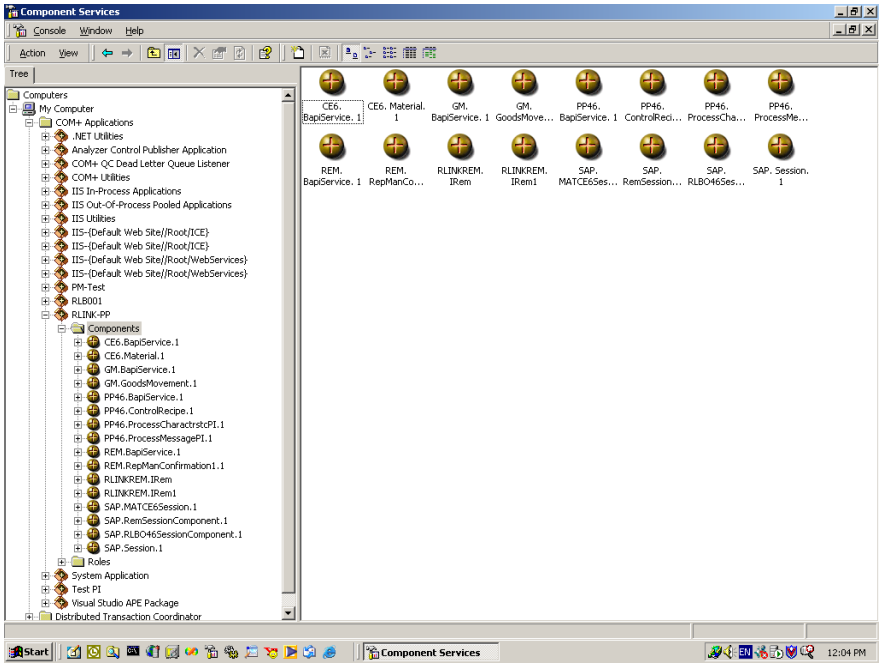
Check that the DB service has been set to automatic.Verify that the Microsoft DB services have been set to automatic upon system startup. Control Panel, Services Icon.

Install PI-SDK

This will be installed as part of the normal setup for version 1.6 of RLINK PPPI. If you are having problems with the PITags connection use the AboutPI-SDK.exe application found in the PIPC\pisdk directory to test the connection and verify it is talking to the correct PI server.

Microsoft Component Server

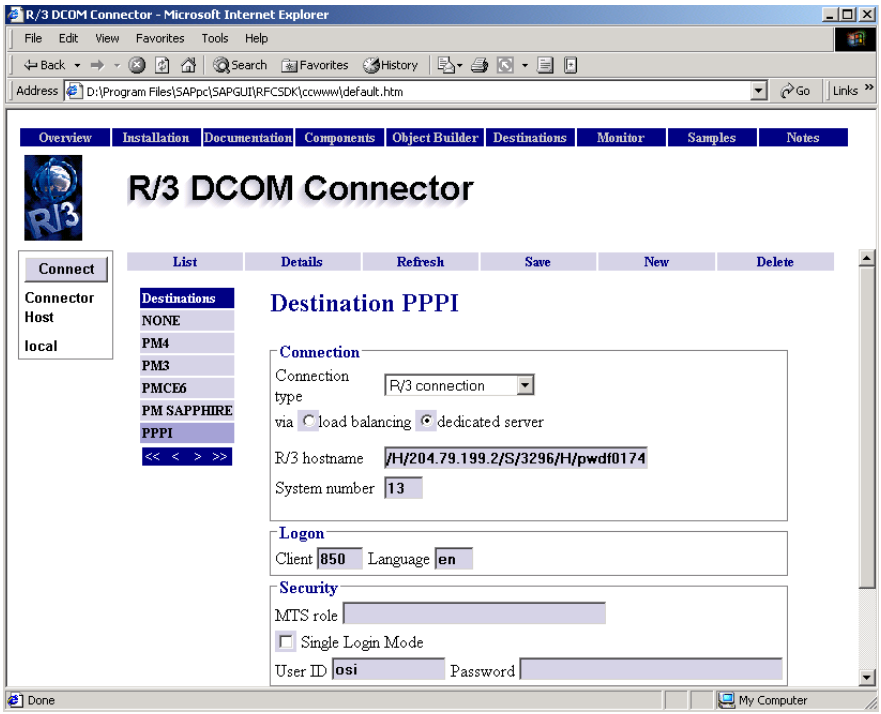
After the entire install of RLINK-PPPI has been completed you will see the components deployed in the component server. After the install of RLINK you will see the following components in the component server.



SAP DCOM Connector

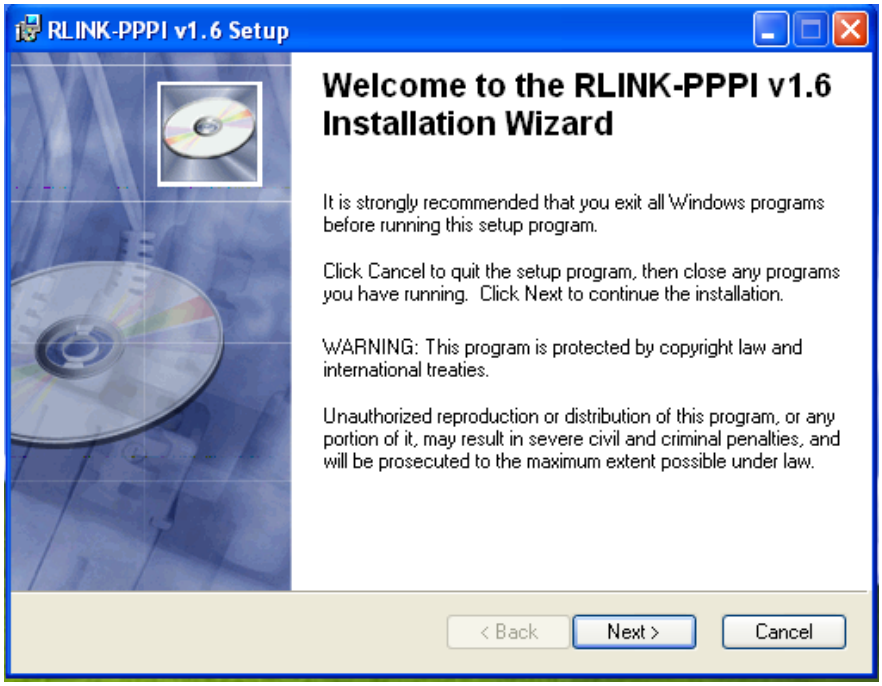
Using the SAP GUI CD start the SAP Setup program. And select the Development Tools. Choose Option change and select RFC SDK Libraries Install DCOM Connector

1. Go to the Destination create a new destination
2. Enter the saprouter ip address value, system number and the client.
3. Test whether a connection is established by going to details for the connection you have configured. Information about the destination should be returned from SAP.

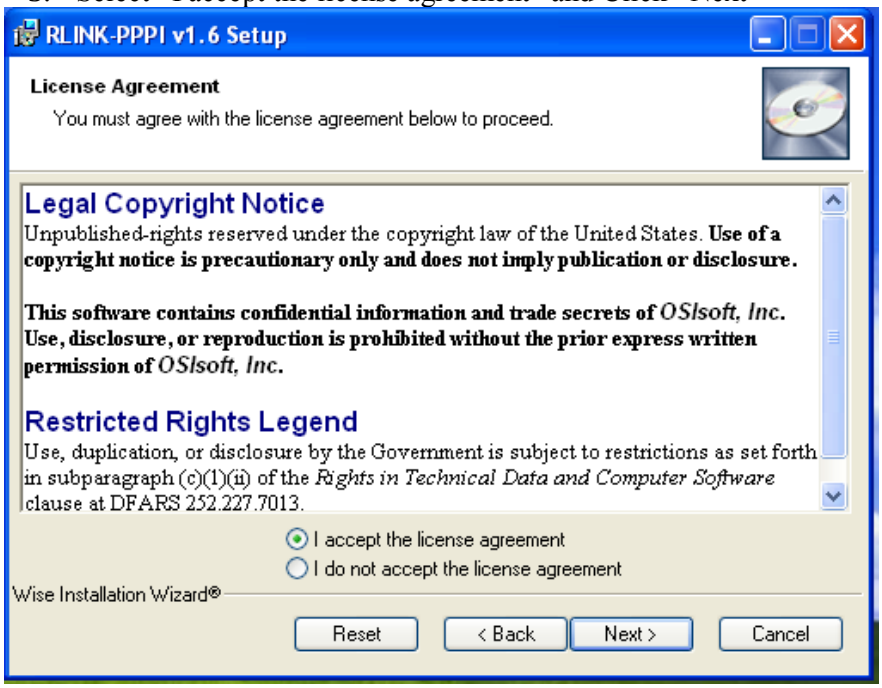


RLINK-PPPI Server Installation

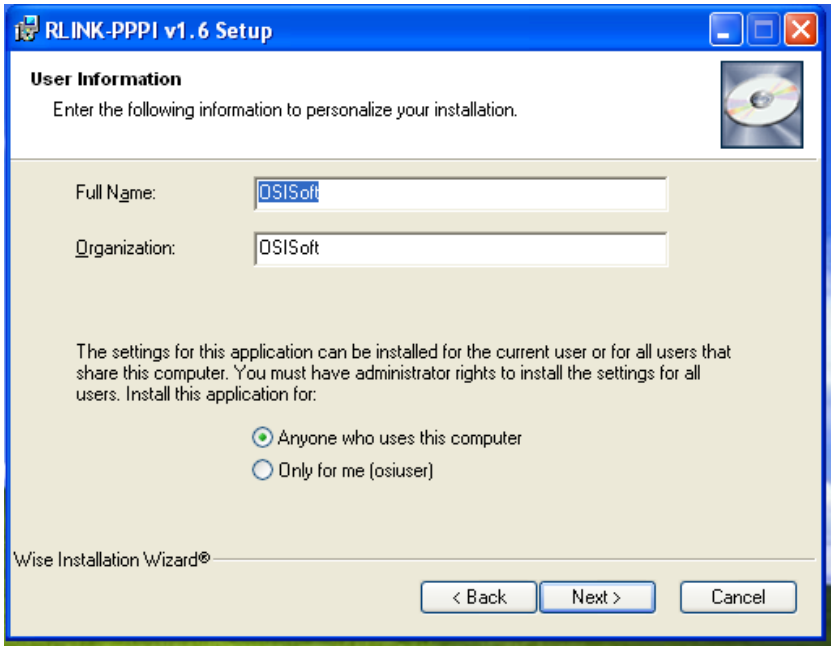
1. Start the installation by Double-Clicking the executable.
2. Click “Next”



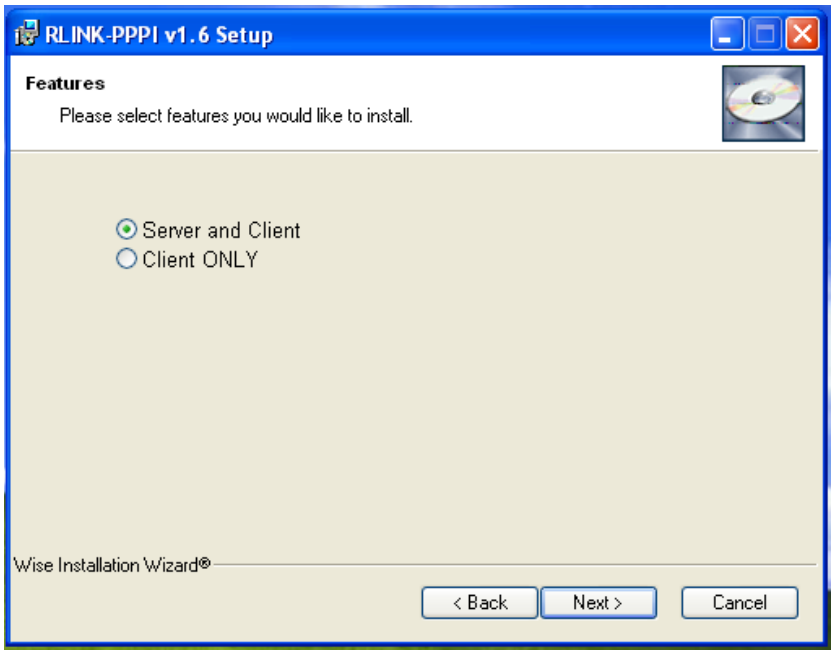
3. Select “I accept the license agreement” and Click “Next”



4. Set appropriate settings and click “Next”



5. Select “Server and Client” and click “Next”



6. Provide the RLINK Database Information as follows:

RLINK Applications:

Database Server: name of the computer that has the database (i.e. the SQL Server name)

Database Name: the name of the database (plant_suite).

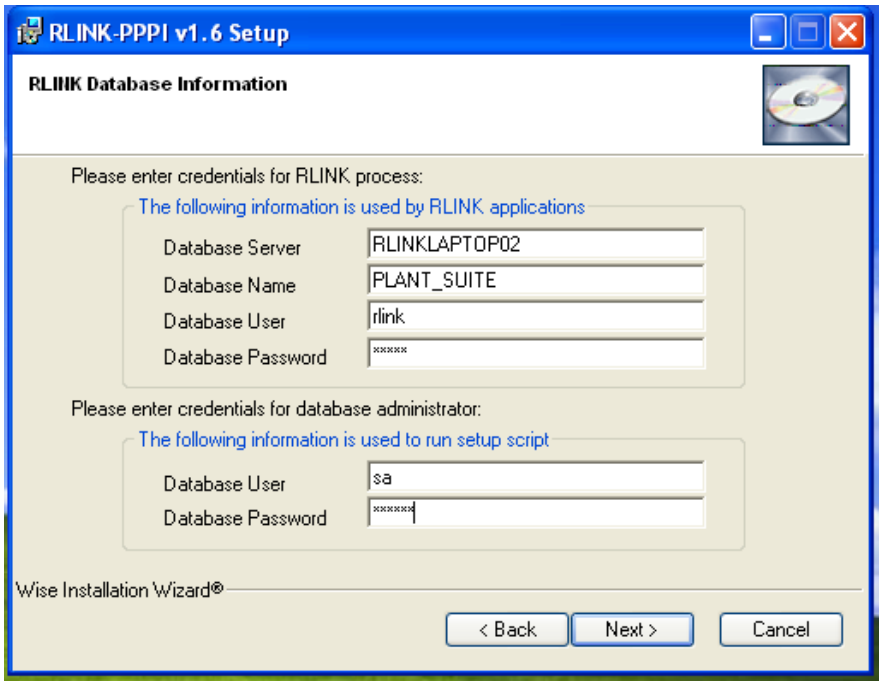
Database User: the user who has priviledge to select, update, delete from the database and execute procedures in the database

Database password: the password for the database user above.

Database Administrator:

Database User: the database administrator (example: sa) user who has privledge to create tables and procedures in the database.

Database password: the password for the database administrator

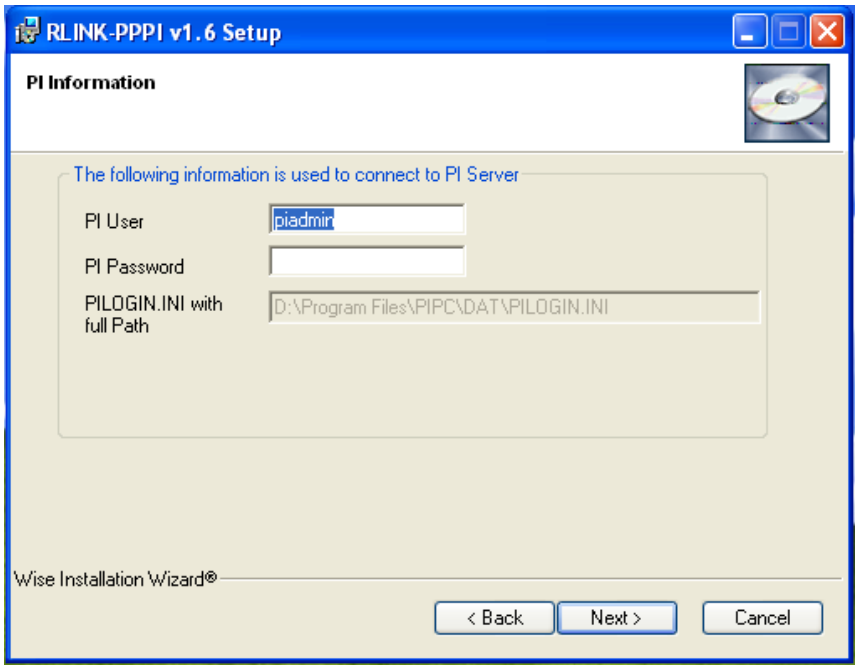


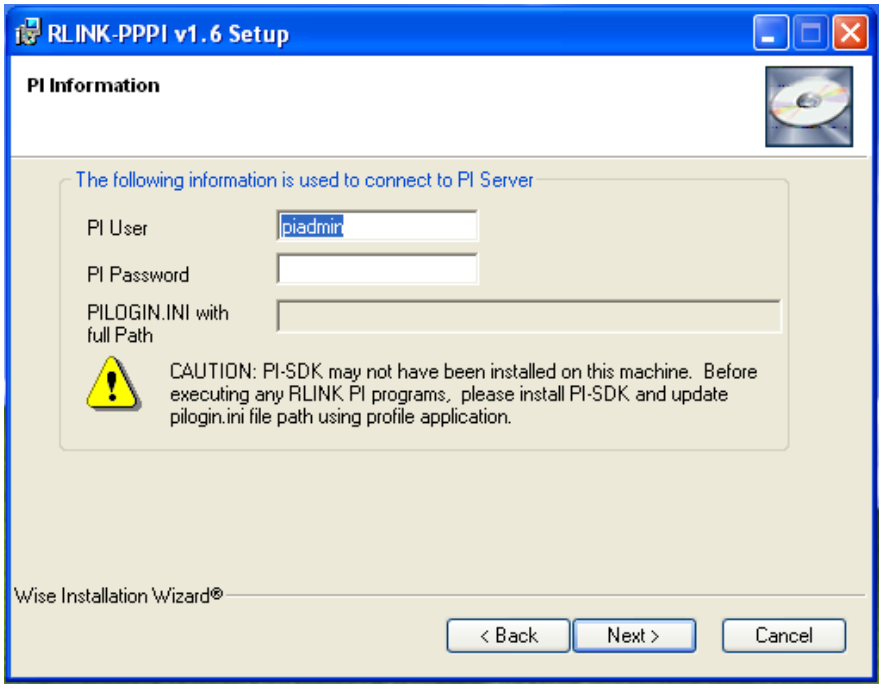
Click “Next”

7. Provide the PI information as follows:

PI User: PI user with read/write access

PI password: password for user above. PILogin ini File Path: full path (include a backslash at the end). If the setup program could not find PIHOME value in PIPC.INI or could not find PIPC.INI file in Windows directory, it shows a warning.





8. Refer to the SAPlogon and your sm59 transaction destination to provide SAP information:

Client: corresponds to "Client" in the SAPLogon

User: user that can connect to SAP

Password: password for the user above.

Language: the correct Language for your regional settings

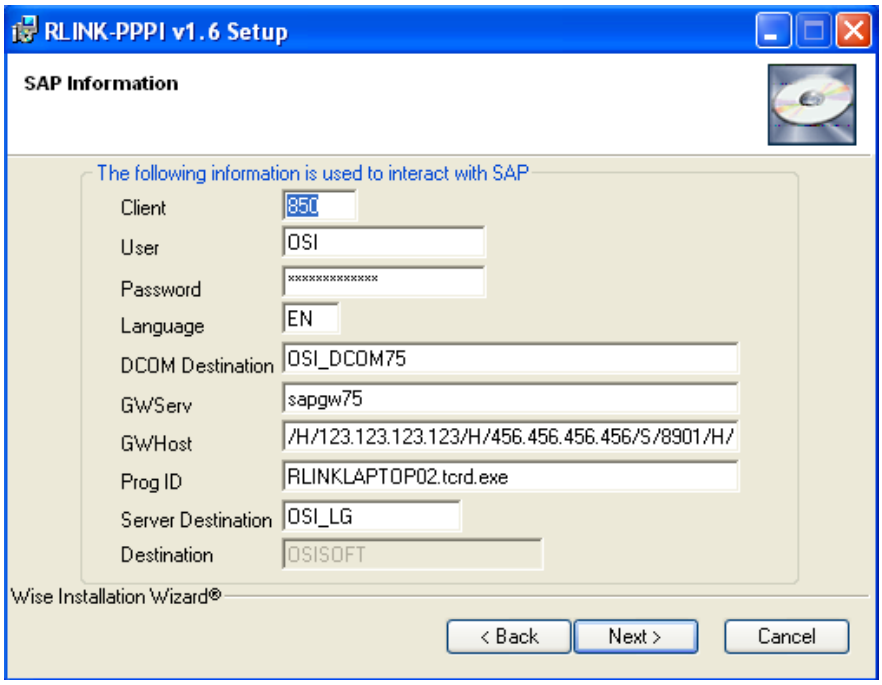
DCOM Destination: SAP DCOM configured for BAPI's

GWServ: the name of your Gateway Server sapgwnn where nn is the system number

GWHost: corresponds to the "SAP Route String" & "/" & "Application Server" in the SAPlogon

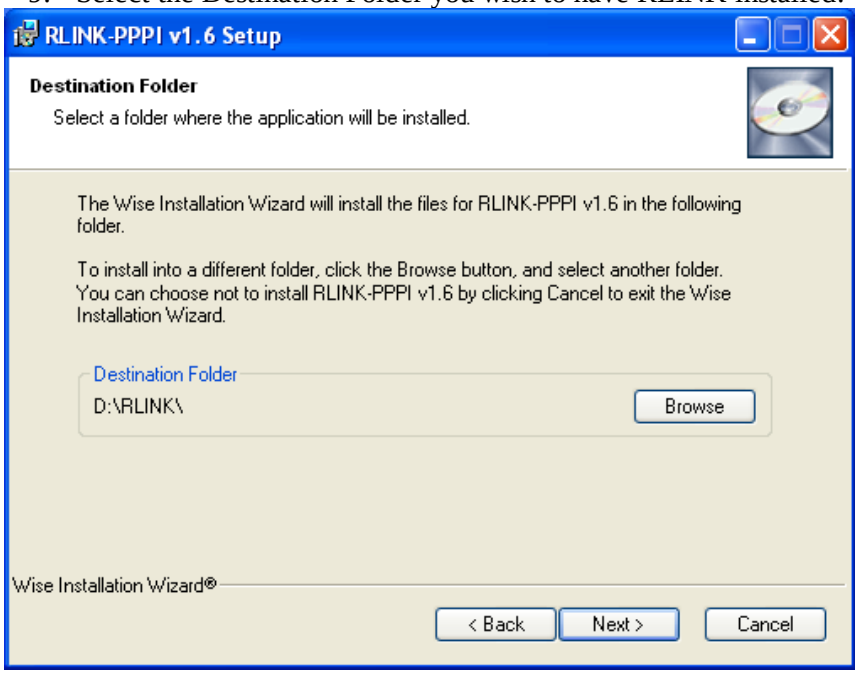
Prog ID: corresponds to the program ID entered in the sm59 transaction destination

Server Destination: corresponds to the "SM59 destination



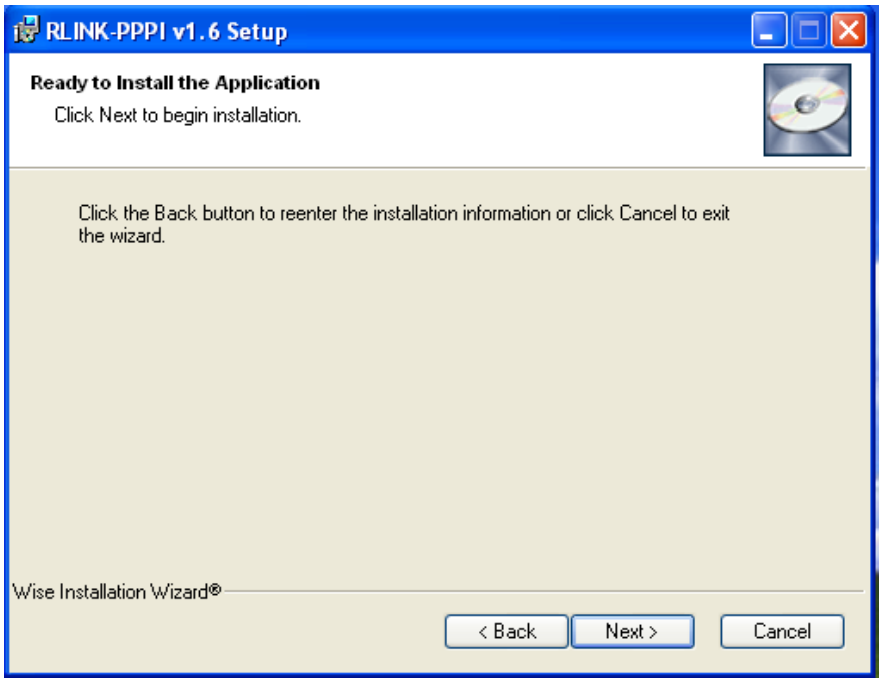
Click “Next”

9. Select the Destination Folder you wish to have RLINK installed:



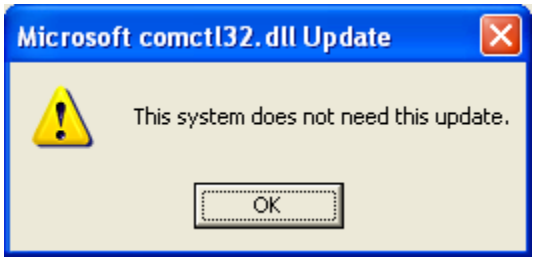
Click “Next”

10. Click “Next” to continue installation.

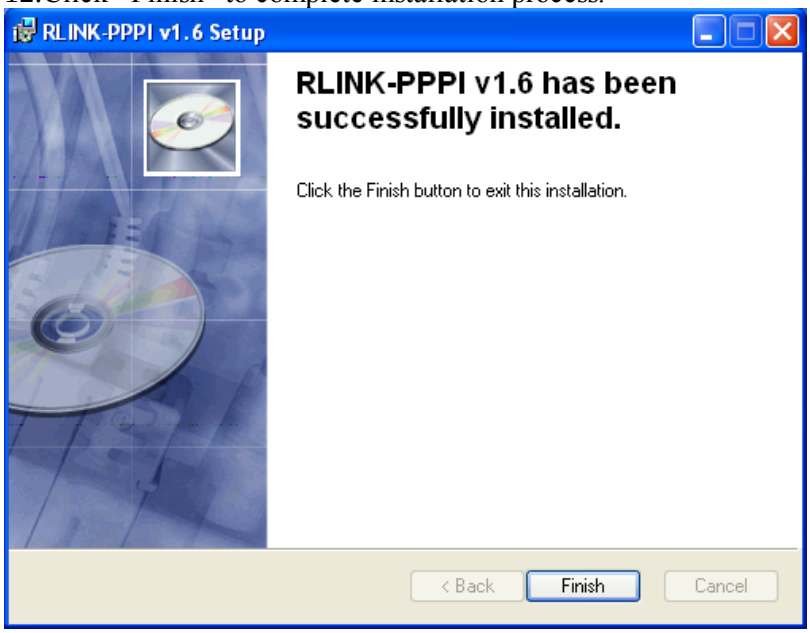


Note: Click “OK” to the KeyInstall warning (this warning may indicate a improper install):

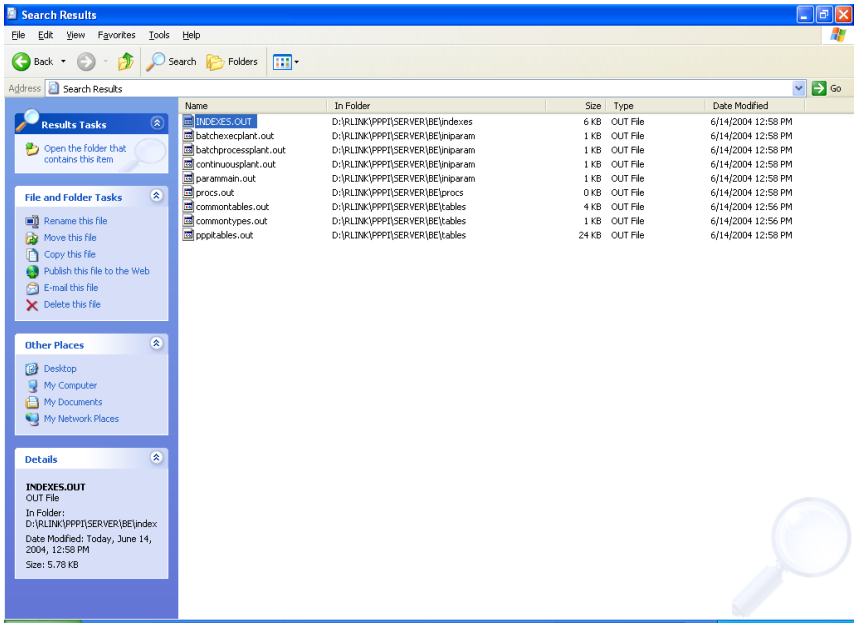
11.Click “OK” to the Microsoft comctl32.dll Update warning (this warning is Okay):



12.Click “Finish” to complete installation process.



13.Go to each directory below the RLINK\PPPI\SERVER\BE directory and check the .out files to verify no errors during installation. The file names are



The installation did the following:

- Creation of directory structure
- Copy of files for RFC handling
- Database install and initial load of tables
- Program executables transfered
- Set up of default registry
- Writes the SAPRFC.INI file
- Sets path RLINK\shared
- Creates and sets environement variables
- After completion of the script you should review that the database was loaded correctly. This is done by reviewing the .OUT files , PPPI_DB.log, PPPI_Install.log.
- The table rlink_version is updated at the completion of the install with the indication of what version is installed.

Create Plant Suite SAP/R3 User

To create the sapuser account, select the SQL Enterprise Manager, Database server, Logins, Manage, Logins. Create the login name sapuser and enter password, give access to the plant_suite database and set the Group to public, default database plant_suite.

Server components and security

If you look at the Microsoft component explorer you should now see the RLINK-PPPI components.

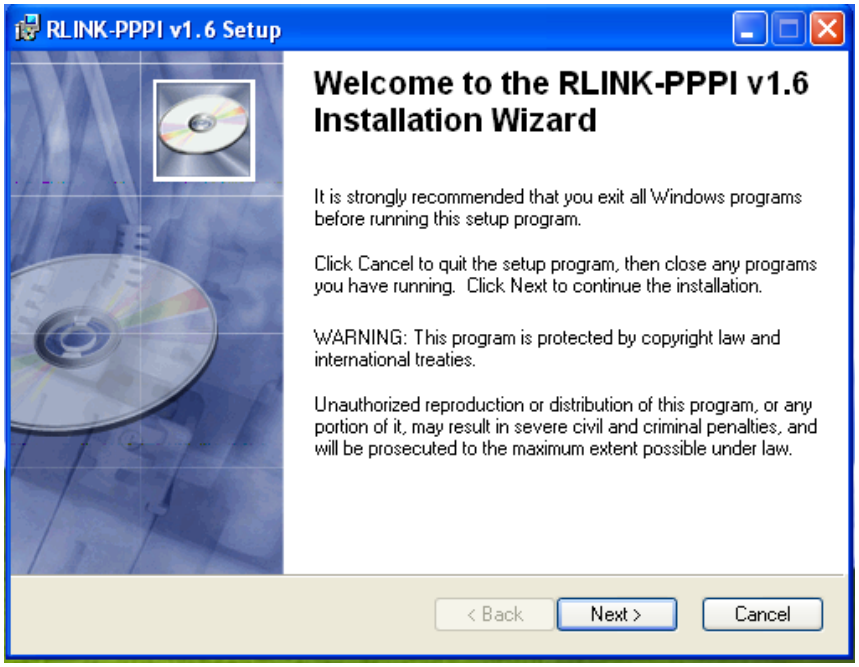
The security must be established for the server components. This is done in the following steps.

1. Create a local User(rlinkuser) in the Server Side without the password
2. Open the Transaction Server
3. Right click on the "RLINK-PPPI" Package and goto properties and the Identity Tab
4. Choose the "This user" and click the browse button select the "rlinkuser" from the list
5. Click the Ok/Apply Button from the "RLINK-PPPI Properties".
6. Right Click on Packages Installed and select Refresh
7. Right Click on "My Computer" and select "Shut Down Server Processes"
8. Similarly Right Click on "My Computer" and select "Refresh All Components"
9. Run the "regedt32.exe" from Start->Run
10. Point the Mouse to HKEY_LOCAL_MACHINE\SOFTWARE\PS_RLINK, Under the Security Menu choose Permissions.
11. Click the "Add" button and select the "rlinkuser" from the list.
12. Click the Ok button and then click Ok button on the Registry key permissions

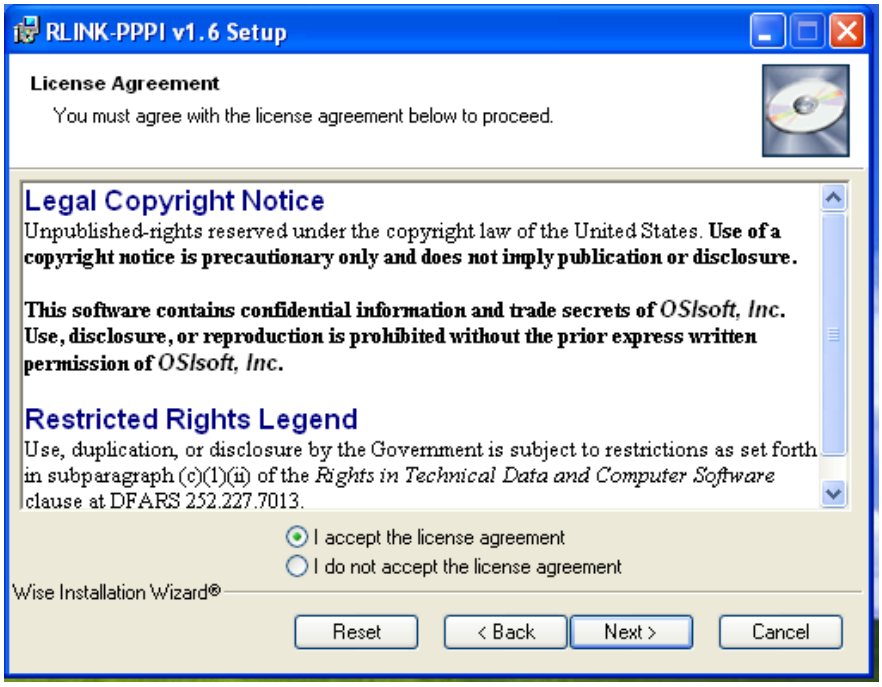
Client Install Setup.exe

RLINK-PPPI Client Installation

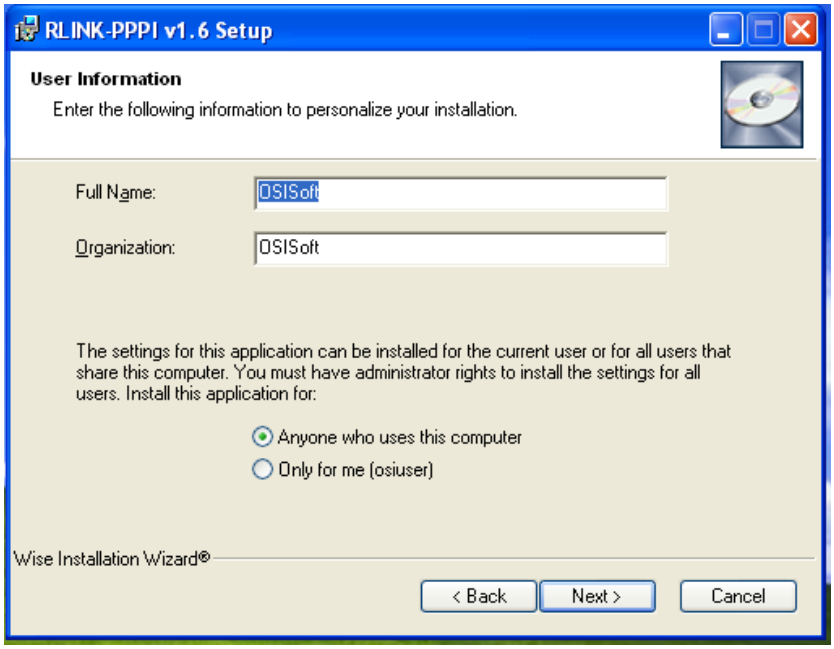
1.Start the installation by Double-Clicking the executable.
2.Click “Next”



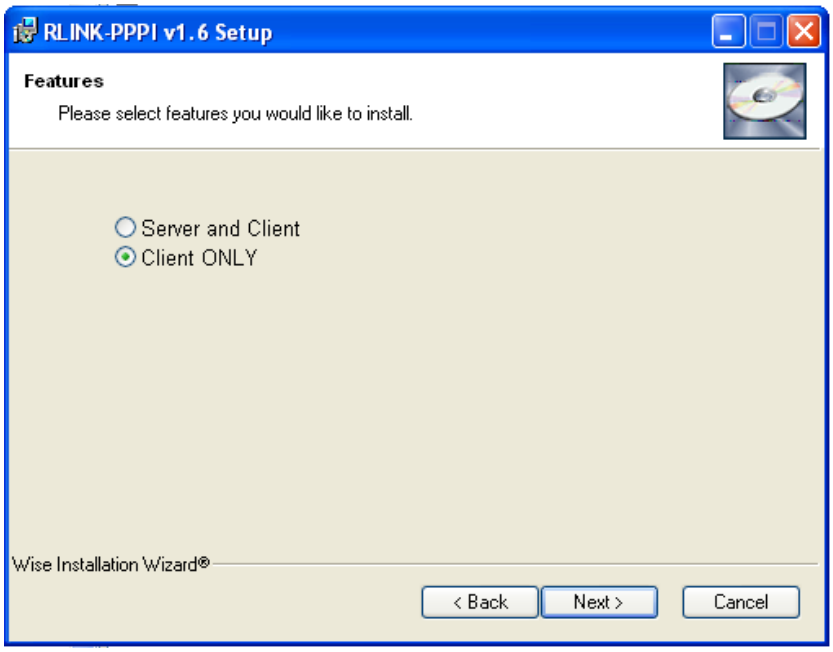
3.Select “I accept the license agreement” and Click “Next”



4.Set appropriate settings and click “Next”



5.Select “Client ONLY” and click “Next”



6.Provide the RLINK Database Information as follows:

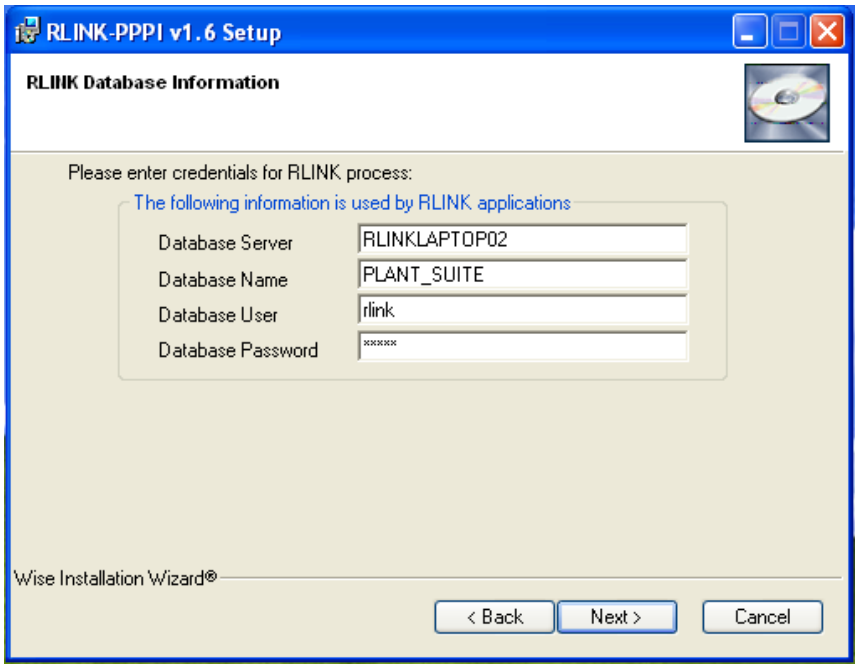
RLINK Applications:

Database Server: name of the computer that has the database (i.e. the SQL Server name)

Database Name: the name of the database (created in Step 2 of "SQL Server SAP Database Configuration" documentation).

Database User: the database user (created in Step 8 of ""SQL Server SAP Database Configuration" documentation).

Database password: the password for the database user above.

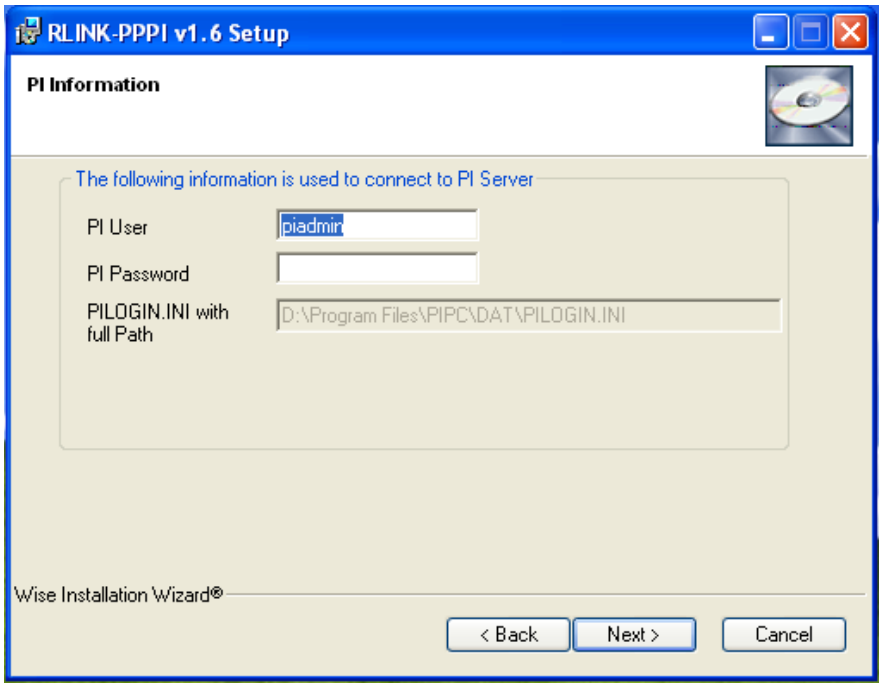


Click “Next”

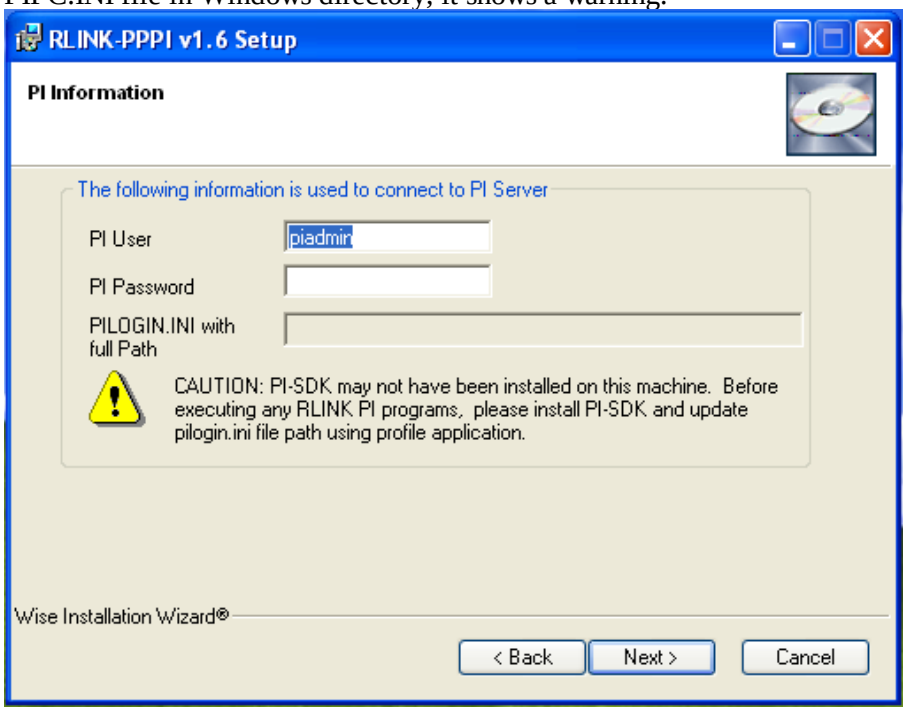
7. Provide the PI information as follows:
PI User: PI user with read/write access

PI password: password for user above.

PILOGIN.INI with full Path: The setup reads PIHOME value in PIPC.INI from Windows directory. PI-SDK creates the file PIPC.INI in Windows directory.



8.If the setup program could not find PIHOME value in PIPC.INI or could not find PIPC.INI file in Windows directory, it shows a warning.



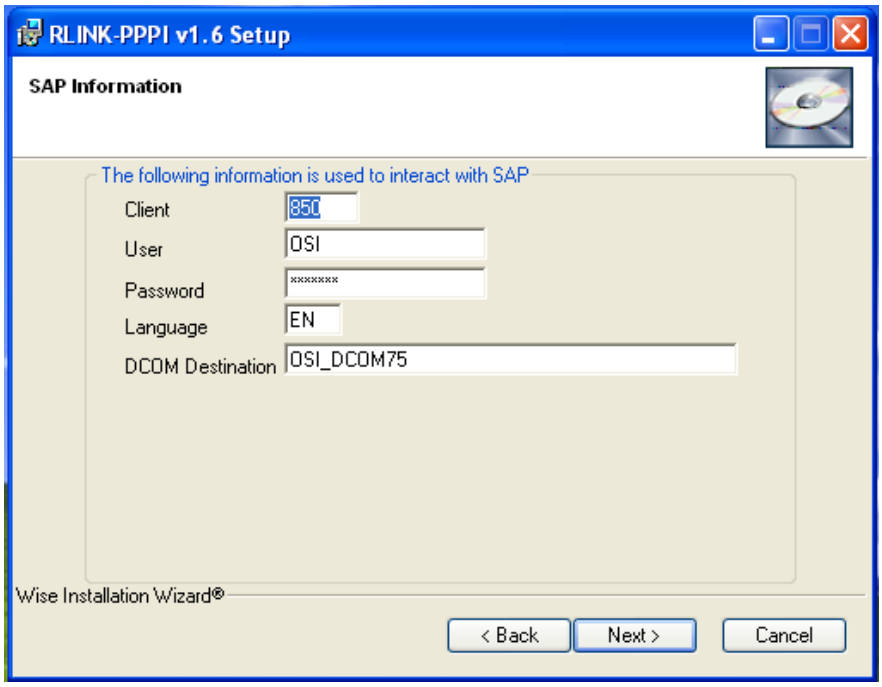
9.Refer to the SAPlogon and your sm59 transaction destination to provide SAP information:
Client: corresponds to client in the SAPLogon

User: user that can connect to SAP

Password: password for the user above.

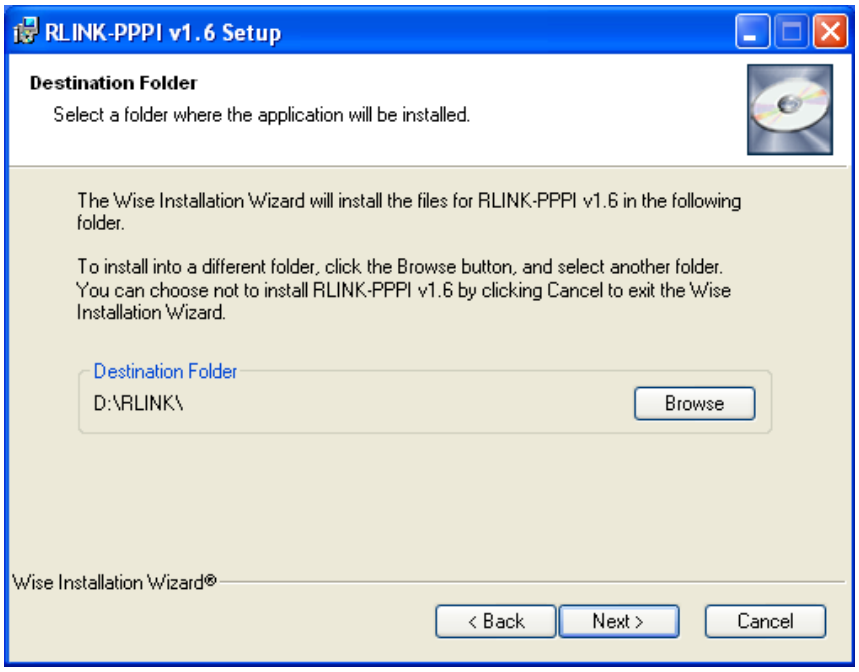
Language: the correct Language for your regional settings

DCOM Destination: corresponds to DCOM Connector Destination



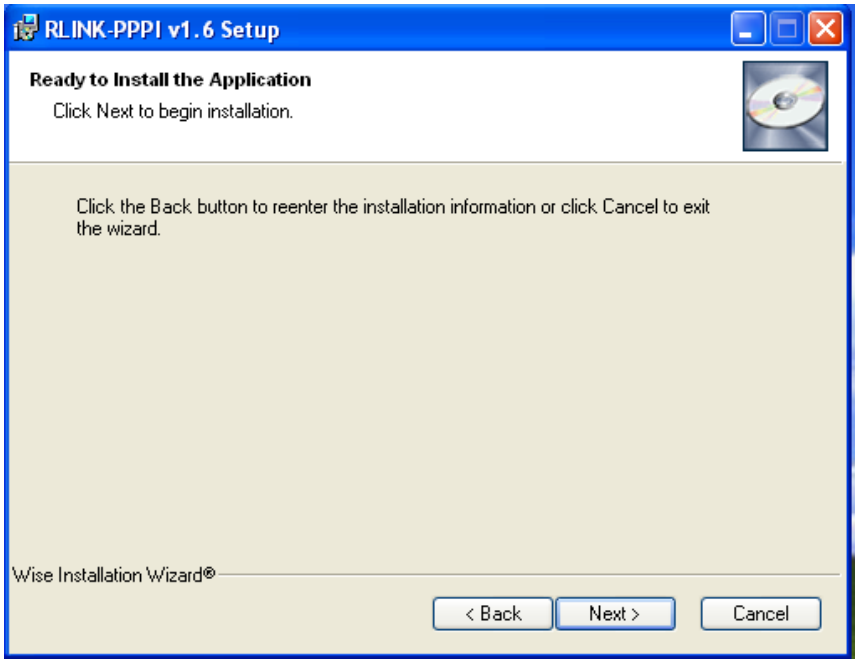
Click “Next”

10.Select the Destination Folder you wish to have RLINK installed:

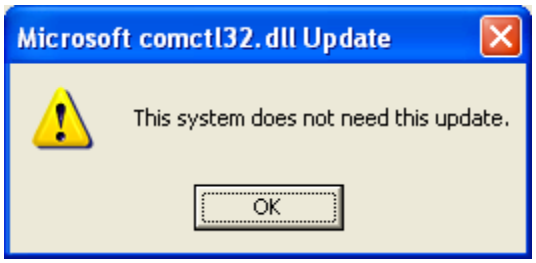


Click “Next”

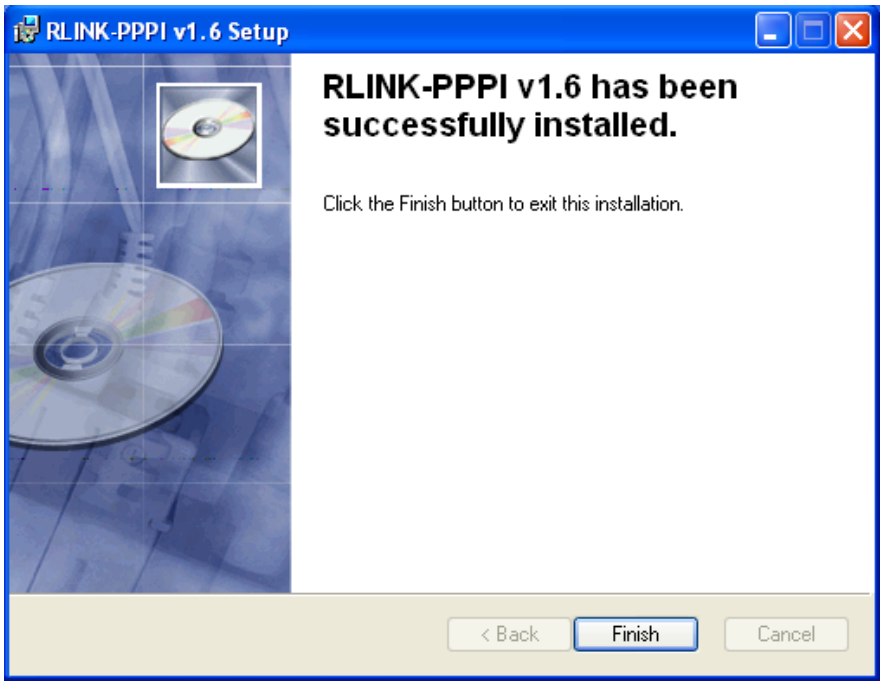
11.Click “Next” to continue installation.



12.Click “OK” to the Microsoft comctl32.dll Update warning (this warning is Okay):



13.Click “Finish” to complete installation process.



This setup is only run on a client machine, for the server this portion was installed with the server install. The purpose of this section is to load the front-end executables, modified the path to include RLINK\shared and register the ocx's. A client machine must have installed the PI-SDK and it should be properly configure if you are going to use the PITag portion of the client tools. The client machine should also have Microsoft Access installed. The ODBC should be configured and the profile application for RLINK should have the plant information tab completed.

ODBC

Setup ODBC to access the database plant_suite using the ODBC icon from the Control Panel. Call the ODBC Data Source Name PSRLINK and set the options to the database plant_suite and the correct server machine. This should be done for the client and the server machine.

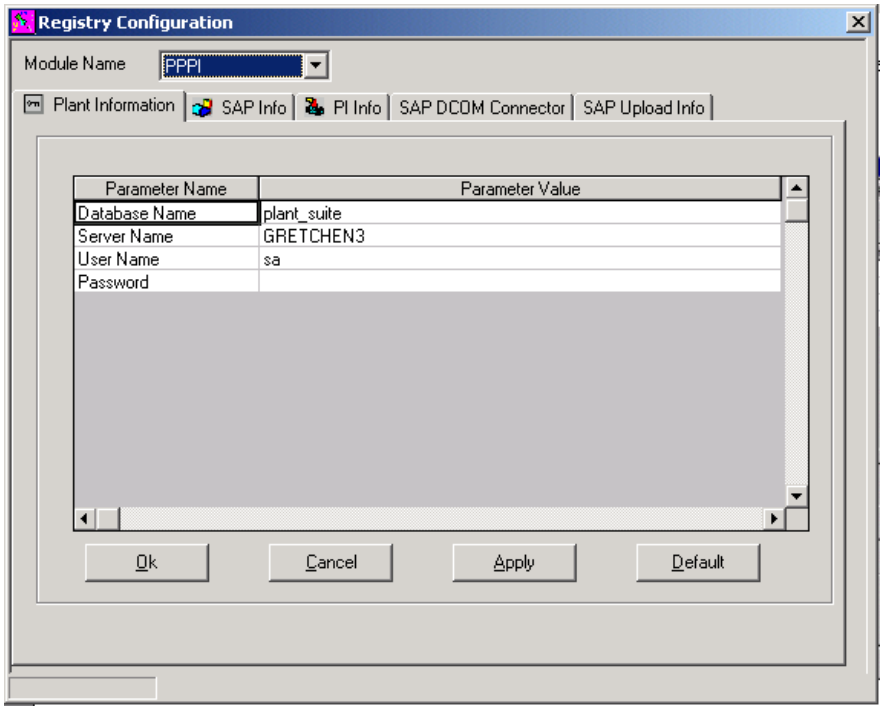
Turn off Nulls padding warning and check the perform translation if working on a foreign language computer setup.

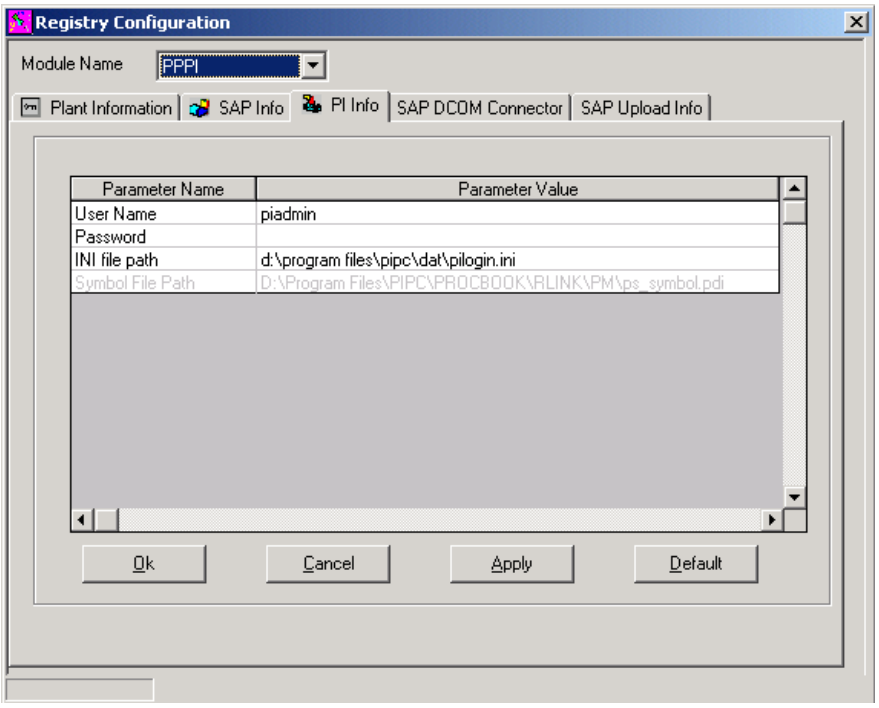
Verify that the ODBC has be configure to use TCP/IP not Named Pipes.

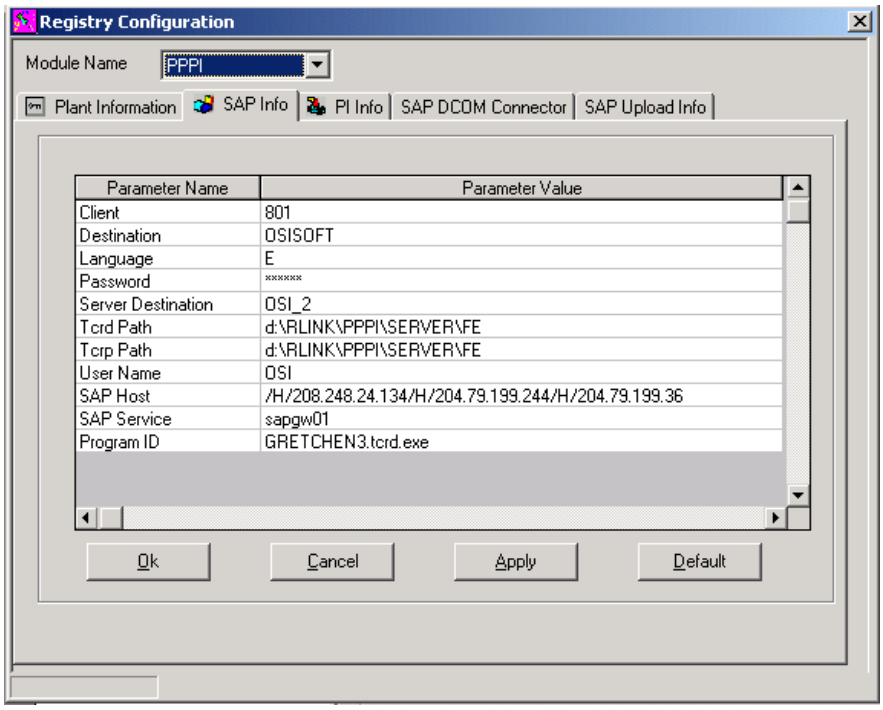
Registry Modifications

The application RLINK Profile will setup the Registry data requirements for the RLINK application. The initial registry entries are created during the installation.

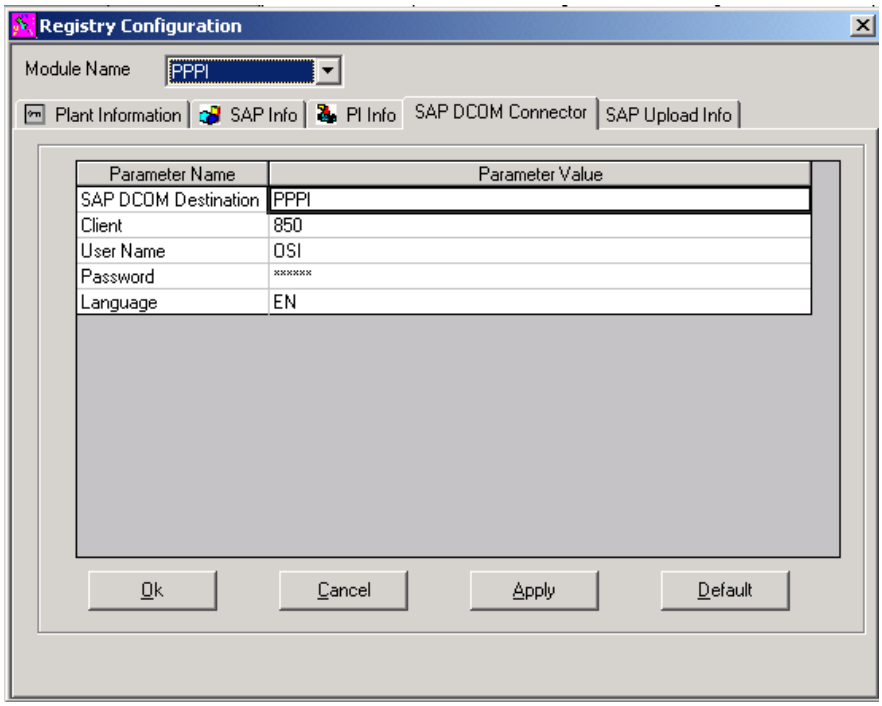
This must be done for the client and for the server machine


Registry Configuration
Module Name
PPPI
Plant Information
SAP Info
PI Info
SAP DCOM Connector
SAP Upload Info
Parameter Name
Parameter Value
Database Name
plant_suite
Server Name
GRETCHEN3
User Name
sa
Password
Ok
Cancel
Apply
Default


Registry Configuration
Module Name
PPPI
Plant Information
SAP Info
PI Info
SAP DCOM Connector
SAP Upload Info
Parameter Name
Parameter Value
User Name
piadmin
Password
INI file path
d:\program files\pipc\dat\pilogin.ini
Symbol File Path
D:\Program Files\PIPC\PROCBOOK\RLINK\PM\ps_symbol.pdi
Ok
Cancel
Apply
Default



If you are using the BAPI's then the SAP DCOM Connector tab must be configured



PS_RLINK

PI_INFO

PI_INIFILEWITHPATH C:\PIPC\DAT\PILOGIN.DAT

PI_PASSWORD password for account with privilege to modify points

PI_USERNAME account with privilege to modify points

PLANT_SUITE_INFO

DBNAME fixed plant_suite

PASSWORD	password for logon
SERVERNAME	server name for the SQL
USERNAME	username to logon to SQL server sapuser

SAP_INFO

CLIENT	SAP/R3 client machine
DESTINATION	fixed name for OSI software
LANGUAGE	E
PASSWORD	password for the SAP/R3 logon
SERVER_DEST	SM59 destination
TCRDPATH	path for RFC files set in install, this is Also the location of the service log files C:\RLINK\PPPI\server\fe
TCRPPATH	path for RFC files set in install C:\RLINK\PPPI\server\fe
USERNAME	SAP/R3 logon username (This user name should be setup with Profile in SAP as language English and delimiter “.”)
SAPService	SAPgwxx where the xx is the system number
SAP Host	Complete path to the SAP machine
PROGRAM ID	The program as configured in the SM59 transaction

SAP DCOM

SAP DCOM Destination	Destination from SAP DCOM
CLIENT	SAP/R3 client machine
USERNAME	SAP/R3 logon username (This user name should be setup with Profile in SAP as language English and delimiter “.”)
LANGUAGE	EN
PASSWORD	password for the SAP/R3 logon

SAPRFC.INI File

During the install the file \RLINK\PPPI\SERVER\FE\SAPRFC.INI is created and saved. The \RLINK\SERVER\PPPI\FE\SAPRFC.INI file must contain an entry for the Registered SM59 destination and for the client logon applications. Two examples are shown below for each. The entries that need some modification for the client are ASHOST and SYSNR. The fields that need to be edited in the Registry are DEST, PROGID, GWHOST and GWSERV.

After this file has been edited it must be placed in the location assigned by the environment variable in the Windows system environment. The **environment variable RFC_INI is the location of this file for example c:\RLINK\PPPI\SESRVER\FE\saprfc.ini.** If you are going to be running the services

this file must be in the path, WINNT\system32 directory. It should also be copied to the c:\RLINK\PPPI\CLIENT\FE directory.

Sample 1 for OSISOFT uses the tcp/ip address for the ASHOST

DEST=OSISOFT

TYPE=A

ASHOST=/H/192.187.177.196/H/204.79.199.2/H/204.79.199.4

SYSNR=01

RFC_TRACE=0

ABAP_DEBUG=0

USE_SAPGUI=0

Sample 2 for OSISOFT uses the name of the ASHOST

DEST=OSISOFT

TYPE=A

ASHOST=spn01int

SYSNR=00

RFC_TRACE=0

ABAP_DEBUG=0

USE_SAPGUI=0

Sample 1 for the Registered SM59 uses the explicit tcp/ip address

DEST=OSI_PI2

TYPE=R

PROGID=GMS_1.tcrd.exe

GWHOST=/H/192.187.177.196/H/204.79.199.2/H/204.79.199.4

GWSERV=sapgw01

RFC_TRACE=1

Sample 2 for Registered SM59 uses the machine name

DEST=OSI_PI2

TYPE=R

PROGID=GMS_1.tcrd.exe

GWHOST=spn01int

GWSERV=sapgw00

RFC_TRACE=1

System Environment Variables

The environment variables are setup during the installation. They include RFC_INI which was already discussed and TRFC_TRACE. The environment variable TRFC_TRACE and set its value to 0 for off. If the value is set to 1 and the trace flag is set in the SAPRFC.INI file then a file will be generated tr*.trc with some trace information.

Services for TCRD, PSRLINK and SAPPOLL

During the install process three services were installed but not started. These services are TCRD, PSRLINK and SAPPOLL when you are sure that communications is working correctly with SAP/R3 and are ready to go into production mode you can start the services. You should start only the sappoll and the psrlink services using the Control Panel, services. The TCRD service should not be started instead it will be started by the sappoll service. When the services are running the output from TCRD and PSRLINK is written to the log files tcrd*.log and psrlink.log and sappoll.log and are located in \RLINK\PPPI\SERVER\FE. Other commands that are useful in working with the services are

Servicename -R to remove the service

Servicename -I to install the service

Servicename -Debug to debug the service

The purpose of the SAPPOLL service is to verify that SAP/R3 is still up and operational. This service will wake up and attempt to log onto SAP/R3 at a regular time interval. If it cannot logon and TCRD is executing then it will shut down TCRD. When it can successfully logon again it with start the TCRD service. The time interval used to execute SAPPOLL is set in the configuration application. This application stores the values in the System_parameter table. You will not see the tcrd service show started in the services window on the control panel but if you look at the task manager you should see it start as a task under the process tab.

To get a log file of the PSRLINK servce you set the parameter on the PSRLINK service. This is done by going to the control Panel, Servcie for the PSRLINK service and then enter the parameter –Debug D in the parameter line and start the service from the same dialog. The trace file will be written to psrlink.log. You do not want to leave this running for an extensive time because the file will get rather large.

In order to get a trace file for the SAPPOLL service the system environment variable TRFC_TRACE value must be assigned to 1

TRFC_TRACE = 0 (will Not create sappoll.log)

TRFC_TRACE =1 (will create sappoll.log)

After you change the trace flag for the environment variable you must reboot the machine for it to take effect.

A copy of the SAPRFC.INI file should be placed in the winnt\system32 directory.

To enable trace options right click on the service and select the properties from the menu, and pass <-Debug D> in the Startup Parameter and then click the start button. Make sure that the saprfc.ini is available in winnt\system32 directory
The log files will be generated in RLINK\PPPI\SERVER\FE directory if the trace option is enabled. The log files for sappoll, tcrd and psrlink are sappoll.log, tcrd<number>.log and psrlink.log respectively. Make sure the entries in your saprfc.ini are in sync with the profile application. I fyou need a trace on tcrd then you will have to start it first before

sappoll and put the –Debug D on the parameter line of TCRD service and start it from the same dialog.

It is recommended that you only switch to the service operation after you see that all processes are running correctly by using the PSRLINK and Download options from the menu and DOS window.

Menu Initialization

The install procedure creates menu entries in the program files under RLINK PPPI.

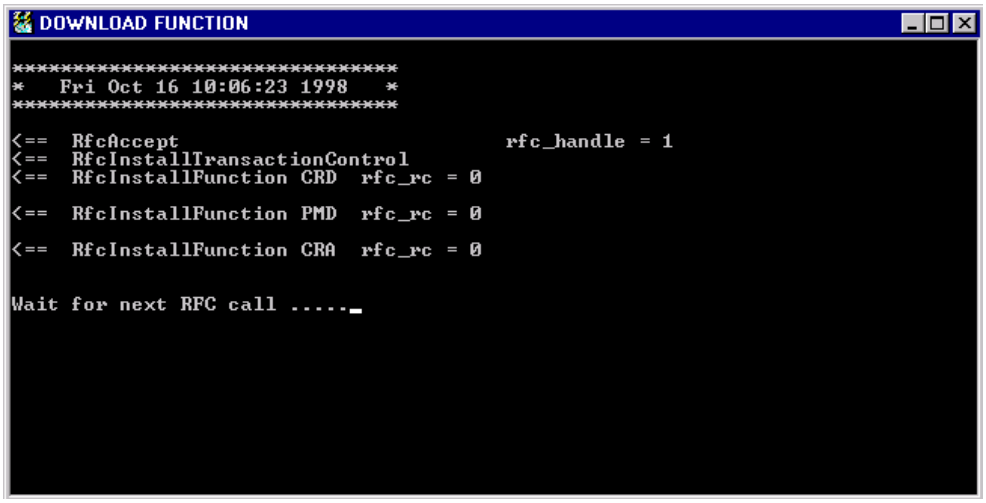
Testing the Link with SAP R/3

At this point you are ready to test the connection to SAP R/3. In a DOS window start the program tcrd.exe with the following command

Tcrd.exe –DXXXXX

Where XXXXX is the name of the SM59 destination that was configured. You should get a handle=1 to the SAP system.

From the SM59 transaction for the entered destination select Test Connection. If the machines have been setup correctly this will test the network link connection.



Upgrade Install

The installation CD will also contain an Upgrade directory. If you are doing an upgrade there are usually two parts. The first you must move any files which are shipped in to the appropriate directories for each of the following:

rlink\PPPI\client\fe

rlink\ PPPI\server\fe

rlink\shared

rlink\ services

If you are shipped a new version of the service you must uninstall the service and then install the service again.

If you are shipped any OCX files or DLL files there will be a readme file naming the files to be registered. These use regsvr32 to do the registration.

If the activeX EXE spprckbk is included in the patch then this must be registered and un-registered as follows:

spprckbk.exe / unregserver

spprckbk.exe / regserver

The second part is requires that you move what is shipped in the psrlink\server\be to a new directory, name it with the upgrade name. Then you must run the upgrade.bat file for upgrading the database with changes.

If you copy from a CD change the property of the files removing the read-only property.

After you have updated the database there might be shipped procedures that will update existing plants. For example in the release to upgrade to 1.3.5 there is a procedure usr_sync_iniparm that must be run from the Microsoft Query Analyzer for SQLServer.

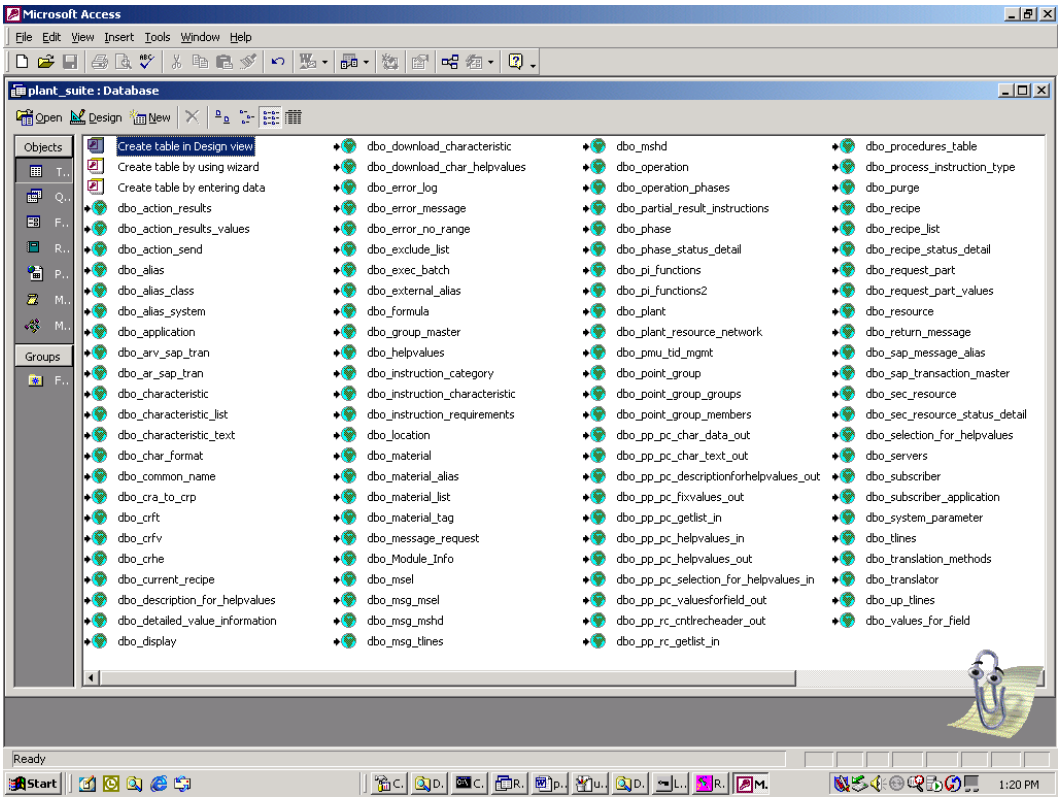
If you are migrating from an earlier install remove the entry for the destination after the program TCRPS.exe. The information for the destination is now located in the registry.

With version 1.6 the date formats in action_send and action_results tables used to set and retrieve values from PI has been modified from DD MMM YYYY HH:MM:SS to YYYY-MM-DD HH:MM:SS format. In order to update your existing database a procedure must be run. Backup the database and then run the procedure usr_update_olddate2iso 'PPPI' then verify that the dates are in the correct format in action_results, action_send tables, ar_sap_tran, pp_gm_ar and pp_ar_rem.

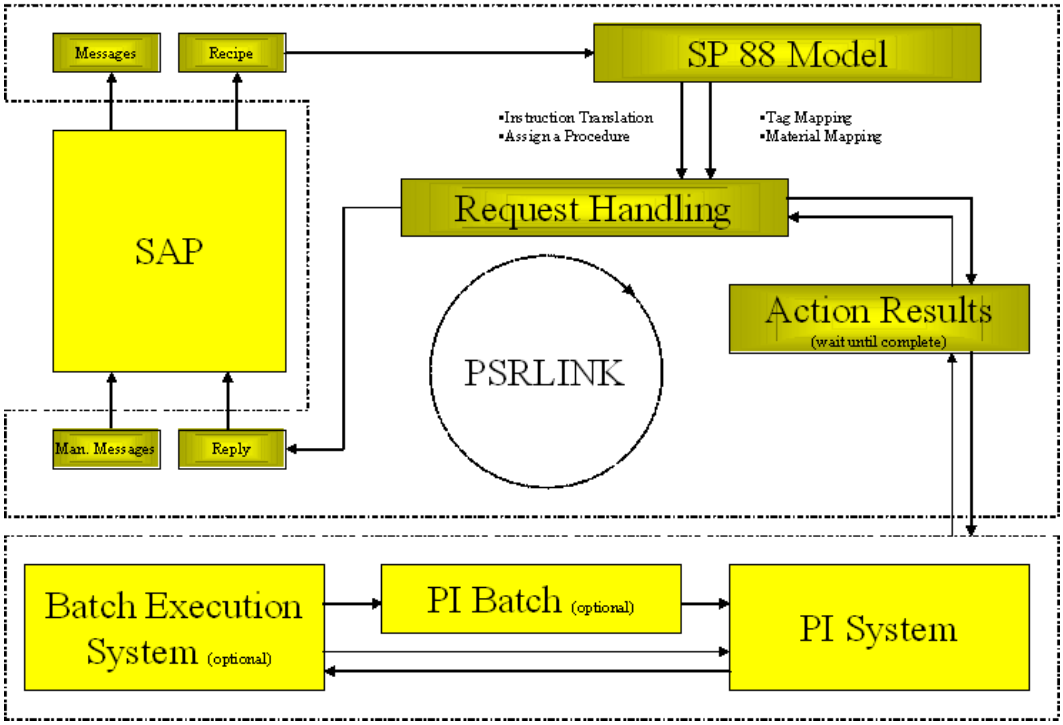
After completion of an update script verify the table rlink_version for the information about what release level you are at.

Database Table Initialization

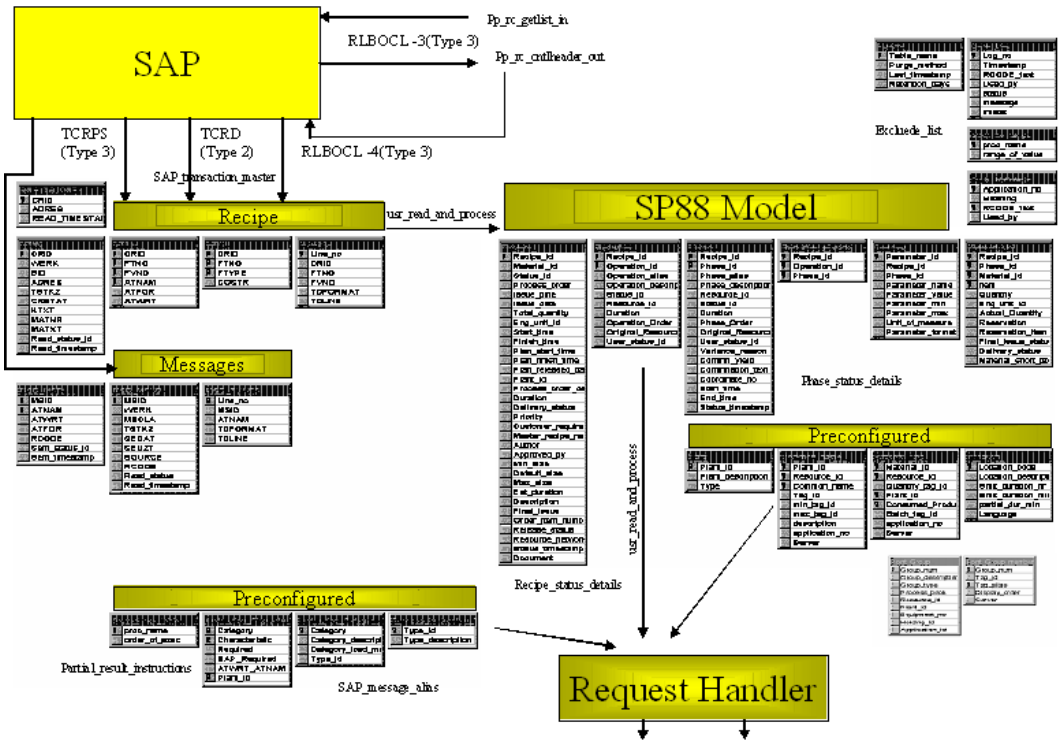
Microsoft Access can be installed to look at the data in the Plant Suite Server database, however it can be configured using the configuration application. If the ODBC driver has not been set up this can be done by using the Control Panel ODBC option. The server is SQL Server and the database is plant_suite. Set up an Access view into the SQL tables shown on the following screen by using the link option for an ODBC data source. The majority of these tables can be configured using the configuration application. The following diagrams show the flow of information during the processing of recipes.



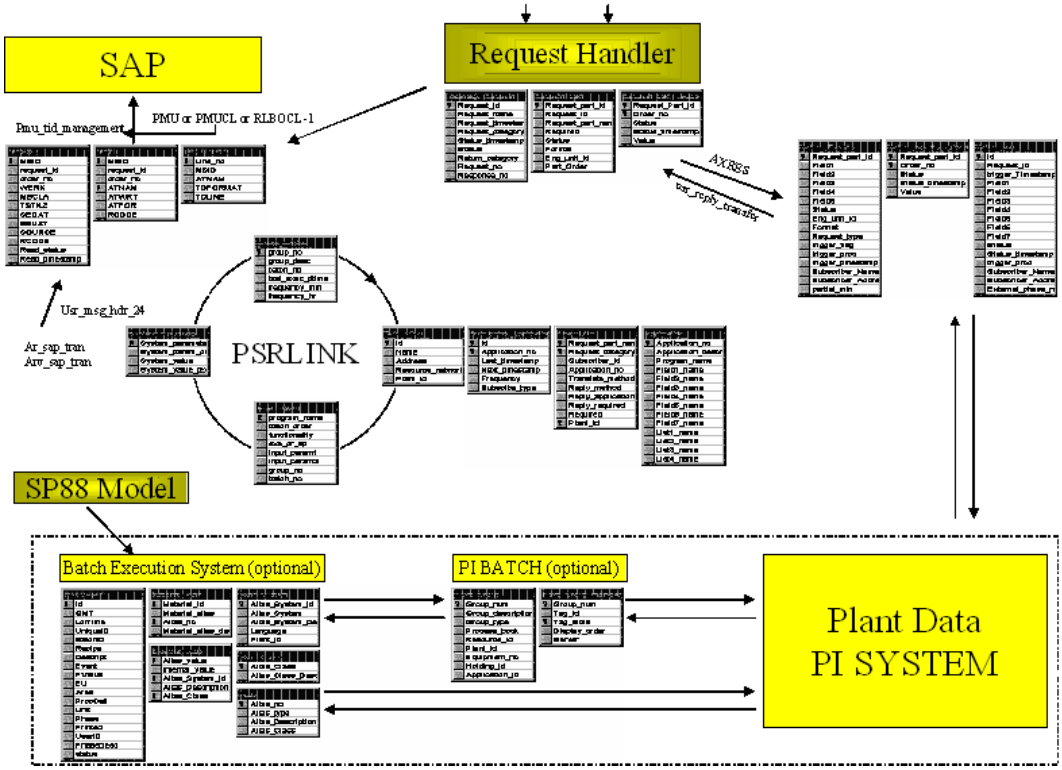
RLINK Gateway to SAP R/3



RLINK Gateway to SAP R/3 - Data Flow (1)



RLINK Gateway to SAP R/3 - Data Flow (2)



PSRLINK executes programs that have been configured in the group_master and exec_batch tables. These tables come loaded with the applications that are delivered. If the site wants to remove some applications from the list that will not be used at that site these tables can be edited. If the user would like to add a program for their site that is to be scheduled for routine execution they can also use this functionality.

- Program execution environment tables

There is functionality within the system to schedule program execution. Programs are scheduled by defining groups of programs to be executed. Each group can consist of multiple batches and each batch can consist of multiple programs. The PSRLINK task starts the programs in this group to continuously execute. Each program group can have a frequency of execution and a sleep time can be set before you check again if there are programs to be executed. There currently is a limit of 64 entries in the group_master table which are supported. The PI programs have been written with a trace option to enable debugging when needed. To turn on this trace option you would change the entry in exec_batch to have the parameter "-T" after the program name for all programs except the combined pimod.exe. For the pimod.exe you can set trace on for the individual program by using the pi_functions table and placing a T under trace_flag for the program you want to write a log for. The debugging information is written to a file with the name of the executable and type txt in the root directory. If you do not want a program to be executed from within pimod but rather as a separate exe then remove its name from the table pi_functions.

There are two execution queues that are provided. The first uses the service psrlink with the table group_master and exec_batch. This is usually all that is required. If the user has other programs that are taking a long time to execute and are independent of the usual RLINK processing then they can be configured in the second queue which uses the table group_master2 and exec_batch2 and the service psrlink2.

dbo_pi_functions : Table

order_no	func_name	param1	trace_flag
1	GETSNAPSHOT		T
3	GETTAGRANGE		T
5	GETSUMMARY		T
6	GETDSUM		T
7	PUTSNAP		T
8	MULTIVAL		T
9	GETDIFF		T
10	GETDIFFWAIT		T
11	GETTAGRANGEWAIT		T
12	MULTIVALWAIT		T
13	GETTAGWAIT		T
14	GETINPVALUE		T
15	GETINPWAIT		T
16	SUMMARYWAIT		T
17	DSUMWAIT		T
18	GETTAGJ		T
19	GETTAGJI		T
20	DELIVERY		T

Record: 18 of 18

In the exec_batch table there is an active column that must be set for which programs are to be run. Since there are multiple options for downloading and uploading data to SAP the method chosen must be selected as active and the others de-activated.

The program clchrval.exe that supports the SAP function for download of allowed characteristic helpvalues is included in the group_master and exec_batch tables. If you are not on at least SAP 4.5 then this should be flagged as in-active in the exec_batch table.

Group_master

Table Field	***Meaning***
Group_no	No. for the group
Group_description	Description of program group to be executed
Batch_no	Batch _no within the group, a group can consist of multiple batches
Last_execution	Last execution time for program group
Frequency_min	No. of minutes to next execution (with version 1.5 decimal minutes is supported)
Frequency_hr	No. of hours to next execution

Exec_batch

Table Field	***Meaning***
Program_name	Program to be executed. If this is an executable it must have the .exe extension present
Batch_order	Order within batch of the program
Functionality	Description of what the program does
exe_or_sp	E or P standing for executable or stored procedure
group_no	No of group
Batch_no	Batch _no within the group, a group can consist of multiple batches
Active	Y or Null is action or N if the progam is in-active

Table Field	***Meaning***
Program_name	Program to be executed. If this is an executable it must have the .exe extension present
Batch_order	Order within batch of the program

Table Field	*Meaning*
Functionality	Description of what the program does
exe_or_sp	E or P standing for executable or stored procedure
group_no	No of group
batch_no	Batch _no within the group, a group can consist of multiple batches

ICON Setup Misc. Tasks

The install creates menu additions in the program menu. You must edit the properties on the entry for rlink\PPPI\server\download functions to include the destination as set up in the SM59 transaction. A typical entry would be as follows where dest is the name in the SM59 transaction.

C:\RLINK\PPPI\SERVER\FE \tcrd.exe -Ddest

Purge Install

Use SQL Enterprise Manager to install the Purge utility.

Configuring RLINK purge Utility on SQL Server 2000

- In the enterprise manager go to your server, management, SQL Server Agent, Jobs
- From Action Menu at the top left, select New Job
- On the New Job Properties, General Tab, Enter the name as “Purge Plant_suite”
- For category, choose “Databes Maintenance” from the list
- For Owner, type either “sa” or any valid SQL User who has permission to delete and update records in the plant_suite database.
- For description, type “Runs a stored procedure topurge processed recipe data”
- On the next tab Step , press “New” button
- On the New Job Step dialog, for step name type “Run stored procedure”
- For Type select “Transact-SQL Script” from the list
- For database choose “plant_suite”
- For command, type “usr_purge” and press OK Button
- On the next tab Schedules press “New Schedule” button
- For Name type “Purge Schedule”
- For Schedule type select “Recurring” option and then press “Change “ button.
- Schedule according to the volume of data. Typically, schedluling purge utility to run once a week is ideal. Please make sure that for Duration “No end date” option is chosen. Press “OK” button on Edit recurring Job Schedule dialog. Then press “OK” button again on “New Job Schedule” dialog. Then press “Apply” to save the changes.

- Optional: By default if the job fails it logs into “Windows Appliation Log”. In addition if you need to inform someone by email or page use the next tab called Notifications and configure the same.
- The day after this is scheduled to run check the status of the Job. If the lasat run is unsuccessful, check the “Windows Application Log” for further details.
- For repetitive manufacturing there will be an entry in the purge table called usr_rem_purge with the duration date for these entries.
- For material movements there will be an entry in the purge table called usr_mm_purge with a duration for these entries.
- SQL Server Agent must be up and running in order to run the Jobs created in SQL Server.

Check that the SQLServer Agent service is started in the Control Panel Services.

The tables that are included in the purge are as follows:

Action_results	Characteristic_text	Pmu_tid_mgmt
Action_result_values	Ledger_history	Recipe
Action_send	Material_list	Recipe_status_detail
Cra_to_crp	Messge_request	Request_part
Crft	Msel	Request_par_values
Crhe	Msg_msel	Sec_resource
Crfv	Msg_mshd	Sec_resource_status_detail
Description_for_helpvalues	Msg_tlines	Selection_for_helpvalues
Detailed_value_information	Mshd	Tlines
Download_char_helpvalues	Operation	Up_lines
Error_log	Operation_phases	Values_for_field
formula	Phase	Sap_transaction_master
Helpvalues	Phase_status_detail	Characteristic_list
Pp_pc_helpvalues_in	Pp_pc_descriptionforhelpvalues_out	Pp_pc_char_data_out
Pp_pc_fixvalues_out	Pp_pc_helpvalues_out	Pp_rc_cntlrecheader_out
Pp_pc_valuesforfield_out	Pp_pc_selection_for_helpvalues	Pp_pc_char_text_out
Mm_gm_cfd_gmcode_in	Mm_gm_cfd_head_01_in2	Mm_gm_cfd_itemcreate_in3
Mm_gm_cfd_s1no_in4	Mm_gm_arv	Pp_rm_datgen_in
Pp_rm_components	Pp_rm_dataorder	Pp_rm_dataserial_in
Pp_rm_flag_in	Pp_arv_rem	

The purge is based on the recipe completion time and status of the recipe being 0004 (Terminated), 0005 (Processed) or 0007 (Discarded before started).

Edit Exec_Batch

Verify that the path of all programs in the table exec_batch are set correctly.

You must verify the status of each individual application to choose which ones will execute. A status of Y or null will cause the program to be executed, while a status of N will skip the program. The initial load has been chosen to be the most frequent configuration.

Server Status PI and SAP

In order to handle the case when there are known shutdowns of the PI servers or SAP with a reduced number of error messages bing generated we have added a table that handles the server status. Ther table is called servers. During the installation you should use Microsoft Access or Enterprise manager to enter the server name to this table.

Table Field	*Meaning*
Servername	Name of the piserver or SAP to represent the SAP instance
isrunning	Y or N to indicate if the server is running

The field in action_results that will correspond to the application and that will be tested for status is given in the table pi_functions2. This table does not need to be modified unless a new application is being added.

Table Field	*Meaning*
id	Unique identifier
Program_name	Program name
servercol	Filed name in action_results that will hold the piserver name

In order to implement the checking of the server status the system parmeter PISRV must be set.

Removing PSRLINK

If you wish to remove the RLINK-PPPI product from your machine then you must do the following:

- Use the Add/Remove programs from the control panel to remove RLINK-PPPI, select the option not to remove any shared programs
- Delete the database and database devices using Enterprise manager
- Remove the data files in the MSSQL\Data directory for the devices
- Delete the directory RLINK
- Remove the services with the service_name –r for SAPPOLL, PSRLINK and TCRD.

ProcessBook

A sample ProcessBook has been provided. This ProcessBook requires that the Microsoft ADO interface be installed on your machine. This can be done by downloading it from http://www.microsoft.com/data/mdac2.htm and select the entry which says “Microsoft

Data Access Components 2.0 Redistribution Typical Setup 6.2MB". The version of this as of the distribution time is included on the CD.

SQLServer Backup

- Steps for automating Scheduled Database dump

1. In Enterprise manager, select the server under console root-Microsoft SQL Servers-SQL Server Group.

2. Under Mangerment, right click on Backup and choose "New Backup Device" option.

3. In "Backup devices property - New Device" Dialog, select "file name" option and then type ps_db_dump in Name. (Note: Write down the full path displayed, which will be used later in "Job" creation)

4. Click on the server again and choose, "New ", "Job" option either from Toolbar or from the menu Action\New\Job.

5. In the general Tab,

Type the following:

Name: Plant suite database backup

Uncheck "Enabled" checkbox.

Choose "Database Maintenance" from Catagory combobox options.

Choose "sa" as owner.

then click on "steps" tab,

Select "New" Button

Type the following

Name: Saving the last database backup file.

Choose "Operating system command" from type combobox option.

Type "copy c:\program files\Microsoft SQL Server\mssql\backup\ps_db_dump.bak c:\program files\Microsoft SQL Server\mssql\backup\old_ps_db_dump.bak"

Press "Apply"

In the steps tab, select "New" again.

Type the following:

Name: Dumping the database

Choose "Transact SQL script" from type combobox option.

Choose "plant_suite" from database option.

Type "BACKUP DATABASE plant_suite TO DISK = 'C:\Program files\Microsoft SQL Server\mssql\backup\ps_db_dump.Bak' WITH INIT"

in command box.

(Note: Give the full path as in step 3)

In the "Advanced" tab, clear the "Run as user" box.

Press "Apply"

then click on "schedules" tab and select "New Schedule".

Type the following:

Name: Schedule for dumping the database

Press "Change" button.

Choose "Weekly" from "Occurs" frame and tick "Sun" in "Weekly" frame.

Press "Ok" and then Apply.

6. Go to General tab and check the "Enabled" option.

- Steps for automating Scheduled Transaction log dump

1. In Enterprise manager, select the server under console root-Microsoft SQL Servers-SQL Server Group.

2. Under Management, right click on Backup and choose “New Backup device” option.

3. In "Backup devices property - New Device" Dialog, select "file name" option and then type ps_log_dump_??? in Name. [Where ??? stands for mon, tue, wed, thu, fri, sat]

(Note: Write down the full path displayed, which will be used later in "Job" creation)

4. Click on the server again and choose, “NewJob” option either from Toolbar or from the menu Action\New\Job. Select "New Job" option either from Toolbar or from the menu Action\New\Job.

5. In the general Tab,

Type the following:

Name: ??? plant_suite log backup

[Where ??? stands for Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]

Uncheck "Enabled" checkbox.

Choose "Database Maintenance" from Catagory combobox options.

Choose "sa" as owner.

then click on "steps" tab,

Select "New" Button

Type the following:

Name: Saving the last log backup file.

Choose "Operating system command" from type combobox option.

Type "copy c:\Program Files\Microsoft SQL Server\mssql\backup\ps_log_dump_???.bak c:\Program Files\Microsoft SQL Server\mssql\backup\old_ps_log_dump_???.bak"

[Where ??? stands for mon, tue, wed, thu, fri, sat]

Press "Apply"

In the steps tab, select "New" again.

Type the following:

Name: ??? transaction log dump

[Where ??? stands for Monday, Tuesday, Wednesday, Thursday, Friday, Saturday]

Choose "Transact SQL script" from type combobox option.

Choose "plant_suite" from database option.

Type "BACKUP LOG plant_suite TO DISK = 'C:\Program Files\Microsoft SQL Server\mssql\backup\ps_log_dump_???.Bak' WITH INIT"

in command box.

[Where ??? stands for mon, tue, wed, thu, fri, sat]

(Note: Give the full path as in step 3)

In the "Advanced" tab, clear the "Run as user" box.

Press "Apply"

then click on "schedules" tab and select "New Schedule".

Type the following:

Name: Schedule for dumping the database

Press "Change" button.

Choose "Daily" from "Occurs" frame and set "Occurs once at" option in "Daily frequency" frame.

Press "Ok" and then Apply.

6. Go to General tab and check the "Enabled" option.

7. Repeat this for creating transaction log dump for Monday through Saturday.

- Enabling SQL Server Agent in Services

1. From the control panel, select services icon.

2. Set focus on SQLServerAgent and press start button.

[If it is already started proceed to the next step]

3. Double click on SQLServerAgent and choose "Automatic" in "Startup type" frame, then press OK.

After Installation

- Backup of Master Database

This is done using SQL Enterprise Manager. Select Tools, Backup, New option, give device name as DUMP_DEV_MASTER, Destination as Disk.

In a similar the plant_suite database can be backed up after the tables have been loaded by selecting the DB for backup as plant_suite.

Optional SAP Gateway Installation

Some sites have found that they cannot rely on the SAPGateway that is installed with the SAP system. There is an option to create a SAPGateway on the same machine that the interface is running on. This should be done only if the provided SAPGateway is found not to be satisfactory. The software for creating a SAPGateway is located on the SAPGUI CD

1. install Gateway from the SAPGUI CD using the program r3gwinst.exe and the directions in the file r3gwinst.txt
2. set up the user account OSIADM with the password OSI, make it a member of the administrators group and give the home directory as c:\users\OSIADM
3. be sure that the option "User Cannot Change Password" is turned off and that password history in the account policies panel is off
4. Use file manager to give a share on the directory users\osiadm with "Full Control for Everyone"
5. Run the program r3gwinst give the name OSI for the SAP system name, password OSI, 0 as the answer to number of users, and take the default as the other options
6. If this completes successfully you should see a program group created and an entry for SAPOSI_00 in the control panel services option.
7. If this does not complete then remove the service by using

   ntscmgr remove SAPOSI_00
8. Install the Microsoft SNA service which can be found on the Visual Studio CD. The folder is SNA4. There is a server setup.
9. .If the service installed correctly then go to the directory usr\sap\osi\sys\exe and execute the program r3gwsideinfo
10. The saprfc.ini file should now be configured with the GWHOST = IP address of machine with gateway that you just loaded GWSERV=sapgw00
11. The SM59 transaction is configured with a gateway where the host is given with the path backwards from the SAP machine to the gateway. The service is sapgw00. The registration option is selected and the progam_id is give to match the one in the SAPRFC.INI file

Errors

- ISQL error MSG 4002 Level4

  c:\SQL60\bin\dbmpipe
- SQL error when executing the PMU program

  Check that you setup the environment variable for RFC_INI
- Unable to register server in Enterprise Management

  Check that the client configuration is using Named Pipes
- Unable to connect to SAP

  Check the registry, SAPRFC.INI file and verify case on SM59 transaction
- Services not working

  Check you moved the SAPRFC.INI to the WINNT\System32 directory
- ProcessBook not working with database connections

  Install the ADO interface from the web site given.

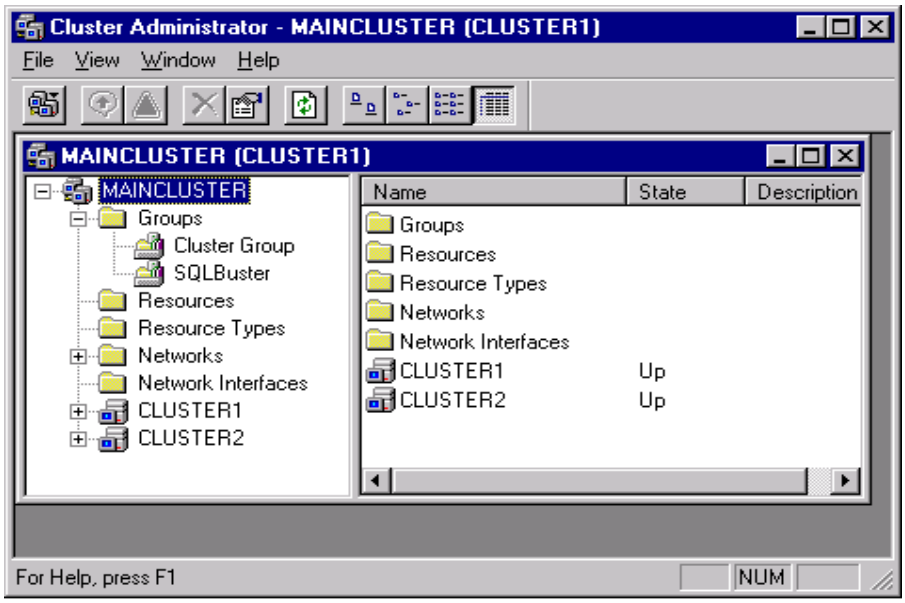
Cluster Support

RLINK has been installed on Windows NT 4.0 and Windows 2000 clusters. The type of failover that is configured is for CPU failure not service failure. Sometimes SAP will go down and the service TCRD will be stopped on purpose by the service SAPPOLL.

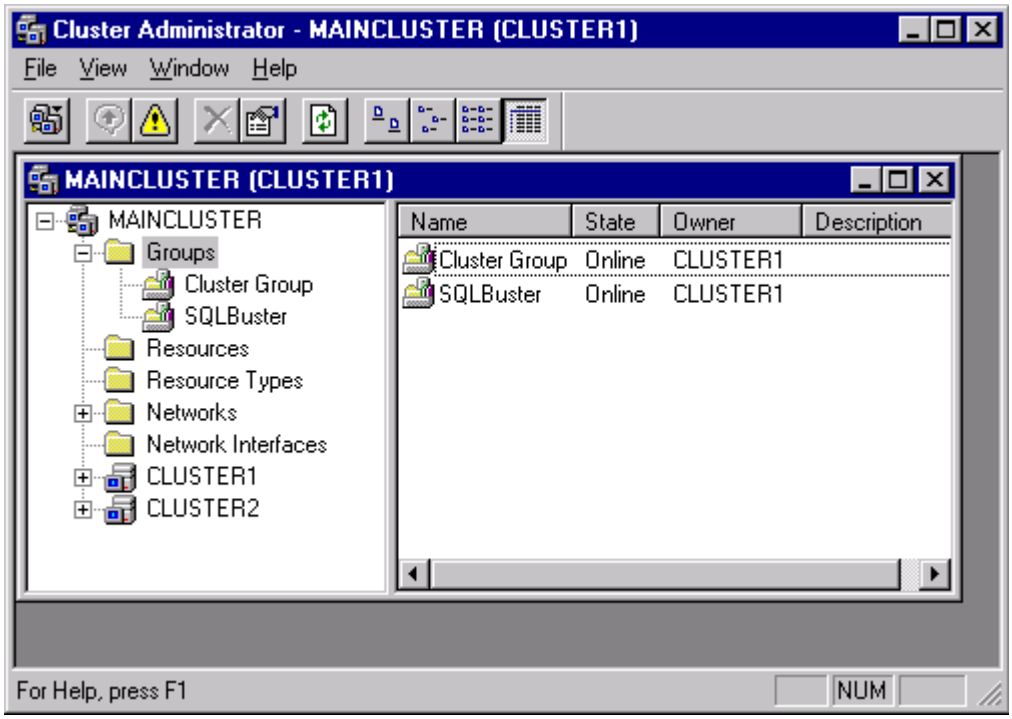
Installation instructions

Preparation of Cluster Group and Resources for RLINK

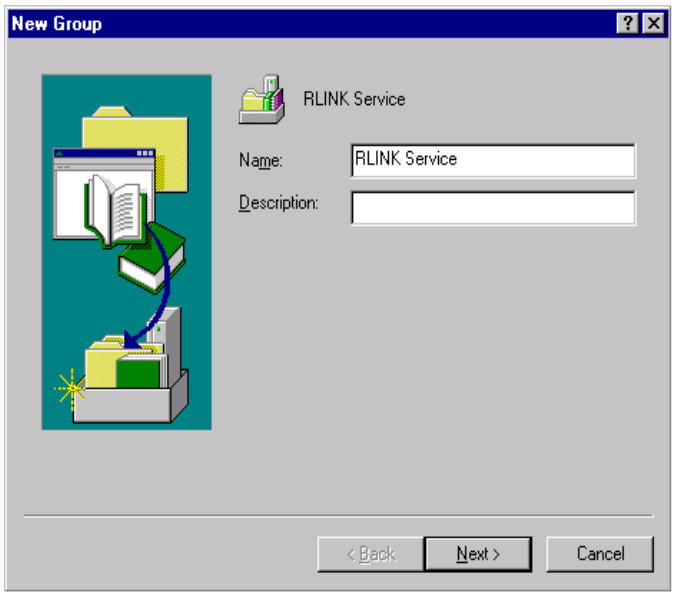
Start the Cluster Administrator on Node 1. On the Start menu, point to Programs, then point to Administrative Tools and click Cluster Administrator.



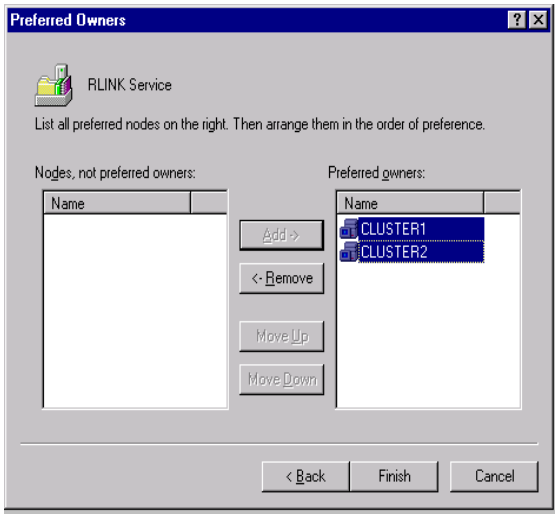
1. Your screen should now have a window similar to screen shot above.



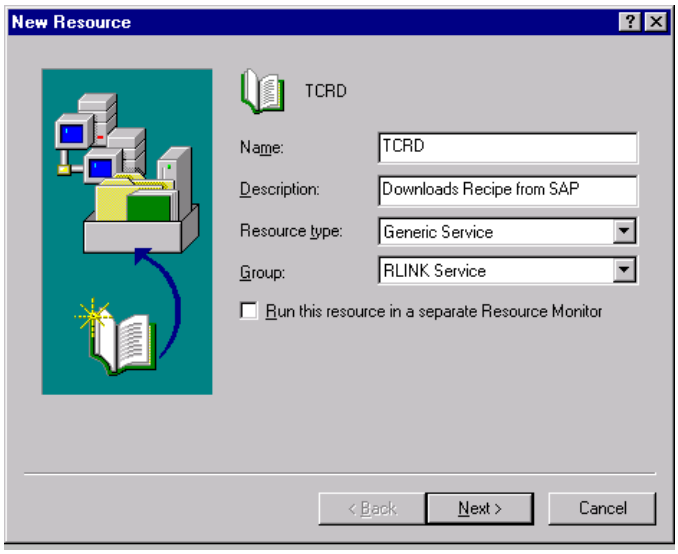
Right-click the Groups and then point to **New**, and then click **Group**



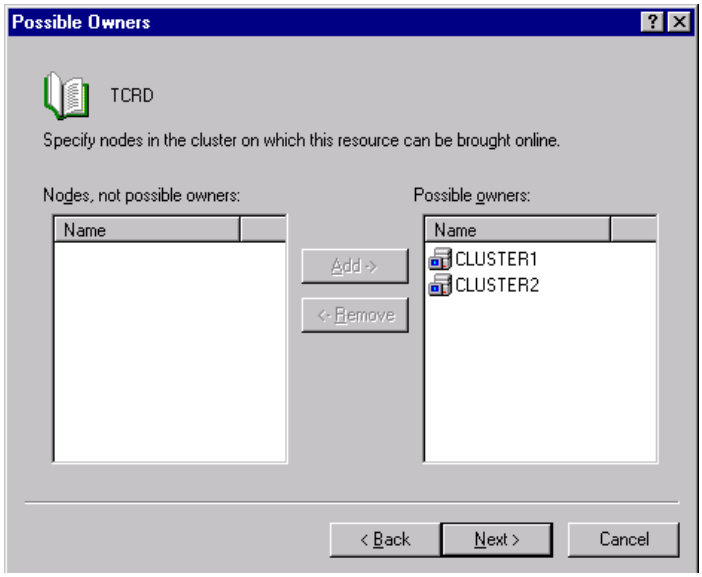
Type the Group name for example, RLINK Service. You can also type an optional description and then click **Next**.



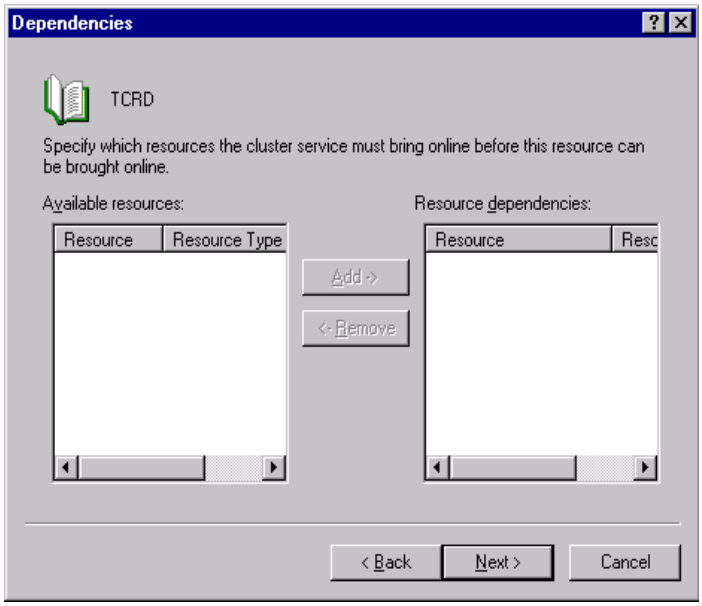
Creating TCRD Resource



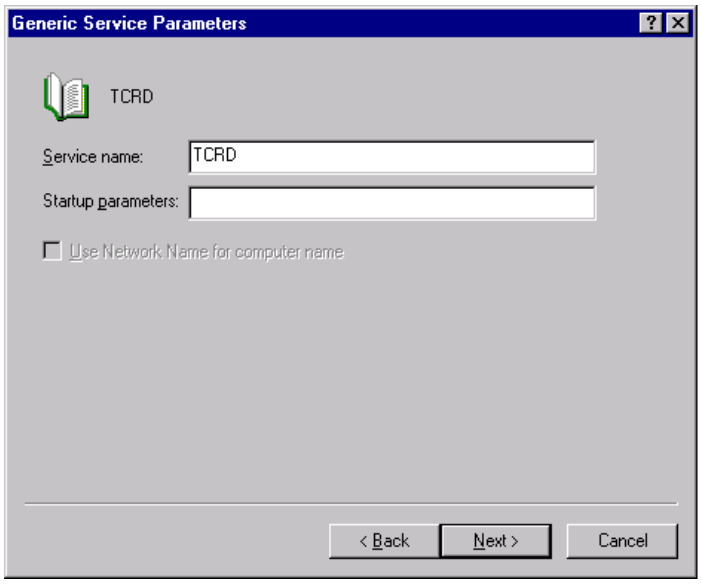
2. Type the resource name for the TCRD, for example,. You can also type an optional description. In the **Resource type** box, Generic Service, and then click **Next**.



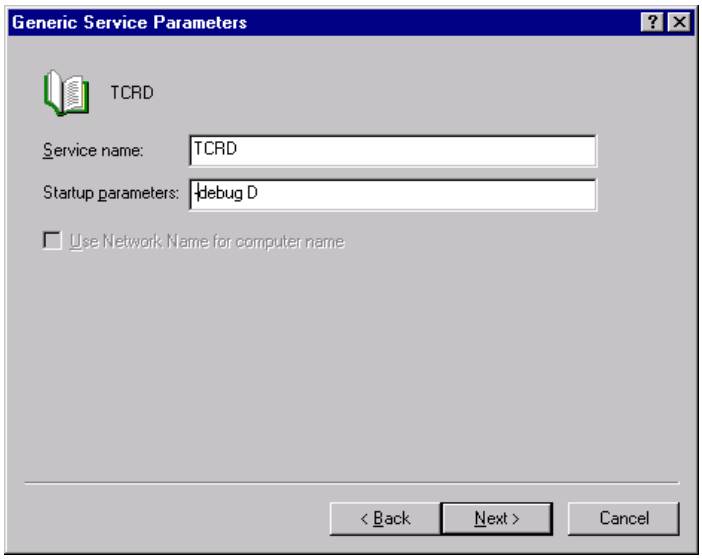
3. Both nodes appear as possible owners. Click **Next**.



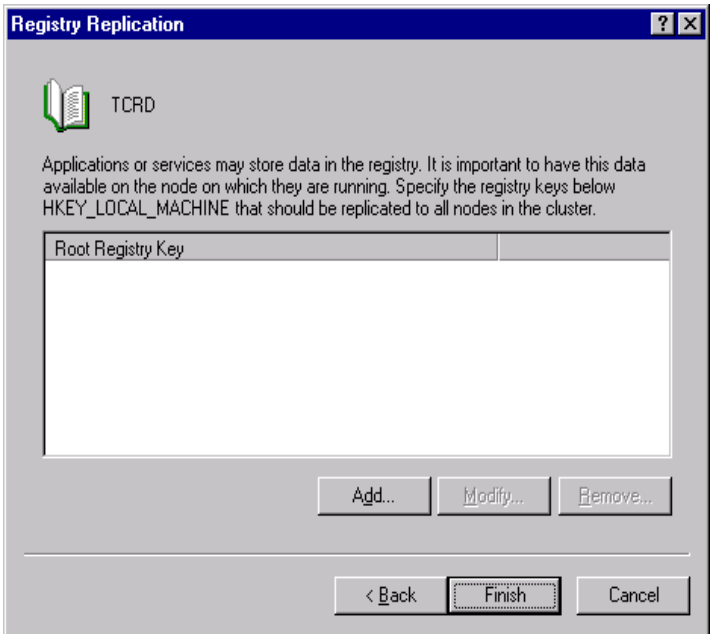
4. Click **Next**



5. Type TCRD as the Service Name

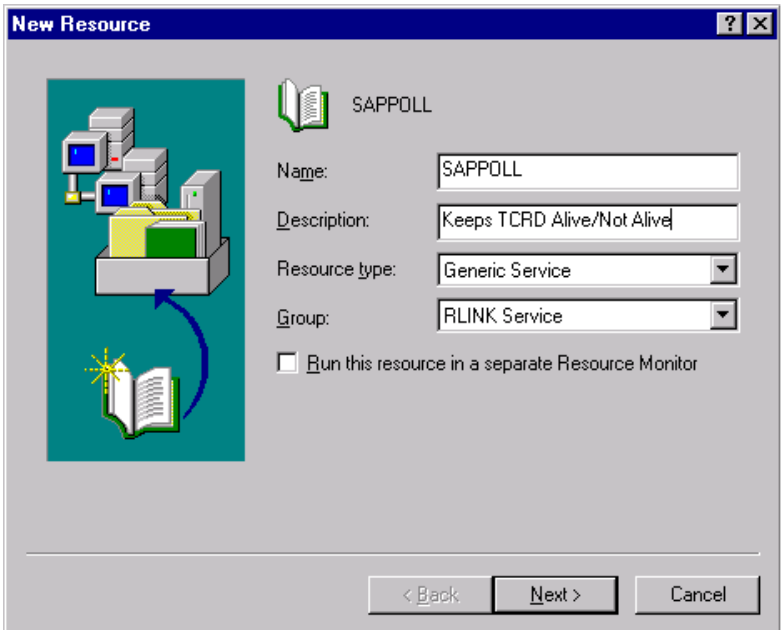


6. If you are doing to enable TRACE options then type –debug D in the Startup parameter.

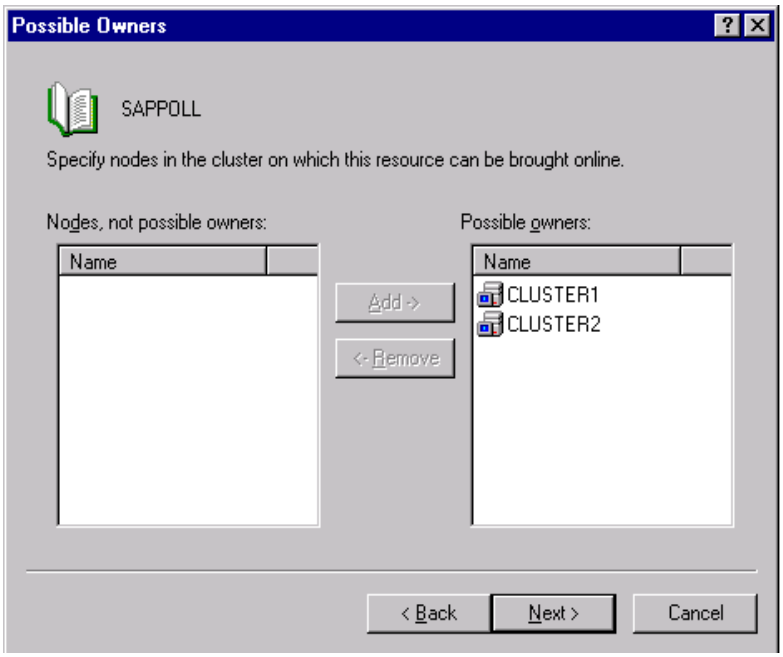


Click **Finish**

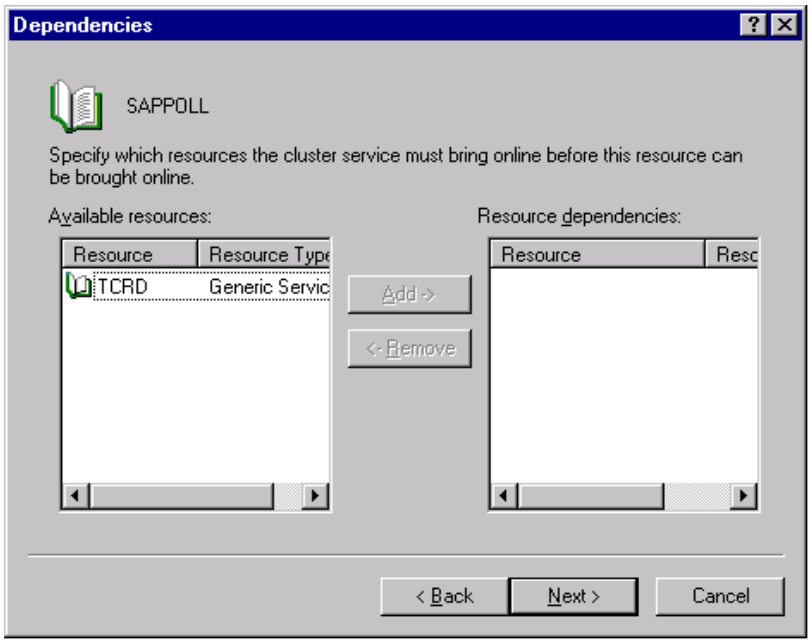
Creating SAPPOLL Resource



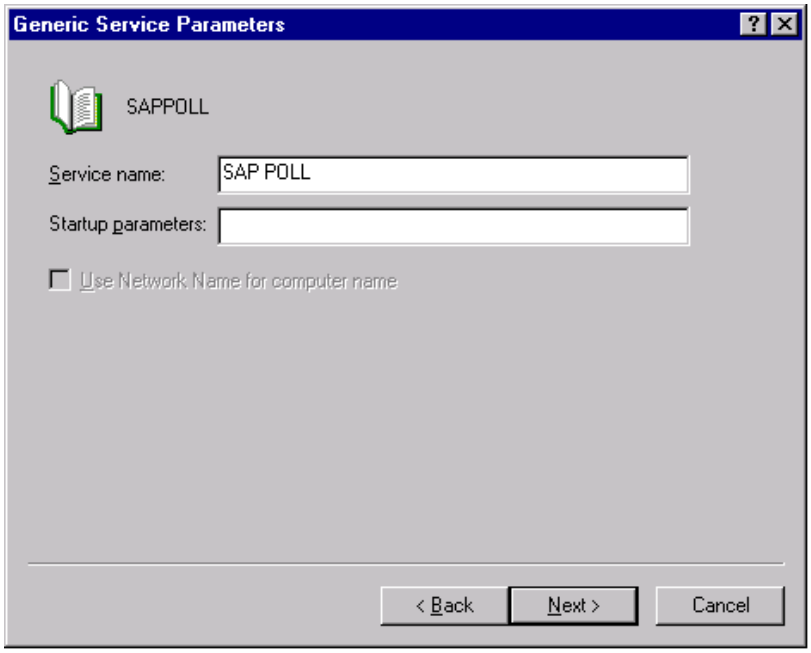
Type the resource name for SAPPOLL for example. SAPPOLL You can also type an optional description. In the **Resource type** box, Generic Service, and then click **Next**.



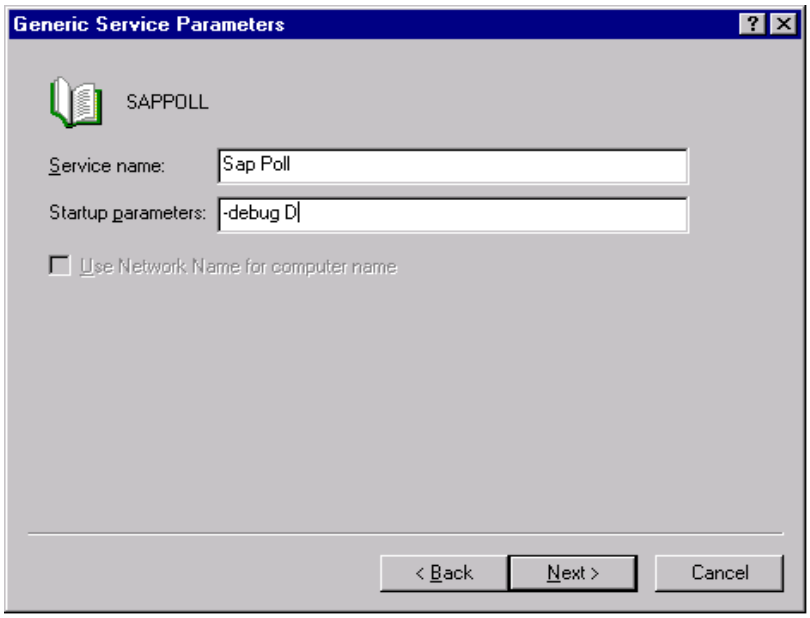
Both nodes appear as possible owners. Click **Next**.



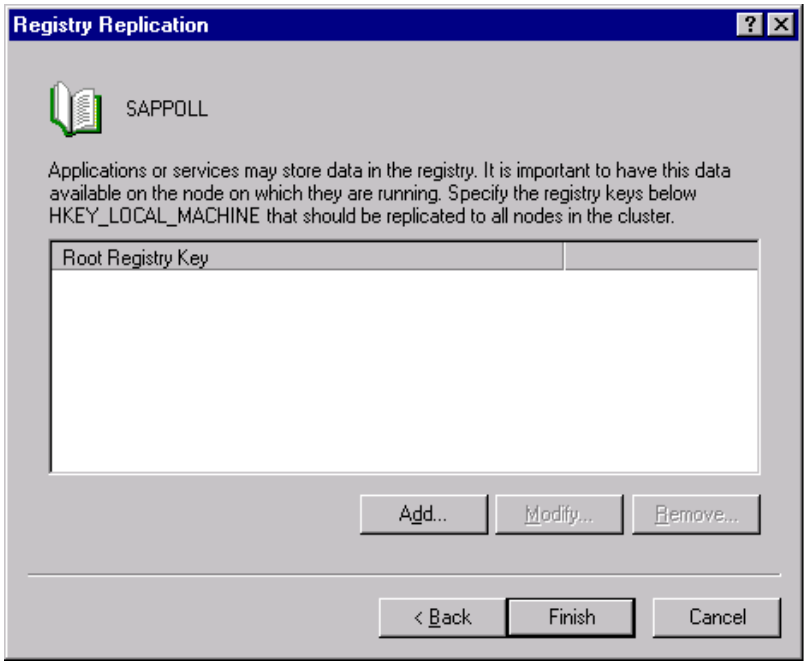
Click Next



Type SAP POLL as the Service Name.

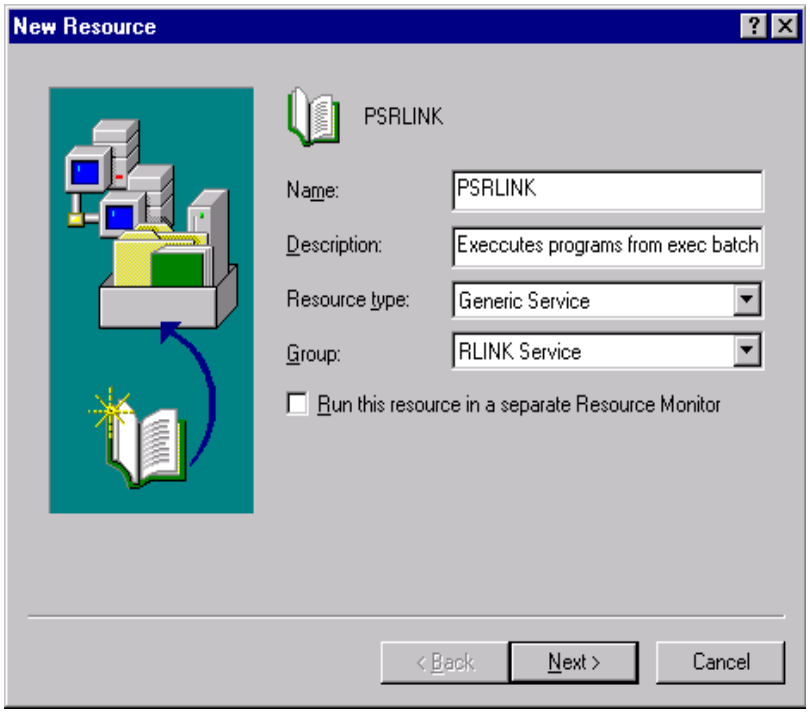


If you are doing to enable TRACE options then type –debug D in the Startup parameter.

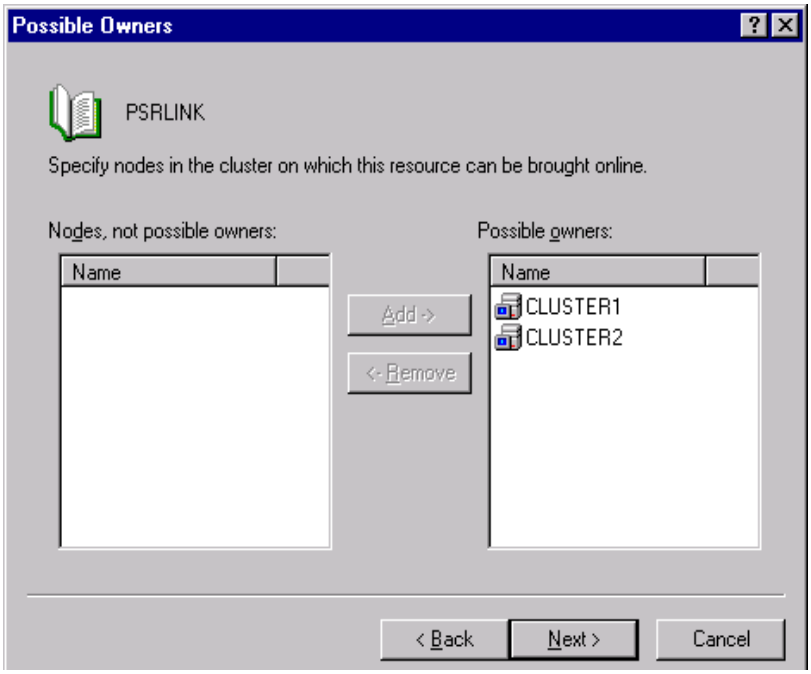


Click Finish.

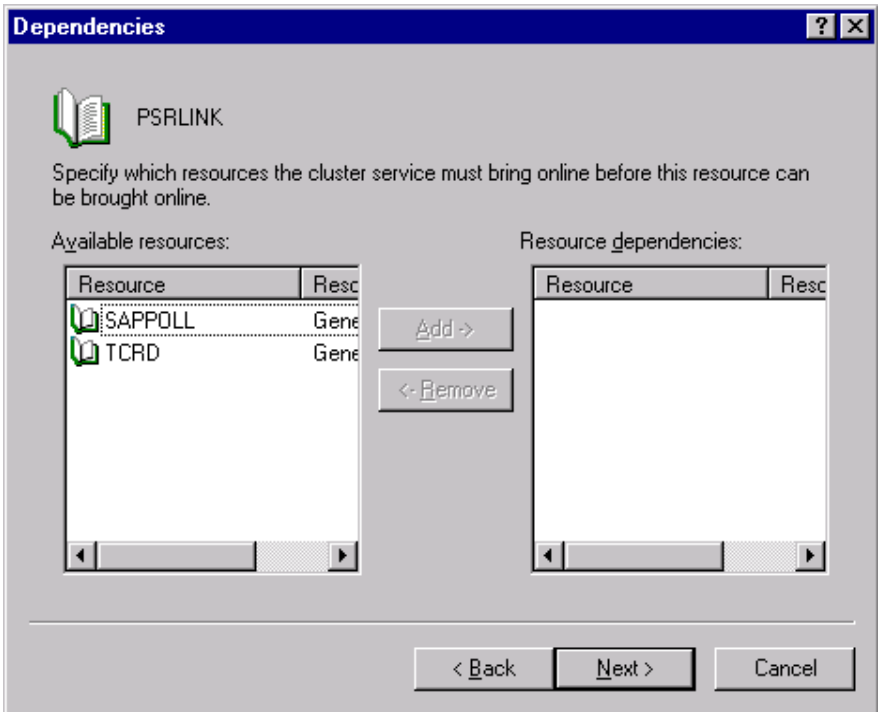
Creating PSRLINK Resource



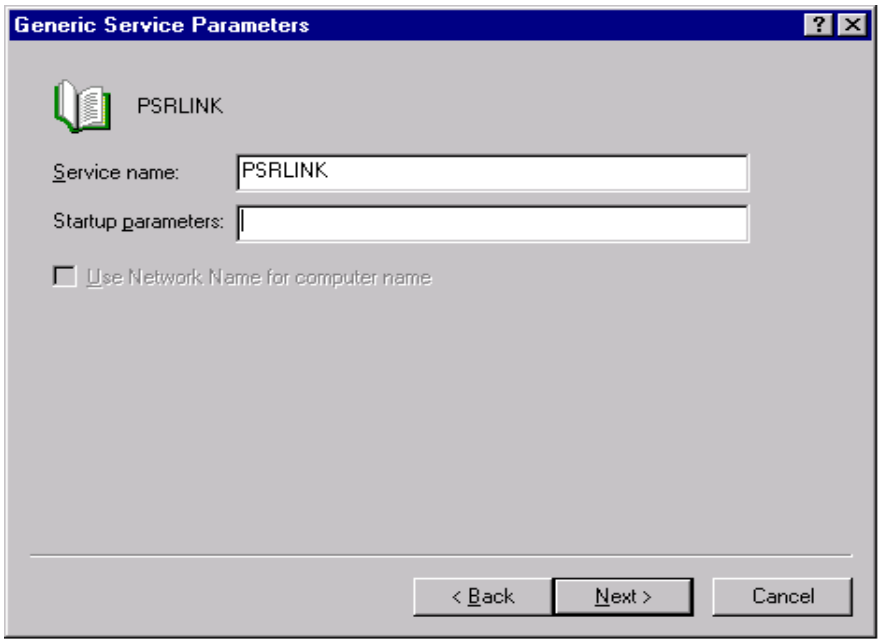
Type the resource name for PSRLINK for example. PSRLINK You can also type an optional description. In the **Resource type** box, Generic Service, and then click **Next**.



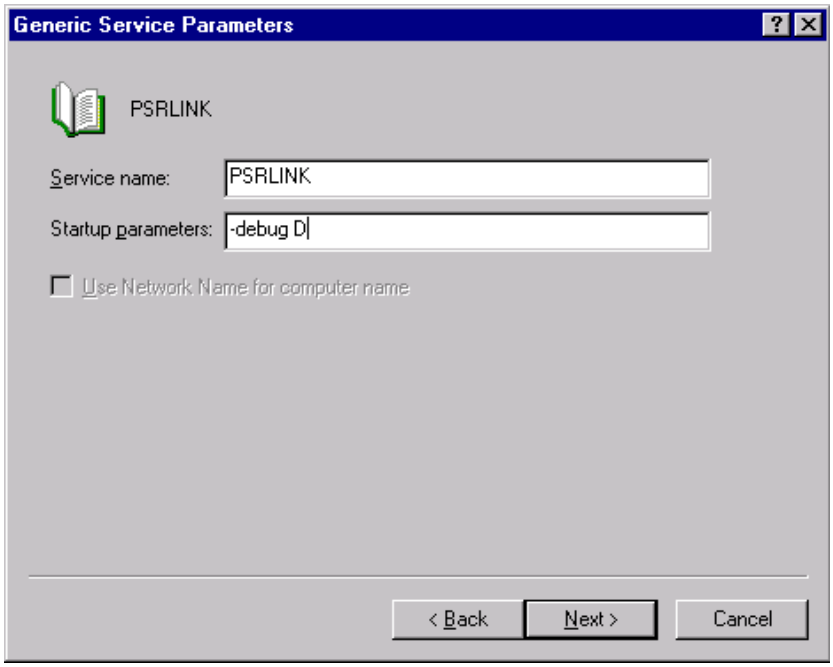
Both nodes appear as possible owners. Click **Next**.



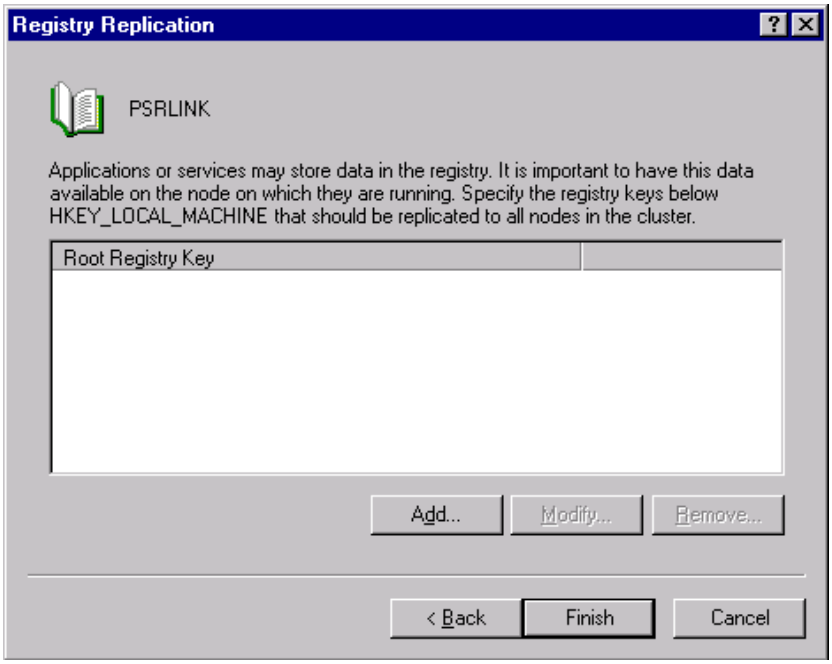
Click Next



Type PSRLINK as the Service Name.

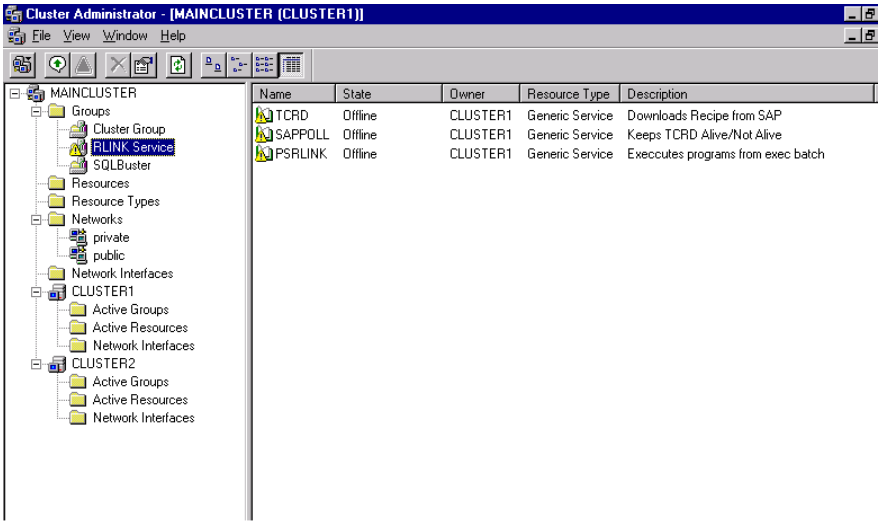


If you are doing to enable TRACE options then type –debug D in the Startup parameter.



Click Next.

RLINK on cluster should look like the one below which is offline

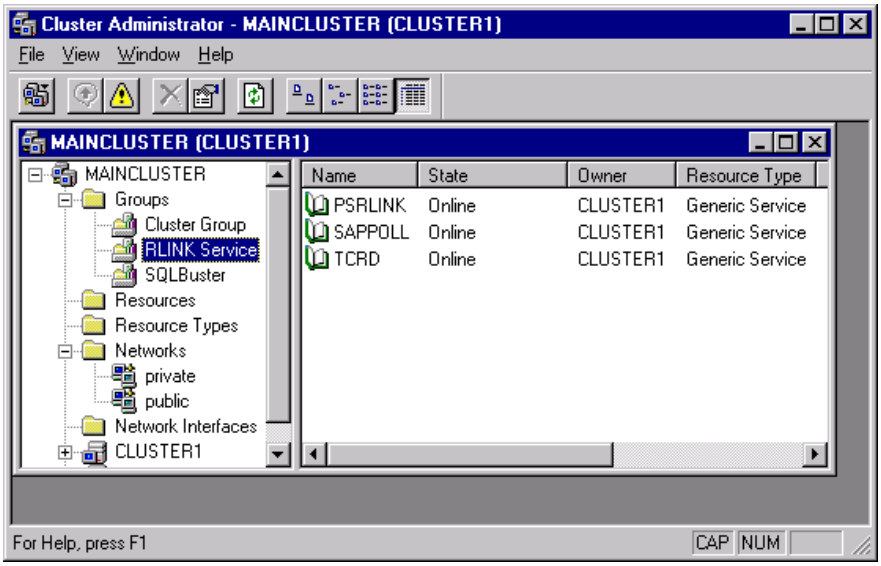


Your screen should now have a window similar to screen shot above.

Now point your mouse on the the SAPPOLL resource and right click a menu pops up in that choose ***Bring Online*** item. TCRD will be brought online after sometime when SAPPOLL becomes online depends on RLINK configuration.

Similarly point your mouse on the the PSRLINK resource and right click a menu pops up in that choose ***Bring Online*** item.

Finally your RLINK on cluster should look like the one below which is Online



To bring your resources/services to a halt follow the same instruction as you do to Bring Online but choose **Bring Offline** by right clicking on the popup menu.

Chapter 3
RFC and BAPI Functions

BAPI Programs

The BAPI calls are executed with the program ROBCL. This requires the SAP DCOM connector and Microsoft Transaction server. The input parameter will say which BAPI is executed. Running ROBCL ? will give the list of parameters and their function. Each is described below in their functional area.

RLBOCL -?	Displays various options in a message box
RLBOCL -1	Creates Process Messages in SAP
RLBOCL -2	Get Characteristics List
RLBOCL -3	Get Recipe List
RLBOCL -4	Recipe Request
RLBOCL -5	Get helpvalues

The BAPI's are located in RLINK\PPPI\Client\FE directory.

Process Message Upload

There are three methods for uploading process messages. The first method is a syncronous RFC call which uses the executable pmu.exe. The second method is a transactional rfc which is the executable pmucl.exe. The third method uses the SAP DCOM connector and transaction server and is a BAPI implementation that is executed with the program ROBCL –1.You must choose which of these options you are going to use in the exec_batch table.

The syncronous RFC call pmu will send up the message and wait for the RCODE from SAP. When the message is sent up to SAP the status in the MSHD table will be set to X. The status will be updated to S when the return code is received. The RCODE is returned by SAP and will tell you of any problems detected by SAP in the message. If there is a problem on the SAP side and no RCODE is returned the message will not be sent up to SAP again. An alarm will be set if you are using the alarming points for diagnostics. If you want to force this message to be sent again the change the status to blank.

The transactional RFC call pmucl operates in two steps. First the message is sent to SAP and a record inserted in the table pmu_tid_mgmt. Once a message is sent to SAP it will not be sent again. The RCODE is returned by a separated transactional job that will update the status to S and set the RCODE value returned from SAP. If you want to force a message to be sent a second time you must clean up the entry in the table pmu_itd_mgmt. If messages remain as sent to SAP with no return message for a

extended period of time then an alarm will be set if you are using the alarming points for diagnostics. If there is loss in communication the return message might be trapped on the SAP system in the RFC Environment on transaction CO54.

In the system_parameters there is a parameter for PMUWT that is the wait time before a message alarm will be created for missing RCODE for messages that have been sent to SAP.

You can see what the meaning of the RCODE values are by using the table error_message or the application MESSAGE CORRECTION from the menu.

- PMU_tid_mgmt Table

Table Field	***Meaning***
msid	Msid matches with MSHD table
Tid	Transactional id
status	Status of transmission to SAP
Sent_timestamp	Timestamp when message was sent to SAP

BAPI for Message Upload

The program which is executed is ROBCL –1. There is a new field added to the table MSHD as follows

Procc_mess_id char(18) Final Process Message ID from SAP

System messages will be written to the error_log tables. The messages include the following

Message Header

E	CB1	303	Errors occurred during message creation
E	CB1	200	Plant & does not exist
E	CB1	201	Process message category & 1 is not defin in plant &2
E	CB1	202	Enter a valid test indicator
E	CB	201	Further processing not possible
E	CB1	206	Unable to create message
S	CB1	308	Messages created successfully

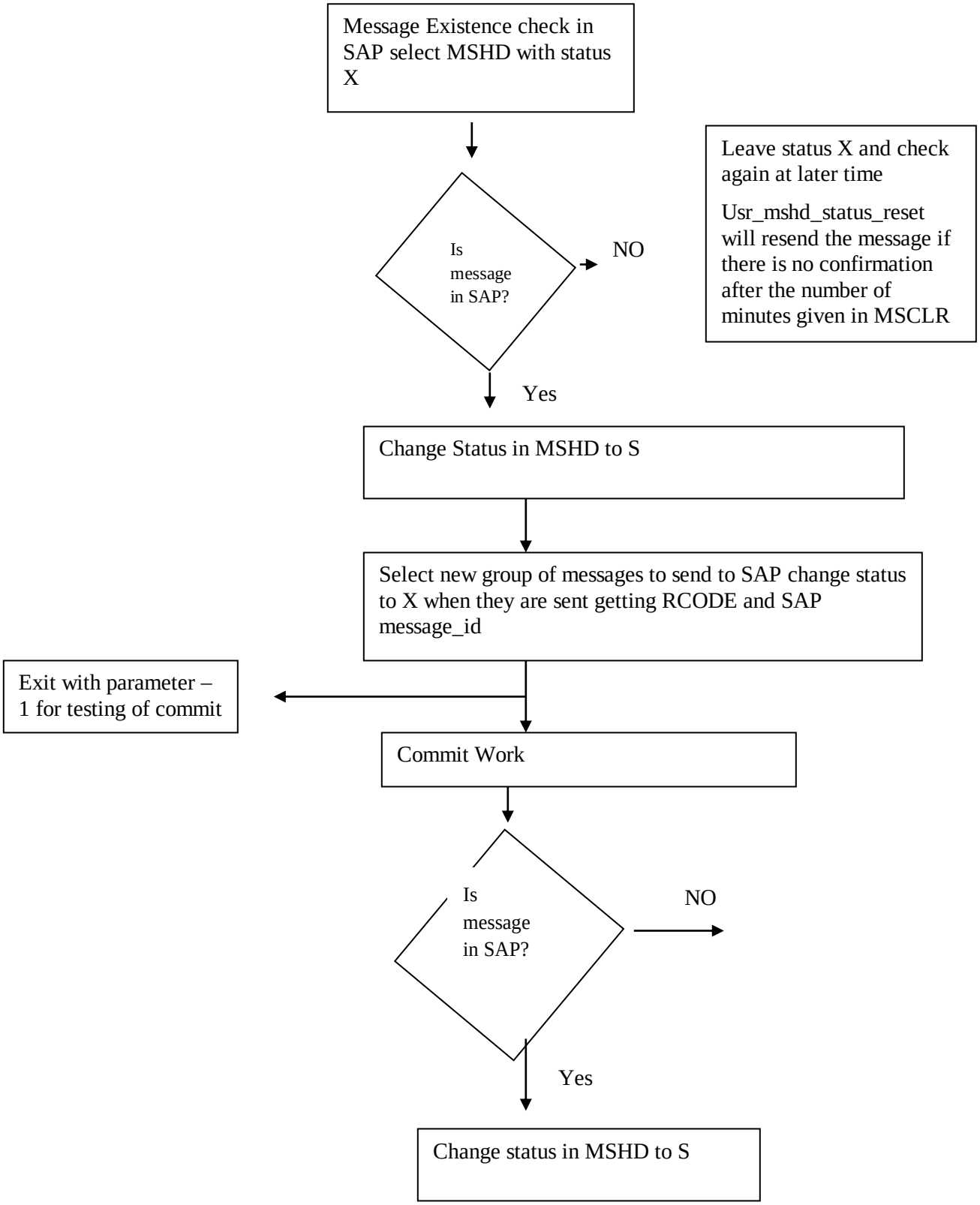
Message Characteristics

E	CB1	304	Error occurred in characteristics during message creation
E	CB	023	no characteristic found with internal number &
E	CB1	207	Characteristic & not created in the system
E	CB1	204	Check characteristic format
E	CB1	301	Error when converting value to format &1
E	CB1	205	Enter a long text for characteristic &
E	CB1	302	No value assigned to message characteristic &
E	CB1	203	Characteristic & has been assigned to messge more than once

Process Messsage Existence Check

E	CB1	001	An internal error has occurred
E	CB1	305	Message & is not created in the system
E	CB1	306	Message & already being processed
S	CB1	307	Message & is created

The following id the program flow for ROBCL –1 to create process messages in SAP.


Message Existence check in SAP select MSHD with status X
Is message in SAP?
NO
Leave status X and check again at later time
Usr_mshd_status_reset will resend the message if there is no confirmation after the number of minutes given in MSCLR
Yes
Change Status in MSHD to S
Select new group of messages to send to SAP change status to X when they are sent getting RCODE and SAP message_id
Exit with parameter – 1 for testing of commit
Commit Work
Is message in SAP?
NO
Yes
Change status in MSHD to S

This process includes the BAPI's to create the process message, commit the work and to check the existence of the process message in SAP. Messages that cannot be confirmed in SAP will remain at a status of X. It is up to the user to change this status back to blank so they will be resent after sufficient review has been made to determine if there is a problem in SAP accepting messages.

Recipe and Message Download

Transactional RFC Download

The program TCRD handles the download from SAP that is done with transactional RFC calls. The status of these transactional messages is recorded in the table SAP_transaction_master. The possible values for the type are

CRID - recipe download

TPMU - message return code from pmucl

CTOP - control recipe available if you are pulling recipes rather than pushing Recipes from SAP

MSID - text messages sent with the CO57 transaction in SAP

Depending on whether you configure SAP to push recipies down or whether they shoulc be pulled down will determine what is done by TCRD. The best method is to configure so SAP pushes the recipe or message down as soon as it is available rather than having to poll for its existence.

If you are have the control system put the recipe then TCRD will send down that a recipe is available and then you use the application TCRPS for pull single or TCRP for pull recipies. These applications will be scheduled to run in PSRLINK group_master and exec_batch tables.

- SAP_transaction_master

Table Field	***Meaning***
Key_id	Key to either recipe or MSG_MSHD or MSHD table
type	Type of transaction, Recipe, message, message rcode, control recipe available
tid	Transactional ID
Rec_datetime	Time of transaction

The download of ad-hoc messages is only handled by the transactional RFC call there is no corresponding BAPI for this. The majority of companies find the support of these essential to their implementation.

BAPI Control Recipe List

There is a table called pp_rc_getlist_in that will setup the requests for recipes. The user should configure the request for recipes that are desired. Multiple requests can be configured.

Table Field	*Meaning*
Id	Indentity
Plant	Plant
Destaddress	Destination Address RFC destination
Desttype	Type of control recipe destination must be set to 3 to be download by the BAPI
Processorder	Process Order Number
Material	Header material of the process order
Testcntlrec	Indicator to read test control recipes set to X for test otherwise it is blank
prodcntlrec	Indicator read productive control recipes default valus is X to read you do not need to set
Cntrecstatus	Control recipe status, 00001 is created can be downloaded
Datefrom	Earliest creation data of control recipe default value 00000000 no restriction
dateto	Latest creation data of control recipe default value is current local time
status	If you set to R then this will not be used in the recipe selection
Status_timestamp	Timestamp of status

The following are sample configurations.

id	plant	destaddress	desttype	processorder	material	testcntlrec	Prodcntlrec	cntlrecstatus	datefrom	dateto	status	status_timestamp
1	1100	OSI_2					X	00001				6/28/2001 11:53:12 AM
3	1100	OSI_2				X		00001				6/28/2001 11:53:12 AM

The program that is run to get the list of control recipes available is ROBCL –3. This program can be scheduled to run a regular intervals in group_master and exec_batch but the download of a list is included in the total download option given below. This is a component dll that runs in Microsoft Transaction server with the SAP DCOM connector. SAP DCOM and the component must be installed.

The output of the call is a list of available recipes. The list of recipes will be in the table pp_rc_cntlrecheader_out.

Table Field	*Meaning*
Id	Indentity
Request_id	Corresponds to the idenity in the pp_rc_getlist_in table
Cntl_rec_id	Control recipe
Plant	Plant
Proc_order	Process Order Number
dest	Control recipe destination
Dest_address	Address on the control recipe destination for RFC
Dest_typ	Type of destination 3 is required for download with BAPI
Cntl_rec_status	Control recipe status, 00001 is created can be downloaded
Test_flag	Indicator if control recipe is for test X
Recipe_text	Short text of the order used in the master recipe
Material	Header material of the process order
Material_text	Material description
Insplot	Inspection lot number of the order
Status	Will be set to C when the recipe itself has been retrieved
Status_timestamp	Status of the entry

Any error messages will be logged in the error_log table. The error messages are

E CB1 403 No control recipes found

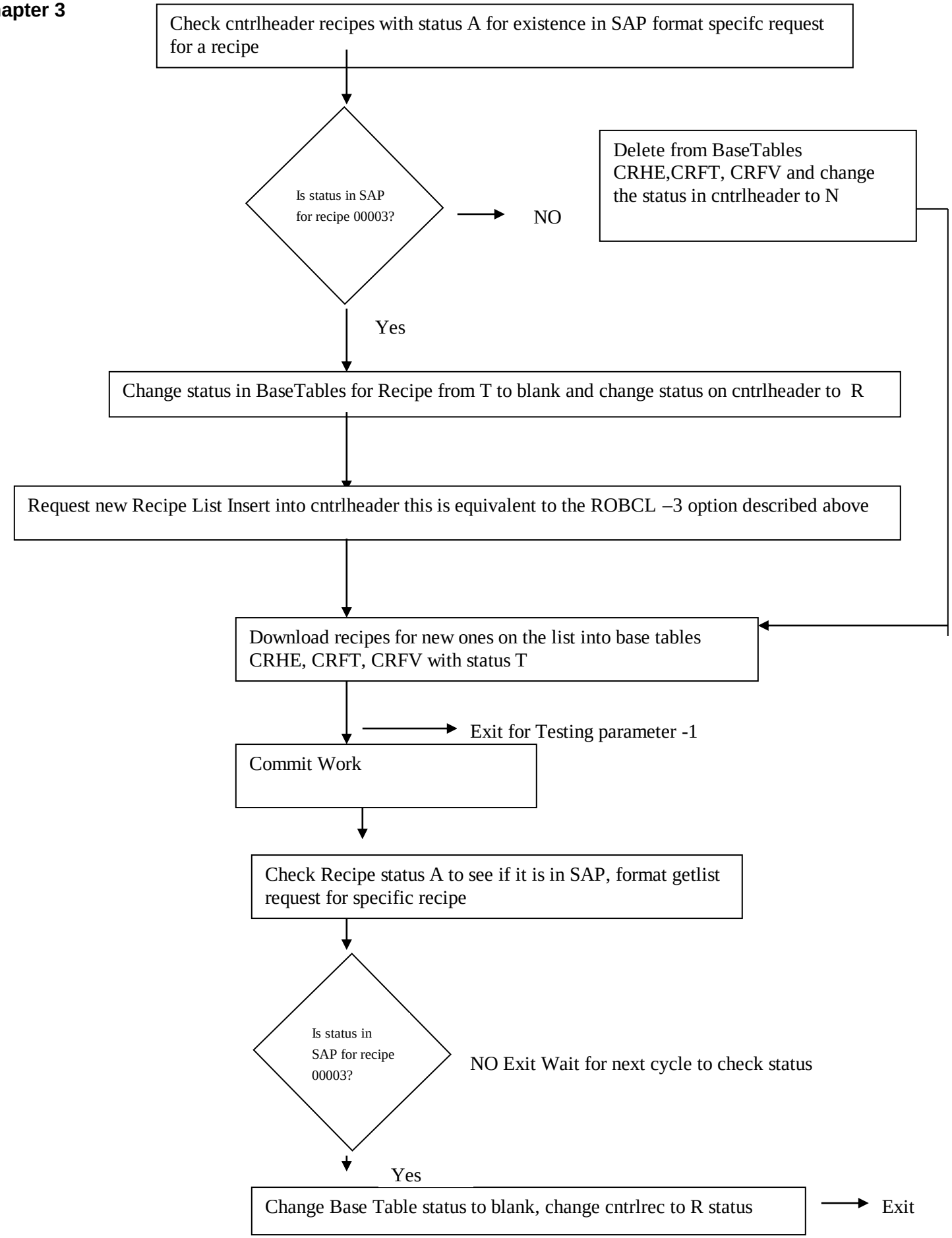
BAPI Control Recipe Download

This uses the RLBOCL application with the –4 option. It will take all recipes in the pp_rc_cntlrecheader_out table and request the details of the recipes. The output of the recipes is stored in the standard CRHE, CRFT , CRFV and TLINES tables used by the transaction RFC call.

There have been modifications to the standard table. In the CRFT table there are two new fields for the process instruction line number and the phase number.

All error messages will be recorded in the error_log table. If a locked error message is received it will try to retrieve the recipe the next time it executes.

When a control recipe has been received the status in the pp_rc_cntlrecheader_out table will be updated.



The System Messages will be recorded in the error_log table and they include

E	CB1	404	You ar not authorized to request control recipes for address &2
E	CB	082	Control recipe & has already been sent
E	CB1	401	Control recipe & not creaed in the system
E	CB	014	Control recipe & locked
A	CB	201	Further processing not possible (contact system administrator)
E	CB1	207	Characteristic & not created in the system
E	CB1	301	Error when converting value to format &1

Download of Characteristic Data

PROC_CHAR_HELPVALUES_GET RFC Function

This is a syncronous RFC call. Any text that comes down with ‘ or “ marks will have these taken out because they are reserved characters.

Input Tables

Download_char_helpvalues

Plant	Char(4)	X	Plant
PPPI_CHAR	Char(30)	X	Characteristic name such as PPPI_BATCH
Max_of_rows			Maximum number of values to be selected
Description	Char(1)		Ind: X or “ “ determines if only description is to be returned if X
ID			Identitity for the request
Requestor			Program or person making the request
Request_id			Number assigned for the request this might be the message_request
Status	Char(14)		S or blank
Timestamp			Timestamp of the status

Selection_for_helpvalues

Shlpname	Char 30	
Shlpfield	Char 30	Name for the field, valid values are from the table description_for_helpvalues in the columnfieldname
Sign	Char 1	Ind. Include or exclude a value or value range in selection I = include value, E= Exclude value
Option	Char 2	EQ = equal to LOW NE = not equal to LOW BT = between lower and upper value NB = ouside lower and upper value CP = contains the search pattern NP = does not contain the search pattern LT = less than LOW LE = less than or equal to LOW GT = greater than LOW GE = greater than or equal to LOW
Low	Char 45	Single value or low limit
high	Char 45	Upper limit
Shlpname	Char 30	
Shlpfield	Char 30	Name for the field, valid values are from the table description_for_helpvalues in the columnfieldname
Sign	Char 1	Ind. Include or exclude a value or value range in selection I = include value, E= Exclude value

For example the selection of the batches for a material would have the following:

Plant	100
PPPI_CHAR	PPPI_BATCH
Max_no_rows	
Description	

Shlpfield	MATNR
Sign	I
Option	EQ
Low	Y-300
high	

Returned values

Values_for_field

Values	Char 255	Only the value is returned here

Description_for_helpvalues

Tabname	Char 30	Table name
Fieldname	Char 30	Field name, can be used for selection_for_helvalues
Langu	Char 1	Language
Position		Position in the table
Offset		Offset exampl 7 = field starts with the 7th character in line
Leng		Field length
Fieldtext	Char 60	Short description
Reptext	Char 55	Heading
Scrtext_s	Char10	Short keyword
Scrtext_m	Char 20	Medium keyword
scrtext_ls	Char 40	Long keyword

Helpvalues

Helpvalues	Char 255	The format is transfeeed in the description for helpvalu how to parse the string

Detailed_value_information

Oper1	Char 10	Operator for lower limiting value ➢ > greater than ➢ < less than ➢ >= greter than or equal to ➢ <= less than or equal to	>
Oper2	Char 10	Operator for upper limiting value	<
String1	Char 30	Lower limiting value	5.5
String2	Char 30	Upper limiting value	12.0
String	Char 30	Allowed input value for characte withoud interval definition, singl value	5.5
Atstd	Char 1	Relevant value is to be displayed default	‘’

This can be used to determine if a batch already exists before the batch characteristics are sent up.

The program clchval.exe will execute the RFC call and is scheduled to run in group_master and exec_batch.

Microsoft Access must be used to configure requests for helpvalues which are to be done manually.

This program is used to support the instruction PI_BTCL, a query is automatically formulated for checking if the batch number exits.

BAPI Get HelpValues

Input Tables

Pp_pc_helpvalues_in

Plant	Char(4)	X	Plant
Char_name	Char(30)	X	Characteristic name such as PPPI_BATCH
maxrows			Maximum number of values to be selected
Descriptiononly	Char(1)		Ind: X or “ “ determines if only description is to be returned if X
ID			Identitity for the request
Status	Char(14)		S or blank
Timestamp			Timestamp of the status

Pp_pc_selection_for_helpvalues_in

id		Identity
Request_id		Request_id
FSelect_fld	Char 30	Name for the field, valid values are from the table description_for_helpvalues in the columnfieldname
fSign	Char 1	Ind. Include or exclude a value or value range in selection I = include value, E= Exclude value

fOption	Char 2	EQ = equal to LOW
		NE = not equal to LOW
		BT = between lower and upper value
		NB = ouside lower and upper value
		CP = contains the search pattern
		NP = does not contain the search pattern
		LT = less than LOW
		LE = less than or equal to LOW
		GT = greater than LOW
		GE = greater than or equal to LOW
fLow	Char 45	Single value or low limit
fHigh	Char 45	Upper limit
Status		Status
Status_Timestamp		Status_Timestamp

For example the selection of the batches for a material would have the following:

Plant	100
PPPI_CHAR	PPPI_BATCH
Max_no_rows	
Description	

Shlpfield	MATNR
Sign	I
Option	EQ
Low	Y-300
high	

Returned values

Pp_pc_valuesforfield_out

Vaid		Identity
Request_id		Request_id
Valuesforfield	Char 255	Only the value is returned here
Status		Status
Status_timestamp		Status Timestamp

Pp_pc_descriptionforhelpvalues_out

ID		Identity
Request_id		Request_id
Tabname	Char 30	Table name
Fieldname	Char 30	Field name, can be used for selection_for_helvalues
Langu	Char 1	Language
Position	Char 4	Position in the table
Offset	Char 6	Offset exampl 7 = field starts with the 7th character in line
Leng	Char 6	Field length
Fieldtext	Char 60	Short description
Reptext	Char 55	Heading
Scrtext_s	Char10	Short keyword
Scrtext_m	Char 20	Medium keyword
scrtext_ls	Char 40	Long keyword
Status		Status
Status_timestamp		Status Timestamp

Pp_pc_helpvalues_out

id		Identity
Request_id		Request_id
Helpvalues	Char 255	The format is transfeeed in the description for helpvalues on how to parse the string
Status		Status
Status_Timestamp		Status Timestamp

Pp_pc_fixvalues_out

id		➢ identity	
Request_id		➢ request_id	
Operator_low	Char 10	Operator for lower limiting value ➢ > greater than ➢ < less than ➢ >= greter than or equal to ➢ <= less than or equal to	>

Operator_high	Char 10	Operator for upper limiting value	<
Limit_low	Char 30	Lower limiting value	5.5
Limit_high	Char 30	Upper limiting value	12.0
Fix_value	Char 30	Allowed input value for characteristics wi interval definition, single value	5.5
Default_flag	Char 1	Relevant value is to be displayed as default	‘’

System messages will be written to the error_log table they include the following

E	CB	616	You ar not authorized to display allowed values
E	0C	010	Enter a valid characteristic
E	CB1	207	Characteristic & not created in the system
E	CB	614	No allowed values defined for characteristic &
A	CB1	210	Unable to determin allowed values for characteristic &
E	CB1	200	Plant & does not exist (Enter a valid plant)
A	CB	251	Further processing not possible (contact system administrator)

Get Characteristic Detail

PROC_CHAR_GET_LIST_WITH_DETAIL RFC

Input Tables

Download_characteristics

Plant	Char 4	X	Plant	
Language	Char 1		Language, default value is logon Language	
Instruction_chars	Char 1	X	Ind. Detail data on process instruction requested, Default value X select process instruction characteristics	X
Message_chars	Char 1	X	Ind. Detail data on process message characteristic requested, Default value X select message characteristics	X

Characteristic_group	Char 10		Characteristic group for which detail characteristic data is required if no value is specified the transfers data for all groups. Default value all characteristic groups	PPPI_01

Output Tables

Characteristic_list

Atname	Char 30	Characteristic name
Atfor	Char 4	Data type of the characteristic CHAR character NUM floating point number DATE YYYYMMDD TIME HHMMSS
Anzst	Char 5	Number of characters
Anzdz	Char 5	Number of decimal places
Atvor	Char 1	Ind. Negative values allowed
Atsch	Char 30	Input template
Atkle	Char 1	Ind. Case sensitive
Atdim	Char 5	Exponent in display 0 display without exponent 1 exponent is set automatically on place before the decimal point 2 display with exponent entered 3 display in scientific format 3 places before the decimal point
Atdex	Char 1	Exponent display format
Atkla	Char 10	Characteristic group
Txtrf	Char 1	Ind. Characteristic value is long text

Characteristic_text

Atname	Char 30	Characteristic name
Atfor	Char 4	Data type of the characteristic CHAR character NUM floating point number DATE YYYYMMDD TIME HHMMSS

Atnam	Char 30	Characteristic name
Spras	Char 1	Language
Atbez	Char 30	Characteristic description
Atue1	Char 30	First line of heading
Atue2	Char 30	Second line of heading

The program cldwchr.exe is used to execute this RFC call.

BAPI Get Characteristic Detail

The BAPI is executed in the program ROBCL –2. This uses the SAP DCOM Connector and Microsoft Transaction Server.

The input request in formulated in the table pp_pc_getlist_in

ID			Indentity column	
Plant	Char 4	X	Plant	
Language	Char 1		Language, default value is logon Language	
InstructionCharsFlag	Char 1	X	Ind. Detail data on process instruction requested, Default value X select process instruction characteristics	X
MessageCharsFlag	Char 1	X	Ind. Detail data on process message characteristic requested, Default value X select message characteristics	X
CharacteristicGgroup	Char 10		Characteristic group for which detail characteristic data is required if no value is specified the transfers data for all groups. Default value all characteristic groups	PPPI_01
Status			Status Value	
Status Timestamp			Timestamp of Status	

The output of the request is received in the tables pp_pc_char_data_out and pp_pc_char_text_out.

CharacteristicData pp_pc_char_data_out

ID		Identity
Request_id		Request_id
Name_char	Char 30	Characteristic name
Data_type	Char 4	Data type of the characteristic CHAR character NUM floating point number DATE YYYYMMDD TIME HHMMSS
Number_digits	Char 5	Number of characters
Number_decimals	Char 5	Number of decimal places
Sign	Char 1	Ind. Negative values allowed
Template	Char 30	Input template
Case_sensitive	Char 1	Ind. Case sensitive
Exponent	Char 5	Exponent in display 0 display without exponent 1 exponent is set automatically on place before the decimal point 2 display with exponent entered 3 display in scientific format 3 places before the decimal point
Exponent_type	Char 1	Exponent display format
Char_group	Char 10	Characteristic group
Longtext_flag	Char 1	Ind. Characteristic value is long text
Status		Status
Status_timestamp		Status Timestamp

Characteristic_text pp_pc_char_text_out

ID		Identity
Request_id		Reguest_id
Name_char	Char 30	Characteristic name
Desc_char	Char 30	Characteristic name
Langu	Char 1	Language
Langu_iso	Char 2	Language key according to ISO 639
Atbez	Char 30	Characteristic description
Hd_line1	Char 30	First line of heading

Hd_line2	Char 30	Second line of heading
Status		Status
Status_timestamp		Status Timestamp

The error messages are written to the error_log table and can include the following.

E	CB1	208	No messages found for the selection criteria you entered
E	CB1	209	Characteristics group& is not release for the desired use
E	CB	615	you are not authorized to display characteristic data

BAPI Material Get Detail and Material Get List

Get a list of materials in a plant and provide the detail information on material available in SAP to the plant floor. Material information can be requested based on material and plant. The information that is provided with this BAPI includes pricing and generic properties of the material.

The BAPI calls that are supported are BAPI_MATERIAL_GETLIST and BAPI_MATERIAL_GET_DETAIL. These will be added as a standard part to the PPPI interface.

Tables for BAPI_MATERIAL_GET_DETAIL

Lo_mat_getdetail_in

This sets up the input request, Material is required, if plant is given the plant data is returned, if Valuation area is given then the Valuation data is returned. Records that have a status of S or Null will be picked up for processing if the results already exist then they will be deleted and updated with the new set of results so only one set of results is maintained. If and error or a warning is returned the status will be updated and the error logged in the error_log table. The columns for the status and status_timestamp are for the customers use while the rlink_status and rlink_timestamp are for the use by the RLINK process.

id	material	plant	valuationarea	valuationtype	Rlink_status	rlink_timestamp	Status	Status_
1	T-HV200	1100			S	10/9/2002 5:36:49 PM		

Lo_mat_doc_out – Material Plant Data Only output if plant is specified

Plant specific information for the material

PUR_GROUP	Purchasing group

ISSUE_UNIT	Unit of issue

id	request_id	Pur_group	Issue_unit	Rlink_status	rlink_timestamp	Status	Status_timestam
1	1			N	10/9/2002 5:36:49 PM		

Lo_mat_dobew_out – Material Valuation Data Only out put if valuation area is specified

PRICE_CTRL	Price control indicator
MOVING_PR	Moving average price/periodic unit price
STD_PRICE	Standard price
PRICE_UNIT	Price unit
CURRENCY	Currency Key
CURRENCY_ISO	ISO code currency

id	request_id	Price_ctrl	Moving_pr	Std_price	Price_unit	Currency	Currency_iso	Rlink_status	rlink_timestamp	Status	Status_timestamp
1	1		0	0	0			N	10/9/2002 5:36:49 PM		

Lo_mat_doa_out – Material General Data

MATL_DESC	Material description
OLD_MAT_NO	Old material number
MATL_TYPE	Material type - Key that assigns the **material** to a group of materials such as raw materials, operating supplies or trading goods
IND_SECTOR	Industry sector- Key that specifies the branch of industry to which the **material** is assigned
DIVISION	Division - A way of grouping materials, products, or services. The system uses divisions to determine the **sales areas** and

	the **business areas** for a material, product, or service.
MATL_GROUP	Material group - Key that you use to group together several **materials** or **services** with the same attributes, and to assign them to a particular material group.
PROD_HIER	Product hierarchy
BASIC_MATL	Basic material (basic constituent of a material) - obsolete
STD_DESCR	Industry Standard Description (such as ANSI or ISO)
LAB_DESIGN	Laboratory/design office
PROD_MEMO	Production/inspection memo
PAGEFORMAT	Page Format of Production Memo
CONTAINER	Container requirements - Key for the regulation that governs which type of container the **material** must be stored and shipped in
STOR_CONDS	Storage conditions
TEMP_CONDS	Temperature conditions indicator
BASE_UOM	Base unit of measure
EAN_UPC	International Article Number (EAN/UPC)
EAN_CAT	Category of International Article Number (EAN)
SIZE_DIM	Size/dimensions
GROSS_WT	Gross weight
NET_WEIGHT	Net weight
UNIT_OF_WT	Weight Unit
VOLUME	Volume - Space that the material occupies per unit of volume. The volume refers to the unit specified in the "Volume unit" field.
VOLUMEUNIT	Volume unit
LENGTH	Length
WIDTH	Width
HEIGHT	Height
UNIT_DIM	Unit of dimension for length/width/height
MANU_MAT	Manufacturer part number
MFR_NO	Manufacturer number
BASE_UOM_ISO	Base unit of measure in ISO code
UNIT_OF_WT_ISO	Unit of weight in ISO code

VOLUMEUNIT_ISO	Volume unit in ISO code
UNIT_DIM_ISO	Unit for length/breadth/height in ISO code
CREATED_ON	Creation date
CREATED_BY	Name of Person who Created the Object
LAST_CHNGE	Date of last change
CHANGED_BY	Name of person who changed object
MATL_CAT	Material category
EMPTIESBOM	Empties Bill of Material
BASIC_MATL_NEW	Basic Material
.INCLU--AP	
LAST_CHNGE_DATE	Last changed date

id	request_id	Matl_desc	Old_mat_no	Matl_type	Ind_sector	Division	Matl_group
1	1	Ice Mix		HALB	M		

Prod_hier	Basic_matl	Std_descr	Lab_design	Prod_memo	Pageformat	Container

Stor_conds	Temp_conds	Base_uom	Ean_upc	Ean_cat	Size_dim	Gross_wt
		KG				0

Net_weight	Unit_of_wt	Volume	Volumeunit	Length	Width	Height
0		0		0	0	0

Unit_dim	Manu_mat	Mfr_no	Base_uom_iso	Unit_of_wt_iso	Volumeunit_iso	Unit_dim_iso	C
			KGM				1

Created_by	Last_chnge	Changed_by	Matl_cat	Emptiesbom	Basic_matl_new	Rlink_status
FISCHER	8/20/199	FISCHER				N

rlink_timestamp	Status	Status_timestamp
10/9/2002 5:36:49 PM		

Tables for BAPI_MATERIAL_GET_LIST

In several of the following tables there are input fields that ask for sign the convention is as follows:

Ind. Include or exclude a value or value range in
selection I = include value, E= Exclude value

In several of the following tables there are input fields that ask for option the convention is as follows:

EQ = equal to LOW
NE = not equal to LOW
BT = between lower and upper value
NB = ouside lower and upper value
CP = contains the search pattern
NP = does not contain the search pattern
LT = less than LOW
LE = less than or equal to LOW
GT = greater than LOW
GE = greater than or equal to LOW

The convention on status changes is that the RLINK program will use the fields marked RLINK_STATUS and RLINK_TIMESTAMP and leave the other status and timestamp fields for customer use.

Lo_mat_getlist_lst_out – materials returned

Field name	Description
Material	Material
Matl_desc	Material description
Material_external	Future development
Material_guid	Future development
Material_version	Future development

id	request_id	Material	Matl_desc	Material_external
1	1	000000000000000170	Rebate settlement: gloss paints	

Material_guid	Material_version	rlink_status	rlink_timestamp	status	status_timestamp

Material_guid	Material_version	rlink_status	rlink_timestamp	status	status_timestamp
		N	10/15/2002 12:35:15 PM		

Lo_mat_getlist_frpn_in4- manufacturer information

Field name	Description
MANU_MAT	Manufacturer part number
MFR_NO	Manufacturer number
INCLU__AP	No description given

Lo_mat_getlist_maxrows_in – sets rows returned

id	maxrows	status	status_timestamp
1	10	R	10/15/2002 12:35:15 PM

Lo_mat_getlist_radc_in8 – DistributionChannelSelection

Field name	Description
Sign	Sign
Option	Option
Distr_chan_low	Distribution Channel From
Distr_chan_high	Distribution Channel To

Lo_mat_getlist_ral_in6 – StorageLocationSelection

Field name	Description
Sign	Sign
Option	Option
Stloc_low	From Storage Location
Stloc_high	To Storage Location

Lo_mat_getlist_ram_in2 –MatnrSelection - material selection

Field name	Description
Sign	Sign
Option	Option

Matnr_low	Material number low
Matnr_high	Material number high

id	request_id	Sign	iOption	Matnr_low	Matnr_high	status	status_timestamp
1	1	I	CP	*		R	10/15/2002 12:35:15 PM

Lo_mat_getlist_raso_in7 – SalesOrganizationSelection

Field name	Description
Sign	Sign
Option	Option
SalesOrg_low	Sales Organization From
SalesOrg_high	Sales Organization To

Lo_mat_getlist_ras_in3 – MaterialShortDescSelection

Field name	Description
Sign	Sign
Option	Option
Descr_low	Material Description From
Descr_high	Material Description To

Lo_mat_getlist_raw_in5 – PlantSelection

Field name	Description
Sign	Sign
Option	Option
Plant_low	From plant
Plant_high	To plant

id	request_id	Sign	iOption	Plant_low	Plant_high	status	status_timestamp
1	1	I	EQ	1100		R	10/15/2002 12:35:15 PM

Group_master

group_no	group_desc	batch_no	last_exec_dti	frequency_m	frequency_h

56	Logistics-Material	1	10/9/2002	1	0

Exec_batch

If you have not installed on the D drive the path must be changed

program_name	batch_order	functionality	exe_	group_	batch
D:\rlink\pppi\fe\matce6.exe –3	1	Get List and material detail	E		

Stored Procedures

Five stored procedures are provided with this customization.

Usr_lo_mat_dobew_out_I

Usr_lo_mat_doa_out_I

Usr_lo_mat_doc_out_I

Usr_lo_mat_general

Usr_lo_mat_stauts_u

Usr_lo_mat_getlist_lst_out_i

Applications

There is one new application matce6.exe. There are tree parameter options for the program.

- 1 This will execute the Get Detail program only based upon data configured in the input tables

- 2 This will execute the Get List program only based upon the data configured in the input tables

- 3 This will first execute the Get List program then use its output to add to the input for Get Detail and then execute the Get Detail program.

Components

There is a component that is added to the transaction server. The setup for the server is executed on the server machine and the setup for the client is executed on machines that are clients only.

Error Messages for Get Detail

- S <blank> Material could be read successfully
- E M3262 Material could not be read (no material number was transferred)
- E M3305 The material does not exist
- E M3853 No authorization to display the material
- E MM302 No authorization to display material master data at client level
- E M3849 No authorization for the authorization group of the material

- E M3852 No authorization for this material group
- E M3851 No authorization for this material type
- E MM357 The currency code for the valuation area is missing

Error Messages for Get List

- W MM 354 More entries exist
- E HV 026 Invalid SIGN entry
- E HV 027 Invalid OPTION entry
- E MM 352 No internal material number could be found for the manufacturer part number
- E MM 353 No entries could be determined for the selection criteria specified

Miscellaneous Notes

None of these values are currently written to PI tags this is a possible future enhancement.

Adding filter to GET_MATERIAL_ DETAIL BAPI:

Add data filter condition to a query, which sets the input for get material detail BAPI filtering the entries return from material get list.

Table data_filter

```
create table data_filter
(
        filter_id          char(16)        not null primary key,
        condition      nvarchar(512)   not null
)
```

Data entry example:

filter_id	condition
MATERIAL	and material like "%G%"

The stored procedure “usr_lo_mat_general” is modified to check data_filter table for the key “MATERIAL”. If exists, the condition is added with standard SQL Query and executed, otherwise just the standard SQL Query is executed.

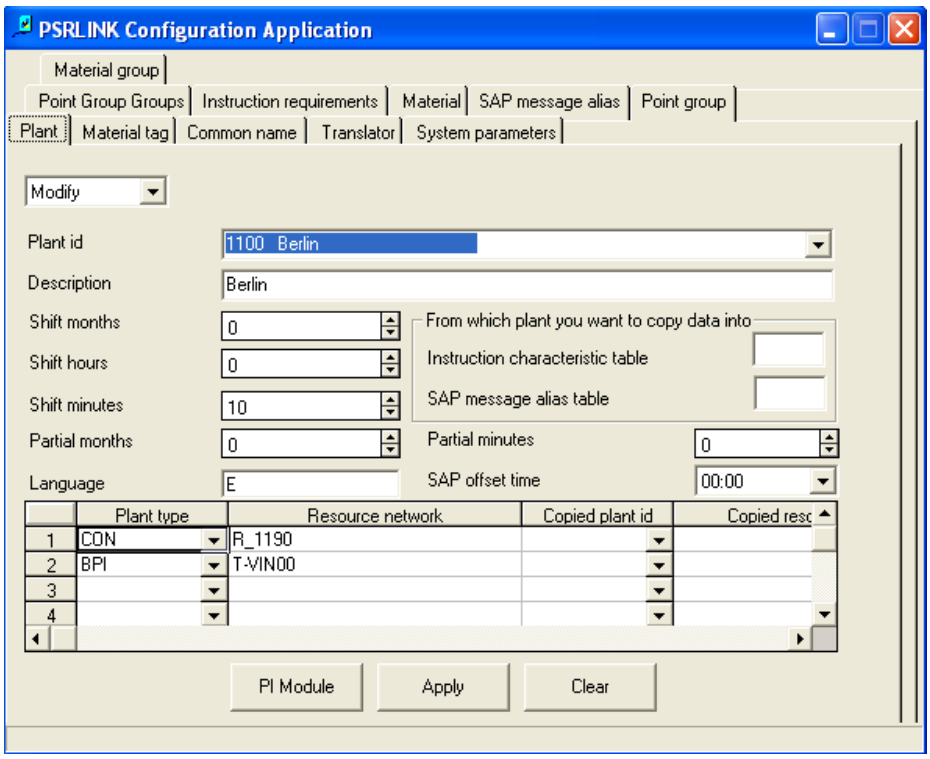
Chapter 4
Configuration

Overview

The Configuration Application is used to set up the tables in the plant_suite database. The program can be started from the menu. The executable is c:\PSRLINK\CLIENT\FE\configure.exe. The user selects the plant that is to be configured on subsequent dialogs on the Configuration Tab.

Plant Information

General Plant information is entered using the Plant Tab. This tab also sets the plant that will be referenced on the other tabs of the dialog. If you are setting up a new plant you can start by copying the samples we have configured for continuous plant 1100, PI-Batch 1200 or Batch Execution System 1300 using the copy from portion of this dialog. The plant resource network should correspond to what you give in the AORD instruction.



The example shown here is for making a new continuous plant. Select the Add option, enter the plant to be copied being plant 1100 and for the resource network specify the resource network information to be copied.

If you want to add a new resource network after the plant exists then select Modify add the new line for the resource network at the bottom and check the line that is to be added. Select Apply.

If you want to delete a resource select that line, select the delete key and then select Apply.

If you want to delete the entire plant then select the delete option and Apply.

The shift times applies to a continuous plant. Shift months will add the number of months to the date. It will increment the month number by the number of months given.

- Plant table.

Table Field	***Meaning***
Plant_id	Name of the plant as it will come down from SAP/R3
Plant_description	Description of the plant

Plant_resource_network- The plant resource concept allows one part of the plant to be continuous and another part operated by a batch execution system.

Table Field	***Meaning***
Plant_id	Name of the plant as it will come down from SAP/R3
Resource_network	This is the resource network which is used in the AORD_1 instruction
Type	Designates whether the plant is a continuous (CON), or Batch Exec (BES) or PI-BATCH(BPI) plant
Crst_disable	Set to X if you do not want the PI_CRST messages to be sent to SAP
Partial_dur_min	No of minutes between partial results in continuous plant for the resource network
Partial_dur_hr	Not operational
Partial_dur_months	Not operational
Shift_dur_min	No of minutes duration for the recipe for the resource network
Shift_dur_hr	No of hours duration for the recipe for the resource network
Shift_dur_mon	No of months duration for the recipe for the resource network

If you are running a plant that is continuous but you have different shift times for each resource network then you can set these through the access tables in the plant_resource_network table.

- Location Table

Table Field	*Meaning*
Location_code	Plant_id
Location_description	Plant description
Shift_duration_hr	Shift duration in hours which will be added to the OSI_START_TIME and OSI_STRAT_DATE
Shift_duration_min	Shift duration in minutes which will be added to the OSI_START_TIME and OSI_STRAT_DATE
Partial_min	No of minutes to be incremented between requests for data in the example of a continuous plant which will ask for updates during the execution of the order on this basis of time increments.
Language	E for English must be in agreement with Alias System code for language
flag	Set to 'Y' if you want the shift drurations calculated at the resource network level for the plant.

- Subscriber Table

Table Field	*Meaning*
Id	Unique id
Name	Name of Subscriber ex. PI
Address	Address of machine or DDE server name
Resource_network	Overall resource network, for example if there is one openbatch server per resource network
Plant_id	Name of the plant this appears in

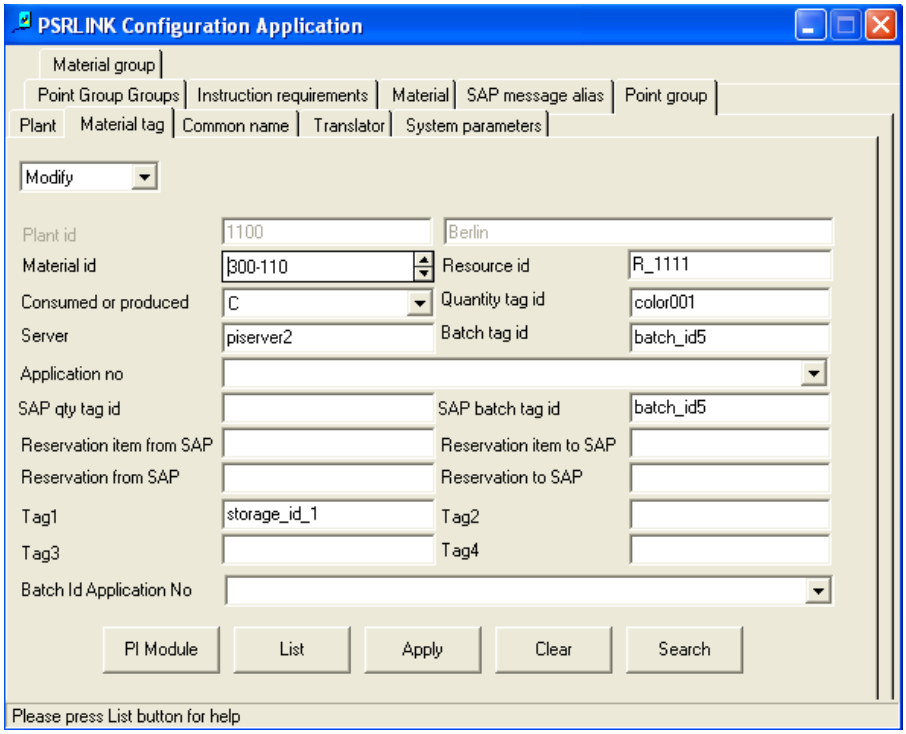
Other tables modified are the Translator and SAP_message_alias tables.

The PI Module button will create a module in the PI-ModuleDatabase for the plant. This is optinal and not required for RLINK execution.

Material Tags

On the Material Tag Tab the user has the option to Add, Modify, Delete an entry for a material in the database. Adding a material is done by selecting add, specifying the material name (you must include the leading SAP/R3 0's) give the resource where the material will be used in the recipe and specifying whether it is 'C' for consumed or 'P' for produced. The user can use the Search button to pull up the Tag Search Dialog. If the standard application for converting the tag information to the value required by SAP/R3 is not to be used another application can be chosen. Finally select 'Apply' to store the information in the database. If you want to send the material quantity from SAP down to a tag in PI then fill out the SAP Quantity tag. If you want to send a batch number from SAP down to a tag in PI then you must include in the AMAT_1 instruction the characteristic PPPI_BATCH with the batch_id and fill in the SAP batch_id tag on this dialog. If there are multiple AMAT's for the same material than the batch will be

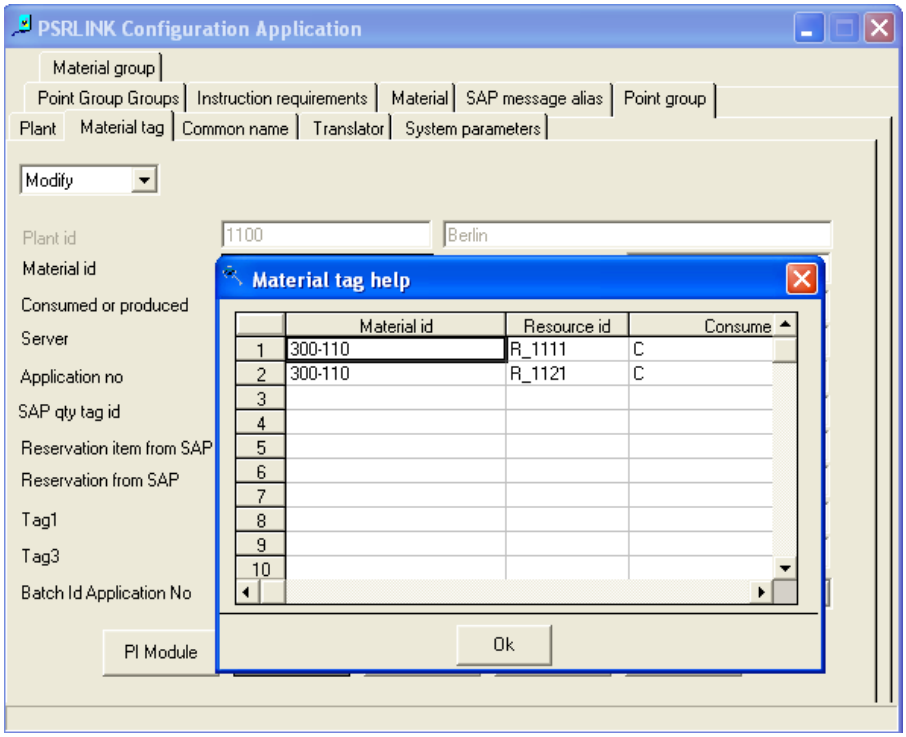
entered into this point at the same timestamp. If you want to store the reservation or reservation_item for material in a PI tag then you must send it down in the AMAT_1 instruction and have a tag in the entry “reservation from SAP” or “reservation item from SAP”. If there are multiple AMAT’s for the same material in the same phase they will be entered in this tag at the same timestamp. You can return the reservation or reservation_item from a tag to SAP by entering a tag in the “Reservation to SAP” or “Reservation_item to SAP”. For getting the values from PI the application that will be used is the standard one given in the translation methods for the batch, resrvation and

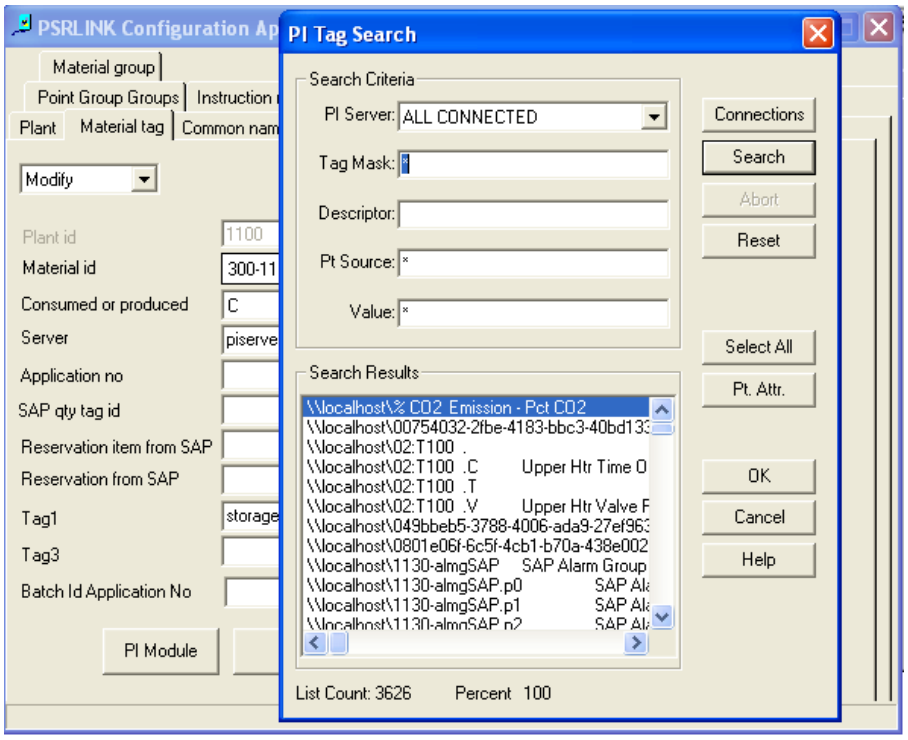


reservation_item. You cannot override this application. The application override given here is only for material quantity.

There entries for 4 additional tag requests from PI that can be used to support additional properties with customization. The labels on the dialog can also be changed with customizing the resource file used for the dialog. The delivery complete program has been set up to use miscellaneous tag 3.

When the user chooses to Modify or Delete a material entry in the data base the List Button can be used to review the current entries in the database and select the correct entry. The user must select Apply for the Delete or Modify changes to take place.





- Material_Tag table.

This table is used to locate the tag for Material Consumed and Produced in a given Phase as well as the Batch_tag if requested.

Table Field	*Meaning*
Material_id	Material number as it will come from SAP/R3 check if you SAP/R3 system sends leading zero's
Resouce_id	Resource were the material will be used as sent down in the Phase instruction APHASE_1
Quantity_tag_id	PI tag for the material quantity
Server	PI Server for the tag
Plant_id	Plant as sent down in the AORD_1 instruction
Consumed_Produced	C or P standing for material consumed or produced it the location
Batch_tag_id	Tag which will hold the batch for that location and material
Application_no	No of application if the default for the instruction is to be overwritten
sap_qty_tag_id	Tag if the material quantity from SAP is to be written to PI
sap_batch_tag_id	Tag if the batch no is to be written to Pi
Reservation_item_from_sap	Tag which will hold the reservation items send down from SAP in the AMAT instructions
Reservation_item_to_sap	Tag for the reservation_item to be returned to SAP in the PI_CONS instruction
Misc_tag1	Misc tag to be used in customizing or with translation method usr_misc1_tag
Misc_tag2	Misc tag to be used in customizing or with translation method usr_misc2_tag
Misc_tag3	Misc tag to be used in customizing or with translation method usr_misc3_tag
Misc_tag4	Misc tag to be used in customizing or with translation method usr_misc4_tag
Misc_tag5	Misc tag to be used in customizing with translation method usr_misc5_tag. This entry does not appear on the configure application but is in the tables.
Reservation_from_sap	Tag which will hold the reservation send down from SAP in the AMAT instructions
Reservation_to_sap	Tag for the reservation to be returned to SAP in the PI_CONS instruction
Batch_id_app	Application number for batch

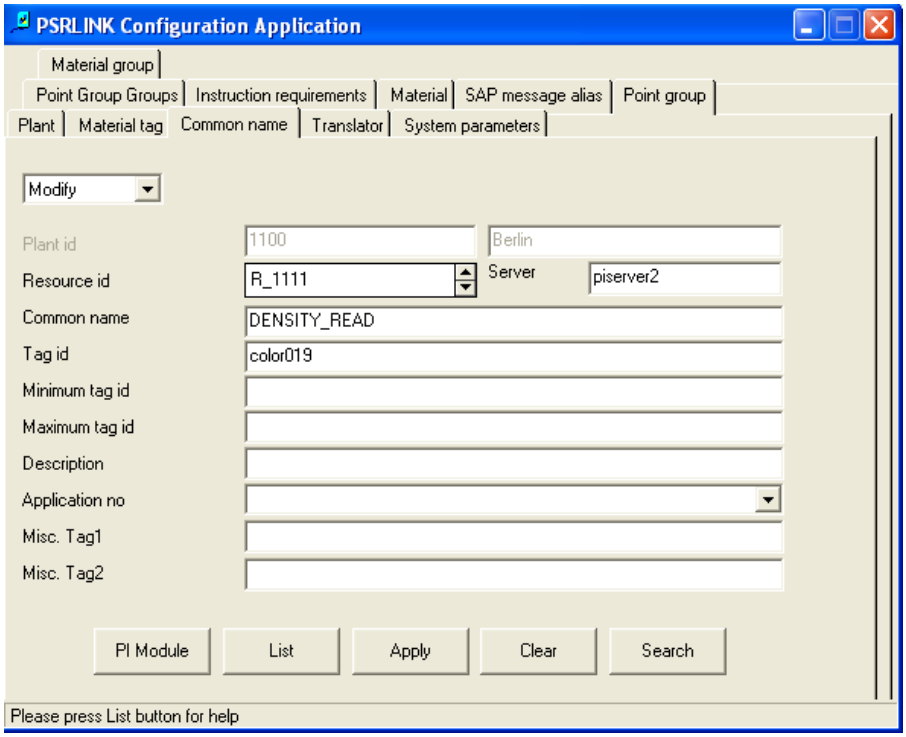
The PI Module button will create a module in the PI Module Database for the material. This is optional and is not required for RLINK execution.

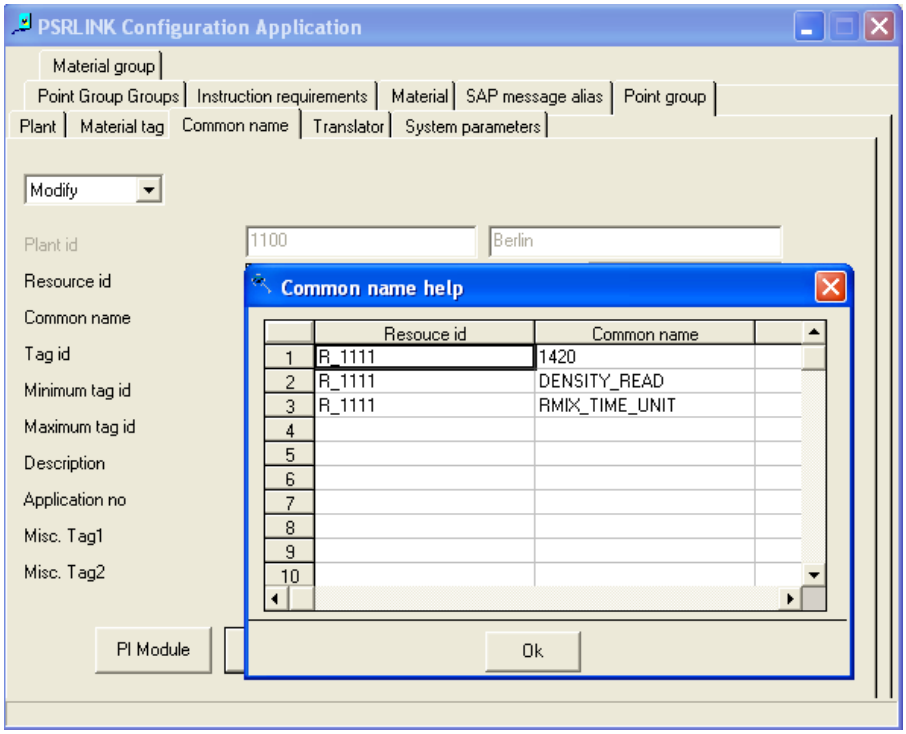
Common Name Tags

The Common Name Tags are used to setup mappings for SAP/R3 instructions such as AREAD1 ,AREAD2, APHACT, ASRACT and APHPAR_1. Just as in the Material selection the user can choose to Add, Modify or Delete an entry. The Search Button brings up the Tag Search Dialog. The user must have the mouse in a field that can receive tag information for the Tag Search Dialog to work. The user must specify the Resource where the reading is located as specified in the SAP/R3 recipe. If you are using this for the ASRACT instruction the resource should be configured to be the secondary resource.

Min. Tag and Max Tag are only used for the APHPAR_1 instruction if the minimum and maximum values are to be set in PI. The standard default application used for the instruction type can be overridden with an entry in application.

If Modify or Delete are chosen you can find the records that already exist in the database by selecting the List Button. For changes to take place you must select Apply.





Common_name Table

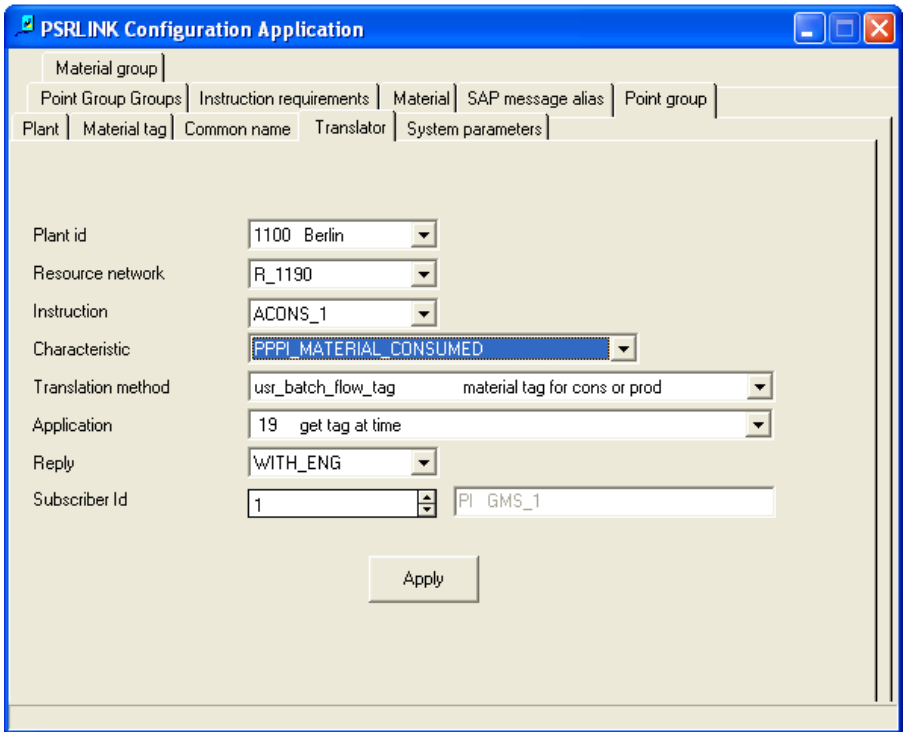
This table is used to store the translations used for characteristics such as DATA_POINT_NAME in AREAD instructions, PARAMETER_NAME for APHAPR instructions and PPPI_STD_VALUE_PARAMETER_ID for the APHACT instructions.

Table Field	***Meaning***
Plant_id	Plant as sent down in AORD_1
Resource_id	Resource as sent in APHASE_1
Common_name	This will be the value sent in DATA_POINT_NAME, PARAMETER_NAME, or PPPI_STDVALUE_PARAMETER_ID
Tag_id	PI Tag name for value
Server	PI server for the tag
Min_tag_id	PI Tag name where the min value will be sent if for APHAPR instructions
Max_tag_id	PI Tag name where the max value will be sent if for APHAPR instructions
Description	Added description to be used to enhance meaning of entries since for certain name sent from SAP/R3 the meaning is unclear. This is especially true for the Activity confirmation instruction.
Application_no	No of application if the default for the instruction is to be overwritten

Table Field	*Meaning*
Misc_tag1	Misc tag to be used in customizing, also used for the PHACT and SRACT instructions PPPI_STATUS_CONFIRMED
Misc_tag2	Misc tag to be used in customizing, also used for the PHACT and SRACT instructions PPPI_CONFIRMATION_SHORT_TEXT

The PI Module button will create a module in the PI Module Database for the material. This is optional and is not required for RLINK execution.

Translator



The translator table can be modified using the Translator tab. This is the default mechanism for retrieving data for a given instruction and characteristic. Execute the Apply button to make the changes.

Application_no	Application which will be used to retrieve value
Translate_method	Method used to translate SAP/R3 request to data which can be retrieved. See list of translation methods in Recipe Setup
Reply_method	Values are WITH, WITHOUT, WITH_ENG, WITHOUT_ENG, UNIT, mean set time and date with this value, do not set time and date with this value, set time and date and engineering unit with this value, do not set time and date with this value but set engineering unit, unit set with activity value.

Translator

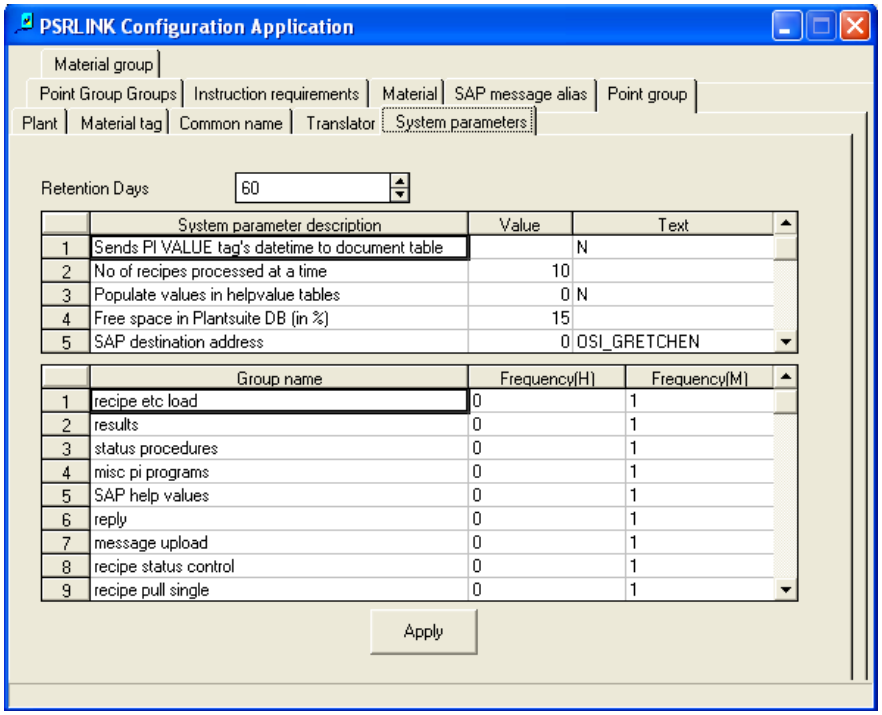
Table Field	*Meaning*
Request_part_name	SAP/R3 Characteristic name
Request_category	SAP/R3 Instruction Name
Subscriber_id	Machine were application will run
Application_no	Application which will be used to retrieve value
Translate_method	Method used to translate SAP/R3 request to data that can be retrieved. See list of translation methods in Recipe Setup
Reply_method	Values are WITH, WITHOUT, WITH_ENG, WITHOUT_ENG, UNIT, mean set time and date with this value, do not set time and date with this value, set time and date and engineering unit with this value, do not set time and date with this value but set engineering unit, unit set with activity value.
Reply_application	Not used
Reply_required	Not used
Required	Not used
Plant_id	Plant as sent in Recipe AORD

Translation_method

Table Field	*Meaning*
Name	Translation method name
Description	Translation method description

System Parameter

The System Parameter and Purge tables can be modified using the System Parameter tab.



The retention days are for the purge of the database.

The No. of Recipes Processed is the number of recipes that will be down loaded at a time from SAP/R3. Setting this value is to prevent the situation that an extremely large number of recipes have been created in SAP/R3 and are downloaded at one time stopping other functions on the server from progressing. This is the number that will be downloaded as a maximum for each cycle of PSRLINK.

Sleep time in minutes-sleep time between cycles of executing all the applications.

SAP/R3 Polling Time- Interval is the time that the SAPPOLL service will sleep before it tries to log onto SAP/R3 again.

Database free space- In order to avoid loss of data, a precautionary measure is taken by TCRD exe, before downloading the data from SAP/R3. When RFC call gets triggered in TCRD, it checks the available space in plant_suite database. If the space is less than the tolerance value (default is 15% of the database size) defined by the user in System_parameter table where system_parameter_code = "DBSIZ", it will write an Error message into error_log table and quit TCRD program.

The No. of Messages uploaded is to set a maximum on the number of messages sent to SAP/R3 each time that an upload occurs. This is done to prevent SAP/R3 from getting overloaded which we have seen occur.

The frequency for the execution of each program is given. The only one that you might choose to change is the message upload that has been set to 30 minutes on the install. This is for uploading messages to SAP/R3

The sleep duration is set in the System_parameter_table.(This table needs to be modified only if there is a change in the SLEEP time or ATATM which stands for the number of recipes which will be processed at a time)

ATATM – no of recipes to be processed at a time. For example if you download 100 recipes type system will only process 10 on each cycle through the application of usr_read_and_process. This prevents the system getting halted at just processing incoming recipes if there is a massive download of recipes from SAP/R3 at one time.

SLEEP- sleep time between cycles of executing all the applications.

IDLE is the time that the SAPPOLL service will sleep before it tries to log onto SAP/R3 again.

DBSIZ In order to avoid loss of data, a precautionary measure is taken by TCRD exe, before downloading the data from SAP/R3. When RFC call gets triggered in TCRD, it checks the available space in plant_suite database. If the space is less than the tolerance value (default is 15% of the database size) defined by the user in System_parameter table where system_parameter_code = "DBSIZ", it will write an Error message into error_log table and quit TCRD program.

PMUWT – is the time in minutes that the system will wait before setting an alarm that a message has been sent to SAP but not RCODE is returned.

ZROCN and ZROPR are used to determine if material of zero quantity is allowed to be sent to SAP. If the system_value_text is set to 'Y' then a value of 0 will be sent. You might have to install patches on SAP to support this option. The SAP patch numbers are 0108952 goods movement with PI_CONS and quantity 0 and 01732742 PI_CONS with quantity 0.

ZROPL – Log message in error_log table when produced material quantity is zero set 'Y'

ZROCL – Log message in error_log table when consumed material quantity is zero set 'Y'

DEST – SAP destination address used by the general SAP transactions

DTTM – This parameter was added to correct the problem that SAP can generate a time of 24:00:00 which is not know by Microsoft. You have the option of re-setting such a time to 23:59:59 for the SAMEDAY or incrementing the day for NEXTDAY. Enter either SAMEDAY or NEXTDAY.

INDCNT – is used to set the number of messages sent to SAP at one time.

PISRV – check the PI server status. This means that if this is set to Y then when an application is run it will check the pi_server2 table for the column that will hold the name of the server for that appliation and then it will check the servers table to see if that pi-server is up. If it is not up it will not execute the request. This is useful when the PI servers are taken for backup. If the value isset to N this check is not made. The setting of the status of the server in the server table is left to the user to implement. The is a dialog application where this can be set or the user can add a program to their PI backup procedure.

BTCLF- If this value is set to "N" the we will not formulate the helpvalue squery to ask if SAP has the batch_id. This is the default setting. If you want to formulate the helpvalues query before sending the value to SAP to verify that the batch_id exists in SAP then set the value to "Y".

MSCLR – The number of minutes that the system will wait and check to see if the message has been posted in SAP. If it does not get a response that the message has been posted in this time the procedure usr_mshd_status_reset will resend the message. The default value is 10. The program usr_mshd_status_reset is scheduled in exec_batch.

STACT- if TRUE usr_msg_hdr22 will not allow a phase status of 0002 to be sent to SAP until al PI-PHACT instructions have been sent.

MSG24 – There are some messages which are dependent on other messages and when reporting to SAP they must be sequenced. This feature is implemented for messages handled by usr_msg_hdr_24 procedure. To incorporate this feature

1. Initial parameter called “MSG24” with string Boolean value (“TRUE” and “FALSE”) is added to system_parameter table.
2. A new table called pp_msg_sequence with the following fields is added
3. Modified usr_msg_hdr_24 procedure to check the system_parameter and pp_msg_sequence.

COLUMN NAME	DATATYPE	COLUMN DESCRIPTION
GROUP_NUM (Primary Key)	INT	Point group number
PLANT_ID	CHAR(4)	Plant id
CATEGORY	CHAR(30)	The main message category.
CHAR_NAME	CHAR(30)	Name of the characteristic to look for when taking the timestamp
FOLLOWED_BY_CATEGORY	CHAR(30)	Dependent category

By default, the “MSG24” is set to “FALSE”. If we want to enable the message category sequencing feature, change the entry in system_parameter for “MSG24” to “TRUE” and make entries in pp_msg_sequence table.

For example

GROUP_NUM: 50

PLANT_ID: 1100

CATEGORY: PI_PROD

CHAR_NAME: PPPI_BATCH (Characteristic Name whose value is matched for in CATEGORY before translating FOLLOWED_BY_CATEGORY msg.)

FOLLOWED_BY_CATEGORY: YXFER

- System_parameter Table

Table Field	*Meaning*
System_parameter_code	Code for system parameter. The current use is for SLEEP
System_parameter_description	This entry give the sleep time in minutes between cycles of program execution
System_value	Value if numeric value entered
System_value_text	Value if text value for system parameter

- Purge Table

Table Field	*Meaning*
Table	Table to be purged

Table Field	*Meaning*
Purge_method	Method to be used to purge the table
Last_ Purge	Last date and time of purge
Retention_days	No. of days of data which should be retained

The following configuration shows the purge set up for the instructions which are general SAP Transactions using the purge method usr_arv_purge. The normal recipe purge is done by the purge method usr_crhe_mtd. This purge method will delete any reference to the recipe from the psrlink product. There is also a more soft purge called usr_crhe_mtd2 which will keep the recipe as it was downloaded put will not keep the data in other tables. When this purge method is used and entry will be made in the crhe table for the field deleted. This will allow the user to re-run this recipe so the results can be seen again in PSRGUI. The re-run recipe will not send messages up to SAP and will purged again the next time the purge program runs. In order to re-run the recipe you must change the status of the recipe in CRHE to blank. Currently this must be done using Microsoft Access.

The purge for recipies is based on the completion time of the recipe with status of 0004 (Terminated), 00005 (Processed) or 00007 (Discarded before started).

dbo_purge : Table

table_name	purge_method	last_timestamp	retention_days
arv_sap_tran	usr_arv_purge	12/6/2001 10:10:01 AM	60
crhe	usr_crhe_mtd	12/6/2001 10:10:01 AM	60
mm_gm_arv	usr_mm_purge	12/6/2001 10:10:01 AM	60
pp_arv_rem	usr_rem_purge	12/6/2001 10:10:01 AM	60

Record: 3 of 4

Point Group

The Point Group and Point Group Members tables can be modified using the Point Group tab. The entries for the point group types required for processing batch recipes are PI_BATCH, RECIPE, and PI_BATCH_O which are described in the chapter on PI and PI-Batch Specifics. Other reserved group types that are being used are QM for the interface to QM and EQUIP for the interface to the PM module.

The ability to manipulate points in groups with alias is a very powerful feature that the user could use for other applications.

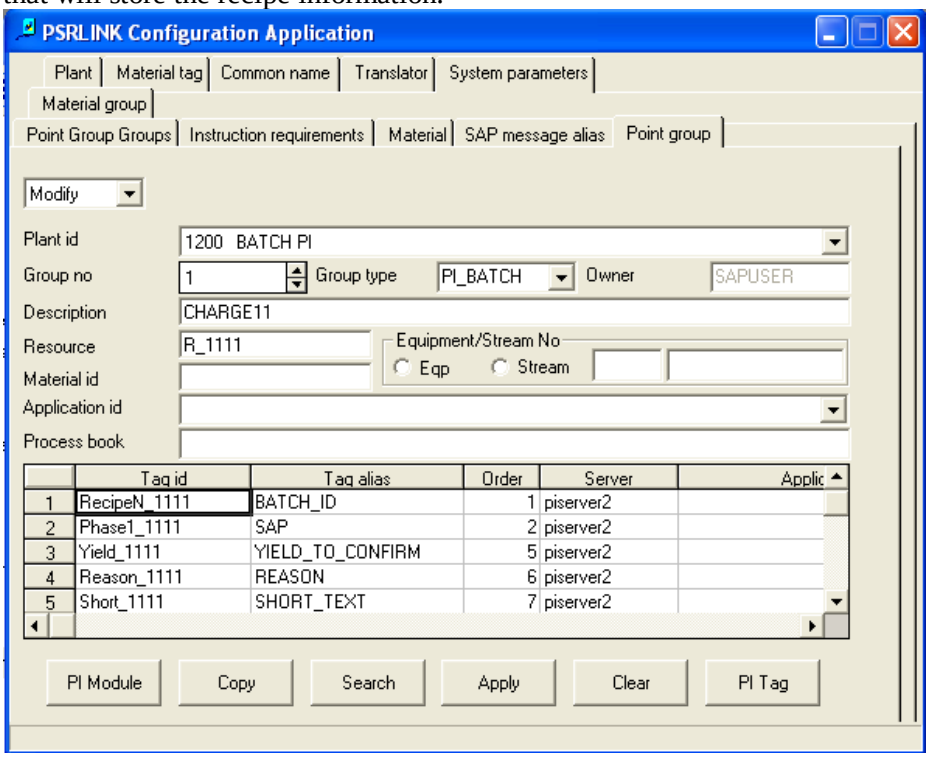
You must have Point Groups if one of the following is true

- The plant is of type BES or BPI
- The recipe requires PI_OPUST or PI_PHUST instructions
- The recipe requires PPPI_YIELD_TO_CONFIRM, PPPI_REASON_FOR_VARIANCE OR PPPI_CONFIRMATION_SHORT_TEXT
- The recipe has QM instruction

- The recipe requires PI_PHCON, PI_SRCON, PI_SRST, PI_BTCL, PI_BTCR instructions

Plants that are of the type BES or BPI require that the Point_group and Point_group_members table must be configured with the point names that will contain the information for the status of the recipe. The digital state tables must be setup as illustrated in the PIConfig script given in the chapter on PI Specifics.

If you want the data for the stop and start of the recipe and the phases from Batch Execution systems such as Openbatch and iBatch to be sent to PI for archiving and also reviewed in PI-Batch the tables for Point_group and Point_group_members must be setup as shown below. There must be a point group for each unit and one point group that will store the recipe information.



The name of the Group Description for the phases must match the PPPI_EXTERNAL_PHASE in the SAP/R3 recipe for each phase. The value for the phase resource must match the resource assigned in the SAP recipe for the phase. The group types that are reserved for this application are “RECIPE” and “PI_BATCH” which stand for a recipe group and a PI_Batch unit respectively. The reserved words in the point_group_members table are “SAP” for the SAP/R3 status tag, “ACTIVE” (optional) for the PI-Batch active tag, “PRODUCT_ID” (optional) for the product id, “BATCH_ID” for the batch tag in PI-Batch this will hold the recipe number, “RECIPE_ID” for the tag that holds the recipe tag, “STATUS” (optional) an active tag for the recipe, “PO” (optional) for the process order.

The digital states for the SAP recipe status are:

00001 Started

00004 Terminated

00005 Processed

00007 Discarded (never started)

The digital states for the SAP phase status are:

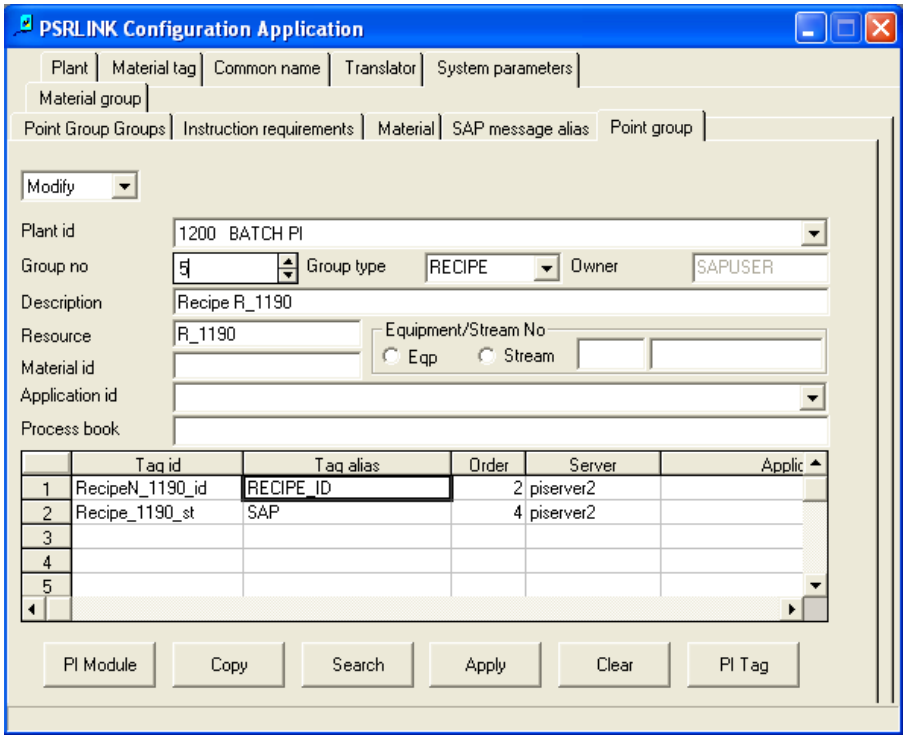
00001 Started

00003 Interuption

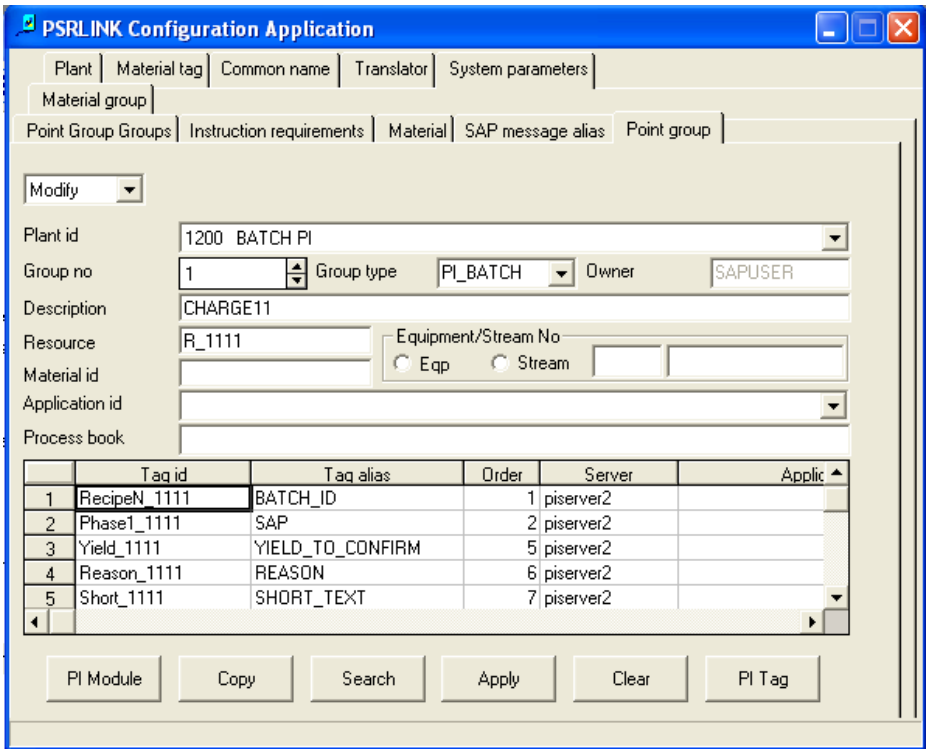
00002 Finished

- Interruption (After an interruption you must give a status of 00001 to start the phase again, if there are parameter values to be written on a phase start they are written each time the phase is started)
- Partial Finish

The following show how to set up a point group for each of the three cases RECIPE, PHASE that is done with the group type PI_BATCH and OPERATION that is done with the group type PI_BATCH_O. The description for the PI_BATCH must correspond to the PPPI_EXTERNAL_PHASE name that is given in the recipe. The description for the PI_BATCH_O must correspond to the Operation number in the recipe.

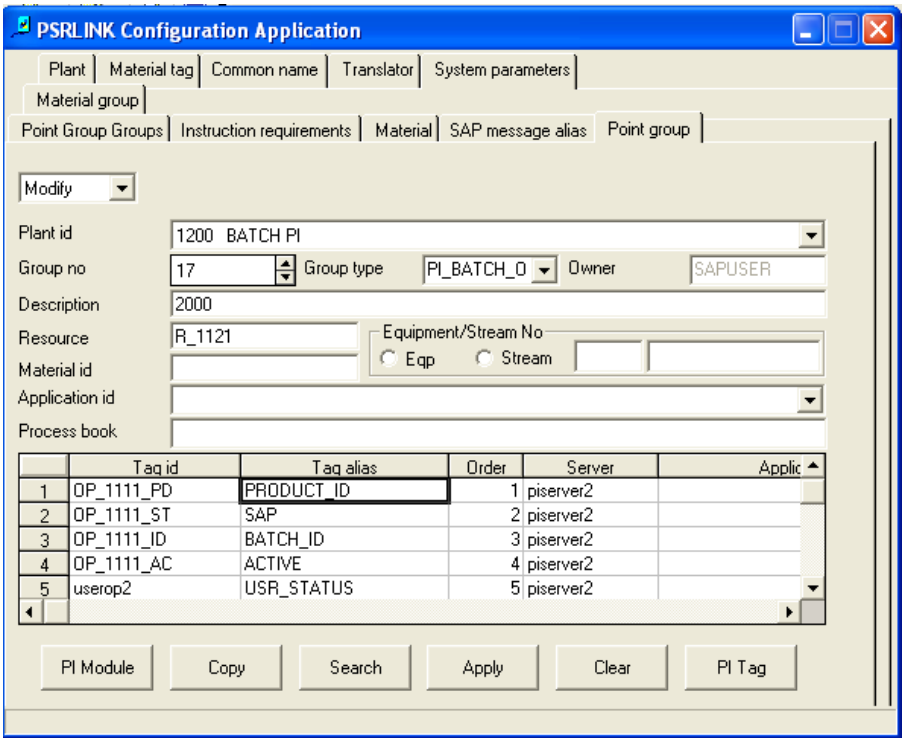


The resource on the point group for the RECIPE must match the plant resource network in the AORD instruction in the recipe.

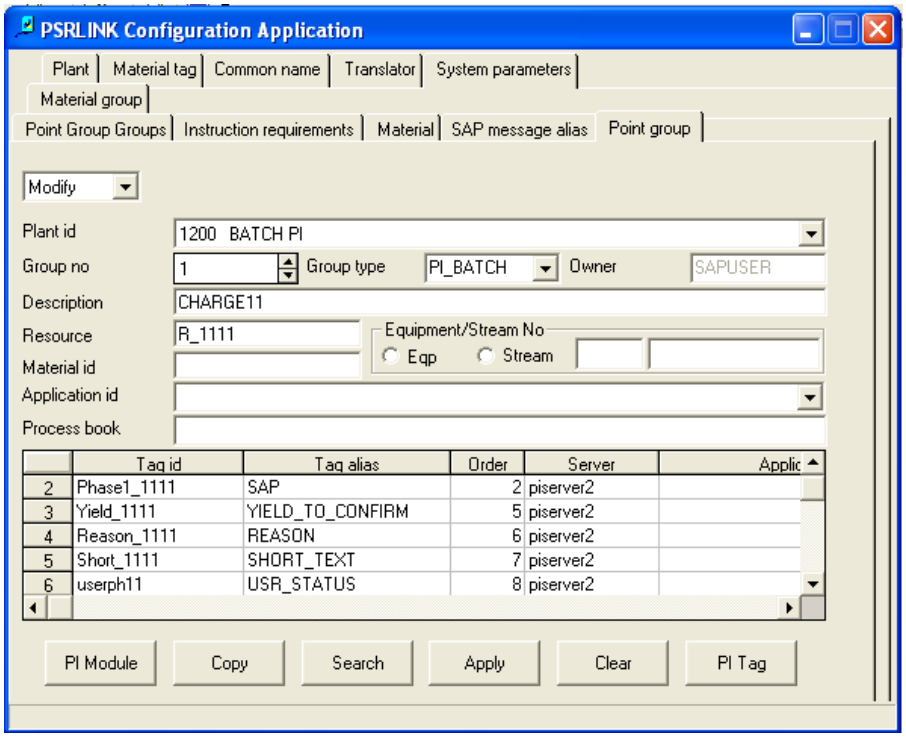


If you want the characteristics PPPI_YIELD_TO_CONFIRM, PPPI_CONFIRMATION_SHORT_TEXT, PPPI_REASON_FOR_VARIANCE or you want to use the PI_PHST instruction for a phase then you must setup the phase point group as shown above with the alias values of YIELD_TO_CONFIRM, SHORT_TEXT and REASON respectively..

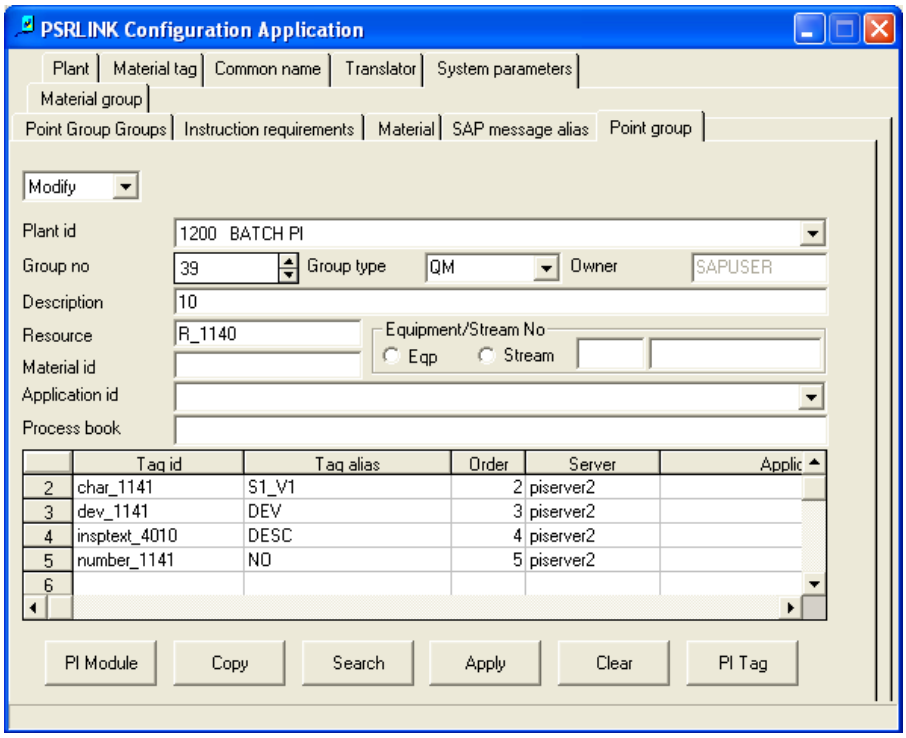
If you want to use the instruction PI_OPHUST then you must setup the operation phase group as follows:



If you want to use the instruction PI_PHUST then you must setup the operation phase group as follows:



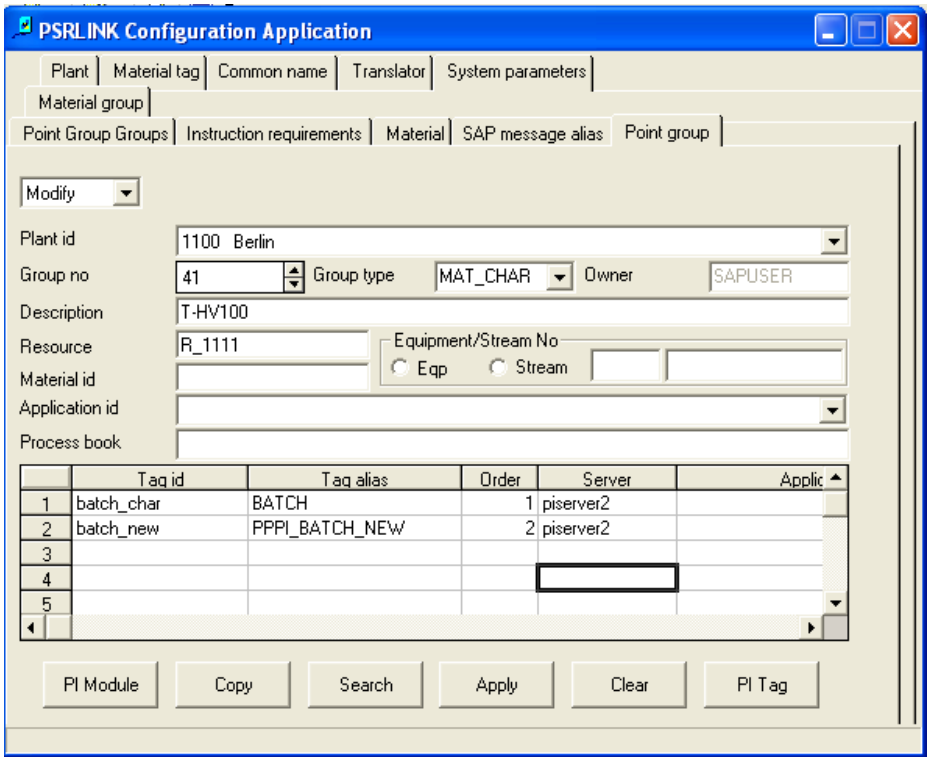
If you want to use the instruction PI_QMSMR then you must setup a QM group as follows:



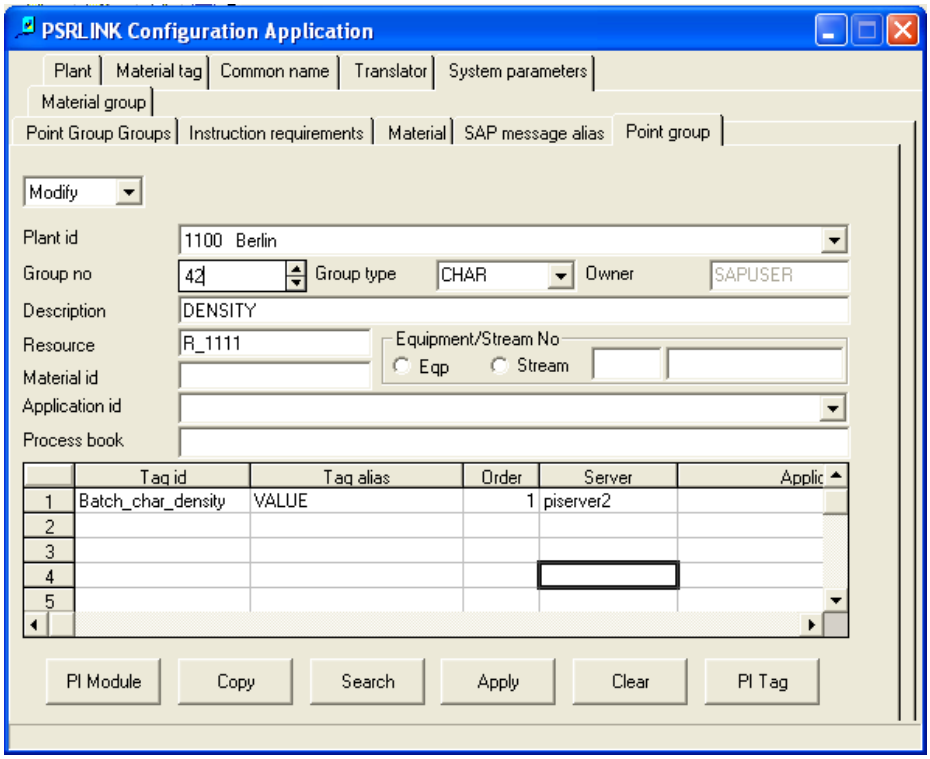
If you want to use the message PI_SRST then the group_type is SEC_RES and the required alias values are SAP, BATCH_ID and optionally REASON and SHORT_TEXT. If PI_SRCON is being used ACTIVITY_1 …ACTIVITY_6 and

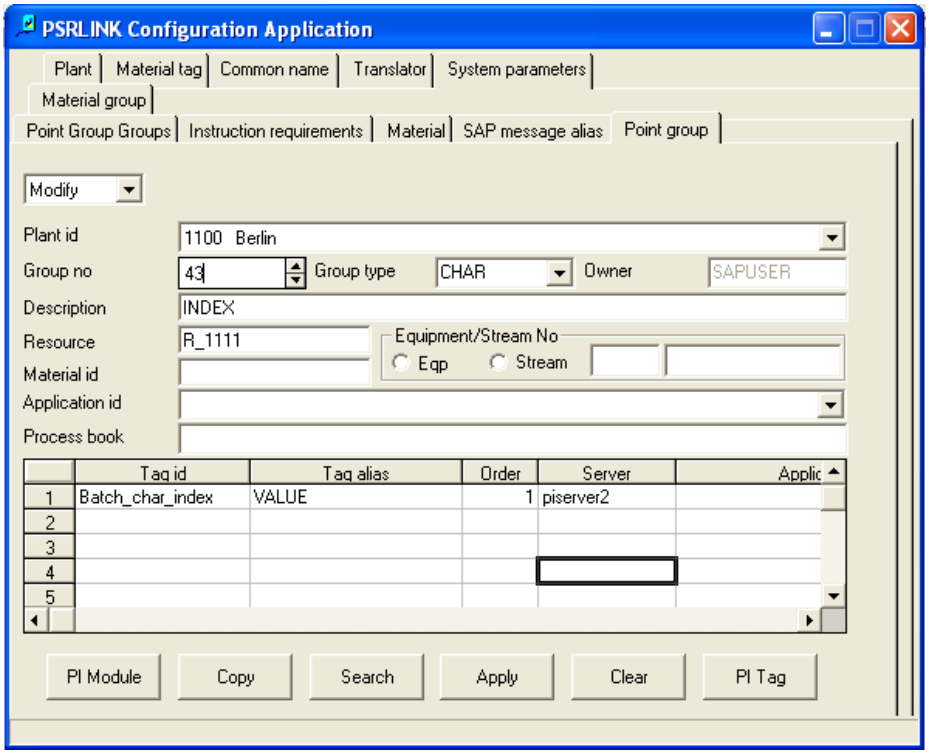
ACTIVITY_FINISH_1 …ACTIVITY_FINISH_6 , POST_DATE and SHORT_TEXT are used in addition to SAP and BATCH_ID. The resource is configured to be the secondary resource, the description is the PPPI_EXTERNAL_PHASE.

If you are going to report batch characteristics with PI_BTCL instruction then you must use the Point grouping functionality. The following example shows what groups must be set up for a sample problem. First you must make the group for the material.

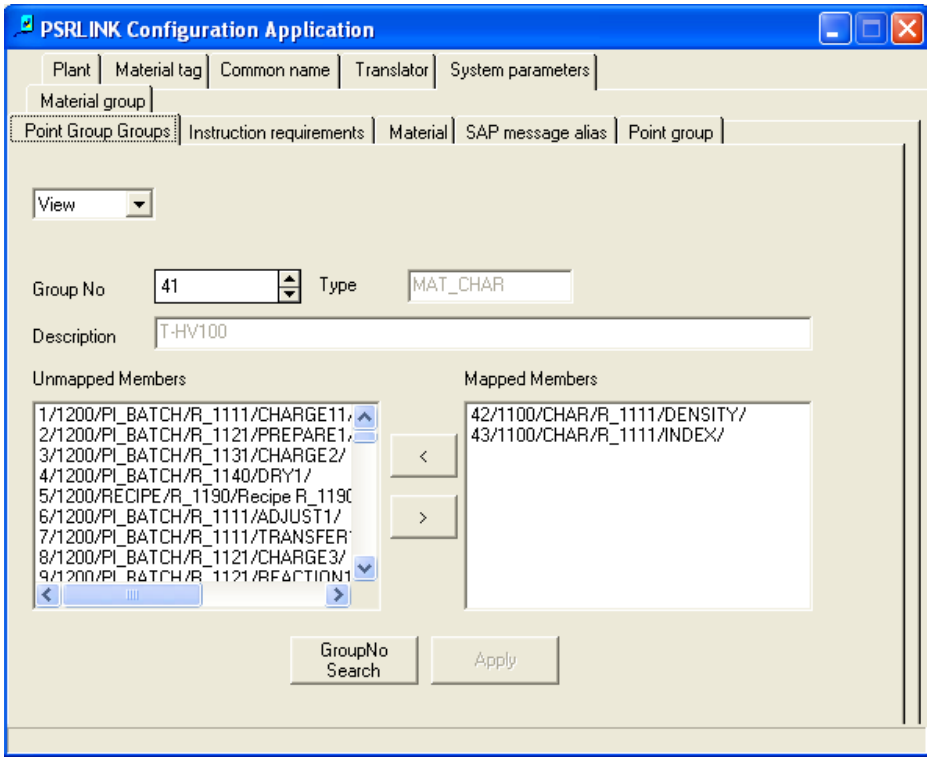


Then there must be a group created for each of the batch characteristics to be monitored. For example if there were two characteristics DENSITY and INDEX there would be two groups defined.





Then you need to group these together.



In this example the grouping would be entered in the table point_group_groups as follows:

point_group_no	point_group_member_no
382	383
382	384

Point_Group Table

The Point_group and Point_group_members tables would be used to associate a group of points together for PI_BATCH or to identify the recipe and phase status tags for a batch execution system. The details of the data entry are discussed in the chapter on PI and PI Batch Specifics.

Table Field	***Meaning***
Group_num	Unique no of group
Group_description	Description of group
Group_type	Type of group for example PI_BATCH
Process_book	Not used
Resource_id	Resource network
Plant_id	Plant id
Equipment_no	Equipment id
Owner	Owner of this point group for change modification
Application	Application which uses this point group

Table Field	*Meaning*
Material_id	Material no

Setup of the Point_Group_Members Table

Table Field	*Meaning*
Group_num	Unique no of group
Tag_id	Tag name
Tag_alias	Alias description for tag
Display_order	Order within group to display the tag
Server	Server for the PI tag
Application	Application Number

Setup of Point_group_groups Table

This table is required if you are going to do a grouping of groups. This is used to for the instruction PI-BTCL. In this case there is a master group for the material and then a group for each of the characteristics of the material with group type CHAR. Entries are made in point_group_groups to associate the characteristic group with the master material group of group type MAT_CHAR. At this time there is no dialog to support configuration of point_groups_groups with the configure application it must be done using Microsoft Access.

Table Field	*Meaning*
Point_group_no	Master group number
Point_group_member_no	Point group of member grout

The reserved names for the point group types and tag alias values are given in the following tables.

Group Type	*Use*
RECIPE	Recipe
PI_BATCH	Phase of the Recipe
PI_BATCH_O	Operation of the Recipe
ERROR	Alarm tags for the RLINK diagnostics
QM	QM
SEC_RES	Secondary Resource
MAT_CHAR	Material which will have characteristics, ie. PI_BTCR and PI_BTCL
CHAR	Specific characteristic of material PI_BTCL
SAP_TRAN	General PP Instruction

Group Type	*Tag Alias*	*Use Description*
RECIPE		PI_CRST
	RECIPE_ID	Control recipe status, stores the recipe number PPPI_CONTROL_RECIPE_STATUS
	SAP	Control recipe status stores the status of the recipe PPPI_CONTROL_RECIPE_STATUS
	PO	Can be used to store the Process Oder of the recipe (optional)
	PRODUCT_ID	Material to be made (optional)
PI_BATCH		PI_PHST, PI_PHUST, PI_PHCON
	BATCH_ID	For each phase stores the recipe used for the phase status. PPPI_PHASE_STATUS
	SAP	For each phase stores the status used for the phase status. PPPI_PHASE_STATUS
	YIELD_TO_CONFIRM	For each phase stores the yield used for the phase status. PPPI_YIELD_TO_CONFIRM
	REASON	For each phase stores the reason used for the phase status. PPPI_REASON_FOR_VARIANCE
	SHORT_TEXT	For each phase stores the short text used for the phase status. PPPI_CONFIRMATION_SHORT_TEXT
	USR_STATUS	For each phase stores the user status used for the phase status. PI_PHUST PPPI_PHASE_USER_STATUS
	ACTIVITY_1	For each phase time ticket use to store PPPI_ACTIVITY_1
	ACTIVITY_2	For each phase time ticket use to store PPPI_ACTIVITY_2
	ACTIVITY_3	For each phase time ticket use to store PPPI_ACTIVITY_3
	ACTIVITY_4	For each phase time ticket use to store PPPI_ACTIVITY_4
	ACTIVITY_5	For each phase time ticket use to store PPPI_ACTIVITY_5
	ACTIVITY_6	For each phase time ticket use to store PPPI_ACTIVITY_6

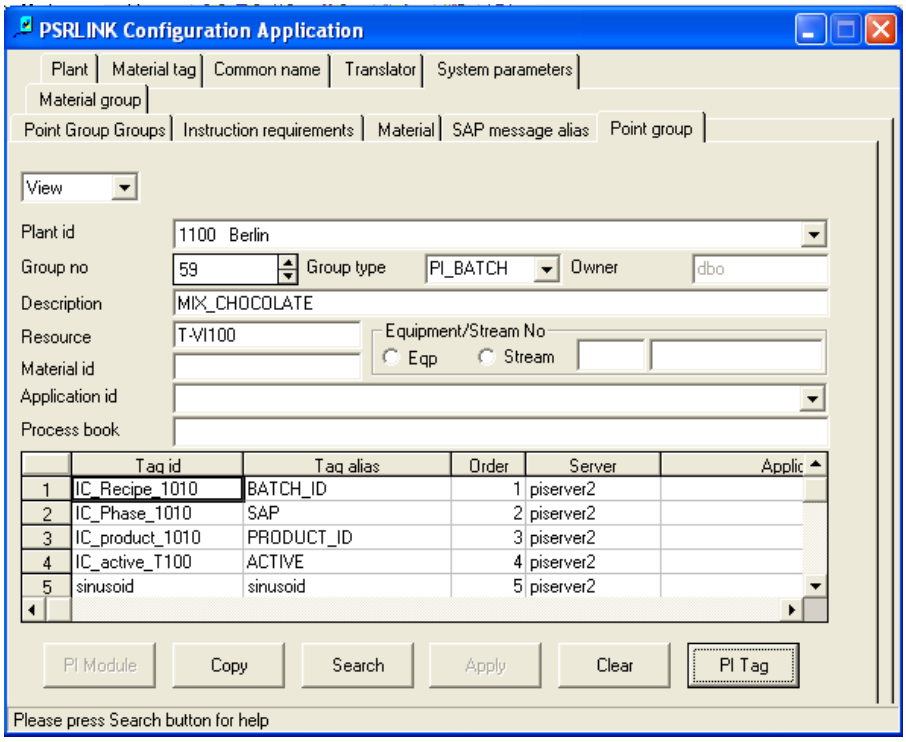
Group Type	*Tag Alias*	*Use Description*
	ACTIVITY_FINISH_1	For each phase time ticket use to store PPPI_ACTIVITY_1_FINISHED
	ACTIVITY_FINISH_2	For each phase time ticket use to store PPPI_ACTIVITY_2_FINISHED
	ACTIVITY_FINISH_3	For each phase time ticket use to store PPPI_ACTIVITY_3_FINISHED
	ACTIVITY_FINISH_4	For each phase time ticket use to store PPPI_ACTIVITY_4_FINISHED
	ACTIVITY_FINISH_5	For each phase time ticket use to store PPPI_ACTIVITY_5_FINISHED
	ACTIVITY_FINISH_6	For each phase time ticket use to store PPPI_ACTIVITY_6_FINISHED
	SCRAP	For each phase time ticket use to store PPPI_SCRAP_TO_CONFIRM
	POST_DATE	For each phase time ticket use to store PPPI_POSTING_DATE
	PRODUCT_ID	Material in Phase used if creating batch records in PI (optional)
	ACTIVE	The tag used for creating batch records in PI (optional)
PI_BATCH_O		PI_OPUST
	PRODUCT_ID	If you want to make a batch record for operation stores the product (optional)
	SAP	If you want to make a batch record for operation stores the status of the operation
	BATCH_ID	If you want to make a batch record for operation stores the recipe
	ACTIVE	If you want to make a batch record for operation stores active tag for making the PI-BATCH record (optional)
	USR_STATUS	For the operation PI_OPUST the PPPI_OPERATION_USER_STATUS
ERROR		
	CRHE	The tag for errors in the CRHE table; ie recipies down from SAP
	MSHD	The tag for errors in the MSHD table; ie messages sent to SAP
	PSRLINK	The tag for errors in the error_log table; ie general error messages
	SAP_RECIPE	The tag for errors in recipies that did not come down. Requires site SAP code.

Group Type	*Tag Alias*	*Use Description*
	SAP_MSHD	The tag for errors in posting messages in SAP. Requires site SAP code.
	PMU_RCODE	The tag for errors in the message went to SAP but no return code was recieved
QM		PI_QMSMR
	S1_V1	Sample one value 1 PPPI_INSPECTION_RESULT
	DEV	PPPI_STANDARD_DEVIATION
	DESC	PPPI_INSPECTION_SHORT_TEXT
	NO	PPPI_NUMBER_OF_INSPECTIONS
	LOT	PPPI_INSPECTION_LOT
SEC_RES		PI_SRST, PI_SRACT, PI_SRCON
	SHORT_TEXT	PPPI_CONFIRMAION_SHORT_TEXT
	BATCH_ID	Holds the recipe id
	ACTIVITY_1	PPPI_ACTIVITY_1
	ACTIVITY_2	PPPI_ACTIVITY_2
	ACTIVITY_3	PPPI_ACTIVITY_3
	ACTIVITY_4	PPPI_ACTIVITY_4
	ACTIVITY_5	PPPI_ACTIVITY_5
	ACTIVITY_6	PPPI_ACTIVITY_6
	ACTIVITY_FINISH_1	PPPI_ACTIVITY_1_FINISHED
	ACTIVITY_FINISH_2	PPPI_ACTIVITY_2_FINISHED
	ACTIVITY_FINISH_3	PPPI_ACTIVITY_3_FINISHED
	ACTIVITY_FINISH_4	PPPI_ACTIVITY_4_FINISHED
	ACTIVITY_FINISH_5	PPPI_ACTIVITY_5_FINISHED
	ACTIVITY_FINISH_6	PPPI_ACTIVITY_6_FINISHED
	POST_DATE	PPPI_POSTING_DATE
	SAP	Status
	REASON	PPPI_REASON_FOR_VARIANCE
MAT_CHAR		
	BATCH	Digital state that is monitored to signal a new batch.
	PPPI_BATCH_NEW	Number of the new batch PPPI_BATCH_NEW
	PPPI_ORDER_ITEM_NUMBER	PPPI_ORDER_ITEM_NUMBER

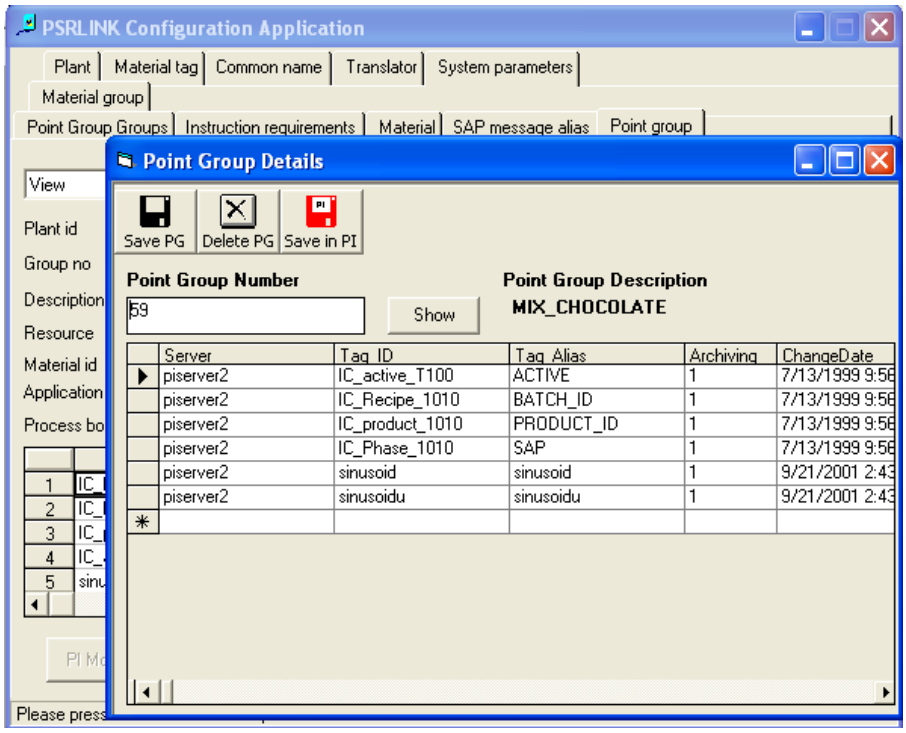
Group Type	*Tag Alias*	*Use Description*
	PO	Holds the process order
	PPPI_STOCK_TYPE	Holds material PPPI_STOCK_TYPE
	PPPI_STORAGE_LOCATION	Holds material PPPI_STORAGE_LOCATON
CHAR		PI_BT_CL
	VALUE	PPPI_BATCH_CHARAC_VALUE
SAP_TRAN		General SAP transaction the characteristics are specific for the transaction used this is only an example
	PPPI_MATERIAL_CONSUMED	
	PPPI_MESSAGE_CATEGORY	
	STATE	
	PPPI_BATCH	
	OTHER_SLOC	
	PPPI_MATERIAL	
	OTHER_MATERIAL	
	OTHER_BATCH	
	PPPI_STORAGE_LOCATION	
	MOVEMENT_TYPE	
	OTHER_PLANT	
	PPPI_PLANT_OF_RESOURCE	

Point Group Tag Generation

From the point group tab you can create and modify the tags that are associated with the group. Select the button called PI Tag and this will display the second screen.



From the Point Group Details screen you can change any of the PI tag information. You select the rows you want to have changed and then the save in PI button and the save PG button. If there are problem with the PI tags verify that the AboutPI-sdk.exe in the pipi\pisdk directory is working correctly and is configured to talk to the correct PI server.

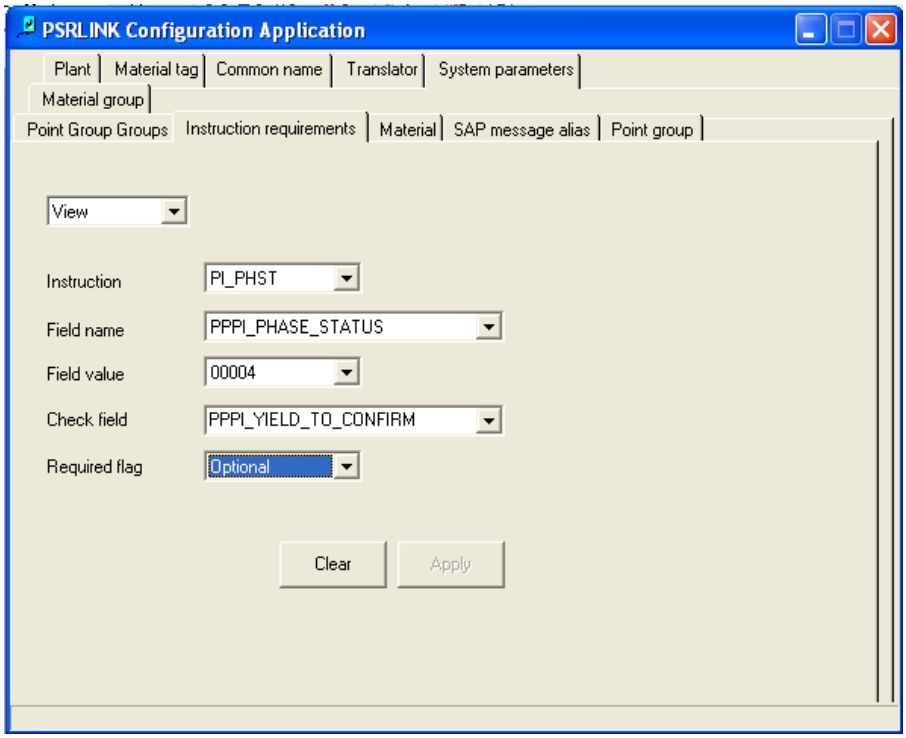


Instruction Requirements

Instruction Requirements table is required to set the status of whether certain characteristics must be present before an instruction is returned to SAP. Currently this is used for PI_PHST, PI_PHCON, PI_SRCON, PI_SRST instructions to specify whether

PPPI_YIELD_TO_CONFIRM and other such characteristics are required. If you are going to send back a yield to confirm you must set up point groups for the phases.

Table Field	*Meaning*
id	Automatic number assigned
Instruction	Instruction you are checking ex PI_PHST
Field_name	Field name which will determine if check_field is to be returned in the Instruction ex PPPI_PHASE_STATUS
Field_value	Value of the field_name ex 0001
Check_field	Name of characteristic which is to be returned
Requrired_flag	Values are Y= Yes, N= No, O= Optional

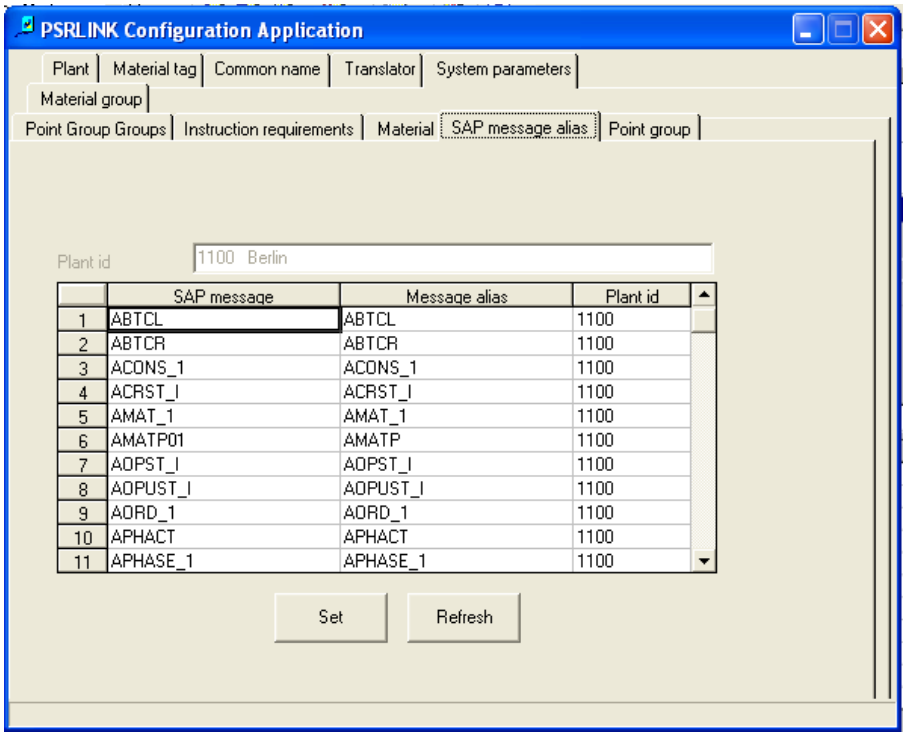


SAP/R3 Message Alias

If the user does not wish to use the standard names for SAP/R3 Instructions an alias table can be configured that will set up an alias name for the standard instructions.

- Setup for table for SAP/R3 Instruction SAP_Message_Alias

Table Field	*Meaning*
SAP_message	Class of Alias for example material
Alias_message	Description of class
Plant_id	Plant



Alias for OSI Characteristics

If you want to use alias values for the OSI characteristics in the SAP recipe OSI_START_TIME, OSI_START_DATE, OSI_FINISH_TIME, OSI_FINISH_DATE, OSI_NO_VLAUES, OSI_AVG_TYPE, OSI_EXTERNAL_RECIPE, OSI_ORDER_QUANITY then you must configure the alias_system, alias_class and external_alias tables. Below is shown a sample configuration for alias values. The description of these tables is given in the chapter on customizing.

Alias_class

alias_class	alias_class_desc
SAP-PPPI	SAP external phase

Alias_system

alias_system_id	alias_system	alias_system_desc	language	plant_id
6	SAP	SAP PP-PI	E	1100

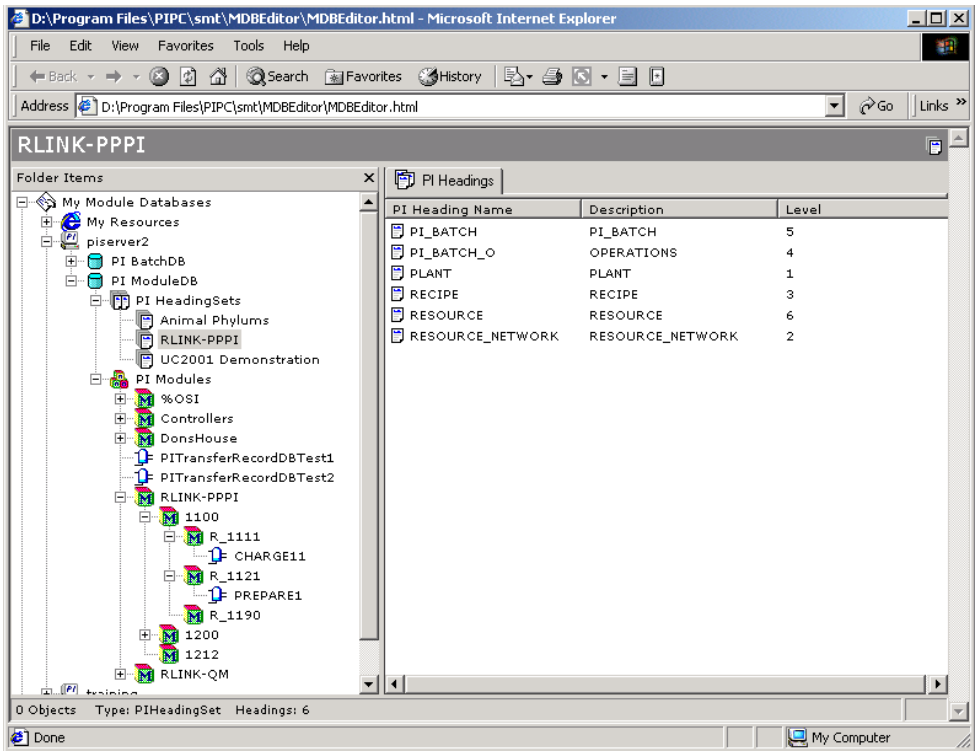
External_alias

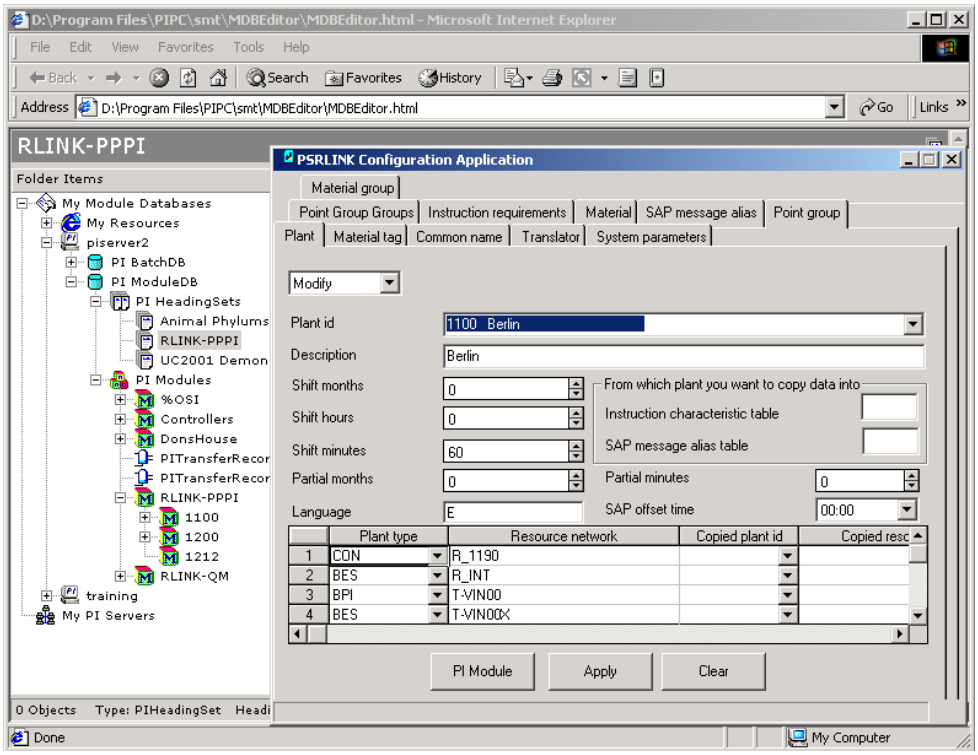
alias_value	internal_value	alias_system_id	alias_description	alias_class
ZOSI_FINISH_DATE	OSI_FINISH_DATE	6	OSI_FINISH_DATE	SAP-PPPI
ZOSI_FINISH_TIME	OSI_FINISH_TIME	6	OSI_FINISH_TIME	SAP-PPPI
ZOSI_START_DATE	OSI_START_DATE	6	OSI_START_DATE	SAP-PPPI
ZOSI_START_TIME	OSI_START_TIME	6	OSI_START_TIME	SAP-PPPI

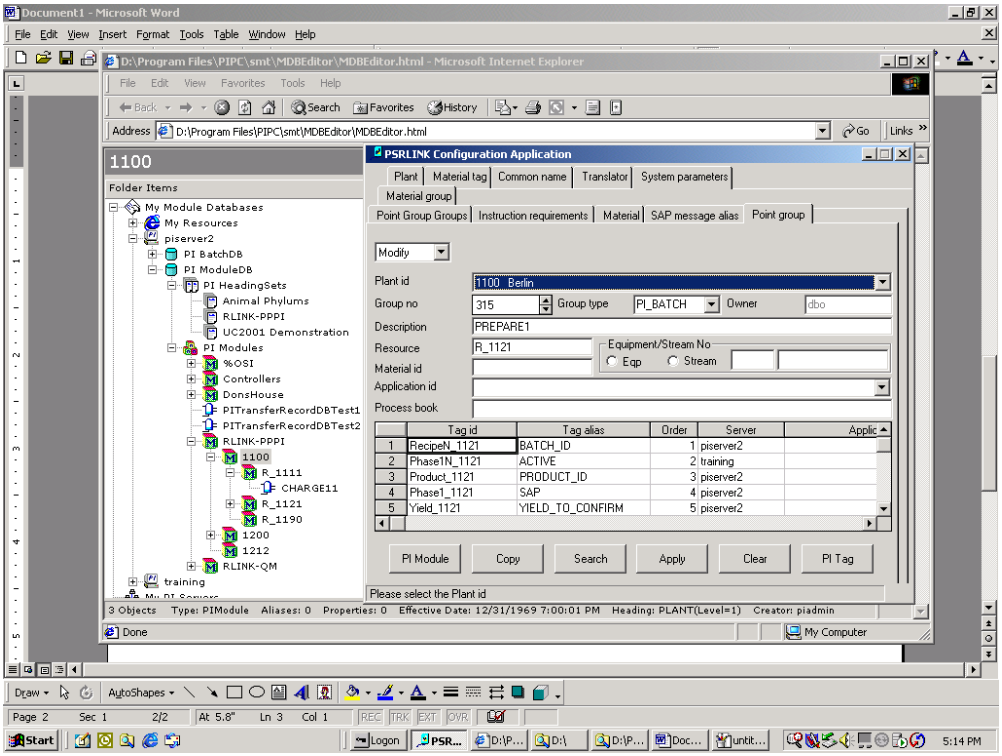
You must also change the table instruction_characteristics for the instruction_characteristics that you give alias values for.

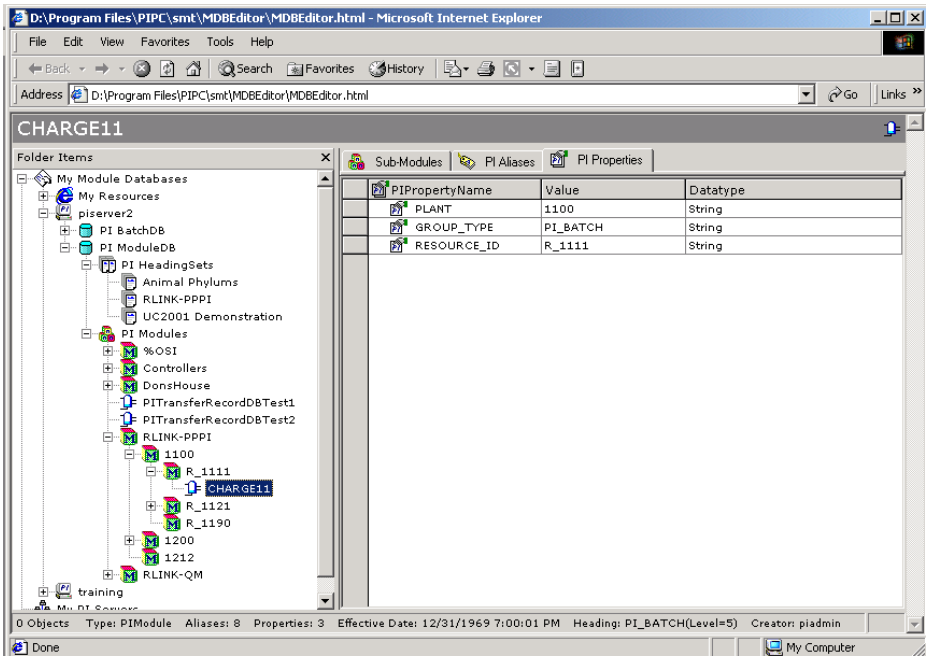
PI Modules

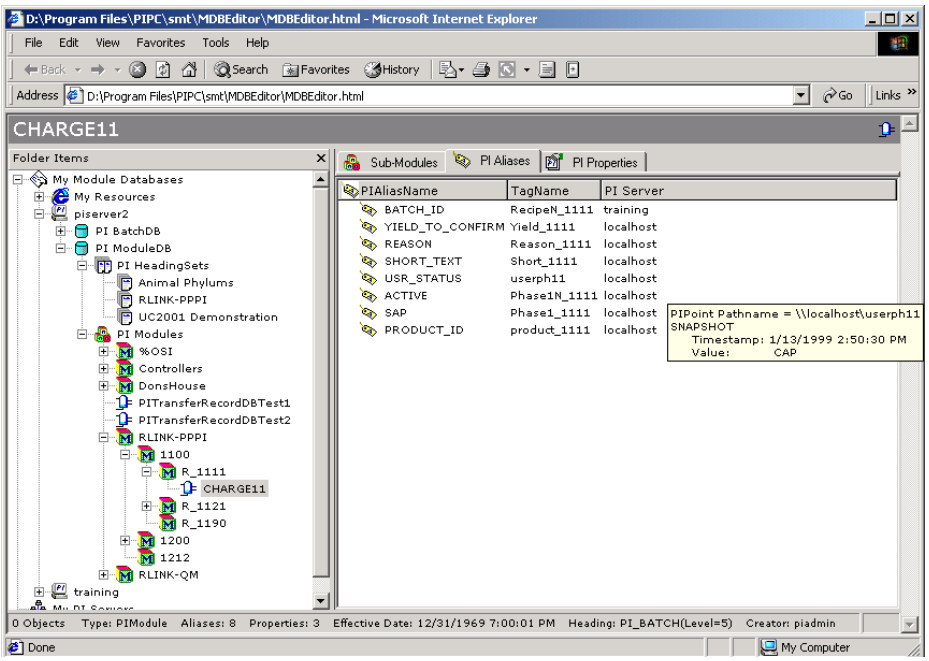
In the configuration application you have the option to create PI-Modules that can be used in ProcessBook type applications. These are created in the structure shown in the following displays. You create them by being in the modify mode on the Configuration Application and selecting the PI-Modules button.

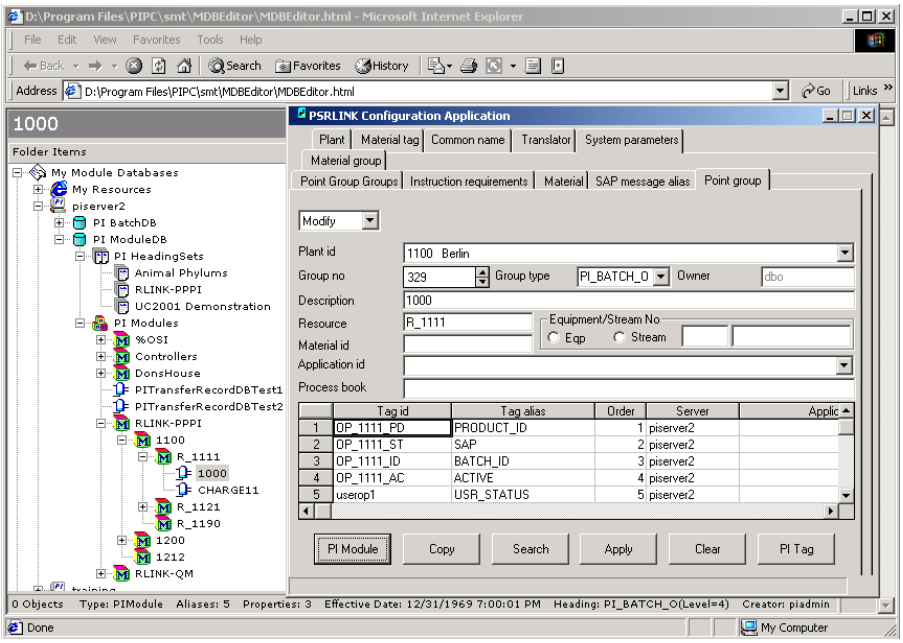


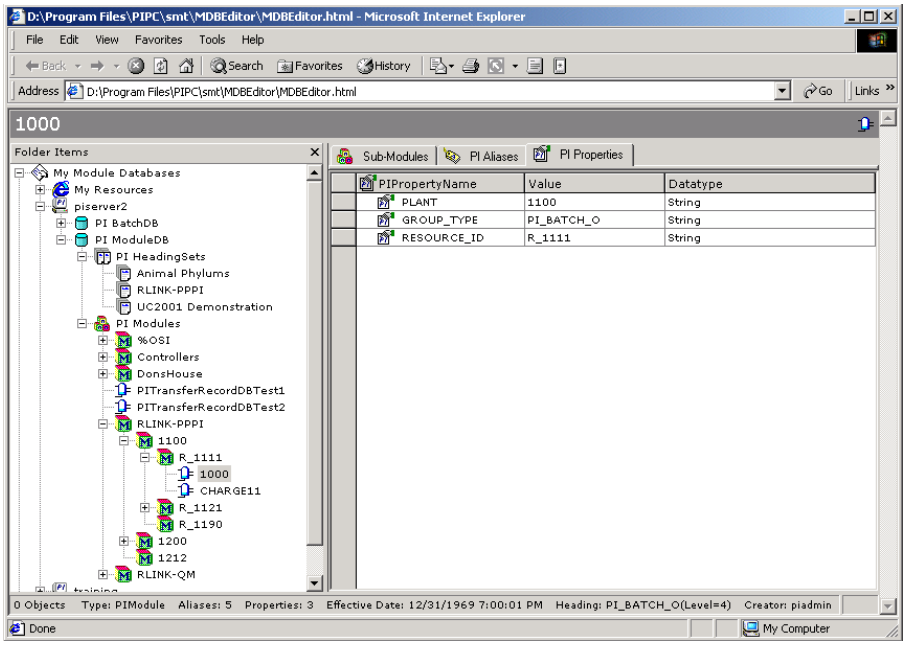


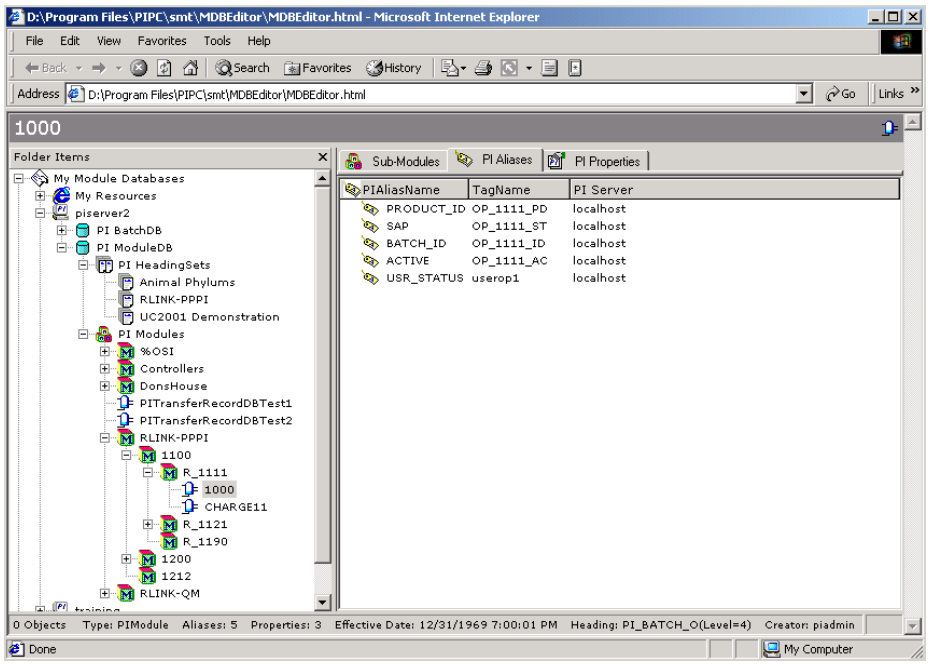

Document1 - Microsoft Word
D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
1100
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
training
PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Material group
Point Group Groups
Instruction requirements
Material
SAP message alias
Point group
Modify
Plant id
1100 Berlin
Group no
315
Group type
PI_BATCH
Owner
dbo
Description
PREPARE1
Resource
R_1121
Equipment/Stream No
Eqp
Stream
Material id
Application id
Process book
Tag id
Tag alias
Order
Server
RecipeN_1121
BATCH_ID
Phase1N_1121
ACTIVE
training
Product_1121
PRODUCT_ID
Phase1_1121
SAP
Yield_1121
YIELD_TO_CONFIRM
PI Module
Copy
Search
Apply
Clear
PI Tag
Please select the Plant id
3 Objects Type: PIModule Aliases: 0 Properties: 0 Effective Date: 12/31/1969 7:00:01 PM Heading: PLANT(Level=1) Creator: piadmin

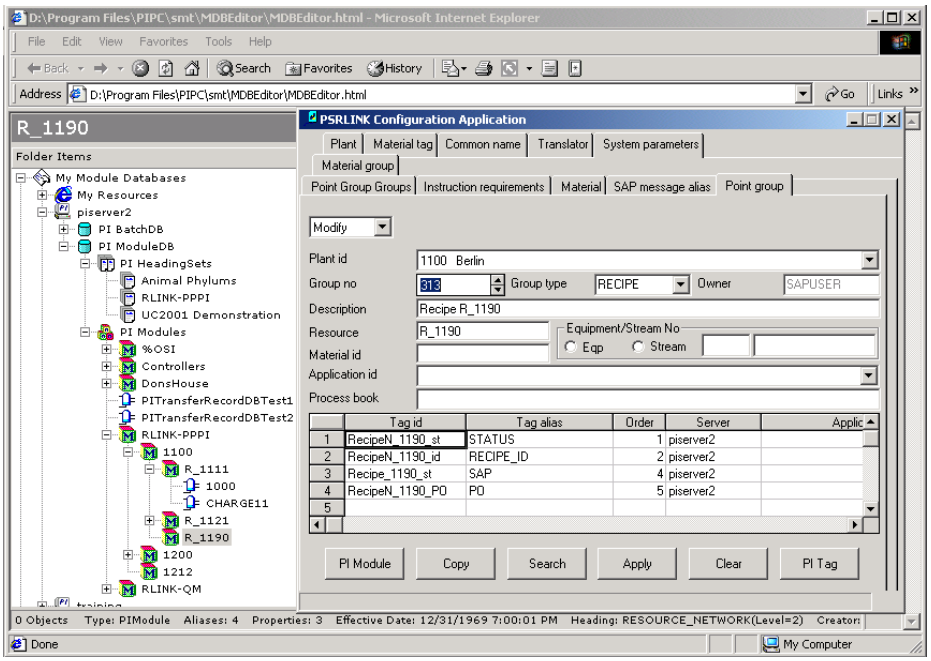

D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
CHARGE11
Folder Items
Sub-Modules
PI Aliases
PI Properties
PIPropertyName
Value
Datatype
PLANT
1100
String
GROUP_TYPE
PI_BATCH
String
RESOURCE_ID
R_1111
String
0 Objects Type: PIModule Aliases: 8 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: PI_BATCH(Level=5) Creator: piadmin

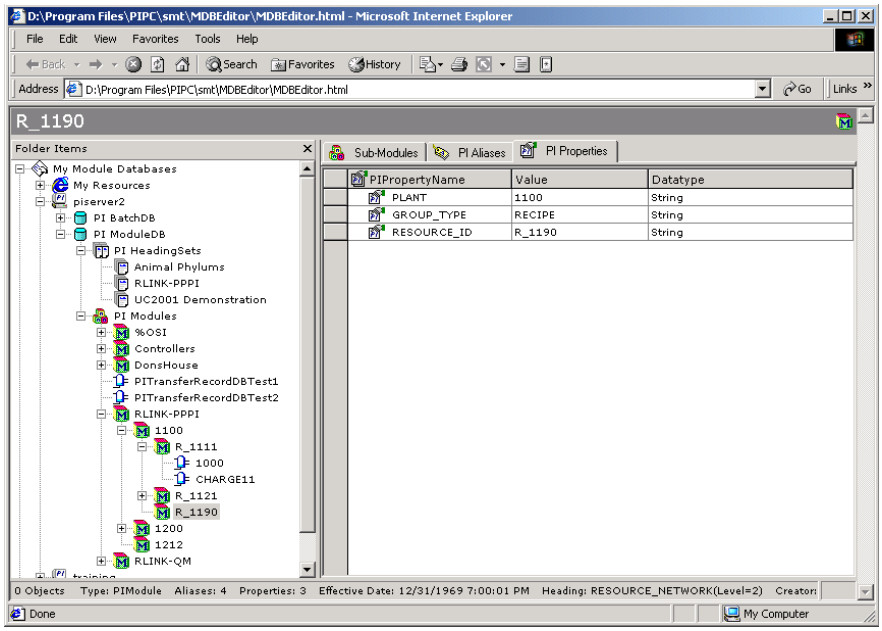

D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
CHARGE11
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
training
Sub-Modules
PI Aliases
PI Properties
PIAliasName TagName PI Server
BATCH_ID RecipeN_1111 training
YIELD_TO_CONFIRM Yield_1111 localhost
REASON Reason_1111 localhost
SHORT_TEXT Short_1111 localhost
USR_STATUS userph11 localhost
ACTIVE Phase1N_1111 localhost
SAP Phase1_1111 localhost
PRODUCT_ID product_1111 localhost
PIPoint Pathname = \\localhost\userph11
SNAPSHOT
Timestamp: 1/13/1999 2:50:30 PM
Value: CAP
0 Objects Type: PIModule Aliases: 8 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: PI_BATCH(Level=5) Creator: piadmin
Done
My Computer


D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
1000
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
1000
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Material group
Point Group Groups
Instruction requirements
Material
SAP message alias
Point group
Modify
Plant id 1100 Berlin
Group no 329
Group type PI_BATCH_O
Owner dbo
Description 1000
Resource R_1111
Equipment/Stream No
Eqp
Stream
Material id
Application id
Process book
Tag id Tag alias Order Server Applic
1 OP_1111_PD PRODUCT_ID 1 piserver2
2 OP_1111_ST SAP 2 piserver2
3 OP_1111_ID BATCH_ID 3 piserver2
4 OP_1111_AC ACTIVE 4 piserver2
5 userop1 USR_STATUS 5 piserver2
PI Module
Copy
Search
Apply
Clear
PI Tag
0 Objects Type: PIModule Aliases: 5 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: PI_BATCH_O(Level=4) Creator: piadmin
Done
My Computer


D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
1000
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
1000
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
Sub-Modules
PI Aliases
PI Properties
PIPropertyName Value Datatype
PLANT 1100 String
GROUP_TYPE PI_BATCH_O String
RESOURCE_ID R_1111 String
0 Objects Type: PIModule Aliases: 5 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: PI_BATCH_O(Level=4) Creator: piadmin
Done
My Computer


D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
1000
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
1000
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
Sub-Modules
PI Aliases
PI Properties
PIAliasName TagName PI Server
PRODUCT_ID OP_1111_PD localhost
SAP OP_1111_ST localhost
BATCH_ID OP_1111_ID localhost
ACTIVE OP_1111_AC localhost
USR_STATUS userop1 localhost
0 Objects Type: PIModule Aliases: 5 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: PI_BATCH_O(Level=4) Creator: piadmin
Done
My Computer


D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
R_1190
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
1000
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Material group
Point Group Groups
Instruction requirements
Material
SAP message alias
Point group
Modify
Plant id
1100 Berlin
Group no
313
Group type
RECIPE
Owner
SAPUSER
Description
Recipe R_1190
Resource
R_1190
Equipment/Stream No
Eqp
Stream
Material id
Application id
Process book
Tag id
Tag alias
Order
Server
1 RecipeN_1190_st STATUS 1 piserver2
2 RecipeN_1190_id RECIPE_ID 2 piserver2
3 Recipe_1190_st SAP 4 piserver2
4 RecipeN_1190_PO PO 5 piserver2
5
PI Module
Copy
Search
Apply
Clear
PI Tag
0 Objects Type: PIModule Aliases: 4 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: RESOURCE_NETWORK(Level=2) Creator:
Done
My Computer


D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\Program Files\PIPC\smt\MDBEditor\MDBEditor.html
R_1190
Folder Items
My Module Databases
My Resources
piserver2
PI BatchDB
PI ModuleDB
PI HeadingSets
Animal Phylums
RLINK-PPPI
UC2001 Demonstration
PI Modules
%OSI
Controllers
DonsHouse
PITransferRecordDBTest1
PITransferRecordDBTest2
RLINK-PPPI
1100
R_1111
1000
CHARGE11
R_1121
R_1190
1200
1212
RLINK-QM
Sub-Modules
PI Aliases
PI Properties
PIPropertyName Value Datatype
PLANT 1100 String
GROUP_TYPE RECIPE String
RESOURCE_ID R_1190 String
0 Objects Type: PIModule Aliases: 4 Properties: 3 Effective Date: 12/31/1969 7:00:01 PM Heading: RESOURCE_NETWORK(Level=2) Creator:
Done
My Computer

Chapter 5
Recipe Setup

SAP/R3 Instructions

A recipe must contain the AORD, AMAT_1 and APHASE_1 instructions since these are used to setup the basic structure of the recipe to be processed. The recipe must also include the ACRST_I and APHST_I instruction since these are the requests that tell SAP/R3 that the recipe or phase have been completed or terminated.

The remaining instructions are added depending upon what data is required by SAP/R3 or what data is to be sent to the Industrial Desktop environment.

- The following instructions can be used in the SAP/R3 recipe

AORD_1	Included	OSI_START_TIME	
	Included	OSI_START_DATE,	
	Included	OSI_FINISH_TIME	
	Included	OSI_FINISH_DATE,	
	Included	OSI_EXTERNAL_RECIPE	
	Included	OSI_ORDER_QUANTITY	
	Included	PPPI_ORDER_QUANTITY	
	Included	PPPI_RESOURCE_NETWORK	
	Included	PPPI_PLANT_OF_RESOURCE	
APHASE_1	Included	PPPI_PHASE	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE_RESOURCE	
	Included	PPPI_PHASE_SHORT_TEXT	
	Included	PPPI_EXTERNAL_PHASE	
AMAT	Included	PPPI_BATCH	
	Included	PPPI_MATERIAL	

	Included	PPPI_MATERIAL_ITEM	
	Included	PPPI_RESERVATION_ITEM	
	Included	PPPI_RESERVATION	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_MATERIAL_QUANTITY	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_STORAGE_LOCATION	V 1.5
	Included	PPPI_MATERIAL_BY_PRODUCT	V 1.6
	Included	PPPI_MATERIAL_CO_PRODUCT	V 1.6
AMATP01	Included	PPPI_BATCH	
	Included	PPPI_MATERIAL	
	Included	PPPI_PHASE (if this does not come down will be assigned to last phase)	
	Included	PPPI_MATERIAL_QUANTITY	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_OPERATION (if this does not come down will be assigned to last operation)	
	Included	PPPI_STORAGE_LOCATION	V 1.5
ACRST_I	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_CONTROL_RECIPE_STATUS	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_CONTROL_RECIPE	
APHST_I	Included	PPPI_PHASE_RESOURCE	
	Included	PPPI_PLANT_OF_RESOURCE (sending down)	
	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_PHASE_STATUS	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_YIELD_TO_CONFIRM	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_CONFIRMATION_SHORT_TEXT	
	Included	PPPI_REASON_FOR_VARIANCE	

AOPST_I	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPPI_OPERATION_STATUS	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_EVENT_DATE	
	Included	PPPI-MESSAGE_CATEGORY	
APROD_1	Not Supported	PPPI_ORDER_ITEM_NUMBER,	
	Included	PPPI_DELIVERY_COMPLETE	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_BATCH	
	Included	PPPI_STORAGE_LOCATION	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_MATERIAL_PRODUCED	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_MATERIAL	
	Included	PPPI_MESSAGE_CATEGORY	
	Included (use misc tag)	PPPI_STOCK_TYPE (sending down)	
ACONS_1	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_MATERIAL	
	Included	PPPI_BATCH	
	Included	PPPI_STORAGE_LOCATION	
	Included	PPPI_RESERVATION	
	Included	PPPI_RESERVATION_ITEM	
	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_MATERIAL_CONSUMED	
	Included	PPPI_UNIT_OF_MEASURE	
	Included (use misc tag)	PPPI_FINAL_ISSUE	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_EVENT_TIME	
APHPAR_1	Included	PPPI_PHASE	
	Included	PPPI_PARAMETER_NAME	

	Included	PPPI_PARAMETER_VALUE	
	Included	PPPI_PARAMETER_VALUE_MIN	
	Included	PPPI_PARAMETER_VALUE_MAX	
	Included	PPPI_UNIT_OF_MEASURE	
APHACT	Included	PPPI_CONFIRMATION_SHORT_TEXT	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_ACTIVITY	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_STD_VALUE_PARAMETER_ID	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_STATUS_CONFIRMED	
AREAD1	Included	OSI_VAG_TYPE	
	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_DATA_POINT_NAME	
	Included	PPPI_DATA_POINT_VALUE	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_PHASE	
	Included	PPPI_OPERATION	
	Included	PPPI_UNIT_OF_MEASURE	
AREAD2	Included	OSI_NO_VALUES	
	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_DATA_POINT_NAME	
	Included	PPPI_DATA_POINT_VALUE	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_PHASE	
	Included	PPPI_OPERATION	
	Included	PPPI_UNIT_OF_MEASURE	
AQMSMR_1	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PHASE	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	

	Included	PPPI_INSPECTION_LOT	
	Included	PPPI_INSPECTION_CHARACTERISTIC	
	Included	PPPI_NUMBER_OF_INSPECTIONS	
	Included	PPPI_INSPECTION_SHORT_TEXT	
	Included	PPPI_INSPECTION_RESULT	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_UNIT_OF_MEASURE	
	Included	PPPI_STANDARD_DEVIATION	
ASRST	Included	PPPI_MESSAGE_CATEGORY	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_SECONDARY_RESOURCE	V 1.35
	Included	PPPI_SECONDARY_RESOURCE_STATUS	V 1.35
	Included	PPPI_REASON_FOR_VARIANCE	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_RESOURCE	V 1.35
	Included	PPPI_PLANT_OF_RESOURCE (sending down)	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
ASRACT	Included	PPPI_MESSAGE_CATEGORY	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_SECONDARY_RESOURCE	V 1.35
	Included	PPPI_STD_VALUE_PARAMETER_ID	V 1.35
	Included	PPPI_ACTIVITY	V 1.35
	Included	PPPI_UNIT_OF_MEASURE	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_STATUS_CONFIRMED	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
APHUST_I	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_PHASE	
	Included	PPPI_PHASE_USER_STATUS	

	Included	PPPI_LANGUAGE_OF_USER_STATUS (sending down)	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_EVENT_DATE	
AOPUST_I	Included	PPPI_MESSAGE_CATEGORY	
	Included	PPPI_PROCESS_ORDER	
	Included	PPPI_OPERATION	
	Included	PPPI_OPERATION_USER_STATUS	
	Included	PPPI_EVENT_TIME	
	Included	PPPI_EVENT_DATE	
	Included	PPPI_LANGUAGE_OF_USER_STATUS (sending down)	
ABTCL	Included	PPPI_MESSAGE_CATEGORY	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_BATCH_CHARAC_VALUE	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_PLANT_OF_BATCH	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_MATERIAL	V 1.35
	Included	PPPI_ORDER_ITEM_NUMBER	V 1.35
	Included	PPPI_BATCH	V 1.35
	Included	PPPI_BATCH_CHARAC_NAME	V 1.35
ABTCR	Included	PPPI_MESSAGE_CATEGORY	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_BATCH_NEW	V 1.35
	Included	PPPI_PLANT_OF_BATCH	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_MATERIAL	V 1.35
	Included	PPPI_ORDER_ITEM_NUMBER	V 1.35
	Included	PPPI_PHASE	V 1.35
APHCON	Included	PPPI_ACTIVITY_1	V 1.35
	Included	PPPI_ACTIVITY_1_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_1_UNIT	V 1.35
	Included	PPPI_ACTIVITY_2	V 1.35
	Included	PPPI_ACTIVITY_2_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_2_UNIT	V 1.35
	Included	PPPI_ACTIVITY_3	V 1.35
	Included	PPPI_ACTIVITY_3_FINISHED	V 1.35

	Included	PPPI_ACTIVITY_3_UNIT	V 1.35
	Included	PPPI_ACTIVITY_4	V 1.35
	Included	PPPI_ACTIVITY_4_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_4_UNIT	V 1.35
	Included	PPPI_ACTIVITY_5	V 1.35
	Included	PPPI_ACTIVITY_5_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_5_UNIT	V 1.35
	Included	PPPI_ACTIVITY_6	V 1.35
	Included	PPPI_ACTIVITY_6_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_6_UNIT	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_PHASE_RESOURCE	V 1.35
	Included	PPPI_PLANT_OF_RESOURCE	V 1.35
	Included	PPPI_POSTING_DATE	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_SCRAP_TO_CONFIRM	V 1.35
	Included	PPPI_UNIT_OF_MEASURE	V 1.35
	Included	PPPI_YIELD_TO_CONFIRM	V 1.35
ASRST	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_SECONDARY_RESOURCE	V 1.35
	Included	PPPI_SECONDARY_RESOURCE_STATUS	V 1.35
	Included	PPPI_REASON_FOR_VARIANCE	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_RESOURCE	V 1.35
	Included	PPPI_PLANT_OF_RESOURCE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
ASRCON	Included	PPPI_ACTIVITY_1	V 1.35
	Included	PPPI_ACTIVITY_1_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_1_UNIT	V 1.35
	Included	PPPI_ACTIVITY_2	V 1.35
	Included	PPPI_ACTIVITY_2_FINISHED	V 1.35

	Included	PPPI_ACTIVITY_2_UNIT	V 1.35
	Included	PPPI_ACTIVITY_3	V 1.35
	Included	PPPI_ACTIVITY_3_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_3_UNIT	V 1.35
	Included	PPPI_ACTIVITY_4	V 1.35
	Included	PPPI_ACTIVITY_4_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_4_UNIT	V 1.35
	Included	PPPI_ACTIVITY_5	V 1.35
	Included	PPPI_ACTIVITY_5_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_5_UNIT	V 1.35
	Included	PPPI_ACTIVITY_6	V 1.35
	Included	PPPI_ACTIVITY_6_FINISHED	V 1.35
	Included	PPPI_ACTIVITY_6_UNIT	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
		PPPI_PHASE_RESOURCE (do not include)	
	Included	PPPI_PLANT_OF_RESOURCE	V 1.35
	Included	PPPI_POSTING_DATE	V 1.35
	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_RESOURCE	V 1.35
	Included	PPPI_STATUS_CONFIRMED	V 1.35
	Included	PPPI_SECONDARY_RESOURCE	V 1.35
ASRACT	Included	PPPI_PROCESS_ORDER	V 1.35
	Included	PPPI_OPERATION	V 1.35
	Included	PPPI_PHASE	V 1.35
	Included	PPPI_EVENT_TIME	V 1.35
	Included	PPPI_EVENT_DATE	V 1.35
	Included	PPPI_CONFIRMATION_SHORT_TEXT	V 1.35
	Included	PPPI_STATUS_CONFIRMED	V 1.35
	Included	PPPI_UNIT_OF_MEASURE	V 1.35
	Included	PPPI_STD_VALUE_PARAMETER_ID	V 1.35
	Included	PPPI_ACTIVITY	V 1.35
	Included	PPPI_SECONDARY_RESOURCE	V 1.35
	Included	PPPI_MESSAGE_CATEGORY	V 1.35
APMMD	Not Supported	PPPI_CODE_CATALOGUE	

	Not Supported	PPPI_CODE_GROUP	
	Not Supported	PPPI_DATA_POINT_NAME	
	Not Supported	PPPI_DATA_POINT_VALUE	
	Not Supported	PPPI_DIFFERENCE_READING	
	Not Supported	PPPI_EVENT_TIME	
	Not Supported	PPPI_EVENT_DATE	
	Not Supported	PPPI_NOTIFICATION_PRIO	
	Not Supported	PPPI_NOTIFICATION_TYPE	
	Not Supported	PPPI_PROCESS_ORDER	
	Not Supported	PPPI_SECONDARY_INDEX	
	Not Supported	PPPI_SHORT_TEXT	
	Not Supported	PPPI_SOURCE	
	Not Supported	PPPI_UNIT_OF_MEASURE	
	Not Supported	PPPI_USER_DATA	
	Not Supported	PPPI_VALUATION_CODE	

For each Instruction there are characteristics whose values must be retrieved and sent back to SAP/R3. In order to do this the characteristic must first have a translation method that finds the information needed to retrieve the value and then an application that actually retrieves the value or values. The translation method sets up the data needed for the application. The following tables give the list of translation methods that are

available and the list of applications for retrieving data from PI that are available.

If you do not want to use the SAP/R3 given names for these instructions setup the SAP/R3 alias name in the configuration application.

Instruction	***Purpose***	***Characteristic***	***Methods of Translation***	***Result of Processing***
AORD	Gives basic data about the order.			The tables Recipe is constructed
AMAT_1	Gives data about each material to be consumed or produced in the order			The table Material_list is loaded for the order
AMATP01	Tells what is the product which is being produced			Will construct a PI_PROD message to be returned.

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
APHASE_1	Gives data about each phase in the recipe			The tables phase, operation and operation_phases are loaded
ACRST_I	Requests the status of the recipe			
		PPPI_CONTROL_ RECIPE_STATUS	USR_GET_RECIPE_STATUS	used for continuous process
		PPPI_CONTROL_ RECIPE_STATUS	USR_RECIPE_MONITOR	used for PI-Batch or PID monitor of recipe status
APHST_I	Requests the status of the phase for time events			
		PPPI_PHASE_STATUS	USR_GET_PHASE_STATUS	used for continuous process
		PPPI_PHASE_STATUS	USR_PHASE_MONITOR	used for monitor of PI-Batch of phase status change
		PPPI_PHASE_STATUS	USR_PHASE_ALIAS_MONITOR	used for PID monitor program of phase status
		PPPI_PHASE_STATUS	USR_PHASE_EQP_MONITOR	used for PID or PI-Batch monitor when you have the same Phase name for several resources. It will select the correct set of points based upon phase name and resource.
		PPPI_YIELD_TO_CONFIRM	USR_YIELD_TO_CONFIRM	
		PPPI_YIELD_TO_CONFIRM_PARTIAL	USR_YIELD_TO_CONFIRM_PARTIAL	For continuous recipe if partial phase status is to be reported and you want the yield
		PPPI_CONFIRMATION_SHORT_TEXT	USR_CONFIRMATION-SHORT_TEXT	
		PPPI_REASON_FOR_VARIANCE	USR_REASON_FOR_VARIANCE	
		PPPI_PHASE_RESOURCE	USR_PHASE_RESOURCE	used to select the possible phases with the same name from the point groups which could be used for the resource, a maximum of 8 different phases can be monitored to see which one has been assigned to the recipe
AOPST_I	Requests the status of the operation			
		PPPI_OPERATION_ STATUS	USR_GET_OPERATION_STATUS	used for continuous process
		PPPI_OPERATION_ STATUS	USR_OPERATION_MONITIR	used of either PI-Batch or PID operation monitor

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
		PPPI_OPERATION_STATUS	USR_OPERATION_MONITOR_NEW	Used to set the tags required to get the status change in campaign manager, requires that the point group be setup for the operation
ACONS_1	Requests data about material consumed in the recipe			
		PPPI_MATERIAL_ CONSUMED	USR_BATCH_FLOW_TAG	Used for both continuous and batch
		PPPI_MATERIAL_ CONSUMED	USR_MATERIAL_FROM_BATCH	Used when you can not guarantee that the batch_id tag values and the material tag values will have the same timestamp. This will look for a batch_id value that is retrieved and then setup to get the material tag value at the timestamp of the batch_id and take the value at that time
		PPPI_STORAGE_ LOCATION	USR_GET_STORAGE_LOCATION	
		PPPI_STORAGE_LOCATION	USR_STORAGE	Used to get storage location from the material list that got filled by the AMAT instruction the application is usr_storage_matlist_app
		PPPI_BATCH	USR_BATCHID_TAG	Used to get the batchid and if multiple batches for the same material_id. The entry for the material consumed tag must have the exact time as the batch tag
		PPPI_RESERVATION	USR_RESERVATION	Used to get the reservation tag from the material_tag table and setup the start and end time for request
			USR_RS_AND_RSI	Used to get reservation from the AMAT instructions sent down based on the batch_id received back into the request looking for the batch_id value
		PPPI_RESERVATION_ITEM	USR_RESESRVATION_ITEM	Used to get the reservation_item tag from the material_tag table and setup the start and end time for request

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
			USR_RS_AND_RSI	Used to get reservation from the AMAT instructions sent down based on the batch_id received back into the request looking for the batch_id value
APROD_1	Requests data about material produced in the recipe			
		PPPI_MATERIAL_ PRODUCED	USR_BATCH_FLOW_TAG	used for continuous and batch
		PPPI_STORAGE_ LOCATION	USR_GET_STORAGE_LOCATION	
		PPPI_STORAGE_LOCATION	USR_STORAGE	Used to get storage location from the material list that got filled by the AMAT instruction the application is usr_storage_matlist_app
		PPPI_BATCH	USR_BATCHID_TAG	Used to get the batchid id multiple batches for the same material_id. The entry for the material consumed tag must have the exact time as the batch tag
		PPPI_RESERVATION	USR_RESERVATION	Used to get the reservation tag from the material_tag table and setup the start and end time for request
			USR_RS_AND_RSI	Used to get reservation from the AMAT instructions sent down based on the batch_id received back into the request looking for the batch_id value
		PPPI_RESERVATION_ITEM	USR_RESESRVATION_ITEM	Used to get the reservation_item tag from the material_tag table and setup the start and end time for request
			USR_RS_AND_RSI	Used to get reservation from the AMAT instructions sent down based on the batch_id received back into the request looking for the batch_id value

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
		PPPI_DELIVERY_COMPLETE	USR_DELIVERY_TAG	Used to select misc. tag3 from the material_tag to be used for the delivery complete tag. This tag must be configured as a digital state with values X or NULL. The application delivery will change the NULL to blank for SAP.
AREAD1	Requests readings for the plant			
		PPPI_DATA_POINT_VALUE	USR_GET_ALIAS_TAG	used for continuous process
		PPPI_DATA_POINT_VALUE	USR_READ1_MONITOR	used for PI-BATCH or PID monitor
APHPAR_1	Sends readings to the plant			Loads the Formula table as well as prepares data to be sent to locations
		PPPI_PARAMETER_NAME	USR_SET_ALIAS_TAG	used for continuous process
APHACT	Requests confirmation of amounts for activities for costing for time events			
		PPPI_ACTIVITY	USR_PHACT_ACTIVITY	used for continuous and batch
		PPPI_DUMMY	USR_DUMMY_MONITOR	used for when the activity value is sent down from SAP and you just want to get the value for date and time based on phase end time. You do not include this characteristic in the instruction, it will be placed there automatically when it detects that only date and time are asked for
		PPPI_STATUS_CONFIRMED	Upr_phact_status_monitor	Users misc tag 1 on common_name after selecting the resource
		PPPI_CONFIRMATION_SHORT_TEXT	Usr_phact_confirmation+_text	Users misc tag 2 on common_name after selecting the resource.
AREAD2	Reads multiple values			
		PPPI_DATA_POINT_NAME	USR_GET_ALIAS_TAG_RANGE	used for continuous process
		PPPI_DATA_POINT_NAME	USR_READ2_MONITOR	used for either PI-BATCH or PID monitor
AQMSMR	Retrieves a value based lab result.			

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
		PPPI_INSPECTION_RESULT	USR_QMSMR1_MONITOR_S1_V1	used for continuous and batch
		PPPI_NUMBER_OF_INSPECTIONS	USR_QMSMR1_MONITOR_NO	used for continuous and batch
		PPPI_STANDARD_DEVIATION	USR_QMSMR1_MONITOR_DEV	used for continuous and batch
		PPPI_INSPECTION_SHORT-TEXT	USR_INSPECTION_SHORT-TEXT	used for continuous and batch
AOPUST	Retrieves user set status .for operation Values must correspond to status profile in SAP			
		PPPI_OPERATION_USER_STATUS	USR_OPERATION_MONITOR_USER	
APHUST	Retrieves user set status .for phases Values must correspond to status profile in SAP			
		PPPI_PHASE_USESR-STATUS	USR_PHASE_MONITOR_USER	
		PPPI_PHASE_USESR-STATUS	USR_PHASE_EQP_MONITOR_USER	
COMM or other name given to message	Sends messages to the plant			
ASRACT	Retrieves the activity of the secondary resources for time events			
		PPPI_CONFIRMATION_SHORT_TEXT	Usr_sract_confirmation_text	Selects the tag from common_name misc tag 2 based on the secondary resource and the std_parameter_id
		PPPI_STATUS_CONFIRMED	Usr_sract_monitor	Selects the tag from common_name misc tag 1 based on the secondary resource and the std_parameter_id
		PPPI_ACTIVITY	Usr_sract_activity	Selects the tag from common_name based on the secondary resource being used and the std_parameter_id, the secondary resource is found in the table sec_resource. The name of the phase and the secondary_resource must be sent down in the instruction

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
ABTCL	Returns characteristic of the batch	PPPI_BATCH_CHAR_VALUE	Usr_batch_char_value	Selects the tag from point_group for the characateristic which will return the value the tag alias is VALUE
ABTCR	Creates a new batch in SAP	PPPI_BATCH	Usr_batch_char_batch	Selects the tag which will hold the batch id if there is one. Returns the assigned batch
		PPPI_BATCH_NEW	Usr-batch_create_ar	Selects the tag which corresponds to the new batch if there is a change in state then a new batch is created
APHCON	Status of the phase for time ticket including activities			
		PPPI_ACTIVITY_1	Usr_activity_1	This will request the value of the activity at a partial or completion status.
		PPPI_ACTIVITY_1_FINISHED	Usr_activity_finished_1	This is a tag which is a digital state which will be X or NULL and then the delivery application is use to change the NULL to a blank or you can stet the tag to a string and use the values of X and “” and the gettag application.
		PPPI_ACTIVITY_1_UNIT	Usr_activity_1_unit	This will select the request part that is finding the answer to the activity and take the engineering unit returned for that tag.
		PPPI_ACTIVITY_2	Usr_activity_2	
		PPPI_ACTIVITY_2_FINISHED	Usr_activity_finished_2	
		PPPI_ACTIVITY_2_UNIT	Usr_activity_3_unit	
		PPPI_ACTIVITY_3	Usr_activity_3	
		PPPI_ACTIVITY_3_FINISHED	Usr_activity_finished_3	
		PPPI_ACTIVITY_3_UNIT	Usr_activity_3_unit	
		PPPI_ACTIVITY_4	Usr_activity_4	
		PPPI_ACTIVITY_4_FINISHED	Usr_activity_finished_4	
		PPPI_ACTIVITY_4_UNIT	Usr_activity_4_unit	
		PPPI_ACTIVITY_5	Usr_activity_5	
		PPPI_ACTIVITY_5_FINISHED	Usr_activity_finished_5	
		PPPI_ACTIVITY_5_UNIT	Usr_activity_5_unit	
		PPPI_ACTIVITY_6	Usr_activity_6	

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
		PPPI_ACTIVITY_6_FINISHED	Usr_activity_finished_6	
		PPPI_ACTIVITY_6_UNIT	Usr_activity_6_unit	
		PPPI_CONFIRMATION_SHORT_TEXT	usr_confirmation_short_text	This will request the confirmation text from a tag at the partial or completion time
		PPPI_POSTING_DATE	usr_post_date	This will request the post date at the partial or completion time
		PPPI_SCRAP_TO_CONFIRM	usr_scrap	This will request the scrap at the partial or completion time
		PPPI_YIELD_TO_CONFIRM	usr_yield_to_confirm	This will request the yield to confirm at the partial or completion time
		PPPI_PHASE_RESOURCE	usr_phase_resource	This will monitor if the recipe has been started in the assigned resource or if a resource has changed. Up to 8 resources can be monitored. All point groups with the same phase name but different resources are selected to be monitored.
		PPPI_STATUS_CONFIRMED	usr_phcon_time_status	This will monitor for a change in status of the phase by checking 2 tags, one will hold the recipe_id and one will hold the status. This tags will be paired by timestamp. Only the status for partial and complete will be returned to SAP
ASRCON	Status of the secondary resource including activities for the time tickets	PPPI_ACTIVITY_1	Usr_activity_1_sec	This will request the value of the activity at a partial or completion status for the secondary resurce.
		PPPI_ACTIVITY_1_FINISHED	Usr_activity_finish_1_sec	This is a tag which is a digital state which will be X or NULL and then the delivery application is use to change the NULL to a blank or you can stet the tag to a string and use the values of X and “” and the gettag application.
		PPPI_ACTIVITY_1_UNIT	Usr_activity_1_unit_sec	This will select the request part that is finding the answer to the activity and take the engineering unit returned for that tag.
		PPPI_ACTIVITY_2	Usr_activity_2_sec	

Instruction	*Purpose*	*Characteristic*	*Methods of Translation*	*Result of Processing*
		PPPI_ACTIVITY_2_FINISHED	Usr_activity_finish_2_sec	
		PPPI_ACTIVITY_2_UNIT	Usr_activity_2_unit_sec	
		PPPI_ACTIVITY_3	Usr_activity_3_sec	
		PPPI_ACTIVITY_3_FINISHED	Usr_activity_finish_3sec	
		PPPI_ACTIVITY_3_UNIT	Usr_activity_3_unit_sec	
		PPPI_ACTIVITY_4	Usr_activity_4_sec	
		PPPI_ACTIVITY_4_FINISHED	Usr_activity_finish_4_sec	
		PPPI_ACTIVITY_4_UNIT	Usr_activity_4_unit_sec	
		PPPI_ACTIVITY_5	Usr_activity_5_sec	
		PPPI_ACTIVITY_5_FINISHED	Usr_activity_finish_5_sec	
		PPPI_ACTIVITY_5_UNIT	Usr_activity_5_unit_sec	
		PPPI_ACTIVITY_6	Usr_activity_6_sec	
		PPPI_ACTIVITY_6_FINISHED	Usr_activity_finish_6_sec	
		PPPI_ACTIVITY_6_UNIT	Usr_activity_6_unit_sec	
		PPPI_CONFIRMATION_SHORT_TEXT	Usr_confirmation_text_sec	
		PPPI_POSTING_DATE	Usr_post_date_sec	
		PPPI_STATUS_CONFIRMED	Usr_time_status_sec	
ASRST	Status of the secondary resource for a phase with time events	PPPI_SECONDARY_RESOURCE_STATUS	Usr_srst_monitor Usr_set_status (for continuous)	This will monitor the status of the secondary resource for a change in state.
		PPPI_CONFIRMATION_SHORT_TEXT	Usr_srst_confirmation_text	This will select the tag for the confirmation from the point group of the secondary resource at the time of the chane in state of the resource
		PPPI_REASON_FOR_VARIANCE	Usr_srst_reason_for_variance	This will select the tag for the reason from the point group of the secondary resource at the time of the chane in state of the resource

Translation Methods and Execution Process

- Translation Methods

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_GET_RECIPE_STATUS	Gets the recipe start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE	Recipe, Location	Puts into action_results the recipe_id and endtime of recipe
USR_GET_OPERATION_STATUS	Gets the operation_id start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE	Recipe, Location, Operation	Puts into action_results the recipe_id, operation_id and endtime of recipe
USR_GET_PHASE_STATUS	Gets the phase_id, start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE	Recipe, Location, Phase	Puts into action_results the recipe_id and Phase_id and the start and endtimes.
USR_GET_ LOCATION	Set the equipment location for a given material	Material, Equipment tables There must be a single piece of equipment for the material	Returns equipment_id in Action Results
USR_GET_ALIAS_TAG_RANGE	Gets the tag and based upon the SAP/R3 characteristic PPPI_DATA_POINT_NAME and selects the range as start and end time of recipe and sets no of values to 10	Common_name	Sets the tag_id and start and endtime and the no of values to be retrieved in the time range.
USR_BATCHID_TAG	Gets the batch tag based upon the material id, with the start and end time of the phase	Material_tag	Sets the batch tag and the start and end time to be searched for values.
USR_GET_ALIAS_TAG	Gets the tag and based upon the SAP/R3 characteristic PPPI_DATA_POINT_NAME	Common_name	Sets the tag_id and recipe endtimestamp for request
USR_PHACT_ACTIVITY	Gets the tag based upon the SAP/R3 characteristic PPPI_STD_VALUE_PARAMETER_ID	Common_name	Sets the tag_id and recipe endtimestamp for request
USR_QMSMR1_MONITOR_S1_V1	Gets the tagname for the quality inspection and lot tag and lot number point based on the characteristic PPPI_INSPECTION_RESULT	Point_Group and Point_group_members	Sets the tag_id and the start andendtimestamp for the request. The lot tag and the lot number
USR_SET_ALIAS_TAG	Selects the tag_id for value to be sent to PI including the tag for the min and max value. Uses SAP/R3 PPPI_DATA_POINT_NAME	Common_name	Selects 3 tags and sets the timestamp to be the beginning of the recipe
USR_PHASE_MONITOR	Selects the points to monitor in PI for the phases of the recipe. Used is PI is going to be the source of information of status on phase	Point group, Point_group_members, phase, recipe	Selects point name for the phase status tag in PI to monitor and sets the timestamp to be the beginning of the recipe. This routine is triggered after the recipe has received notice that it has started.
USR_PHASE_EQP_MONITOR	Selects the points to monitor in PI for the phases of the recipe. The points are selected based on phase name and resource. Thus yu can have the same phase name used in multiple resources. Used is PI is going to be the source of information of status on phase	Point group, Point_group_members, phase, recipe	Selects point name for the phase status tag in PI to monitor and sets the timestamp to be the beginning of the recipe. This routine is triggered after the recipe has received notice that it has started. The point group is selected based on the phase name and resource of the point group.
USR_RECIPE_MONITOR	Used in both the batch execution method and PI reading method to watch for the status change in the batch.	CHRE, recipe	Sets the starttime to monitor to be the time when the recipe was read from SAP/R3.

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_READ1_MONITOR	Selects the tags to monitor for the read instruction after the status of the phase for the read has been set to complete. Used in either the PI or batch execution method of executing a recipe	Common_name, phase	Sets the tag and the timestamp to be the end time of the phase
USR_READ2_MONITOR	Selects the tags to monitor for the read instruction after the status of the phase for the read has been set to complete. Used in either the PI or batch execution method of executing a batch.	Common_name, phase	Sets the tag, and start and end time of the phase along with no of points to be read.
USR_QMSMR1_MONITOR_DESC	Selects the tag to be read for short text	Point_group and point_group_members	Sets the tag and the start and end timestamp for the result with the lot no and the lot tag
USR_YIELD_TO_CONFIRM	Selects the tag to be read for yield to confirm	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_YIELD_TO_CONFIRM_PARTAIL	Selects the tag to be read for yield to confirm	Point_group and point_group_members	Sets the tag and the start and end timestamp for partial status in continuous
USR_REASON_FOR_VARIANCEO	Selects the tag to be read for reason for variance	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_CONFIRMATION_SHORT_TEXT	Selects the tag to be read for confirmation short text	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_OPERATION_MONITOR_USER	Selects the tag to be read for operation user status	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_PHASE_MONITOR_USER	Selects the tag to be read for phase user status	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_PHASE_EQP_MONITOR_USER	Selects the tag to be read for phase user status based on resource	Point_group and point_group_members	Sets the tag and the start and end timestamp
USR_QMSMR1_MONITOR_DEV	Selects the tag to be read for deviation	Point_group and point_group_members	Sets the tag and the start and end timestamp for the result with the lot no and the lot tag
USR_QMSMR1_MONITOR_NO	Selects the tag to be read for number	Point_group and point_group_members	Sets the tag and the start and endtimestamp for the result with the lot no and the lot tag
USR_BATCH_FLOW_TAG	Sets the tagid for the material consumption or production with the start time and the end time based upon the phase the material is being consumed or produced in.	material_tag, phase	sets the tag_id and the start and endtime for the material
USR_OPERATION_MONITOR	Monitors if all the phases in the operation have been completed and then sets the operation status to be the completion time of the last phase	Recipe, Phase	Sets the operation and the timestamp for the start of the recipe to monitor that all phases in the operation are complete
USR_PHASE_ALIAS_MONITOR	Selects the alias name of the phase which is being used in the batch execution system . These are selected after it is determined that the recipe has been started by the batch execution system.	Phase, recipe	Selects the phase alias and the start time of the recipe to start monitoring for the start of the phase.
USR_DUMMY_MONITOR	Used for activity when all the information is sent down except the date and time, will set up a request for the phase status time	Phase, phase_status_details	Sets a dummy value at the endtime of the phase which will then be assigned to the time of the activity
USR_RESERVATION	Selects the reservation tag from the material tag table	Material_tag	Sets a tag to obtain the reservation from
USR_RESERVATION_ITEM	Selects the resesrvation_item tag from the material tag table	Material_tag	Sets a tag to obtain the reservation item from

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_DELIVERY_TAG	Selects the delivery tag, misc tag 3 from the material tag table miscellaneous item 3.	Material_tag	Sets a tag to obtain the delivery complete flag from
USR_PHASE_RESOURCE	Selects all phases with the same name in the same plant and will take the batch_id tag for these phases	Point Group	Selects up to 8 phase resources with the same name to watch to see which one has been assigned to the recipe
USR_RS_AND_RSI	Used to get reservation from the AMAT instructions sent down based on the batch_id received back into the request looking for the batch_id value	Material_List	Selects the request_id for the batch assigned to this material and sets whether a reservation or reservation_item should be retrieved from the material_list
USR_MATERIAL_FROM_BATCH	Used when you can not guarantee that the batch_id tag values and the material tag values will have the same timestamp. This will look for a batch_id value that is retrieved and then setup to get the material tag value at the timestamp of the batch_id and take the value at that time	Material_tag, Action_result_values	Selects the request_id for the batch and selects the timestamp from the results of the batch_id which have been retrieved. It will then setup a request based on the batch_id's timestamp
USR_PHACT_STATUS_MONITOR	Sets the status of the activity in the PHACT instruction	Common_name, phase	Selects misc tag 1 from common name and sets the timestamp to be that of the change in phase status
USR_PHACT_CONFIRMATION_TEXT	Sets the confirmation text	Common_name, phase	Select misc tag 2 from common name and sets the timestamp to be that of the change in phase status
USR_SRACT_ACTIVITY	Sets the tag for locating the activity of the secondary resource as required in the PPPI_STD_PARAMETER_ID	Common_name, sec_resource	Selects the tag from common_name based on the secondary resource and the STD_PARAMETER_ID at the time of the secondary resource status
USR_SRACT_CONFIRMATION_TEXT	Sets the confirmation text	Common_name, sec_resource	Select misc tag 2 from common name and sets the timestamp to be that of the change in secondary resource status
USR_SRACT_MONITOR	Sets the status of the activity in the SRACT instruction	Common_name, sec_resource	Selects misc tag 1 from common name and sets the timestamp to be that of the change in secondary resource status

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_BATCH_CHAR_VALUE	Sets the value of the characteristic for the batch, The value of the characteristic and the batch_id which is in the tag that has the alias BATCH are matched by having the same timestamp.	Point_group, point_group_members and point_group_groups	For a given material it will find the point group with type MAT_CHAR and then for the point_group_groups which are members of this group it will select the point group for the give characteristic where the group typ is CHAR and the group description is the name of the characteristic. It will select the tag alias VALUE and find the result at the time of the batch. It searches from the start of the phase to the partial confirmation or end of phase.
USR_BATCH_CREATE_AR	Determines if a new batch should be created. A tag that is a digital state will signal if a batch should be created. If the value is 00001 a new batch is to be created. Sets up a request which includes the tag with alias BATCH to return the batch_id and the tag that holds the digital value. If the BATCH tag does not hold a value at the time then the PPPI_BATCH_NEW will be assigned to blank which means that SAP will create the batch.	Point_group, point_group_members, and point_group_groups	The tag with the alias PPPI_BATCH_NEW is selected to monitor between status changes of the phase.
USR_BATCH_CHAR_BATCH	If batch characteristics are to be returned this will return the batch_id. The characteristic value and the batch_id will be matched by timestamp.	Point_group, point_group_members and point_group_groups	The tag with alias BATCH is selected at the timestamp of the batch creation tag.
USR_ACTIVITY_1 USR_ACTIVITY_2 USR_ACTIVITY_3 USR_ACTIVITY_4 USR_ACTIVITY_5 USR_ACTIVITY_6	This will select the corresponding alias from point_group_members to retrieve the activity value. The point group that is used is PI-BATCH	Point_group, Point_group_members	The tag will be the value of the activity
USR_ACTIVITY_FINISHED_1 USR_ACTIVITY_FINISHED_2 USR_ACTIVITY_FINISHED_3 USR_ACTIVITY_FINISHED_4 USR_ACTIVITY_FINISHED_5 USR_ACTIVITY_FINISHED_6	This will select the tag from the point_group_members with the alias ACTIVIT Y_FINISHED_n, the group is the PI-BATCH	Point_group, Point_group_members	The tag can be a string tag with a X or “” and the application get tag can be used or the tag can be a digital state with values X or NULL and the delivery application will be used to change the NULL to blank for the return to SAP
USR_ACTIVITY_1_UNIT USR_ACTIVITY_2_UNIT USR_ACTIVITY_3_UNIT USR_ACTIVITY_4_UNIT USR_ACTIVITY_5_UNIT USR_ACTIVITY_6_UNIT	This will select the request_part corresponding to the activity for the phase and will setup to return the engineering unit of that request part with the application usr_eng_unit. The point group is of the type PI-BATCH	Point_group, Point_group_members	This will use the application usr_eng_unit to return the engineering unit for the activity. The reply method must be set to UNIT
USR_POST_DATE	This will select the tag from the point_group_members with the alias POST_DATE the group is the PI_BATCH type to satisfy the PHCON posting date characterisitc	Point_group, Point_group_members	This will return from the tag the post date
USR_SCRAP	This will select the tag from the point_group_members with the alias SCRAP the group is the PI_BATCH type to satisfy the PHCON scrap characteristic	Point_group, Point_group_members	This will return from the tag the scrap quantity

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_PHCON_TIME_STATUS	This will select the tag from the point_group_members with the alias SAP the group is the PI_BATCH type	Point_group, Point_group_members	This will monitor for a change of status if the value is 00004 or 00002 then it will be returned to SAP and all the other values will be found at this time
USR_SRST_MONITOR	This will select the tag SAP that will hold the status of the secondary resource and the BATCH_ID tag which will hold the recipe which is using the secondary resource. This is setup for the application srstatus.exe	Point_group, Point_group_members Sec_resource	The secondary resource can not change from what is planned in the recipe.
USR_SRST_CONFIRMATION_TEXT	This will select the tag for the confirmation text of the secondary resource at the change in state of the resource	Point_group, Point_group_members Sec_resource	The tag will be the confirmation text
USR_SRST_REASON_FOR_VARIANCE	This will select the tag for the reason of the secondary resource at the change in state of the resource	Point_group, Point_group_members Sec_resource	The tag will be the reason value
USR_ACTIVITY_1_SEC USR_ACTIVITY_2_SEC USR_ACTIVITY_3_SEC USR_ACTIVITY_4_SEC USR_ACTIVITY_5_SEC USR_ACTIVITY_6_SEC	This will select the tag from the point_group_members with the alias ACTIVIT Y_n, the group is the SEC_RES type	Point_group, Point_group_members Sec_resource	The tag will be the value of the activity
USR_ACTIVITY_FINISH_1_SEC USR_ACTIVITY_FINISH_2_SEC USR_ACTIVITY_FINISH_3_SEC USR_ACTIVITY_FINISH_4_SEC USR_ACTIVITY_FINISH_5_SEC USR_ACTIVITY_FINISH_6_SEC	This will select the tag from the point_group_members with the alias ACTIVIT Y_FINISHED_n, the group is the SEC_RES type	Point_group, Point_group_members Sec_resource	The tag can be a string tag with a X or “” and the application get tag can be used or the tag can be a digital state with values X or NULL and the delivery application will be used to change the NULL to blank for the return to SAP
USR_ACTIVITY_1_UNIT_SEC USR_ACTIVITY_2_UNIT_SEC USR_ACTIVITY_3_UNIT_SEC USR_ACTIVITY_4_UNIT_SEC USR_ACTIVITY_5_UNIT_SEC USR_ACTIVITY_6_UNIT_SEC	This will select the request_part corresponding to the activity for the secondary resource and will setup to return the engineering unit of that request part with the application usr_eng_unit	Point_group, Point_group_members Sec_resource	This will use the application usr_eng_unit to return the engineering unit for the activity. The reply method must be set to UNIT
USR_CONFIRMATION_TEXT_SEC	This will select the tag from the point_group_members with the alias SHORT_TEXT the group is the SEC_RES type	Point_group, Point_group_members Sec_resource	This will return from the tag the confirmation text
USR_POST_DATE_SEC	This will select the tag from the point_group_members with the alias POSI_DATE the group is the SEC_RES type	Point_group, Point_group_members Sec_resource	This will return from the tag the post date
USR_TIME_STATUS_SEC	This will select the tag from the point_group_members with the alias SAP the group is the SEC_RES type	Point_group, Point_group_members Sec_resource	This tag will be monitored for change in status, only the values of 00004 and 00002 are returned. When there is a value all the other data is requested.

Translation Method	*Purpose*	*Data Tables Used*	*Result of Processing*
USR_OPERATION_MONITOR_NEW	Monitors if all the phases in the operation have been completed and then sets the operation status to be the completion time of the last phase, also sets the tags required to get the change in active unit for the operations if they have been set up as point groups and PI-BATCH units	Recipe, Phase Point_group Point_group_members	Sets the operation and the timestamp for the start of the recipe to monitor that all phases in the operation are complete
USR_MISC1_TAG USR_MISC2_TAG USR_MISC3_TAG USR_MISC4_TAG	These can be used for any characteristics on the ACONS or APROD that are not handled explicity by another method. They will pick up the miscellaneous tag specified for the material	Material_tag	Sets the tag and start and end time of the phase.
USR_STORAGE	Sets the storage location to be the location passed in the AMAT or AMATP01 instructions	Material_list	Sets the storage location
Usr_batchid_tag_kk (named for Kellogg, Keebler)	Used to setup a request for batch_id of a material in order to have multiple recipes running on the same resource at the same time. It results in the following settings field 1 = batch tag on material_tag, filed 2 finish time, field 3 start time, field 4 recipe tag from misc tag1, filed 5 recipe no, filed 6 is material quantiy tag and field 7 material no.	Mateial_tag	This is used with the getprop application to search for the recipe and match with the material to get the batch when there are multiple recipes running on the same phase at the same time
Usr_batch_flow_tag_kk	When multiple recipes running on the same phase at the same time to match recipe and quantiy for a material. Filed 1 quantity tag, filed 2 finish time, field 3 start time, field 4 recipe tag from misc tag1, field 5 recipe no, field 6 material quantiy tag, field 7 material no.	Material_tag	This is used with the getmatqty application to search for the recipe and match with the material to get the quantity when there are multiple recipes running on the same phase at the same time

- Application Programs for execution

Some of the PI applications have been consolidated into one PI program called pimod. This consolidation reduces the number of logons for PI. When you are setting up the translation method you still use the individual application although the executable that is called will be pimod.exe. The programs included in pimod are given in the table pi_function. The functions included with this release are shown below. The function getsummary allows a parameter value that can be entered in this table in the parameter column. For the getsummary and summarywait calculations are passed the percent good, the default is assumed to be 100 % if no entry is made. The getdiff and getdiffwait will take the absolute value between the values.

Func_name
GETSNAPSHOT
GETTAGRANGE
GETSUMMARY
GETDSUM
PUTSNAP
MULTIVAL

Func_name
GETDIFF
GETDIFFWAIT
GETTAGRANGEWAIT
MULTIVALWAIT
GETTAGWAIT
GETINPVALUE
GETINPWAIT
SUMMARYWAIT
DSUMWAIT
GETTAGJ
GETTAGJI
DELIVERY

Application Programs	*Data Required*	*Result of Processing*
Getsnapshot (snap.exe)	tag_id	Returns PI snapshot and eng unit for tag_id
Putsnap(putsnap.exe)	tag_id	Sets PI snapshot for tag_value, min_tag_id and max_tag_id
Gettag(value.exe) or gettagwait(valuew) or getinpvalue(valuei) or getinpwait(gviw)	tag_id, timestamp	Retrieves PI value and eng unit at the given time
Getsummary(summary.exe) or summarywait(summaryw)	PI config tag must be set with totalcode = 0 and convers for the correct time conversion of the point	Retrieves totalized value from PI for the tag specified between the start and end times.
Summaryplus	Tag_id, start time, end time, Tag_id	Two tags are supplied the first tag is totalized and the second tag is added to the total
Putvalue(putvalue.exe)	tag_id, timestamp	Sets PI value for tag_value, min_tag_id and max_tag_id at the given time
Usr_set_location	Equipment_group, Equipment_group_members, Equipment	Sets the Equipment location value based on the equipment tables
Usr_set_status	recipe_id, (operation_id or Phase_id) start and end time	Sets the recipe, operation and phase status. For the Phase both a start and end time status are set.

Application Programs	*Data Required*	*Result of Processing*
Usr_set_status_partial	recipe_id, (operation_id or Phase_id) start and end time	Sets the phase status. For the Phase both a start and end time status are set and partial status are handled for continuous recipe
Gettagrange(interpv.exe) or gettagrangewait(interpvw)	tag_id, starttime,endtime, no of values	Gets a range of values for a specified tag from the start to the endtime. No of values returned is set OSI_NO_VALUES otherwise the default of 10 is returned
Getdsum(getdsum.exe) or dsumwait (getdsumw)	tag_id, starttime, endtime	Sums discrete values from the start time to the ending time and returns the total. Useful for scale weight totalizing.
Openbatch(opnbatch.exe)	recipe , selects the materials and the formula values.	Puts a recipe on the batch list for openbatch
Control_monitor(stsctrl.exe)	tagid	Monitors PI for status change of recipe tags used if this is a PI-Batch plant
Phase_monitor(phstctrl.exe) A cmdline parameter C to accept repeat count is incorporated. By default, phsctrl function will be executed once. If the paremter is passed then it will validate the parameter and run phsctrl that many times specified by “repeat count”. The max repeat count is 25.	Tagid	Monitors the status of the phase tags if this is a PI-Batch plant
Usr_open_batch_recipe(obrecipe.exe)	recipe	Monitors the status of the recipe if this is a PID plant that data is to be taken from Openbatch Batchhis
Usr_open_batch_phase(obphase.exe)	phase alias, recipe	Monitors the status of the phase if this is a PID plant that data is to be taken from Openbatch Batchhis
Usr_operation_status	recipe, operation_id	Monitors the operation status for either PI-Batch plant or Openbatch from Batchhis plant
Multival(multiv.exe) or multivalwait(multivw)	Tag_id, starttime, endtime	Returns sets of values at times from the start to the end. Use if multiple consumptions or productions from different batches during the time and you want to report back each individual result.
Getdiff(getdiff.exe) or getdiffwait (getdiffw)	Tag_id, starttime, endtime	Takes the value of the start time and subtracts the value at the end time. Used in tank rundown.

Application Programs	*Data Required*	*Result of Processing*
Getqmval(qm.exe)	Tag_id, starttime, endtime, Lot tag, Lot number	Finds the timestamp for the lot number and then finds value with that timestamp.
Vbatchr.exe	Recipe	Monitors the status of the recipe as output from iBatch by looking in the SQL archive
Vbatchp.exe	Phase alias, recipe	Monitors the status of the phase from iBatch as put in the SQL archive
batchvb	Recipe, selects the materials and formula values	Puts a recipe on the batchlist for iBatch
resource	Batch_id tags for the common phase name and the recipe_id	Sets the phase to the one where the recipe is actually being processed
Usr_rs_and_rsi-app	Batch_id	Selects for the material_list the reservation and reservation_item based on the batch_id returned
Jvalue,jvaluei	Batch_id	Sets the value for material after a batch_id has been returned, either the last value or an interpolated value will be returned
Delivery.exe		Reads a digital state tag which has a value of X or NULL if the value is NULL changes this to blank for SAP
Newbatch (newbat.exe) newbatch_wait (newbatw.exe)	Batch_id tag, start time, end time and tag batch digital state	Checks the digital state tag. If the value is 00001 then it reads the batch_id tag. If there is no value in the batch_id tag then assigns a blank. If there is a value returns the batch_id.
Usr_eng_unit	Request part which will be used for the engineering unit	Selects the engineering unit found for the request part assigned.
Sec_status (Srstatus.exe)	Tag for the secondary status and tag for the batch_id corresponding to the secondary status	Returns the status of the secondary resource as it is used for the given recipe. It will update the timestamps of the other items required, confirmation and reason
Usr_operation_status_new	Point group for the operation	Sets the operation status and also sets the tags in action_send so that a change of state can be detected by the campaign manager.
Usr_storage_matlist_app	Request_id, material_id, resource_id and the start and end time	Sets the storage location from the material list table.

Application Programs	*Data Required*	*Result of Processing*
Getmatqty (getmatqty.exe)	Field 1 quantity tag, field 2 finish time, field 3 start time, field 4 recipe tag from misc tag 1, field 5 recipe no, field 6 material quantity tag, field 7 material no	Finds the material batch for materials for the given recipe
Max	Tag_id, start time, end time	Returns the maximum value in the time range.
Putoption	Tag_id, timestamp	Sets value in PI but gives option to replace the existing value
Getprop (getprop.exe) A cmdline parameter C to accept repeat count is incorporated. By default, Getprop function will be executed once. If the paremeter is passed then it will validate the parameter and run Getprop that many times specified by “repeat count”. The max repeat count is 25.	Field batchtag from material_tag, field2 finish time, field 3 start time, field 4 recipe tag from misc tag1, filed 5 recipe no , filed 6 material quantity tag, field 7 material no	Finds the material quantity for materials for the given recipe

Translation Procedures that are valid for each application are given in the following table.

Application Programs	*Translation Procedures*

Application Programs	*Translation Procedures*
Getsnapshot (snap.exe)	USR_READ2_MONITOR(starttime) USR_GET_ALIAS_TAG_RANGE(starttime) USR_READ1_MONITOR USR_PHACT_ACTIVITY USR_GET_ALIAS_TAG USR_BATCHID-TAG USR_BATCH_FLOW_TAG USR_YIELD_TO_CONFIRM USR_REASON_FOR_VARIANCE USR_CONFIRMATION-SHORT_TEXT USR_OPERATION_MONITOR_USER USR_PHASE_MONITOR_USER USR_PHASE_EQP_MONITOR_USER USR_ACTIVITY_N USR_ACTIVITY_N_SEC USR_ACTIVITY_FINISH_N_SEC USR_ACTIVITY_FINISHED_N USR_BATCH_CHAR_BATCH USR_BATCH_CHAR_VALUE USR_CONFIRMATION_TEXT_SEC USR_PHACT_CONFIRMATION_TEXT USR_PHACT_STATUS_MONITOR USR_POST_DATE USR_POST_DATE_SEC USR_SCRAP USR_SRACT_ACTIVITY USR_SRACT_CONFIRMATION_TEXT USR_SRACT_MONIT USR_SRST_CONFIRMATION_TEXT USR_SRST_REASON_FOR_VARIANCE
Putsnap(putsnap.exe)	USR_SET_ALIAS_TAG

Application Programs	*Translation Procedures*
Gettag(value.exe) or gettagwait (valuew) or getinpvalue (valuei) or getinpwait (gviw)	USR_READ2_MONITOR(starttime) USR_GET_ALIAS_TAG_RANGE(starttime) USR_READ1_MONITOR USR_PHACT_ACTIVITY USR_GET_ALIAS_TAG USR_BATCHID_TAG USR_BATCH_FLOW_TAG USR_YIELD_TO_CONFIRM USR_REASON_FOR_VARIANCE USR_CONFIRMATION-SHORT_TEXT USR_OPERATION_MONITOR_USER USR_PHASE_MONIOTR_USER USR_PHASE_EQP_MONITOR_USER USR_ACTIVITY_N USR_ACTIVITY_N_SEC USR_ACTIVITY_FINISH_N_SEC USR_ACTIVITY_FINISHED_N USR_BATCH_CHAR_BATCH USR_BATCH_CHAR_VALUE USR_CONFIRMATION_TEXT_SEC USR_PHACT_CONFIRMATION_TEXT USR_PHACT_STATUS_MONITOR USR_POST_DATE USR_POST_DATE_SEC USR_SCRAP USR_SRACT_ACTIVITY USR_SRACT_CONFIRMATION_TEXT USR_SRACT_MONIT USR_SRST_CONFIRMATION_TEXT USR_SRST_REASON_FOR_VARIANCE
Getsummary(summary.exe) or summarywait (summary)	USR_BATCH_FLOW_TAG
Summaryplus	USR_BATCH_FLOW_PLUS
Putvalue(putvalue.exe)	USR_SET_ALIAS_TAG
Usr_set_location	USR_GET_ LOCATION
Usr_set_status	USR_GET_RECIPE_STATUS, USR_GET_OPERATION_STATUS USR_GET_PHASE_STATUS
Usr_set_status_partial	USR_GET_PHASE_STATUS
Gettagrange(interpv.exe) or gettagrangewait (interpvw)	USR_GET_ALIAS_TAG_RANGE USR_READ2_MONITOR
Getdsum(getdsum.exe) or dsumwait(getdsumw)	USR_BATCH_FLOW_TAG
Control_monitor(stsctrl.exe)	USR_RECIPE_MONITOR
Phase_monitor(phstctrl.exe)	USR_PHASE_MONITOR USR_PHASE_EQP_MONITOR USR_PHCON_TIME_STATUS
Usr_open_batch_recipe(obrecipe.exe)	USR_RECIPE_MONITOR

Application Programs	*Translation Procedures*
Usr_open_batch_phase(obphase.exe)	USR_PHASE_ALIAS_MONITOR
Usr_operation_status	USR_OPERATION_MONITOR
Multival(multv.exe) or multivalwait (multivw)	USR_BATCHID_TAG USR_BATCH_FLOW_TAG USR_OPERATION_MONITOR_USER USR_PHASE_MONIOTR_USER USR_PHASE_EQP_MONITOR_USER
Getdiff(getdiff.exe) or getdiffwait (getdiffw)	USR_GET_ALIAS_TAG_RANGE(start-end) USR_READ2_MONITOR(start-end) USR_BATCH_FLOW_TAG(start-end)
Getqmval(qm.exe)	USR_QMSMR1_S1_V1(start-end) USR_QMSMR1_DEV(start-end) USR_QMSMR1_DESC(start-end) USR_QMSMR1_NO(start-end)
Vbatchr.exe	USR_RECIPE_MONITOR
Vbatchp.exe	USR_PHASE_ALIAS_MONITOR
Phaseres.exe (resource)	USR_PHASE_RESOURCE
Gettagj(jvalue.exe) or gettagji(jvaluei)	USR_MATERIAL_FROM_BATCH
Usr_rs_and_rsi_app	USR_RS_AND_RSI
Delivery	USR_ACTIVITY_FINISH_N_SEC USR_ACTIVITY_FINISHED_N USR_DELIVERY_TAG
Newbatch or newbatch_wait	USR_BATCH_CREATE_AR
Usr_eng_unit	USR_ACTIVITY_N_UNIT, USR_ACTIVITY_N_UNIT_SEC
Sec_status	USR_SRST_MONITOR, USR_TIME_STATUS_SEC
Usr_operation_status_new	USR_OPERATION_MONITOR_NEW
max(max.exe) or maxwait (max.exe) A mdline parameter –P [for percentgood] is added for max application, it will use this value, otherwise the default check is made with 100%.	USR_BATCH_FLOW_TAG
Getprop.exe	USR_BATCHID_TAG_KK
Getmatqty.exe	USR_BATCH_FLOW_TAG_KK
Usr_storage_matlist_app	USR_STORAGE

The PI routines make a distinction between wait and no wait. This means that if you choose the wait option a value with a timestamp greater than or equal to the requested time must appear in the snapshot for the point otherwise it will not return a value on that request. You must be aware that values can appear in the snapshot but not yet moved to the archive for PI, therefore if requests are made to the archive they must check the snapshot. The mode used in the PI request has been noted in the following table.

Application	*Wait/No Wait*	*Exe Name*	*Program Name*	*Pi for Numbers*	*PI for digital*	Pi for strings
gettagrange no wait	N	interpv	Gettagrange	piar_interpvalues		piar_getarcvaluesx –comp
get tag at time no wait	N	value	gettag	piar_value mode(1)	piar_value mode(1) pisn_getsnapshotx	piar_getarcvaluex mode (3)
get flow total no wait	N	summary	Getsummary	piar_summary		
get total discrete value in time range no wait	N	getdsum	Getdsum	piar_compvalues		
multiple pdat	N	multiv	Multival	piar_compvalues	piar_compvalues	piar_getarcvaluesx –comp
difference	N	getdiff	Getdiff	piar_compvalues		
difference value between start and end wait	W	getdiffw	Getdiffwait	piar_compvalues , pisn_getsnapshotx		
gettagrange between start and end wait	W	interpvw	Gettagrangewait	piar_interpvalues , pisn_getsnapshotx		piar_getarcvlaluesx comp pisn_getsnapshotx
multiple values pdate the start and end wait	W	multivw	Multivalwait	piar_compvalues, pisn_getsnapshotx	piar_compvalues, pisn_getsnapshotx	piar_getvaluesx – comp pisn_getsnapshotx
get tag at exact time wait	W	valuew	Gettagwait	piar_value mode(1) pisn_getsnapshotx	piar_value mode(1) pisn_getsnapshotx	piar_getarcvaluex mode(3) pisn_getsnapshotx
get interpolated tag	N	valuei	Getinpvalue	piar_value mode(3)	piar_ pdat mode(3)	piar_getarcvlaluex mode (3)
get interpoloate tag value at exact time wait snapshot time > endtime	W	gviw	Getinpwait	piar_value mode(3) pisn_getsnapshotx	piar_value mode(3) pisn_getsnapshotx	piar_getarcvaluex Mode(3) pisn_getsnapshotx
get flow total with wait this routine could wait for the time in snapshot to be greater than the time (must be greater not equal to assume data moved to archive)	W	summaryw	Summarywait	piar_summary pisn_getsnaphotx		

Application	*Wait/No Wait*	*Exe Name*	*Program Name*	*Pi for Numbers*	*PI for digital*	Pi for strings
discrete total with wait this routine could wait for the time in snapshot to greater than the time if the time is == in the snapshot then take this as the value at the endtime	W	getdsumw	Dsumwait	piar_compvalues pisn_getsnaphotx		
QM result. It finds the lot number and selects the timestamp and then takes the value at that time. If the lot is still in the snapshot it will not find it until it moves to archive	W	getqm	Qm	piar_getarcvaluex mode(3)	piar_getarcvaluex mode(3)	Piar_getarcvaluesx comp piar_getarcvaluex mode(3)
gettagji	N	jvaluei	gettagji	piar_value mode(3)	piar_ pdat mode(3)	Pisar_getarcvlaluex mode (3)
get tagj at time no wait	N	jvalue	gettagj	piar_value mode(1)	piar_value mode(1) pisn_getsnapshotx	Piar_getarcvaluex mode (3)
Delivery	N	Pimod(delivery)	delivery		piar_value Mode(1)	
Newbatch	N	Newbat	Newbatch		Piar_compvalues Mode(1) Piar_getarcvaluex	
Newbatch_wait	W	newbatw	Newbatch_wait		Piar_compvalues Mode(1) Piar_getarcvaluex	
Max	N	Max	Max	Piar_summary Code = 6		
Maxwait	W	maxwait	maxwait	Piar_summary Code = 6		
Getprop	N	Getprop				Piar_getarcvaluesx
Getmatqty	N	Getmatqty				Piar_getarcvaluesx

Translation Procedure	*Data fields Returned*						
usr_batchid_tag	batch_tag	endtime	Starttime				

Translation Procedure	*Data fields Returned*						
usr_batch_flow_tag	material_tag	endtime	Starttime				
Usr_batch_flow_plus	Material_tag	Endtime	Starttime			Plus tag_id	
usr_phact_activity	common_name_tag for activity	endtime	Starttime				
usr_get_alias_tag	common_name_tag	endtime					
usr_get_alias_tag_range	common_name tag	starttime	Endtime	no_values			
usr_yield_to_confirm	Yield_to_confirm tag	endtime	Starttime				
usr_yield_to_confirm_partial	Yield_to_confirm tag	endtime	Starttime				
usr_reason_for_variance	Reason for variance tag	endtime	Starttime				
usr_confirmation-short-text	Confirmation short text tag	endtime	Starttime				
usr_get_location	equipment_id	endtime					
usr_get_operation_status	recipe_id	operation_id	Endtime				
usr_get_phase_status	recipe_id	phase_id	Endtime				
usr_get_recipe_status	recipe_id		Endtime				
usr_operation_monitor	recipe_id	operation_id	Timestamp				
usr_phact_monitor	common_name_tag for activity	endtime					
usr_phase_alias_monitor	recipe_id	phase_alias_name					
usr_phase_monitor	recipe_id	phase_id	phase_status_tag	recipe_tag	timestamp		
usr_qmsmr1_monitor_desc	Qm short text tag	endtime	Starttime	Lot tag	Lot Number		
usr_qmsmr1_monitor_dev	Qm deviation tag	endtime	Starttime	Lot tag	Lot Number		
usr_qmsmr1_monitor_no	Qm number tag	endtime	Starttime	Lot tag	Lot Number		
usr_qmsmr1_monitor_s1_v1	Qm result tag	endtime	Starttime	Lot tag	Lot Number		
usr_read1_monitor	common_name tag	endtime					
usr_read2_monitor	common_name tag	starttime	Endtime	no_values			
usr_recipe_monitor	recipe_id	recipe_status_tag	recipe_tag	timestamp			
usr_set_alias_tag	common_name	starttime	Value	min_tag	value		
usr_phase_monitor_user	Phase user status tag	endtime	Starttime				
usr_phase_eqp_monitor_user	Phase user status tag	endtime	Starttime				
usr_operation_monitor_user	Operation user status tag	endtime	Starttime			max_tag	value
usr_dummy_monitor	recipe_id	phase_id	phase_status_tag	recipe_tag	timestamp		
usr_reservation	Reservation Tag from material_tag	endtime	Starttime				
usr_resesrvation_item	Reservation_item Tag from material_tag	endtime	Starttime				

Translation Procedure	*Data fields Returned*						
usr_delivery_tag	Tag, misc tag from material_tag	endtime	Starttime				
usr_phase_resource	Starttime	Batch_id tag for phase1	Batch_id tag for phase2	Batch_id tag for phase etc.			
Usr_rs_and_rsi	Request_part for corresponding batch_id	endtime	Starttime	Material	phase		
Usr_material_from_batch	Request_part_id for corresponding batch_id	Endtime					
Usr_activity_n	Tag from point group for PI-BATCH with alias ACTIVITY	endtime	Starttime				
Usr_activity_n_sec	Tag from point group for SEC_RES with alias ACTIVITY	endtime	Starttime				
Usr_activity_n_unit	Request_part for characteristic PPPI_ACTIVITY PI-BATCH with alias ACTIVITY	endtime	Starttime				
Usr_activity_finish_n_sec	Tag from SEC_RES with alias ACTIVITY_FINISH_n	endtime	Starttime				
Usr_activity_finished_n	Tag from PI_BATCH with alias ACTIVITY_FINISH_n	endtime	Starttime				
Usr_activity_1_unit_sec	Request_part for characteristic PPPI_ACTIVITY SEC_RES with alias ACTIVITY	endtime	Starttime				
Usr_batch_char_batch	Tag from point_group MAT_CHAR with alias BATCH_ID	endtime	Starttime				
Usr_batch_char_value	Tag from point_group CHAR with alias VALUE	endtime	Starttime				
Usr_batch_create_ar	Tag from point_group MAT_CHAR with alias BATCH_ID	endtime	Starttime	Tag from point_group MAT_CHAR with alias PPPI_BATCH_NEW			
Usr_confirmation_text_sec	Tag from PI_BATCH with alias SHORT_TEXT	endtime	Starttime				

Translation Procedure	*Data fields Returned*						
Usr_phcon_time_status	recipe_id	phase_id	phase_status_tag, PI-BATCH alias SAP	recipe_tag	timestamp		
Usr_post_date	Tag from PI_BATCH with alias POST_DATE	endtime	Starttime				
Usr_post_date_sec	Tag from SEC_RES with alias POST_DATE	endtime	Starttime				
Usr_scrap	Tag from PI_BATCH with alias SCRAP	endtime	Starttime				
Usr_sract_activity	Common_name tag for secondary resource	endtime	Starttime				
Usr_sract_confirmation_text	Common_name misc tag1 for secondary resource	endtime	Starttime				
Usr_sract_monitor	Common_name misc tag2 for secondary resource	endtime	Starttime				
Usr_srst_confirmation_text	Tag from SEC_RES with alias SHORT_TEXT	endtime	Starttime				
Usr_srst_monitor	Recipe_id	Phase_id	Phase_tag	Recipe_tag	starttime		
Usr_srst_reason_for_variance	Tag from SEC_RES with alias REASON	endtime	Starttime				
Usr_time_status_sec	Recipe_id	Phase_id	Phase_tag	Recipe_tag	starttime		
Usr_phact_confirmation_text	Common_name misc tag1 for resource	endtime	Starttime				
Usr_phact_status_monitor	Common_name misc tag2 for resource	endtime	Starttime				
Usr_misc1(4)_tag	Misc_tag from material_tag	Endtime	starttime				
Usr_storage	Request_id	Endtime	Starttime	Material_id	Resource_id	PPPI_STORAGE_LOCATION	
Usr_batchid_tag_kk	Batch_id tag from matrial_tag	Endtime	Starttime	Recipe tag from misc tag1	Recipe no	Material quantiy tag	Material no
Usr_batch_flow_tag_kk	Quantity tag from material_tag	Endtime	Starttime	Recipe tag from misc_tag1 of material tag	Recipe number	Material quantity tag	Material no

SAP By-Products

A by-product in SAP R/3 requires that a PI_CONS instruction be returned to SAP but it also requires that the AMAT_1 is generated with a negative quantity. For a by-product

you would make a specific ACONS_1 instruction in the recipe for the by-product. You would also be required to make all APROD_1 instructions in the recipe specific, ie.enter the phase, operation and the material. The material would be entered in the configuration application at type “C”.

With version 1.6 of RLINK we have added the support of the characteristics PPPI_MATERIAL_CO_PRODUCT and PPPI_MATERIAL_BY_PRODUCT. These are characteristics of a single character and if the value is X it indicates if they are a by-product or co-product. If a material is a by-product it will be returned to SAP as a PI_CONS and if the material is a co-product it will be returned as a PI_PROD. In the AMAT instruction you would include this characteristic. This only works for the AMAT not the AMATP01. You are able to set up an alias for the characteristics by using the external_alias tables.

Notes on Instructions

- The minimum instruction set for a recipe is AORD, AMAT (for each material), APROD, ACONS, APHASE, ACRST, AOPST, APHST.
- AMATP01 is for materials produced. Normally this does not come down with an operation or a phase and it is assigned to be the last operation and phase in the recipe. If and operationor phase is given it will accept those values.
- Text notes added to any of the above instructions will be loaded into the SQL Server database for retrieval but they have not been moved for storage in PI at this time. The user can retrieve them for use in their own application if that is desired.
- Messages received are stored in the SQL Server database but these are also not downloaded into PI at this time, however like text comments added to instructions they can be retrieved to be used in their own application.
- If the recipe is defined with only one ACONS instruction which does not give a specific phase, operation and material in the instruction the system will automatically create messages for all phases, operations and materials which are consumed which were defined in the recipe
- If the recipe is defined with only one APROD instruction that does not give a specific phase, operation and material in the instruction the system will automatically create messages for all phases, operations and materials which are produced which were defined in the recipe. It determines a material to be produced by a neqative quantity. If the material to be produced does not have a AMAT with a negative quantity then you must define a APROD instruction in the recipe and give the material, phase and operation.
- If the recipe is defined with only one APHST instruction that does not give a specific phase in the instruction the system will automatically create messages for all phases that were defined in the recipe.
- If the recipe is defined with only one AOPHST instruction that does not give a specific operation in the instruction the system will automatically create messages for all phases that were defined in the recipe.
- The EVENT_TIME and EVENT_DATE requests are satisfied by taking the date and time one of the value characteristics in the message request. The characteristic which is used is determined in the setup of the Translator table by assigning the WITH or WITH_ENG to the Reply_method. Time is handled to the level of second resolution through out the application.

- The UNIT_OF_MEASURE request is satisfied if a value is not supplied in the SAP/R3 download by using the Engineering Unit returned with one of the characteristic values assigned. The characteristic that is used is determined in the setup of the Translator table by assigning WITH_ENG to the Reply_method.
- If the recipe is defined with only one APHUST instruction that does not give a specific phase in the instruction the system will automatically create messages for all phases that were defined in the recipe.
- If the recipe is defined with only one AOPUST instruction that does not give a specific operation in the instruction the system will automatically create messages for all phases that were defined in the recipe.
- If you want to automatically detect a change in resource for a phase this in done by including the PPPI_PHASE_RESOURCE as a requested value in the APHST_I instruction. You can configure up to a maximum of 8 resources with the same phase name. Build a point_group for each resource with the same description that is set to the PPPI_EXTERNAL_PHASE. Each of these point groups will have a different set of tags. The system will detect the resource that is being used by monitoring the set of tags for all the phases that apply. When it detects that the recipe is active in a given set it will change the resource to correspond to the active phase location.
- OSI_FINISH_DATE and OSI_FINISH_TIME and be added to a continuous or batch recipe. In the continuous recipe this will override the time that would be calculated with the start time and date plus the shift duration. In the case of a batch recipe it is put into the recipe table and no further processing is done with it unless the customer uses it for their own purposes.
- The APMMD message is not supported because the PM interface is more appropriate for this. If this message was used there is no return of the measurement document and notification thus no correspondence can be stored for malfunction diagnosis.
  An equipment malfunction is a function of time not the process order thus requesting this in conjunction with a process order does not make since.
- ABTCL- You can use a single ABTCL instruction in a recipe for a given material in a given phase (phase and material are sent down in the instruction along with PPPI_ORDER_ITEM_NUMBER, PPPI_PROCESS_ORDER, PPPI_PLANT_OF_BATCH) The characteristics that are to be reported back are determined by looking at the point groups which have been configured for that material on that resource. You should go to the section on point groups to understand how they should be configured for the batch characteristic.
  The values for the batch characteristics can not be sent to SAP until the batch has been created or they will fail to post. The RFC get help values is used to verify that the batch has been created in SAP. When a batch characteristic is to be sent up a request is first formulated to check if the batch exists. Only after it is know that the batch exists is the message for PI_BTCL sent to the table MSHD.
- For the message PI_BT_CL there is an option to check if SAP has the batch number already. This is done with the systemparameter BTCLF that is taken by default to be “N” for not checking. If you want to formulate the check then set the value to “Y”.
- APHCON- if time tickets are to be used in a recipe instead of time events then replace the APHST instruction with APHCON. This instruction also handles activities for time tickets. The tables instruction_requirements and return_message must be loaded to support this instruction. There is a reply_method called UNIT to

handle the engineering unit assignment for the activities of this instruction to assign the PPPI_ACTIVITY_n_UNIT to the engineering unit of the tag for the activity. For APHCON only a status of 00004 and 00002 are returned in PPPI_STATUS_CONFIRMED. The point group type used for configuration is PI-BATCH. The characteristic PPPI_UNIT_OF_MEASURE is used only if PPPI_YIELD_TO_CONFIRM is present.

- Secondary resources are reported using PI_SRST and PI_SRACT if you are using time events and PI_SRCON if you are using time tickets. PI_SRST and PI_SRCON require a point group of type SEC_RES to be configured.
- In the APHAPR instruction the requirement is to have PPPI_PARAMETER_VALUE was removed. This means that you can send down only minimum or maximum values. You are still required to enter a tag in the configuration however. This can be set to a dummy tag.
- Miscellaneous tag translation methods exist for the support of characteristics such as PPPI_FINAL_ISSUE, PPPI_STORAGE_LOCATION, PPPI_STOCK_TYPE and PPPI_DELIVERY_COMPLETE that are associated with a material but a specific translation method has not been given. These translation methods are usr_misc1_tag, etc.
- To prevent PI_CRST messages from going to SAP an field was added to the plant_resource_network. This was added to support a situation in SAP where PI_PHCON message were used and there was a problem in SAP if it got PI_CRST messages. This is configure with a setting in the field crst_disable. If you set the value to “X” then the CRST message will not be sent to SAP.
- How to change the name of PI_CONS and PI_PROD instruction and add additional characteristics. Configuration of the table partial_result_instructions as shown below is required. The new characteristics would have to be entered in the table instruction_characteristics, characteristic and char_format. Characteristic and char_format only hve to be entered if the data is not of format CHAR. Additions must also be made to the translator table for the new instructions and instruction_characteristics.

Partial_result_instructions

return_categor	request_part_name
ZPI_CONS	PPPI_MATERIAL_CONSUMED
ZPI_PROD	PPPI_MATERIAL_PRODUCED

- The only message_categories that would have problems if the message_category name is changed from the SAP standard are PI_CRST, PI_BT_CR, PI_BT_CL, PI_QMSMR and PI_PHST.
- In regard to what characteristic names we use. What follows is a list of the characteristic names we use and expect to find in the instructions. These are standard SAP instruction_characteristics.

PPPI_PHASE

PPPI_RESOURCE

PPPI_ACTIVITY_1 THRU 6

PPPI_CONTROL_RECIPE_STATUS

PPPI_PHASE_STATUS

PPPI_OPERATION_STATUS

PPPI_UNIT_OF_MEASURE

PPPI_EVENT_DATE

PPPI_EVENT_TIME

PPPI_PHASE_RESOURCE

PPPI_PHASE_USER_STATUS

PPPI_REQUESTED_VALUE

PPPI_MATERIAL

PPPI_BATCH_CHARAC_NAME

PPPI_BATCH_NEW

PPPI_MATERIAL_CONSUMED

PPPI_BATCH

PPPI_MESSAGE_CATEGORY

PPPI_MATERIAL_PRODUCED

PPPI_ACTIVITY

PPPI_BATCH_CHAR_VALUE

PPPI_DATA_POINT_NAME

PPPI_DATA-POINT_VALUE

PPPI_STD_VALUE_PARAMETER_ID

PPPI_PARAMETER_NAME

PPPI_PARAMETER_VALUE

PPPI_PARAMETER_MIN

PPPI_PARAMETER_MAX

PPPI_INSPECTION_CHARACTERISTIC

PPPI_MATERIAL_ITEM

PPPI_MATERIAL_SHORT_TEXT

PPPI_MATERIAL_QUANTITY

PPPI_RESERVATION_ITEM

PPPI_RESERVATION

PPPI_EXTERNAL_PHASE

PPPI_SHORT_TEXT

PPPI_ORDER_QUANTITY

PPPI_RESOURCE_NETWORK

PPPI_ACTIVITY_1_UNIT THRU 6

PPPI_DATE_REQUEST_TYPE

PPPI_STATUS_CONFIRMED

PPPI_MESSAGE_TEXT

PPPI_SECONDARY_RESOURCE

PPPI_YIELD_TO_CONFIRM

PPPI_SECONDARY_RESOURCE

PPPI_INSPECTION_LOT

PPPI_STANDARD_DEVIATION

PPPI_INSPECTION_RESULT

PPPI_NUMBER_OF_INSPECTIONS

- If you want to pass information from PI to SAP in message characteristics that are not standard SAP messages characteristics. This can be done as follows:
  1. If this is an entirely new message_category you can use the general SAP transactions or create their own entry in the tables MSHD and MSEL
  2. If these are characteristics added to existing instructions then you can use the standard tools in the product
- If you want to pass SAP information to PI through instruction characteristics that are not standard. This can be done as follows:
  1. You can send down an Ad-hoc message and write a procedure about what to do with the data. A sample procedure is included in the manual.
  2. If the information is about a material you can use the functionality of write to PI for a material.
  3. You can send values to PI by using the APHPAR instruction for parameter values.
  4. If you send down some instruction in the recipe that is totally undefined then a procedure must be written that knows what to do with it. There are instructions in the manual on how to write these.
- All the CONS and PROD interpretation procedures where changed to support the addition of PPPI_POSTING_DATE added to these instructions changing the format of this characteristic to DATE
- For one customer PHCON had several additional custome characteristics and translation methods added they are documented here. ZPPPI_ACTIVITY_5_UNIT was configure with the alias CHG_CODE and the translation method usr_chg_code. The characteristic ZPPPI_ACTIVITY_6_UNIT with alias UNCHG_CODE and usr_unchg_code was added. PPPI_REASON_FOR_VARIANCE with the alias REASON and translation procedure fo usr_reason_phcon.
- An application for the sesolution of the PPPI_RESERVATION and PPPI_RESERVATION_ITEM based on the material_list table not including batch was added with application usr_rs_and_rsi_app_nobatch. It really makes not sense to have reservations when there are no batches.
- The procedure usr_msg_hdr22 sequences PHACT messages after PHST messages.
- ZI_PHST2 is added to be supported in addition to PI_PHST in the corresponding messages. If the user choses this message_category then they must change the instruction_requirements table accordingly.
- The APHPAR instruction was modified to allow for alias values for PPPI-PARAMETER_VALUE and PPPI_UNIT_OF_MEASURE. This enables these

values to be set by a program in SAP. In order for these values to be accepted you need to configure the external alias table as shown below.

alias_value	internal_value	alias_system_id	alias_description
ZPPI_PARAMETER_VALUE	PPPI_PARAMETER_VALUE	5	PPPI_PARAMETER_VALU
ZPPI_UNIT_OF_MEASURE	PPPI_UNIT_OF_MEASURE	5	PPPI_UNIT_OF_MEASUR

Notes on Applications

- The standalone summary application was modified to accept the parameter for the percentage good results. In execute batch create an entry for summary.

  Param1: -T for tracing (Default is no trace)
  Param2: -P90.5 for passing percentage (Default value is 100%)
  Param3: -C use conversion value from PI. (Default is no conversion)
  When this program is to be used remove the entry SUMMARY from the table PI_FUNCTIONS (These are the programs to be executed by pimod.exe)

- The standalone max application has the parameter –P for percent good added. If this param eter is passed it will use this value otherwise the default check is made with 100%. The –C will indicate not to use conversion.

- A new program summaryplus was added that to a totalized value will add a value from a second tag that is configured in material_tag misc_tag4 and is set in field6 of action_results.

  Param1: -T for tracing (Default is no trace)
  Param2: -P90.5 for passing percentage (Default value is 100%)
  Param3: -C use conversion value from PI. (Default is no conversion)

- Updated the program gettag to trap memory problems. These updates are only in the standalone programs and thus entries in exec_batch are required to run them and the entry in the table pi_function should be removed for GETTAG so they will not be executed by pimod.

- Putoption application

Putoption application is used to put multiple values at the same timestamp in PI.

Putoption takes 2 cmdline parameters –T and –M<mode>. Both are optional. The default mode used in the application is 4 (append). eg., Putoption.exe –T –M4
-T is for logging debug information into a flat file.
-M<mode> takes PI archive mode as parameter.

The valid archive modes and its descriptions are

ARCNOREPLACE 3 /* add unless event(s) exist at same time (PI2.x) */
ARCAPPEND 4 /* add event regardless of existing events */
ARCREPLACE 5 /* add event, replace if event at same time */
ARCREPLACEX 6 /* replace existing event (fail if no event at time) */
ARCDELETE 7 /* remove existing event */
ARCAPPENDX 8 /* add event regardless of existing events, no compression*/

The stored procedure usr_write_data_pi, populates record in action_send for group_type 'WTPIM'. The default trigger_proc set for these records are ‘putvalue’. If we want to change the trigger_proc to be ‘putoption’, pass parameter 2, to the stored procedure usr_write_data_pi.

Putoption application looks for records with trigger_proc equal to ‘putoption’ in action_send. According to the mode parameter, putoption application will set the value in PI.

Putoption application requires the following configuration in rlink.

1. An entry for putoption in pi_functions2.

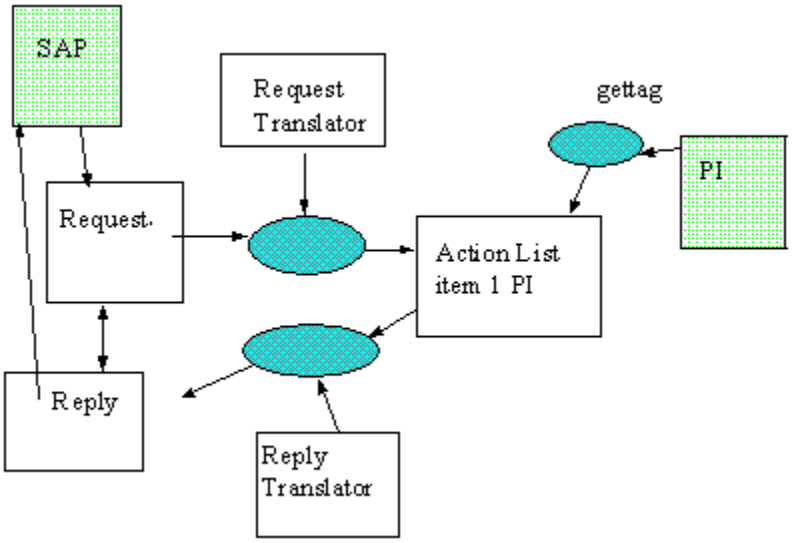
id	program_name	servercol
42	putoption	piserver

2. An entry for putoption in exec_batch with mode parameter as required. Note that the putoption should be added in the same group_no as putvalue.exe. The functionality is set to “Append or replace value in PI.

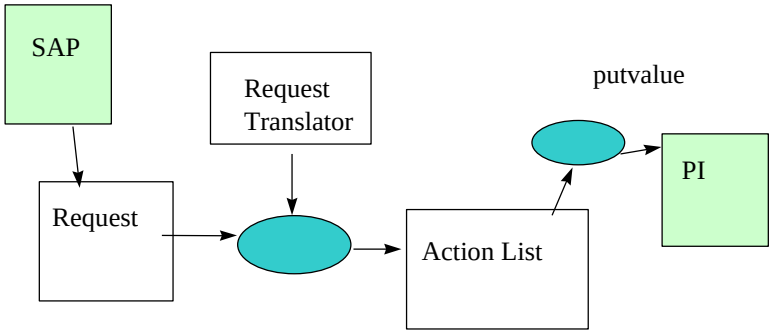
3. Update on usr_write_data_pi entry in exec_batch to pass parameter 2 by changing the entry for the program name to usr_write_data_pi 2

Data Flow Diagrams

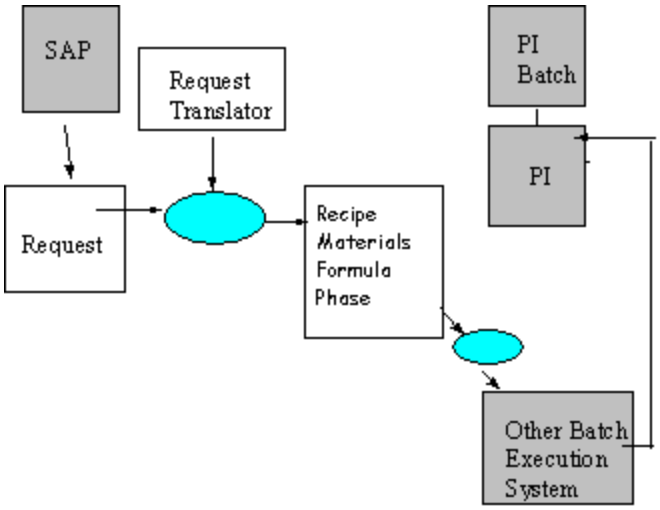
- SAP/R3 Message Data Request



- SAP/R3 Message Data Send



- SAP/R3 to Batch Execution System



Specialized Configuration Senerios

Senario 1

At one customer the assignment of material to a resource was highly variable. The configuration of every material for every possible resource would require excessive configuration. Instead it was desired to configure a set of points for each resource and allow the operator to assign at run time the resource that would be used for the material. The following is a description of the configuration for this situation.

Application

	application_descripti	program_name	field1_name	field2_name	field3_nam	field4_name
71	line selection	line	TAGID	TIMESTAMP		

Group_master

group_no	group_desc	batch_no	last_exec_dtime	frequency_mi	frequency_hr
47	line	1	8/24/2000 6:17:17 PM	1	0

Exec_batch

If you have not installed on the D drive the path must be changed

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
d:\psrlink\server\fe\lineres.exe	1	line selection	E	47	1

Point_group and point_group_members

The first group will hold the line selection for a material. The value that goes into the tag must have the format recipe_id/material_id/line where line is the group_description for the individual equipment-lines.

group_num	group_description	group_type	resource_id	plant_id	application_id	owner	eqp_str
191	LINE_SEL	LINE_SEL	X	BYBA		dbo	NU

group_num	tag_id	tag_alias	display_order	server	application_no
191	Short_1111	LINE_SEL	1	piserver2	

For each equipment-line

A group must be defined for each equipment-line.

group_num	group_description	group_type	resource_id	plant_id	application_id	owner	eqp_stream_flag
129	FEEDER_1	MAT_CON	CODO13	BYBA		dbo	NU
130	FEEDER_2	MAT_CON	CODO13	BYBA		dbo	NU
131	FEEDER_3	MAT_CON	CODO13	BYBA		dbo	NU
132	FEEDER_4	MAT_CON	CODO13	BYBA		dbo	NU
133	FEEDER_5	MAT_CON	CODO13	BYBA		dbo	NU
134	FEEDER_6	MAT_CON	CODO13	BYBA		dbo	NU
135	FEEDER_7	MAT_CON	CODO13	BYBA		dbo	NU
136	FEEDER_8	MAT_CON	CODO13	BYBA		dbo	NU
137	FEEDER_9	MAT_CON	CODO13	BYBA		dbo	NU
138	FEEDER_10	MAT_CON	CODO13	BYBA		dbo	NU

Members are shown for only one group

If the group is a product resource for production then replace the PPPI_MATERIAL_CONSUMED with PPPI_MATERIAL_PRODUCED. Custom characteristics are shown here in addition to the standard characteristics.

group_num	tag_id	tag_alias	display_order	server	application_no
129	m6	BATCH_ID	6	piserver2	29
129	m7	ZPPPI_LGTYP	7	piserver2	19
129	reason_1111	PPPI_BATCH	2	piserver2	29
129	reason2_1131	PPPI_MATERIAL	1	piserver2	19
129	color009	PPPI_MATERIAL_CONSUMED	5	piserver2	29
129	m4	ZPPPI_LGNUM	4	piserver2	19
129	m3	ZPPI_LGPLA	3	piserver2	19

Translation_methods

name	description
usr_line_selection	selects line

Translator

request_part_name	request_category	subscriber_id	application_no	translate_method	reply_method	rep	plant_id	resource_network
PPPI_BATCH	ACONS_1	11	71	usr_line_selection	WITHOUT		BYBA	X
PPPI_MATERIAL_CONSUMED	ACONS_1	11	71	usr_line_selection	WITH_ENG		BYBA	X
ZPPPI_LGNUM	ACONS_1	11	71	usr_line_selection	WITHOUT		BYBA	X
ZPPPI_LGPLA	ACONS_1	11	71	usr_line_selection	WITHOUT		BYBA	X
ZPPPI_LGTYP	ACONS_1	11	71	usr_line_selection	WITHOUT		BYBA	X
PPPI_DELIVERY_COMPLETE	APROD_1	11	71	usr_line_selection	WITHOUT		BYBA	X

Stored Procedures

Usr_line_selection – This procedure is used to set up the request in action_results for finding the sub-resource assignment

Usr_line_ar_upd – this procedure is used to change the entry in action_results for the specific instruction characteristic base on the point group and alias and application once the sub-resource is known.

Applications

The application lineres.exe is provide which will read the PI tag which holds the recipe/material/sub-resource. This application will match the recipe/material provided and find the sub-resource. The application will then call the stored procedure usr_line_ar_upd to update the action_results table based on the sub-resource assigned.

Dynamic Point Assignment

Dynamic Point Assignment works by first selecting the tag that will hold the assignment of the for the equipment or sub-resource assignment. The tag which will hold this value is the in the point_group with the type LINE_SEL as the group type and with the tag alias of LINE_SEL. A Bayer application will write to this tag in the following format

Recipe_no/material_no/sub-resource where the sub-resource must match the description field of one of the point_groups to be selected for the material.

The translation method and application must be configured in the translator table that will use this line selection method. This is shown above configured for the ACONS_1 instruction. The translation method usr_line_selection and the application 71 are chosen that corresponds to lineres.exe as configured in the application table.

The application that is to be used to retrieve the specific value for the point must be coded in the point_group_members table. For example if multival is to be used then application 29 is selected.

Requesting material to an additional Feeder

The purpose of this material to feeder customization is to allow the assignment of the same material to multiple feeders.

The basic assumption is that there will be only one ACONS in the recipe. This ACONS will generate multiple message requests based upon the number of consumed materials sent down in the recipe.

This update will allow for the generation of additional message requests for the assignment of materials to multiple feeders.

We would need to change the convention of the tag to be

Recipe/material/occurrence number/feeder

By default the one ACONS in the recipe would be assigned to occurrence number 1.

This would require a change in the following

1. Translation method usr_line_selection2 to include the occurrence = 1 will replace the translation method usr_line_selection
2. The program the consultants wrote to write a value to the PI tag to now write the new format of Recipe/material/occurrence number/feeder

The stored procedure usr_material_duplicate_cons is used to create a new message request for the same material in a different feeder. The variable input to this procedure is Recipe, Material, Phase Alias and occurrence number. This is a change from the original specification. We had to substitute Phase Alias for Resource in case at a future time you would have more than one phase. The following shows how this would be called given the sample recipe as of Aug. 2000.

usr_material_duplicate_cons "100000000000000506", "000000000004950925", "31", 4

This procedure inserts into messge_request, request_part , request_part_values and action_results. After this insertion the regular processing continues as before.

If you clean up a recipe to re-execute it you must also re-execute the above procedure.

If the batch_id is also to be included then the user would select the translation method usr_line_selection3 that will format the request as recipe/material_id/occurance/batch_id/feeder.

Translation Method

name	description
usr_line_selection2	selects line multiple material
Usr_line_selection3	Selects line multiple material with batch

Translator

Note that you should change the translation method in the translator table to usr_line_selection2 or 3 depending on the one required.

request_part_name	request_category	subscriber_id	application_no	translate_method	reply_method	r e p	plant_id	resource_network
PPPI_BATCH	ACONS_1	11	71	usr_line_selection2	WITHOUT		BYBA	X
PPPI_MATERIAL_CONSUMED	ACONS_1	11	71	usr_line_selection2	WITH_ENG		BYBA	X
ZPPPI_LGNUM	ACONS_1	11	71	usr_line_selection2	WITHOUT		BYBA	X
ZPPPI_LGPLA	ACONS_1	11	71	usr_line_selection2	WITHOUT		BYBA	X
ZPPPI_LGTYP	ACONS_1	11	71	usr_line_selection2	WITHOUT		BYBA	X
PPPI_DELIVERY_COMPLETE	APROD_1	11	71	usr_line_selection2	WITHOUT		BYBA	X

Three stored procedures are provided with this customization.

Usr_line_selection2 – This procedure is used to set up the request in action_results for finding the sub-resource assignment including the occurrence number of a material

Usr_line_selection3 – This procedure is used to set up the request in action_results for finding the sub-resource assignment including the occurrence number of a material and batch_id

Usr_material_duplicate_cons- Inserts records for a duplicate cons of the same material

Scenario 2

The purpose of this customization is to provide a method of using one set of tags per resource for the collection of information about material production and consumption. Provided are two methods for accomplishing this. Method 1 uses the material_tag table for configuration and method 2 uses the point_groups for configuration. The advantage of the point grouping is that it does not require new configuration if an additional material is added.

Applications have been added that check a group of tags for values all at the same timestamp checking for the recipe number and material number before retrieving the material quantity or the batch.

The number of tags which are required are 4 per resource.

Table Changes

Application

application_no	application_description	program_name	field1_name	field2_name	field3_name	field4_name	field5_name	field6_name	field7_name
69	batch id	getbatchid	BATCH TAG ID	ENDTIME	STARTTIME	RECIPE TAG ID	RECIPE ID	MATERIAL TAG ID	MATERIAL ID
70	batch id with wait	getbatchidwait	BATCH TAG ID	ENDTIME	STARTTIME	RECIPE TAG ID	RECIPE ID	MATERIAL TAG ID	MATERIAL ID
71	batch quantity	getbatchqty	QUANTITY TAG ID	FINISHTIME	STARTTIME	RECIPE TAG ID	RECIPE ID	MATERIAL TAG ID	MATERIAL ID and field8 BATCH
72	batch quantity with wait	getbatchqtywait	QUANTITY TAG ID	FINISHTIME	STARTTIME	RECIPE TAG ID	RECIPE ID	MATERIAL TAG ID	MATERIAL ID and field8 BATCH

Group_master

group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
51	batch material match group	1	9/7/00 1:43:45 PM	2	0

Exec_batch

If you have not installed on the D drive the path must be changed

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
d:\psrlink\server\fe\batchid.exe	1	batch id app with wait and no wait	E	51	1
d:\psrlink\server\fe\batqty.exe	2	batch quantity app with wait and no wait	E	51	1

Translation_methods

name	description
usr_batch_flow_tag2	material 3Tag Applic
usr_batch_flow_tag3	Material point group

name	description
usr_batchid_tag2	batch_id 3Tag Applic
usr_batchid_tag3	Batch_id point group

Translator

request_part_name	request_category	subscriber_id	application_no	translate_method	reply_method	plant_id	resource_network
PPPI_BATCH	ACONS_1	87	70	usr_batchid_tag3	WITHOUT	7000	R_OSI
PPPI_MATERIAL_CONSUMED	ACONS_1	87	72	usr_batch_flow_tag3	WITH_ENG	7000	R_OSI

Stored Procedures

Usr_batch_flow_tag2 – translation method using materiall_tag for material quantity

Usr_batch_flow_tag3 – translation method using point_group for material quanity

Usr_batchid_tag2 – translation method for batch_id using material_tag

Usr_batchid_tag3 – translation method for batch_id using point_group

Applications

The applications are batchid.exe and batqty.exe

Batchqty works as follows

1. First the tag for the recipe is located and from the start to the end time is searched to see if there is an entry for the given recipe.
2. The tag for the material is located and matched to the material requested.
3. If both the recipe and material match then the quantity is retrieved at the given time
4. The batch is then gotten at that time
5. An array is built of material, batch, quantity, timedate
6. At the end of the time range all the entries with the same batch are added and the last timedate for the batch is used.
7. The wait option will wait for the recipe tag to have a value at the endtime or a value after that time.

Batchid.exe works as follows

1. First the tag for the recipe is located and from the start to the end time is searched to see if there is an entry for the given recipe.
2. The tag for the material is located and matched to the material requested.
3. If both the recipe and material match then the batch_id is retrieved at the given time
4. An array is built of material, batch, timedate

5. At the end of the time range all the entries with the same batch are reviewd and the last timedate for the batch is used.
6. The wait option will wait for the recipe tag to have a value at the endtime or a value after that time.

Translation Methods

Translation method using material_tag for configuration for the quantity tag using general tags.

Use misc tag 3 for the recipe tag and misc tag 4 for the material tag

The translation method usr_batch_flow_tag2 will set up the following request in action_results

Field 1 quantity tag

Field 2 finish time

Field 3 start time

Field 4 recipe tag from misc tag 3

Field 5 recipe no

Field 6 material tag from misc tag4

Field 7 material no

Field 8 batch tag

The translation method usr_batch_flow_tag3 uses the point_group and point_group_members configuration. In this case only one group is configured for a resource. This procedure will construct the same input into action_results selecting the alias for PPPI_BATCH, RECIPE_ID, PPPI_MATERIAL_CONSUMED or PPPI_MATERIAL_PRODUCED and PPPI_MATERIAL.

Point_group and point_group_members

The group_description corresponds to the PPPI_EXTERNAL_PHASE name that is given in the recipe.

group_num	group_description	group_type	resource_id	plant_id	owner
517	PHASE_NAME_1	MAT_CONS	R_1111	1200	dbo
518	PHASE_NAME_2	MAT_PROD	R_1111	1200	dbo

group_num	tag_id	tag_alias	display_order	server	application_no
517	GENERIC_C1_ST	PPPI_BATCH	1	piserver2	
517	GENERIC_C3_ST	PPPI_MATERIAL	2	piserver2	
517	GENERIC_c1_FL	PPPI_MATERIAL_CONSUMED	3	piserver2	
517	GENERIC_C8_ST	RECIPE_ID	4	piserver2	
518	GENERIC_P1_ST	PPPI_BATCH	1	piserver2	

group_num	tag_id	tag_alias	display_order	server	application_no
518	GENERIC_P3_ST	PPPI_MATERIAL	2	piserver2	
518	GENERIC_P1_FL	PPPI_MATERIAL_PRODUCED	3	piserver2	
518	GENERIC_P7_ST	RECIPE_ID	4	piserver2	

Translation method using material_tag for configuration for the batch_id using general tags.

Use misc tag 3 for the recipe tag and misc tag 4 for the material tag

The translation method usr_batchid_tag2 will set up the following request in action_results

Field 1 batch tag

Field 2 finish time

Field 3 start time

Field 4 recipe tag from misc tag 3

Field 5 recipe no

Field 6 material tag from misc tag4

Field 7 material no

The translation method usr_batchid_tag3 uses the point_group and point_group_members configuration. In this case only one group is configured for a resource. This procedure will construct the same input into action_results selecting the alias for PPPI_BATCH, RECIPE_ID, and PPPI_MATERIAL.

Extending this for storage location and storage type requires additional application programs and translation methods. It would be recommended to send these values down with specific ACONS instructions in the recipe.

Scenario 3

Purpose

At Janssen they have one tank that is feeding 2 different phases at the same time. The configuration of a material and a resource is not sufficient to get 2 different sets of tags for the material and batch_id. Instead the use of the point_group table based on the PPPI_EXTERNAL_PHASE will be used where the different recipes will have different values for the PPPI_EXTERNAL_PHASE name

Point_group for Tag assignments

The two stored procedures used by Janssen for the assignment of the material tag and the batch_id tag have been modified. These procedures will now look for the table phase_alias. The two changed procedures are usr_material_from_batch (mttagbch.qry) and usr_batchid_tag (batchbat.qry). If an entry has been configured in this table for the phase_alias as given in the characteristic PPPI_EXTERNAL_PHASE then the

assignment of the tags will be made using the point_group and point_group_members tables.

Table Changes

New table called Phase_alias

This table is used to signal that the point groups should be used rather than the material_tag configuration for the selection of tags for the materials.

Phase_alias	Resource_id	flag
S21P1_SOLV_CHARGE	?? put in resource	Y
S21P2_SOLV_CHARGE	?? put in resource	Y

In the point group for the phase alias you must include entries for the material and batch with the alias names MATERIAL and MAT_BATACH. Suppose you have two point groups 129 and 130 corresponding to each of the different PPPI_EXTERNAL_PHASE then the added members of the point_group_members table will be as follows:

group_num	tag_id	tag_alias	display_orde	server
129	STF1_S21P1_QNT	MATERIAL	6	piserver2
129	STF1_S21P1_TxLotid	MAT_BATCH	7	piserver2
130	STF1_S21P2_QNT	MATERIAL	6	piserver2
130	STF1_S21P2_TxLotid	MAT_BATCH	7	piserver2

Chapter 6
Recipe Execution

Steps to Process and Monitor a Continuous Process Recipe

1. Setup the recipe in SAP/R3. For a continuous Process the OSI_START_DATE and OSI_START_TIME should be set in the AORD instruction. The OSI_FINISH_DATE and OSI_FINISH_TIME ar optional and are used if you want to fix an end time in a continuous recipe different from what would be calculated with the duration.
2. Check that for all data values that are requested a translation has been setup in either Material_tag or Common_name tables.
3. Check that an entry exists in the translator table for every request and entries in this table reference your plant.
4. If any of the default application methods for the characteristic are to be overridden these should be noted in the Material_tag or Common_name table.
5. Verify that the Location table is configured correctly with the duration of the standard process order (ie. Shift or day etc) if partial readings are to be returned during this period set up the time increment on the partial reading. If partial readings are to be made you must also indicate on which instructions in the partial_results_instruction table.
6. If you hve a plant that has different durations based on resourse networks then these are configured in the plant resource network table. If given these values override the set for the plant.
7. Verify that the other tables have been loaded correctly for your installation ie. Plant, SAP_message_alias.
8. Startup the process TCRD and PSRLINK.
9. Create the recipe for test and review the status of the recipe in the SAP/R3 Control Monitor, transaction CO53. If the recipe destination has been set up to push the recipes down to PSRLINK then you must send the recipe. If the recipe destination is set to type 3 meaning they will be pulled by PSRLINK then a record is inserted in CRA_TO_CRP notifying that a recipe is available. There will be a delay until the TCRPS process is executed. Programs will be executed in the order and at the frequency specified in the group_master and exec_batch tables. You can force execution of TCRPS from the Icon setup
10. Once the recipe is down it is further checked for accuracy. If an error is detected it will be marked with the explanation of the problem. If there are no errors processing will continue. These errors appear in the error_log table.
11. The process followed is that each request is translated by the translation method and setup for data retrieval at the correct time.

12. When an answer has been retrieved it will be translated into the format required for reply to SAP/R3 and will be sent to SAP/R3 when the PMU task is executed. The messages are in the tables MSHD and MSEL.
13. You can force execution of PMU by calling it from the Icon setup.
14. You can monitor the reply in SAP/R3 by using SAP/R3 transaction CO54.

Steps to Execute and Monitor a Batch Process Recipe

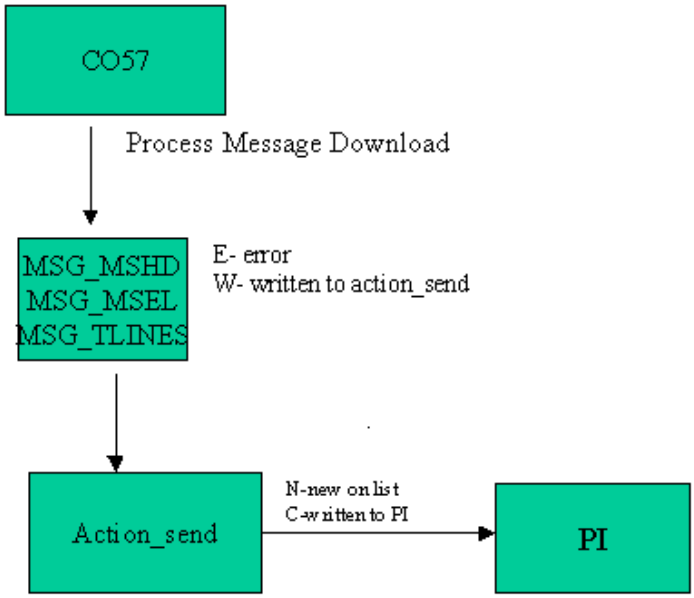
1. Setup the recipe in SAP/R3. The OSI_EXTERNAL_RECIPE should be set in the AORD instruction. The table Instruction_Characteristics that is used to verify the recipes must have the required field changed for OSI_EXTERNAL_RECIPE, OSI_START_TIME and OSI_START_DATE from the default value as delivered. If OSI_FINISH_TIME and OSI_FINISH_DATE are given then these values are moved to the recipe table but not used further.
2. Check that for all data values that are requested a translation has been setup in either Material_tag or Common_name tables.
3. Check that an entry exists in the translator table for every request and entries in this table reference your plant.
4. If any of the default application methods for the characteristic are to be overridden these should be noted in the Material_tag or Common_name table.
5. There must be entries in the Alias_Class, Alias_system and External_Alias tables to support the language conversions for the batch execution system. A material alias table is required for the material names.
6. The control recipe status, phase status and operational status tags should be configured in Point_Group and Point_group_members. The alias tables should be setup to translate for material alias and unit alias. The subscriber table should be setup if there are multiple recipe servers on the network. Each recipe server should be mapped to a SAP/R3 resource network.
7. Verify that the other tables have been loaded correctly for you installation ie. Plant, SAP_message_alias.
8. There must be an executable program that knows how to translate your recipes into those for the batch execution system. There must be entries in the Application table and Group_master and Exec_batch that correspond to this application.
9. The Subscriber and Subscriber_application table must be setup to reference the batch execution system as setup in Exec_batch.
10. Startup the process TCRD and PSRLINK.
11. Create the recipe for test and review the status of the recipe in the SAP/R3 Control Monitor, transaction CO53. If the recipe destination has been set up to push the recipes down to PSRLINK then you must send the recipe. If the recipe destination is set to type 3 meaning they will be pulled by PSRLINK then a record is inserted in CRA_TO_CRP notifying that a recipe is available. There will be a delay until the TCRPS process is executed. Programs will be executed in the order and at the frequency specified in the group_master and exec_batch tables. You can force execution of TCRPS from the Icon setup.

12. Once the recipe is down it is further checked for accuracy. If an error is detected it will be marked with the explanation of the problem. If there are no errors processing will continue.
13. The process that is followed is that each request is translated by the translation method and setup for data retrieval.
14. The recipe is setup in the recipe, formula, material_list tables. When the Openbatch or iBatch program is executed it will select recipes for recipe that have not yet been sent to a recipe processor. It will send them based upon the subscriber setup for the SAP/R3 resource network. The recipe with formula and material list will be sent to the Batch Execution System batch list and then the recipe will be marked as sent.
15. The recipe monitor application which must be setup in the group_master and exec_batch tables with then proceed to monitor the status of the recipe status tag in PI. When it is detected that the recipe has started it will proceed to monitor the status of the operation and phase status tags in PI as configured in Point_group and point_group_members. The change in status of the phase will trigger the reading of data for APROD_1 and ACONS_1 instructions.
16. When an answer has been retrieved it will be translated into the format required for reply to SAP/R3 and will be sent to SAP/R3 when the PMU task is executed. The messages will be in the tables MSHD and MSEL.
17. You can force execution of PMU by calling it from the Icon setup.
18. You can monitor the reply in SAP/R3 by using SAP/R3 transaction CO54.

Chapter 7
SAP Ad-Hoc Messages

Data Flow

Ad-hoc messages can be created in SAP and sent to RLINK using a CO57 transaction. A message destination for ad-hoc messages was created in the installation chapter. When the message is sent down it will be stored in the tables msg_mshd, msg_msel and msg_tlines. This data can be sent to PI with the addition of a custom program which will be discussed here.



Uses of Ad-Hoc Messages

Ad-hoc messages have been used for a number of purposes to exchange data for which there was not an existing defined method of exchange or that was not created at the time of the recipe creation. Some examples of use are as follows:

- The batch ID of a tank changed after the recipe was sent down due to materials being added to the tank. The need was to bring down the new batch ID and assign it to a point at the timestamp of the change so that it would be picked up correctly for the recipe.
- A tank mixing calculation was done in SAP and the properties of the tank were needed at the plant level. The properties were sent down and sent to PI points.
- It was to be decided at the SAP level that a control recipe was to be stopped. The request to stop the recipe was sent down and all the corresponding phases where stoped and the recipe was terminated.

- The properties of the material from a sales order in SAP were sent down to the plant floor for further use.
- This could also be a mechanism for sending down price data and other data required by the plant but there does not exist a function in SAP for data transfer.

Formating the Message

When creating the ad-hoc message you should use a characteristic such as PPPI_SOURCE to distinguish one type of message from another. This will allow you to have multiple message types that can be processed by different program.

The details of the message would be in the PPPI_MESSAGE_TEXT characteristic. If you use the format keyword = value then the data can easily be parsed.

Writing Material Data to PI

We have a generalized procedure for writing material data to PI. Usr_write_to_pi – writes material quantity data to PI that is sent down in ad-hoc message from SAP. This procedure matches on the message class that as set up in the external_alias table to correspond to OSI_WTPIM. It selects the point group that corresponds to the group_type for the given plant using the characteristic PPPI_PLANT_OF_BATCH and the material given in PPPI_MATERIAL. The procedure will loop over the characteristics in msg_msel and if the characteristic has been configured in point_group_members then it will write the value to the tag configured. The characteristics that will not be handled are PPPI_EVENT_DATE, PPPI_EVENT_TIME and PPPI_UNIT_OF_MEASURE. The value is written at the date and time given by PPPI_EVENT_DATE and PPPI_EVENT_TIME. The standard message name is OSI_WTPIM, if this is not to be used than an alias message must be entered in the external_alias table for the alias system SAP.

Group_master

Group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
51	Write data to PI	1	9/7/00 1:43:45 PM	2	0

Exec_batch

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
Usr_write_data_pi	1	Writes the material quantity to PI	P	51	1

Point_group

group_num	group_description	group_type	resource_id	plant_id	equipment_no	application_id	owner
520	000000000010017850	WTPIM		1030			dbo

Point_group_members

group_num	tag_id	tag_alias	display_order	server	application_no
520	tag1	PPPI_BATCH_CHAR	2	piserver2	
520	tag2	PPPI_MATERIAL_CONSUMED	1	piserver2	
520	tag3	PPPI_STORAGE_LOCATION	3	piserver2	
520	tag4	Z0050_001_BEWEGUNGSART	4	piserver2	

If the entry is configured as an alias in the point_group_members table it will write the value sent from SAP to the corresponding PI tag.

OSI_WTPIM

PPPI_BATCH	Batch
PPPI_EVENT_DATE	Date
PPPI_EVENT_TIME	Time
PPPI_MATERIAL	Material number
PPPI_MATERIAL_CONSUMED	Material Consumed
PPPI_PLANT_OF_BATCH	Plant of Batch
PPPI_STORAGE_LOCATION	Storage location
PPPI_UNIT_OF_MEASURE	Unit of Measure
Z0050_0001_BEWEGUNGSART	property

An Example of the received general message in Rlink is:

msg_mshd:

msid	werk	mscla	tstkz	sedat	seuzt	source
100000000000003505	1030	OSI_WTPIM	20000310	123512	U016701	

msg_msel:

msid	atnam	atwrt	atfor
100000000000003505	PPPI_BATCH	33003474	CHAR
100000000000003505	PPPI_EVENT_DATE	20000310	DATE
100000000000003505	PPPI_EVENT_TIME	123512	TIME
100000000000003505	PPPI_MATERIAL	000000000010017850	CHAR
100000000000003505	PPPI_MATERIAL_CONSUMED	1.5000000000000000E+04	NUM
100000000000003505	PPPI_PLANT_OF_BATCH	1030	CHAR
100000000000003505	PPPI_STORAGE_LOCATION	1288	CHAR
100000000000003505	PPPI_UNIT_OF_MEASURE	KG	CHAR
100000000000003505	Z0050_0001_BEWEGUNGSART	101	CHAR

Sending Data to PI

Once the data has been received by RLINK and is located in the tables MSG_MSHD, MSG_MSEL and MSG_TLINES it can be moved to PI points. The method of moving this data is writing a stored procedure which will read the data and find the points it should be mapped to and then update the table action_send with the PI tag and value information. The program putvalue will then move the data to PI.

Configuration to map the data to tags can be done by using the common_name table or by using the point_group and point_group_members tables to store configuration information.

The following is a sample query that could be used for reading a message, mapping it to tags and then inserting it into action_send. In this example mapping is not done rather a fixed naming convention is used for the tag. After the record is inserted into action_send the status of msg_mshd is changed to W.

```
create proc usr_ad_hoc_pi
as
begin
/*
        File name       : ad_hoc_pi.sql
        Modification history

        id      datetime        comment

*/

declare @msid           char(32),
        @mat_id                 char(128),
        @lot_no                 char(128),
        @dt             char(32),
        @tm             char(32),
        @szdttime       char(32),
        @mincnt         int,
        @maxcnt                 int,
        @msg            char(128),
        @resnet                 char(32),
        @address        char(32),
        @plant_id       char(4)

        create table    #temp1
        (
```

```
id              int     identity,
msid            char(18)        null,
plant_id        char(4)         null
)

insert into #temp1(msid, plant_id)
select  distinct mm.msid, mm.werk
from    msg_mshd        mm,
        msg_msel        ms,
        msg_tlines      mt
where   mm.rcode        = “”
and     mm.msid                 = ms.msid
and     ms.atnam        = “PPPI_SOURCE”
and     ms.atwrt        = “LOT_NO”
and     mt.msid                 = mm.msid

select  @mincnt         = min(id),
        @maxcnt         = max(id)
from    #temp1

while @mincnt <= @maxcnt
begin
        select  @msid           = msid,
                @plant_id       = plant_id
        from    #temp1
        where   id      = @mincnt

        select  @mat_id         = LTRIM(RTRIM(substring(mt.tdline,
charindex( “=”, mt.tdline) + 1, datalength(mt.tdline) – charindex( “=”, mt.tdline))))
        from    msg_mshd        mm,
                msg_msel        ms,
                msg_tlines      mt
        where   mm.rcode        = “”
        and     mm.msid                 = ms.msid
        and     ms.atnam        = “PPPI_SOURCE”
        and     ms.atwrt        = “LOT_NO”
        and     mt.msid                 = mm.msid
```

```
and     mt.msid                 = @msid
and     LOWER(mt.tdline) like "%material%"

select  @resnet         = LTRIM(RTRIM(substring(mt.tdline,
charindex( "=", mt.tdline) + 1, datalength(mt.tdline) – charindex( "=", mt.tdline))))
from    msg_mshd        mm,
        msg_msel        ms,
        msg_tlines      mt
where   mm.rcode        = ""
and     mm.msid                 = ms.msid
and     ms.atnam        = "PPPI_SOURCE"
and     ms.atwrt        = "LOT_NO"
and     mt.msid                 = mm.msid
and     mt.msid                 = @msid
and     LOWER(mt.tdline) like "%resource%"

select  @lot_no         = LTRIM(RTRIM(substring(mt.tdline,
charindex( "=", mt.tdline) + 1, datalength(mt.tdline) – charindex( "=", mt.tdline))))
from    msg_mshd        mm,
        msg_msel        ms,
        msg_tlines      mt
where   mm.rcode        = ""
and     mm.msid                 = ms.msid
and     ms.atnam        = "PPPI_SOURCE"
and     ms.atwrt        = "LOT_NO"
and     mt.msid                 = mm.msid
and     mt.msid                 = @msid
and     LOWER(mt.tdline) like "%lot_no%"

select  @dt     = LTRIM(RTRIM(atwrt))
from    msg_msel
where   msid    = @msid
and     atnam   = "PPPI_EVENT_DATE"

select  @tm     = LTRIM(RTRIM(atwrt))
from    msg_msel
where   msid    = @msid
```

```
                and     atnam   = "PPPI_EVENT_TIME"

                select  @szdttime       = convert(char(12), convert(datetime, @dt),
106) +
                                        substring(@tm, 1, 2) + ":" +
                                        substring(@tm, 3, 2) + ":" +
                                        substring(@tm, 5, 2)

                select  @address        = address
                from    subscriber
                where   plant_id        = @plant_id
                and     name            = "PI"
                and     resource_network= @resnet

                if @mat_id != "" and @mat_id != NULL and @lot_no != "" and
@lot_no != NULL
                begin
                        select  @mat_id         = "TAG_NAME" + @mat_id

                        insert into action_send
                        (
                                trigger_timestamp       ,
                                field1                  ,
                                field2                  ,
                                field3                  ,
                                status                  ,
                                status_timestamp        ,
                                trigger_proc            ,
                                subscriber_name         ,
                                subscriber_address
                        )
                        select  getdate()       ,
                                @mat_id                 ,
                                @szdttime       ,
                                @lot_no                 ,
                                "N"             ,
                                getdate()       ,
```

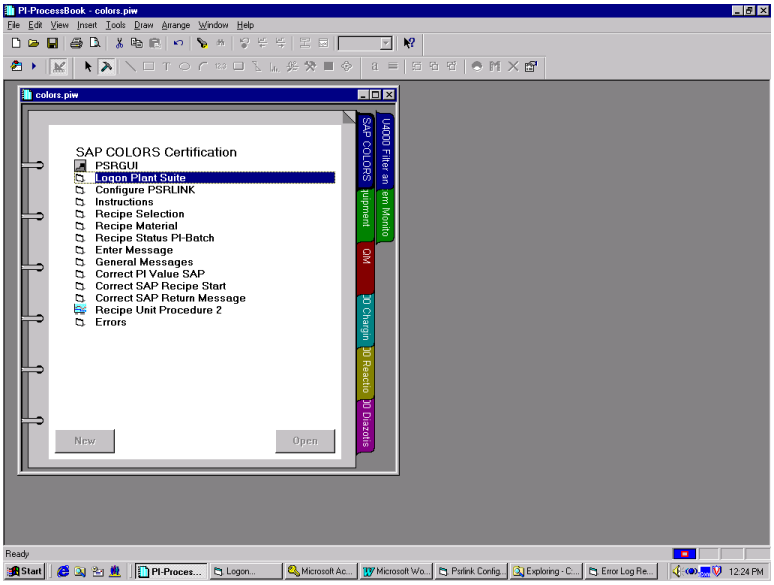
```
                    "putvalue"      ,
                    "PI"            ,
                    @address

                update  msg_mshd
                set     rcode   = "W"
                where   msid    = @msid
            end
            else
            begin
                select  @msg   = "Material id: " + RTRIM(@mat_id) + " Lot
no: " + RTRIM(@lot_no)
                exec    usr_error_log_rfc_i "usr_ad_hoc_pi", @msg, "Material
id or lot no may be null", @msid
            end

            select  @mincnt = @mincnt  + 1
        end

        drop table #temp1
end
```

Once the query has been written you can schedule it for execution by creating an entry in the group_master and exec_batch tables. This will execute the procedure on a regular frequency.

Chapter 8
Graphics Interface

RLINK ProcessBook

The RLINK ProcessBook is provided as a sample ProcessBook for access to the applications useful in reviewing the status of the recipe. It can be used as a prototype for your own application.



The ProcessBook RLINK is provided on the product CD. There are entries in this ProcessBook for each of the Visual Basic and PSRGUI applications to follow. If you have installed ProcessBook on a different directory you might have to correct the paths in the data sets and for the ProcessBook entries.

PSRGUI

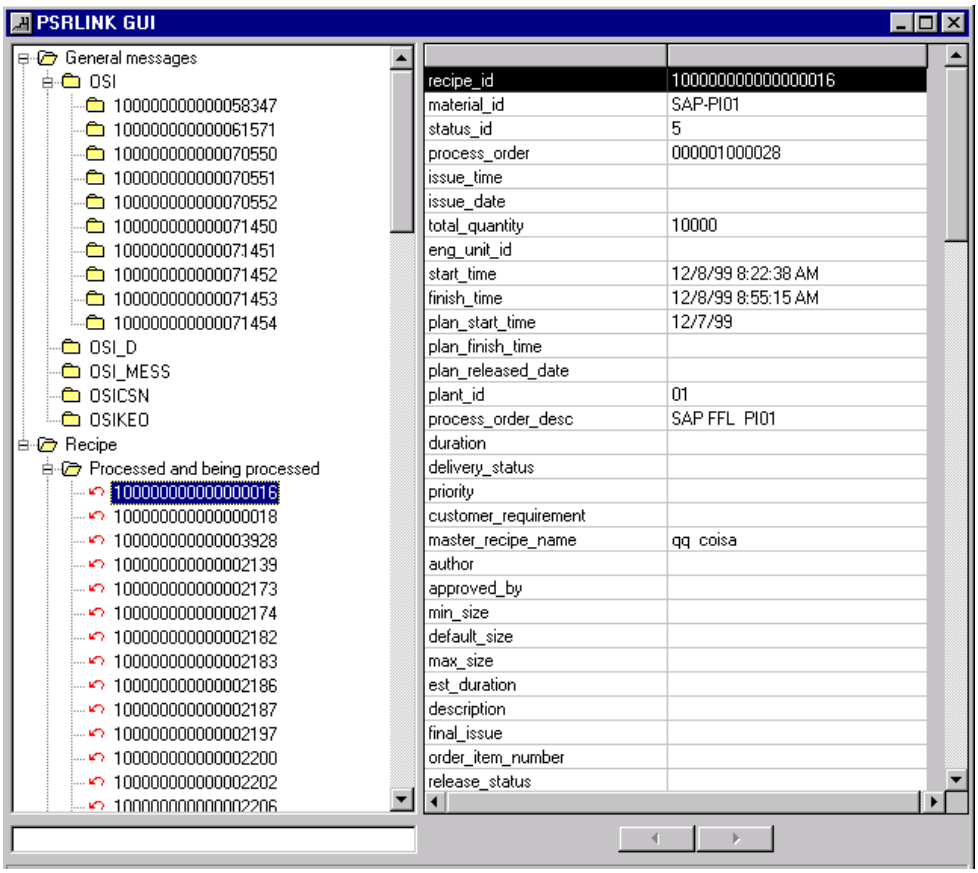
Recipe

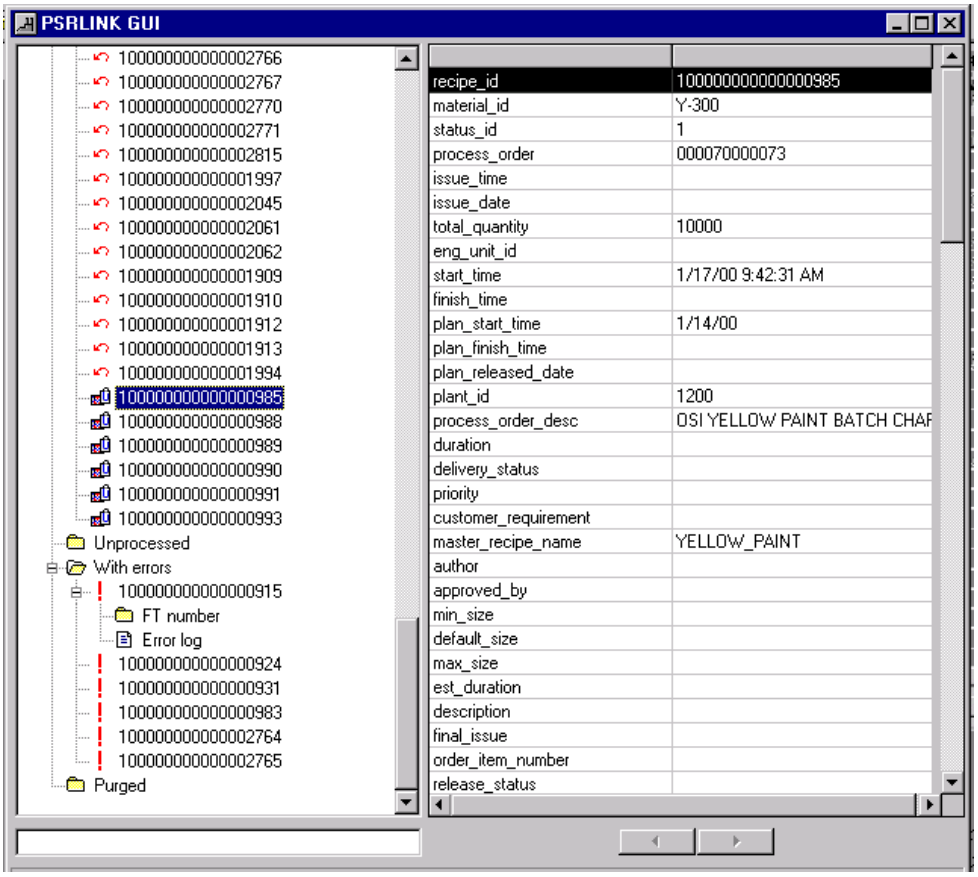
The PSRGUI can be used to review the recipe processing by the system. Recipes are broken down into those processed or being processed, recipes with errors in their formulation and recipes for which processing has not started.

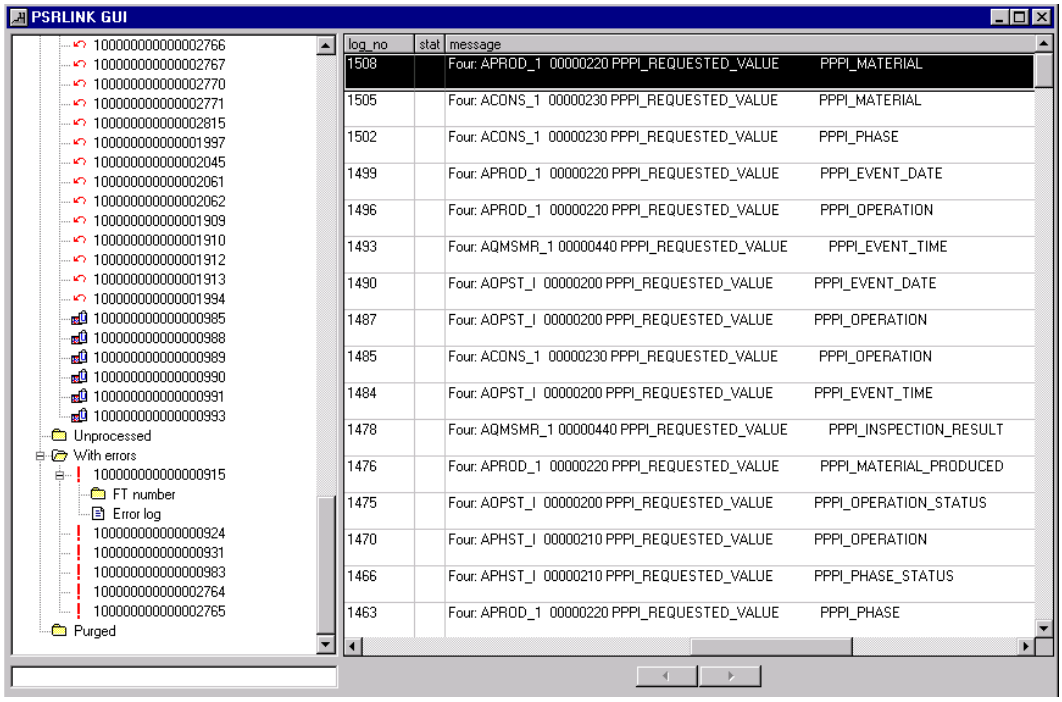
If a recipe is processed or being processed it will be subdivided into Phases, Operations, Message Requests and Instruction Text messages that have been sent. The Phase is divided into materials and formula values for that phase.

If a recipe is in error you can obtain the error that was detected by showing the values in this section.

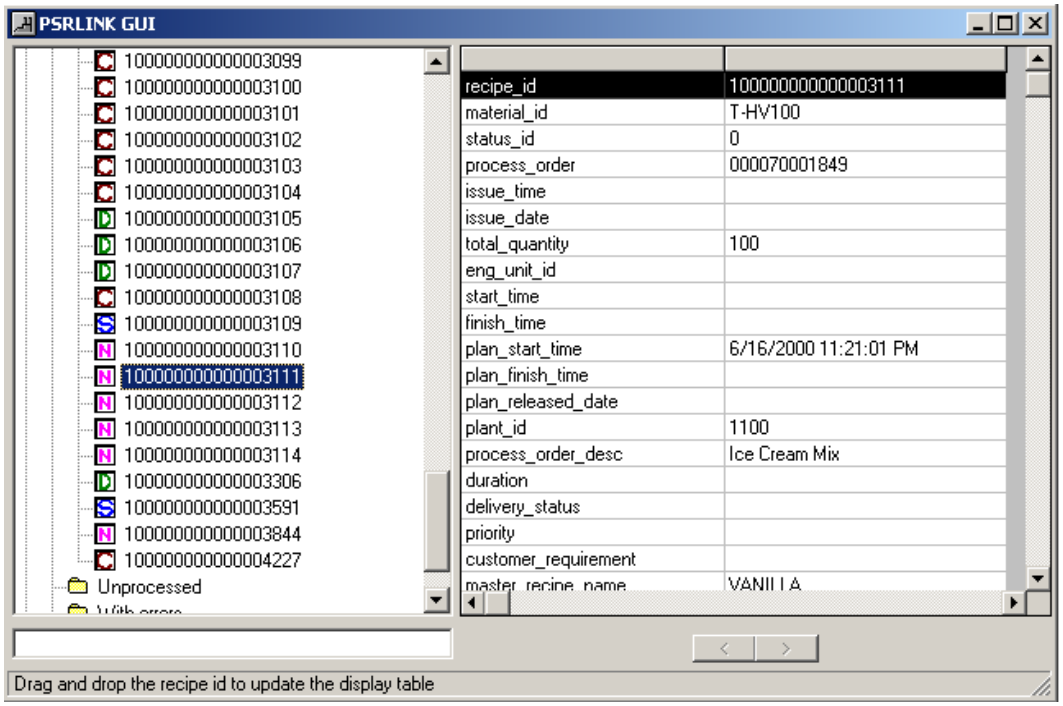
If a recipe has not been processed yes it will display the original data as sent from SAP/R3.


PSRLINK GUI
General messages
OSI
100000000000058347
100000000000061571
100000000000070550
100000000000070551
100000000000070552
100000000000071450
100000000000071451
100000000000071452
100000000000071453
100000000000071454
OSI_D
OSI_MESS
OSICSN
OSIKEO
Recipe
Processed and being processed
100000000000000016
100000000000000018
100000000000003928
100000000000002139
100000000000002173
100000000000002174
100000000000002182
100000000000002183
100000000000002186
100000000000002187
100000000000002197
100000000000002200
100000000000002202
recipe_id 100000000000000016
material_id SAP-PI01
status_id 5
process_order 000001000028
issue_time
issue_date
total_quantity 10000
eng_unit_id
start_time 12/8/99 8:22:38 AM
finish_time 12/8/99 8:55:15 AM
plan_start_time 12/7/99
plan_finish_time
plan_released_date
plant_id 01
process_order_desc SAP FFL PI01
duration
delivery_status
priority
customer_requirement
master_recipe_name qq coisa
author
approved_by
min_size
default_size
max_size
est_duration
description
final_issue
order_item_number
release_status


PSRLINK GUI
100000000000002766
100000000000002767
100000000000002770
100000000000002771
100000000000002815
100000000000001997
100000000000002045
100000000000002061
100000000000002062
100000000000001909
100000000000001910
100000000000001912
100000000000001913
100000000000001994
100000000000000985
100000000000000988
100000000000000989
100000000000000990
100000000000000991
100000000000000993
Unprocessed
With errors
100000000000000915
FT number
Error log
100000000000000924
100000000000000931
100000000000000983
100000000000002764
100000000000002765
Purged
recipe_id 100000000000000985
material_id Y-300
status_id 1
process_order 000070000073
issue_time
issue_date
total_quantity 10000
eng_unit_id
start_time 1/17/00 9:42:31 AM
finish_time
plan_start_time 1/14/00
plan_finish_time
plan_released_date
plant_id 1200
process_order_desc OSI YELLOW PAINT BATCH CHAF
duration
delivery_status
priority
customer_requirement
master_recipe_name YELLOW_PAINT
author
approved_by
min_size
default_size
max_size
est_duration
description
final_issue
order_item_number
release_status



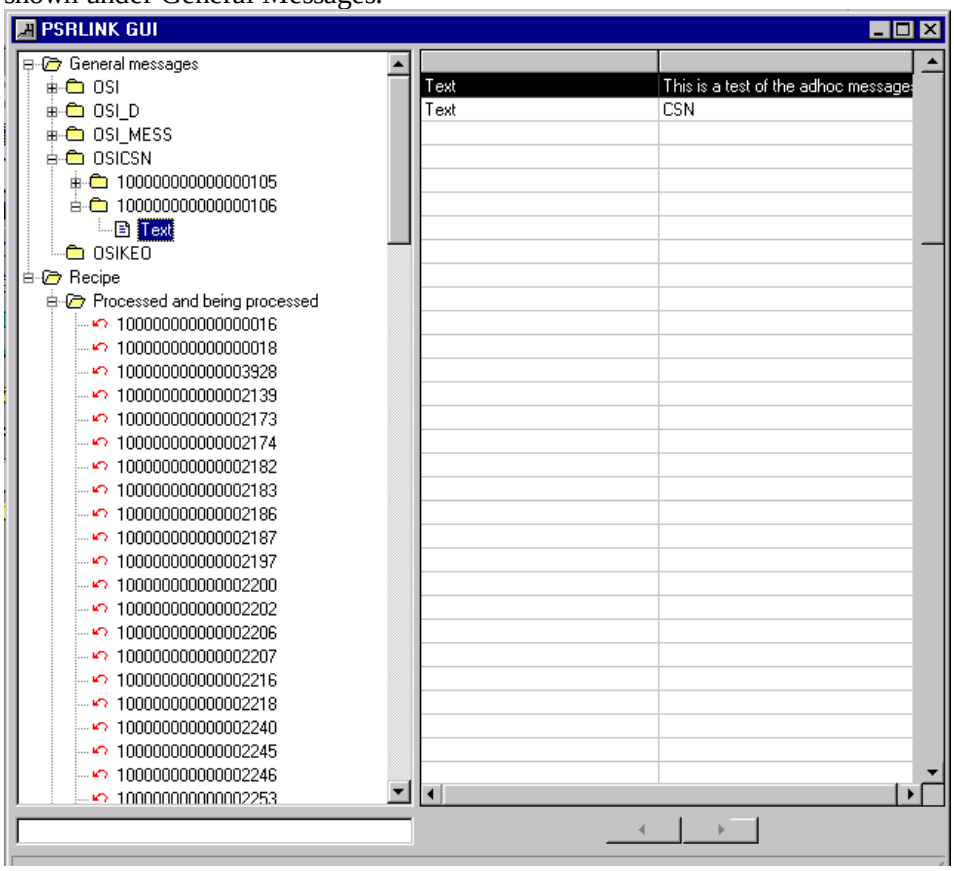
To see the detail of any entry, select it and then right click. If a message has multiple return values such as the phase status you can move between values using the arrows below the right hand portion of the dialog. Reviewing Status of Processing Process-Book



The icons in the PSRGUI application have been changed to reflect the status of the recipe. The icon meaning is N = new, S= Started, D = Discarded, T = Termnated, and C = Complete.

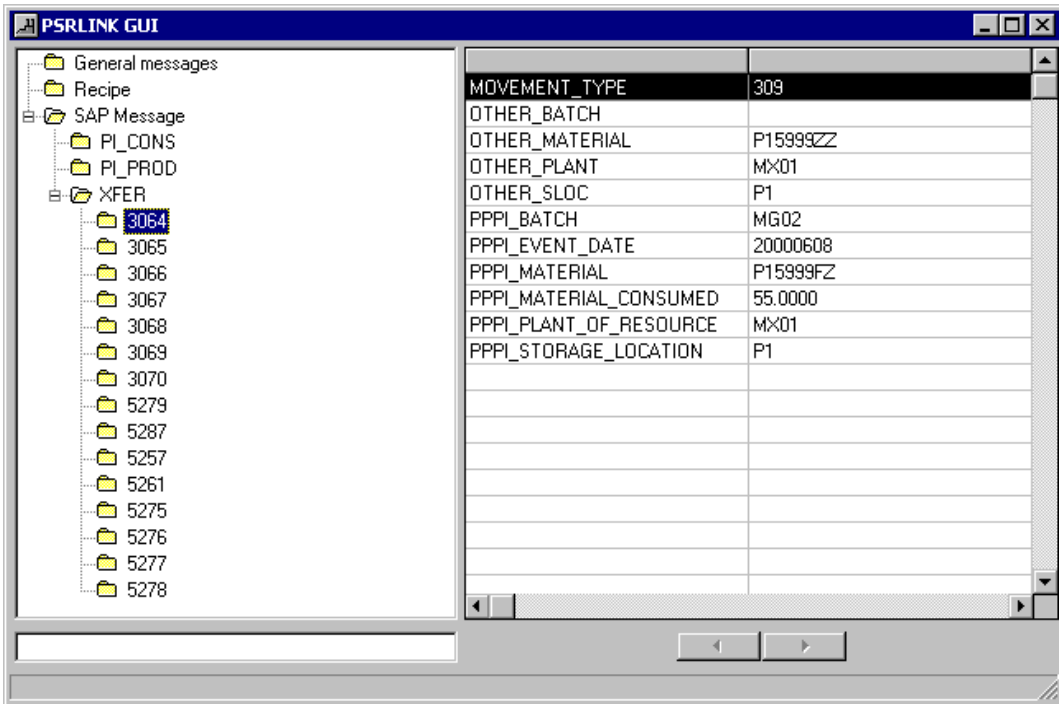
Gerneral Text Messages

General Text messages that are sent which cannot be related to any individual recipe are shown under General Messages.



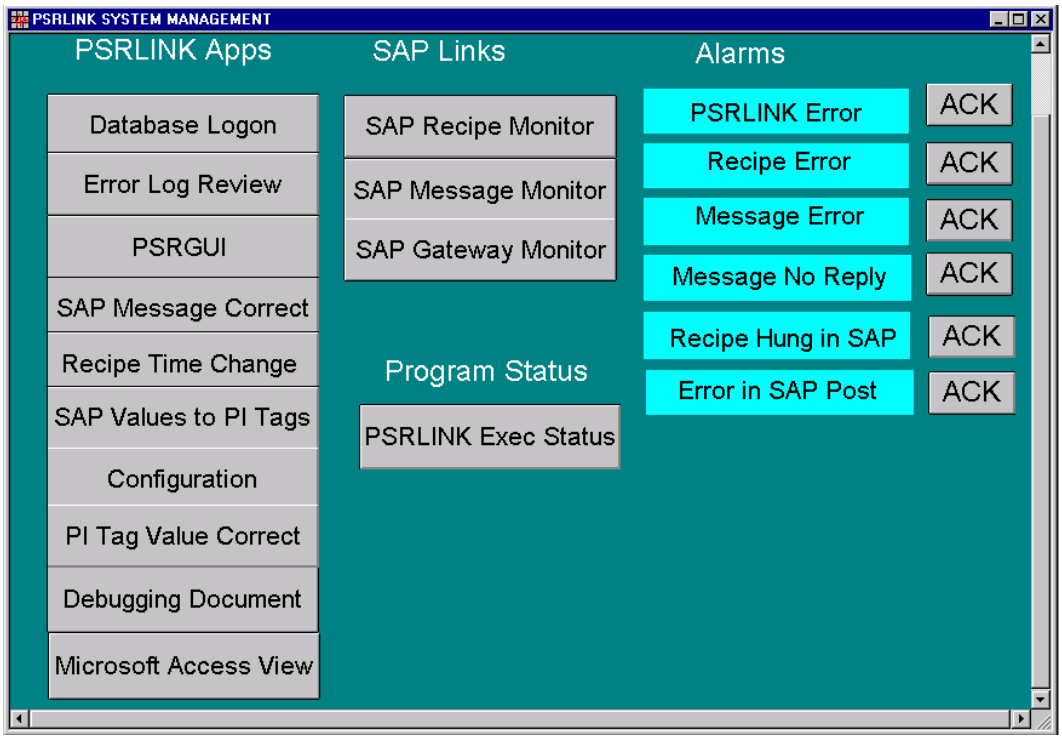
General SAP Transactions

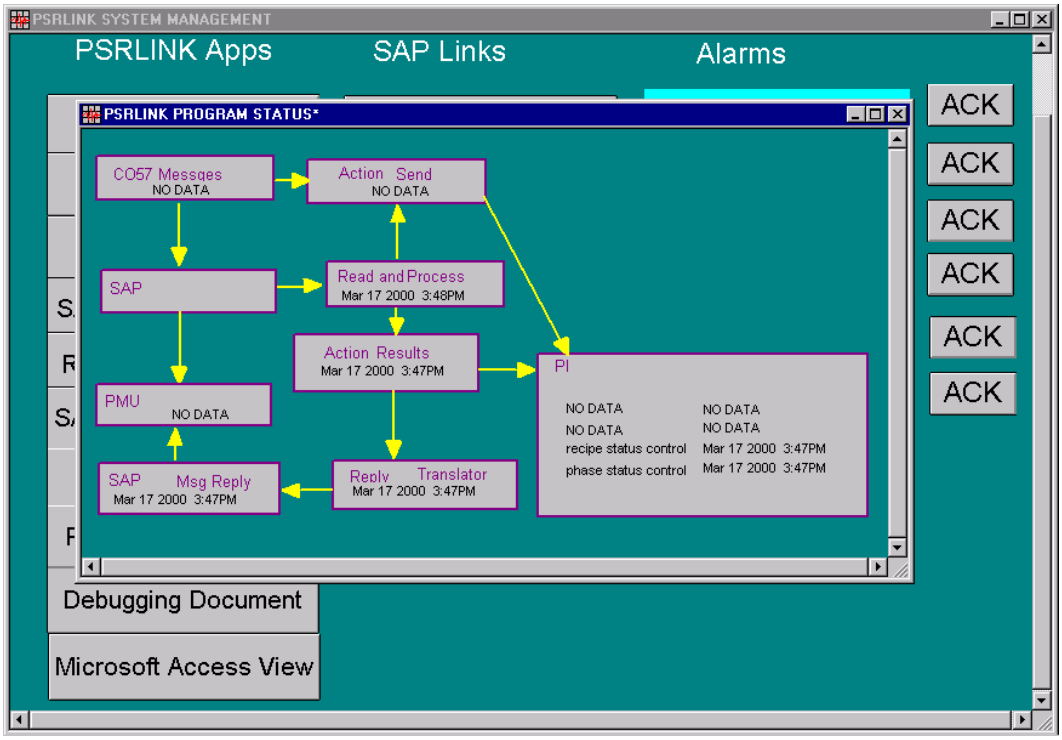
Messages that are created with the genral SAP Transaction can also be displayed in PSRGUI.



Monitor RLINK

A process-book display can be setup so that you can easily monitor that PSRLINK processes are running. A sample is shown here.





There is a point group of type ERROR with the following members

CRHE	- Errors have occurred in interpreting the recipe
MSHD	- SAP sent back an error message for a returned message
PMU_RCODE	- A message has gone to SAP but no RCODE received
PSRLINK	- An error message has been recorder in error_log
SAP_MSHD	- Used to indicate a message is hung in message monitor
SAP_RECIPE	- Used to indicate a recipe is hung in recipe monitor

The errors for SAP_MSHD and SAP_RECIPE require that a user exit be written on the SAP side to create a CO57 message with will set the value of the tag for this alarm.

The SAP LINKs require that a shortcut be set for SAP that will be attached behind these buttons to link to the appropriate screen in SAP.

The stored procedure usr_set_alarm is used to set the alarm values for the tags. If there messages in the error_log table that you do not want to alarm on then you should use the table exclude_list to enter the value in the error_log table status field that you want to exclude.

The ACK buttons will reset the tag using the application resetalr.exe. This application takes a parameter as follows

PSRLINK	- PSRLINK errors
CRHE	- Recipe errors
MSHD	- Message errors
SAP_RECIPE	- SAP recipe in the control recipe monitor
SAP_MSHD	- SAP message in the message monitor
PMU_RCODE	- Message RCODE return missing

Visual Basic Dialogs

In the following sections a number of Visual Basic Dialogs will be given for reviewing the recipe, instructions, messages etc. These are given so that they can be incorporated in ProcessBook applications. The code can also be furnished so that they can be customized. Our intent is to provide a reasonable starting set not meet all needs and specifics about how the recipe data will be reviewed. We have provided a general logon mechanism and template VB application that uses this mechanism.

Logon to Plant Suite Dialog

Purpose: This utility is provided to allow the user to logon on only once for Plantsuite applications. The logon information is shared across the applications until this application is closed. This application does not actually log onto the database, rather it gathers the information need for that logon and shares it between applications. When we move to Microsoft transaction server this functionality will be updated.

Executable Name: Logon

Stored Procedure Called: None

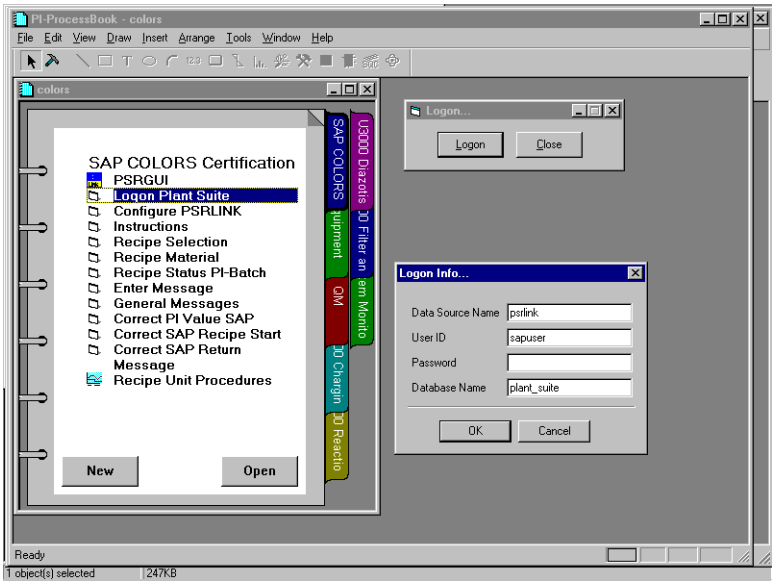
Controls Required: None

Function Keys or Mouse Key Actions:

Buttons: Logon – brings up the dialog for entering the SQL server machine, username, password and database that is set by default to Plant Suite.

Close will stop the sharing of the logon information between the Plant Suite dialog applications.

Tables Modified: None



Receiving Messages Dialog

Purpose:

The following Dialog is available for reviewing general messages that are sent from SAP/R3. The dialog is available as an executable and Active-X control for incorporation in ProcessBook.

Executable Name: Msg.exe

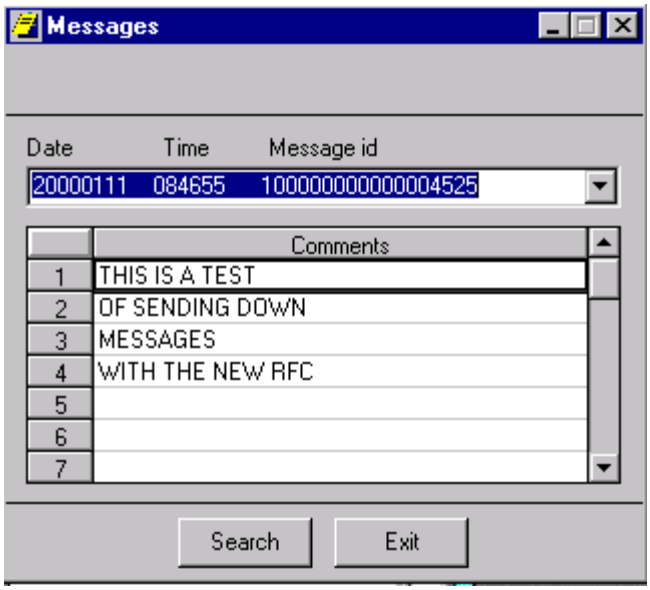
Stored Procedure Called: usr_general_msg_sel “L”

Controls Required: ss32x25.ocx

Function Keys or Mouse Key Actions:

Buttons:

Tables Modified: None



Search Messages Dialog

Purpose:

This dialog is used for selecting the message or range of messages to be reviewed.
Executable Name: None

Stored Procedure Called:

usr_general_msg_sel “CHECK” for checking

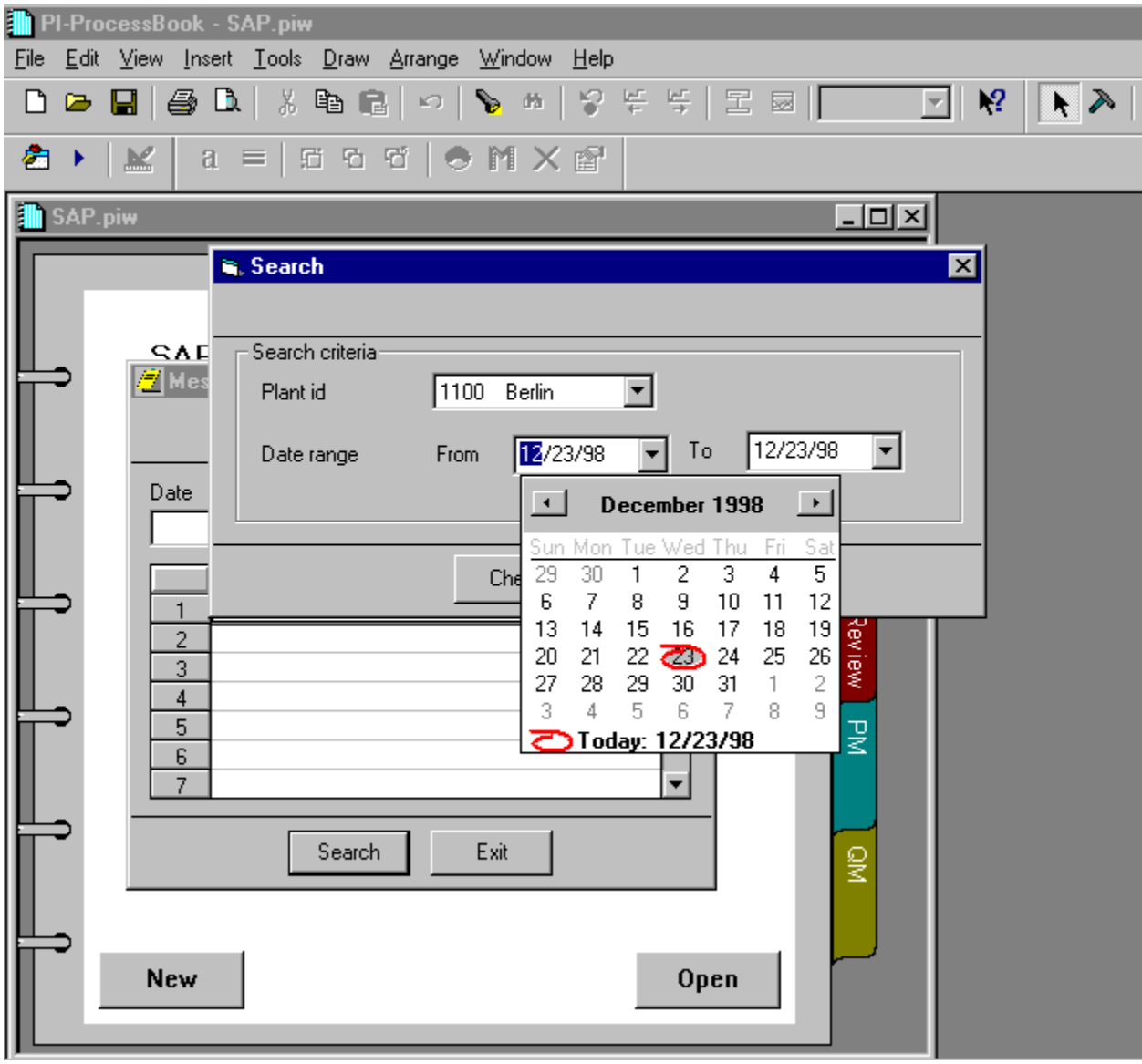
usr_general_msg_sel “GETVALLUE” to fill the combo box in message dialog

Controls Required: edt32x20.ocx

Function Keys or Mouse Key Actions: Date entry – right mouse click gives calendar

Buttons: Check selects the messages that satisfy the criteria

Tables Modified: None



Uploading Messages Dialog

Purpose:

This dialog is used for entering messages to be uploaded into SAP/R3. It is available as and executable and an ActiveX control.

Executable Name: MsgMak

Stored Procedure Called:

usr_make_msg is called for saving the message

usr_plant_all “R” is called to fill the combo box

usr_crhe_all “R” is called to fill PO combo box for the chosen plant

usr_general_rtr “MSGMK”, ‘1’ for filling phase combobox

Controls Required:

Function Keys or Mouse Key Actions:

Buttons: Save- saves the message in the database to be sent to SAP/R3 on the next scheduled upload of messages

Clear- clears the message that is being typed in

Tables Modified: MSHD, MSEL, UP_TLINES

Instructions for Recipe Dialog

Purpose:

This dialog is used to review instructions that have been sent down with the recipe. If a recipe has already be set with the Recipe selection dialog then this will go immediately to the instructions for that recipe.

Executable Name: Instruct.exe

Stored Procedure Called:

usr_recipe_all "GETVALUE" is used to fill Recipe combobox

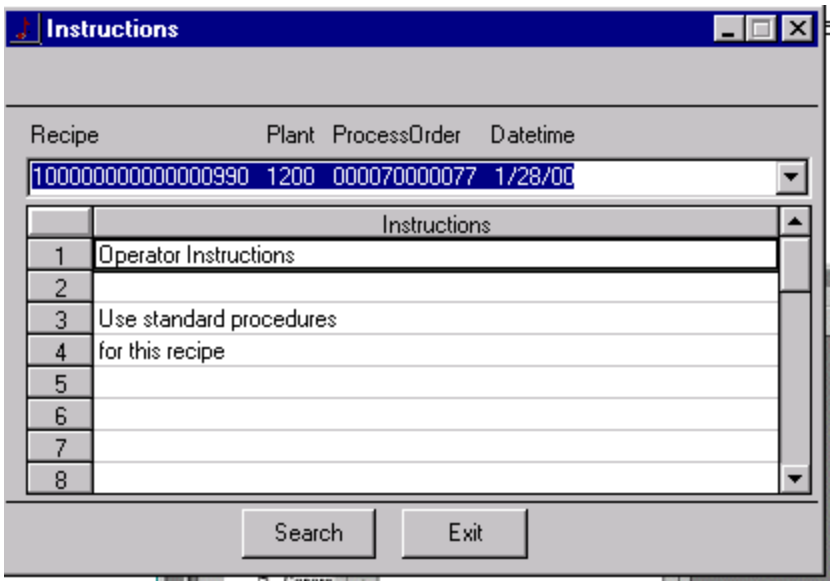
usr_tlines_all 'C' is used to fill the Instruction pdate i

Controls Required: ss32x25.ocx

Function Keys or Mouse Key Actions:

Buttons: Search switches to search for the recipe dialog

Tables Modified: None



Search for Recipe Dialog

Purpose:

This dialog is used for searching for the recipe or range of recipes to be examined in greater detail.

Executable Name: None

Stored Procedure Called:

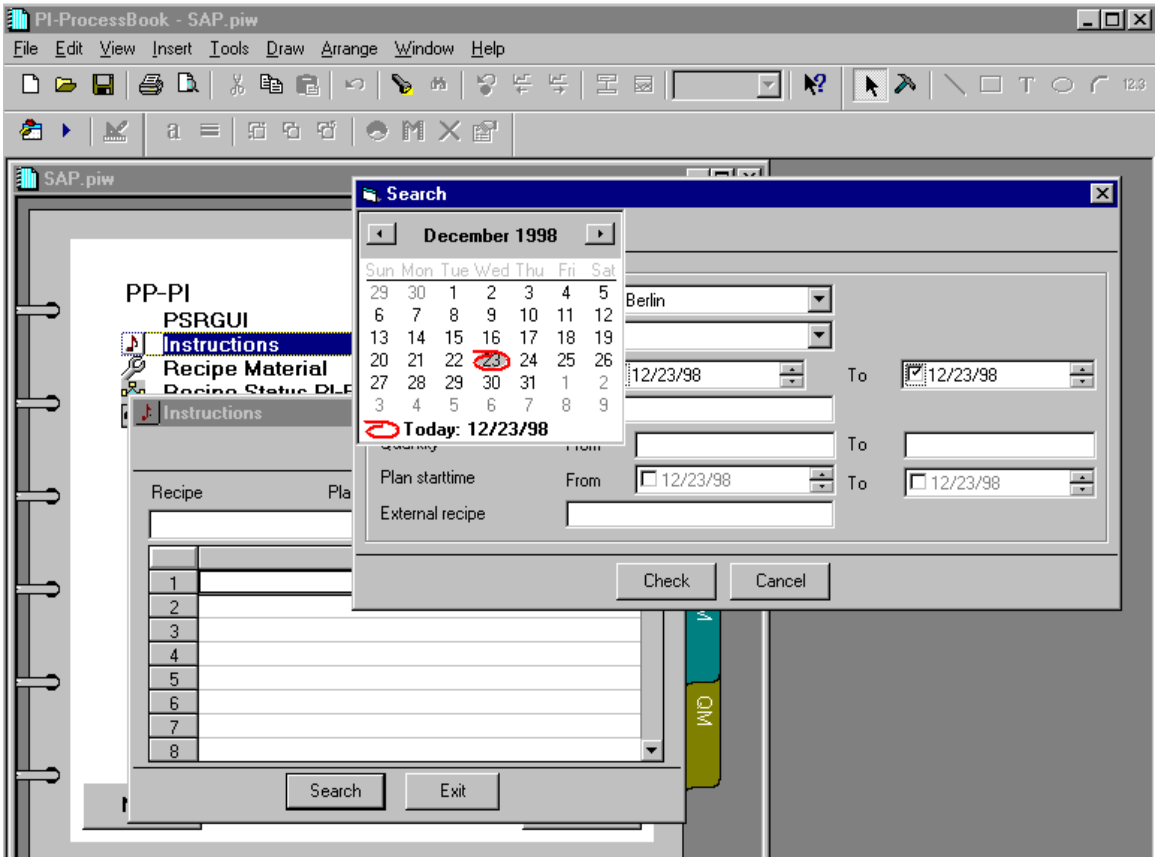
usr_recipe_all “CHECK” is used to check whether there is a result for the selected criteria. If “YES” then the same procedure with same criteria and “GETVALUE” is called to fill in the recipe combo box

Controls Required: edt32x20.ocx

Function Keys or Mouse Key Actions: Date entry – right mouse click gives calendar, to activate the date you must mark check box.

Buttons: Check searches for the recipes that meet the selected criteria

Tables Modified: None



Material Dialog

Purpose:

This dialog is used to retrieve the material list for a given recipe. If a recipe has already been selected with the recipe selection dialog this dialog will show immediately the material for that recipe.

Executable Name: Material.exe

Stored Procedure Called:

usr_recipe_all “GETVALUE” is called to fill in the Combo box

usr_phase_sel “K” is called to fill in the phase combobox

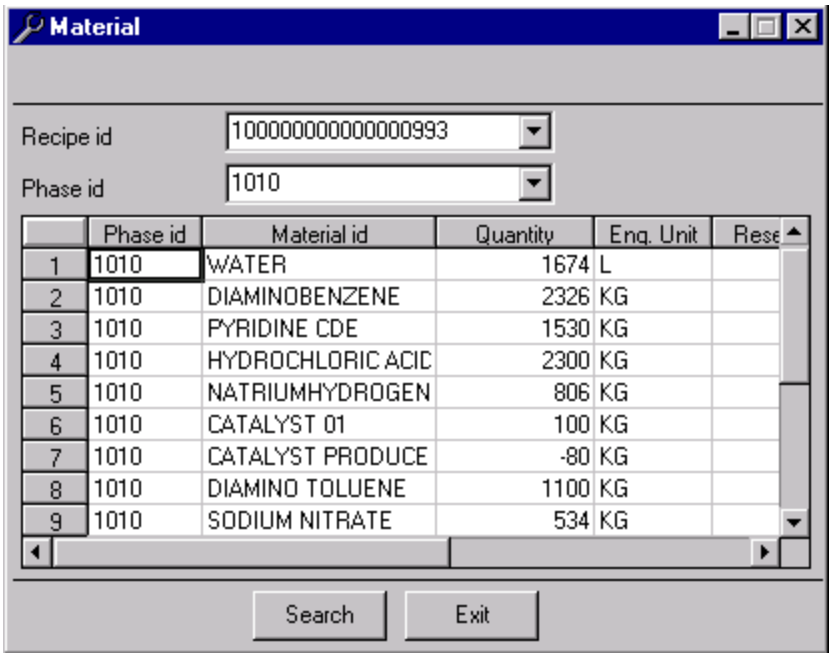
usr_mat_list_sel “L” is called to fill in the materials

Controls Required: ss32x25.ocx

Function Keys or Mouse Key Actions:

Buttons: Search switches to the recipe search dialog

Tables Modified: None



Selecting Recipe to Review Dialog

Purpose:

This dialog is used to select a recipe that will be reviewed in the ProcessBook displays and the other dialogs without having to select multiple times the recipe of interest. It updates a table in the database called Display that allows one active recipe to be set per user. The clear option on this dialog removes the currently set recipe for this user. This application is available as an executable and as an ActiveX control.

Executable Name: Recipe.exe

Stored Procedure Called:

usr_recipe_all “GETVALUE” is called to fill the recipe Combo box

usr_display_all “SET” is called to set recipe_id in the Display table

usr_crfv_all “r” is called to fill the information about the recipe

Controls Required: ss32x25.ocx

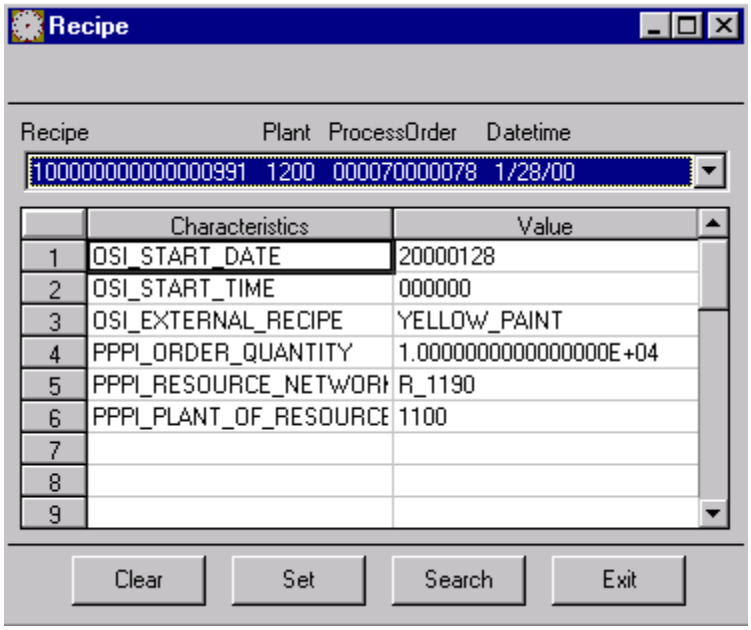
Function Keys or Mouse Key Actions:

Buttons: Search – switches to the recipe search dialog

Set- will set the Display table for the chose recipe

Clear- will clear the Display table for the current user

Tables Modified: DISPLAY



Setting Status of Recipe Dialog

Purpose: Application will set the PI status points for starting and stopping of a recipe and phase and change the resource for a phase by selecting the new resource.

Executable Name: Setpibatch.exe

Stored Procedure Called:

usr_recipe_all “PIBATCH” is called to fill recipe Combo box

usr_phase_all “A” is called to fill phase_ids in phase combo box for the selected recipe

usr_general_rtr “PIBATCH” is called to check for the given recipe whether all the phases are completed when the recipe status is 00005

usr_acton_send_set “R” is used to set the recipe status

usr_action_send_set “P” is used to set phase status.

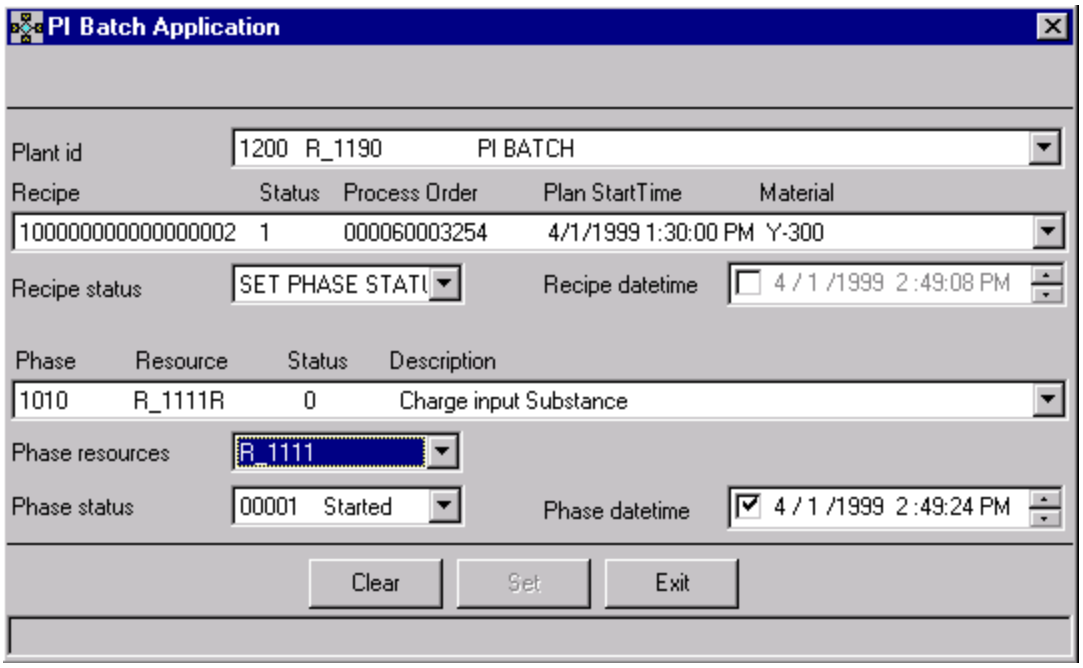
Controls Required: edt32x20.ocx

Function Keys or Mouse Key Actions: Activate the date by marking the checkbox

Buttons: Set will cause the status to be set and the record to be written in Action_send followed by the call to put the values in PI and retrieve the data from PI to assure that it has been entered. The programs putvalue.exe, phsctrl.exe and stsctrl.exe are called

Clear will clear the dialog

Tables Modified: Action_Send



SAP/R3 Message Correction Dialog

Purpose:

This dialog is used to correct messages that have been sent to SAP/R3 and are returned in error. The dialog will allow the user to correct a problem with the message and re-send or re-send the message after a correction has been made in the SAP/R3 system. After changes are made the message must be saved. Even if changes are not required in the PSRLINK side the SAVE button must be executed to reset the status of the message so that it can be sent again. The Send option will send the message immediately.

Executable Name: Msgcr.exe

Stored Procedure Called:

Usr_msg_correct ‘E’ to retrieve error messages

Usr_msg_correct ‘L’ to retrieve MSEL

Usr_msg_correct ‘M’ to retrieve MSHD

Usr_MSEL_U2 “MSEL” to update MSEL

Usr_MSEL_U2 “MSHD” to update MSHD

Usr_plant_all

Controls Required: ss32x25.ocx

Function Keys or Mouse Key Actions:

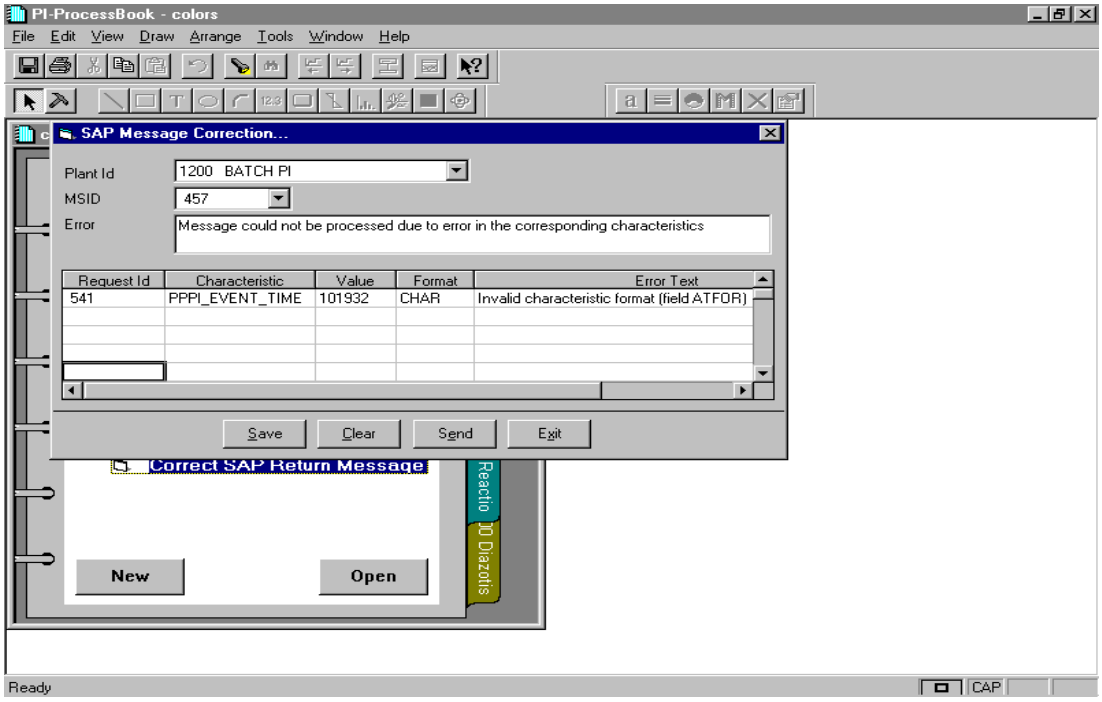
Buttons: Save- resets the status of the records in MSHD and MSEL and makes the corrections in the records

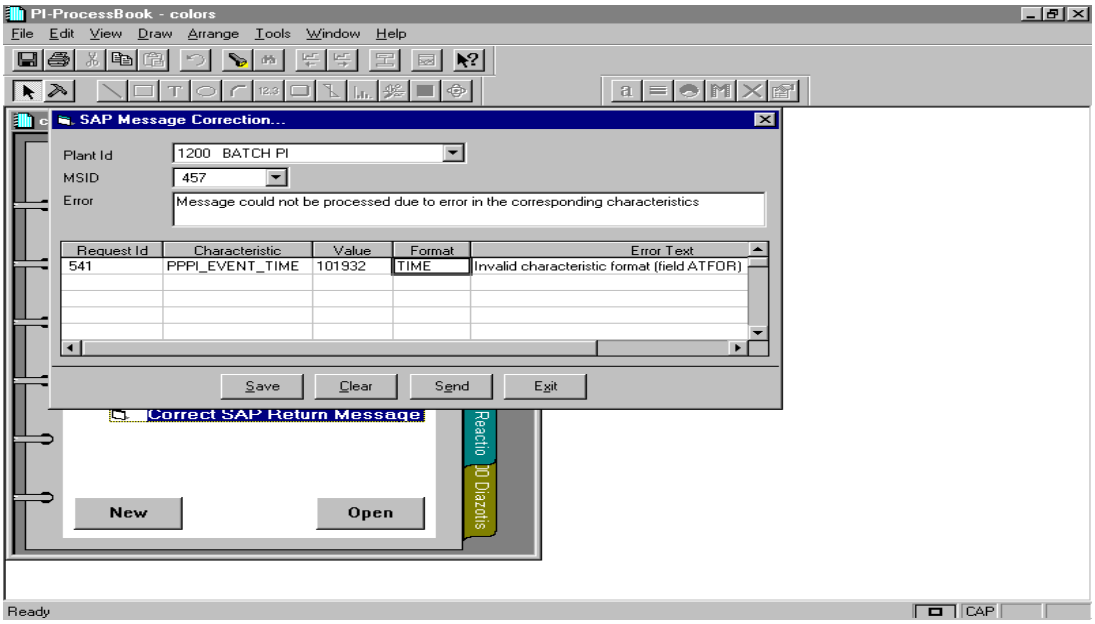
Clear-erases the current data with no changes made

Send-calls the PMU executable to upload the messages immediately

Exit-exits the application

Tables Modified: MSHD, MSEL





The table that holds the error codes for SAP messages is error_message

application_no	meaning	rcode_text	used_by
1	No error discovered during characteristic check	0	MSEL
2	Message processed correctly	0	MSHD
1	Characteristic is not created(field ATNAM)	1	MSEL

application_no	meaning	rcode_text	used_by
2	Plant does not exist(field WERK in table MSHD)	1	MSHD
1	Invalid characteristic format (field ATFOR)	2	MSEL
2	Message category not created in the plant(field MSCLA in table MSHD)	2	MSHD
1	According to the characteristic definition in PP-PI, the characteristic value should be a long text. However, the table TLINES does not contain a corresponding entry.	3	MSEL
2	Invalid test indicator (field TSTKZ in table MSHD)	3	MSHD
1	No value assigned to characteristic. This is not allowed according to the characteristic definition within PP-PI	4	MSEL
2	The system does not have the authorization to create process messages in the specified plant	4	MSHD
2	Message could not be processed due to error in the corresponding characteristics	99	MSHD
4	Control recipe does not exist	CONTROL_RECIPE_NOT_FOUND	CRA
4	Control recipe status does not permit download	CONTROL_RECIPE_STATUS_NOT_VALID	CRA
3	Address not valid for this type of communication	DESTINATION_NOT_VALID	CRP
4	Addres not known	DESTINATION_NOT_VALID	CRA
3	Address not known	DESTINATION_UNKNOWN	CRP
5	Internal error	INTERNAL_ERROR	PMU
3	System error	SYSTEM_FAILURE	CRP
4	System error	SYSTEM_FAILURE	CRA
3	Error when editing control recipe texts	TEXT_WORK_UP_FAILURE	CRP
4	Error when editing control recipe texts	TEXT_WORK_UP_FAILURE	CRA

Adjusting the Recipe Start Time Dialog

Purpose:

This dialog is used to reset the window for searching for the start and end of a recipe in PI. It would be used in the case that SAP/R3 was down for an extended period of time and plant operation continued. In this case when the recipe is created after SAP/R3 comes back up the recipe start time is after when the recipe actually executed. In this case the window is readjusted back by this dialog so that the actual time of execution can be found.

Executable Name: Setaxr.exe

Stored Procedures Called:

usr_axr_sel "SPECIFIC" picks up a particular recipe record from action_results with trigger_proc "control_monitor"

Usr_axr_sel "UPDATE" updates action_results where trigger_proc is control_monitor" or "Phase_monitor"

Usr_axr_sel "ALL" picks up field1 of action_results whre trigger_proc = "control_monitor"

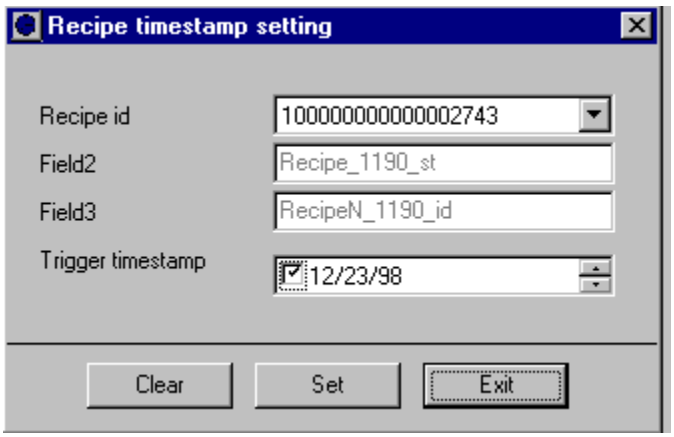
Controls Required: edt32x20.ocx

Function Keys or Mouse Key Actions: To activate the time you must mark the checkbox

Buttons: Clear-clears the current screen making no changes

Set will change the starting time range for looking for the recipe results to the time given in Trigger Timestamp

Tables Modified: Action_Results



Correcting Failed Result Collection Dialog

Purpose:

This screen is used to enter values for results that failed to find a value. Reasons for such failure could be that the point could not be found in PI or that the result in PI was not a valid value. The changes here will not be made in PI but only in the local tables and the fact that a change has been made will be made in the audit tables.

Executable Name: Jchange.exe

Stored Procedures Called:

usr_ar_sel "CLOSERPD" updates action_results status to "S" and calls usr_ad_I

usr_ar_sel "SPIN" retrieves records from action_results with status "F"

Usr_ar_sel "UPDATE" inserts records into action_result_values table, puts record into audit_data by calling usr_ad_I

Usr_ar_sel "RETRIEVE" retrieves the failed frcords from action_results

Usr_ar_sel "SELARV" selects a specific record afrom action_results_ pdate for the given request_part_id

Usr_ar_sel "DELARV" Delets a specific record from action result_values for the given request_part_id, order_no

Usr_ad_i inserts record into audit_data

Controls Required: edt32x20.ocx

Function Keys or Mouse Key Actions:

Buttons: Set- The entered value with be set in Action_result_values

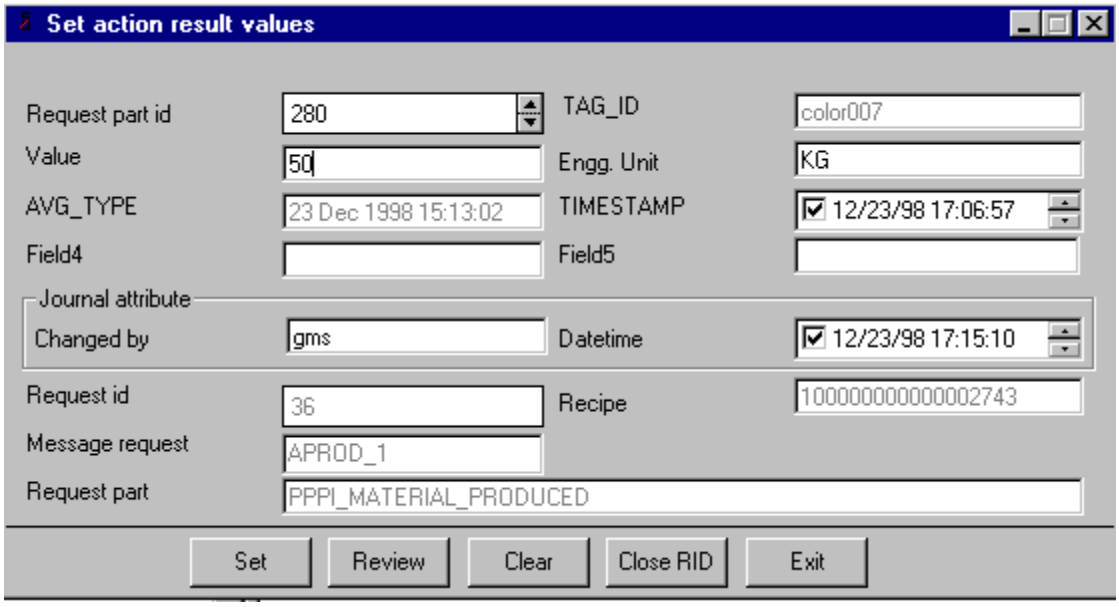
Review-Allows the user to review what values exist in Action_result_values. From the review screen values can be marked for deletion

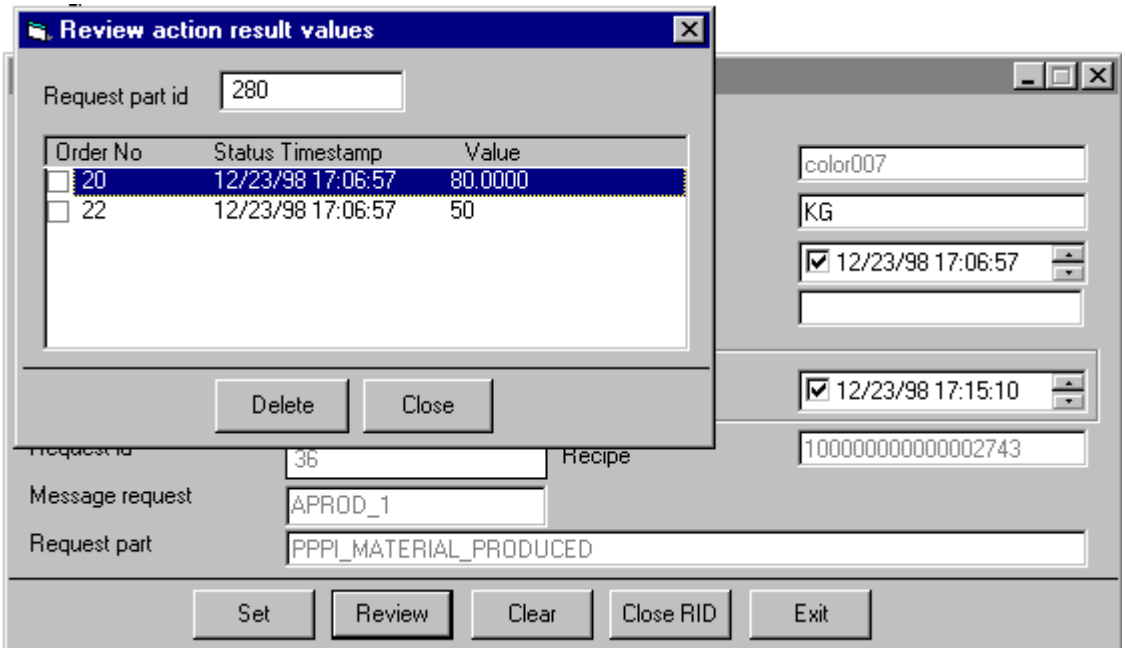
Clear-Clears all entries making no changes

Close RID- Closes the request ID and no further values can be enterd. The status will be marked as 'S' and regular processing of the data will progress.

Exit –exits the application

Tables Modified: Action_Result, Action_Result_values, Audit_data





Error Log Review Dialog

Purpose:

The display is used to review the error messages that have occurred and are logged in the error_log table. If you want to search for all errors from a given date enter clear then enter the date and scroll forward.

Executable Name: Trend.exe

Stored Procedure Called

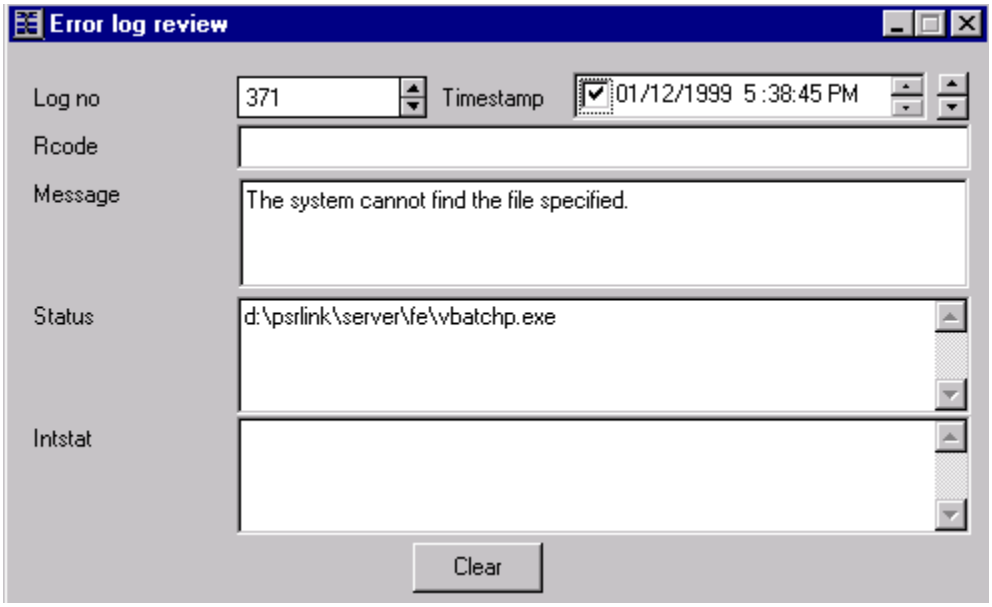
Controls Required: None

Function Keys or Mouse Key Actions:

Buttons: If you spin the button assigned to the Log No the application will loop around the log number. If you check the timestamp box and select a date and spin the buttons associated with the timestamp then you will loop around the date.

Tables Modified: None

The table that holds the error log is error_log.



Adjusting a Trend Time in ProcessBook

Purpose:

The Trend application is provided to readjust the start and end times of the current active trend display in the open ProcessBook for the start and end times of the recipe and operation that is being reviewed. This routine sets the start time to be the start of the recipe and the end time to be the end of the recipe for the entire display.

Executable Name: Trend.exe

Stored Procedure Called:

usr_display_all "Time" is used to get start and end time for the recipe is is set in the ProcessBook

Controls Required: None

Function Keys or Mouse Key Actions:

Buttons:

Tables Modified: None

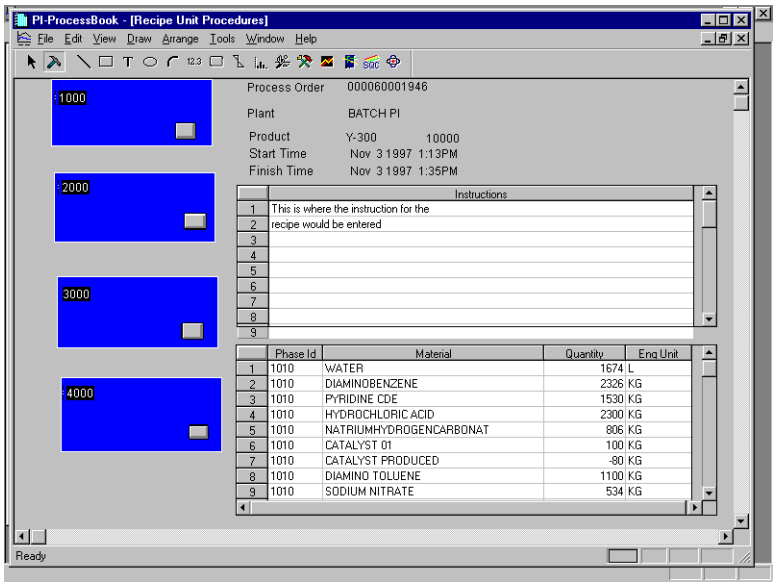
Using the Plant Suite Logon In Your Applications

As mentioned in the section on the Logon Dialog the information can be shared across multiple applications. The module that handles the communication to be included in your application is ModComn.bas and a starting project is given in PS_Sample.

Process Book Review of Recipe Using ODBC DataSets

The following illustrates how you can use ProcessBook to review the details of a recipe that was executed. The ODBC data sets that were used to create these displays are shown below. This ProcessBook has been built for ProcessBook version 2.0.

Recipe data set



select recipe.status_id,recipe.recipe_id,plant.plant_description,

recipe.process_order,recipe.material_id,

recipe.total_quantity,

recipe.resource_network,

recipe.master_recipe_name,

start_time = convert(char(24),recipe.start_time),

finish_time = convert(char(24),recipe.finish_time)

from recipe,

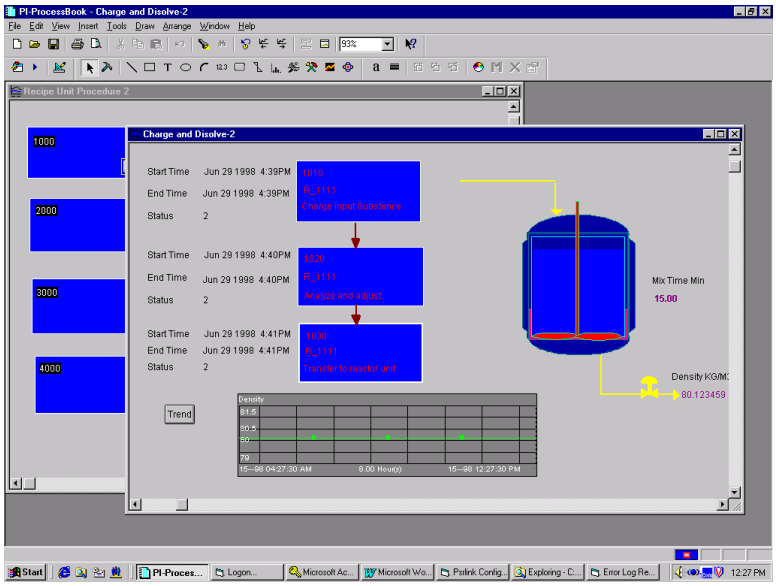
display,plant

where display.display_type = ‘RECIPE’

and recipe.recipe_id = display.criteria1

and plant.plant_id = recipe.plant_id

and display.host_name=host_name()



On Open Function in ProcessBook

The following procedure is used for the opening of the Processbook the value passed to the procedure execute usr_phase is the phase number for that operation.

```
Private Sub Display_Open()
Dim szQry          As String
   gConnected = False
   gszHostName = gFnGetHostName
   DoConnect
   szQry = "execute usr_phase 1"
   Set rdoRs = rdoCn.OpenResultset(szQry, rdOpenForwardOnly, rdConcurReadOnly)
  End Sub
```

Block Query

The ID is set to the number of the phase

```
select phase_id= col1, phase_desc=RTRIM(col2),resource=RTRIM(col3),
start_time=RTRIM(col4),end_time=RTRIM(col5),status=RTRIM(col6),status_time=RT
RIM(col7)
from    pi_process_book
where id          = 1
and host_name= host_name()
```

Program Monitor Type ODBCDataSet

```
select date = convert(char(22),last_exec_dtime,100),eb.functionality
from exec_batch eb,
```

group_master gm

where eb.group_no = gm.group_no

and eb.program_name = ‘c:\psrlink\fe\phsctrl.exe’

Unit operation first block

select operation_id,operation_description

from operation,

display

where display.display_type = ‘RECIPE’

and operation.recipe_id = display.criteria1

and operation.operation_id = display.criteria2

Unit Operation other blocks

select operation_id,operation_description

from operation,

display

where display.display_type = ‘RECIPE’

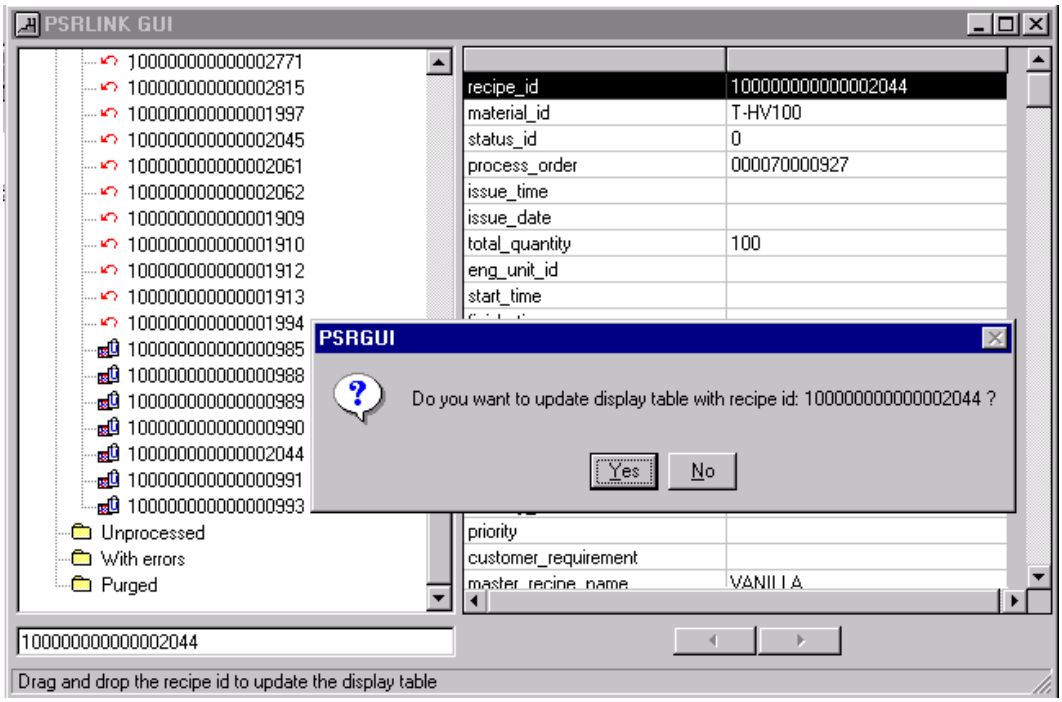
and operation.recipe_id = display.criteria1

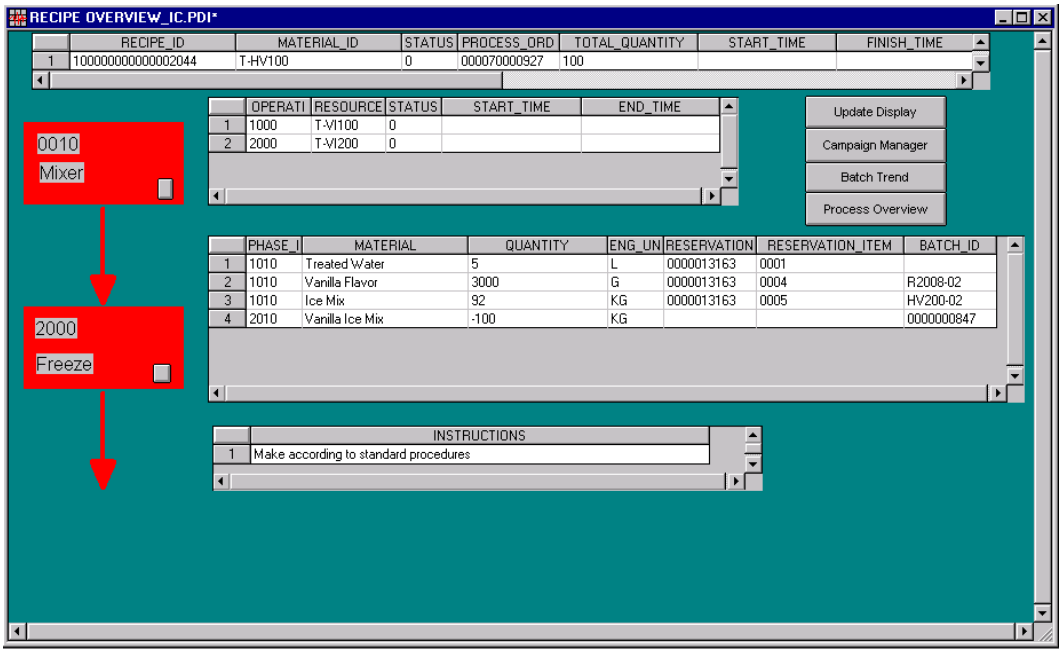
and operation.operation_id = display.criteria3

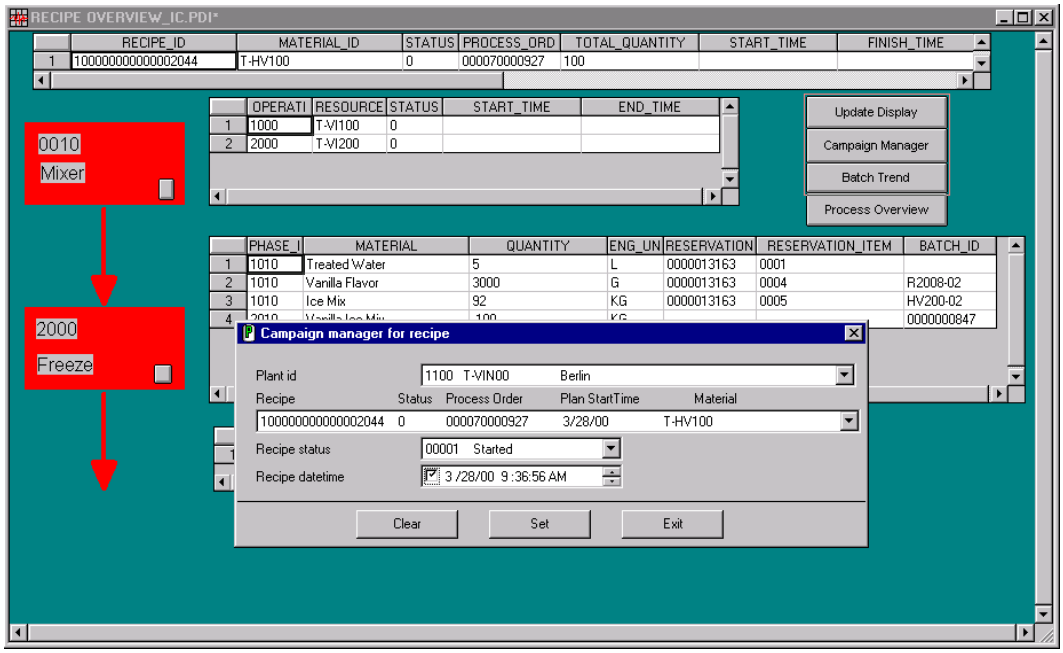
Campaign Manager

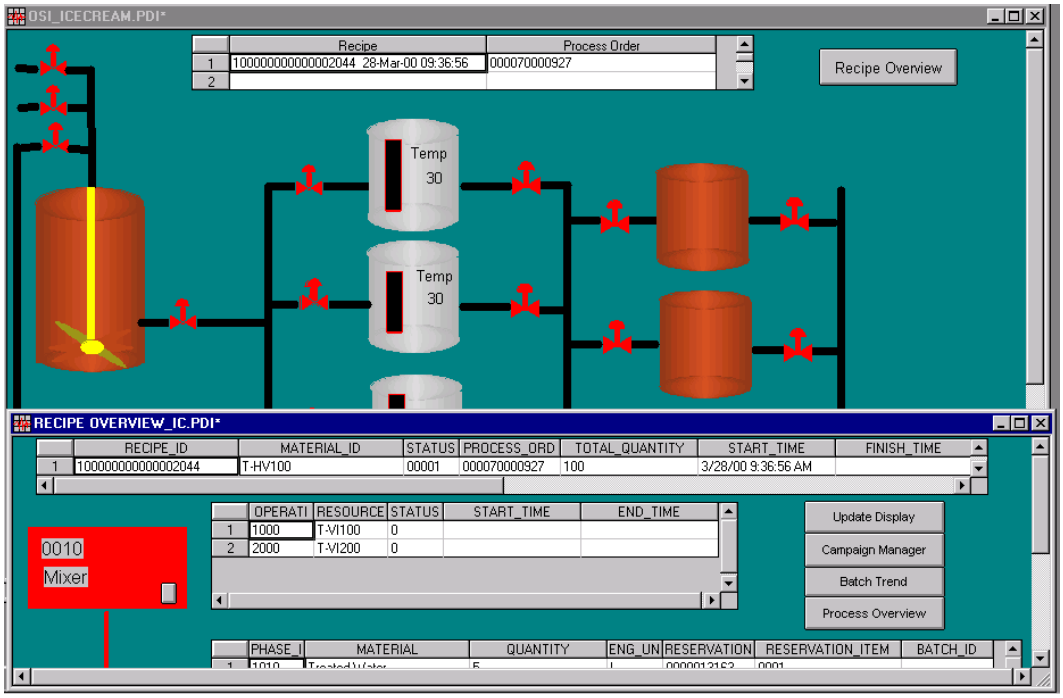
You can use ProcessBook as a campaign manager. We have provided applications which allow you to set the status of the recipe and the phase. These programs are CMRCP.exe and CMPHS.exe respectively.

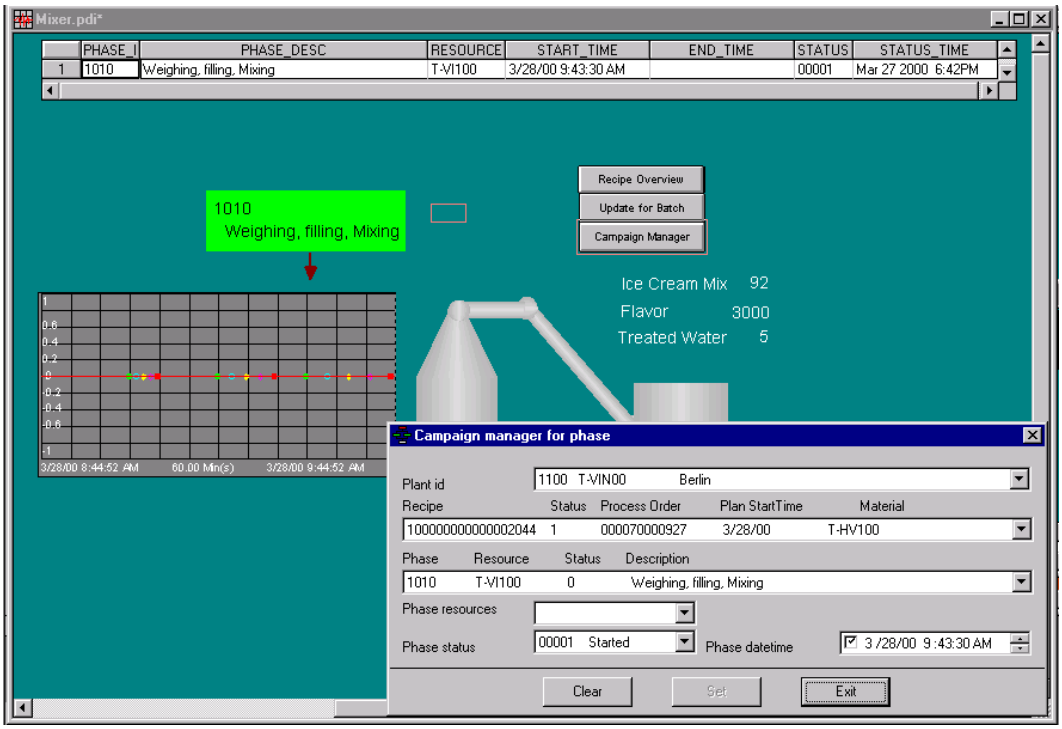
First you use PSRGUI to select the recipe that is to be reviewed in campaign manager by dragging the recipe number down to the bottom left. This will set the active recipe in the table display. This example uses the ietimer.ocx from Microsoft IE4 service pack 4, the Microsoft mshflxgd.ocx for the grid control and Microsoft msadodc.ocx for database access. The ProcessBooks are provided so you can see the code which is required. The share2.pdi is password protected and is attached as a reference to the pdi file you are creating.

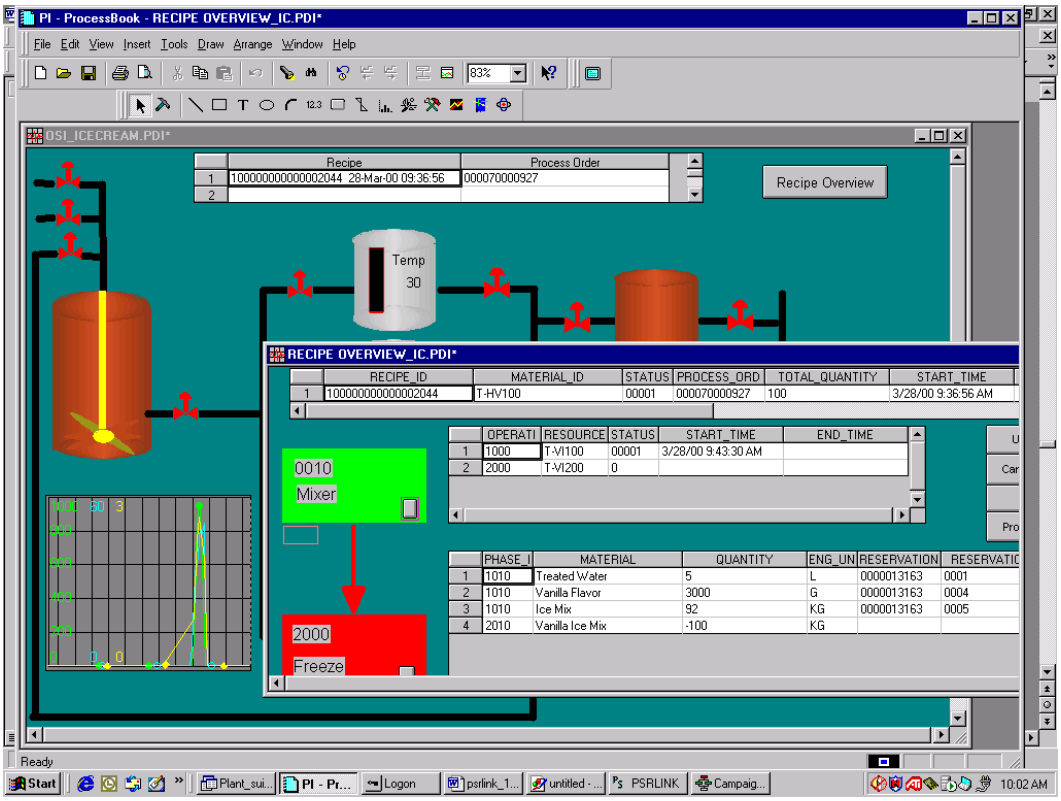

PSRLINK GUI
100000000000002771
100000000000002815
100000000000001997
100000000000002045
100000000000002061
100000000000002062
100000000000001909
100000000000001910
100000000000001912
100000000000001913
100000000000001994
100000000000000985
100000000000000988
100000000000000989
100000000000000990
100000000000002044
100000000000000991
100000000000000993
Unprocessed
With errors
Purged
recipe_id
100000000000002044
material_id
T-HV100
status_id
0
process_order
000070000927
issue_time
issue_date
total_quantity
100
eng_unit_id
start_time
priority
customer_requirement
master_recipe_name
VANILLA
PSRGUI
Do you want to update display table with recipe id: 100000000000002044 ?
Yes
No
100000000000002044
Drag and drop the recipe id to update the display table


RECIPE OVERVIEW_IC.PDI*
RECIPE_ID
MATERIAL_ID
STATUS
PROCESS_ORD
TOTAL_QUANTITY
START_TIME
FINISH_TIME
100000000000002044
T-HV100
0
000070000927
100
OPERATI
RESOURCE
STATUS
START_TIME
END_TIME
1000
T-VI100
0
2000
T-VI200
0
Update Display
Campaign Manager
Batch Trend
Process Overview
0010
Mixer
2000
Freeze
PHASE_I
MATERIAL
QUANTITY
ENG_UN
RESERVATION
RESERVATION_ITEM
BATCH_ID
1010
Treated Water
5
L
0000013163
0001
1010
Vanilla Flavor
3000
G
0000013163
0004
R2008-02
1010
Ice Mix
92
KG
0000013163
0005
HV200-02
2010
Vanilla Ice Mix
-100
KG
0000000847
INSTRUCTIONS
Make according to standard procedures


RECIPE OVERVIEW_IC.PDI*
RECIPE_ID
MATERIAL_ID
STATUS
PROCESS_ORD
TOTAL_QUANTITY
START_TIME
FINISH_TIME
100000000000002044
T-HV100
0
000070000927
100
OPERATI
RESOURCE
END_TIME
1000
T-VI100
2000
T-VI200
0010
Mixer
2000
Freeze
Update Display
Campaign Manager
Batch Trend
Process Overview
PHASE_I
MATERIAL
QUANTITY
ENG_UN
RESERVATION
RESERVATION_ITEM
BATCH_ID
1010
Treated Water
5
L
0000013163
0001
Vanilla Flavor
3000
G
0004
R2008-02
Ice Mix
92
KG
0005
HV200-02
0000000847
Campaign manager for recipe
Plant id
1100 T-VIN00 Berlin
Recipe
Status
Process Order
Plan StartTime
Material
3/28/00
Recipe status
00001 Started
Recipe datetime
3 /28/00 9 :36:56 AM
Clear
Set
Exit


OSI_ICECREAM.PDI*
Recipe
Process Order
100000000000002044 28-Mar-00 09:36:56
000070000927
Recipe Overview
Temp
30
Temp
30
RECIPE OVERVIEW_IC.PDI*
RECIPE_ID
MATERIAL_ID
STATUS
PROCESS_ORD
TOTAL_QUANTITY
START_TIME
FINISH_TIME
100000000000002044
T-HV100
00001
000070000927
100
3/28/00 9:36:56 AM
OPERATI
RESOURCE
END_TIME
1000
T-VI100
2000
T-VI200
0010
Mixer
Update Display
Campaign Manager
Batch Trend
Process Overview
PHASE_I
MATERIAL
QUANTITY
ENG_UN
RESERVATION
RESERVATION_ITEM
BATCH_ID





For each of these displays there is attached a reference file called share2.pdi in which we ship the code for doing the database access for this application. This file is password protected. Using reference files like this is a method to have common code shared between ProcessBooks.

Chapter 9
Customization

Adding a New Instruction

1. Construct new instruction in SAP/R3
2. Write new procedure that will know how to translate the Instruction Characteristics into requests for data. You can use the sample for AREAD2 given in the installation as an example.
3. Make entry in the Procedure_table for your new instruction.
4. Set up a translation method for each Instruction characteristic that must be found. You can use the sample for usr_get_alias_tag.
5. If there is a new application for retrieval of the data this must be written which can be done similar to the gettag application. Update the application table for the new application. Add this new application that must run to the group_master and exec_batch tables.
6. Update the Translation table for each new characteristic in the new Instruction that must be retrieved.
7. Update the Instruction_characteristic table for what are the required characteristics in the Instruction to be used by the recipe checking system.

Adding a New Characteristic

In the material_tag table there have been added 4 miscellaneous tags that can be used to support a characteristic associated with a material. These 4 tags also appear on the material tag dialog in the configure application. You can add a translation method for your new characteristic that uses one of these fields. In the future we will write a general procedure to support this automatically.

Adding a New Destination Interface

(To be completed with a later version)

Adding a New Source Interface

If data is to be retrieved for a source other than PI then the following must be implemented.

I new program executable would be written which would retrieve the data from the new source. The input to this program are field1-field5 in Action_results along with the timestamp. This program is written per the example for getting a PI Tag. It first executes the query on the application table if there are any requests waiting for this

application which passes the outstanding request. It then uses the information that is passed to request the data of the new source and then it formats the reply as illustrated in the PI Tag example to insert the reply.

This new application must be entered into the application table, and setup for execution in the group_master and exec_batch tables. If it is to be used as the default application for one of the characteristics then the translation table should be changed to the new application.

There will probably be a new translation method to setup the input to the new program using the fields in Action_result. The translation method could make use of the tables Common_name or material_tag or a new table that the user creates. The result of the translation method must return the fields similar to what is given in the sample procedure get_alias_tag. The new translation procedure should be stored as a stored procedure in the database and the translation table updated.

Message Comment Interface

If you want to build an interface for sending text comments to SAP/R3 then the Tables that must be inserted into are as follows. This interface could be used to monitor the error log table and send messages to SAP.

MSHD

Table Field	***Meaning***
MSID	automatically assigned when insert is made in this table
request_id	null
order_no	null
WERK	Plant
MSCLA	PI_COMM
TSTKZ	test flag set to X for test
SEDAT	date in format YYYYMMDD
SEUZT	time in format HHMMSS
Source	name of DEST system sending data

MSEL

This table is used to supply the characteristics of the PI_COMM instruction. There should be an entry for each of the following

ATNAM	ATFOR
PPPI_EVENT_TIME	TIME
PPPI_EVENT_DATE	DATE
PPPI_MESSAGE_TEXT	CHAR
PPPI_OPERATION	CHAR
PPPI_PHASE	CHAR
PPPI_PROCESS_ORDER	CHAR

PPPI_SOURCE CHAR

Table Field	Meaning
MSID	value from MSHD table
request_id	null
Order_no	null
ATNAM	Enter ATNAM as given above
ATWRT	char 30 field with value
ATFOR	Enter ATFOR as given above

UP_TLINES

Table Field	Meaning
Line_no	automatic assignment of line number on insert
MSID	value from MSHD table
ATNAM	PPPI_MESSAGE_TEXT
TDFORMAT	*
TDLINE	Enter text up to char132

We provide you an application to insert new messages.

User Exit PPPI_EXTERNAL_PHASE

If you require on SAP to set the name of the PPPI_EXTERNAL_PHASE by a user exit program and can not use the standard instruction characteristic PPPI_EXTERNAL_PHASE then a method has been provided for you to alias this instruction characteristic.

The tables you must modify are alias_system and external_alias. In alias system there is an entry with the alias_system_description of SAP PP_PI and an alais_system of SAP. You must make a similar entry for your plant. In the table external_alias you make an entry for your new alias_system with the alias_value being the name of the new characteristic and internal_value is assigned PPPI_EXTERNAL_PHASE and the alias class is SAP-PPPI. There is a sample entry created in the standard load for plant 1100 for you to use for comparison. If no alias system and external_alias is created the system will use the standard instruction PPPI_EXTERNAL_PHASE.

Table Modification

A table follows which summarizes for each type of modification the tables that must be modified.

Translation_method- this table would only be modified if you were adding you own translation methods.

Table Field	***Meaning***

Table Field	*Meaning*
Name	Translation method name
Description	Translation method description

Instruction_category (Only modified if adding your own instruction)

Table Field	*Meaning*
Category	SAP/R3 Instruction Name
Category_description	SAP/R3 Category description
Category_load_method	Used if partial results should be returned for a continuous process against one process order. This is the only field which needs to updated with the value partial if a partial result at fixed increments is to be returned.
Type_id	Type of instruction 1=process parameter, 3= subscription, 2=process data request

Application (This table is changed only if a new interface is being added)

Table Field	*Meaning*
Application_no	Number of application
Application_description	Description of application
Program_name	Name of program to be executed to satisfy the data query
Field_name 1-7	Data input fields required by the application
LIST_NAME 1-4	List structure if data input requires

Instruction_characteristics - This table is used to do error checking on the recipe sent down to assure that the correct characteristics were sent and that there are no duplicates.

Table Field	*Meaning*
Category	SAP/R3 Instruction Name
Characteristic	Name of characteristic
Required	Y/N/O if the characteristic is required to be in the recipe for processing or is optional

Table Field	*Meaning*
SAP required	If the characteristic is required for successful return of the message to SAP/R3
ATWRT_ATNAM	Enter ATWRT if it must appear in ATWRT, ATNAM if it must appear as an ATNAM request or EITHE if it could appear as either ATWRT or ATNAM
Plant_id	Plant id for the plant

Partial_result_instructions

If a message category is to return partial results during the execution of the recipe then an entry must be placed in this table.

Table Field	*Meaning*
Return_Category	Message Category which is to have partial results returned during the execution of the recipe
Request_part_name	Main characteristic which is used to set the time and date for the message category to be returned

Return_message

To restrict messages with certain status from returning to SAP the following configuration table has been introduced. It is used for PI_PHST, PI_SRST, PI_PHCON and PI_SRCON instructions.

Table Field	*Meaning*
Message_category	Message_category for monitoring the status
Instruction	Status instruction_characteristic
Value	Value of status
Return_flag	Y or N to return for the status value

Char_format Table

SAP requires different number of decimal digits to be returned. The number is taken as 4 unless it is given in this table. The characteristic PPPI_BATCH_CHAR_VALUE can only take one value regardless of the type of value. This will only apply to values that have a “.” .

Table Field	*Meaning*
Return_category	Message category
Char_name	Characteristic name for decimal digits

Table Field	*Meaning*
Format_len	Number of decimal digits allowed in SAP

Procedures_table Table

The programs that are used to translate the recipe down are configured in this table.

Table Field	*Meaning*
Proc_name	Procedure for translation in recipe
Order_of_exec	Order of execution of the procedure

Procedures

A table follows which summarizes for each type of modification the procedures that must be modified.

- Application Procedure

ex. Get a PI value: value

- Instruction Translation Procedure

ex. AREAD1

- Characteristic Translation Methods

ex. Usr_get_alias_tag

- Recipe execution Procedures

  ex. Usr_obatch_recipe

Table	***I*** nstruction	***C*** haracteristic	***A*** pplication	***B*** atch Execution	***D*** ata Destination	***D*** ata Source	***T*** ranslate Method
Procedure_table	X						
Translator	X					X	X
Group_master			X	X		X	
Exec_batch			X	X		X X	
Subscriber				X			
Subscriber_application				X			
Instruction_category	X						
Instruction_characteristics	X	X					
Application			X	X			
Partial_result_instructions	X						
Translation_method	X	X					X

Procedure	***I*** nstruction	***C*** haracteristic	***A*** pplication	***B*** atch Execution	***D*** ata Destination	***D*** ata Source
Application Procedure			X			X
Instruction Translation Procedure	X					X
Characteristic Translation Method	X	X	X			X
Recipe Execution Procedure				X		

Language Customization

All of the dialogs are made using a resource file that contains all the labels for the dialog. If you wish to change these labels for another language this can be done.

With this release we have changed the tag configuration on the configure application. If you have a previous translated dll you will have to change entry IDS_CONF_PLT for the new tab definitions.

Steps to build a new language version of PSRES.DLL

Note: To make all the resources appear in a different language, the resource file is edited in VC++, then the .rc file is compiled on the command prompt. The output of resource compilation is the .RES file which is attached with visual Basic project to build PSRES.DLL

1. In Microsoft VC++, open the file PSRES.RC.
2. Select the string table and modify the strings.

- Save the RC file after modification.
- On the command prompt, go to the resource dll directory and issue “RC PSRES.RC”.

This will generate PSRES.RES file.

- Open the PSRRES.DLL project from Visual Basic build the new PSRES.DLL.
- Copy this psres.dll on the \psrlink\shared directory and issue the following command to

  register the resource dll in the registry.

Regsvr32 /u psres.dll to unregister the dll

regsvr32 /c psres.dll ‘to register the dll

Guide lines for upgrading the resource file of PSRLINK application. If you want to add lines to an existing resource file you would follow the following example.

- Updating resource.h
- Open resource.h in Microsoft developer studio.
- Add the following line after 145

  #define IDS_MSG_PRESS_BTN_FOR_HELP 146
- Add the following line after 1484

  #define IDS_SET_PIBCH_CUSTOM_MSG2 1485

- Save and close the resource.h file.
- Updating psres.rc
- Open psres.rc in Microsoft developer studio and double click the string table.

2.2. Go to the end of the string table and double click on the last EMPTY line. A “string properties” dialog appears.

- In the ID field, type the following

  IDS_MSG_PRESS_BTN_FOR_HELP

  and in the caption, type the equivalent of the following message

  Please press % button for help
- Repeat the same to add the following resource ID and the string.

  IDS_SET_PIBCH_CUSTOM_MSG2

  Cannot be set. Recipe has open Phase(s).
- Save the psres.rc file.
- From the dos prompt, go to the psres.dll project directory.
- Type the following

  rc psres.rc
- Open psres.dll Visual basic project and build the dll.
- Copy the dll in \psrlink\shared directory.
- Execute and check the applications.

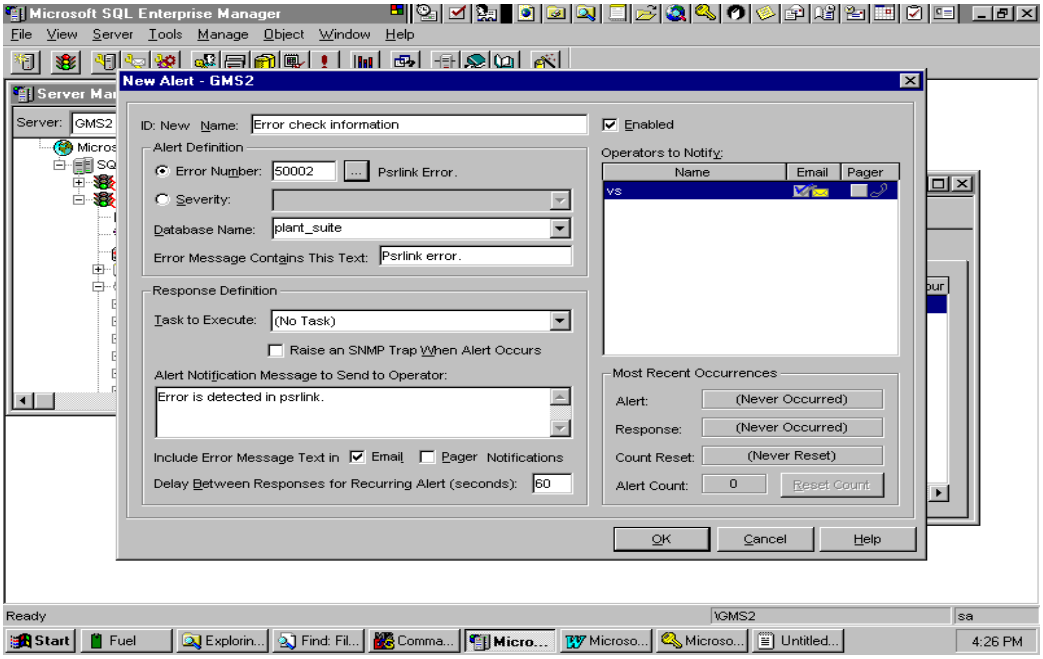
In order to support language changes for the profile application you must follow the same steps for the commoners.dll changing those fields that apply to the profile application.

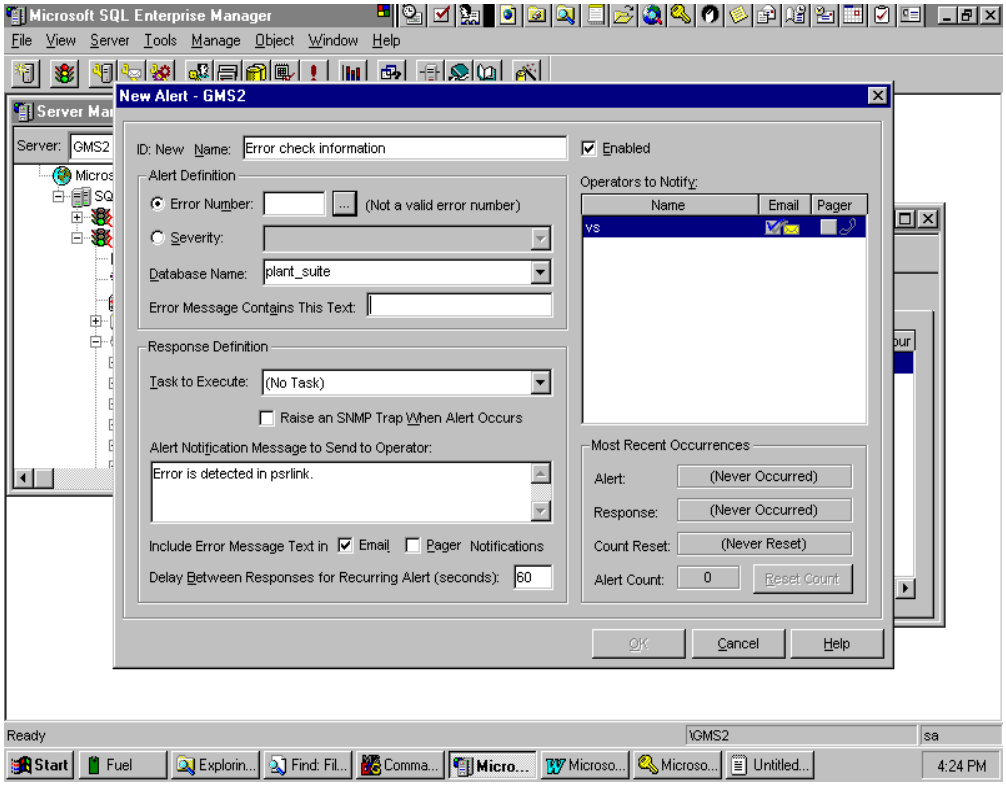
Chapter10
System Management

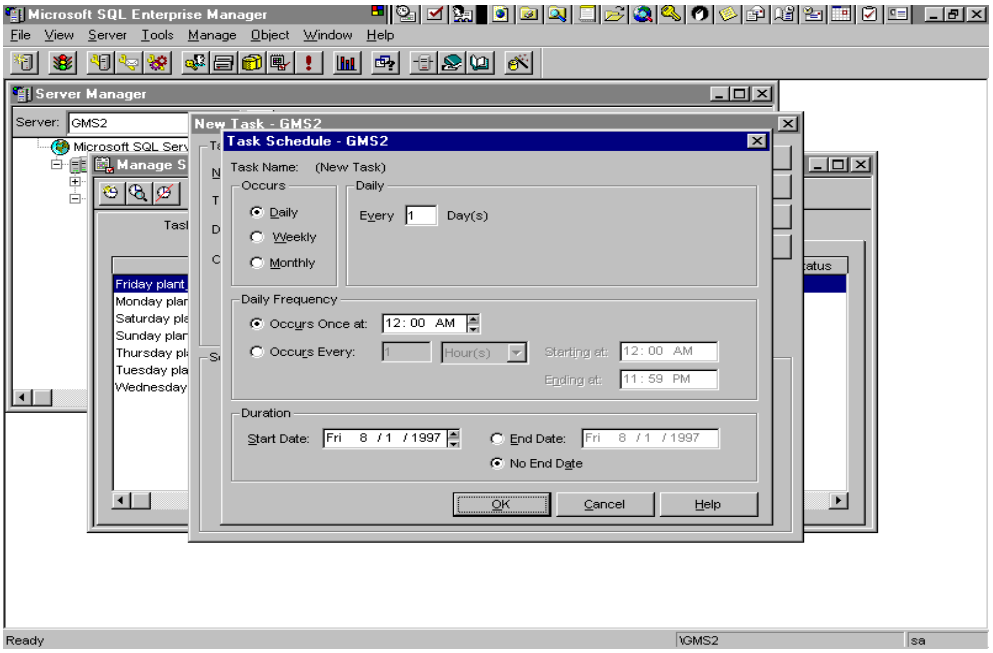
Error Log Monitoring

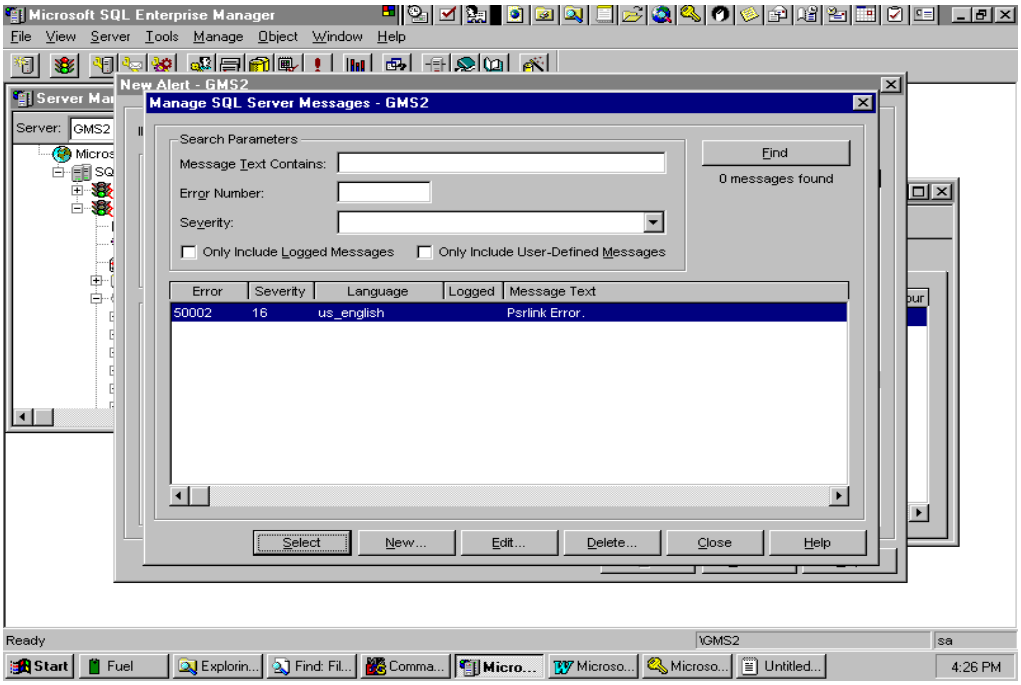
If there is a problem with some one of the PSRLINK processes this will be logged in the SQL Server table Error_log. If there is a problem with a recipe that has been downloaded which causes the translation of the recipe to fail it will also be noted here. The system will continue to operate on the remaining recipes. The user can evaluate the messages in the error log that can often assist with identification of the recipe error.

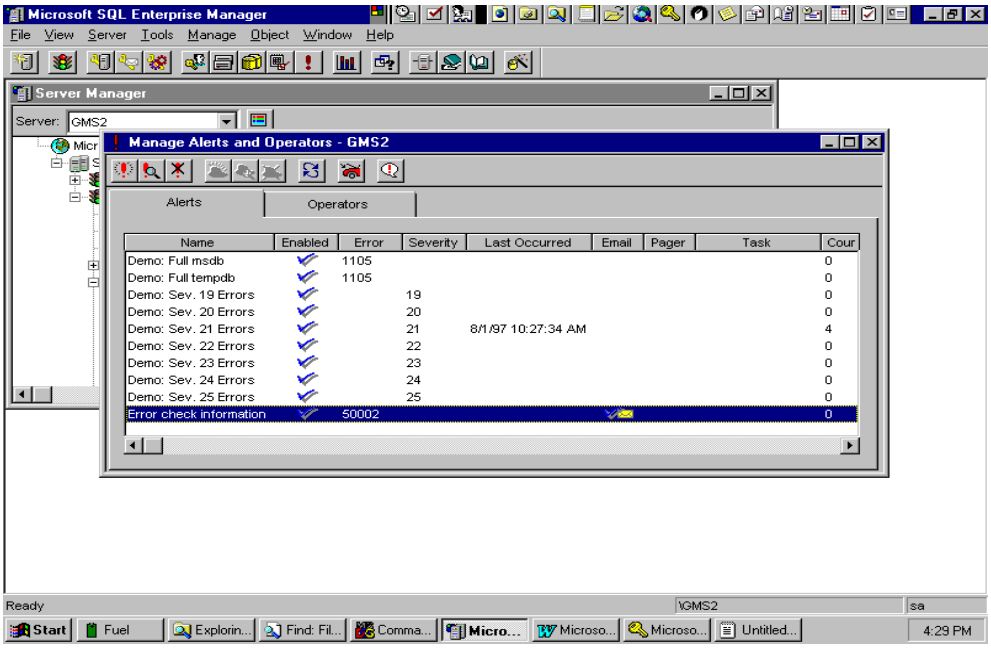
You can configure sample queries to run against the error log which if a record is detected it can send a mail message to the person responsible for that class of errors using standard Microsoft tools. To setup automatic notification if there is a problem use SQL Enterprise Manager, Server, Manage Scheduled Tasks. Setup a new task against the plant_suite database using one of the following queries. Select Options for automatic notification to e-mail or a pager. The following series of dialogs shows how this would be done.

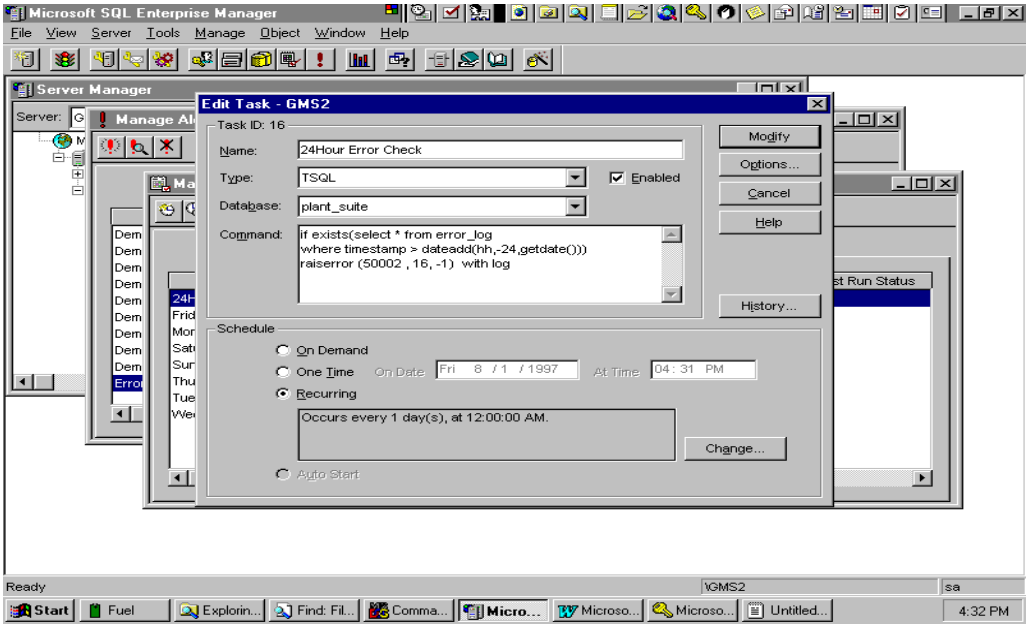


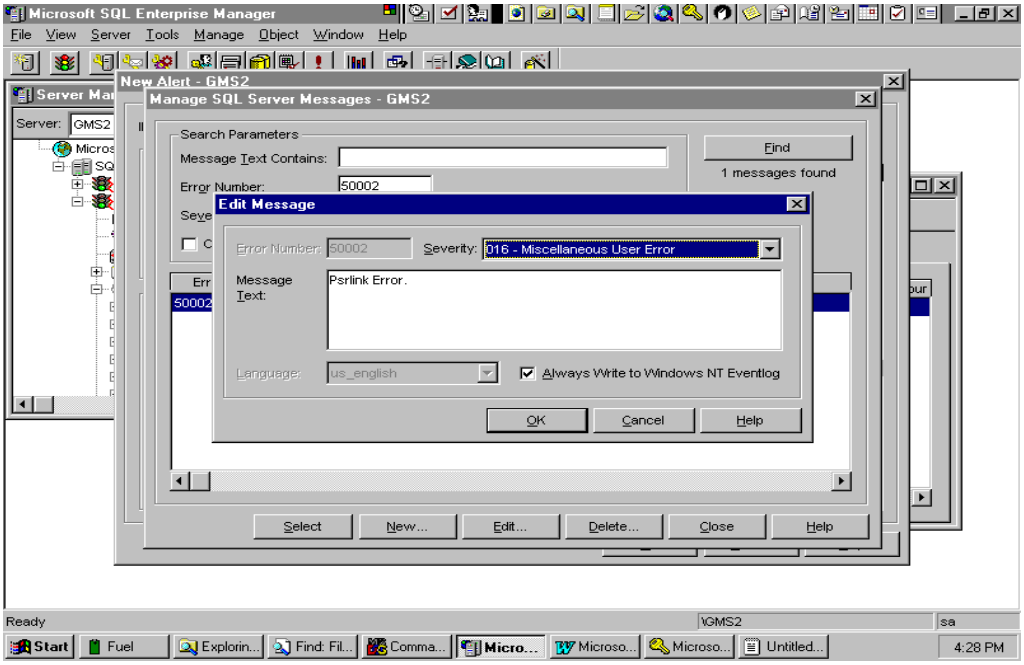

Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
New Alert - GMS2
ID: New Name: Error check information
Enabled
Alert Definition
Error Number:
(Not a valid error number)
Severity:
Database Name: plant_suite
Error Message Contains This Text:
Operators to Notify:
Name Email Pager
vs
Response Definition
Task to Execute: (No Task)
Raise an SNMP Trap When Alert Occurs
Alert Notification Message to Send to Operator:
Error is detected in psrlink.
Include Error Message Text in Email Pager Notifications
Delay Between Responses for Recurring Alert (seconds): 60
Most Recent Occurrences
Alert: (Never Occurred)
Response: (Never Occurred)
Count Reset: (Never Reset)
Alert Count: 0
Reset Count
OK Cancel Help
Ready
\GMS2
sa
Start
4:24 PM

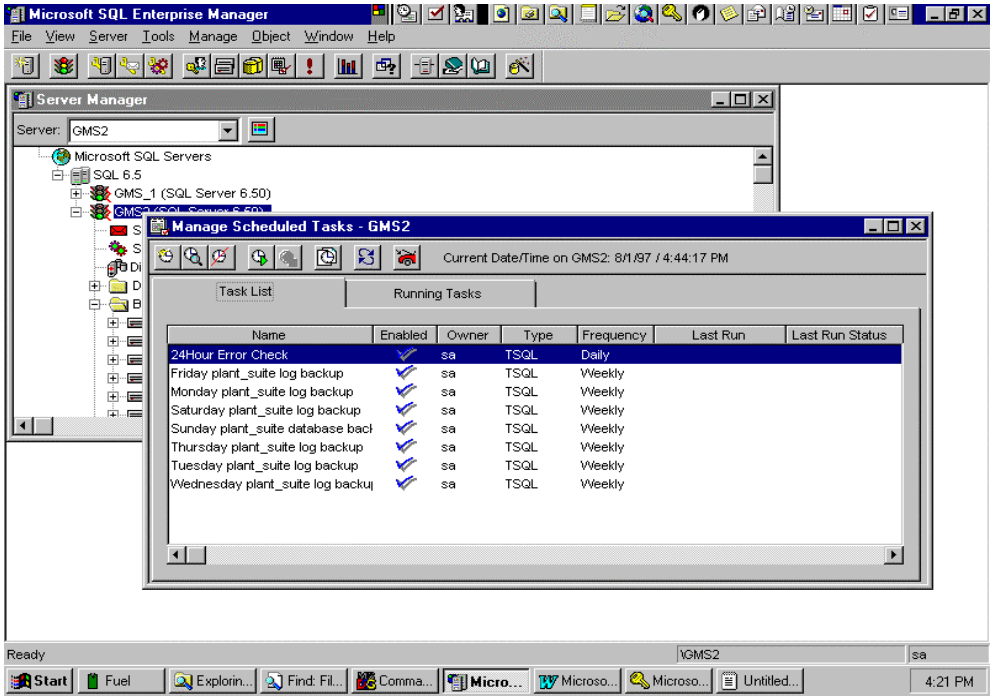

Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
Server Manager
Server: GMS2
New Task - GMS2
Task Schedule - GMS2
Task Name: (New Task)
Occurs
Daily
Weekly
Monthly
Every 1 Day(s)
Daily Frequency
Occurs Once at: 12:00 AM
Occurs Every: 1 Hour(s)
Starting at: 12:00 AM
Ending at: 11:59 PM
Duration
Start Date: Fri 8 /1 /1997
End Date: Fri 8 /1 /1997
No End Date
OK Cancel Help
Friday plant
Monday plan
Saturday pla
Sunday plan
Thursday pl
Tuesday pla
Wednesday
Ready
\GMS2
sa


Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
New Alert - GMS2
Manage SQL Server Messages - GMS2
Search Parameters
Message Text Contains:
Error Number:
Severity:
Only Include Logged Messages
Only Include User-Defined Messages
Find
0 messages found
Error Severity Language Logged Message Text
50002 16 us_english Psrlink Error.
Select New... Edit... Delete... Close Help
Ready
\GMS2
sa
Start Fuel Explorin... Find: Fil... Comma... Micro... Microso... Microso... Untitled...
4:26 PM


Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
Server Manager
Server: GMS2
Manage Alerts and Operators - GMS2
Alerts
Operators
Name Enabled Error Severity Last Occurred Email Pager Task Cour
Demo: Full msdb 1105 0
Demo: Full tempdb 1105 0
Demo: Sev. 19 Errors 19 0
Demo: Sev. 20 Errors 20 0
Demo: Sev. 21 Errors 21 8/1/97 10:27:34 AM 4
Demo: Sev. 22 Errors 22 0
Demo: Sev. 23 Errors 23 0
Demo: Sev. 24 Errors 24 0
Demo: Sev. 25 Errors 25 0
Error check information 50002 0
Ready
\GMS2
sa
Start Fuel Explorin... Find: Fil... Comma... Micro... Microso... Microso... Untitled...
4:29 PM


Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
Server Manager
Edit Task - GMS2
Task ID: 16
Name: 24Hour Error Check
Type: TSQL
Enabled
Database: plant_suite
Command: if exists(select * from error_log
where timestamp > dateadd(hh,-24,getdate()))
raiserror (50002 , 16, -1) with log
Modify
Options...
Cancel
Help
History...
Schedule
On Demand
One Time
On Date Fri 8 / 1 / 1997
At Time 04 : 31 PM
Recurring
Occurs every 1 day(s), at 12:00:00 AM.
Change...
Auto Start
Ready
\GMS2
sa
Start
4:32 PM


Microsoft SQL Enterprise Manager
File View Server Tools Manage Object Window Help
New Alert - GMS2
Manage SQL Server Messages - GMS2
Search Parameters
Message Text Contains:
Error Number: 50002
Find
1 messages found
Edit Message
Error Number: 50002
Severity: 016 - Miscellaneous User Error
Message Text: Psrlink Error.
Language: us_english
Always Write to Windows NT Eventlog
OK
Cancel
Help
Select
New...
Edit...
Delete...
Close
Help
Ready
\GMS2
sa
Start
4:28 PM



Following are sample queries of the error log.

PI is Down

select * from error_log

where message = “PI Logon”

and status = “failure”

and timestamp > dateadd(hh,-24,getdate())

SAP/R3 is Down

select * from error_log

where rcode_text =”RFC_IO5”

and timestamp > dateadd(hh,-24,getdate())

PID is Down

select * from error_log

where message = “BATCH Logon”

and status =”failure”

and timestamp > dateadd(hh,-24,getdate())

Error in retrieving PI Data

select * from error_log

where status like “Returned status value%”

OR status like “Returned percentage value%”

```
    OR status like "istat value%"
    OR  status like "Returned istat value%"
and timestamp > dateadd(hh,-24,getdate())
```

PI Data Marked in Error when PI was being Started

```
select intstat, message from error_log
where  status like "Returned status value%"
    OR status like "Returned percentage value%"
    OR status like "istat value%"
    OR  status like "Returned istat value%"
and timestamp > dateadd(hh,-24,getdate())

update action_results
set status = 'N'
where request_part_id = "Enter the instat no here you want to change"
```

Error in SAP/R3 Recipe Detected by PRECHECK

```
select * from error_log
where rcode_text ="PRECHK"
and timestamp > dateadd(hh,-24,getdate())
```

Error Message Process by SAP/R3

```
select * from MSHD
where rcode != "0"
and timestamp > dateadd(hh,-24,getdate())
```

Tag Does not exits

```
select * from error_log
where status like "pipt_findpoint error%"
    OR status like "Tag does not exists%"
    OR  status like "Tag not found or not yet connected to a server%"
and timestamp > dateadd(hh,-24,getdate())
```

Any error since yesterday

```
select * from error_log
where timestamp > dateadd(hh,-24,getdate())
```

No Quantity Tag found for Instruction

```
select * from error_log
where message = "No quantity tag id found"
```

and timestamp > dateadd(hh,-24,getdate())

No Common Name Tag found for Instruciton

select * from error_log

where message = “Tag id not found”

and timestamp > dateadd(hh,-24,getdate())

Error in application file name for PSRLINK

select * from error_log

where message like “The system cannot find the file specified%”

and timestamp > dateadd(hh,-24,getdate())

Purge Monitor

Use the SQL Enterprise Manager to check the status of the Purge Utility. Select Tools, Task Schedule, History Button. Here you will find a log of the purge function.

Transaction RFC Log on SAP/R3

If there has been a failure in communication between SAP/R3 and the PlantSuite RLINK system one can examine the status of the communication in SAP/R3 for those recipes or messages that were being sent from SAP/R3. This is done for example for the Control Recipes from the CO53 transaction. Select the recipe, choose Environment and then RFC log. The user can then review the status of the recipe and re-send the recipe.

The downloading of information is traced in a log file called trfcserv.tid. This file can get big over time. Periodically you should stop the TCRD process thenrename this file and start TCRD again. This will then create a new file.

Failure in PI data Retrieval

If an error has been detected in the retrieval of PI data it will be logged in the error_log table with the reason for the error that was detected. You must then correct the value in PI. A dialog is provided for correcting these values and making a journal entry for the result. The value is not changed in PI. See the Graphics Interface Chapter, Correcting Failed Result Collection.

SAP/R3 Down

If SAP/R3 has been down for an extended period of time and the plant is a PI-Batch type of plant then if operations have continued without the downloaded recipes operations must keep track manually of the recipe start and end times. Once the recipes are downloaded the planned start time must be corrected using the dialog “Adjusting the Recipe Start Time” described in the Chapter “Graphics Interface”.

PI Down

If PI has been down and no values have been collected then you can stop PSRLINK, manually enter values in PI for the points and times required, and then restart PSRLINK. If you do not want to totally take down PSRLINK you can change the name of the programs in the table exec_batch which get data from PI until you have had time to enter the data manually. This is done by just entering an ‘x’ for example on the name of the

file in this table for the programs, value, interpv, summary, getdsum, stsctrl, phsctrl. After the data has been entered manually in PI change these names back again.

If when you are restarting PI PSRLINK captures data when the archive is not completely up you might PI values marked as failed. If this has occurred you can reset their status so another attempt is made at retrieving the data. The queries required to do this are given in the error log monitoring section.

If PI goes down and data is being collected in a buffering node then you should shut down PSRLINK until the data has been caught up in PI and then restart PSRLINK.

Database Problems

Problem: It takes an unusal length of time to process recipes

Soution: Change the entry in exec_batch for usr_read_and_process to be "usr_read_and_process 1" this will write statistics to the table "timetable" which will help us detect if a procedure has gotten corrupted or an index.

Problem: A procedure has gotten corrupted

Solution: Create a new directory, copy the QRY file for the corrupted procedure to this directory. Copy dec from winnt\system32 to this directory. Drop the procedure using ISQL and then execute the program dec from the directory you have moved it to.

Problem: Tempdb is full.

Solution: Expand the size of the tempdb database

First from ISQL do dbcc checkdb(plant_suite)

Using the enterprise manager create a new device for temp called dev_temp and make it 8 MB. Then go into the database and double click on tempdb and expand with the new device.

SQLServer Logspace full

In order to correct the situation do the following

1. Execute Enterprise Manager
2. Select "SQL Query Tool" option from "Tools" menus. Type "dump transaction plant_suite with no_log" and press execute (green arrow).
3. Select "Database Backup/Restore" option from "Tools" menu. There will be two property sheets one for Backup and the other for Restore. Choose the Backup option. Choose plant_suite database from the database combo box. In option frame select "Initialize device" then Press "New" button. In the location edit box, type full path where you want the backup to be made to. In the name edit box, type "plant_suite_backup". Choose "plant_suite_backup" device from the backup devices box. Check that the plant_suite database is chosen from the database combo box. Press "Backup Now" button.
4. Check the logspace by entering the command "dbcc sqlperf(LOGSPACE)" in the SQL Query Tool.
5. To verify the state of the database execute the following command in ISQL. "dbcc checkdb(plant_suite)"
6. Check the amount of database storage used with sp_spaceused

Time Issues

The systems person should be cautioned on how to handle daylight savings time. Since the recipe from SAP/R3 come down as a specific time setting there will be an ambiguous time issue when the time changes. If the recipe ends on the hour the time changes it will not know which result for that hour to retrieve, the first occurrence or the second. If a flow is being totaled and the change is included in the time range the total will include both hours. The standard rules for PI time apply.

The system operator is also responsible for keeping time in synch between PI, the PSRLINK and the SAP/R3 machine.

Update Database Statistics

Updating the database statistics might be required if there has been a large number of changes to the database tables. This can be done using the procedure that has been provided by going to ISQL for the plant_suite database and executing the procedure usr_run_updatestatistics.

Corrupted Index on Table

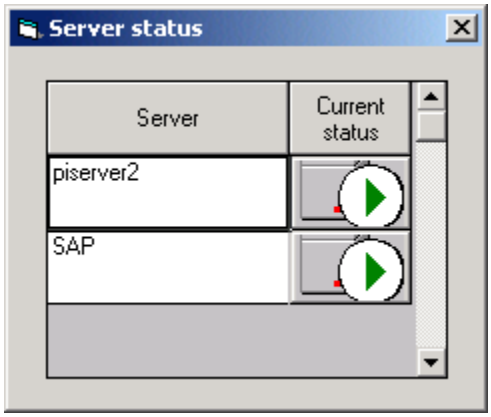
If an index on a table has become corrupted you can recreate it by using SQL Enterprise, selecting the database, the table and then the index and choose rebuild.

PI and SAP Down for Backup

If your PI server or SAP are taken down for backup, that is a known shutdown of the server you can avoid getting error messages that must be validated if you change the status of the server in the RLINK system during the shutdown. This is done by using the application srstatus.exe. This application will allow the user to change the status of a PI server or the SAP server. The available PI servers are configured in the table called servers. If a PI server is not running then the applications for that server will not be executed. If SAP is not running then the PMU or the PMUCL applications will not run. The status of a server can also be changed using the stored procedure

usr-server_status ‘U’,’servername’, ‘Yor N’

For each application the field that is used to hold the server name is configured in the table pi_functions2.



Another option for handeling PI not being available during backup is to stop the service for PSRLINK while the PI backup is being done. This can be done by constructing a .bat file with the directory path of the rlink service and then giving the command net stop psrlink or net start psrlink. A sample file is

d:

cd rlink\services

net stop psrlink

Exits for Certification Testing

The application RLBOCL.EXE has the following options

rlbocl -?	Displays various options in a message box
rlbocl -1	Create process message (Multiple)
rlbocl -2	Get characteristics list
rlbocl -3	Get recipe list
rlbocl -4	Recipe request
rlbocl -5	Get helpvalues

rlbocl –1 -U Create process message (Multiple) with User Exit option

With –U command line parameter, the application will skip transaction commit call. By not committing, the status of the messages sent up to SAP would have the status 'X' in mshd table of plant_suite database.

rlbocl –4 -U Recipe request with User Exit option

With –U command line parameter, the application will skip transaction commit call. By not committing, the recipe status in SAP would still be 'created'.

TCRD.EXE application has the following functions

control_recipe_download

process_mess_download

control_recipe_available

process_mess_get_return_code

In TCRD.EXE application, we can specify user exit option by setting the command line parameter as the following

TCRD.EXE –U or TCRD.EXE –u

If the parameter "-U" is passed, then before calling RfcSendData, at the prompt, the application would ask for user input with the following question

"Do you want to execute RfcSendData (Type Y/N then press ENTER):" and wait for the user input.

If 'Y' is entered then

It would commit plant_suite database transaction and call RfcSendData.

If 'N' is entered then

It would rollback the plant_suite database transaction and call RfcAbort with "TCRD: User has requested to abort the operation." Message.

PMUCL.EXE application: Client Program for transactional RFC process message upload

In PMUCL.EXE application, we can specify user exit option by setting the command line parameter as the following

PMUCL.EXE –U or PMUCL.EXE –u

If the parameter "-U" is passed, then before calling RfcIndirectCallEx, at the prompt, the application would ask for user input with the following question

"Do you want to execute RfcIndirectCallEx (Type Y/N then press ENTER):" and wait for the user input.

If 'Y' is entered then

It would call RfcIndirectCallEx followed by RfcConfirmTransID

If 'N' is entered then

It would log error message "PMUCL: User has terminated the operation." And return RFC_FAILURE.

TCRP.EXE application: Client Program for transactional RFC control recipe pull

In TCRP.EXE application, we can specify user exit option by setting the command line parameter as the following

TCRP.EXE –U or TCRP.EXE –u

If the parameter "-U" is passed, then before calling RfcIndirectCallEx, at the prompt, the application would ask for user input with the following question

"Do you want to execute RfcIndirectCallEx (Type Y/N then press ENTER):" and wait for the user input.

If 'Y' is entered then

It would call RfcIndirectCallEx followed by RfcConfirmTransID

If 'N' is entered then

It would log error message "TCRP: User has terminated the operation." And return RFC_FAILURE.

TCRPS.EXE application: Client Program for transactional RFC control recipe pull single

In TCRPS.EXE application, we can specify user exit option by setting the command line parameter as the following

TCRPS.EXE –U or TCRPS.EXE –u

If the parameter "-U" is passed, then before calling RfcIndirectCallEx, at the prompt, the application would ask for user input with the following question

"Do you want to execute RfcIndirectCallEx (Type Y/N then press ENTER):" and wait for the user input.

If 'Y' is entered then

It would call RfcIndirectCallEx followed by RfcConfirmTransID

If ‘N’ is entered then

It would log error message “TCRPS: User has terminated the operation.” And return RFC_FAILURE.

Chapter11
Debugging RLINK

Diagnosing problems in PSRLINK

Below is a list of frequently occurring problems and how to diagnois them. This list will be updated on the OSI web site for RLINK.

Problem	Method of Diagnosis and solution
TCRD Service Will not start	1. Does TCRD start in a DOS window. If it does then there is a problem with the system environment variable for RFC_INI not being set up as a system environment variable with the complete path and file name. 2. If TCRD does not start in the DOS window then there is something wrong with the SM59, or the saprfc.ini file or the RLINK registry application.
No messages being returned to SAP	1. Check the error_log or the error_log appplication for any error_messages. 2. Are the messages in the table MSHD ready to be returned to SAP, the Rcode is blank in MSHD. If no then there is a problem with your recipe, configuration or PI values are not getting set. 3. If there are messages waiting to go up then check the group_master table or the configuration application for the time frequency of sending data to SAP. If this is correctly then check that you have set up the account and password correctly in the RLINK registry application. Check the saprfc.ini file that the OSISOFT destination has been setup correctly.
TCRD will not start in DOS Window	This can be cause by one of the following: 1. The SAPRFC.INI file is not setup correctly 2. The SM59 transaction in SAP is not set up correctly 3. The PSRLINK Registry Application is not

	setup correctly 4. The environment variable for RFC_INI in system environment variables is not set up as a system variable and does not include the path name and file name
No material PI_CONS or PI_PROD messages	1. Are there any messages in the error_log that would indicate that the tag could not be found and thus a problem with configuration. 2. Has the recipe started and has the phase started and for the phase that has the material you are looking for has there been a partial or completion set for the phase. You can check the phase_status_details to check the status of a phase. 3. If you are using material after batch as the application have you gotten the batch_id yet.
SAPPOLL service will not start	1. The PSRLINK registry application is not setup correctly 2. The SAPRFC.INI file is not setup correctly for the OSISOFT Destination. 3. The RFC_INI environment variable has not been setup in the system parameters section.
No messages being marked as complete in PSRGUI	1. Need to check if the control recipe has started 2. Messages will only move to the completed section when the phase has been stopped.
Everything seems to halt processing	Check the database space. This is a problem which can occur on SQL6.5 . You can use Enterprise manager to increase the size of the database.
SAP is dropping the connection and the progam is not registered.	Install the gateway locally on the NT server
No values are going back for the PI-PHST instruction	1. Is this a recipe that you are using a different resource than the one that came down in the recipe? If so did the recipe include a PPPI_PHASE_RESOURCE in the APHST_I instruction. 2. Do you have a point group configured for the phase with the correct resource id and a matching name to the PPPI_EXTERNAL_PHASE.
Too much material being reported back on partial consumption	Is the PI tag set to be zeroed after the partial phase if it is a totalizer point.

Phases not completing	Have you sent a final status of 0002 for the phase
The recipe is not starting	1. Did a OSI_START_DATE and OSI_START_TIME come down with the recipe and is the start time between 000000 and 235959. This can be check using the PSRGUI. Choose the AORD instruction under FT. Also check the recipe table and see that there is a plan start time. 2. Is there a Point Group configured for the recipe and resource network
Return Code from SAP indicates there is something wrong with the numerical value sent to SAP	Is the user account on SAP set up to use the delimiter of "." and not ","
PISETBATCH does not seem to get the status updated	1. What is the clock time on the client machine that you are working on compared to the PI server. IF it is ahead in time the values will not write to PI until that time. 2. Check the table action send. Did the values get written to PI and thus have a status of C or is the status still N. If the status is F there should be a message in the error log 3. If the values wrote to PI and the status still did not change check the phase_status_details time. If there is no entry for the time you are looking at there is something wrong with the configuration of the point group or you have changed resource but have not configured resource change in the recipe. 4. Is the time you are entering consistent with the timezone of the PI server
Overlapping Phases return PI_CONS for both recipes	Cannot run more than one recipe in the same phase resource at the same time. Change the application to get tag application as a work around.
Problem in Polimeri with customization in the procedure_table	The numbers should be adjusted to allow for the 2 custom procedures being added
You get the message that you are not authorized to logon to the target system	Something is wrong with the registry configuration for the logon to SAP or the SAP logon has not been set up correctly
Error message of 1000119 in error_log and recipe marked in E	The AMATP01 instruction is missing from the SAP_MESSAGE_ALIAS table
Multiple recipe in same resource network at same time	Need to create a separate resource network and point group for the recipe and

	resource_network.
Redundant database switch-over	The servername was not changed on action_results table to the other server
PI_CONS and PI_PROD message not being answered	PI values are not at the same time for Batch tag and material tag and the multival application is used.
Error number 2627	This usually means that there is a problem with one of the indexes from ISQL select the database plant_suite and then run DBCC checkdb from the ISQL window. You can also try resetting the index. This is done using ISQL and executing the following command DBCC dbreindex (“plant_suite.dbo.request_part”) Request_part is the table name which would be replaced with the table name that is giving problems. The tables that exhibit some problems when there are many cleanups or formula, message_request, request_part, tlines. If this problem persists there is a separte task available to be scheduled to handle rebuilding the index.
No material being reported for phase	Is the material configured for the phase in the recipe sent down from SAP. Check the material_tag table. Was the material configured in the Recipe
Messages for CO57 not being downloaded	The message number was the same as already present in the database, this can be caused by a change in client number. When you switch client you have to clean msg_mshd, msg_msel, msg_tines
Live to stand by switch over no values	Change the server name in action_results for the correct server, material_tag_common_name and point_group_members
Translation method for activity should be usr_phact_activity not usr_phact_monitor	Change the name in the translation_methods table from usr_phact_monitor to usr_phact_activity
The patch did not seem to take effect	Check the *.out files. Run the bat file with no parameters and check that you have input the parameter list in the correct order.
Install of the patch for build resulted in 1000118 usr_load_all error message	Add AMATP01 in SAP_MESSAGE_ALIAS and add to the procedure table usr_prod_mpo9 with the next number
Usr_msgdr24 error about DEST for the new release	In the system_paramter table add in the text column the SM59 destination

Cannot restore database dump	Must select the option to override the existing database
Continuous plant get error on initial process of the recipe about date conversion	No shift duration has been set up
In a pi program get an entry in the error_log table something like System error: Error No. number	There is something wrong with the PI logon. Check the error code with the PI documentation or the pistatus.h file for more detail.
The recipe will not come down and the log says there is something wrong with the material short text	The character ‘ cannot be used in any of the text descriptions. This is a reserved character. You can use ‘’ or “ instead.
On PI putvalue 01149	There is something wrong with the time relative to the PI time. Are you sending a valid time for the machine which holds PI
PRECHK errors	Check recipe and Instruction_characteristics table for compatibilty
Between a partial and finish report back the last value for material before the last partial	If no value between the last partial and finish exists must put a 0 value in PI for this time range
Tag_id not found	Check the material_tag table and common_name table are configured correctly
2627 error on a table which indicates that something has happened to the index	Dbcc checktable (table_name) Dbcc dbreindex (table_name) Update statistics table_name
RCODE other than 0 coming back in MSHD and MSEL	Check the Message Correction Application or the error_message table for further information on why SAP would not accept the message.
String tags do not seem to be picked up with the correct timestamp	What version of the Piapi32.dll do you have. You need version 1.3.1.3
Error in PSGUI multiline	New release patch
Error in Pisetbatch of invalid procedure	New release patch
If PPPI_YIELD_TO_CONFIRM did not go back with the phase instruction	Check instruction_requirements that this has been configured to be required.
PI programs are giving an error of –1	There is a problem with logging on to PI
PI programs are giving an error code of 2	There is a system problem with the network
Material Tag not found	Check the configuration for the material tag. If there are leading 0’s in the material number that comes down from SAP these must match. The materials that came down in a recipe can be seen in the material list table. The configuration is in the material_tag table. You can also see the materials that came down in the recipe using PSRGUI and selecting the FT number for the AMAT instruction. You can see what materials are configured by using the

	configuration application.
You get no material cons being reported although you have all the parts in request_part as status C	Are the timestamps the same for the PPPI_BATCH and PPPI_MATERIAL_CONSUMED. If not you probably used the wrong applications. Check your translator configurations.
After PI has gone down I am not getting any messages returned	Check action_results that status of F on any of the items might be because PI went down. Reset to N or P depending on the type of request.
I see error messages in the log of greater tha 0 for PI	This means there was some system type failure with PI. If everything is now OK with PI then check action_results for any items with status of F that might have gotten set and not need to be reset to N or P depending on the request.
TCRD gets error in TID management	Do you have the correct path set in the registry application
Phases do not seem to start but the correct data is in PI	Check the table action_results for the phase status. Do you have a valid date in trigger timestamp not 1/1/1900. If there a timestamp in field5. Check that you have sent down a valid resource with the recipe and that this resource is configured in the point group table.
The communication with SAP works but a recipe does not come down fails after TID check	Either the path in the registry for the .lck files is not correct or the .lck files are read only.
Error –119 on the install with a dll name	There is a problem with registering this dll on the machine. This could be do to security or read protection or the dll being registered
Installation of version 1.35 fails on the librfc32.dll for SAP	Remove the old version of SAP GUI and install the 4.6 version.
Installation of version 1.35 fails on the comcat.dll	Replace with older version and then bring back new one after install
No procedure with name usr_sync_inparm Version 1.35 upgrade	The correct name for the procedure is usr_sync_iniparam
Sappoll service will not start automatically with Windows 2000 but will start manually	We have found a problem with the librfc32.dll from SAP that version 4.6A will work but the other versions will not, we have gotten 4.6C with 2000 professional to work
2627 error message when processing a recipe	Check your recipe that you do not have mulitple lines requesting or setting information about the same characteristic
1000131 usr_load_all	You are missing the AORD instruction or have something wrong with it in your recipe
No programs seem to be running	Check that there are entries in the PI_functions table. These entries should be

	GETSNAPSHOT GETTAG GETTAGRANGE PUTVALUE GETSUMMARY GETDSUM PUTSNAP MULTIVAL GETDIFF GETDIFFWAIT GETTAGRANGEWAIT MULTIVALWAIT GETTAGWAIT GETINPVALUE GETINPWAIT SUMMARYWAIT DSUMWAIT GETTAGJ GETTAGJI DELIVERY
1003001 usr_batch_create	If this is an upgrade from a 1.34 system to a 1.35 system then you must check you SAP_message_alias table for all plants and add the following new instructions for the plants that are missing them. ABTCL, ABTCR, ASRST, ASRCON, APHCON and ASRACT
Want to report 0 quantity	If you want 0 quantity for cons but not prod then use patch6, if you want 0 quantity for both use patch4
Memory error with multival using pimod.exe	Apply patch5
Cluster support of services	Apply patch 8
Support interrupt status	Apply patch 7
Do not get any results back for cons or prod	Are you using different applications like multival for PPPI_BATCH and gettag for PPPI_CONSUMED or PRODUCED
Upgrade fro 1.34 to 1.35 and having problems	Check translator table that you do not have entries for Zetc. Instructions.
A lot of error messages are occurring because of point not found and entries are appearing for errors in action_send	The point group of type ERROR needs to be check and the points for the members created in PI, if you do not want to do this then remove the procedure usr_set_alarm from group_master and exec_batch
Not getting character set support for unique	multi linqual 850 must be

language characters	chosen when installing SQL
usr_axr_sel the application to change the date does not support the date format on some none English systems	Patch3
Support of a finish time and date on continuous recipe	Patch3
Hot keys for program execution	Some of the hot key assignments conflict with keyboards remove from properties on menu entry
PI-PHCON has the wrong translation method in the ini parameters ist should be usr_phcon_time_status not usr_time_status	Change usr_time_status to usr_phcon_time_status in the translator table
Using the SAP_TRAN a characteristic does not appear in the message	Has the format been setup in characteristic
SAP_the message does not get to mshd	SAP_TRAN be sure that exec_batch has the following order for program execution, arsptran 1, usr_msg_hdr_24 2 and usr_ar_sap_tran 3
Recipe application requires old version of ocx	New version of recipe available in patch 9
Pisetbatch on a separate client machine does not respond	Verify that the plant information tab is configured. Also that your time zoned is compatible with the PI server
PI Tags for the PI-SDK does not work	Verify that PI-sdk is configure properly. Using the AboutPIsdk.exe utility.
ODBC connection does not connect. The logon applicaion will not connect to a remote server	Verify that ODBC is configured with TCP/IP not named pipes.
Error file not found	Have you set the path environment variable to include the RLINK\shared
Error are occurring when PI is taken down for backup	Use the server status to set status of the PI server before and after backup

This procedure is for checking status of an individual instruction. (Using the filter option in Microsoft Access will help to find the items more quickly)

1. In the PSRGUI for the recipe check the MESSAGES for the instruction that is of interest. Note the number of the instruction appears in the message list. If you right click on the message you will see in the right window the values that are currently available.
2. With the number that is noted in step 1 use the Access Database and the table request_part. Select the rows that have a entry in the second column that correspond to the value taken from step 1. Check the status of each entry. If the status is C then the instruction characteristic is complete. If the status is A then the characteristic has passed the translation and is in the table Action_results waiting to find a value.

If the status is N then the characteristic has not been translated. If the status is W then the value has already been written back to the tables for return to SAP.

3. Status is A for the request part. In this case go to the table for Action_results for the number is the first column from request_part corresponds to the characteristic you are interested in. Select the row in action_results that corresponds to this number. Check what is the status of the entry by looking at the status column. If the status is N then no value has been found for this yet. If the value is P then a partial value has been found. If the status is S then all values have been found. The tags that are used for the request are also shown here. You should use these tag names to check the values in PI. If you think that these are not the correct tag names then you must return to configuration to see what has been setup incorrectly.
4. The actual values that have been retrieved to date are in action_result_values using the request_part_id as a location.
5. In action results check that a trigger procedure exists and a trigger timestamp has been set. The program file name that corresponds to trigger procedure is given in the following table. If you question that the correct values are being returned from PI then you can execute the program in a dos window set to the path rlink\server\fe program_name -T

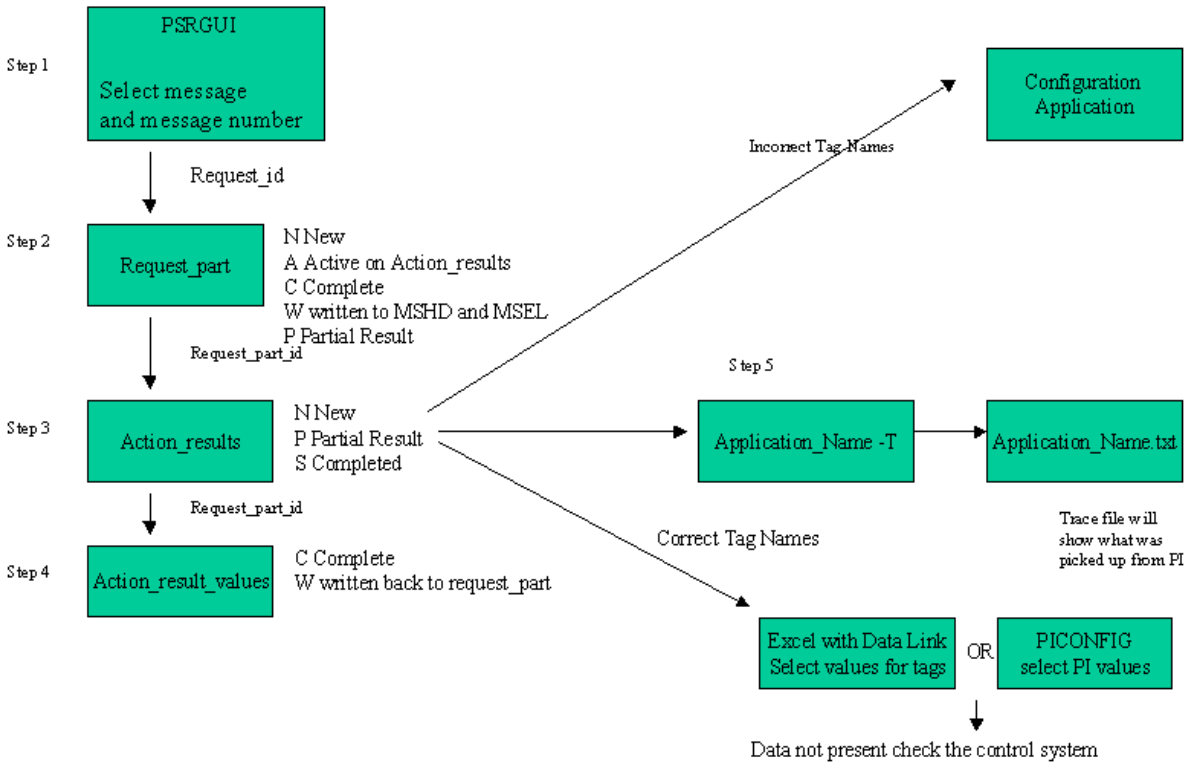
   The –T option will cause a trace file to be written with the name of the program_name.txt. You can then use this file to trace what was requested and what values were gotten from PI.

Application Name	Executable Name
gettagrange	Interpv.exe
getsnapshot	Snap.exe
putvalue	Putvalue.exe
gettag	Value.exe
getsummary	Summary.exe
putsnap	Putsnap.exe
getdsum	Getdsum.exe
control_monitor	Stsctrl.exe
phase_monitor	Phsctrl.exe
multival	Multiv.exe
getdiff	Getdiff.exe
gettagrangewait	Interpvw.exe
getdiffwait	Getdiffw.exe
multivalwait	Multiw.exe
gettagwait	valuew.exe
getinpvalue	Valuei.exe
getinpwait	gviw.exe
summarywait	Summaryw.exe

dsumwait	Getdsumw.exe
getqmval	Qm.exe
delivery	Delivery.exe
gettagji	Jvaluei.exe
gettagj	Jvalue.exe
resource	Phaseres.exe

6. When looking at the PI values if they are for control recipe status or phase status they must have entries for the status and value at the same timestamp.

Tracing the path a message request from SAP



Places to Trace errors in the system

1. The error_log table
2. The Log Review application
3. The -T parameter on every application
4. The log files for the sappoll, psrlink service

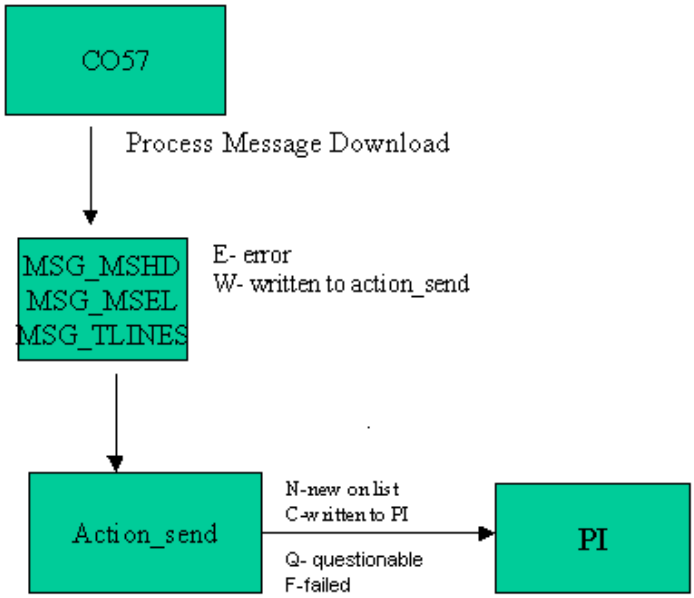
5. PSRGUI for reply status of the messages
6. The message correction application for SAP messages, check meaning of returned message in the error_message table

Cleaning up Recipes in the Server

There are three methods for cleaning up recipes in the server. You should shut down the PSRLINK service before you execute any of these procedures. The following procedures can be executed from the Query Analyser or SQL. Change the database to plant_suite.

1. usr_clean_a_recipe "recipe_no" – this procedure is used to clean up a single recipe so that it can be started again. The entry for readstatus in the CRHE table will be set to blank.
2. usr_clean_rcp_from_base_tables "recipe_no" – this procedure will delete all information about a recipe including the tables CRHE, CRFT, CRFV and TLINES.
3. usr_clean_up 'YES' – this deletes information about the recipes in tables other than CRHE, CRFT, CRFV and TLINES. The status of the recipe in CRHE is not reset to blank so it will not run again unless you change this status.

CO57 Messages from SAP



The putvalue application has a status of "Q" for questionable. This is to handle the case when there is some network problems with connecting to PI. A status will be set to Q instead of F. This will allow a retry of sending the value to PI but will alert the system manager that there is a potential problem.

Recovery from Down Servers

	SAP Server	PSRLINK Server	PI Server	Action
Case	UP	UP	UP	No Action Required

1				
Case 2	UP	DOWN	UP	When the PSRLINK and SAPPOLL service restart you will have to send down the recipes that have been captured in SAP. These are found using SAP transaction CO53 and selecting Environment and then TRFC Log. Execute the Download function for those that have not been sent.
Case 3	UP	UP	DOWN	Either data is buffered in the DCS or in PI-NET nodes, or manually written down in the case of manual data. When PI comes back up the buffered data will be recovered. The data that is usually entered in ProcessBooks would have to be entered. Consideration has to be given to any calculations that are being done. Check the table action_results for any status codes of F. If these occurred during the down period rechange status to N or P depending on request type. (if there have been partial results to data then you should set the status to P. The can be validated by looking at request_part status). Set the status of the PI server to down with the server status application to avoid excessive error messages in the error_log table Pi goes down and you need to recover archive. You need to stop PSRLINK until the data has been recovered in the PI system otherwise since data is filled into PI most recent first you will loose possible events.
Case 4	DOWN	UP	UP	No recipes or CO57 message will be coming down from SAP. If the recipes that are current are already in PSRLINK ther is no problem continuing to process against them. The messages will stay in the PSRLINK server until the SAP server comes back up. If there are no recipes in PSRLINK you will have to manually write down the postings until SAP can come up and a recipe can be created. At the time you send

				down the recipe only the current time can be put on the recipe, you will have to readjust the starttime of the recipe to be when actual production was taking place. This can be done using the application SAP Recipe Time. You will have to manually update the PI Tags with the correct control recipe reference. Instead of using the application SAP Recipe Time another option is to stop PSRLINK service, send the recipe from SAP and then update the CRFV for the OSI Start Time and OSI Start Date and set the read_time in CRHE to be before production started.
Case 5	UP	DOWN	DOWN	Case 2, Case 3
Case 6	DOWN	UP	DOWN	Case 4 , Case 3
Case 7	DOWN	DOWN	UP	Case 4 , Case 2
Case 8	DOWN	DOWN	DOWN	Case 2, Case 3, Case 4

Tables Trace Execution

The Recipe from SAP is downloaded into the following tables:

- CRHE
- CRFT
- CRFV
- CRA_TO_CRP
- TLINES
- Msg_mshd
- Msg_msel
- Msg_tlines

Data is returned to SAP with the following tables:

- Mshd
- Msel
- Up_lines

The recipe is translated in SP88 in the following tables:

- Formula
- Recipe

- Recipe_status_detail
- Phase
- Operation_phases
- Phase_status_detail
- Material_list
- Operation
- Sec_resource
- Sec_resource_status_detail

The recipe uses the following tables to process the request for information

- Action_send
- Action_results
- Action_result_values
- Message_request
- Request_part
- Request_part_values

Tables used in General PP Transactions

- Ar_sap_tran
- Arv_sap_tran

Error Code Messages

RLINK Error code and its description.

PROCEDURE	ERROR	ERROR DESCRIPTION
usr_load_all	1000116	Entry not found for AMAT_1 in sap_message_alias
	1000117	Entry not found for APHASE_1 in sap_message_alias
	1000118	Entry not found for AORD_1 in sap_message_alias
	1000119	Entry not found for AMATP01 in sap_message_alias
	1000101 1000121	Error in inserting records into material_list for AMAT_1 Error in inserting records into material_list for AMATP01
	1000102	Error in inserting records into operation
	1000103	Error in inserting records into operation_phases
	1000104 1000120	Error in inserting records into phase
	1000105	Error in inserting record into recipe

	1000124	plant_type is null or blank in plant_resource_network
	1000107	Error in inserting record into recipe_datetime_temp
	1000115	Error in inserting record into material
	1000126	Problem with AORD
	1000131	No OSI_START_DATE or alias not configured
usr_cons_mpo1	1000208	Entry not found for ACONS_1 in sap_message_alias
	1000201	Error in inserting record into message_request
	1000202 1000204	Error in inserting record into request_part
	1000203	Error in inserting record into request_part_values
usr_cons_mpo2	1000312	Entry not found for ACONS_1 in sap_message_alias
	1000302	Error in inserting record into message_request
	1000303 1000304	Error in inserting record into request_part
	1000305 1000306 1000307 1000308	Error in inserting record into request_part_values
usr_cons_mpo3	1000411	Entry not found for ACONS_1 in sap_message_alias
	1000402	Error in inserting record into message_request
	1000403 1000404	Error in inserting record into request_part
	1000405 1000406 1000407	Error in inserting record into request_part_values
usr_cons_mpo4	1000510	Entry not found for ACONS_1 in sap_message_alias
	1000502	Error in inserting record into message_request
	1000503 1000504	Error in inserting record into request_part
	1000505 1000506	Error in inserting record into request_part_values

usr_cons_mpo5	1000609	Entry not found for ACONS_1 in sap_message_alias
	1000602	Error in inserting record into message_request
	1000603 1000604	Error in inserting record into request_part
	1000605 1000606 1000607	Error in inserting record into request_part_values
usr_cons_mpo6	1000711	Entry not found for ACONS_1 in sap_message_alias
	1000702	Error in inserting record into message_request
	1000703 1000704	Error in inserting record into request_part
	1000705 1000706 1000707	Error in inserting record into request_part_values
usr_cons_mpo7	1000808	Entry not found for ACONS_1 in sap_message_alias
	1000802	Error in inserting record into message_request
	1000803 1000804	Error in inserting record into request_part
	1000805 1000806	Error in inserting record into request_part_values
usr_cons_mpo8	1000908	Entry not found for ACONS_1 in sap_message_alias
	1000902	Error in inserting record into message_request
	1000903 1000904	Error in inserting record into request_part
	1000905 1000906	Error in inserting record into request_part_values
usr_crst	1001007	Entry not found for ACRST_I in sap_message_alias
	1001001	Error in inserting record into message_request
	1001002	Error in inserting record into request_part

	1001004	
	1001003	Error in inserting record into request_part_values
usr_opst	1001109	Entry not found for AOPST_I in sap_message_alias
	1001101	Error in inserting record into message_request
	1001102 1001104	Error in inserting record into request_part
	1001103 1001107	Error in inserting record into request_part_values
usr_phst	1001212	Entry not found for APHST_I in sap_message_alias
	1001201	Error in inserting record into message_request
	1001202 1001204	Error in inserting record into request_part
	1001203 1001207	Error in inserting record into request_part_values
usr_prod_mpo1	1001308	Entry not found for APROD_1 in sap_message_alias
	1001301	Error in inserting record into message_request
	1001302 1001304	Error in inserting record into request_part
	1001303	Error in inserting record into request_part_values
usr_prod_mpo2	1001413	Entry not found for APROD_1 in sap_message_alias
	1001402	Error in inserting record into message_request
	1001403 1001404	Error in inserting record into request_part
	1001405 1001406 1001407 1001408	Error in inserting record into request_part_values
usr_prod_mpo3	1001512	Entry not found for APROD_1 in sap_message_alias
	1001502	Error in inserting record into message_request
	1001503	Error in inserting record into request_part

	1001504	
	1001505 1001506 1001507	Error in inserting record into request_part_values
usr_prod_mpo4	1001611	Entry not found for APROD_1 in sap_message_alias
	1001602	Error in inserting record into message_request
	1001603 1001604	Error in inserting record into request_part
	1001605 1001606	Error in inserting record into request_part_values
usr_prod_mpo5	1001709	Entry not found for APROD_1 in sap_message_alias
	1001702	Error in inserting record into message_request
	1001703 1001704	Error in inserting record into request_part
	1001705 1001706 1001707	Error in inserting record into request_part_values
usr_prod_mpo6	1001812	Entry not found for APROD_1 in sap_message_alias
	1001802	Error in inserting record into message_request
	1001803 1001804	Error in inserting record into request_part
	1001805 1001806 1001807	Error in inserting record into request_part_values
usr_prod_mpo7	1001911	Entry not found for APROD_1 in sap_message_alias
	1001902	Error in inserting record into message_request
	1001903 1001904	Error in inserting record into request_part
	1001905 1001906	Error in inserting record into request_part_values

usr_prod_mpo8	1002011	Entry not found for APROD_1 in sap_message_alias
	1002002	Error in inserting record into message_request
	1002003 1002004	Error in inserting record into request_part
Usr_msg_hdr23	1002852	Problem inserting records into msel. Check records in request_part_values for the request_part_id/request_id that is logged
Usr_msg_hdr22	1002702	Problem inserting records into msel. Check records in request_part_values for the request_part_id/ requst_id that is logged.

The following table tells you which of the message formatting routines is used for a message.

Procedure Name	MSHD Trace_flag	MSEL Trace_flag
Usr_msg_hdr	W,V	1
Usr_msg_hdr22	Y	3,4,8,9,o
Usr_msg_hdr23	U	0,6,7
Usr_msg_hdr24	K	L

Chapter 12
Batch Execution Systems

Interfaces exist to Batch Execution Systems such as OpenBatch and iBatch. This chapter will also describe the table structure required so that other batch execution system interfaces can be developed.

To interface a new Batch Execution System you follow the following steps:

- Construct a stored procedure similar that queries the recipe table for any recipes that need to be processed. It will gather data from recipe, material_list, formula, phase. This procedure must set the status in the recipe table with a unique identifier for the destination of the recipe.
- The Subscriber, Subscriber_application and Application tables must be setup. For each SAP/R3 resource_network there is defined a subscriber (the batch execution system). The application that the subscriber will run is defined in the Subscriber_application table as it is setup in the application table.
- Schedule the execution of the new application in the group_master and exec_batch tables.
- There must be an interface that takes the data out of the batch execution system and loads PI.
- The tags to monitor for the control recipe and phases should be setup in the point_group and point_group_members table as usual.

Make a point group for the recipe status information for each resource network within a plant. Set the Group_type = RECIPE and enter the resource network for the resource_id.

In the point_group_members table make one point for the status and one for the recipe_id. The status tag should be a digital state tag with values 00004 = Terminated, 00005 = Processed, 00007 = Discarded, 00001 = started.

Make a point group for each unit with group_type = PI_BATCH and resource_id set to the resource. The points that should be created in the point_group_members table should contain one for the recipe_id in PI_BATCH that is referred to as the Batch_id, one for each phase on the unit, one for the tag which will signal a batch active on the unit, one for the master recipe which in PI_BATCH is referred to as the Product_id. The Phase tags should have the External_phase alias name as the tag alias. The tag_alias for the Product_id should be PRODUCT_ID and the tag_alias for the Batch_id should be BATCH_ID. The tag_alias for the active point for the unit should be ACTIVE.

If there are any other tags you want to group with the unit you can also add these to the point_group_members.

- The following is the mapping of terms between PI_BATCH, PSRLINK and SAP/R3 PP-PI

PI_BATCH	SAP/R3 PP-PI
Unit	Resource
Product_id	Master_recipe
Batch_id	CRID or recipe_id
Phase_1	Unique Phase
Phase_2	Unique Phase
etc.	etc.

In PI_BATCH the Phase tags are made up as digital states for each unique phase that can run on a unit or resource. Only one recipe_id can be running in a Unit or resource at a given time but more than one phase can be running since you can have parallel phases.

- Setup of Subscriber_application Table (This table is only required for interfaces to batch execution systems at this time)

Table Field	***Meaning***
Id	Unique id for subscriber
Application_no	Application number from Application table as used in batch execution systems
Last_timestamp	Not used
Next_timestamp	Not used
Frequency	Not used
Subscribe_type	Not used

- Setup of tables for Alias Descriptions

Table Field	***Meaning***
Alias_class	Class of Alias for example material
Alias_class_desc	Description of class

- Alias_class (This is used for Batch Execution Systems such as Openbatch and iBatch only)

Table Field	***Meaning***
Alias_class	Class of Alias for example material
Alias_class_desc	Description of class

- Alias_system (This table needs to be edited only if new languages are to be added)

Table Field	*Meaning*
Alias_system	System uses the alias for example, SAP/R3, Openbatch, PI etc.
Alias_system_desc	Description of System
Language	Language for the Alias System matches that set up in Location E is English
Plant_id	Plant Id

- External_alias (This table needs to edited only if new languages are added)

Table Field	*Meaning*
Alias_value	Alias value
Internal_vlaue	Internal value used in Plant Suite
Alias_system	Alias System
Alias_class	Alias Class
Alias_description	Description of alias

Alias for Languages

The tables Alias System, Alias Class and External Alias must be configured to have the translations for key values. Openbatch configuration is used as an example in the following tables. Setup the tables as follows:

Alias System

Alias_system_id	Alias_system	Alias_System_ Desc	Language	Plant_id
Enter a unique no	**Enter PID**	**Enter a description for system similar to one shown**	**Enter Language indicator same as used in Location table**	**Enter plant id same as in SAP/R3 and in plant and location tables**
1	PID	Openbatch English	E	1100

Alias Class

Alias_Class	Alias_Class_description
Enter the values as shown below	**Enter the values as shown below**

STATUS	PID Status values
EVENT	PID Event types

External Alias

Alias_value	Internal_value	Alias_System_id	Alias_description	Alias_class
Enter the Openbatch language specific value. This must match exactly the string being used by openbatch	**Enter the English version for the Openbatch string**	**Enter Alias system ID being used for Obenbatch**	**Enter descriptions similar to those shown**	**Enter the corresponding class as shown**
State Change	State Change	1	English for State Change	EVENT
System Message	System Message	1	English for System Message	EVENT
Beginning of Batch	Beginning of Batch	1	English for Beginning of Batch	STATUS
Complete	Complete	1	English for Complete	STATUS
End of Batch	End of Batch	1	English for End of Batch	STATUS

Batch Execution System

A plant that uses a batch execution system should be configured as a BES plant in the plant table.

Material Alias Configuring

Setup of material alias needed for Batch Execution System Translation.

Alias

Table Field	Meaning
Alias_no	Unique no
Alias_type	Set to “PID” if for use with Openbatch
Alias_description	Set to “PID Material” for PID
Alias_class	Set to “Material”

Material Alias

Table Field	Meaning
Material_id	Unique material id, not the material no because this is not unique across phases, use the material short description
Material_alias	Name of the material in the batch execution system
Alias_no	Alias Entry in Alias table
Material_Alias_description	Description of material, the short text in the material instruction

Subscriber and Subscriber Application

In this interface we map a Plant Resource Network to a single Batch Execution System server.

Subscriber

Table Field	Meaning
ID	Unique no
Name	Set to “BATCH” if for use with Openbatch
Address	Set to address name for Batch Execution system Server
Resource_network	Resource Network of recipes which will be sent to this server
Plant_id	Plant which will use this server

Subscriber application

Table Field	Meaning
ID	Unique no
Application_no	Set to number corresponding to the batch execution system application to put a recipe on the batch list
Last_timestamp	Do not use
Next_timestamp	Do not use
Frequency	Do not use
Subscribe_type	Set to 1

Point Groups and Point Group Members

For the stop and start of the recipe and the phases from Openbatch to be sent to PI for archiving and also reviewed in PI-Batch then you must set up the tables for Point_group and Point_group_members as described in the PI and PI Batch specifics chapter. There must be a point group for each phase, unit operation and one point group that will store the recipe information. See the point group section under the configuration application.

Configuration Example

The plant must be configured as a BES type plant

The following tables must be configured

Subscriber

id	name	address	resource_net	plant_id
85	BATCH	machineaddre	TT	1100

Subscriber_application

id	application_	last_timesta	next_timesta	frequency	subscibe_ty
85	1	7/14/1999			1

Application

application_	application_description	program_name	f	fiel	fi	fi	f	f	fi	li	l	l	li
1	BES put control recipe	BES_PROGRAM_NAME											

Material_alias

material_id	material_alias	alias_no	material_alias_desc
SAP	Name of material in BES	4	Description of material as it comes from

Alias_system

alias_system_id	alias_system	alias_system_des	language	plant_id
4	BES_PROGRA	Master_recipe_na	E	1100

Group_master

group	group_desc	batch_no	last_exec_dtime	frequen	frequency_h
41	BES_PROGRAM_group	1	8/9/1999 11:49:04	1	0

Exec_batch

program_name	batc	functionality	exe_	i	group_n	batch_
d:\psrlink\server\fe\recipe_list.exe	1	puts visualbatch batch list	E		41	1
Stored procedure or exe for phase	3	set visualbatch phase	P or E		41	1
Stored procedure or exe for recipe	2	set visual batch recipe	P or		41	1

Translator – this table would be configured as the usual PI-BATCH type of plant.

Program Requirements

To interface to a batch execution system there are usually three programs required. The first program puts the recipe on the batch list for the batch execution system. The remaining two read the status of the recipe and the status of the phase from the batch

execution system and formulate the data which must be set in PI to record this data. Once these programs have been written they must be scheduled to run by configuring them in the tables application, group_master and exec_batch. If you construct all the points in PI as required by the PI-BATCH type of plant then the translator table can be configured as a PI-BATCH type of plant..

Putting the recipe on the Batch Execution System

Application must be configured in Application table for selecting a recipe that will go on batch list. For example call this program recipe_list.exe

This query checks to see if there is a recipe that is waiting to be put on the batch execution system. This query selects the recipe name that came in the SAP OSI_EXTERNAL_RECIPE in the AORD instruction. It also selects the address of the BES server

```
select  @recipe_id      = r.recipe_id,
        @master_rname           = r.master_recipe_name,
        @plant_id       = r.plant_id,
        @name           = s.name,
        @address        = s.address
                from    recipe  r,
                        subscriber s,
                        application a,
                        subscriber_application sa
                where   r.release_status = NULL
                and     r.resource_network      = s.resource_network
                and     s.id                    = sa.id
                and     s.plant_id              = r.plant_id
                and     sa.application_no       = a.application_no
                and     a.program_name                  = "BES_PROGRMA_NAME"
                and     s.address               = @server_name
```

This query would be used to select the values of parameters that are to be set in the recipe. These would have come down in the recipe as APHAPR_1 instructions.

```
Select  @recipe_id,
","+RTRIM(f.parameter_name)+","+convert(char(30),convert(real,f.parameter_value))
                        from    formula f
                        where   f.recipe_id = @recipe_id
```

This query would be used to select the materials that are to be set. The material alias, alias_system and alias tables must be configured

```
select  @recipe_id,
“,”+RTRIM(ma.material_alias)+”,”+convert(char(30),abs(ml.quantity))
                    from    material_list     ml,
                            material_alias    ma,
                            alias_system      asy

                    where   ml.recipe_id              = @recipe_id
                    and     ma.material_id            = ml.material_id
                    and     ma.material_alias_desc =
ml.material_short_text
                    and     ma.alias_no               = asy.alias_system_id
                    and     asy.alias_system_desc     = @master_rname
                    and     asy.alias_system        = “BES_PROGRAM”
                    and     asy.plant_id              = @plant_id
```

After the recipe has been selected for the batch execution system its status must be updated. Set the status to P or F if failure. Serial number is assigned to the number given by the batch execution system if there is one.

```
Update recipe
set     release_status   = @select_flag,
        document         = “Unique Sl No : “ + convert(char,
@serial_no)
where   recipe_id        = @crid

update  subscriber_application
set     last_timestamp           = GETDATE()
from    recipe r,
        subscriber s,
        subscriber_application sa
where   sa.id                    = s.id
and     s.resource_network       = r.resource_network
and     r.recipe_id              = @crid
```

Retrieving the Data for Phase and Recipe Start and End Times

The recipe start and end times must be set by some mechanism. The phase start and end times must be set by some mechanism. One method of handling this is to get the values for these into the PI points defined in the PI_BATCH and RECIPE point groups. The recipe tag and the status tag must be set at the same timestamp. The application that is to retrieve the status of the recipe and the phase is configured as in a PI-BATCH type of plant. Schedule the programs that put values in the PI points with the exec_batch table. These programs are noted above as "Stored procedure or exe for phase" and "Stored procedure or exe for recipe". These programs can make use of the action_send table. If a value is placed in the action_send table it will be sent to PI by the PSRLINK putvalue application.

Retrieving the Remainder of the Data

The remainder of the data would be retrieved as usual from the associated PI points.

Batch Execution System Specifics

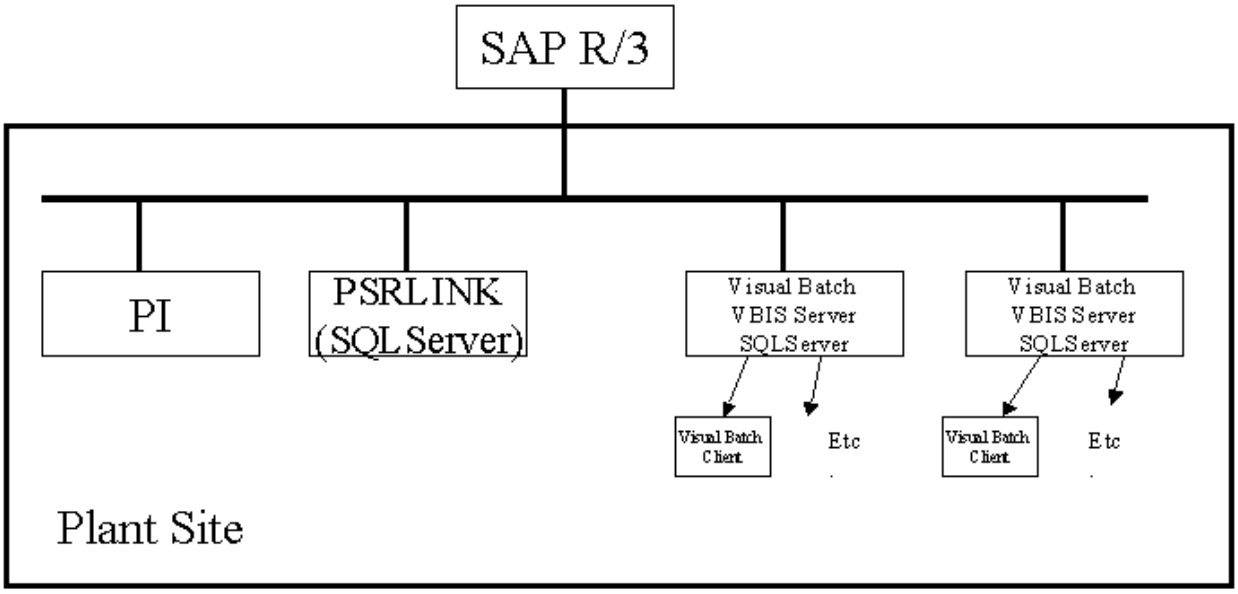
Openbatch

The interface to Openbatch uses the Batchhis table that receives the dump of the event log from Openbatch. When configuring Openbatch you must configure SQLServer to be the destination of the archive log. The archive log is updated upon deletion of the batch from the Openbatch system unless configure otherwise. The SQL script used to configure Batchhis must be the one provided with RLINK not the one provided by Openbatch. This table is installed as part of our installation procedure. A separate script is provided on the CD.

RLINK Interface to Visual Batch

There are two parts to the interface with Visual Batch. First the recipe from SAP must be translated and formatted for input into Visual Batch. The second part is reading the results from Visual Batch formatting the data to be stored in PI and collecting the data from PI and Visual Batch for return to the requested information from SAP.

The system can be configured that one PSRLINK server supports a network of VBIS and Visual Batch servers or the VBIS and SQLServer for Visual Batch could be running on the same machine as PSRLINK. DCOM is used to communicate between RLINK and the distributed VBIS servers. SQLServer database replication is used to keep results from the Visual Batch executions in sink with the PSRLINK server. PI can be located on the same server or a different server depending on the volume of data it is collecting.



The RLINK product install must be completed before installation of the Visual Batch interface. The install for the Visual Batch Interface is available as a separate install on the CD. Install the interface for Visual Batch by executing the the setup program and then the psrlink_vbatch.bat file. This will load the corresponding tables and create the stored procedures.

To run the application without installing VisualBatch you must do the following. Support on this configuration should be obtained from Intellution:

1. Copy VBISPS.DLL to the RLINK machine and register it with regsvr32
2. Open and edit the included file VBIS.REG
3. Modify the InprocServer32 file location so that it is accurate for the VBISPS.DLL file
4. Save your changes and merge the file into the Registry.
5. If you are going to use the Visual Basic IDE you might also want to copy the VBISSRV.TLB file to the machine as well.

You must also configure DCOM on the server by using DCOMCNFG.EXE located in the NT System32 direcory. Select the VisualBatch Integration Service and set up the properties.

Visual Batch tables

The following are the tables that are used to receive the results from the Visual Batch run and transfer those results into RLINK. The data can be gotten into these tables by either having the version of these tables in the PSRLINK database be the database that is directly written into by Visual Batch or by setting up database shadowing from the Visual Batch SQLServer to the RLINK SQLServer. The original SQL script for the creation of these tables that is received from Intellution has been modified in 2 ways. The first is that the name of the table has been changed to lower case. This is done so the code works on an instance of SQLServer which has been installed as case sensitive. The second is that a status column has been added which we use to mark the rows which have been processed by RLINK.

The tables which must be shadowed into RLINK are batch_proc and phase_proc. If you are writing all data into the PSRLINK database and you have a case sensitive version of SQLServer installed then you must change the following tables delivered by Intellution to lowercase, param, batch, unit_proc, and unit_operation_proc.

Batch_proc

```
create table batch_proc
(
        server_name             varchar(64),
        batch_serial_no         int,
        time_stamp              datetime,
        event_serial_no         int,
        batchproc_id            varchar(40),
        event_type              varchar(40),
        event_subtype           varchar(40)     null,
        process_value           varchar(255)    null,
        step_type               varchar(32)     null,
        user_id                 varchar(80)     null,
        execution_counter       int             null,
        batch_id                varchar(40),
        status                  char(1)         null,
        primary key (server_name, batch_serial_no, event_serial_no)
)
```

Phase_proc

```
create table phase_proc
(
        server_name             varchar(64),
        batch_serial_no         int,
        time_stamp              datetime,
        event_serial_no         int,
        phase_id                varchar(40),
        batchproc_id            varchar(40) ,
        unitproc_id             varchar(40) ,
        unitoperproc_id         varchar(40)     null,
        event_type              varchar(40),
        event_subtype           varchar(40),
        process_value           varchar(255)    null,
        step_type               varchar(32)     null,
```

```
        user_id                 varchar(80)     null,
        process_cell                    varchar(255),
        unit                    varchar(255),
        execution_counter           int,
        batch_id                varchar(40)     null,
        phaseproc_id            varchar(40)     null,
        status                  char(1)                 null,
        primary key (server_name, batch_serial_no, event_serial_no)
    )
```

Table configuration

The recipe that is being used to illustrate the setup for the interface configuration is the ICE-Cream recipes from the Interkama 1999 Demonstration.

The plant is setup to be of type BES using the configuration application in RLINK.

plant_id	resource_net	type
1100	R_INT	BES

Subscriber

The subscriber table is used to set up the address for the servers to be used. The BATCH entry is the computer which will be running Visual Batch.

id	name	address	resource_net	plant_id
82	PI	piserver2	R_INT	1100
83	BATCH	GRETCHEN	R_INT	1100

Subscriber_application

The subscriber application table maps the subscriber_id to the application for Visual Batch

id	application_	last_timesta	next_timesta	frequency	subscibe_ty
83	1	6/17/1999			1

Application

There is an entry in the application table for the Visual Batch application

appli	application_description	program_name	field1_name	f
1	VisualBatch put control recipe	VISUALBATCH		

Translator

The translator table is used to setup the method of translation and data retrieval for SAP requests.

request_part_name	request_ca	su	appl	translate_method	reply_metho	r	plant	resou
PPPI_ACTIVITY	APHACT	82	19	usr_phact_activity	WITH_ENG		1100	R_INT
PPPI_BATCH	ACONS_1	82	19	usr_batchid_tag	WITHOUT		1100	R_INT
PPPI_BATCH	APROD_1	82	19	usr_batchid_tag	WITHOUT		1100	R_INT
PPPI_CONFIRMATION_SHORT_T	APHST_I	82	19	usr_confirmation_short_tex	WITHOUT		1100	R_INT
PPPI_CONTROL_RECIPE_STATU	ACRST_I	83	2	usr_recipe_monitor	WITH		1100	R_INT
PPPI_DATA_POINT_VALUE	AREAD1	82	19	usr_read1_monitor	WITH_ENG		1100	R_INT
PPPI_DATA_POINT_VALUE	AREAD2	82	6	usr_read2_monitor	WITH_ENG		1100	R_INT
PPPI_DELIVERY_COMPLETE	APROD_1	82	19	usr_delivery_tag	WITHOUT		1100	R_INT
PPPI_DUMMY	APHACT	82	19	usr_dummy_monitor	WITH		1100	R_INT
PPPI_INSPECTION_RESULT	AQMSMR_1	82	39	usr_qmsmr1_monitor_s1_v	WITH_ENG		1100	R_INT
PPPI_INSPECTION_SHORT_TEX	AQMSMR_1	82	39	usr_qmsmr1_monitor_desc	WITHOUT		1100	R_INT
PPPI_MATERIAL_CONSUMED	ACONS_1	82	19	usr_batch_flow_tag	WITH_ENG		1100	R_INT
PPPI_MATERIAL_PRODUCED	APROD_1	82	19	usr_batch_flow_tag	WITH_ENG		1100	R_INT
PPPI_NUMBER_OF_INSPECTION	AQMSMR_1	82	39	usr_qmsmr1_monitor_no	WITHOUT		1100	R_INT
PPPI_OPERATION_STATUS	AOPST_I	82	25	usr_operation_monitor	WITH		1100	R_INT
PPPI_OPERATION_USER_STATU	AOPUST_I	82	19	usr_operation_monitor_use	WITH		1100	R_INT
PPPI_PARAMETER_NAME	APHPAR_1	82	12	usr_set_alias_tag			1100	R_INT
PPPI_PHASE_RESOURCE	APHST_I	82	64	usr_phase_resource	WITHOUT		1100	R_INT
PPPI_PHASE_STATUS	APHST_I	83	3	usr_phase_alias_monitor	WITH		1100	R_INT
PPPI_PHASE_USER_STATUS	APHUST_I	82	29	usr_phase_monitor_user	WITH		1100	R_INT
PPPI_REASON_FOR_VARIANCE	APHST_I	82	19	usr_reason_for_variance	WITHOUT		1100	R_INT
PPPI_RESERVATION	ACONS_1	82	19	usr_reservation	WITHOUT		1100	R_INT
PPPI_RESERVATION_ITEM	ACONS_1	82	63	usr_rs_and_rsi	WITHOUT		1100	R_INT
PPPI_STANDARD_DEVIATION	AQMSMR_1	82	39	usr_qmsmr1_monitor_dev	WITHOUT		1100	R_INT
PPPI_STORAGE_LOCATION	ACONS_1	82	18	usr_get_location	WITHOUT		1100	R_INT
PPPI_STORAGE_LOCATION	APROD_1	82	18	usr_get_location	WITHOUT		1100	R_INT
PPPI_YIELD_TO_CONFIRM	APHST_I	82	19	usr_yield_to_confirm	WITHOUT_EN		1100	R_INT

Alias_system

The alias system concept is to setup a unique id which will be used to select the set of translations for a given system. In configuring a recipe we have three types of systems. The first type is use to distinguish materials by recipe and plant. For this type the individual alias values will be given in material_alias.

The second type is to setup an alias characteristic for an instruction_characteristic used in the SAP recipe for PPPI_EXTERNAL_PHASE. If this characteristic cannot be used and another is used then the name of the corresponding characteristic will be setup in external_alias.

The last type of alias system is for VBATCH itself which is used to handle the language changes in key fields which are used internally. This alias system is also used to handle the resource changed name.

The only name change which is not handled in the system is the parameter names which are used in the APHAPR instructions in SAP. These must match the name which has been configured in Visual Batch.

alias_system_id	alias_system	alias_system_desc	Language	plant_id
4	VBATCH	VANILLA	E	1100
5	VBATCH	CHOCO	E	1100
6	SAP	SAP PP-PI	E	1100
7	VBATCH	VBATCH	E	1100

External_alias

The first row here illustrates using an alias for the SAP characteristic PPPI_EXTERNAL_PHASE. The entries for the alias_class EVENT and STATUS show setting up a language translation for the key fields needed to translate the output of Visual Batch. The foreign language translation would be put in the column alias_value.

The entries for alias_class of RESOURCE are used to translate the resource as given from SAP into a resource in Visual Batch. The Visual Batch resource name is given in the alias_value column.

alias_value	internal_value	alia	Alias_description	alias_class
IK_EXTERNAL_PHAS	PPPI_EXTERNAL_PHAS	6	PPPI_EXTERNAL_ALI	SAP-PPPI
State Change	State Change	7	English State Change	EVENT
State Command	State Command	7	English State	EVENT
T_VI100	T-VI100	7	Resource	RESOURCE
T_VI200	T-VI200	7	Resource	RESOURCE
T_VI210	T-VI210	7	Resource	RESOURCE
T_VI220	T-VI220	7	Resource	RESOURCE
T_VI230	T-VI230	7	Resource	RESOURCE
COMPLETE	COMPLETE	7	English Complete	STATUS
RUNNING	RUNNING	7	English Running	STATUS
START	START	7	English Start	STATUS

Alias_class

alias_class	alias_class_desc
EVENT	Event types
RESOURCE	Resource
SAP-PPPI	SAP external
STATUS	Status values

Recipe_list

This table is only required if the OSI_EXTERNAL_RECIPE has not been added to the AORD_1 instruction in the recipe. If this is the situation then the recipe name

will be selected by matching the material_id with the material that comes down in the header of the SAP recipe.

osi_external_re	plant_id	material_id	resource_net
CHOCO	1100	T-HV300	R_INT
VANILLA	1100	T-HV100	R_INT

Phase_UP

This table is used to configure which units must be bound in the recipe. For those units which have selected to be bound it will use the UP_phase name as the unit name and it will set the binding to be the resource assigned in the SAP recipe. If the resource name is not the same in SAP as in Visual Batch it will use the name setup in the external alias for that resource.

plant_i	resour	phase_alias	UP_phase	externa	UO_phase	bi
1100	R_INT	FREEZE	UP_FREEZE:1	CHOC	OP_FREEZE:1	Y
1100	R_INT	FREEZE	UP_FREEZE:1	VANILL	OP_FREEZE:1	Y
1100	R_INT	MIX_CHOCOL	UP_MIX_CHOCO	CHOC	OP_MIX_CHOC	N
1100	R_INT	MIX_VANILLA	UP_MIX_VANILL	VANILL	OP_MIX_VANIL	N

Material_alias

The material name may not be the same in SAP as it is in Visual Batch. The material_alias table is used to do the conversion. A material can have a different name in different recipies. The material_alias_desc must match the material_short_text which comes down in the AMAT instructions in the recipe. The alias_no corresponds to an alias_system for the recipe and plant. The material_alias value is the material name in Visual Batch.

alias_sys	alias_system	alias_system_des	Language	plant_id
4	VBATCH	VANILLA	E	1100
5	VBATCH	CHOCO	E	1100

material_id	material_alias	alias_no	material_alias_desc
T-HV100	T_HV100	4	Vanilla mix
T-HV200	T_HV200	4	Ice Cream Mix
T-IC-R2008	T_IC_R2008	4	Vanilla flavor
T-IC-R3006	T_IC_R2006	4	Treated Water
T-HV200	T_HV200	5	Ice Cream Mix
T-HV300	T_HV300	5	Chocolate mix
T-IC-R2007	T_IC_R2007	5	Chocolate Syrup
T-IC-R3006	T_IC_R2006	5	Treated Water

Point_group

A point group must be setup for each phase and one for the recipe. The group_description must match the PPPI_EXTERNAL_PHASE given in the APHASE_1 instruction in the SAP recipe. The resource_id for the phases must match the resource_id in the the APHASE_1 instruction. The point group for the recipe must have a resource_id which matches the plant_resource_network in the AORD instruction of the SAP recipe.

group_nu	description	group_type	Resource_id	plant_id	application_i	owner
359	MIX_VANILLA	PI_BATCH	T-VI100	1100		SAPUSER
360	FREEZE	PI_BATCH	T-VI200	1100		SAPUSER
361	FREEZE	PI_BATCH	T-VI210	1100		SAPUSER
362	FREEZE	PI_BATCH	T-VI220	1100		SAPUSER
363	FREEZE	PI_BATCH	T-VI230	1100		SAPUSER
365	FAT	QM	T-VI200	1100		SAPUSER
366	FAT	QM	T-VI210	1100		SAPUSER
367	FAT	QM	T-VI220	1100		SAPUSER
368	FAT	QM	T-VI230	1100		SAPUSER
369	MIX_CHOCOLATE	PI_BATCH	T-VI100	1100		SAPUSER
370	RECIPE	RECIPE	R_INT	1100		SAPUSER

Point_group_members

The point group members are used to map to the individual PI tags.

group_num	tag_id	tag_alias	display_orde	server	applicat
359	IC_active_T100	ACTIVE	3	piserver2	
359	IC_Recipe_1010	BATCH_ID	1	piserver2	
359	IC_product_1010	PRODUCT_ID	4	piserver2	
359	IC_Phase_1010	SAP	2	piserver2	
360	IC_active_T200	ACTIVE	3	piserver2	
360	IC_Recipe_2010-	BATCH_ID	1	piserver2	
360	IC_product_2010_T	PRODUCT_ID	4	piserver2	
360	IC_Phase_2010_T2	SAP	2	piserver2	
361	IC_active_T210	ACTIVE	3	piserver2	
361	IC_Recipe_2010-	BATCH_ID	1	piserver2	
361	IC_product_2010_T	PRODUCT_ID	4	piserver2	
361	IC_Phase_2010_T2	SAP	2	piserver2	
362	IC_active_T220	ACTIVE	3	piserver2	
362	IC_Recipe_2010-	BATCH_ID	1	piserver2	
362	IC_product_2010_T	PRODUCT_ID	4	piserver2	
362	IC_Phase_2010_T2	SAP	2	piserver2	
363	IC_active_T230	ACTIVE	3	piserver2	
363	IC_Recipe_2010-	BATCH_ID	1	piserver2	
363	IC_product_2010_T	PRODUCT_ID	4	piserver2	
363	IC_Phase_2010_T2	SAP	2	piserver2	
365	IC-FAT_text	DESC	3	piserver2	
365	IC_FAT_LOT	LOT	1	piserver2	
365	IC-FAT_NO	NO	4	piserver2	
365	IC-FAT	S1_V1	2	piserver2	
366	IC-FAT_text	DESC	3	piserver2	
366	IC_FAT_LOT	LOT	1	piserver2	

366	IC-FAT_NO	NO	4	piserver2	
366	IC-FAT	S1_V1	2	piserver2	
367	IC-FAT_text	DESC	3	piserver2	
367	IC_FAT_LOT	LOT	1	piserver2	
367	IC-FAT_NO	NO	4	piserver2	
367	IC-FAT	S1_V1	2	piserver2	
368	IC-FAT_text	DESC	3	piserver2	
368	IC_FAT_LOT	LOT	1	piserver2	
368	IC-FAT_NO	NO	4	piserver2	
368	IC-FAT	S1_V1	2	piserver2	
369	IC_active_T100	ACTIVE	3	piserver2	
369	IC_Recipe_1010	BATCH_ID	1	piserver2	
369	IC_product_1010	PRODUCT_ID	4	piserver2	
369	IC_Phase_1010	SAP	2	piserver2	
370	IC_Recipe_po	PO	3	piserver2	
370	IC_Recipe_id	RECIPE_ID	1	piserver2	
370	IC_Recipe_st	SAP	2	piserver2	

Group_master, exec_batch

The group_master and exec_batch tables are used to setup the execution of the programs required for the Visual Batch interface.

group	group_desc	batch_no	last_exec_dtime	frequency	frequency_
41	VisualBatch	1	6/17/1999 5:31:19	1	0

program_name	batc	Functionality	exe	i	group_n	batch_
usr_vbatch_phase	3	set visualbatch phase	P		41	1
usr_vbatch_recipe	2	set visual batch recipe	P		41	1
d:\rlink\server\fe\batchvb.exe	1	puts visualbatch batch list	E		41	1

Executables

There is on executable batchvb.exe which is a VBIS application that selects the recipe and formualtes it to be placed on the correct VBIS server.

Procedures

Procedure	Purpose	File Name
Usr_vbatch_recipe	Queries data from batch_proc and formulates input to PI and sets the results in PSRLINK	Vbatrcp
Usr_vbatch_phase	Queries phase_proc and formulates the input to PI and sets	Vbatphas

	the results in PSRLINK	
Usr_vbatch	Selects the recipes, material list, resource and formula values for input to Visual Batch	vbatch

Replicating tables from different SQL servers on the VBIS servers

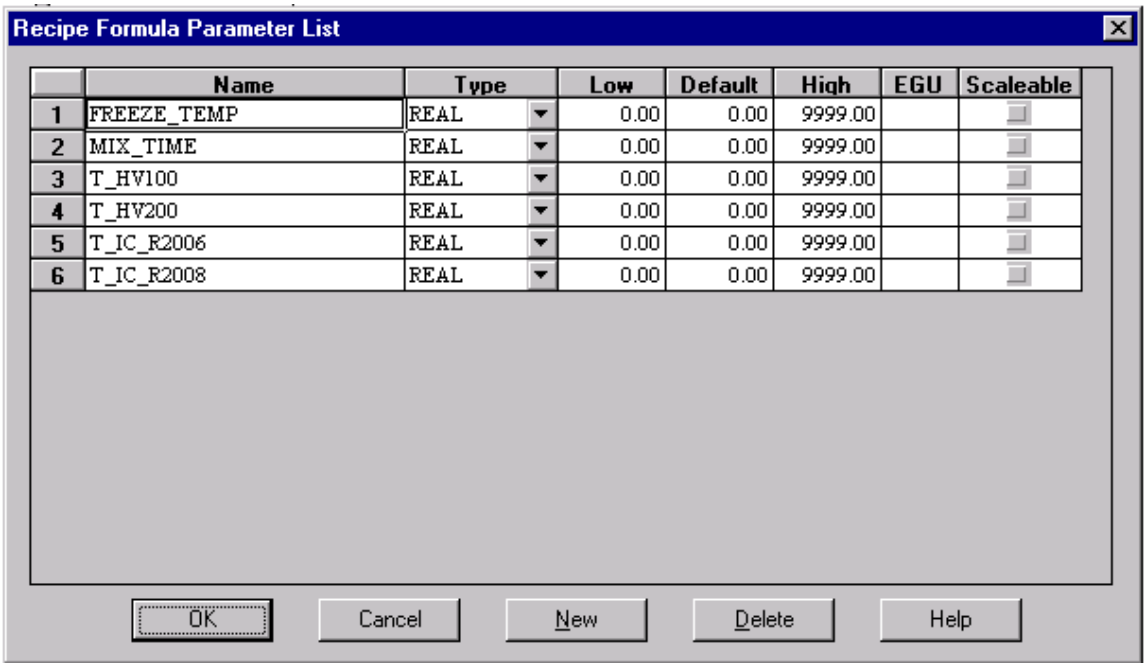
Table replication is required if you are using a separate SQLServer instance for recording your data from Visual Batch than the RLINK database. The tables that must be replicated into the RLINK database are batch_proc and phase_proc. If you need assistence in setting this up see the separate write-up on table replication.

Recipe setup in Visual Batch assumptions

In configuration you should choose Version 4 for the archiving prosess.

The Visual Batch setup requires that the Recipe Formula List be setup to accept the material list as sent down from SAP and that it also include the parameters as sent in APHAPR instructions from SAP.

In this example the first two entries match APHAPR instructions and the rest correspond to the material list in the SAP recipe.



	Name	Type	Low	Default	High	EGU	Scaleable
1	FREEZE_TEMP	REAL	0.00	0.00	9999.00		
2	MIX_TIME	REAL	0.00	0.00	9999.00		
3	T_HV100	REAL	0.00	0.00	9999.00		
4	T_HV200	REAL	0.00	0.00	9999.00		
5	T_IC_R2006	REAL	0.00	0.00	9999.00		
6	T_IC_R2008	REAL	0.00	0.00	9999.00		

PI Tag Configuration

*create material tags, aread and aphapr tags

@table pipoint

@ptclass classic

@mode create,t

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,span,shutdown,compressing

IC_T-HV200,T-HV200 Ice cream mix,kg,L,Float32,10000,off,off,

IC_T-IC-RC2007,T-IC-RC2007 chocolate syrup,g,L,Float32,10000,off,off,

IC_T-IC-R2006,T-IC-R2006,l,L,Float32,10000,off,off,

IC_T-HV300-HV100-T200,T-HV300 OR T-HV100 T-VI200,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T210,T-HV300 OR T-HV100 T-VI210,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T220,T-HV300 OR T-HV100 T-VI220,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T230,T-HV300 OR T-HV100 T-VI230,kg,L,Float32,10000,off,off,

IC_T-IC-R2008,T-IC-R2008,g,L,Float32,10000,off,off,

IC-MIX,Ice cream mix time,MIN,L,Float32,100,off,off,

IC-DENSITY-T200,Ice cream density T200,,L,Float32,100,off,off,

IC-DENSITY-T210,Ice cream density T210,,L,Float32,100,off,off,

IC-DENSITY-T220,Ice cream density T220,,L,Float32,100,off,off,

IC-DENSITY-T230,Ice cream density T230,,L,Float32,100,off,off,

IC-FREEZE-T200,Ice cream freeze temp T200,C,L,Float32,100,off,off,

IC-FREEZE-T210,Ice cream freeze temp T210,C,L,Float32,100,off,off,

IC-FREEZE-T220,Ice cream freeze temp T220,C,L,Float32,100,off,off,

IC-FREEZE-T230,Ice cream freeze temp T230,C,L,Float32,100,off,off,

IC-FAT,Ice cream Fat content,%,L,Float32,100,off,off,

IC-FAT_no,Ice cream Fat content no,,L,Int16,100,off,off,

IC_T-HV200_SAP,T-HV200 Ice cream mix,kg,L,Float32,10000,off,off,

IC_T-IC-RC2007_SAP,T-IC-RC2007 chocolate syrup,g,L,Float32,10000,off,off,

IC_T-IC-R2006_SAP,T-IC-R2006,l,L,Float32,10000,off,off,

IC_T-HV300-HV100-T200_SAP,T-HV300 OR T-HV100 T-VI200 SAP,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T210_SAP,T-HV300 OR T-HV100 T-VI210 SAP,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T220_SAP,T-HV300 OR T-HV100 T-VI220 SAP,kg,L,Float32,10000,off,off,

IC_T-HV300-HV100-T230_SAP,T-HV300 OR T-HV100 T-VI230 SAP,kg,L,Float32,10000,off,off,

IC_T-IC-R2008_SAP,T-IC-R2008 SAP,g,L,Float32,10000,off,off,

@endsection

*create batch_id tags for material , qm lot tag, reservation and reservation_item tags

@table pipoint

@ptclass classic

@mode create,t

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing

IC_HV200_batch_id,Ice Cream HV200 batch_id ,,L,string,off,off,

IC_RC2007_batch_id,Ice Cream RC2007 batch_id ,,L,string,off,off,

IC_RC2008_batch_id,Ice Cream RC2008 batch_id ,,L,string,off,off,

IC_RC2006_batch_id,Ice Cream RC2006 batch_id ,,L,string,off,off,

IC_HV300-HV100-T200_batch_id,Ice Cream HV300-HV100 T200 batch_id ,,L,string,off,off,

IC_HV300-HV100-T210_batch_id,Ice Cream HV300-HV100 T210 batch_id ,,L,string,off,off,

IC_HV300-HV100-T220_batch_id,Ice Cream HV300-HV100 T220 batch_id ,,L,string,off,off,

IC_HV300-HV100-T230_batch_id,Ice Cream HV300-HV100 T230 batch_id ,,L,string,off,off,

IC_FAT_LOT,Ice Cream Fat Lot_no,,L,string,off,off,

IC_HV200_reservation,Ice Cream HV200 reservation ,,L,string,off,off,

IC_HV200_reservation_item,Ice Cream HV200 reservation_item ,,L,string,off,off,

IC_RC2007_reservation,Ice Cream IC_RC2007 reservation ,,L,string,off,off,

IC_RC2007_reservation_item,Ice Cream IC_RC2007 reservation_item ,,L,string,off,off,

IC_RC2008_reservation,Ice Cream IC_RC2008 reservation ,,L,string,off,off,

IC_RC2008_reservation_item,Ice Cream IC_RC2008 reservation_item ,,L,string,off,off,

IC_RC2006_reservation,Ice Cream IC_RC2006 reservation ,,L,string,off,off,

IC_RC2006_reservation_item,Ice Cream IC_RC2006 reservation_item ,,L,string,off,off,

IC_HV300-HV100-T200_reservation,Ice Cream IC_HV300-HV100-T200 reservation ,,L,string,off,off,

IC_HV300-HV100-T200_reservation_item,Ice Cream IC_HV300-HV100-T200 reservation_item ,,L,string,off,off,

IC_HV300-HV100-T210_reservation,Ice Cream IC_HV300-HV100-T210 reservation ,,L,string,off,off,

IC_HV300-HV100-T210_reservation_item,Ice Cream IC_HV300-HV100-T210 reservation_item ,,L,string,off,off,

IC_HV300-HV100-T220_reservation,Ice Cream IC_HV300-HV100-T220 reservation ,,L,string,off,off,

IC_HV300-HV100-T220_reservation_item,Ice Cream IC_HV300-HV100-T220 reservation_item ,,L,string,off,off,

IC_HV300-HV100-T230_reservation,Ice Cream IC_HV300-HV100-T230 reservation ,,L,string,off,off,

IC_HV300-HV100-T230_reservation_item,Ice Cream IC_HV300-HV100-T230 reservation_item ,,L,string,off,off,

```
IC-FAT_text,Ice cream Fat content short text,,L,string,off,off,
@endsection
* create products
@table pipoint
@ptclass classic
@mode create,t
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing
IC_product_recipe,Ice cream product_recipe,,L,string,off,off,
IC_product_1010,Ice cream product phase 1010,,L,string,off,off,
IC_product_2010_T200,Ice cream product phase 2010 T-VI200,,L,string,off,off,
IC_product_2010_T210,Ice cream product phase 2010 T-VI210,,L,string,off,off,
IC_product_2010_T220,Ice cream product phase 2010 T-VI220,,L,string,off,off,
IC_product_2010_T230,Ice cream product phase 2010 T-VI230,,L,string,off,off,
@endsection
*create digital states
@table pids
@mode create,t
@istructure set, state, ...
recipe4,00000,00001,00002,00003,00004,00005,00006,00007
phase4,00000,00001,00002,00003,00004
@endsection
*create SAP states
@table pipoint
@mode create,t
@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing
IC_Recipe_st,Ice cream recipe status,recipe4,digital,off,off,
IC_Phase_1010,Ice cream phase 1010 tank T-VI100,phase4,digital,off,off,
IC_Phase_2010_T200,Ice cream status phase 2010 tank T-VI200,phase4,digital,off,off,
IC_Phase_2010_T210,Ice cream status phase 2010 tank T-VI210,phase4,digital,off,off,
IC_Phase_2010_T220,Ice cream status phase 2010 tank T-VI220,phase4,digital,off,off,
IC_Phase_2010_T230,Ice cream status phase 2010 tank T-VI230,phase4,digital,off,off,
@endsection
*create recipe tags
@table pipoint
@ptclass classic
@mode create,t
```

```
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing
IC_Recipe_id,Ice cream Recipe_id,,L,string,off,off,
IC_Recipe_po,Ice cream Recipe_process_order,,L,string,off,off,
IC_Recipe_1010,Ice Cream Phase Recipe_id 1010,,L,string,off,off,
IC_Recipe_2010-T200,Ice Cream Phase Recipe_id 2010 T200,,L,string,off,off,
IC_Recipe_2010-T210,Ice Cream Phase Recipe_id 2010 T210,,L,string,off,off,
IC_Recipe_2010-T220,Ice Cream Phase Recipe_id 2010 T220,,L,string,off,off,
IC_Recipe_2010-T230,Ice Cream Phase Recipe_id 2010 T230,,L,string,off,off,
@endsection

*create Active states
@table pipoint
@mode create,t
@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing
IC_active_T200,Active T200,phase4,digital,off,off,
IC_active_T210,Active T210,phase4,digital,off,off,
IC_active_T220,Active T220,phase4,digital,off,off,
IC_active_T230,Active T230,phase4,digital,off,off,
IC_active_T100,Active T100,phase4,digital,off,off,
@endsection

* create the unit
@table pibaunit
@mode create,t
@istr unitname,activetag,bidexpr,prodexpr,description,activetype
MIX,IC_active_T100,'"IC_Recipe_1010"','"IC_product_1010"',"MixTI100",pulse,
FREEZE-T200,IC_active_T200,'"IC_Recipe_2010-
T200"','"IC_product_2010_T200"',"Freeze TI-200",pulse,
FREEZE-T210,IC_active_T210,'"IC_Recipe_2010-
T210"','"IC_product_2010_T210"',"Freeze TI-210",pulse,
FREEZE-T220,IC_active_T220,'"IC_Recipe_2010-
T220"','"IC_product_2010_T220"',"Freeze TI-220",pulse,
FREEZE-T230,IC_active_T230,'"IC_Recipe_2010-
T230"','"IC_product_2010_T230"',"Freeze TI-230",pulse,
@endsection
```

Chapter 13
PI and PI-Batch

PI Database configuration for Sample Color Recipe

The following highlights the PI configuration for a typical recipe. The tag configuration for the SAP certification recipe will be shipped with the product, please reference this for complete tag configuration information.

```
@table pipoint
@ptclass classic
@mode create
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,span,shutdown,compressing
color001,CONS 300-110,L,L,Float32,10000,off,off
color002,CONS 300-120,KG,L,Float32,10000,off,off
color003,CONS 300-130,KG,L,Float32,10000,off,off
color004,CONS 300-140,KG,L,Float32,10000,off,off
color005,CONS 300-150,KG,L,Float32,10000,off,off
color006,CONS 300-160,KG,L,Float32,10000,off,off
color007,PROD 300-160,KG,L,Float32,10000,off,off
color008,CONS 300-170,KG,L,Float32,10000,off,off
color009,CONS 300-180,KG,L,Float32,10000,off,off
color010,CONS 300-190,KG,L,Float32,10000,off,off
color011,CONS 300-200,KG,L,Float32,10000,off,off
color012,CONS 300-210,KG,L,Float32,10000,off,off
color013,PROD Y-300,KG,L,Float32,10000,off,off
color014,PROD P-300,KG,L,Float32,10000,off,off
color015,CONS 300-220,KG,L,Float32,10000,off,off
color016,PROD 300-230,L,L,Float32,10000,off,off
```

color017,CONS 300-240,ST,L,Float32,10000,off,off

color018,Mix time set phase 1020,MIN,L,Float32,100,off,off

color019,Density read phase 1030,KG/M3,L,Float32,100,off,off

color020,PH set phase 2010,PH,L,Float32,100,off,off

color021,PH read end of phase 2010,PH,L,Float32,100,off,off

color022,PH set phase 2030,PH,L,Float32,100,off,off

color023,PH read end of phase 2030,PH,L,Float32,100,off,off

color024,Temp read phase 3020,C,L,Float32,500,off,off

color025,Temp set phase 4010,C,L,Float32,500,off,off

color026,Inspec char 10,,L,Float32,100,off,off

@endsection

- create products

@table pipoint

@ptclass classic

@mode create

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing

Product_recipe,product_recipe,,L,string,off,off

product_1111,product phase 1111,,L,string,off,off

product_1121,product phase 1121,,L,string,off,off

product_1131,product phase 1131,,L,string,off,off

product_1141,product phase 1141,,L,string,off,off

@endsection

*create digital states

@table pids

@mode create

@istructure set, state, ...

recipe4,00000,00001,00002,00003,00004,00005,00006,00007

phase4,00000,00001,00002,00003,00004

@endsection

@table pipoint

@mode create

@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing

RecipeN_1190_st,recipe status,recipe4,digital,off,off

Phase1N_1111,phase 1010,phase4,digital,off,off

Phase2N_1111,phase 1020,phase4,digital,off,off

Phase3N_1111,phase 1030,phase4,digital,off,off

```
Phase1N_1121,phase 2010,phase4,digital,off,off
Phase2N_1121,phase 2020,phase4,digital,off,off
Phase3N_1121,phase 2030,phase4,digital,off,off
Phase4N_1121,phase 2040,phase4,digital,off,off
Phase1N_1131,phase 3010,phase4,digital,off,off
Phase2N_1131,phase 3020,phase4,digital,off,off
Phase3N_1131,phase 3030,phase4,digital,off,off
Phase4N_1131,phase 3040,phase4,digital,off,off
Phase1N_1141,phase 4010,phase4,digital,off,off
Phase2N_1141,phase 4020,phase4,digital,off,off
Phase3N_1141,phase 4030,phase4,digital,off,off
@endsection
@table pipoint
@ptclass classic
@mode create
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing
RecipeN_1190_id,Recipe_id R_1190,,L,string,off,off
RecipeN_1111,Phase Recipe_id R_1111,,L,string,off,off
RecipeN_1121,Phase Recipe_id R_1121,,L,string,off,off
RecipeN_1131,Phase Recipe_id R_1131,,L,string,off,off
RecipeN_1141,Phase Recipe_id R_1141,,L,string,off,off
@endsection
*create SAP states
@table pipoint
@mode create
@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing
Recipe_1190_st,SAP recipe status,recipe4,digital,off,off
Phase1_1111,phase 1010 SAP,phase4,digital,off,off
Phase2_1111,phase 1020 SAP,phase4,digital,off,off
Phase3_1111,phase 1030 SAP,phase4,digital,off,off
Phase1_1121,phase 2010 SAP,phase4,digital,off,off
Phase2_1121,phase 2020 SAP,phase4,digital,off,off
Phase3_1121,phase 2030 SAP,phase4,digital,off,off
Phase4_1121,phase 2040 SAP,phase4,digital,off,off
Phase1_1131,phase 3010 SAP,phase4,digital,off,off
Phase2_1131,phase 3020 SAP,phase4,digital,off,off
```

Phase3_1131,phase 3030 SAP,phase4,digital,off,off

Phase4_1131,phase 3040 SAP,phase4,digital,off,off

Phase1_1141,phase 4010 SAP,phase4,digital,off,off

Phase2_1141,phase 4020 SAP,phase4,digital,off,off

Phase3_1141,phase 4030 SAP,phase4,digital,off,off

@endsection

- create the unit

@table pibaunit

@mode create

@istr unitname,activetag,bidexpr,prodexpr,description,activetype

CHARGE11,Phase1N_1111,"'RecipeN_1111'","'product_1111'","Charge R_1111",pulse

PREPARE1,Phase1N_1121,"'RecipeN_1121'","'product_1121'","Prepare R_1121",pulse

ADJUST1,Phase2N_1111,"'RecipeN_1111'","'product_1111'","Adjust R_1111",pulse

TRANSFER1,Phase3N_1111,"'RecipeN_1111'","'product_1111'","Transfer R_1111",pulse

CHARGE3,Phase2N_1121,"'RecipeN_1121'","'product_1121'","Charge R_1121",pulse

REACTION1,Phase3N_1121,"'RecipeN_1121'","'product_1121'","Reactor R_1121",pulse

DISCHARGE1,Phase4N_1121,"'RecipeN_1121'","'product_1121'","Discharge R_1121",pulse

CHARGE2,Phase1N_1131,"'RecipeN_1131'","'product_1131'","Charge R_1131",pulse

HEAT1,Phase2N_1131,"'RecipeN_1131'","'product_1131'","Heat R_1131",pulse

CONDENS1,Phase3N_1131,"'RecipeN_1131'","'product_1131'","Condensor R_1131",pulse

DISCHARG1,Phase4N_1131,"'RecipeN_1131'","'product_1131'","Discharge R_1131",pulse

DRY1,Phase1N_1141,"'RecipeN_1141'","'product_1141'","Dryer R_1141",pulse

CHARGE2F,Phase2N_1141,"'RecipeN_1141'","'product_1141'","Charge R_1141",pulse

DISCHARG2,Phase3N_1141,"'RecipeN_1141'","'product_1141'","Discharge R_1141",pulse

@endsection

@exit

If you want the operations to be saved as points and the operations to be units in PI-BATCH then the following would be a sample of the point configuration.

- create products

@table pipoint

@ptclass classic

@mode create

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing

OP_1111_PD,product OPERATION 1000,,L,string,off,off

OP_1121_PD,product OPERATION 2000,,L,string,off,off

OP_1131_pd,product OPERATION 3000,,L,string,off,off

OP_1141_PD,product OPERATION 4000,,L,string,off,off

@endsection

@table pipoint

@mode create

@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing

OP_1111_AC,OPERATION 1000,phase4,digital,off,off

OP_1121_AC,OPERATION 2000,phase4,digital,off,off

OP_1131_AC,OPERATION 3000,phase4,digital,off,off

OP_1141_AC,OPERATION 4000,phase4,digital,off,off

@endsection

@table pipoint

@ptclass classic

@mode create

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing

OP_1111_ID,OPERATION Recipe_id R_1111,,L,string,off,off

OP_1121_ID,OPERATION Recipe_id R_1121,,L,string,off,off

OP_1131_ID,OPERATION Recipe_id R_1131,,L,string,off,off

OP_1141_ID,OPERATION Recipe_id R_1141,,L,string,off,off

@endsection

*create SAP states

@table pipoint

@mode create

@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing

OP_1111_ST,OPERATION 1000 SAP,phase4,digital,off,off

OP_1121_ST,OPERATION 2000 SAP,phase4,digital,off,off

OP_1131_ST,OPERATION 3000 SAP,phase4,digital,off,off

OP_1141_ST,OPERATION 4000 SAP,phase4,digital,off,off

@endsection

- create the unit

@table pibaunit

@mode create

@istr unitname,activetag,bidexpr,prodexpr,description,activetype

R_1111,OP_1111_AC,"'OP_1111_ID'","'OP_1111_PD'","R_1111",pulse

R_1121,OP_1121_AC,"'OP_1121_ID'","'OP_1121_PD'","R_1121",pulse

R_1131,OP_1131_AC,"'OP_1131_ID'","'OP_1131_PD'","R_1131",pulse

R_1141,OP_1141_AC,"'OP_1141_ID'","'OP_1141_PD'","R_1141",pulse

@endsection

@exit

If you are going to use the flow totalization function then you must configure your points with the totalcode and the convers parameters. The totalcode is set to 0 and the convers parmeter is set according to the conversion factor required. In this example the conversion was from hours and thus used 24. This is required if you are using the summary application.

@table pipoint

@ptclass classic

@mode create,t

@stype delimited

@istr
tag,descriptor,EngUnits,pointsource,pointtype,span,shutdown,compressing,totalcode,convers,

color001,CONS 300-110,L,L,Float32,10000,off,off,0,24,

color002,CONS 300-120,KG,L,Float32,10000,off,off,0,24,

color003,CONS 300-130,KG,L,Float32,10000,off,off,0,24,

color004,CONS 300-140,KG,L,Float32,10000,off,off,0,24,

color005,CONS 300-150,KG,L,Float32,10000,off,off,0,24,

color006,CONS 300-160,KG,L,Float32,10000,off,off,0,24,

color007,PROD 300-160,KG,L,Float32,10000,off,off,0,24,

color008,CONS 300-170,KG,L,Float32,10000,off,off,0,24,

color009,CONS 300-180,KG,L,Float32,10000,off,off,0,24,

color010,CONS 300-190,KG,L,Float32,10000,off,off,0,24,

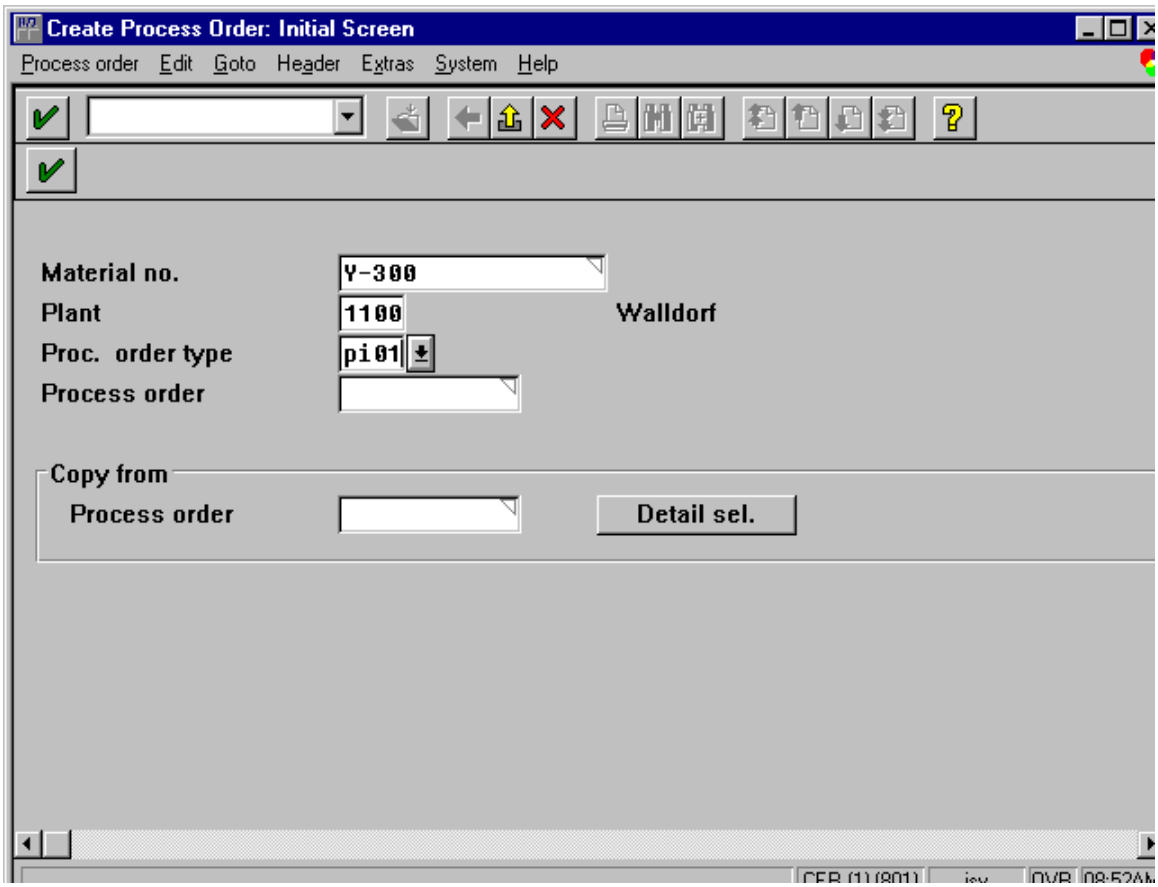
```
color011,CONS 300-200,KG,L,Float32,10000,off,off,0,24,
color012,CONS 300-210,KG,L,Float32,10000,off,off,0,24,
color013,PROD Y-300,KG,L,Float32,10000,off,off,0,24,
color014,PROD P-300,KG,L,Float32,10000,off,off,0,24,
color015,CONS 300-220,KG,L,Float32,10000,off,off,0,24,
color016,PROD 300-230,L,L,Float32,10000,off,off,0,24,
color017,CONS 300-240,ST,L,Float32,10000,off,off,0,24,
@endsection
@exit
```

Chapter 14
SAP/R3 Recipe

The following are screen captures demonstrate the creation of a recipe in SAP/R3 and sending it to RLINK.

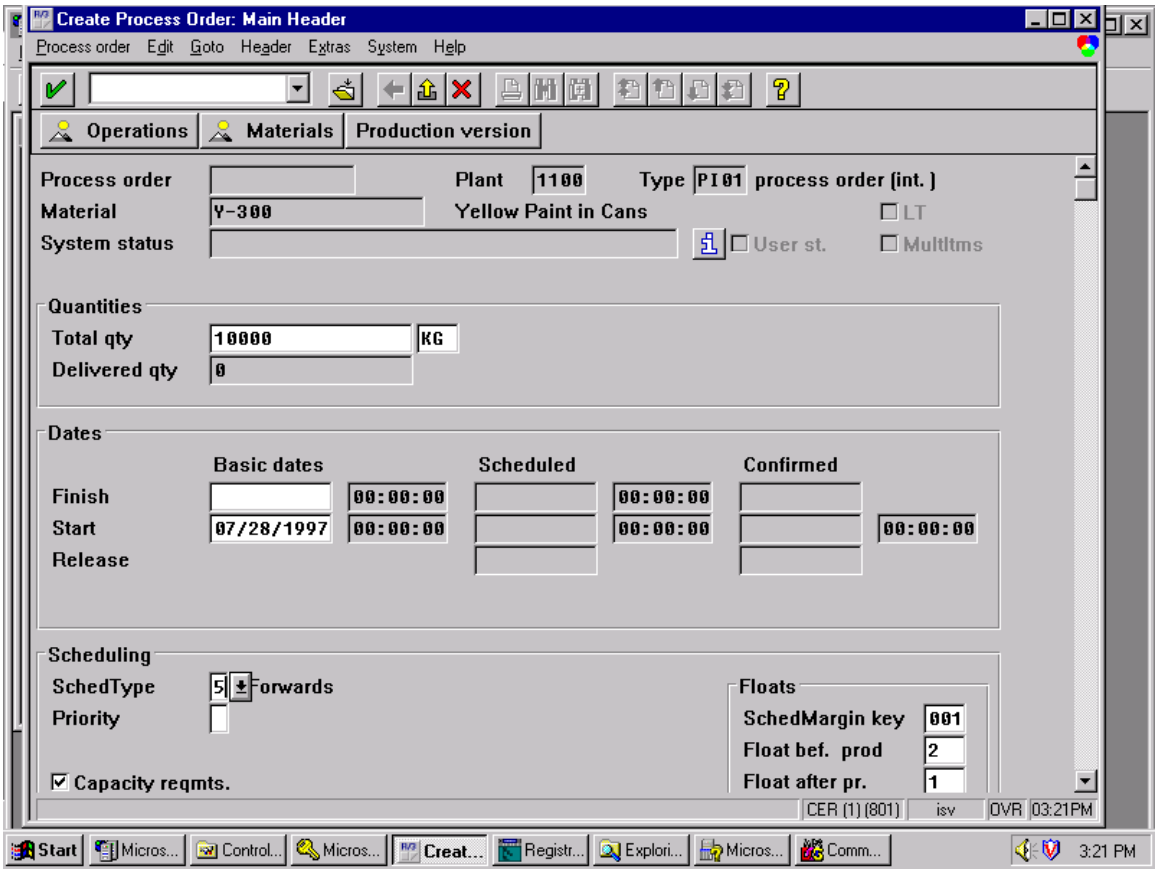
This screen is the first in creation of a process order.

You reach this screen through the menu Logistics, Process Management, Process Order.



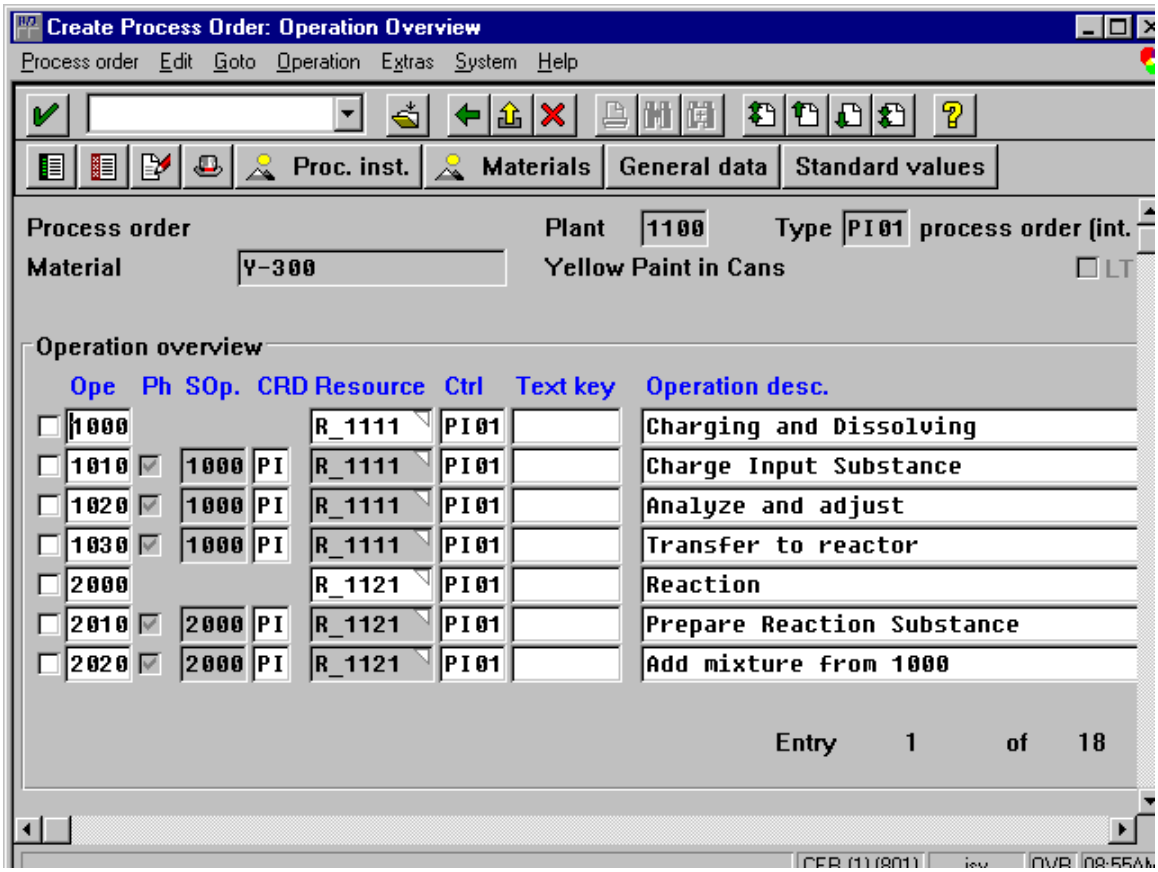
We will illustrate creating a Process Order for the material Y-300 in plant 1100.

The following screen shows entering the quantity and the date and time for the process order.



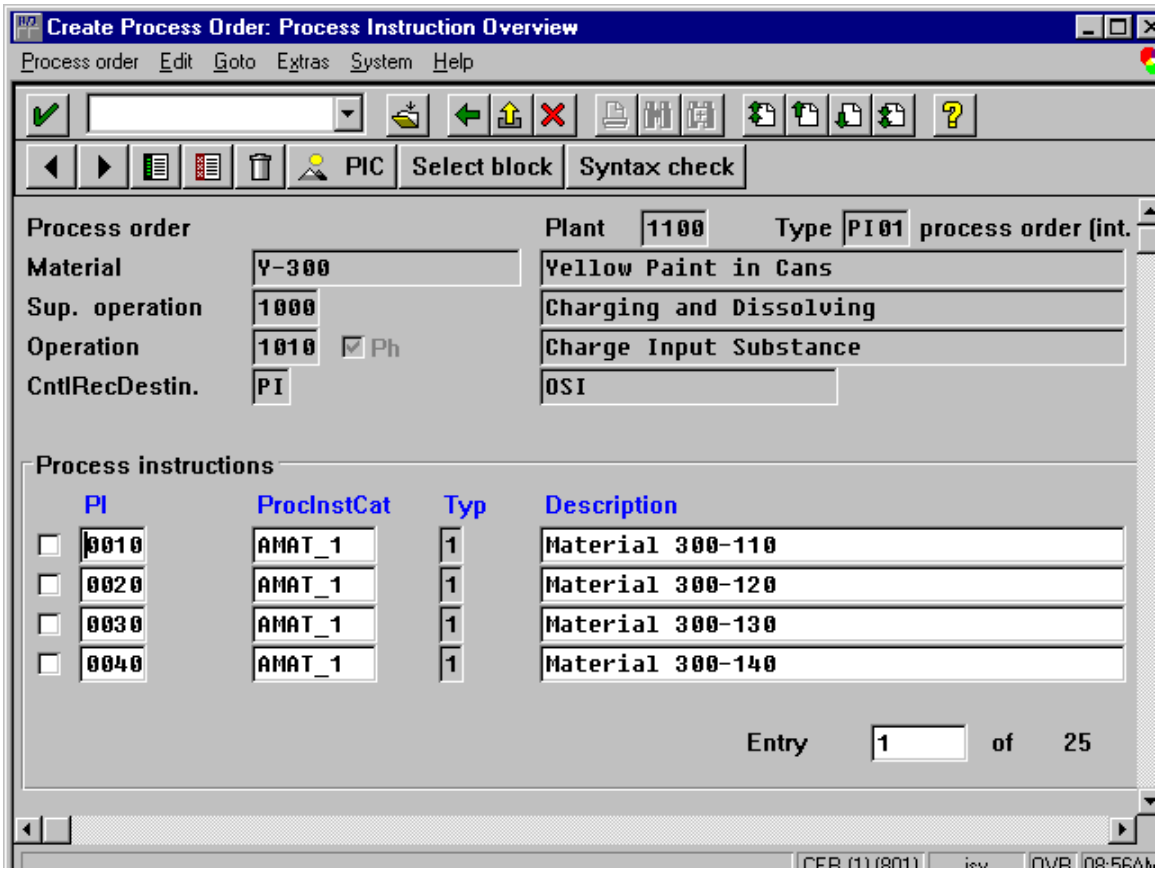
You enter the quantity, start date and select the production version. After the master recipe is copied in the time of the recipe start is reset to 00:00:00 so the time must be entered again. In version 3.1G the definition of the scheduling has been changed in SAP/R3. In order to enter the start time you must select a scheduling option of 5 or 6.

The following screen shows the operations and phases for the process order.



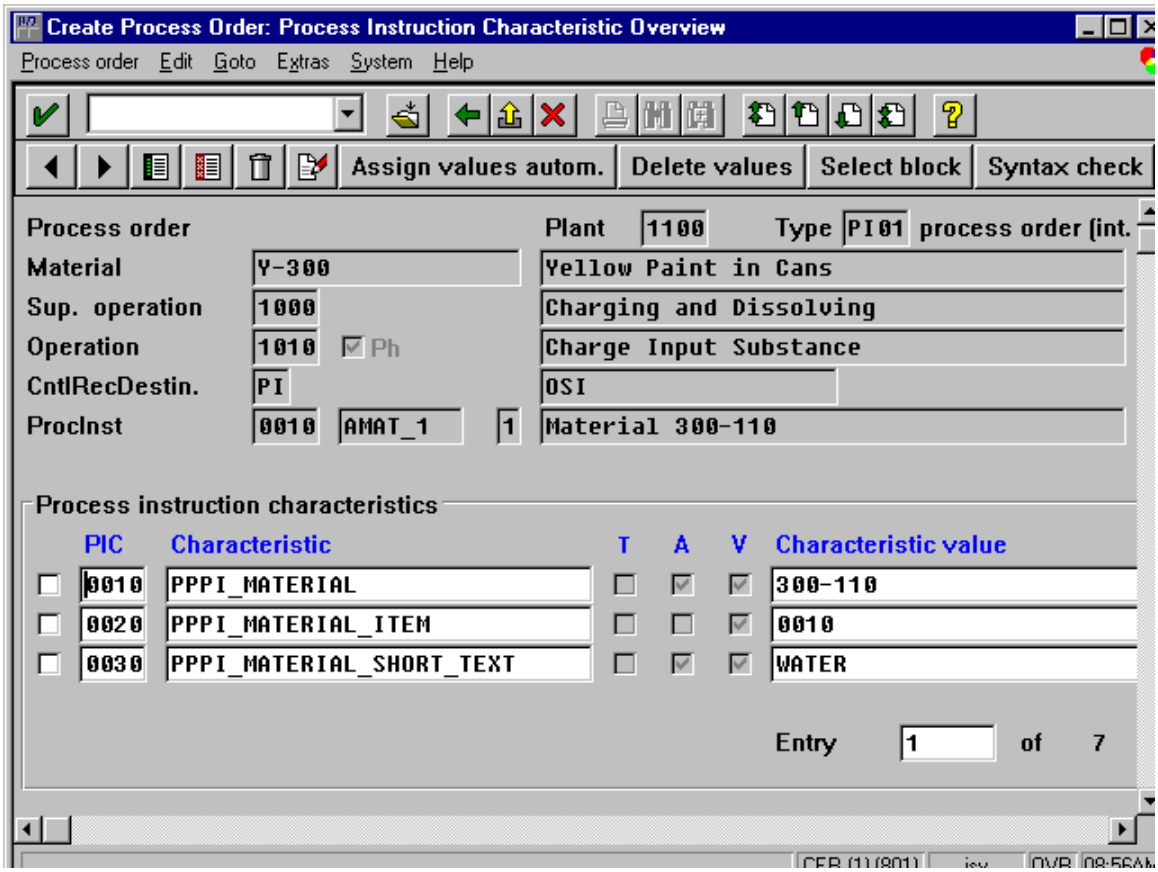
We will select phase 1010 to look at in further detail. The following screen shows the list of process instructions for phase 1010.

Any modifications to the recipe are entered now. Any changes in instruction text are entered now.



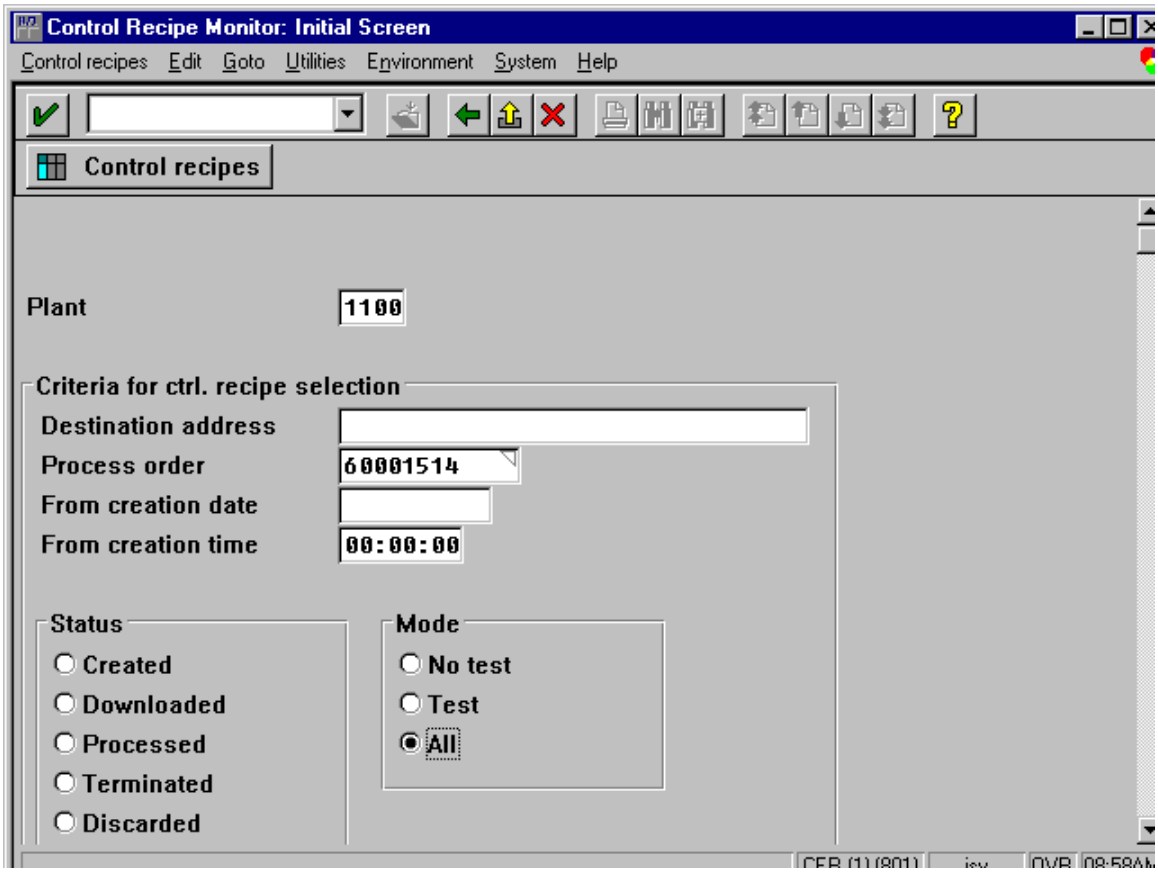
We will select the first process instruction AMAT_1 for material 300-110 to look at in detail.

The following screen shows the process instruction characteristics for this instruction.

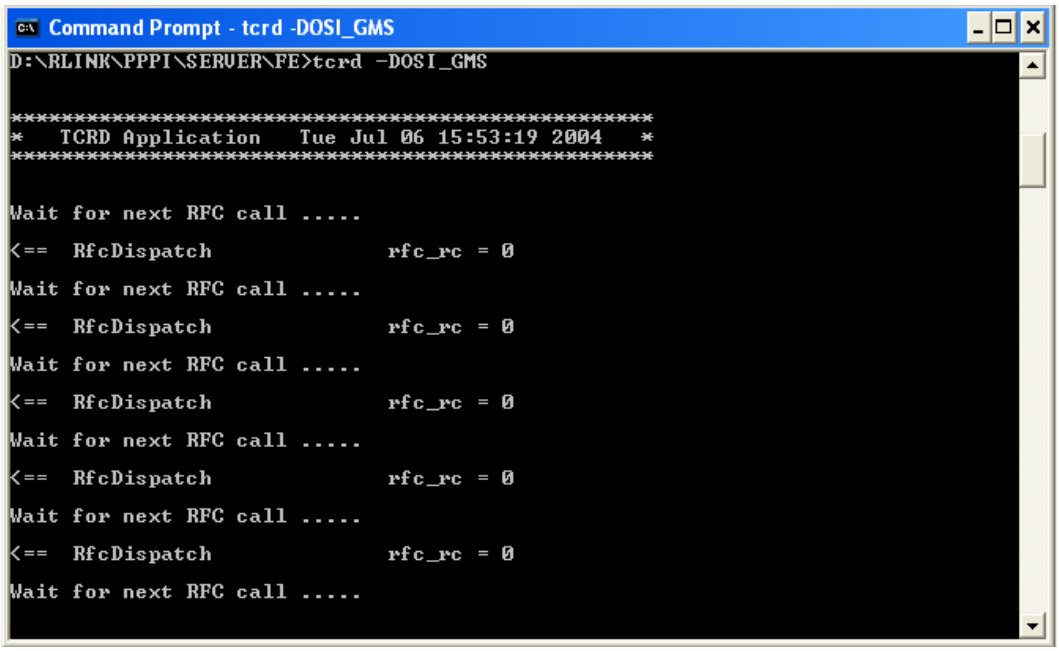


After making any corrections to the recipe the recipe is released, a control recipe is generated and the process order is saved.

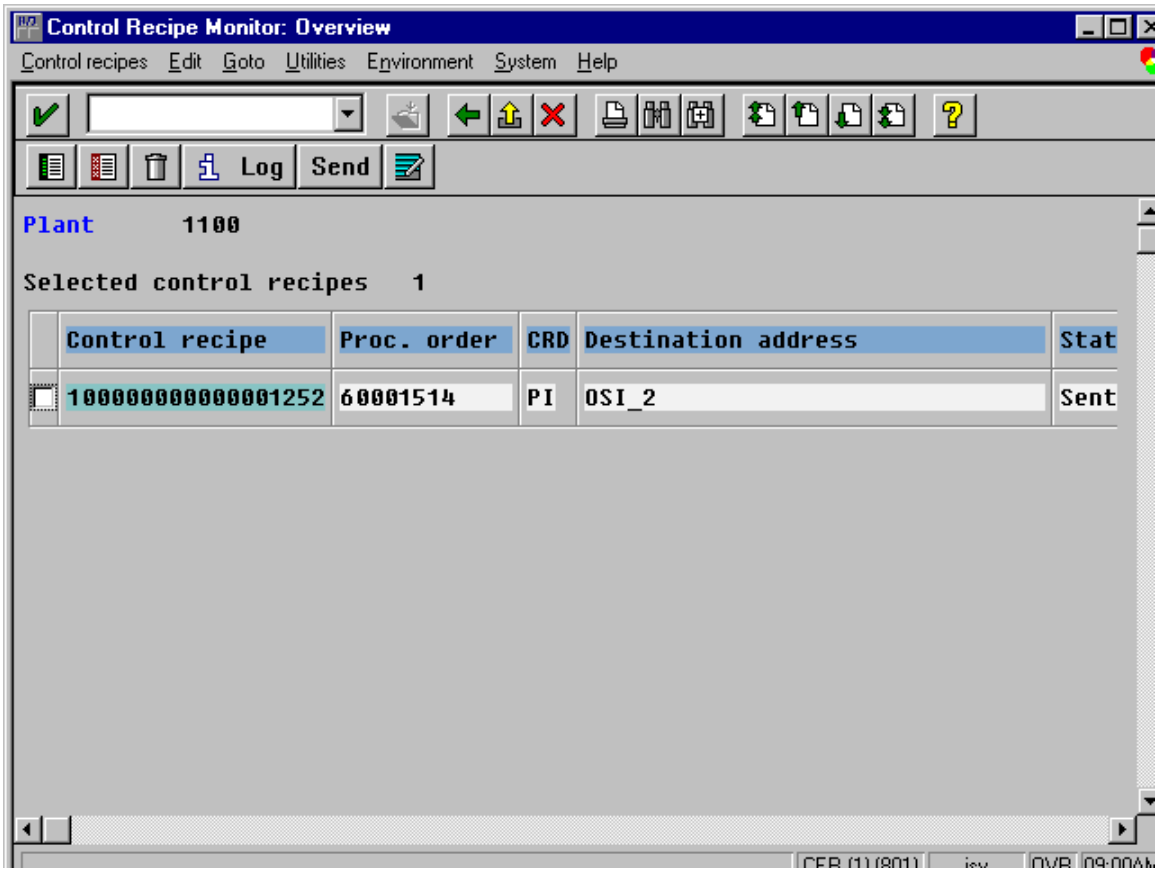
We then switch to the control Recipe Monitor Screen (/nCO53) shown below.



Switching to the RLINK product the following screen shows the background process of TCRD that has established a connection to SAP/R3.

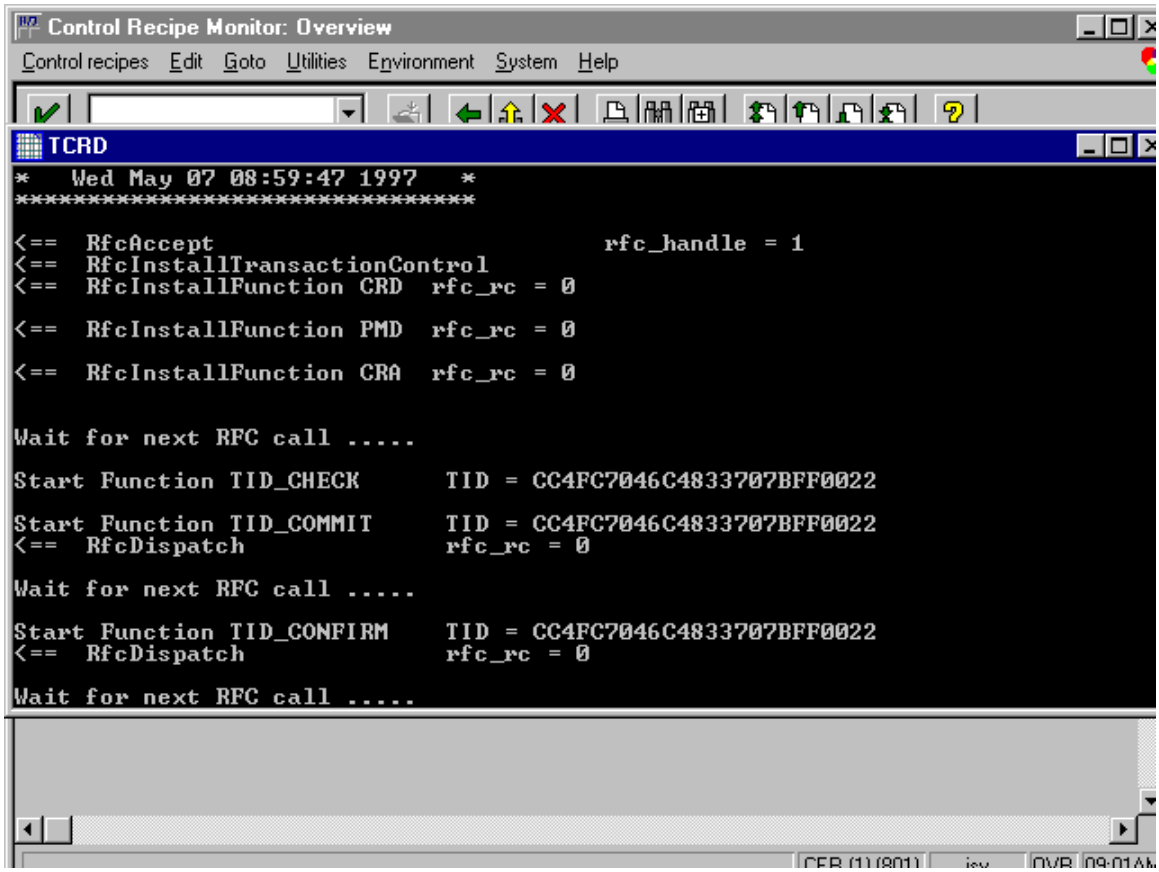


Back to SAP/R3 we now prepare to send the control recipe that has been created down to RLINK.

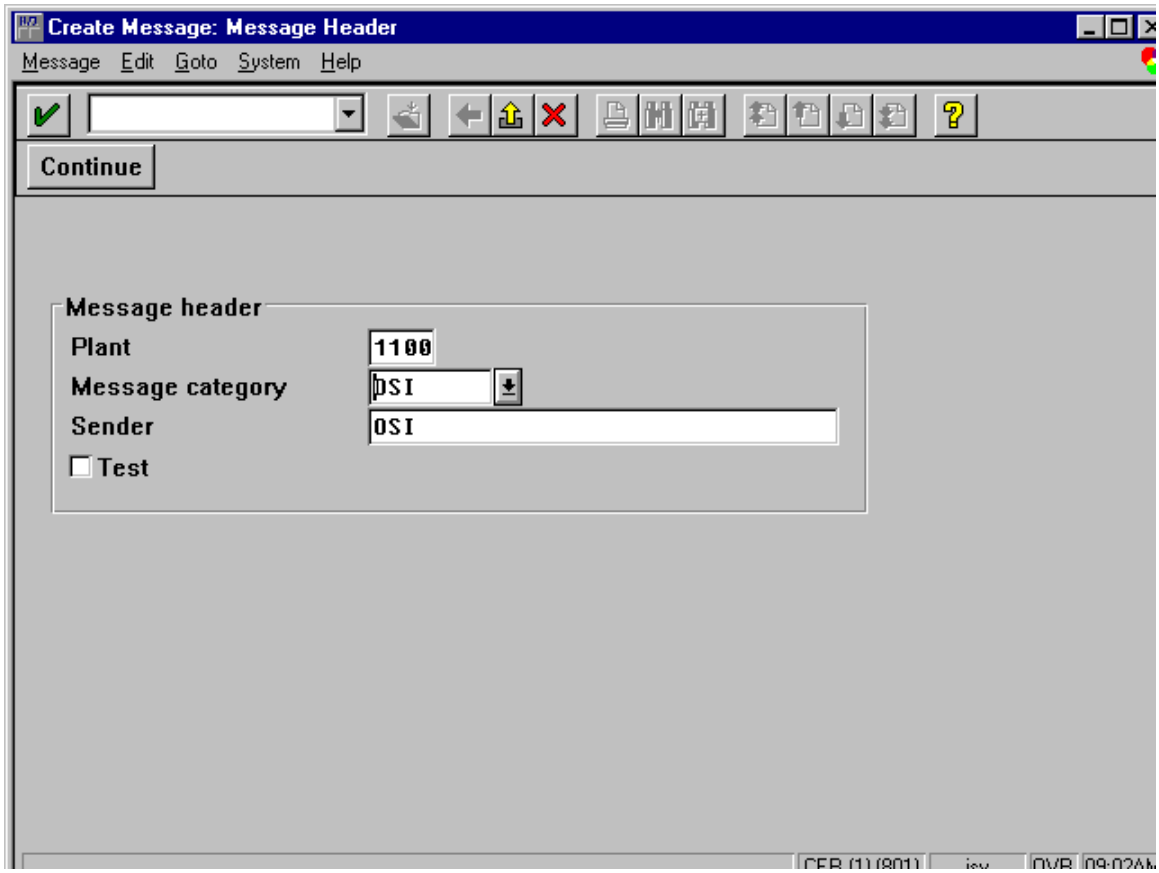


Switching once again back to RLINK we see the transactions have been processed to receive the recipe.

If there is an error in transmission it is found by looking in Environment, tRFCLog. A recipe can be resent by selecting the process and then Edit and execute the program.

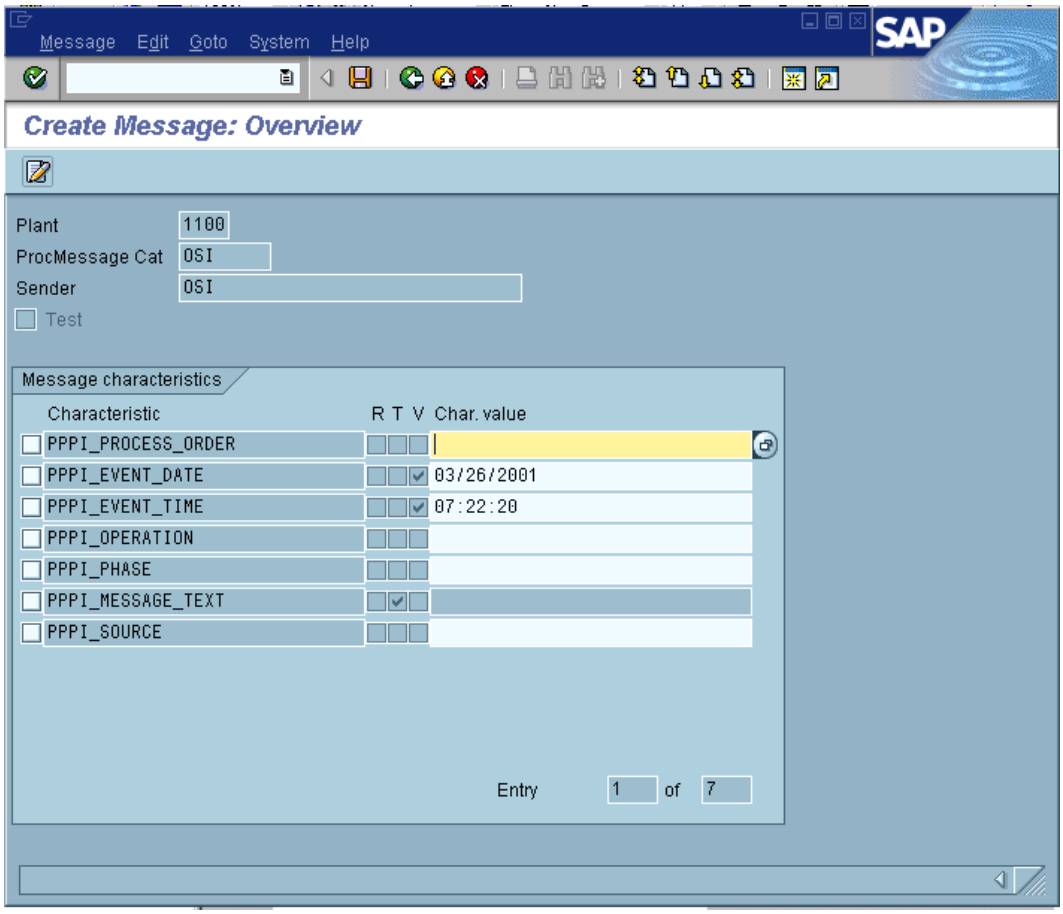


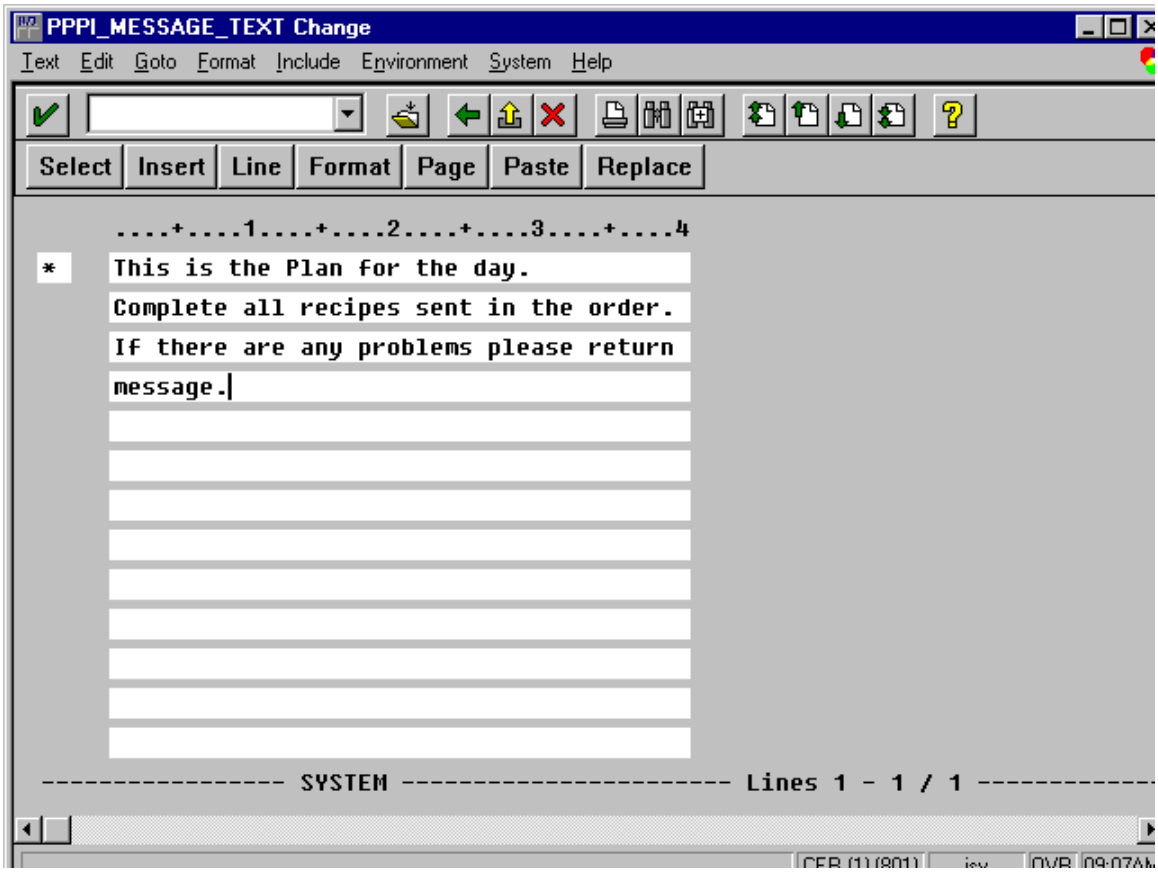
We will now illustrate how a message is sent from SAP/R3 to RLINK. From the Message Monitor SAP/R3 screen we will create a new message.



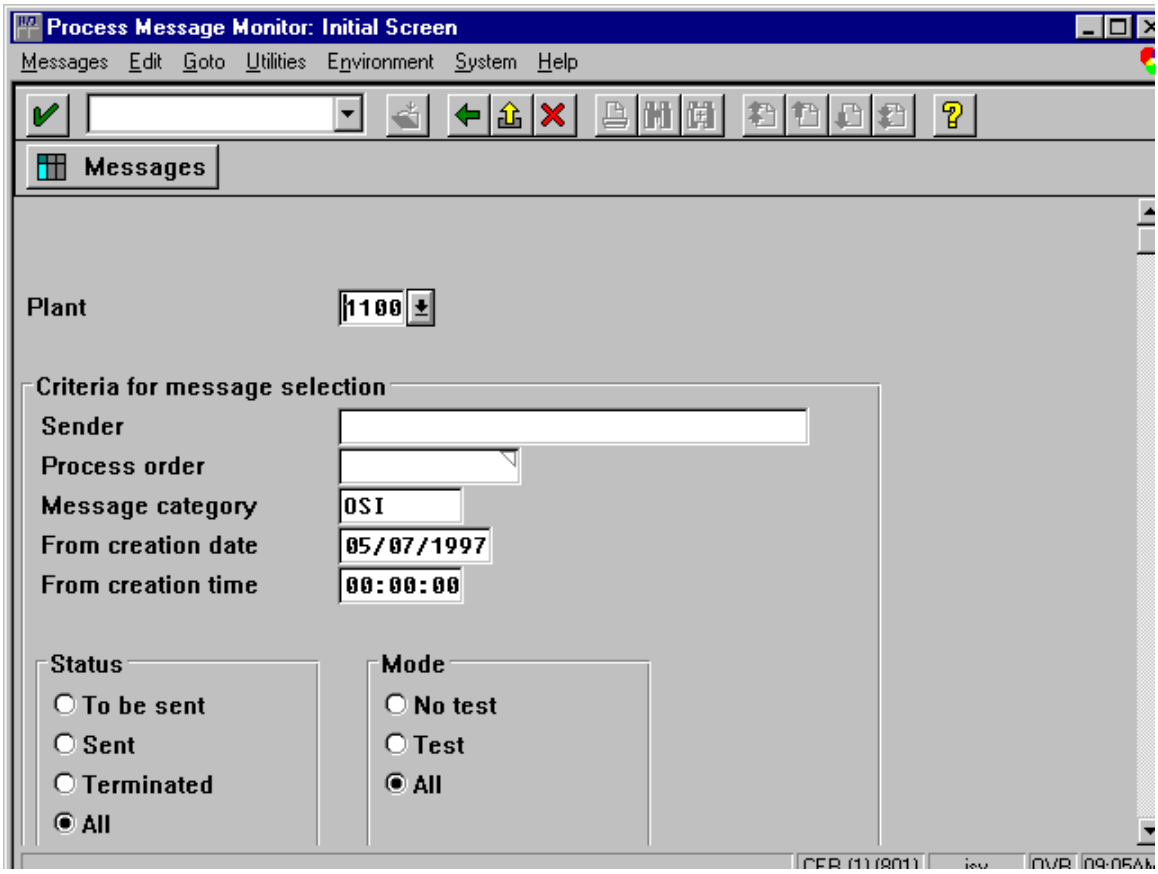
The message is entered using transaction /nco57.

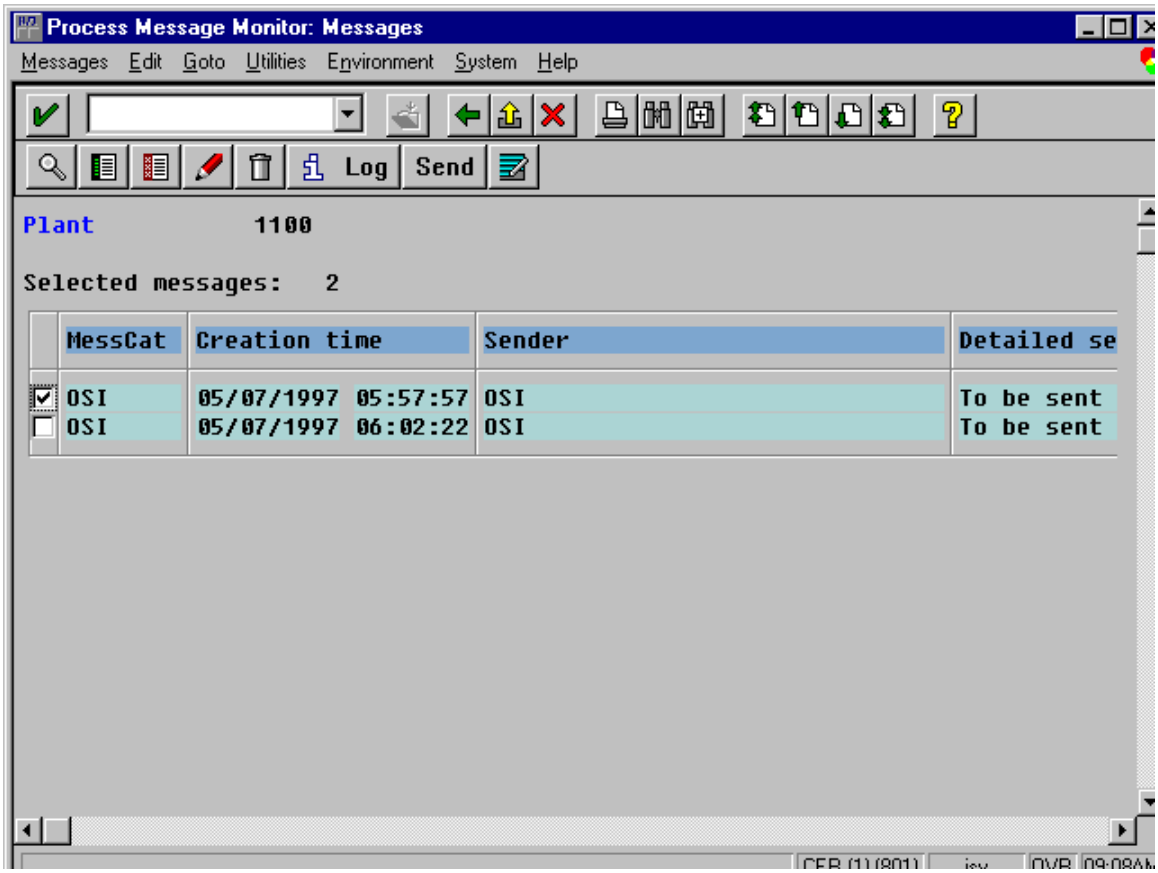
The details of that message are entered on the following screens.

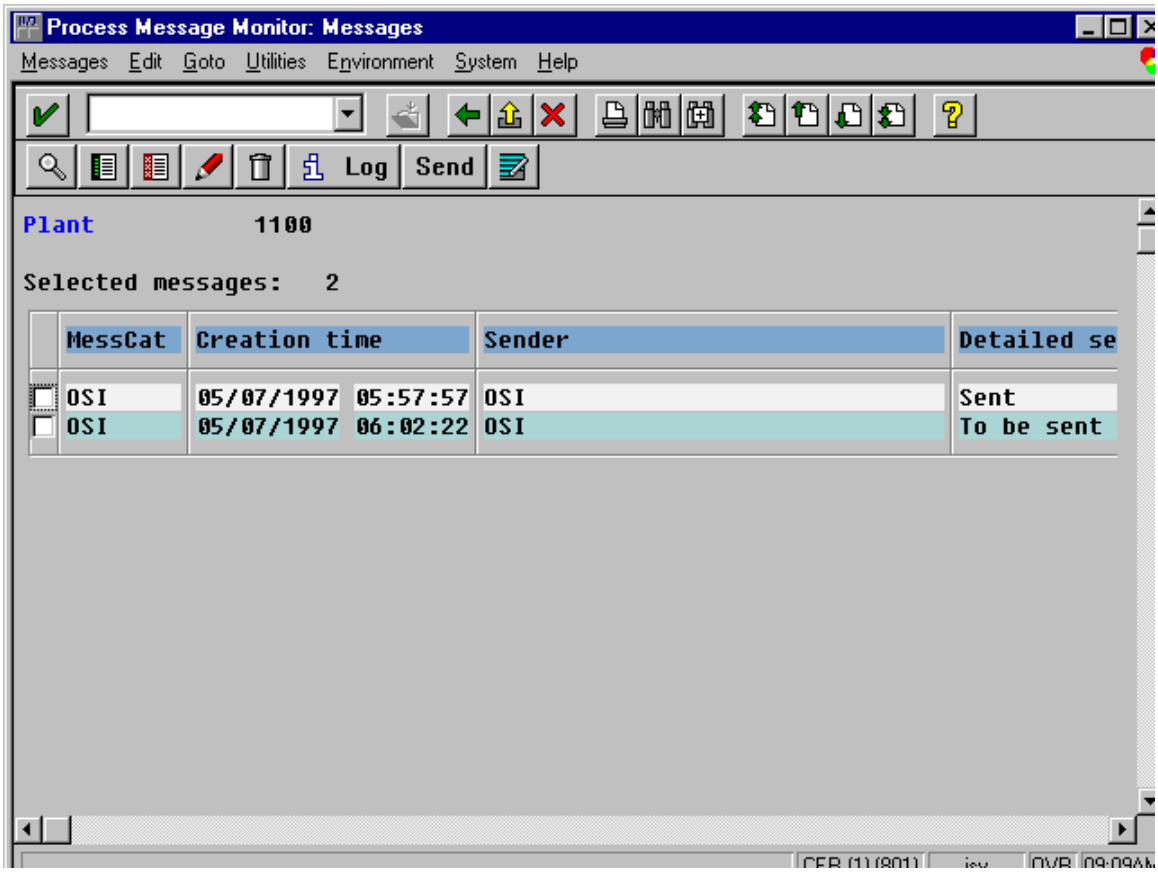




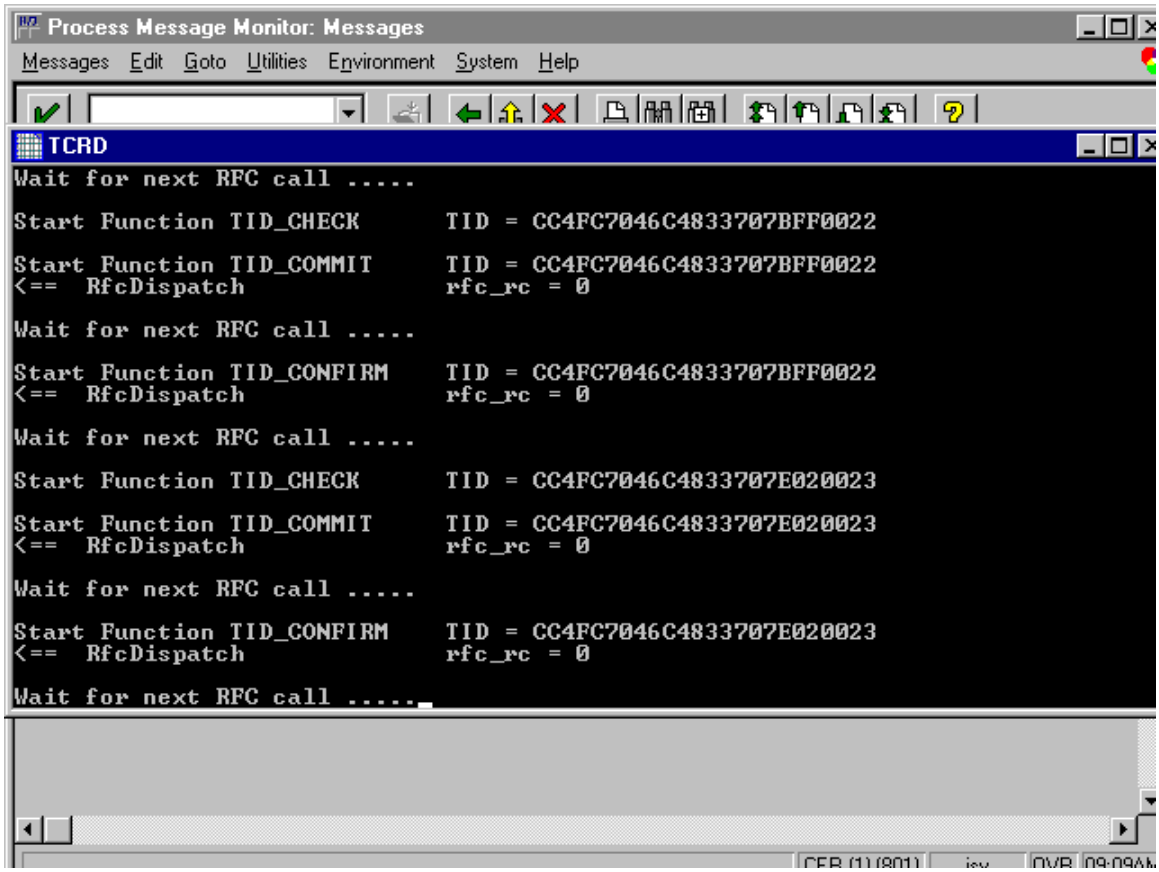
Now we prepare to send the message to RLINK from the Message Monitor /nCO54.




Process Message Monitor: Messages
Messages Edit Goto Utilities Environment System Help
Log
Send
Plant 1100
Selected messages: 2
MessCat
Creation time
Sender
Detailed se
OSI 05/07/1997 05:57:57 OSI To be sent
OSI 05/07/1997 06:02:22 OSI To be sent

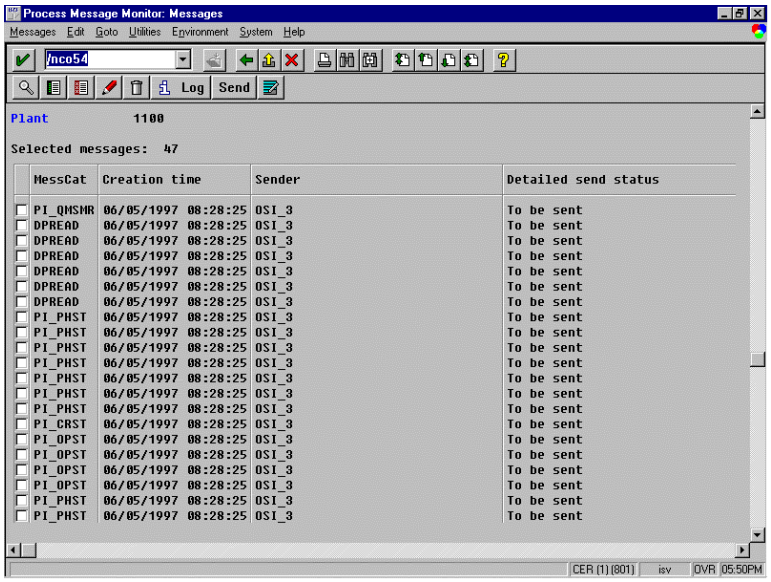


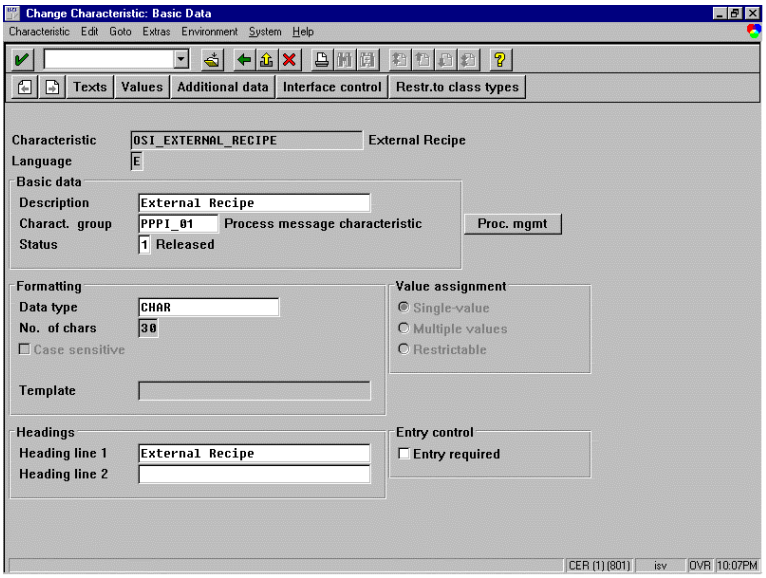
Switching back to RLINK we see in the following screen that the message has been received.

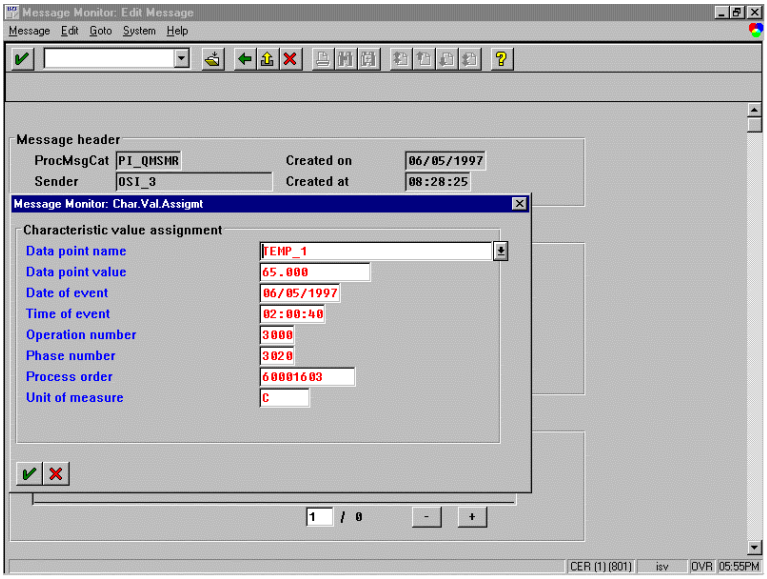


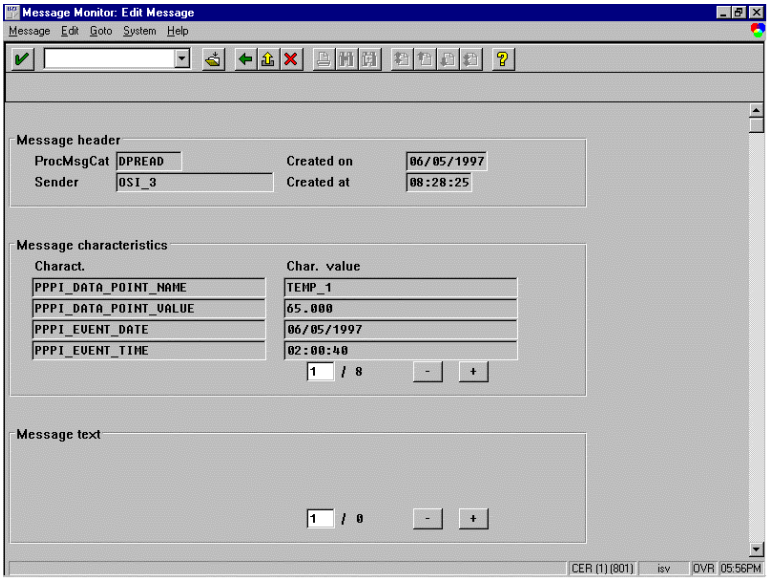
Now both the Recipe and the Message have been stored in the SQLServer and the process continue to execute the recipe and retrieve the information to satisfy the SAP/R3 requests. The data can be reviewed via the tools in the Graphics Interface Chapter.

The recipe would then be processed either through a batch execution system or PI-Batch or as a continuous plant. When the answers to the SAP/R3 recipe have been found the messages would be sent back up to SAP/R3. At this point they can be reviewed in the SAP/R3 message monitor /nco54. Corrections can be made and the message can be sent to the other SAP/R3 processes. These steps are shown in the following screens.




Change Characteristic: Basic Data
Characteristic Edit Goto Extras Environment System Help
Texts
Values
Additional data
Interface control
Restr.to class types
Characteristic
OSI_EXTERNAL_RECIPE
External Recipe
Language
E
Basic data
Description
External Recipe
Charact. group
PPPI_01
Process message characteristic
Proc. mgmt
Status
1 Released
Formatting
Data type
CHAR
No. of chars
30
Case sensitive
Template
Value assignment
Single-value
Multiple values
Restrictable
Headings
Heading line 1
External Recipe
Heading line 2
Entry control
Entry required
CER (1) (801)
isv
OVR
10:07PM


Message Monitor: Edit Message
Message Edit Goto System Help
Message header
ProcMsgCat
PI_QMSMR
Created on
06/05/1997
Sender
OSI_3
Created at
08:28:25
Message Monitor: Char.Val.Assigmt
Characteristic value assignment
Data point name
TEMP_1
Data point value
65.000
Date of event
06/05/1997
Time of event
02:00:40
Operation number
3000
Phase number
3020
Process order
60001603
Unit of measure
C
1 / 0
CER (1) (801)
isv
OVR
05:55PM


Message Monitor: Edit Message
Message Edit Goto System Help
Message header
ProcMsgCat
DPREAD
Created on
06/05/1997
Sender
OSI_3
Created at
08:28:25
Message characteristics
Charact.
Char. value
PPPI_DATA_POINT_NAME
TEMP_1
PPPI_DATA_POINT_VALUE
65.000
PPPI_EVENT_DATE
06/05/1997
PPPI_EVENT_TIME
02:00:40
1 / 8
Message text
1 / 0
CER (1) (801)
isv
OVR
05:56PM

Chapter 15
SAP/R3 Transactions

Using the RLINK product to do other SAP Transactions

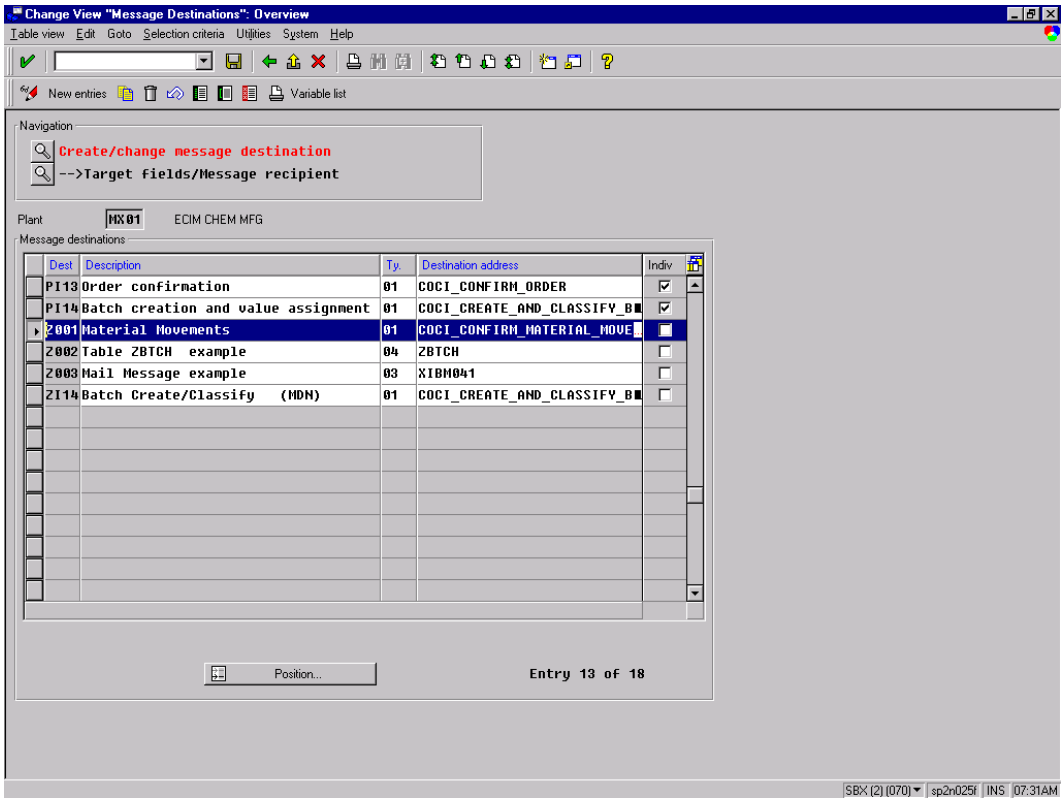
The RLINK product has can set up to do other SAP Transactions that are configured in SAP customization to be initiated via a set of messages through the message monitor. The following is a description of how the SAP system is configured as well as the configuration of the RLINK product.

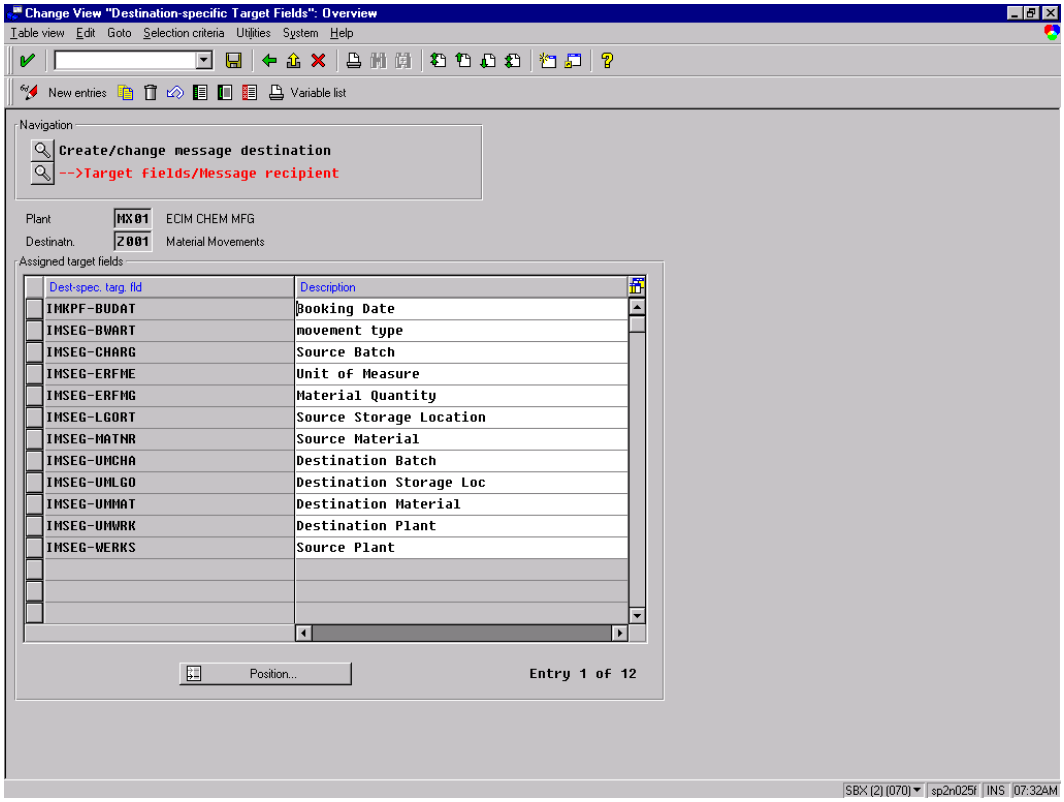
These transactions are ones that are continuously monitored for updates such as movements.

The functionality described here can be used to send on demand messages of any type. Configure a point group for the messge type and set a digital tag when you want to create the message. This has be used to construct messages for unplanned material in recipes.

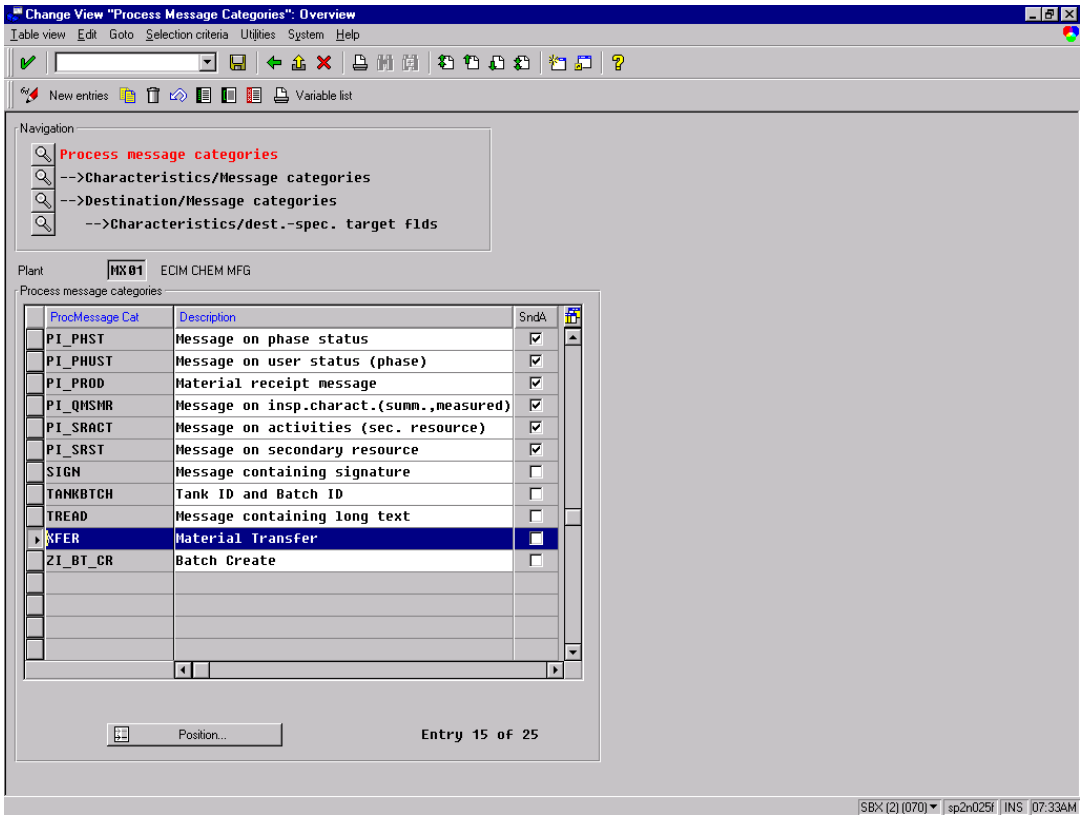
SAP Set-up of Message to do Material Movements

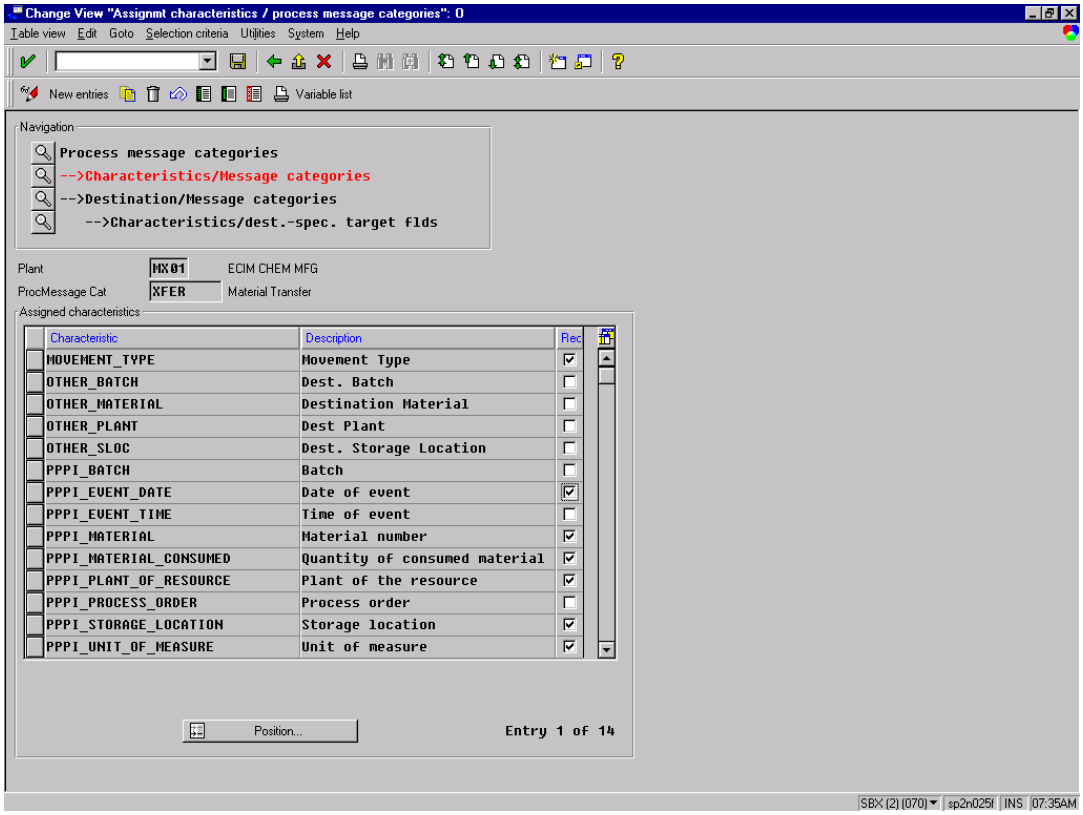
This is the set-up of the Message Destination:

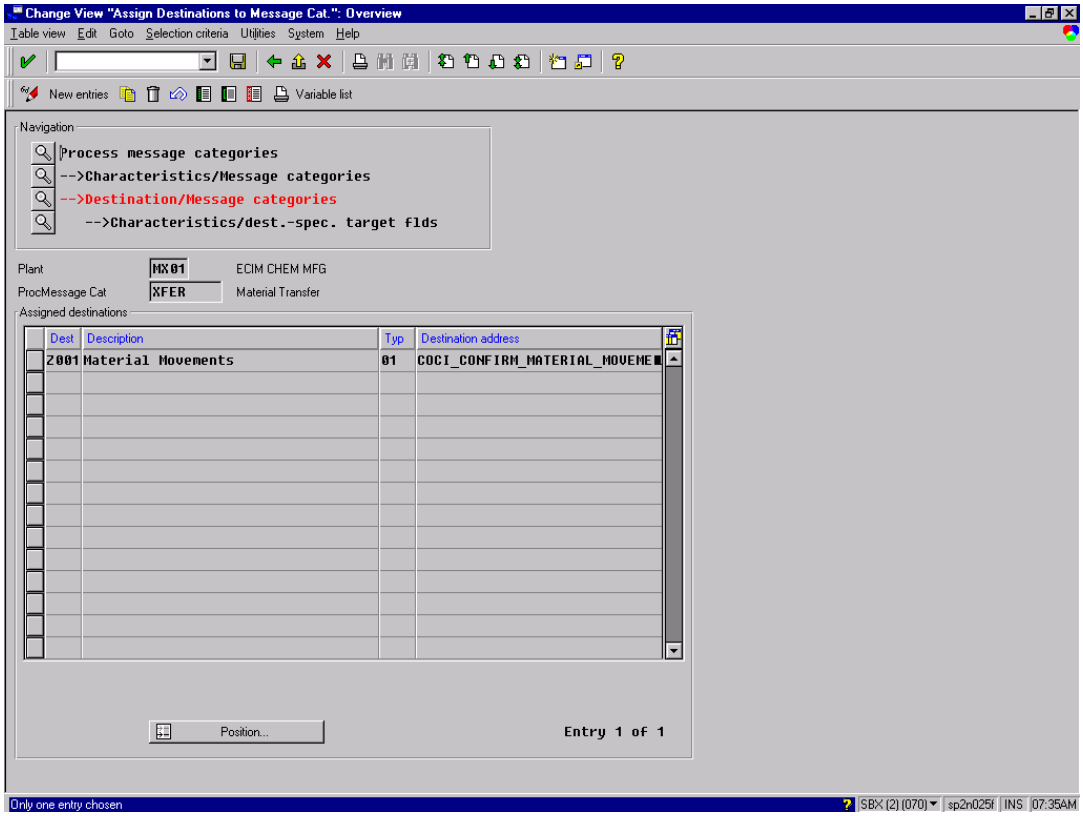

Change View "Message Destinations": Overview
Table view Edit Goto Selection criteria Utilities System Help
New entries
Variable list
Navigation
Create/change message destination
-->Target fields/Message recipient
Plant MX01 ECIM CHEM MFG
Message destinations
Dest Description Ty. Destination address Indiv
PI13 Order confirmation 01 COCI_CONFIRM_ORDER
PI14 Batch creation and value assignment 01 COCI_CREATE_AND_CLASSIFY_B
Z001 Material Movements 01 COCI_CONFIRM_MATERIAL_MOVE
Z002 Table ZBTCH example 04 ZBTCH
Z003 Mail Message example 03 XIBM041
ZI14 Batch Create/Classify (MDM) 01 COCI_CREATE_AND_CLASSIFY_B
Position...
Entry 13 of 18
SBX (2) (070) sp2n025f INS 07:31AM

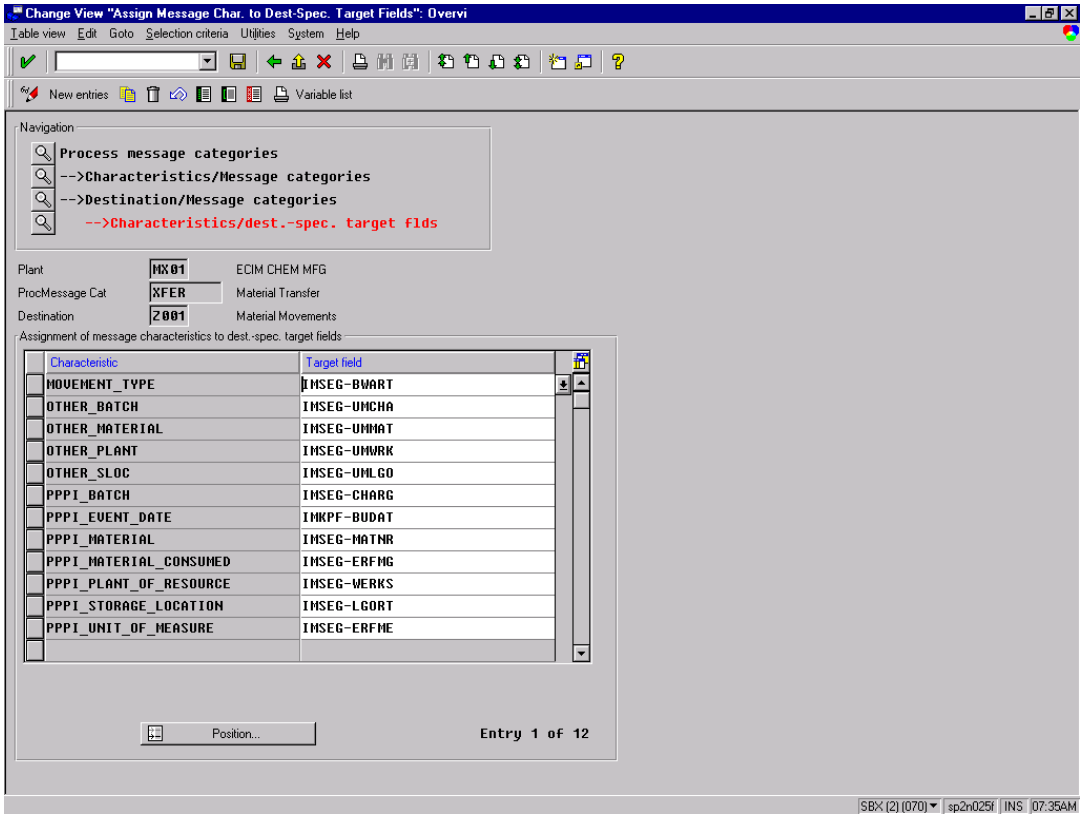

Change View "Destination-specific Target Fields": Overview
Table view Edit Goto Selection criteria Utilities System Help
New entries
Variable list
Navigation
Create/change message destination
-->Target fields/Message recipient
Plant MX01 ECIM CHEM MFG
Destinatn. Z001 Material Movements
Assigned target fields
Dest-spec. targ. fld Description
IMKPF-BUDAT Booking Date
IMSEG-BWART movement type
IMSEG-CHARG Source Batch
IMSEG-ERFME Unit of Measure
IMSEG-ERFMG Material Quantity
IMSEG-LGORT Source Storage Location
IMSEG-MATNR Source Material
IMSEG-UMCHA Destination Batch
IMSEG-UMLGO Destination Storage Loc
IMSEG-UMMAT Destination Material
IMSEG-UMWRK Destination Plant
IMSEG-WERKS Source Plant
Position...
Entry 1 of 12
SBX (2) (070) sp2n025f INS 07:32AM

This is the set-up of the Message Category:

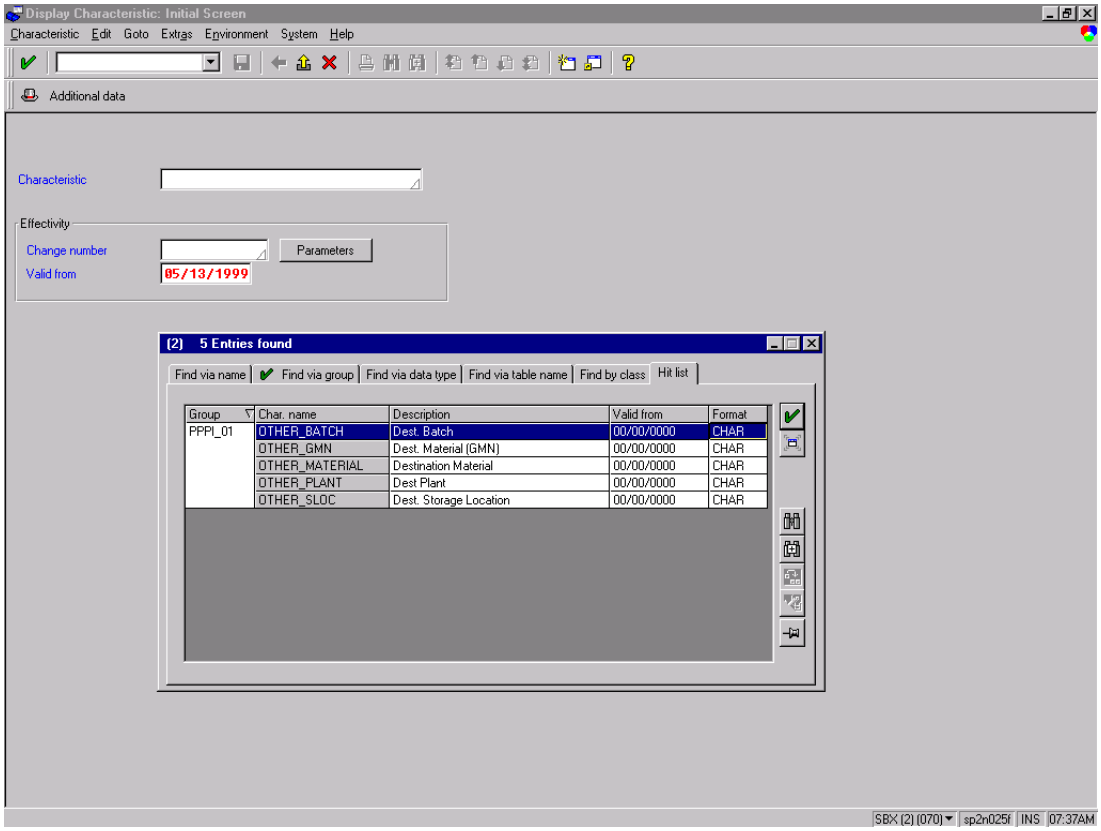




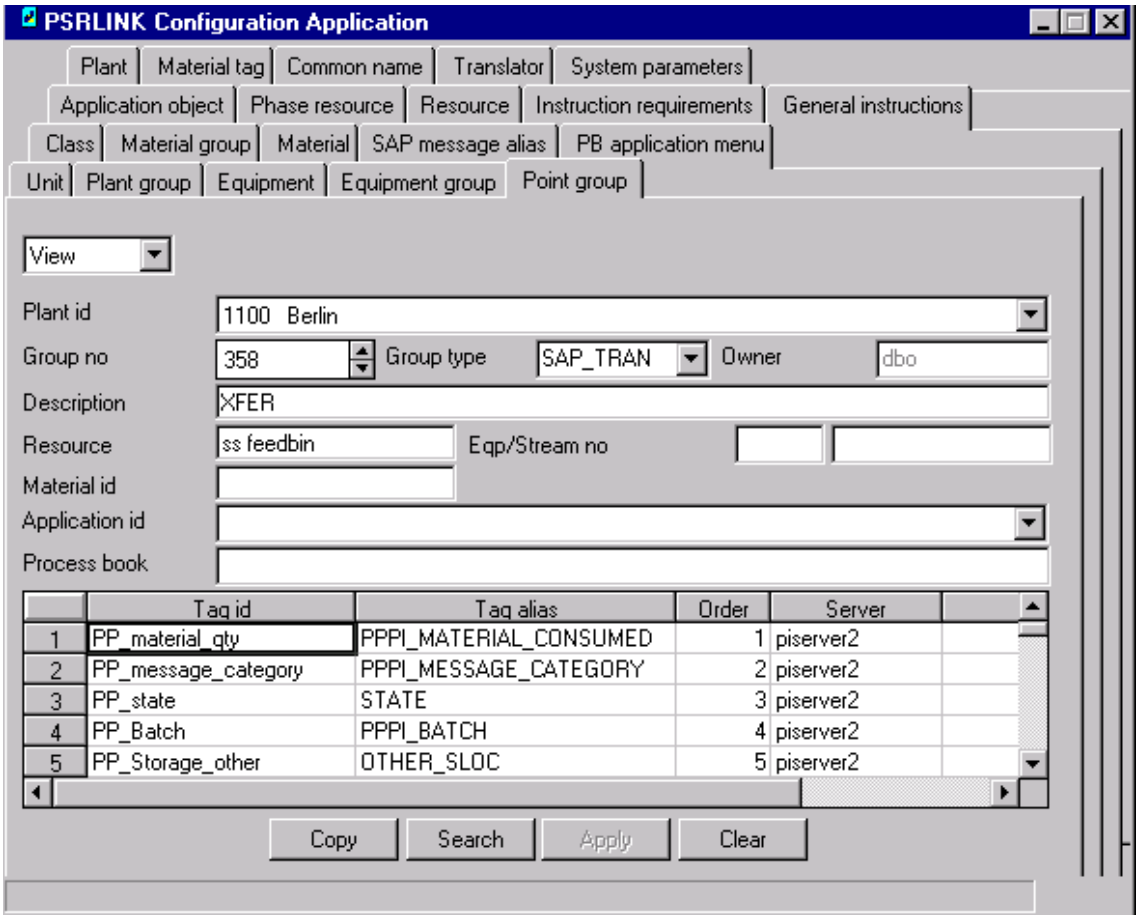


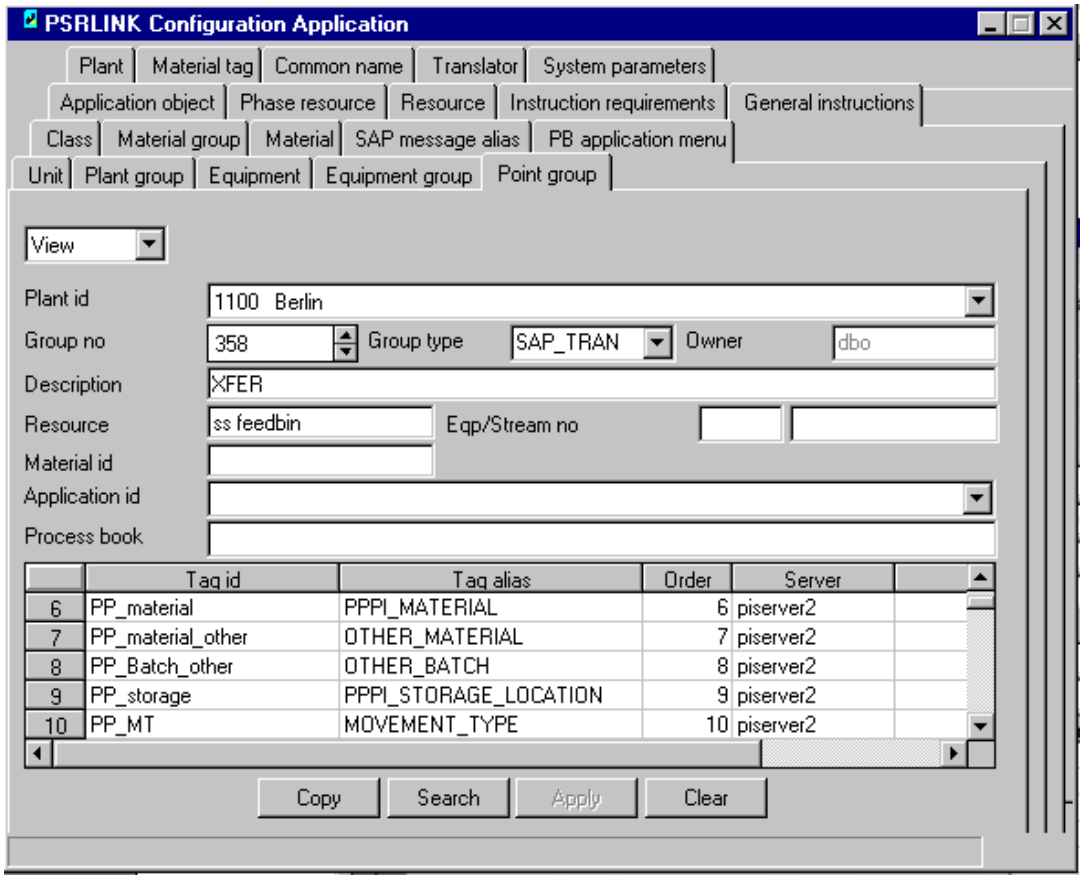


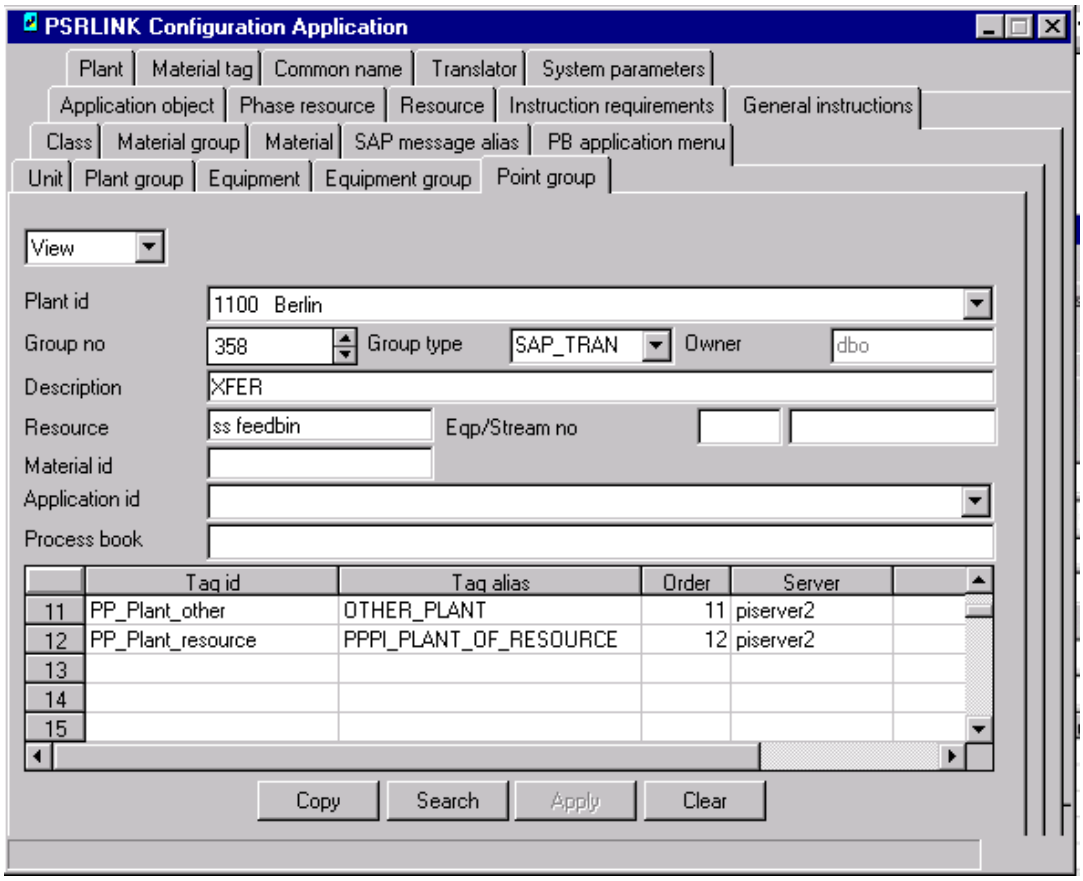
These are the characteristics that will have to be set up. The characteristic formats are just like the stanadard ones (PPPI)– you just need an extra set.



Point Group and Point Group Member Table Configuration


PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Application object
Phase resource
Resource
Instruction requirements
General instructions
Class
Material group
Material
SAP message alias
PB application menu
Unit
Plant group
Equipment
Equipment group
Point group
View
Plant id
1100 Berlin
Group no
358
Group type
SAP_TRAN
Owner
dbo
Description
XFER
Resource
ss feedbin
Eqp/Stream no
Material id
Application id
Process book
Tag id
Tag alias
Order
Server
1 PP_material_qty PPPI_MATERIAL_CONSUMED 1 piserver2
2 PP_message_category PPPI_MESSAGE_CATEGORY 2 piserver2
3 PP_state STATE 3 piserver2
4 PP_Batch PPPI_BATCH 4 piserver2
5 PP_Storage_other OTHER_SLOC 5 piserver2
Copy
Search
Apply
Clear


PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Application object
Phase resource
Resource
Instruction requirements
General instructions
Class
Material group
Material
SAP message alias
PB application menu
Unit
Plant group
Equipment
Equipment group
Point group
View
Plant id
1100 Berlin
Group no
358
Group type
SAP_TRAN
Owner
dbo
Description
XFER
Resource
ss feedbin
Eqp/Stream no
Material id
Application id
Process book
Tag id
Tag alias
Order
Server
6 PP_material PPPI_MATERIAL 6 piserver2
7 PP_material_other OTHER_MATERIAL 7 piserver2
8 PP_Batch_other OTHER_BATCH 8 piserver2
9 PP_storage PPPI_STORAGE_LOCATION 9 piserver2
10 PP_MT MOVEMENT_TYPE 10 piserver2
Copy
Search
Apply
Clear


PSRLINK Configuration Application
Plant
Material tag
Common name
Translator
System parameters
Application object
Phase resource
Resource
Instruction requirements
General instructions
Class
Material group
Material
SAP message alias
PB application menu
Unit
Plant group
Equipment
Equipment group
Point group
View
Plant id
1100 Berlin
Group no
358
Group type
SAP_TRAN
Owner
dbo
Description
XFER
Resource
ss feedbin
Eqp/Stream no
Material id
Application id
Process book
Tag id
Tag alias
Order
Server
11 PP_Plant_other OTHER_PLANT 11 piserver2
12 PP_Plant_resource PPPI_PLANT_OF_RESOURCE 12 piserver2
13
14
15
Copy
Search
Apply
Clear

Conventions used to set up this application are that the GROUP TYPE will be set to SAP_TRAN. There will be one point alias called STATE that will be monitored to see if a new entry has been added. This point will be a digital state.

The characteristic PPPI_UNIT_OF_MEASURE will be assigned to the entry with order = 1 and alias not STATE. The characteristics PPPI_EVENT_TIME and PPPI_EVENT_DATE will be assigned to the timestamp of the STATE point. The Tag_alias is set to be the characteristic name that is to be returned to SAP. If you want to store the engineering unit in a point and read this value to satisfy the PPPI_UNIT_OF_MEASURE then include an alias in the point group with the name PPPPI_UNIT_OF_MEASURE. In this case it will use theis value instead of the engineering unit from the tag with order = 1. If you do not want any unit of measure be sure no member has a display order of 1.

Case 1 STATE= 1 Date, Time no Eng Unit

Case 2 = 1 & not STATE if PPPI_UNIT_OF_MEASURE exists take tag otherwise tag that has display order of 1

Case 3 NEQ 1 then no Date, Time or Unit of Measure

Three new tables were added to PSRLINK.

The Characteristic Table that is used to specify the format of the characteristic to be returned to SAP

name	format
MOVEMENT_TYPE	CHAR
OTHER_BATCH	CHAR
OTHER_MATERIAL	CHAR
OTHER_PLANT	CHAR
OTHER_SLOC	CHAR
PPPI_BATCH	CHAR
PPPI_EVENT_DATE	DATE
PPPI_EVENT_TIME	TIME
PPPI_MATERIAL	CHAR
PPPI_MATERIAL_CONSUMED	NUM
PPPI_PLANT_OF_RESOURCE	CHAR
PPPI_STORAGE_LOCATION	CHAR
PPPI_UNIT_OF_MEASURE	CHAR

Ar_sap_tran and arv_sap_tran used for internal processing.

A new system parameter was created called DEST to be the destination returned in MSHD, the SM59 Destination. Currently this parameter has not been updated in the configuration dialog and must be entered manually with Microsoft Access to the system_parameter table.

System_parameter_code	system_param_description	system_value	system_value_text
DEST	SAP destination address		OSI_2

The PI program arsptran has been added to monitor any values in the state point and retrieve all values for the other members of the point group. This program gets the timestamp for the STATE tag and then requests a piar_getarcvaluex with the mode specified for all the other items in the point group. You can specifiy the mode in the exec_batch entry by using the format arsptran –P1 for mode 1. You can also put on the trace option for debugging by entering arsptran –T –P1.

The is an additional parameter –CY that when it is used it will not send back values when the digital state is set to 0000. If no C parameter is give it will use any value for the state tag to collect the other values. Some examples are

Arsptran.exe –T –CY

Arsptran.exe –P1 –CY

Arsptran.exe –T –P1 –CY

Arsptran.exe -CY

New entries must be made in the group_master and exec_batch tables for the three programs that must run. The first is a stored procedure that will insert new entries from point group into the table ar_sap_tran. This program is usr_ar_sap_tran_i. The second entry is for the PI program arsptran used to retrieve data from PI. The third entry is usr_msg_hdr_24 that translates the results to MSHD and MSEL for SAP.

Group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
48	PP sap transaction	1	6/2/1999 8:30:55 PM	1	0

program_name	batch_order	functionality	exe_or_sp	input_param2	group_
d:\psrlink\server\fe\arsptran.exe –P1	2	PP values for group	E		
usr_ar_sap_tran_i	1	PP insert ar_sap_tran	P		
usr_msg_hdr_24	3	PP insert MSHD,MSEL	P		

The new procedures that were added are as follows:

Procedure	
Usr_ar_sap_tran_i	Looks for new point groups which have been created of type SAP_TRAN and adds to the

	table ar_sap_tran
Usr_ar_sap_tran_sel	Selects an entry from the table ar_sap_tran
Usr_ar_sap_tran_upd	Updates and entry in the table ar_sap_tran and arv_sap_tran
Usr_msg_hdr_24	Selects results from ar_sap_tran and arv_sap_tran and constructs the results in the table mshd and msel

The program sequencing is as follows:

1. A new point group is detected and placed on ar_sap_tran. The entry for the STATE is given the timestamp associated with the point group detection.
2. When a value is found for the STATE point the timestamp is set for all the other members of the group an entry is made in arv_sap_tran with status of C and the status of ar_sap_tran is set to P.
3. The PI program will then find the values for the other members by taking the last value for the points. The status is changed to P and an entry of the results is made in arv_sap_tran with the status of C. The timestamp in field3 and trigger_time is set to the time the value is found at.
4. When a complete set of results exists with the same timestamp usr_msg_hdr24 translates the results to the tables MSHD and MSEL and changes the status to W in arv_sap_tran.
5. The PI program will continue to look for values adding 1 second to the last time found until the point group has been removed from the point_group table.
6. The results will be purged according to the duration setup in the purge table. The following entry must be made in the table purge.

Table_name	purge_method	last_timestamp	retention_days
arv_sap_tran	usr_arv_purge	6/7/1999 11:00:01 AM	10

A sample tag file for this transaction is as follows. Note that shutdown and compressing must be off for the points.

```
*create PP movement tags which are reals
@table pipoint
@ptclass classic
@mode create,t
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,span,shutdown,compressing
PP_material_qty,PP material quantity,KG,L,Float32,10000,off,off,
@endsection
```

```
*create PP movement tags which are strings
@table pipoint
@ptclass classic
@mode create,t
@stype delimited
@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing
PP_MT,PP Movement type,,L,string,off,off,
PP_Batch_other,PP Other Batch,,L,string,off,off,
PP_Material_other,PP Material Other,,L,string,off,off,
PP_Plant_other,PP Plant other,,L,string,off,off,
PP_Storage_other,PP Storage Location other ,,L,string,off,off,
PP_Batach,PP Batch,,L,string,off,off,
PP_Material,PP Material,,L,string,off,off,
PP_Plant_resource,PP Plant of Resource ,,L,string,off,off,
PP_Storage,PP Storage Location,,L,string,off,off,
PP_message_category,PP message category,,L,string,off,off
@endsection

*create digital states for PP
@table pids
@mode create,t
@istructure set, state, ...
PP_state,00000,00001,00002
@endsection

*create PP states
@table pipoint
@mode create,t
@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing
PP_state,PP state,PP_state,digital,off,off,
@endsection
```

Sample BAPI for writing CO57 Message

The following is sample code provided by a customer to illustrate how a CO57 message can be created in SAP to send data to RLINK and the PI System.

```
*************************************************************************
```

```
* Program Name: /EMN/BLENDPROCESSMSG (Txn YSB9)
*
*  Devel Class: /EMN/MD01
*
*  Description: Send Vessel Characteristic Values to OSI-PI
*            This program will send (for each vessel selected) the
*            vessel's current characteristic values to OSI-PI via
*            the PI-PCS interface using process message OSI_COMM.
*            The SAP BAPI, BAPI_PROCESS_MESSAGE_CREATEMLT is used
*            to create the process messages.
*
*         PPPI_EVENT_DATE        06/23/1999
*         PPPI_EVENT_TIME        14:18:24
*         YOSI_MESSAGE_TYPE      TAGUPDATE
*         PPPI_SOURCE            SAPSBX070 /EMN/BLENDPROCESSMSG
*         PPPI_PLANT_OF_RESOURCE  MX01
*         PPPI_RESOURCE          SG24
*         PPPI_MATERIAL          P15904FZ
*         PPPI_MESSAGE_TEXT      MX01_IVS                0.769
*                        MX01_L_INDEX            86.14
*                        MX01_A_INDEX            -0.71
*                        MX01_B_INDEX            0.77
*                        MX01_IVS_INTR           0.83
*                        LOBM_QSCORE             93
*
*         PPPI_MESSAGE_TEXT:  Offset  Length    Value
*                       0      30      characteristic name
*                      30      30      characteristic value
*
* Parameters/Arguments: Plant
*                  Vessel Type
*                  Vessel Id
*
REPORT /EMN/BLENDPROCESSMSG LINE-SIZE 132
                     MESSAGE-ID /EMN/MD.
TABLES: /EMN/MD001. " Blend Batch (Vessel) Header
INCLUDE /EMN/MD01_CONSTANTS.
```

```
DATA: MSGHDR   TYPE BAPI_RCOMHAPI OCCURS 0 WITH HEADER LINE,
      MSGCHR   TYPE BAPI_RCOMEAPI OCCURS 0 WITH HEADER LINE,
      TXTLIN   TYPE BAPI_RCOMSTXT OCCURS 0 WITH HEADER LINE,
      MSGNEW   TYPE BAPI_RCOMH    OCCURS 0 WITH HEADER LINE,
      MSGHDRRC TYPE BAPI_RCOMHRTC OCCURS 0 WITH HEADER LINE,
      MSGCHRRC TYPE BAPI_RCOMERTC OCCURS 0 WITH HEADER LINE,
      RETURN   TYPE BAPIRET2      OCCURS 0 WITH HEADER LINE.
DATA: BLEND_BATCH     TYPE /EMN/MD001-BLEND_BATCH,
      BLEND_QTY       TYPE MCHB-CLABS,
      BLEND_UOM       TYPE MARA-MEINS,
      CHAR_VALUE      TYPE AUSP-ATWRT,
      CHAR_UNIT(6)    TYPE C,
      EVENT_DATE      TYPE SY-DATUM,
      EVENT_TIME      TYPE SY-UZEIT,
      MD001           TYPE /EMN/MD001 OCCURS 0 WITH HEADER LINE,
*     system_timezone type tzonref-tzone,
      TIMESTAMP       TYPE TZONREF-TSTAMPS,
      TIMEZONE        TYPE TZONREF-TZONE,
      TMP_MSGID       TYPE BAPI_RCOMHAPI-PROC_MESS_ID_TMP,
      VALUE_TAB       TYPE API_VALI OCCURS 0 WITH HEADER LINE.
SELECTION-SCREEN BEGIN OF BLOCK PARAMS WITH FRAME.
  PARAMETERS:     P_PLANT  TYPE T001W-WERKS
                  OBLIGATORY
                  MEMORY ID WRK.
  SELECT-OPTIONS: SO_VTYPE FOR /EMN/MD001-VESSEL_TYPE
                  MATCHCODE OBJECT /EMN/MD011,
              SO_BATCH FOR /EMN/MD001-BLEND_BATCH
                  MATCHCODE OBJECT /EMN/MD001.
SELECTION-SCREEN END OF BLOCK PARAMS.

INITIALIZATION.
AT SELECTION-SCREEN.
* Authority check (Function: YSB9, Activity: Execute).
* authority-check object 'Y-SILOBLND'
*                id 'WERKS'      field p_plant
*                id 'SBFUNCTION' field k_func_ysb9_16.
* if sy-subrc ne 0.
```

```
*   message e067 with p_plant.
* endif.

* Read the blend vessel header record.
  SELECT * FROM /EMN/MD001 INTO TABLE MD001
          WHERE BLEND_PLANT  EQ P_PLANT
            AND BLEND_BATCH  IN SO_BATCH
            AND VESSEL_TYPE  IN SO_VTYPE
            AND SEND_TO_PIMS EQ K_TRUE.

  IF SY-SUBRC NE 0.
    MESSAGE E023 WITH P_PLANT SO_BATCH.
  ENDIF.

END-OF-SELECTION.

  TMP_MSGID = 0.

  CLEAR   MSGHDR.
  REFRESH MSGHDR.
  CLEAR   MSGCHR.
  REFRESH MSGCHR.
  CLEAR   TXTLIN.
  REFRESH TXTLIN.

* Get current timestamp (UTC).
  GET TIME STAMP FIELD TIMESTAMP.
* call function 'TZ_SYSTEM_GET_TZONE'
*      importing tzone_system = system_timezone.

* For each blend vessel record selected...
  LOOP AT MD001.

    CLEAR:  BLEND_QTY,
            BLEND_UOM,
            VALUE_TAB.
    REFRESH VALUE_TAB.
```

```
IF MD001-BLEND_TYPE EQ K_BT_BINMASTER.
  BLEND_BATCH = MD001-CURRENT_BATCH.
ELSE.
  BLEND_BATCH = MD001-BLEND_BATCH.
ENDIF.

*   Get batch stocks for blend batch.
CALL FUNCTION '/EMN/MD01_GET_BATCH_STOCKS'
    EXPORTING  MATL_NO        = MD001-BLEND_MATERIAL
               PLANT_NO       = MD001-BLEND_PLANT
               STOR_LOC       = MD001-BLEND_STLOC
               BATCH_NO       = BLEND_BATCH
    IMPORTING  TOTAL_STOCK     = BLEND_QTY
               UOM             = BLEND_UOM
    EXCEPTIONS BATCH_NOT_FOUND = 1
               OTHERS          = 2.

*   Get the current characteristic values for the blend vessel.
CALL FUNCTION 'QC01_BATCH_VALUES_READ'
    EXPORTING
        I_VAL_MATNR    = MD001-BLEND_MATERIAL
        I_VAL_CHARGE   = BLEND_BATCH
    TABLES
        T_VAL_TAB      = VALUE_TAB
    EXCEPTIONS
        NO_CLASS       = 1
        INTERNAL_ERROR = 2
        NO_VALUES      = 3
        NO_CHARS       = 4
        OTHERS         = 5.
CHECK SY-SUBRC EQ 0.

ADD 1 TO TMP_MSGID.

*   Build message header record.
MSGHDR-PROC_MESS_ID_TMP   =  TMP_MSGID.
```

```
    MSGHDR-PLANT            =  MD001-BLEND_PLANT.
    MSGHDR-PROC_MESS_CATEGORY = 'OSI_COMM'.
    MSGHDR-TEST_FLAG         =  K_FALSE.
    MSGHDR-SENDER_NAME       =  SY-UNAME.
    APPEND MSGHDR.

*   Build message characteristic records.
    CALL FUNCTION 'SD_TZONE_PLANT'
       EXPORTING  PLANT            = MD001-BLEND_PLANT
       IMPORTING  TIMEZONE         = TIMEZONE
       EXCEPTIONS MISSING_PLANT      = 1
                  NON_EXISTENT_PLANT = 2
                  OTHERS             = 3.
    IF SY-SUBRC EQ 0.
      CONVERT TIME STAMP TIMESTAMP TIME ZONE TIMEZONE
            INTO DATE EVENT_DATE TIME EVENT_TIME.
    ELSE.
      EVENT_DATE = SY-DATUM.
      EVENT_TIME = SY-UZEIT.
*     timezone   = system_timezone.
    ENDIF.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'PPPI_EVENT_DATE'.
    MSGCHR-CHAR_VALUE       =  EVENT_DATE.
    MSGCHR-DATA_TYPE        = 'DATE'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'PPPI_EVENT_TIME'.
    MSGCHR-CHAR_VALUE       =  EVENT_TIME.
    MSGCHR-DATA_TYPE        = 'TIME'.
    APPEND MSGCHR.

*   msgchr-proc_mess_id_tmp =  tmp_msgid.
*   msgchr-name_char        = 'YOSI_TIME_ZONE'.
*   msgchr-char_value       =  timezone.
```

```
*   msgchr-data_type        = 'CHAR'.
*   append msgchr.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'YOSI_MESSAGE_TYPE'.
    MSGCHR-CHAR_VALUE       = 'TAGUPDATE'.
    MSGCHR-DATA_TYPE        = 'CHAR'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP  =  TMP_MSGID.
    MSGCHR-NAME_CHAR         = 'PPPI_SOURCE'.
    MSGCHR-CHAR_VALUE(3)     = 'SAP'.
    MSGCHR-CHAR_VALUE+3(3)   =  SY-SYSID(3).
    MSGCHR-CHAR_VALUE+6(3)   =  SY-MANDT.
    MSGCHR-CHAR_VALUE+10(20) =  SY-REPID.
    MSGCHR-DATA_TYPE         = 'CHAR'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'PPPI_PLANT_OF_RESOURCE'.
    MSGCHR-CHAR_VALUE       =  MD001-BLEND_PLANT.
    MSGCHR-DATA_TYPE        = 'CHAR'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'PPPI_RESOURCE'.
    MSGCHR-CHAR_VALUE       =  MD001-BLEND_BATCH.
    MSGCHR-DATA_TYPE        = 'CHAR'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
    MSGCHR-NAME_CHAR        = 'PPPI_MATERIAL'.
    MSGCHR-CHAR_VALUE       =  MD001-BLEND_MATERIAL.
    MSGCHR-DATA_TYPE        = 'CHAR'.
    APPEND MSGCHR.

    MSGCHR-PROC_MESS_ID_TMP =  TMP_MSGID.
```

```
    MSGCHR-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    MSGCHR-DATA_TYPE        = 'CHAR'.
    APPEND MSGCHR.

*   Build message characteristic text lines.
    LOOP AT VALUE_TAB.
      CHECK ( VALUE_TAB-ATNAM(4) EQ 'EMNQ' OR
              VALUE_TAB-ATNAM    EQ 'LOBM_QSCORE' ).
      TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
      TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
      TXTLIN-TDFORMAT         =  K_NEW_LINE.
      TXTLIN-TDLINE(30)       =  VALUE_TAB-ATNAM.
*     TXTLIN-TDLINE+30(30)    =  VALUE_TAB-ATWRT.
      SPLIT VALUE_TAB-ATWRT AT SPACE INTO CHAR_VALUE CHAR_UNIT.
      TXTLIN-TDLINE+30(30)    =  CHAR_VALUE.
      APPEND TXTLIN.
    ENDLOOP.

    TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
    TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    TXTLIN-TDFORMAT         =  K_NEW_LINE.
    TXTLIN-TDLINE(30)       = 'BATCH_QUANTITY'.
    TXTLIN-TDLINE+30(30)    =  BLEND_QTY.
    CONDENSE TXTLIN-TDLINE+30(30).
    APPEND TXTLIN.

    TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
    TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    TXTLIN-TDFORMAT         =  K_NEW_LINE.
    TXTLIN-TDLINE(30)       = 'QUANTITY_UNIT'.
    TXTLIN-TDLINE+30(30)    =  BLEND_UOM.
    APPEND TXTLIN.

    TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
    TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    TXTLIN-TDFORMAT         =  K_NEW_LINE.
    TXTLIN-TDLINE(30)       = 'STORAGE_LOCATION'.
```

```
    TXTLIN-TDLINE+30(30)    =  MD001-BLEND_STLOC.
    APPEND TXTLIN.

    TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
    TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    TXTLIN-TDFORMAT         =  K_NEW_LINE.
    TXTLIN-TDLINE(30)       = 'VESSEL_ON_HOLD'.
    IF MD001-VESSEL_ON_HOLD EQ K_TRUE.
      TXTLIN-TDLINE+30(30)  = 'YES'.
    ELSE.
      TXTLIN-TDLINE+30(30)  = 'NO'.
    ENDIF.
    APPEND TXTLIN.

    TXTLIN-PROC_MESS_ID_TMP =  TMP_MSGID.
    TXTLIN-NAME_CHAR        = 'PPPI_MESSAGE_TEXT'.
    TXTLIN-TDFORMAT         =  K_NEW_LINE.
    TXTLIN-TDLINE(30)       = 'VESSEL_RESERVED'.
    IF MD001-VESSEL_RESERVED EQ K_TRUE.
      TXTLIN-TDLINE+30(30)  = 'YES'.
    ELSE.
      TXTLIN-TDLINE+30(30)  = 'NO'.
    ENDIF.
    APPEND TXTLIN.

  ENDLOOP.

* Create process message(s).
  CALL FUNCTION 'BAPI_PROCESS_MESSAGE_CREATEMLT'
      TABLES
          PROCMESSHEADER       = MSGHDR
          PROCMESSCHARAC       = MSGCHR
          PROCMESSTEXTLINES    = TXTLIN
          PROCESSMESSAGENEW    = MSGNEW
          PROCMESSHEADERRETURN = MSGHDRRC
          PROCMESSCHARACRETURN = MSGCHRRC
          RETURN               = RETURN.
```

```
  COMMIT WORK.

* Log process message(s) created.
  LOOP AT MSGNEW.
    READ TABLE MD001 INDEX MSGNEW-PROC_MESS_ID_TMP.
    MESSAGE I050 WITH MSGNEW-PROC_MESS_ID
                      MD001-BLEND_PLANT
                      MD001-BLEND_BATCH.
  ENDLOOP.

* Log process message errors.
  LOOP AT MSGHDRRC WHERE RETURN_CODE NE '00'.
*   read table md001 index msghdrrc-proc_mess_id_tmp.
    MESSAGE ID MSGHDRRC-ID
          TYPE 'I'
        NUMBER MSGHDRRC-NUMBER
          WITH MSGHDRRC-MESSAGE_V1
               MSGHDRRC-MESSAGE_V2
               MSGHDRRC-MESSAGE_V3
               MSGHDRRC-MESSAGE_V4.
  ENDLOOP.

* Log process message characteristic errors.
  LOOP AT MSGCHRRC WHERE RETURN_CODE NE '00'.
*   read table md001 index msgchrrc-proc_mess_id_tmp.
    MESSAGE ID MSGCHRRC-ID
          TYPE 'I'
        NUMBER MSGCHRRC-NUMBER
          WITH MSGCHRRC-MESSAGE_V1
               MSGCHRRC-MESSAGE_V2
               MSGCHRRC-MESSAGE_V3
               MSGCHRRC-MESSAGE_V4.
  ENDLOOP.

* Signal failure/success of process message(s) creation.
  MESSAGE ID RETURN-ID
```

```
  TYPE RETURN-TYPE
NUMBER RETURN-NUMBER
  WITH RETURN-MESSAGE_V1
     RETURN-MESSAGE_V2
     RETURN-MESSAGE_V3
     RETURN-MESSAGE_V4.
```

Chapter 16
SQLServer Data Source

Ad-hoc messages from SQL database access

The purpose of this extension is to provide SQL Database access rather than PI for support of Ad-Hoc messages to SAP. This is and extension to the general SAP transactions provided in RLINK using a SQL Database as the source of information. The example SQL database used is one provided by PolyOne.

The conventions used in the following example are

Point_group_table

1. The group_description is configured to be the name of the table
2. Plant_id is the name of the plant
3. In the process_book field put the name of the field that holds the plant

Point_group_members table

1. Tag_alias is the name of the characteristic to go to SAP
2. The tag_id is the name of the field in the table where the value will be retrieved from
3. The server is the name of the database in the SQLServer that will hold the data
4. There is no entry with display_order of 1 because you have specific entries in the SQL database for PPPI_EVENT_TIME, PPPI_EVENT_DATE and PPPI_UNIT_OF_MEASURE

Characteristics table

Any characteristics that you use in your messages must be defined in the characteristic table with the required SAP format.

Configuration of the point groups

Goods receipt

group_num	group_description	group_type	resource_id	plant_id	owner	Process_book
660	ah_goods_receipt	SAP_SQL		1100	dbo	Plant_id

group_num	tag_id	tag_alias	display_order	server	application_no
660	batch_id	PPPI_BATCH	5	RLINK_SQL	
660	event_date	PPPI_EVENT_DATE	6	RLINK_SQL	
660	event_time	PPPI_EVENT_TIME	7	RLINK_SQL	
660	material_id	PPPI_MATERIAL	4	RLINK_SQL	
660	quantity_produced	PPPI_MATERIAL_PRODUCED	8	RLINK_SQL	
660	message_type	PPPI_MESSAGE_CATEGORY	11	RLINK_SQL	
660	operation_id	PPPI_OPERATION	2	RLINK_SQL	
660	phase_id	PPPI_PHASE	3	RLINK_SQL	
660	process_order	PPPI_PROCESS_ORDER	12	RLINK_SQL	
660	unit_of_measure	PPPI_UNIT_OF_MEASURE	9	RLINK_SQL	
660	Quantity_produced	STATE	10	RLINK_SQL	

Goods issue

group_num	group_description	group_type	resource_id	plant_id	owner	Process_book
659	ah_goods_issue	SAP_SQL		1100	dbo	Plant_id

group_num	tag_id	tag_alias	display_order	server	application_no
659	batch_id	PPPI_BATCH	5	RLINK_SQL	
659	event_date	PPPI_EVENT_DATE	6	RLINK_SQL	
659	event_time	PPPI_EVENT_TIME	7	RLINK_SQL	
659	material_id	PPPI_MATERIAL	4	RLINK_SQL	
659	quantity_consumed	PPPI_MATERIAL_CONSUMED	8	RLINK_SQL	
659	message_type	PPPI_MESSAGE_CATEGORY	11	RLINK_SQL	
659	operation_id	PPPI_OPERATION	2	RLINK_SQL	
659	phase_id	PPPI_PHASE	3	RLINK_SQL	
659	process_order	PPPI_PROCESS_ORDER	12	RLINK_SQL	
659	unit_of_measure	PPPI_UNIT_OF_MEASURE	9	RLINK_SQL	
659	Quantity_consumed	STATE	10	RLINK_SQL	

Applications

Customer specific routines

- Usr_customer_saptran_state

(sample code will be provided, called by usr_sql_arsaptran)

- Usr_customer_saptran_other

  (sample code will be provided, called by usr_sql_arsaptran)

RLINK-SQL application procedures

- Usr_sql_arsaptran

Processing of new group-type SAP_SQL

- Usr_arsap_sql_i - this procedure inserts new groups of group_type SAP_SQL in the table ar_sap_tran table it calls the customer specific routine for additional insertion field specifications
- Usr_customer_sap_tran_i – this procedure adds additional information that is customer specific for insertion into ar_sap_tran. A version that matches the tables for PolyOne is delivered. (code will be provided)

Sample SQLDatabase

Msg_goods_issue added the following fields

- data_process_state

```
create table ah_goods_issue
(
msg_id                  int             primary key,
process_order           char(12),
recipe_id               char(18),
operation_id            char(4),
phase_id                char(4),
material_id             char(18),
batch_id                char(10),
event_date              char(8),
event_time              char(6),
plant_id                char(4),
quantity_consumed       float,
unit_of_measure         char(10),
last_value              char(1),
interface_status        char(1),
interface_timestamp     datetime,
message_type            char(8),
data_process_state      char(1)         null
)
```

A typical entry in this table is shown below. When an entryis made in this table the field data_process_state is null. The columns interface_status and interface_timestamp are not used by RLINK they are there for the customer use. Note that the date and time are entered into event_date and event_time as shown.

msg_id	process_order	operation_id	phase_id	material_id	batch_id	event_date	event_time	plant_id
1	0001	1000	1010	MAT01	BATCH01	20010925	141400	1100
2	0002	2000	2010	MAT02	BATCH02	20010926	120000	1100

quantity_consumed	unit_of_measure	interface_status	interface_timestamp	message_type	data_process_state
100	Lbs	N	25-Sep-01	PI_CONS	C
50	Lbs	N	26-Sep-01	PI_CONS	

- Msg_goods_receipt added the following fields
  - data_process_state

```
create table ah_goods_receipt
(
msg_id                  int             primary key,
process_order           char(12),
recipe_id               char(18),
operation_id            char(4),
phase_id                char(4),
material_id             char(18),
batch_id                char(10),
event_date              char(8),
event_time              char(6),
plant_id                char(4),
quantity_produced       float,
unit_of_measure         char(10),
interface_status        char(1),
interface_timestamp     datetime,
message_type            char(8),
delivery_complete       char(1),
```

```
last_value          char(1),
data_process_state  char(1)     null
)
```

Group_master and exec_batch

Group_master table

group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
53	Ad-Hoc SQL	1	9/26/2001 5:16:51 PM	0	0

Exec_batch table

The program usr_adhoc_helper N where N is the number of times to recall the sql procedures is used to increase the number of times that the SQL programs are called. The programs that this will call are usr_ar_sap_sql_I and usr_sql_arsptran.

program_name	batch_order	functionality	exe_or_sp	input_param1	group_no	batch_no
Usr_adhoc_helper 9	3	Multiple calls for sql	P		53	1

Installation

The installation follows the standard RLINK patch update methodology. A script is provided that updates the database and the programs I the other directories are moved to the corresponding RLINK directories.

Recipe Processing from SQLDatabase

Translation method

The translation method usr_sql_customer is used if data is to be configured to come from a SQL database. This procedure can use any combination of information in the databases to setup the query of the external system and then calls usr_insert_ar. It will pass request_part_id and it will get returned all the fields needed to update action_results.

Application

The application usr_application_sql will be used to return data from the SQL database. This procedure selects the first row from action_results for the application that has an open status. It then selects all other members from the same request_id. It then searches the configured table to see if there are any responses to this request. If there is a response it will complete the replies for all members in the request_id. It will continue for all rows that it finds that match in the SQL database. After it has completed this request_id it will move on to the next request_id. Internally it calls usr_customer_application passing all the fields in action_results and getting returned value and timestamp and then the program usr_application_sql calls usr_upd_rcp_ar if the characteristic is for the recipe status, usr_upd_phase_ar if it is for the phase status and usr_updae_action_list for all others with the value and timestamp. For instructions other than PI_CRST and PI_PHST there must be a lastvalue signal maintained in order to determine when the last value has been received. The lastvalue must be a blank or

"X" to indicate the last value. When the entry in the SQL database is read the procedure usr_customer_application will update the interface_status to 'Y' if successful and 'E' if it fails and update the interface timestamp with the current time.

Iniparameters

To the table translation_method must be added "usr_sql_customer"

name	description
usr_sql_customer	SQL recipe

To the table application must be added "usr_application_sql"

application_no	application_description	program_name
79	SQL application	usr_application_sql

Application "usr_application_sql" added to group_master and exec_batch. The procedure usr_application_sql calls usr_sql_customer which can be modified by the customer.

Group_master table

group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
53	Ad-Hoc SQL	1	9/26/2001 5:16:51 PM	0	0

Exec_batch table

program_name	batch_order	functionality	exe_or_sp	input_param1	group_no	batch_no
usr_application_sql	3	SQL request recipe	P		53	1

Translator Table

The Translator table must be changed to support the new application and translation method for the characteristics. In configuring the translator table configure the reply_method for who should assign the Engineering Unit and timestamp.

Translation Methods Fields

For recipe translation PI_CRST into action_results the following data is required

Recipe_no

Table

Database

Field_name that holds recipe status

Field_name that holds recipe_no

Field_name that holds recipe_timestamp

State field

For phase translation PI_PHST into action_results the following data is required

Recipe_no

Phase_id

Table

Database

Field that holds recipe_no

Field that holds phase_id

Field that holds phase_status

Field that holds phase_timestamp

State field

For translation of goods issue PI_CONS into action_results the following data is required

Recipe_no

Phase_id

Material_id

Table

Database

Field that holds recipe_no

Field that holds phase_id

Field that holds material_id

Field for the characteristic being returned

Field for timestamp

Characterisitic name

Field that holds lastvalue

Request_id

For translation of goods issue PI_PROD into action_results the following data is required

Recipe_no

Phase_id

Material_id

Table

Database

Field that holds recipe_no

Field that holds phase_id

Field that holds material_id

Field for the characteristic being returned

Field for timestamp

Field that holds lastvalue

Characteristic name

Request_id

Sample SQL database

Msg_control_recipe_status (recipe_timestamp is used for time)

```
create table msg_control_recipe_status
        (
        msg_id                  int             primary key,
        process_order           char(12),
        recipe_id               char(18),
        recipe_status           char(5),
        event_date              char(8),
        event_time              char(6),
        recipe_timestamp        datetime,
        interface_status        char(1),
        interface_timestamp     datetime,
        message_type            char(8),
        data_process_state      char(1)         null
        )
```

The recipe_status must hold the valid values for SAP that are

00005 Processed

00004 Terminated

00007 Discarded

Msg_phase_status (status_timestamp is used for time)

```
create table msg_phase_status
    (
    msg_id          int             primary key,
    recipe_id       char(18)        ,
```

```
operation_id          char(4)       ,
phase_id              char(4)       ,
phase_status          char(5)       ,
status_timestamp      datetime      ,
interface_status      char(1),
interface_timestamp   datetime,
data_process_state    char(1)       null
)
```

The phase_status must hold the valid values for SAP that are

00001 Started

00002 Finished

00003 Interupted

00004 Partial

Msg_goods_issue (event_timestamp is used for time)

```
create table msg_goods_issue
(
msg_id                int           primary key,
process_order         char(12)      ,
recipe_id             char(18)      ,
operation_id          char(4)       ,
phase_id              char(4)       ,
material_id           char(18)      ,
batch_id              char(10)      ,
event_date            char(8)       ,
event_time            char(6)       ,
event_timestamp       datetime      ,
plant_id              char(4)       ,
quantity_consumed     float         ,
unit_of_measure       char(10)      ,
last_value            char(1)       null,
message_type          char(8)       ,
interface_status      char(1),
interface_timestamp   datetime,
data_process_state    char(1)       null
)
```

Msg_goods_receipt (event_timestamp is used for time)

```
create table msg_goods_receipt
(
msg_id                  int             primary key,
process_order           char(12)        ,
recipe_id               char(18)        ,
plant_id                char(4)         ,
operation_id            char(4)         ,
phase_id                char(4)         ,
material_id             char(18)        ,
batch_id                char(10)        ,
event_date              char(8)         ,
event_time              char(6)         ,
event_timestamp         datetime        ,
quantity_produced       float           ,
unit_of_measure                 char(10)        ,
last_value              char(1)         null,
message_type            char(8)         ,
interface_status        char(1),
interface_timestamp     datetime,
delivery_complete       char(1),
data_process_state      char(1)         null
)
```

Msg_phact (status_timestamp is used for time)

```
create table msg_goods_receipt
(
msg_id                  int             primary key,
process_order           char(12)        ,
recipe_id               char(18)        ,
plant_id                char(4)         ,
operation_id            char(4)         ,
phase_id                char(4)         ,
activity                varchar(50)     ,
parameter_id            varchar(50)     ,
event_timestamp         datetime        ,
status_timestamp        datetime        ,
```

```
unit_of_measure        char(10)      ,
last_value             char(1)       null,
message_type           char(8)       ,
interface_status       char(1),
interface_timestamp    datetime,
data_process_state     char(1)       null
)
```

Configuration

Point_group and point_group_members

group_num	group_description	group_type	process_book	resource_id	plant_id
669	msg_phase_status	SQL_PHASE			1100
668	msg_control_recipe_status	SQL_RECIPE			1100
670	msg_goods_issue	SQL_CONS			1100
671	msg_goods_receipt	SQL_PROD			1100
672	msg_phact	SQL_PHACT			1100

group_num	tag_id	tag_alias	display_order	server
668	recipe_id	PPPI_CONTROL_RECIPE	1	RLINK_SQL
668	recipe_status	PPPI_CONTROL_RECIPE_STATUS	2	RLINK_SQL
668	recipe_timestamp	PPPI_EVENT_TIMESTAMP	3	RLINK_SQL
668	data_process_state	STATE	4	RLINK_SQL
669	operation_id	PPPI_OPERATION	3	RLINK_SQL
669	phase_id	PPPI_PHASE	4	RLINK_SQL
669	status_id	PPPI_PHASE_STATUS	5	RLINK_SQL
669	data_process_state	STATE	6	RLINK_SQL
669	Recipe_id	PPPI_CONTROL_RECIPE	7	RLINK_SQL

group_num	tag_id	tag_alias	display_order	server
670	batch_id	PPPI_BATCH	7	RLINK_SQL
670	recipe_id	PPPI_CONTROL_RECIPE	1	RLINK_SQL
670	event_timestamp	PPPI_EVENT_TIMESTAMP	2	RLINK_SQL
670	material_id	PPPI_MATERIAL	3	RLINK_SQL
670	quantity_consumed	PPPI_MATERIAL_CONSUMED	4	RLINK_SQL
670	phase_id	PPPI_PHASE	5	RLINK_SQL
670	unit_of_measure	PPPI_UNIT_OF_MEASURE	6	RLINK_SQL
670	Last_value	PPPI_LAST_VALUE	7	RLINK_SQL
670	data_process_state	STATE	8	RLINK_SQL
670	operation_id	PPPI_OPERATION	9	RLINK_SQL
670	plant_id	PPPI_PLANT	10	RLINK_SQL
670	Process_order	PPPI_PROCESS_ORDER	11	RLINK_SQL
671	batch_id	PPPI_BATCH	7	RLINK_SQL
671	recipe_id	PPPI_CONTROL_RECIPE	1	RLINK_SQL
671	event_timestamp	PPPI_EVENT_TIMESTAMP	2	RLINK_SQL
671	material_id	PPPI_MATERIAL	3	RLINK_SQL
671	quantity_produced	PPPI_MATERIAL_PRODUCED	4	RLINK_SQL
671	phase_id	PPPI_PHASE	5	RLINK_SQL
671	unit_of_measure	PPPI_UNIT_OF_MEASURE	6	RLINK_SQL
671	Last_value	PPPI_LAST_VALUE	7	RLINK_SQL

group_num	tag_id	tag_alias	display_order	server
671	data_process_state	STATE	8	RLINK_SQL
672	activity	PPPI_ACTIVITY	1	RLINK_SQL
672	data_process_state	STATE	6	RLINK_SQL
672	recipe_id	PPPI_CONTROL_RECIPE	2	RLINK_SQL
672	status_timestamp	PPPI_EVENT_TIMESTAMP	3	RLINK_SQL
672	Last_value	PPPI_LAST_VALUE	4	RLINK_SQL
672	Operation_id	PPPI_OPERATION	5	RLINK_SQL
672	Phase_id	PPPI_PHASE	7	RLINK_SQL
672	Plant_id	PPPI_PLANT	8	RLINK_SQL
672	Process_order	PPPI_PROCESS_ORDER	9	RLINK_SQL
672	Parameter_id	PPPI_STD_VALUE_PARAMETER_ID	10	RLINK_SQL
672	Unit_of_measure	PPPI_UNIT_OF_MEASURE	11	RLINK_SQL

Chapter 17
Repetitive Manufacturing

The business object Repetitive Manufacturing Backflush is a confirmation on the status of production in repetitive manufacturing. There is no reference to long-term orders. In a repetitive manufacturing backflush you can confirm the quantities produced and the activities required to produce these quantities. The components consumed are backflushed.

Procedures

usr_pp_rm_ar_rem_I	Inserts new point groups into pp_ar_rem
usr_pp_rm_ar_rem_upd	Updates results from PI into pp_ar_rem and pp_arv_rem
usr_ar_rem_sel	Selects values from pp_ar_rem
usr_pp_rm_bapi_general	Selects data for BAPI calls
usr_pp_rm_status_u	Updates return status and results from BAPI call
usr_pp_rm_arem_2_r3	Procedure reads from action_results and inserts into SAP like tables
usr_pp_rm_putvalue_2_pi	Confirmation number written to PI by sending to action_send

Application

Application arem.exe gets data from PI

Repman –1	MTS
Repman –2	Cancel
Repman –3	Check Existence
Repman –4	MTP
Repman –5	MTO

Tables

PP_arv_rem	results from PI for the requests
Pp_Ar_rem	data being requested from PI
Pp_rm_datagen_in	BFLUSHDATAGEN
pp_rm_datserial_in	SERIALNR
pp_rm_flag_in	BFLUSHFLAGS
pp_rm_dataorder	sales order and order item for MTO

pp_rem_existencecheck	table for existence check contains confirmation_no and status return
pp_rm_cancel	entries to be canceled in SAP
pp_rm_prod_ver	holds configured production version

RETURN messages are sent to the error_log.

Components

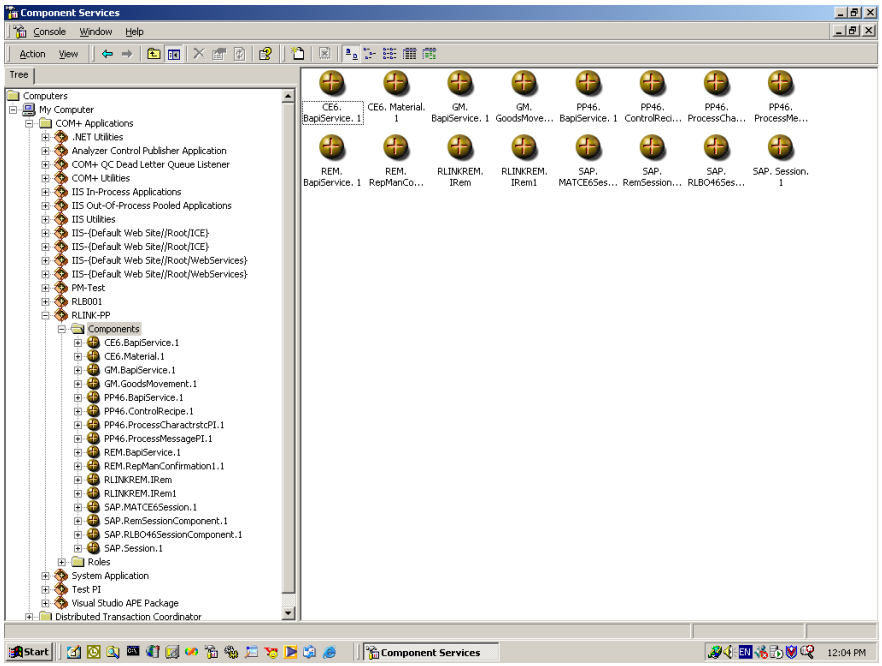
The components are:

REM.BapiService

Rem.RepManConfirmation1

RLINKRem.IRem

SAP.RemSessionComponent



Group_master and exec_batch

group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
54	REM-Processing	1	10/10/2001 11:53:46 AM	1	0

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
d:\rlink\pppi\server\fe\repman.exe –1	4	REM-Sends Values to SAP and receives confirmation	E	54	1
d:\rlink\pppi\server\fe\arem.exe	2	Get values to	E	54	1

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
		pp_arv_rem table			
usr_pp_rm_putvalue_2_pi	5	REM-puts value into pi	P	54	1
usr_pp_rm_ar_rem_I	1	Populates records into pp_ar_rem table	P	54	1
usr_pp_rm_arem_2_r3	3	Populates values to SAP tables	P	54	1
d:\rlink\pppi\server\fe\repman.exe –2	6	REM-Sends Values to SAP and receives confirmation	E	54	1

MTS- made to stock

BflushFlags

These flags determine the type and scope of the backflush to be posted.

BCKFLTYPE – Type of the backflush to be posted

MTS

01 Final Backflush

02 Reporting Point – REQUIRES REPPOINT

10 Separate Activity Backflush

11 Separate Component consumption backflush

12 Separate component scrap backflush

RP_SCRAPTYPE – Type of reporting point backflush it is only required if BCKFLTYPE is 02 and you are backflushing scrap

i.e. Value in SCRAPQUANT. The default value is 1

1 scrap at reporting point

2 Scrap up to the entered reporting point

3 Excess component consumption at reporting point

ACTIVITES_TYPE – defines the type of separate activity backflush it is only filled if a values of 10 is given in BCKFLTYPE

1 Activities of all operations

2 Only activities at reporting point

3 Only activities of operations after the last reporting point

COMPONENTS_TYPE- parameter defines the type of separate goods issue posting or a separate component scrap posting it is only filled if the values of 11 or 12 are assigned to BCKFLTYPE.

1 Components of all operations

2 Only components reporting point

3 Only components after last reporting point

SERIALNR – If serial numbers have to be maintained for the finished product to be backflushed, you can use the parameter to transfer the serial numbers to be used for posting the goods receipt of the finished product. Note in certain circumstances the number of serial numbers to be transferred must correspond exactly to the back flush quantity (this depends on the serial number settings in SAP). If in such a case too few serial numbers are transferred, the system cannot carry out the backflush successfully. You can use the serial number to identify individual instances of a material for individual tracking purposes. A prerequisite for performing serialization of a material is the entry of a serialization profile at material plant level.

CONFIRMATION_NO – returned from the MTS backflush and is the backflush number

MTS – made to stock

You must enter a material and a plant or a planned order

You must enter a posting date or a document date

If the repetitive manufacturing profile is set so that an online corrections mandatory when withdraw errors occur and if a withdrawal error occurs when backflushing the backflush cannot be carried out and an error code is returned. IF the profile is set so that correction is optional the system creates post-processing records for the components that could not be withdrawing. Therefore make sure that the creation of post processing records is allowed in the repetitive manufacturing profile. Otherwise the errors are not logged and there is no system-aided way of post posting of components.

BflushData

REPPOINT – milestone operation number in the routing. Must be filled if you are to carry out a reporting point backflush or if you want to carry out a separate activity posting, a separate goods issue posing or a separate component scarp posting with reference to a reporting point. You can only enter a reporting point after you have selected the indicator RP backflush.

BflushDataGen – This is independent of MTS, MTO or MTP

Parameter	Description	Size	Comments
PDC_NUMBER	PDC number	12	You must enter this number

	given by external system		when reversion a backflush, the system only reverses the posting corresponds to the number
MATERIALNR	Material number	18	
PRODPLANT	Plant	4	
PLANPLANT	Planning plant	4	Specifically identifies the plant where you want to post the goods receipt for an assembly. You only have to enter the planning plant if a material is produced in a different plant from where it is planned. In this case you do not have to enter anything in the filed plant (that is where the material is produced – production plant) It is determined automatically from a special procurement key in the material master record or from the planned order. If you backflush data, the system posts the goods receipt to the planning plant and backflushes the components from the production plant. In the assembly's material master record, for the planning plant you must have maintained a special procurement key for production in another plant. When you carry out the planning run in the planning plant, the system records the planning plant and the production plant in the planned orders.
STORAGELOC	Receiving storage location	4	The system determines the receiving storage location automatically if you entered it in the production version. If no it can be entered here.
PRODVERSION	Production version	4	Key that determines the various production techniques according to which a material can be manufactured. The production version determines the BOM alternative for a

			BOM explosion, the task list type the task list group and the task list group counter, lost size restrictions
PRODLINE	Production line	8	Describes the capacities, can be represented in the system either by a work center or by a line hierarchy. The production line is configure when you configure the production version.
PLANNINGID	Planning identification2	8	Enables you to group various material either by location or by time for planning and evaluation purposes. For example you can assign a planning ID to all materials that are manufactured on a certain production line. The planning ID is assigned to a production version. If you create a material with a certain production version the planning ID of this version is transferred automatically. If you want to use the planning ID instead of the production version in planning you must create a work center and enter this work center in the production version in the production line field.
BATCH	Receiving batch	10	Number of the batch according to which the material is posted in backflushing
POSTDATE	Posting date YYYYMMDD	8	Data that is used when entering the document in Financial accounting or controlling. When entering documents the system checks whether the posting date entered is allowed by means of the posting period permitted.
DOCDATE	Document date YYYYMMDD	8	Date on which the original document was issued
DOCHEADERTXT	Document header text	25	The document header text contains explanations or notes which apply to the document

			as a whole
BACKFLQUANT	Quantity in unit of entry	7	Specifies the quantity to be moved in the unit of entry. The quantity is automatically converted to the stock-keeping unit. If the relevant unit of measure has not been defined in the material master record the system uses the stockeeping unit. If you do not enter a unit of measure SAP uses the following units of measure, order unit in goods receipts against purchase orders, production unit in goods receipts against production orders, unit of issue in other goods movement
SCRAPQUANT	Scrap quantity	7	Quantity of scrap recorded. In one transaction you can backflush either a yield or a scarp quantity. IF you want to backflush scrap enter the appropriate quantity.
UNITOFMEASURE	Unit of measure	3	You can enter the base unit of measure that is maintained in the material master record. If you have maintained conversion factors in the material master record you can also use the units of measure specified. If you enter nothing the system automatically copies the base unit of measure from the material master record.
UNITOFMEASURE_ISO	ISO code for unit of measure	3	
SCRAPREASON	Reason for scrap	4	You can enter the reasons for scrap in order to give more details in the scrap posting. However, this information is not processed any further by the system. Reasons could be for example damage to machinery, human error or material error.
REVLEVEL	Revision level	2	Revision level together with a change number identifies a version of a material or document. You can assign a

			revision level if you use a change number that has a specific valid from date to make a change.
PLANORDER	Planned order number	10	
ORDERCOSTS	Indicator Post with order costs	1	
INCLCOMPSCRAP	Indicator Post with component scrap	1	
NATERIALNR_EXTERNAL	Long material number future	40	
MATERIALNR_GUID	External GUID future	32	
MATERIALNR_VERSION	Version future	10	

Point Groups

Point_group_groups

point_group_no	point_group_member_no
649	650

Point_group

group_num	group_description	group_type	resource_id	plant_id
649	MTS	REM		1100
650	MTS_DATA	REMD		1100

Point_group_members

group_num	tag_id	tag_alias	display_order	server
group_num	**tag_id**	**tag_alias**	**display_order**	**server**
649	pp_ACTIVITIES_TYPE	ACTIVITIES_TYPE	4	PISERVER2
649	pp_BCKFLTYPE	BCKFLTYPE	2	PISERVER2
649	pp_COMPONENTS_TYPE	COMPONENTS_TYPE	5	PISERVER2
649	pp_CONFIRMATION_NO_MTS	CONFIRMATION_NO	7	PISERVER2
649	pp_REPPOINT_MTS	REPPOINT	1	PISERVER2
649	pp_RP_SCRAPTYPE	RP_SCRAPTYPE	3	PISERVER2
649	pp_SERIALNR_MTS	SERIALNR	6	PISERVER2

group_num	tag_id	tag_alias	display_order	server
649	Pp_state_MTS	STATE	15	PISERVER2
650	pp_BACKFLQUANT_MTS	BACKFLQUANT	13	PISERVER2
650	pp_BATCH_MTS	BATCH	9	PISERVER2
650	pp_DOCDATE_MTS	DOCDATE	11	PISERVER2
650	pp_DOCHEADERTXT_MTS	DOCHEADERTXT	12	PISERVER2
650	pp_INCLCOMPSCRAP	INCLCOMPSCRAP	21	PISERVER2
650	pp_MATERIALNR_MTS	MATERIALNR	2	PISERVER2
650	pp_ORDERCOSTS	ORDERCOSTS	20	PISERVER2
650	pp_PDC_NUMBER_MTS	PDC_NUMBER	1	PISERVER2
650	pp_PLANNINGID_MTS	PLANNINGID	8	PISERVER2
650	pp_PLANORDER_MTS	PLANORDER	19	PISERVER2
650	pp_PLANPLANT_MTS	PLANPLANT	4	PISERVER2
650	pp_POSTDATE_MTS	POSTDATE	10	PISERVER2
650	pp_PRODLINE_MTS	PRODLINE	7	PISERVER2
650	pp_PRODPLANT_MTS	PRODPLANT	3	PISERVER2
650	pp_PRODVERSION_MTS	PRODVERSION	6	PISERVER2
650	pp_REVLEVEL_MTS	REVLEVEL	18	PISERVER2
650	pp_SCRAPQUANT_MTS	SCRAPQUANT	14	PISERVER2
650	pp_SCRAPREASON_MTS	SCRAPREASON	17	PISERVER2
650	pp_STORAGELOC_MTS	STORAGELOC	5	PISERVER2
650	pp_UNITOFMEASURE_MTS	UNITOFMEASURE	15	PISERVER2
650	pp_UNITOFMEASURE_ISO_MTS	UNITOFMEASURE_ISO	16	PISERVER2

Sample of input before sending to SAP

Table pp_rm_datagen_in

request_id	method	Pdc_number	Materialnr	Prodplant	Planplant	Storageloc	Prodversion
1	MTS	OSI3	AM2-500	1000	1000	0001	0001

Prodline	Planningid	Batch	Postdate	Docdate
			20011009	20011009

Docheadertxt	Backflquant	Scrapquant	Unitofmeasure	Unitofmeasure_iso	Scrapreason	Revlevel

Docheadertxt	Backflquant	Scrapquant	Unitofmeasure	Unitofmeasure_iso	Scrapreason	Revlevel
	8.	0.				

Planorder	Ordercosts	Inclcompscrap	Materialnr_external	Materialnr_guid	Materialnr_version	confirmation	status	status_timestamp
						1362	W	10/10/2001 3:23:49 PM

Table pp_rm_flag_in

id	request_id	Bckfltype	Rp_scraptype	Activities_type	Components_type	prodlot	reppoint	status	status_timestamp
1	1	01							

Cancel Confirmation

CONFIRMATION_NO – the transferred backflush number is reversed. This can include material documents, activity documents and post-processing documents.

POSTDATE – The posting data is the date on which the reversal document is created. If no posting date is transferred the local system data of the user is then used as the posting date. Char(8)

CancPDCollNr – This parameter can contain a PDC backflush number. The number is saved with the reversal backflush document created by the system. This number is optional and is given by the caller. Char(12)

CancConfirmation – contains the number of the reversing backflush. This is the number of the backflush used to reverse the backflush transferred via the parameter CONFIRMATION char(10)

Point_group

group_num	group_description	group_type	process_book	resource_id	plant_id
651	CANCEL	REM			1100

Point_group_members

group_num	tag_id	tag_alias	display_order	server
651	pp_CANC_PDCOLLNR_cancel	CANC_PDCOLLNR	2	PISERVER2
651	pp_CANCCONFIRMATION_cancel	CANCCONFIRMATION	3	PISERVER2
651	pp_CONFIRMATION_CANCEL	CONFIRMATION	4	PISERVER2

group_num	tag_id	tag_alias	display_order	server
651	pp_POSTDATE_CANCEL	POSTDATE	1	PISERVER2
651	pp_STATE_CANCEL	STATE	5	PISERVER2

Data after sending to SAP the CancConfirmation is returned form SAP

id	ConfirmationNo	CancConfirmation	PostDate	CancPDCollNr	pitimestamp	status	status_timesta
1	1362	0000001372	20011010		10/10/2001 4:56:18 PM	W	

MTO – made to order

The backflush is carried out for a sales order. Depending on the stock category (valuated or non-valueated) the costs are collected at either a product cost collector or for the sales order.

Our implementation only allows one sales order with multiple items per request call of MTO function.

BflushFlags

BflushDatGen

SERIALNR

A sales order and/or a planned order must be transferred.

A postdate and a docdate must be transferred

The fields prodplant or Planplant must be filled depending on whether you are backflushing a yield quantity or a scarp quantity, you must fill one of fields BACKFQUANT or SCRAPQUANT. The field PLANORDER must be filled if the backflush is carried out specifically for a planned order.

You must also enter a posting date or a document date

The backflush number given by the system is returned to the caller via the parameter CONFIRMATION

BflushDataMTO

Parameter	Description	Format	Comments
SORDER	Sales order number	10	
SORDERITEM	An item within a sales order	6	

BCKFLTYPE only 01 is allowed and other flags are of no significance in make to order.

Point_group

Group_num	group_description	group_type	process_book	resource_id	plant_id
664	MTO	REM			1100
665	MTO_DATA	REMD			1100
666	MTO_DATA	REMS			1100

Point_group_groups

point_group_no	point_group_member_no
664	665
664	666

Point_group_members

group_num	tag_id	tag_alias	display_order	server
664	pp_BCKFLTYPE_MTO	BCKFLTYPE	2	PISERVER2
664	pp_REPPOINT_MTO	REPPOINT	1	PISERVER2
664	pp_SERIALNR_MTO	SERIALNR	6	PISERVER2
664	pp_STATE_MTO	STATE	15	PISERVER2
665	pp_BACKFLQUANT_MTO	BACKFLQUANT	13	PISERVER2
665	pp_BATCH_MT0	BATCH	9	PISERVER2
665	pp_CONFIRMATION_NO_MTO	CONFIRMATION	17	PISERVER2
665	pp_DOCDATE_MTO	DOCDATE	11	PISERVER2
665	pp_DOCHEADERTXT_MTO	DOCHEADERTXT	12	PISERVER2
665	pp_INCLCOMPSCRAP_MTO	INCLCOMPSCRAP	21	PISERVER2
665	pp_MATERIALNR_MTO	MATERIALNR	2	PISERVER2
665	pp_ORDERCOSTS_MTO	ORDERCOSTS	20	PISERVER2
665	pp_PDC_NUMBER_MTO	PDC_NUMBER	1	PISERVER2
665	pp_PLANNINGID_MTO	PLANNINGID	8	PISERVER2
665	pp_PLANORDER_MTO	PLANORDER	19	PISERVER2
665	pp_PLANPLANT_MTO	PLANPLANT	4	PISERVER2
665	pp_POSTDATE_MTO	POSTDATE	10	PISERVER2
665	pp_PRODLINE_MTO	PRODLINE	7	PISERVER2
665	pp_PRODPLANT_MTO	PRODPLANT	3	PISERVER2
665	pp_PRODVERSION_MTO	PRODVERSION	6	PISERVER2

group_num	tag_id	tag_alias	display_order	server
665	pp_REVLEVEL_MTO	REVLEVEL	18	PISERVER2
665	pp_SCRAPQUANT_MTO	SCRAPQUANT	14	PISERVER2
665	pp_SCRAPREASON_MTO	SCRAPREASON	17	PISERVER2
665	pp_STORAGELOC_MTO	STORAGELOC	5	PISERVER2
665	pp_UNITOFMEASURE_MTO	UNITOFMEASURE	15	PISERVER2
665	pp_UNITOFMEASURE_ISO_MTO	UNITOFMEASURE_ISO	16	PISERVER2
666	pp_SORDER	SORDER	1	PISERVER2
666	pp_SORDERITEM1	SORDERITEM	2	PISERVER2

In the tag for Sales order there will be one value at the state time. In the tag for the sales order item there will be multiple values at the state time.

MTP

MTP – you carry out a backflush for the production by lot scenario. The costs are collected at the production lot.

A production lot and a plant must be transferred.

You must enter a posting date and a document date.

You must either file BACKFLQUANT or SCRAPQUANT.

The field PLANORDER must be filled if the backflush is to be carried out for one particular planned order.

BflushDataMTP

Parameter	Description	Format	Comments
PRODLOT	Production lot	8	Must be filled. If on the parameter PRODLOT is transferred and several planned orders exist for this production lot the system cannot carry out the backflush as it cannot select a planned order automatically. In such a case enter the planned order as well as filling in the fields

			for the production lot.

BCKFLTYPE – The only allowed value is 01, and the other parameters are of no significance.

Point_group

Group_num	group_description	group_type	process_book	resource_id	plant_id
662	MTP	REM			1100
663	MTP_DATA	REMD			1100

Point_group_groups

point_group_no	point_group_member_no
662	663

Point_group_members

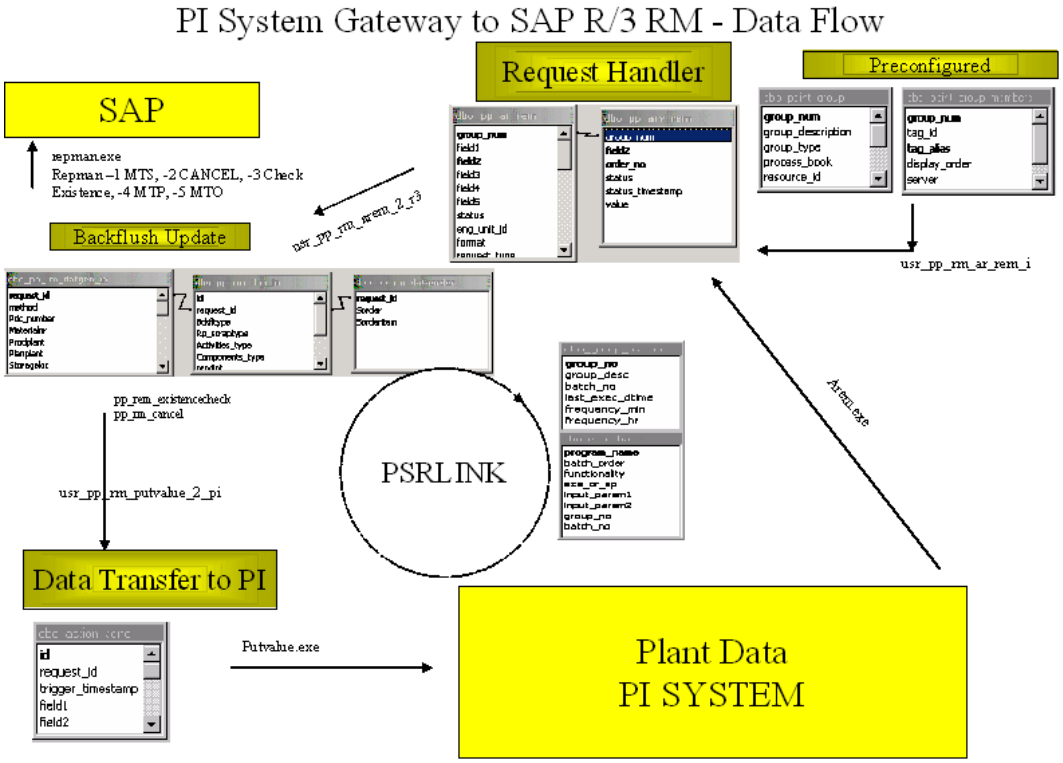
group_num	tag_id	tag_alias	display_order	server
662	pp_BCKFLTYPE_MTP	BCKFLTYPE	2	PISERVER2
662	pp_PRODLOT	PRODLOT	6	PISERVER2
662	pp_REPPOINT_MTP	REPPOINT	1	PISERVER2
662	pp_SERIALNR_MTP	SERIALNR	6	PISERVER2
662	pp_STATE_MTP	STATE	15	PISERVER2
663	pp_BACKFLQUANT_MTP	BACKFLQUANT	13	PISERVER2
663	pp_BATCH_MTP	BATCH	9	PISERVER2
663	pp_CONFIRMATION_NO_MTP	CONFIRMATION	17	PISERVER2
663	pp_DOCDATE_MTP	DOCDATE	11	PISERVER2
663	pp_DOCHEADERTXT_MTP	DOCHEADERTXT	12	PISERVER2
663	pp_INCLCOMPSCRAP_MTP	INCLCOMPSCRAP	21	PISERVER2
663	pp_MATERIALNR_MTP	MATERIALNR	2	PISERVER2
663	pp_ORDERCOSTS_MTP	ORDERCOSTS	20	PISERVER2
663	pp_PDC_NUMBER_MTP	PDC_NUMBER	1	PISERVER2
663	pp_PLANNINGID_MTP	PLANNINGID	8	PISERVER2
663	pp_PLANORDER_MTP	PLANORDER	19	PISERVER2
663	pp_PLANPLANT_MTP	PLANPLANT	4	PISERVER2
663	pp_POSTDATE_MTP	POSTDATE	10	PISERVER2

group_num	tag_id	tag_alias	display_order	server
663	pp_PRODLINE_MTP	PRODLINE	7	PISERVER2
663	pp_PRODPLANT_MTP	PRODPLANT	3	PISERVER2
663	pp_PRODVERSION_MTP	PRODVERSION	6	PISERVER2
663	pp_REVLEVEL_MTP	REVLEVEL	18	PISERVER2
663	pp_SCRAPQUANT_MTP	SCRAPQUANT	14	PISERVER2
663	pp_SCRAPREASON_MTP	SCRAPREASON	17	PISERVER2
663	pp_STORAGELOC_MTP	STORAGELOC	5	PISERVER2
663	pp_UNITOFMEASURE_MTP	UNITOFMEASURE	15	PISERVER2
663	pp_UNITOFMEASURE_ISO_MTP	UNITOFMEASURE_ISO	16	PISERVER2

ExistenceCheck

Existence Check– the system uses the parameter CONFIRMATION to check whether a backflush already exists in the system for this number. This is the backflush number given by SAP. If no backflush with the transferred number is found in the system you will get a return message.

CONFIRMATION – key field



Components

To append the components the user must modify the BAPI on the SAP side. SAP provides no way to send up the actual components. We have added the ability to retrieve the component data but this is only useful if the BAPI has been modified on the SAP side. This portion is delivered only by request.

Table

Pp_rm_comonents Holds the component data to be sent to SAP

Request_id		Request_id
MATNR	char(18)	Material number
ERFMG_R	char(13)	Quantity
ERFME	char(3)	Unit of measure
WERKS	char(4)	Plant
LGORT	char(4)	Storage Location
PRVBE	char(10	Supply area
POSNR_R	char(4)	BOM item number
CHARG	char(8)	Batch

Procedures

A new version of these procedures that support the added point_group type for components is installed

Usr_pp_rm_ar_rem_i	inserts new point_groups into pp_ar_rem
Usr_pp_rm_bapi_general	Selects data for BAPI call
Usr_pp_rm_arem_2_r3	Reads from actions_results and inserts into SAP like tables

Application

The application is repmanal.exe -1 for MTS

A new componet that supports the component table is Remal.

The interface dll remains with the same name rlinkrem.dll. It has a new class called IRem2. repmanal.exe is using IRem2.

TAGS

The following tags are required:

Need tag for plant

Need tag to hold blank

Need quantity tag for material

Need storage location tag for material

pp_BCKFLTYPE must be set to 01

pp_STATE_MTS

pp_CONFIRMATION_NO_MTS_1

Need tag to hold date string format of posting must be of the form YYYYMMDD

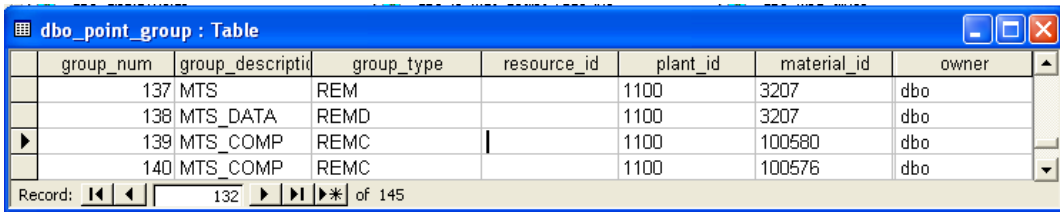
pp_PDC_NUMBER_MTS

Need tag for production version

Tag for unit of measure of each material or if they are all the same then one tag

Configuration

Point_groups

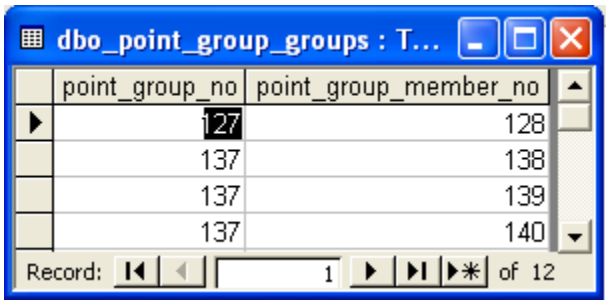


dbo_point_group : Table

group_num	group_descriptio	group_type	resource_id	plant_id	material_id	owner
137	MTS	REM		1100	3207	dbo
138	MTS_DATA	REMD		1100	3207	dbo
139	MTS_COMP	REMC		1100	100580	dbo
140	MTS_COMP	REMC		1100	100576	dbo

Record: 132 of 145

point_group_groups



dbo_point_group_groups : T...

point_group_no	point_group_member_no
127	128
137	138
137	139
137	140

Record: 1 of 12

Point_group_members

group_num	tag_id	tag_alias	display_order	server
137	mm_blank	ACTIVITIES_TYPE	4	PISERVER2
137	pp_BCKFLTYPE	BCKFLTYPE	2	PISERVER2
137	mm_blank	COMPONENTS_TYPE	5	PISERVER2
137	mm_blank	REPPOINT	1	piserver2
137	mm_blank	RP_SCRAPTYPE	3	PISERVER2
137	mm_blank	SERIALNR	6	PISERVER2
137	pp_STATE_MTS	STATE	15	PISERVER2
138	Need quantity tag	BACKFLQUANT	13	PISERVER2
138	mm_blank	BATCH	9	PISERVER2
138	pp_CONFIRMATION_NO_MTS_1	CONFIRMATION	17	Piserver2
138	pp_DOCDATE_MTS_1	DOCDATE	11	PISERVER2
138	mm_blank	DOCHEADERTXT	12	PISERVER2

group_num	tag_id	tag_alias	display_order	server
138	mm_blank	INCLCOMPSCRAP	21	PISERVER2
138	Product.SAP	MATERIALNR	2	piserver2
138	mm_blank	ORDERCOSTS	20	PISERVER2
138	pp_PDC_NUMBER_MTS	PDC_NUMBER	1	piserver2
138	mm_blank	PLANNINGID	8	PISERVER2
138	mm_blank	PLANORDER	19	PISERVER2
138	Need tag for plant	PLANPLANT	4	PISERVER2
138	pp_POSTDATE_MTS_1	POSTDATE	10	PISERVER2
138	mm_blank	PRODLINE	7	PISERVER2
138	Need tag for plant	PRODPLANT	3	PISERVER2
138	pp_PRODVERSION_MTS_1	PRODVERSION	6	PISERVER2
138	mm_blank	REVLEVEL	18	PISERVER2
138	mm_blank	SCRAPQUANT	14	PISERVER2
138	mm_blank	SCRAPREASON	17	PISERVER2
138	Need storage location	STORAGELOC	5	PISERVER2
138	pp_UNITOFMEASURE_MTS_1	UNITOFMEASURE	15	PISERVER2
138	mm_blank	UNITOFMEASURE_ISO	16	PISERVER2
139	Component1.E	ERFME	1	PISERVER2
139	Need tag for plant	ERFMG_R	1	PISERVER2
139	Component1.STO	LGORT	1	PISERVER2
139	Component1.MAT	MATNR	1	PISERVER2
139	Need tag for plant	WERKS	1	PISERVER2
140	x	ERFME	1	PISERVER2
140	Component2.E	ERFMG_R	1	PISERVER2
140	Component2.STO	LGORT	1	PISERVER2
140	Component2.MAT	MATNR	1	PISERVER2
140	Need tag for plant	WERKS	1	PISERVER2

Chapter 18
Material Movements

Installation

The Material Movements is delivered as an incremental install to PP-PI and it requires that the SAP DCOM portion of the install be completed. After the PP-PI install you run the script.bat file that is supplied with the material movement upgrade and execute the program DepSrvComp.exe to deploy the components.

BAPI's Supported

BAPI_GOODSMVT_CREATE - posts a goods movement

BAPI_GOODSMVT_CANCEL – Reverse goods movement

Procedures

Procedure Name	**Purpose**
Usr_mm_gm_ar_i	Populates mm_gm_ar table with goods movement tags for which mmtran.exe application will look for data in PI. There are 3 possible trigger procedures. Mmtran_other2 is used for the cancel. Multi is used for the items of the movement and gettag is used for the header and code.
Usr_mm_gm_ar_upd	Called by mmtran application to populate value into mm_gm_arv, mm_gm_status_detail table
Usr_mm_gm_build	Scans mm_gm_arv table and prepares data in the input tables to create goods movement document
Usr_mm_gm_to_pi	Populates action_send table with material document and year data with goods movement start and end times to set the values in PI
Usr_mm_gm_general	All applications developed for goods movement will use this pplicati for selecting data from plant_suite database
Usr_mm_gm_sel	Mmtran application uses this procedure to select records from mm_gm_ar table
Usr_mm_gm_status_u	Goods movement bapi application uses this procedure to update the status of the processed records

usr_mm_gm_to_pi2	Populates action_send table with material document and year data for cancelled goods movement document to set the values in PI
usr_mm_gm_build2	Scans mm_gm_arv table and prepares data in the input tables to cancel goods movement document.
Usr_mm_gm_ar_upd2	Called by mmtran application to populate value into mm_gm_arv table.

Applications

Rlbogm.exe –1	Transfers the material movement to SAP
Rlbogm.exe –2	Transfers the cancel movement to SAP
Mmtran.exe	Monitors the CODE-STATE tag for changes if there is a changed value then calls usr_mm_ar_upd for CODE and STATE and collectes the rest of the data at that time.

Tables

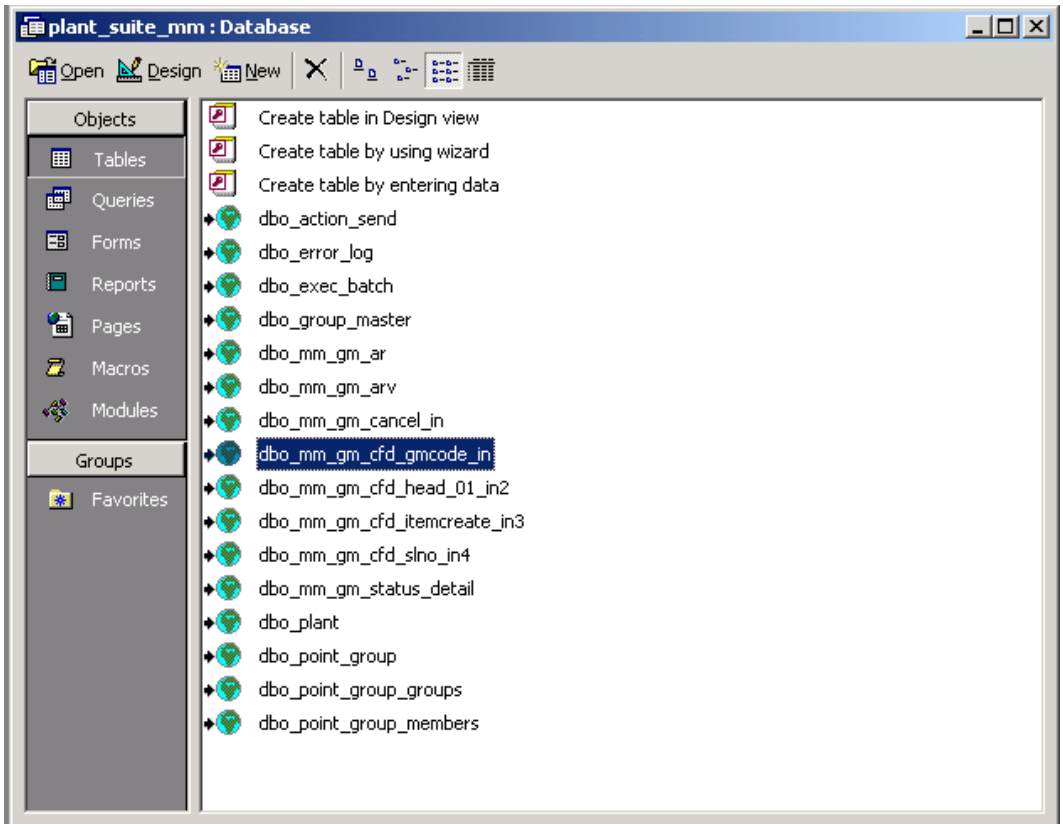


Table	Purpose
mm_gm_cfd_gmcode_in (code table plus pi_end_time, Pi_start_time)	GOODSMVT_CODE table described in SAP description

mm_gm_cfd_head_01_in2	GOODSMVT_HEADER table described in SAP description
mm_gm_cfd_itemcreate_in3	GOODSMVT_ITEM table described in SAP description
mm_gm_cfd_slno_in4	GOODSMVT_SERIALNUMBER table described in SAP description
mm_gm_status_detail	Keeps the start and end time of a movement for a point group
mm_gm_ar	material movement points to monitor
mm_gm_arv	results returned from material movements values
mm_gm_cancel_in	Used to cancel a material movement

Point_group and point_group_members and point_group_groups

The point_group_members that are not having a value recorded do not need to be in the group. The following includes the blank members for documentation completion.

Point_group

group_num	group_description	group_type	resource_id	plant_id
75	GOODSMVT_HEADER	MM_HEADER	R_1111	1100
76	GOODSMVT_CODE	MM_CODE	R_1111	1100
77	GOODSMVT_ITEM	MM_ITEM	R_1111	1100
78	GOODSMVT_SERIALNUMBER	MM_SERIAL	R_1111	1100
116	GOODSMVT_CANCEL	MM_CANCEL	R_1111	1100

Point_group_groups

point_group_no	point_group_member_no
76	75
76	77
76	78

Point_group_members

group_num	tag_id	tag_alias	display_order	server
75	mm_blank	BILL_OF_LADING	4	PISERVER2
75	mm_doc_date	DOC_DATE	2	PISERVER2
75	mm_blank	EXT_WMS	10	PISERVER2
75	mm_blank	GR_GI_SLIP_NO	5	PISERVER2

group_num	tag_id	tag_alias	display_order	server
75	mm_blank	HEADER_TXT	7	PISERVER2
75	mm_blank	PR_UNAME	6	PISERVER2
75	mm_post_date	PSTNG_DATE	1	PISERVER2
75	mm_blank	REF_DOC_NO	3	PISERVER2
75	mm_blank	VER_GR_GI_SLIP	8	PISERVER2
75	mm_blank	VER_GR_GI_SLIPX	9	PISERVER2
76	mm_doc_year	DOC_YEAR	5	PISERVER2
76	mm_codevalue	GM_CODE	2	PISERVER2
76	mm_mat_doc	MAT_DOC	4	PISERVER2
76	mm_STATE-value	STATE	1	PISERVER2
76	mm_blank	TEST_RUN	3	PISERVER2
77	mm_blank	ACTIVITY	57	PISERVER2
77	mm_blank	ACTTYPE	101	PISERVER2
77	mm_blank	AMOUNT_LC	59	PISERVER2
77	mm_blank	AMOUNT_SV	60	PISERVER2
77	mm_blank	ASSET_NO	37	PISERVER2
77	mm_Batch	BATCH	6	PISERVER2
77	mm_blank	CALC_MOTIVE	36	PISERVER2
77	mm_blank	CMMT_ITEM	68	PISERVER2
77	mm_blank	CO_BUSPROC	100	PISERVER2
77	mm_blank	COMP_SHIP	28	PISERVER2
77	mm_blank	COST_OBJ	52	PISERVER2
77	mm_blank	COSTCENTER	33	PISERVER2
77	mm_blank	CUSTOMER	14	PISERVER2
77	mm_blank	DELIV_ITEM	113	PISERVER2
77	mm_blank	DELIV_ITEM_TO_SEARCH	77	PISERVER2
77	mm_blank	DELIV_NUMB	112	PISERVER2
77	mm_blank	DELIV_NUMB_TO_SEARCH	76	PISERVER2
77	mm_blank	EAN_UPC	75	PISERVER2
77	mm_material_qty	ENTRY_QNT	4	PISERVER2
77	mm_qty_UOM	ENTRY_UOM	5	PISERVER2

group_num	tag_id	tag_alias	display_order	server
77	mm_blank	ENTRY_UOM_ISO	21	PISERVER2
77	mm_blank	EXPIRYDATE	64	PISERVER2
77	mm_blank	FUNC_AREA	109	PISERVER2
77	mm_blank	FUND	66	PISERVER2
77	mm_blank	FUNDS_CTR	67	PISERVER2
77	mm_blank	GL_ACCOUNT	72	PISERVER2
77	mm_blank	GR_NUMBER	93	PISERVER2
77	mm_blank	GR_RCPT	31	PISERVER2
77	mm_blank	GR_RCPTX	117	PISERVER2
77	mm_blank	IND_PROPOSE_QUANX	73	PISERVER2
77	mm_blank	ITEM_TEXT	30	PISERVER2
77	mm_blank	MATDOC_TR_CANCEL	96	PISERVER2
77	mm_Material	MATERIAL	1	PISERVER2
77	mm_blank	MATERIAL_EXTERNAL	103	PISERVER2
77	mm_blank	MATERIAL_GUID	104	PISERVER2
77	mm_blank	MATERIAL_VERSION	105	PISERVER2
77	mm_blank	MATITEM_TR_CANCEL	97	PISERVER2
77	mm_blank	MATYEAR_TR_CANCEL	98	PISERVER2
77	mm_blank	MOVE_BATCH	46	PISERVER2
77	mm_blank	MOVE_MAT	43	PISERVER2
77	mm_blank	MOVE_MAT_EXTERNAL	106	PISERVER2
77	mm_blank	MOVE_MAT_GUID	107	PISERVER2
77	mm_blank	MOVE_MAT_VERSION	108	PISERVER2
77	mm_blank	MOVE_PLANT	44	PISERVER2
77	mm_blank	MOVE_REAS	49	PISERVER2
77	mm_blank	MOVE_STOC	45	PISERVER2
77	mm_MT	MOVE_TYPE	7	PISERVER2
77	mm_blank	MOVE_VAL_TYPE	46	PISERVER2
77	mm_blank	MVT_IND	47	PISERVER2
77	mm_blank	NB_SLIPS	114	PISERVER2
77	mm_blank	NB_SLIPSX	115	PISERVER2

group_num	tag_id	tag_alias	display_order	server
77	mm_blank	NETWORK	56	PISERVER2
77	mm_blank	NO_MORE_GR	29	PISERVER2
77	mm_blank	NO_PST_CHGNT	92	PISERVER2
77	mm_blank	NO_TRANSFER_REQ	99	PISERVER2
77	mm_blank	ORDER_ITNO	35	PISERVER2
77	mm_blank	ORDERID	34	PISERVER2
77	mm_blank	ORDERPR_UN	23	PISERVER2
77	mm_blank	ORDERPR_UN_ISO	24	PISERVER2
77	mm_blank	PAR_COMPCO	111	PISERVER2
77	mm_blank	PART_ACCT	58	PISERVER2
77	mm_Plant	PLANT	2	PISERVER2
77	mm_blank	PO_ITEM	26	PISERVER2
77	mm_blank	PO_NUMBER	25	PISERVER2
77	mm_blank	PO_PR_QNT	22	PISERVER2
77	mm_blank	PROD_DATE	65	PISERVER2
77	mm_blank	PROFIT_CTR	54	PISERVER2
77	mm_blank	PROFIT_SEGM_NO	53	PISERVER2
77	mm_blank	REF_DATE	51	PISERVER2
77	mm_blank	REF_DOC	62	PISERVER2
77	mm_blank	REF_DOC_IT	63	PISERVER2
77	mm_blank	REF_DOC_YR	61	PISERVER2
77	mm_blank	RES_ITEM	40	PISERVER2
77	mm_blank	RES_TYPE	41	PISERVER2
77	mm_blank	RESERV_NO	39	PISERVER2
77	mm_blank	RL_EST_KEY	50	PISERVER2
77	mm_blank	S_ORD_ITEM	16	PISERVER2
77	mm_blank	SALES_ORD	15	PISERVER2
77	mm_blank	SCHED_LINE	17	PISERVER2
77	mm_blank	SERIALNO_AUTO_NUMBER ASSIGNMENTI	78	PISERVER2
77	mm_blank	SHIPPING	27	PISERVER2
77	mm_blank	SPEC_STOCK	12	PISERVER2

group_num	tag_id	tag_alias	display_order	server
77	mm_blank	ST_PL_STCK_2	86	PISERVER2
77	mm_blank	ST_UN_QTYY_1	83	PISERVER2
77	mm_blank	ST_UN_QTYY_1_ISO	84	PISERVER2
77	mm_blank	ST_UN_QTYY_2	87	PISERVER2
77	mm_blank	ST_UN_QTYY_2_ISO	88	PISERVER2
77	mm_blank	STCK_TYPE	11	PISERVER2
77	mm_blank	STG_BIN	81	PISERVER2
77	mm_blank	STG_TYPE_ST	94	PISERVER2
77	mm_blank	STGE_BIN_PC	91	PISERVER2
77	mm_blank	STGE_BIN_ST	95	PISERVER2
77	mm_Storage	STGE_LOC	3	PISERVER2
77	mm_blank	STGE_TYPE	80	PISERVER2
77	mm_blank	STGE_TYPE_PC	90	PISERVER2
77	mm_blank	SU_PL_STCK_1	82	PISERVER2
77	mm_blank	SUB_NUMBER	38	PISERVER2
77	mm_blank	SUPPL_VEND	102	PISERVER2
77	mm_blank	TR_PART_BA	110	PISERVER2
77	mm_blank	UNITTYPE_1	85	PISERVER2
77	mm_blank	UNITTYPE_2	89	PISERVER2
77	mm_blank	UNLOAD_PT	32	PISERVER2
77	mm_blank	UNLOAD_PTX	8	PISERVER2
77	mm_blank	VAL_S_ORD_ITEM	70	PISERVER2
77	mm_blank	VAL_SALES_ORD	69	PISERVER2
77	mm_blank	VAL_TYPE	18	PISERVER2
77	mm_blank	VAL_WBS_ELEM	71	PISERVER2
77	mm_blank	VENDOR	13	PISERVER2
77	mm_blank	VENDRBATCH	79	PISERVER2
77	mm_blank	WBS_ELEM	55	PISERVER2
77	mm_blank	WITHDRAWN	42	PISERVER2
77	mm_blank	XSTOB	74	PISERVER2
78	mm_blank	MATDOC_ITM	1	PISERVER2

group_num	tag_id	tag_alias	display_order	server
78	mm_blank	SERIALNO	2	PISERVER2
116	mm_doc_year_C_R	DOC_YEAR	5	PISERVER2
116	mm_doc_year_C	DOC_YEAR_CANCEL	8	PISERVER2
116	mm_mat_doc_C_R	MAT_DOC	4	PISERVER2
116	mm_mat_doc_C	MAT_DOC_CANCEL	7	PISERVER2
116	mm_mat_doc_item_C	MATDOC_ITEM	3	PISERVER2
116	mm_pr_name_C	PR_NAME	2	PISERVER2
116	mm_pstng_date_C	PSTNG_DATE	6	PISERVER2
116	mm_STATE-C	STATE	1	PISERVER2

Group_master and exec_batch

group_master table data

group_no	group_desc	batch_no	last_exec_dtime	frequency_min	frequency_hr
56	MM-Processing	1	7/12/2002 11:11:48 AM	1	0

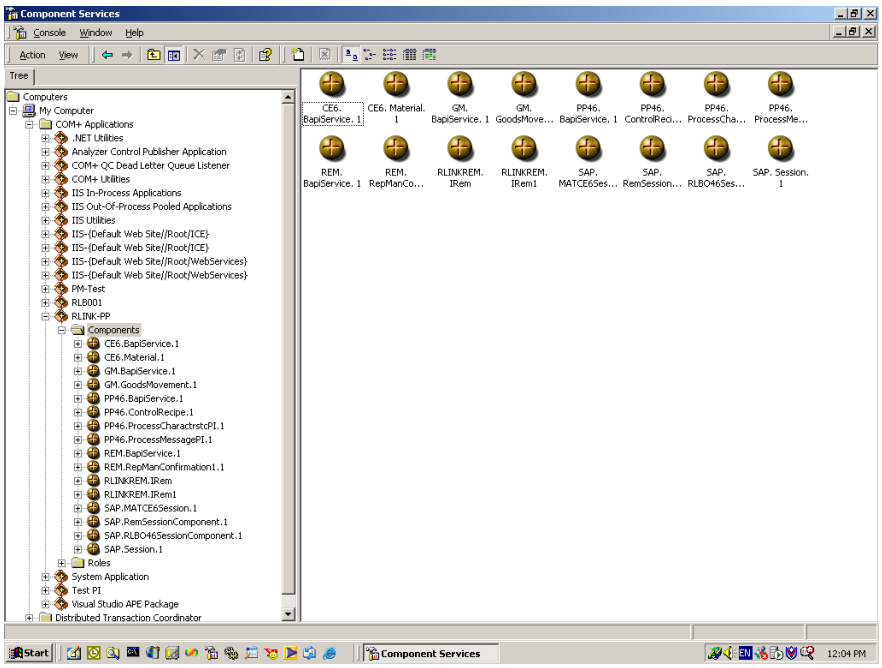
exec_batch table data

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
usr_mm_gm_ar_I	1	MM-insertion of rec into mm_gm_ar	P	56	1
usr_mm_gm_build	4	MM-Builds SAP reply for doc Creation	P	56	1
usr_mm_gm_build2	7	MM-Builds SAP reply for doc Cancel	P	56	1
usr_mm_gm_to_pi	5	MM-Sends data to PI for created doc	P	56	1
usr_mm_gm_to_pi2	8	MM-Sends data to PI for cancelled doc	P	56	1
d:\rlink\pppi\server\fe\mmtran.exe	2	MM-Gets goods movement data from PI	E	56	1
d:\rlink\pppi\server\fe\rlbogm.exe -1	3	MM-Creates	E	56	1

program_name	batch_order	functionality	exe_or_sp	group_no	batch_no
		goods movement document using BAPI			
d:\rlink\pppi\server\fe\rlbogm.exe -2	6	MM-Cancels goods movement document	E	56	1

Components

The components used for the material movements are GM.BapiService, GM.GoodsMovement and SAP.Session.



SAP Descriptions

Goods receipt for purchase order

Code 01 movement indicator B

Required

Purchase order

Purchase order item

Movement type

Movement indicator

Quantity in unit of entry

ISO code unit of measure for unit of entry

Some cases

- Shelf life expiration date
- Reason for movement
- Batch
- Storage location

Optional

- Stock type
- Item text
- Unloading point
- Delivery completed indicator

Cannot fill in

- Account assignment fields
- Reservation
- Receiving/issuing material
- Receiving/issuing plant
- Receiving/issuing storage location

Pucrhase order unknown shipping notification known

- Must be filled
  - Deliver
  - Delivery item
  - Movement type

Purchase order known, purchase order should be created automatically

- Must be filled
  - Material number
  - Plant
  - Storage location
  - Vendor
  - Movement type
  - Movement indicator
  - Quantity in unit of entry
  - ISO code unit of measurement of unit of entry

Purchase order unknown a purchase order should not be created

- Must be filled
  - Material number
  - Plant
  - Storage location

Vendor

Movement type

Movement indicator

Quantity in unit of entry

ISO code unit of measurement for unit of entry

Some cases

Shelf life expiration date

Reason for movement

Batch

Optional

Special stock indicator

Item text

Unloading point

Goods recipient

Cannot be filled

Account assignment

Reservation

Receiving/ issuing material

Receiving / issuing plant

Receiving /issuing storage location

Receiving / issuing batch

Goods receipt for production order

Code 02 movement indicator F

Must be filled

Order

Movement type

Movement indicator

Quantity in unit of entry

ISO code unit of measurement

Some cases

Shelf life expiration date

Reason for movement

Batch

Storage location

Can be filled

Order item (co-product)

Stock type

Item text

Unloading point

Delivery completed indicator

Cannot be filled

Account assignment fileds

Reservation

Receiving/issuing material

Receiving/issuing plant

Receiving/issuing storage location

Receiving/issuing batch

Goods Issue

Code 03 movement indicator blank

Without reference to a reservation

Must be filled

Material number

Plant

Storage location

Movement type

Movement indicator

Quantity in unit of enty

ISO code unit of measurement for unit of entry

Some cases

Special stock (eq. Sales order, project, vendor)

Shelf life expiration date

Reason for movement

Batch

Account assignment fields

Can be filled

Special stock indicator

Item text

Unloading point

Goods recipient

Cannot be filled

Reservation

Receiving/issuing material

Receiving/issuing plant

Receiving/issuing storage location

Receiving/issuing batch

With reference to a reservation

Must be filled

Reservation number

Reservation item

Record type of the reservation

Movement indicator

Quantity in unit of entry

ISO code unit of measurement for unit of entry

Some cases

Shelf life expiration date

Reason for movement

Batch

Storage location

Can be filled

Special stock indicator

Item text

Unloading point

Goods recipient

Cannot be filled

Movement type

Material

Plant

Accounting assignment fields

Transfer posting

Code 04 movement indicator blank

Transfer posting without reference to a reservation

Must be filled

Material number

Plant

Storage location

Movement type

Movement indicator

Quantity in unit of entry

ISO code unit of measurement for unit of entry

Some cases

Receiving material

Receiving plant

Receiving storage location

Receiving batch

Receiving/issusing special stock

Shelf life expiration date

Reason for movement

Batch

Can be filled

Special stock indicator

Item text

Account assignment fields

Cannot be filled

Reservation

Transfer posting with reference to reservation see goods issue with reference to reservation

Other goods receipts

Code 05 movement indicator blank

Without reference to a reservation see goods issue without reference to a reservation

With reference to a reservation see goods issue with reference to a reservation

Reversal of goods movement

Code 06 movement indicator blank

You can use the cancel method to reverse goods movments but you can also use this if you do not want to reference a material document.

Set the appropriate value in BAPI_GM_ITEM_CREATE_XSTOB by enterining in GOODSMVT_ITEM.XSTOB. Or set movement type to reversal and leave BAPI_GM_ITEM_CREATE_XSTOB blank by setting GOODSMVT_ITEM.XSTOB to blank.

SAP Table Descriptions

GOODSMVT_HEADER

Field	Description
PSTNG_DATE	Posting date
DOC_DATE	Document dat
REF_DOC_NO	Reference document number
BILL_OF_LADING	Nuber of bill of lading at time of goods receipt
GR_GI_SLIP_NO	Goods receipt/issue spli number
PR_UNAME	User name

Field	Description
HEADER_TXT	Document header text
VER_GR_GI_SLIP	Version fro printing GR/GI slip
VER_GR_GI_SLIPX	Updated information in related user data field
EXT_WMS	Control posting for external WMS (warehouse management system)

GOODSMVT_CODE

Field	Description
DOC_YEAR	Returned value for the calendar year in which the material document was posted
GM_CODE	Assign code to BAPI for goods movement
MAT_DOC	Returned value for the number of the material document
TEST_RUN	Indicator of test run

GOODSMVT_SERIALNUMBER

Field	Description
MATDOC_ITM	Item in material document, number that uniquely identifies a item
SERIALNO	Serial number

GOODSMVT_ITEM

Field	Description
MATERIAL	Material number
PLANT	Plant
STGE_LOC	Storage location
BATCH	Batch number
MOVE_TYPE	Movement type
STCK_TYPE	Stock type
SPEC_STOCK	Special stock indicator (consignment stock)
VENDOR	Vendor account number
CUSTOMER	Account number of customer
SALES_ORD	Sales order number
S_ORD_ITEM	Item number in sales order
SCHED_LINE	Delivery schedule for sales order

Field	Description
VAL_TYPE	Valuation type uniquely identifies separately valuated stocks of material If a material is valuated according to its origin (valuation category H) you can define the possible countries of origin as valuation types
ENTRY_QNT	Quantity in unit of entry
ENTRY_UOM	Unit of entry default values order unit for goods receipts with reference to purchase orders, the production unit for goods receipts with reference to production orders, unit of issue for other goods movements. If not defined stock-keeping unit
ENTRY_UOM_ISO	ISO code for unit o measurement
PO_PR_QNT	Quantity in purchase order price unit
ORDERPR_UN	Order price unit (purchasing)
ORDERPR_UN_ISO	ISO code for unit of measurement
PO_NUMBER	Purchase order number
PO_ITEM	Item number of purchasing document
SHIPPING	Shipping instructions
COMP_SHIP	Compliance with shipping instructions
NO_MORE_GR	Deliver completed indicator
ITEM_TEXT	Item text start with *
GR_RCPT	Goods recipient
UNLOAD_PT	Unloading point
COSTCENTER	Cost center
ORDERID	Order number
ORDER_ITNO	Order item number
CALC_MOTIVE	Accounting indicator
ASSET_NO	Main asset number
SUB_NUMBER	Asset sub-number
RESERV_NO	Number of reservation/dependent requirements
RES_ITEM	Item number of reservation/dependent requirements
RES_TYPE	Record type
WITHDRAWN	Final issue for this reservation (The indicator is set automatically for a goods movement when the total reserved

Field	Description
	quantity has been withdrawn or delivered. In the case of a partial delivery, you can manually set the indicator if no further goods movement are expected in respect of the relevant reservation item)
MOVE_MAT	Receiving/issuing material
MOVE_PLANT	Receiving plant/issuing plant
MOVE_STOC	Receiving/issuing storage location
MOVE_VAL_TYPE	Valuation type of transfer batch key used in split valuation of materials (that is the separate valuation of different stocks of the same material) to permit stocks of a transfer batch to be differentiated according to different criteria
MOVE_BATCH	Receiving/issuing batch
MVT_IND	Movement indicator (type of document such as purchase order or delivery note that constitutes the basis for the movement, derived from the transaction code. This indicator is necessary to enable a distinction to be made between a goods receipt for a purchase order and a goods receipt for a production order.
MOVE_REAS	Reason for movement
RL_EST_KEY	Internal key for real estate object
REF_DATE	Reference data for settlement, is used to identify the settlement period for invoice account assignment. You always have to enter the reference data when making posting on settlement units
COST_OBJ	Cost object activity based costing CO-OM-ABC, cost for intangible goods and services CO-PC-OBJ, product cost by period CO-PC-OBJ product cost controlling information system CO-PC
PROFIT_SEGM_NO	Profitability segment number (CO-PA)
PROFIT_CTR	Profit center
WBS_ELEM	Work breakdown structure element
NETWORK	Network number for account assignment
ACTIVITY	Operation number determines in which order the operations of a sequence are cared out in production planning, number that identifies and activity in

Field	Description
	project systems
PART_ACCT	Partner account number
AMOUNT_LC	Externally entered posting amount in local currency
AMOUNT_SV	Externally entered sales value in local currency
REF_DOC_YR	Fiscal year of a reference document
REF_DOC	Document number of a reference document
REF_DOC_IT	Item of a reference document
EXPIRYDATE	Shelf life expiration date
PROD_DATE	Date of production of the batch
FUND	Key which uniquely identifies the fund
FUNDS_CTR	Funds center
CMMT_ITEM	Commitment item
VAL_SALES_ORD	Sales order number of valuated sales order stock
VAL_S_ORD_ITEM	Sales order item of valuated sales order stock
VAL_WBS_ELEM	Work breakdown structure element
GL_ACCOUNT	G/L account number
IND_PROPOSE_QUANX	Specifies that the quantity is suggested This indicate is used when calling the function module MB_CREATE_GOODS_MOVEMENT and controls whether the quantity is preset by the calling program or whether the function module is to re-determing the quantity (for example the quantity still open for the order item, reservation or production order)
XSTOB	Use reversal movement type indicator
EAN_UPC	International Article number
DELIV_NUMB_TO_SEARCH	The number that uniquely identifies the delivery
DELIV_ITEM_TO_SEARCH	The number that uniquely identifies the item in a delivery
SERIALNO_AUTO_NUMBERASSIGNMENTI	Indicate that the system automatically created the required serial number for the item if there are not enough serial numbers

Field	Description
VENDRBATCH	Vendor batch number
STGE_TYPE	Storage type is a subdivision of a complex, physical warehouse. Different storage types are identified by their warehousing technique, form of organization or their function. A typical warehouse could have the following storage type, goods receipt are, picking area, high-rack storage area, bulk storage area, open storage area, goods issue area
STG_BIN	Storage bin or slot, exact location in the warehouse where goods are stored. A storage bin can be sub divided into bin sections. Several different material quantities can be stored in one bin at a time.
SU_PL_STCK_1	Number of storage units to be place into storage.
ST_UN_QTYY_1	Quantity per storage unit to be place into sock in alt. UoM
ST_UN_QTYY_1_ISO	ISO code for unit of measurement
UNITTYPE_1	Storage unit type
ST_PL_STCK_2	Number of storage units to be placed into storage
ST_UN_QTYY_2	Specifies which quantity is required for a storage unit
ST_UN_QTYY_2_ISO	ISO code for unit of measurement
UNITTYPE_2	Storage unit type
STGE_TYPE_PC	Storage type for transfer posting
STGE_BIN_PC	Storage bin for transfer posting
NO_PST_CHGNT	Indicator do not create posting change notice
GR_NUMBER	Goods receipt number
STG_TYPE_ST	Storage type for stock transfer
STGE_BIN_ST	Storage bin for stock transfer
MATDOC_TR_CANCEL	Material doc no of transfer requirement to be cancelled
MATITEM_TR_CANCEL	Material doc item of transfer requirement item to be cancelled
MATYEAR_TR_CANCEL	Material doc year of transfer requirement to be cancelled the material document year together with

Field	Description
	the document number forms the key that is used to access a material document
NO_TRANSFER_REQ	Indicator no transfer requirement created.
CO_BUSPROC	business process
ACTTYPE	Activity types describe the activity produced by a cost center and are measured in units of time or quantity
SUPPL_VEND	Supplying vendor
MATERIAL_EXTERNAL	Future
MATERIAL_GUID	Future
MATERIAL_VERSION	Future
MOVE_MAT_EXTERNAL	Future
MOVE_MAT_GUID	Future
MOVE_MAT_VERSION	Future
FUNC_AREA	Functional area such as manufacturing, administration, sales and distribution and research and development
TR_PART_BA	Trading partners business area
PAR_COMPCO	Clearing company code
DELIV_NUMB	Delivery
DELIV_ITEM	Delivery item
NB_SLIPS	Number of GR/GI slips to be printed
NB_SLIPSX	Updated information in related user data field
GR_RCPTX	Updated information in related user data field
UNLOAD_PTX	Updated information in related user data field

Cancel of Goods Movement

Only one document can be cancelled with each call. SAP does not delete the original document it creates a new material document that is a reversal.

The following authorizationobject is check when this method is used

M_MSEG_WMB material documents :plant

M_MSEG_BMB material documents:movement type

Once the system has successfully cancelled a material document it returs the material document number and material document year key fields.

This data is sent to SAP in the tables mm_cancel_in.

Field	Description
MAT_DOC	Material document returned
DOC_YEAR	Material document year returned
PSTNG_DATE	Posting date
PR_UNAME	User name, the user name is required if you want the system to print a goods receipt/issue slip when the material document is posted
MATDOC_ITEM	Document item to be cancelled. If no items are trandferred all the items in the material document are cancelled.
MAT_DOC_CANCEL	Material document to be cancelled
DOC_YEAR_CANCEL	Document year to be cancelled

PI Tag Definitions

*create MM movement tags which are reals

@table pplicat

@ptclass classic

@mode create,t

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,span,shutdown,compressing

mm_material_qty,mm material quantity,KG,L,Float32,10000,off,off,

@endsection

*create MM movement tags which are strings

@table pplicat

@ptclass classic

@mode create,t

@stype delimited

@istr tag,descriptor,EngUnits,pointsource,pointtype,shutdown,compressing

mm_Batch_other,PP Other Batch,,L,string,off,off,

mm_Material_other,PP Material Other,,L,string,off,off,

mm_Plant_other,PP Plant other,,L,string,off,off,

mm_Storage_other,PP Storage Location other ,,L,string,off,off,

mm_Storage,PP Storage Location,,L,string,off,off,

mm_blank,mm blank,,L,string,off,off,

```
mm_mat_doc,mm material_doc,,L,string,off,off,
mm_doc_year,mm doc year,,L,string,off,off,
mm_doc_date,mm doc date,,L,string,off,off,
mm_post_date,mm posting date,,L,string,off,off,
mm_Batch,PP Batch,,L,string,off,off,
mm_Material,PP Material,,L,string,off,off,
mm_Plant,PP Plant of Resource ,,L,string,off,off,
mm_MT,PP Movement type,,L,string,off,off,
mm_qty_UOM,MM QTY UOM,,L,string,off,off,
mm_mat_doc_C,mm material_doc cancel,,L,string,off,off,
mm_doc_year_C,mm doc year cancel,,L,string,off,off,
mm_pstng_date_C ,mm posting date cancel,,L,string,off,off,
mm_mat_doc_item_C,mm mat doc item cancel,,L,string,off,off,
mm_pr_name_C ,mm pr name cancel,,L,string,off,off,
mm_mat_doc_C_R,mm material_doc cancel return,,L,string,off,off,
mm_doc_year_C_R,mm doc year cancel retrun,,L,string,off,off,
@endsection

*create digital states for MM
@table pids
@mode create,t
@istructure set, state, ...
mm_state,00000,00001,00002
MM_CODE,01,02,03,04,05,06
@endsection

*create MM states
@table    pplicat
@mode create,t
@istructure tag, descriptor, digitalset, pointtype,shutdown,compressing
mm_state-value,PP state,mm_state,digital,off,off,
mm_CODEVALUE,PP state,mm_CODE,digital,off,off,
mm_STATE-C ,PP state mm cancel,mm_state,digital,off,off,
@endsection
```

Recording Movements

Data would be recorded in the PI tags as follows:

Movement 1 Endtime Tags in the point group CODE are filled out and the status is given the value 00002

Movement 2 Starttime (MM! End time + 1 sec) Tags associated with the point group HEADER and CODE are filled out the status tag is given the value 00001

MM2 Item1 + 1 sec Tags in the point group ITEM and SERIAL if needed are filled out

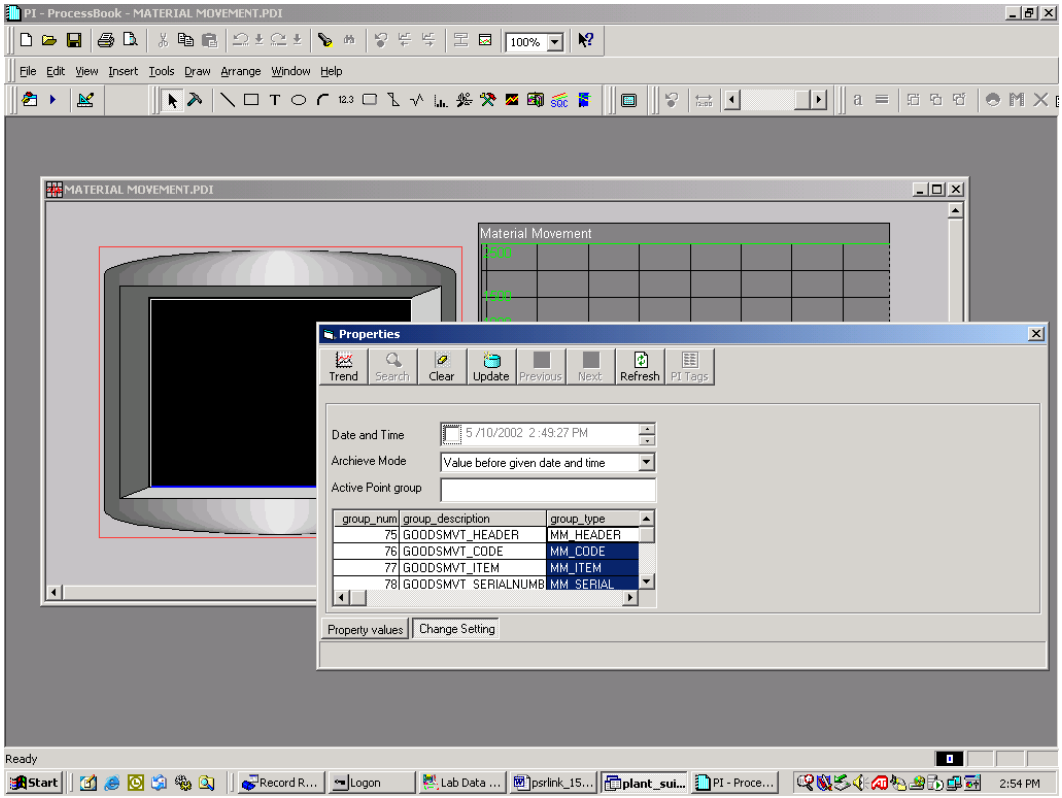
MM2 Item2 + 1 sec Tags in the point group ITEM and SERIAL if needed are filled out

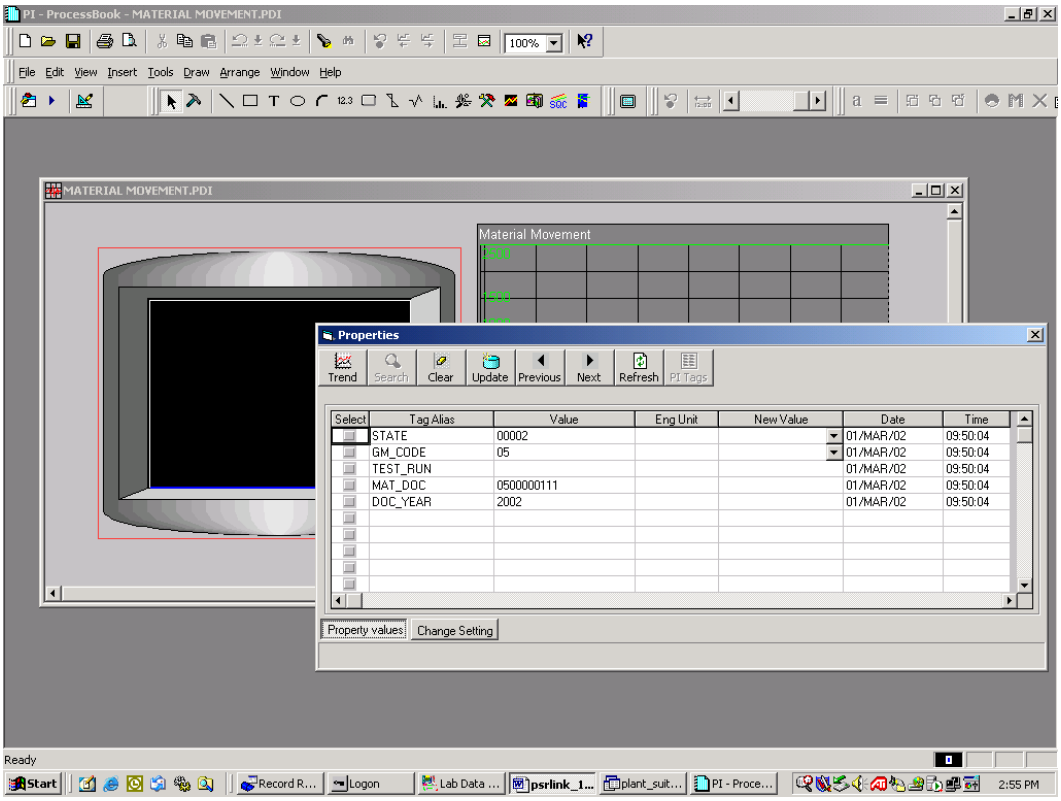
MM2 Item 3 + 1 sec Tags in the point group ITEM and SERIAL if needed are filled out

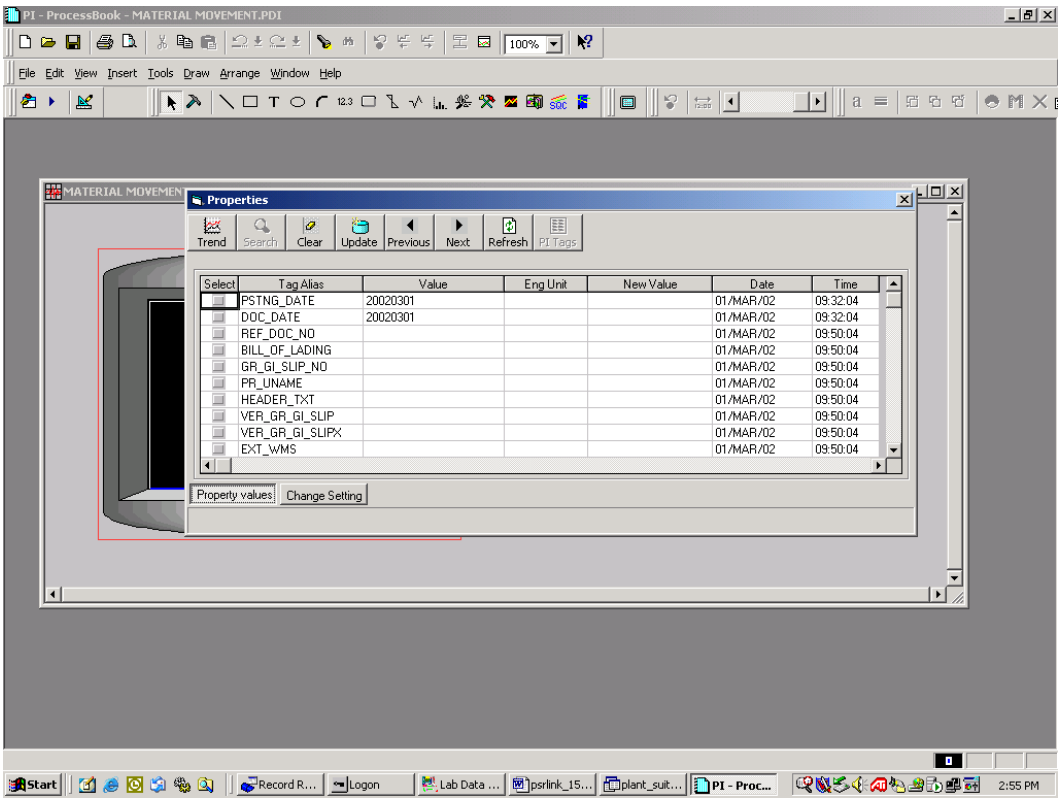
Movement 2 Endtime Tags in the point group CODE are filled out and the status is given the value 00002

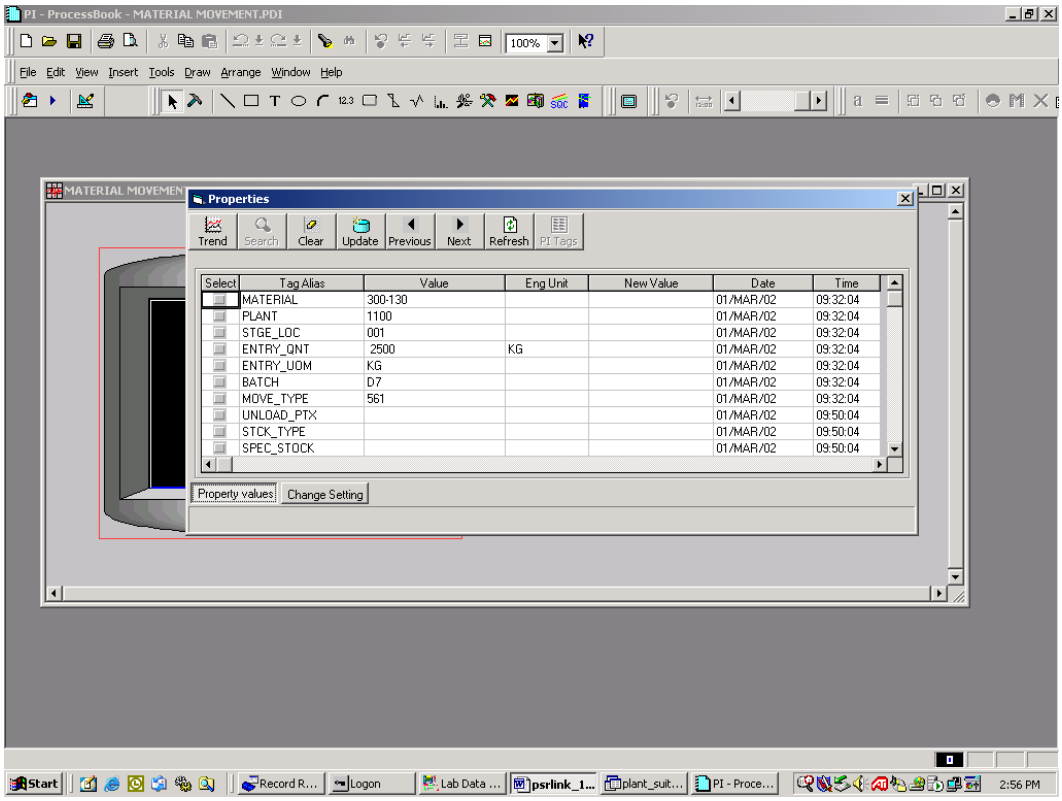
ProcessBook Display

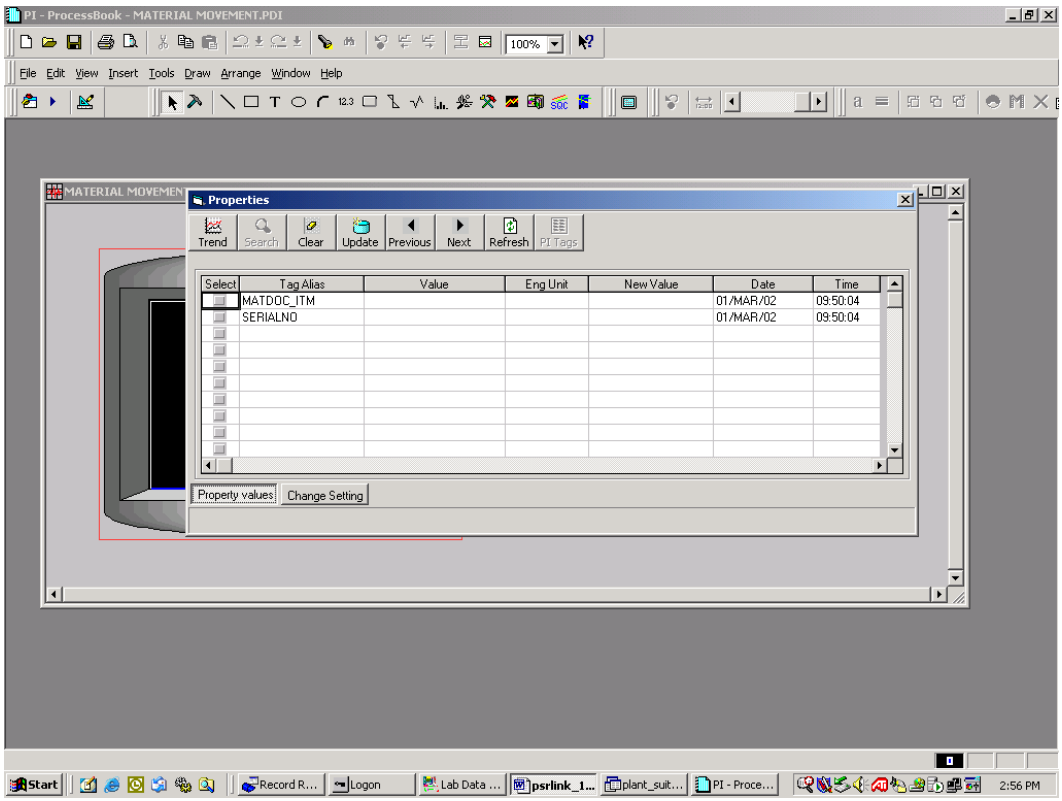
These displays show all point_group_members but the ones that have blank values would be deleted from the point_group_members table and thus only show the ones that have to be entered.

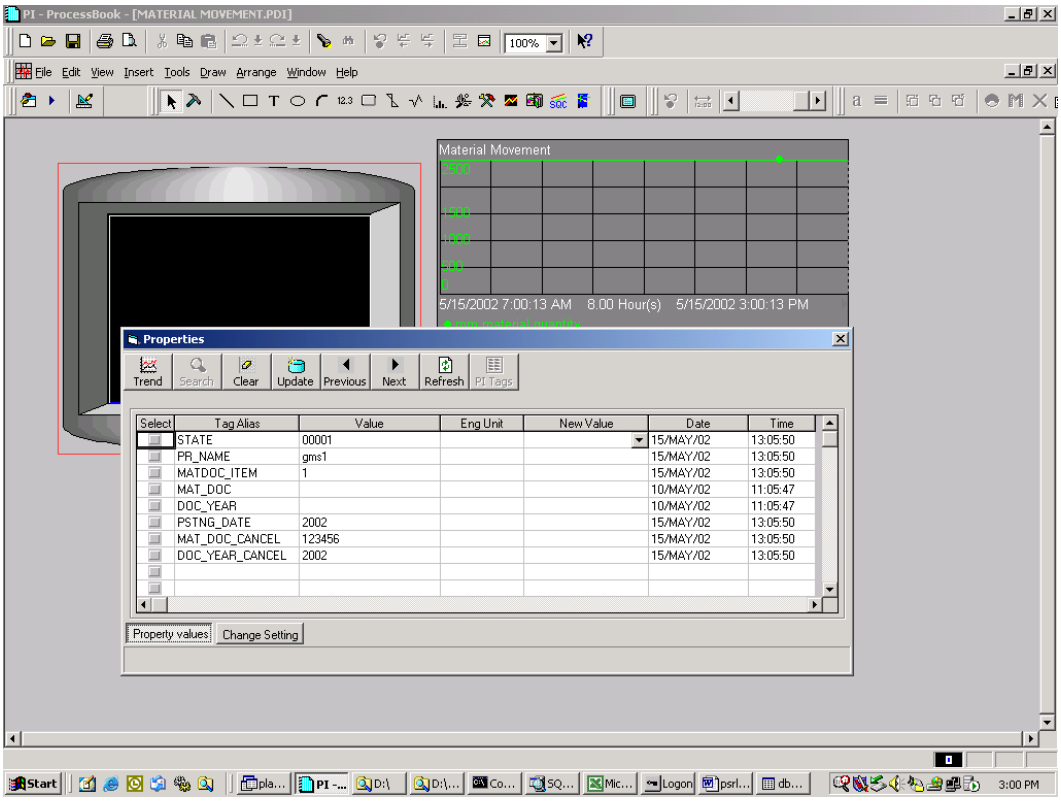



PI - ProcessBook - MATERIAL MOVEMENT.PDI
File Edit View Insert Tools Draw Arrange Window Help
MATERIAL MOVEMENT.PDI
Material Movement
Properties
Trend Search Clear Update Previous Next Refresh PI Tags
Select Tag Alias Value Eng Unit New Value Date Time
STATE 00002 01/MAR/02 09:50:04
GM_CODE 05 01/MAR/02 09:50:04
TEST_RUN 01/MAR/02 09:50:04
MAT_DOC 0500000111 01/MAR/02 09:50:04
DOC_YEAR 2002 01/MAR/02 09:50:04
Property values Change Setting
Ready
Start
2:55 PM

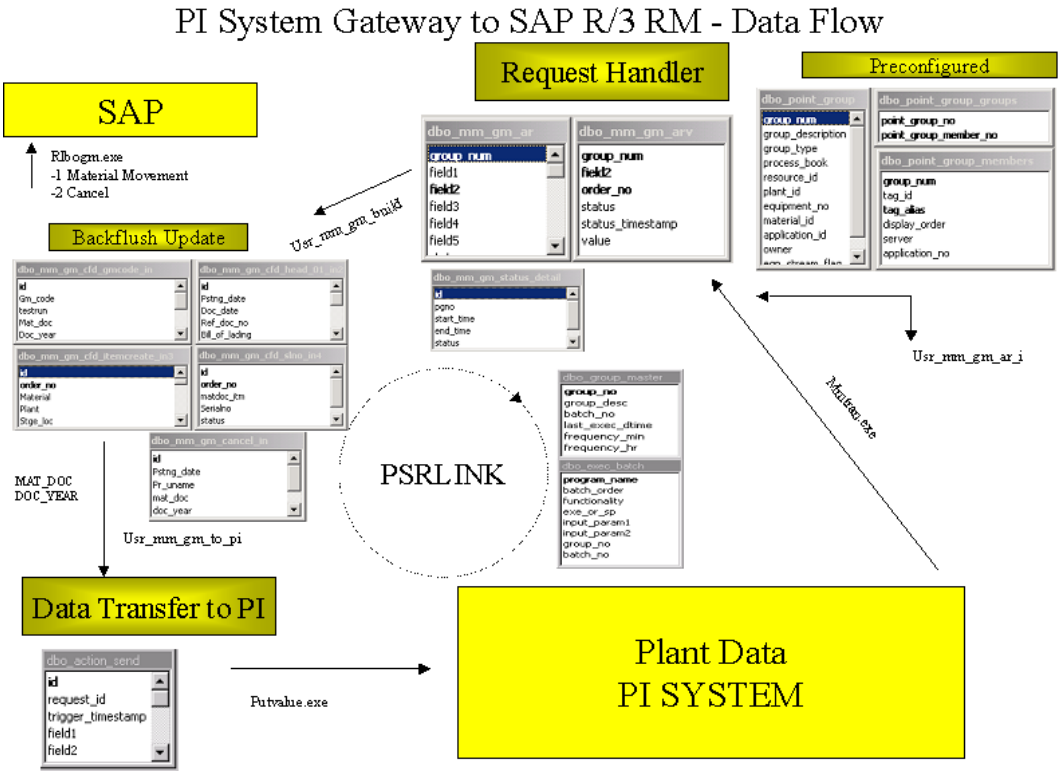

PI - ProcessBook - MATERIAL MOVEMENT.PDI
File Edit View Insert Tools Draw Arrange Window Help
Properties
Trend Search Clear Update Previous Next Refresh PI Tags
Select Tag Alias Value Eng Unit New Value Date Time
PSTNG_DATE 20020301 01/MAR/02 09:32:04
DOC_DATE 20020301 01/MAR/02 09:32:04
REF_DOC_NO 01/MAR/02 09:50:04
BILL_OF_LADING 01/MAR/02 09:50:04
GR_GI_SLIP_NO 01/MAR/02 09:50:04
PR_UNAME 01/MAR/02 09:50:04
HEADER_TXT 01/MAR/02 09:50:04
VER_GR_GI_SLIP 01/MAR/02 09:50:04
VER_GR_GI_SLIPX 01/MAR/02 09:50:04
EXT_WMS 01/MAR/02 09:50:04
Property values Change Setting
Start
2:55 PM


PI - ProcessBook - MATERIAL MOVEMENT.PDI
File Edit View Insert Tools Draw Arrange Window Help
MATERIAL MOVEMEN
Properties
Trend Search Clear Update Previous Next Refresh PI Tags
Select Tag Alias Value Eng Unit New Value Date Time
MATERIAL 300-130 01/MAR/02 09:32:04
PLANT 1100 01/MAR/02 09:32:04
STGE_LOC 001 01/MAR/02 09:32:04
ENTRY_QNT 2500 KG 01/MAR/02 09:32:04
ENTRY_UOM KG 01/MAR/02 09:32:04
BATCH D7 01/MAR/02 09:32:04
MOVE_TYPE 561 01/MAR/02 09:32:04
UNLOAD_PTX 01/MAR/02 09:50:04
STCK_TYPE 01/MAR/02 09:50:04
SPEC_STOCK 01/MAR/02 09:50:04
Property values Change Setting
Start
Record R...
Logon
Lab Data ...
psrlink_1...
plant_suit...
PI - Proce...
2:56 PM


PI - ProcessBook - MATERIAL MOVEMENT.PDI
File Edit View Insert Tools Draw Arrange Window Help
MATERIAL MOVEMEN
Properties
Trend Search Clear Update Previous Next Refresh PI Tags
Select Tag Alias Value Eng Unit New Value Date Time
MATDOC_ITM 01/MAR/02 09:50:04
SERIALNO 01/MAR/02 09:50:04
Property values Change Setting
Start
Record R...
Logon
Lab Data ...
psrlink_1...
plant_suit...
PI - Proce...
2:56 PM


PI - ProcessBook - [MATERIAL MOVEMENT.PDI]
File Edit View Insert Tools Draw Arrange Window Help
Material Movement
5/15/2002 7:00:13 AM 8.00 Hour(s) 5/15/2002 3:00:13 PM
Properties
Trend Search Clear Update Previous Next Refresh PI Tags
Select Tag Alias Value Eng Unit New Value Date Time
STATE 00001 15/MAY/02 13:05:50
PR_NAME gms1 15/MAY/02 13:05:50
MATDOC_ITEM 1 15/MAY/02 13:05:50
MAT_DOC 10/MAY/02 11:05:47
DOC_YEAR 10/MAY/02 11:05:47
PSTNG_DATE 2002 15/MAY/02 13:05:50
MAT_DOC_CANCEL 123456 15/MAY/02 13:05:50
DOC_YEAR_CANCEL 2002 15/MAY/02 13:05:50
Property values Change Setting
Start
3:00 PM

Data Flow


PI System Gateway to SAP R/3 RM - Data Flow
SAP
Rlbogm.exe
-1 Material Movement
-2 Cancel
Request Handler
Preconfigured
Backflush Update
Usr_mm_gm_build
Usr_mm_gm_ar_i
Matbar.exe
MAT_DOC
DOC_YEAR
Usr_mm_gm_to_pi
PSRLINK
Data Transfer to PI
Putvalue.exe
Plant Data
PI SYSTEM

Error Messages

Error messages that have been seen to be returned from SAP include the following:

Error	**Correction**
Posting only possible in periods … in company code … RLINK PPPI BUSINESS OBJECT Error No: 53	Need to correct the posting periods in SAP transaction mmpv

Chapter 19
Stored Procedure

The following detail information is provided for procedures that customers have used in their own applications.

Stopping a recipe

Usr_as_set_recipe “R”, plant_id, Recipe_id, ‘ ‘, date time, “00005”

Date time should be in the format YYYY-MM-DD hh:mm:ss

This procedure will put in action send what is required to stop the recipe and then it will call usr_as_set_phase to set the status of every phase that has a status of 0 or 1 or 4 to be the status 00003 for interruption. It will put the required values in action_send and then they will be sent to PI.

The same procedure can be used to terminate a recipe giving the status “00007”.

Changing status of Phase

The stored procedure for changing the status of the phase is as follows:

Usr_as_set_phase “P “, plant_id, Recipe_id, phase_id, date time, status

Date time should be in the format YYYY-MM-DD HH:mm:ss

The procedure will put in action send what is required to change the status of the phase.

Changing a Phase Resource

The stored procedure for changing the resource assigned by SAP in the recipe is usr_change_phase_resource. This procedure is called as follows

usr_change_phase_resource “recipe_no”, “operation_id”, “phase_id”, “new resource”

We have also provided a sample pplication called MRLINK_MODEL_APP that shows how to use this in conjunction with our standard database logon.

Changing the plan start date of recipe

The procedure usr_axr_sel was exposed to allow the user to incorporate this in their own front end for changing the data of a recipe. The format for calling is

usr_axr_sel ‘UPDATE’, recipe_id, “MM/DD/YYYY hh:mm:ss”

Stored Procedure	***Description***	***Input values***	Output values	***Filename***

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_action_executioner	select a record from action_send for processing			axexec
usr_action_results_I	Insert records into action results to satisfy new message requests in the message request and request part tables			axresi
usr_action_results_values_I	Inserts values into the action result_value table which are obtained from PI or other sources			axrvi
usr_action_send_set	Inserts records into action_send to be later sent to other destinations.			Asendset
usr_activity	Process the SAP/R3 activity instruction APHACT into message_request and request parts			prgexec2
Usr_activity_n (1-6)	Translation method for PI_PHCON PPPI_ACTIVITY_N			actn
Usr_activity_n_sec (1-6)	Translation method for PI_SRCON PPPI_ACTIVITY_N			actns
Usr_activity_n_unit (1-6)	Translation method for PPPI_ACTIVITY_UNIT for PI_PHCON			actnu
Usr_activity_finished_n (1-6)	Translation method for PPPI_ACTIVITY_FINISHED for PI_PHCON			actfinn
Usr_activity_n_unit_sec (1-6)	Translation method for PPPI_ACTIVITY_UNIT for PI_SRCON			actnus
Usr_activity_finish_n_sec (1-6)	Translation method for PPPI_ACTIVITY_FINISHED for PI_SRCON			actfinns
Usr_adhoc_helper	Used in SQL interface to do multiple calls			
Usr_ad_i	Inserts records into audit_data			Adi
Usr_app_all	Selects from Application table			Appall
Usr_app_obj_all	To insert, update, delete and select the record from applic_obj table			Apobjall
Usr_app_pgm_from_menu	Inserts a record into Applic_obj and point_group tables, if required. Populates point group members. Inserts record into symbol_menu table			Apfrmmnu
Usr_ar_rem_sel	Selects values from pp_ar_mem			aremsel
Usr_ar_sap_tran_i	Inserts a record in ar_sap_tran for the SAP_TRANS point group monitoring			arsti

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_ar_sap_tran_sel	Selects a member for the ar_sap_tran for SAP_TRANS monitoring			arstsel
Usr_ar_sap_tran_upd	Updates a record in ar_sap_tran for the SAP_TRANS monitoring			arstupd
Usr_ar_sap_tran_upd2				
Usr_ar_sel	Selects recors from Action_results for spin action “DELARV” deletes record from action result values “SELARV” selects record from action result values for the given request_part_id “CLOSERPID” updates action_results status “S” “RETRIEVE” selects record from action_results for the given request_part_id whose status is F			Arsel2
Usr_as_set_phase	Inserts record into action_send when the status is set in PISETBATCH.exe			Assetphs
Usr_arv_purge	Purges entries in arv_sap_trans for values for the SAP_TRAN			arvpurge
Usr_as_set_recipe	Inserts into action_send when we set the recipe_status using PISETBATCH.exe			Assetrcp
usr_atnam_correction	updates atnam of crfv when pppi_automatic_value is given instead of pppi_requested_value and deletes the record from crfv where atnam is pppi_input_request , this is a recipe checking routine			Atnamcor
Usr_axr_sel	ALL-selects all records from action_results for trigger_proc = control_monitor SPECIFIC- selects a record from Action_results for the given id UPDATE-updates action_results timestamps			Arsel
usr_batch_batchid	Gets the batch tag name from material tag when the phase has completed with start and end time of phase, used for the monitor mode			Batchid
Usr_batch_char	Procedure for recipe translation of the instruction ABTCL			procbatc

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_batch_char_batch	Translation method for the characteristic PPPI_BATCH in the instruction ABTCL			batchrbt
Usr_batch_char_monitor	Used for getting special batch characteristics when in monitor mode, this was added for customer and requires customized instructions. Uses common name table			Batmon
Usr_batch_char_value	Translation method for the characteristic value in PI_BT_CL			batchrva
Usr_batch_create	Procedure for the original instruction ABTCR in the recipe			procbtcr
Usr_batch_create_ar	Translation method for the PPPI_NEW_BATCH characteristic in ABTCR			batcrar
usr_batch_flow	Used to set up a tag in action results which will do a flow calculation. Used for BES and BPI plants			Batflow
Usr_batch_flow_plus	Used to set up a tag in action results which will do a flow calculation and then adds a second tag value			
usr_batch_flow_tag	Translation method for material which combines continuous and batch			batchtag
Usr_batch_flow_tag_kk	Translation method for quantity of material that combines recipe no search			
Usr_batch_flow_tag2	Translation method using material_tag for quantity- Polyone			Battag2
Usr_batch_flow_tag3	Translation method usnign point_group for material quantity- Polyone			Battag3
Usr_batch_flow_tag4	Partial solution-temporary Polyone			Battag4
usr_batchid_tag	Translation method for batch_id which combines continuous and batch			batchbat
Usr_batchid_tag2	Translation method for batch_id using material_tag- Polyone			Batidtg2
Usr_batchid_tag3	Translation method for batch_id using point_group- Polyone			Batidtg3
Usr_batchid_tag_kk	Translation method for batch_id that combines recipe and material			

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_batch_instr	For ZABTCCH, ZABTCDT, ZABTCNM, ZABTCTM instruction. Inserts records into message_request, request_part and request_part_values table. These instructions were added for a customer to return batch characteristics			Batinst
usr_build_action_results	selects records to insert into action results by selecting request parts and translation methods etc.			Axbuild
Usr_change_phase_resource	Changes phase resource			chphsres
Usr_char_list_all	Inserts record into characteristic_list table			Chrlstal
Usr_char_text_all	Inserts record into characteristic_text table			chrtxtal
Usr_check_stact_sequencerule	Used to set sequence of Activity and phase			
usr_check_time	checks when program has last run			Checktime
Usr_check_time2				
Usr_chg_code	Translation method on phase status for alias CHG_CODE			Chg
Usr_class_all	To insert, update, delete and select the record from class table			Clsall
Usr_clean_a_recipe	Cleans up a single recipe			cleanrcp
usr_clean_up ‘YES’	Used to clean out all results from processing the recipe. Used for demo mode use.			Cleanup
Usr_clean_rcp_from_basetable	Used to clean all information about a recipe including the crhe, crft, crfv and tline tables			clrcpbas
Usr_cn_all	ADD,UPDATE,DELETE, RETRIEVE flags do the respective action on common_name			cmnsel
Usr_cn_spin	Selects a record when spin operation on common_name			Cnspin
Usr_comp_pgm_from_menu	Populates records in point_group, point_group_members and point_group_groups			Cofrmmnu
Usr_comp_pgm_sel	Selects from point_group_members table			Copgmsel
usr_confirmation_short_text	Translation method for confirmation_short_text			short

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_confirmation_text_sec	Translation method for the confirmation text in the secondary resource ASRACT			shortsec
Usr_conf2_general	Used to retrieve general values from the table structures for conf2.exe			Conf2gen
usr_cons_mpo1	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmp01
usr_cons_mpo2	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmpo2
usr_cons_mpo3	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmp03
usr_cons_mpo4	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmp04
usr_cons_mpo5	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmpo5
usr_cons_mpo6	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmp06

Stored Procedure	*Description*	*Input values*	Output values	*Filename*
usr_cons_mpo7	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			consmp07
usr_cons_mpo8	This procedure is used to process the ACONS instruction. There are 8 permutations on the data given in an ACONS nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case.			Consmp08
Usr_cra_to_crp_I	if the destination is set to type 3 in SAP/R3 this will insert a record into the CRA_TO_CRP table that a recipe is available for download to be used by the process tcrps.exe			cr_atopi
usr_cra_to_crp_sel	Selects any recipe id which has been sent down to the destination when the interface is configured as a type 3. This is then used as input to pull down the recipe			Cratpsel
usr_crft_I	inserts records into crft on the original download of the SAP/R3 recipe			Crfti
usr_crfv_all	retrieves data from crfv			Crfvall
usr_crfv_I	inserts records into crfv on the original download of the SAP/R3 recipe			Crfvi
usr_crhe_all	retrieves data from crhe			Crheall
usr_crhe_I	inserts records into crhe on the original download of the SAP/R3 recipe			Crhei
usr_crhe_mtd	purges the records based on the recipe date in crhe			Crhemtd
Usr_crhe_mtd2	Purge for the soft purge that will leave the crhe, crft, crfv tables to run recipe again.			Crhemtd2
usr_crst	translates the original ACRST_I instructions into message requests and request parts			Proccrst
Usr_delete_modelplantdata				
usr_delivery_tag	Translation method for delivery tag			deliver
Usr_desc_for_hlpval_all	Helpvalues description			deshpall

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_detail_val_info_all	Detail information for help values			vlinfall
usr_display_all	selects records from display table			Dispyall
usr_display_u	updates the display table which is used to assist with the ODBC data query for a selected recipe			Displayu
Usr_dload_char_helpval_sel	Selection criteria for help values			selchrhlp
usr_doc_no_all				Docnoall
Usr_download_char_sel	Helpvalues download			dwnchrsl
usr_dummy_monitor	Translation procedure for PPPI_ACTIVITY when you only want to assign date and time			Phsdummy
Usr_eng_unit	Application to assign unit for activities			engunit
Usr_eq_del	Deletes records from symbol_menu, point_group, point_group_groups, point_group_members, equipment, applic_ob, Stream depending on passed symbol			Eqdel
Usr_eq_type_all				Eqtypall
Usr_eqalias_all	Selects from equipment_alias table			Eqpalias
Usr_eqp_all	Equipment groups			Eqpall
Usr_eqp_gp_all	Equipment group			Eqpgall
usr_eqp_sel	Selects a record from equipment table.			Eqpsel
usr_eqpalias_all	Used for inserting, updating and retrieval from equipment_alias table.			Eqpalias
Usr_eqpgpm_all	Equipment group members			Eqpgpmal
Usr_errlog_all	Selects from error_log table			Errlgall
usr_error_log_rfc_I	inserts an entry onto the error log table			Errlgrfi
usr_exec_batch_sel	selects records from exec batch to execute by psrlink			Exebats
usr_formula_sel (removed 1.4)	selects record from formula table			Forsel
Usr_formula_sel2	Modified form of usr_formula_sel			Forsel2
usr_general_msg_sel	selects from msg_mshd			Gmsgsel

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_general_msg_sel2	Modified form of usr_general_msg_sel			Gmsgsel2
usr_general_rtr	used to retrieve general values from the table structures			General
Usr_get_action_list (removed version 1.4)	inserts records into action results and action result values			Getaxon
Usr_get_action_send	Selects the items which are to be sent to another destination such as PI			Getsend
Usr_get_activity	Gets the tag based upon the SAP/R3 characteristic PPPI_STD_VALUE_PARAMETER_ID			Gtactivi
Usr_get_alias_tag	Selects the tag_id for value to be sent to PI including the tag for the min and max value. Uses SAP/R3 PPPI_DATA_POINT_NAME			Gtalstag
Usr_get_alias_tag_range	Gets the tag and based upon the SAP/R3 characteristic PPPI_DATA_POINT_NAME and selects the range as start and end time of recipe and sets no of values to 10			Gtaltagr
Usr_get_batch_char	Used to get the batch id and the end time for special batch characteristics. This was added for a customer and requires customized instructions. Uses common name table			Gtbach
Usr_get_batch_tag				Gtbattag
Usr_get_batch_tag_name	Gets the batch_id tag from material with the endtime. Used for continusous process			Getbatagl
Usr_get_flow_tag	Used to set up a tag in action results which will do a flow calculation. Used for continuous plants			Gtflowta
Usr_get_inspect	Gets the tagname for the quality inspection point based on the characteristic PPPI_INSPECTION_RESULT			Gtlinsp
Usr_get_location	Set the equipment location for a given material			Gtloc
Usr_get_material_tag				gtmtlta1
usr_get_operation_status	Gets the operation_id start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE			Gtoprsts

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_get_phase_status	Gets the phase_id, start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE			Gtphasts
Usr_get_recipe_status	Gets the recipe start time and endtime by adding shift duration to the OSI_START_TIME and OSI_START_DATE			Gtrcpsts
usr_gi				procgi
usr_gi_all				giall
usr_gp_master_I	Used to insert and update group_master table.			Gpmasi
usr_gp_master_sel	Used to select records from group_master table.			Gpmassel
usr_group_master_u	Updates the group master table for the last time the program group has been executed			Gpmasu
Usr_group_master_u2	Updates the group master table in queue 2 for the last time the program group was executed			
usr_helper				Hlpproc
Usr_helpvalues_all	Used to get helpvalues for RFC get help values			hlpvlall
usr_ir_all				irall
Usr_isspace_enough	In PMU.exe , checks whether is there enough space is available in the DB before inserting a recipe			Usrspace
usr_ledger_history_i	Inserts data into ledger_history			Ledhisi
usr_ledger_history_r	Removes data from ledger_history			Ledhisr
usr_ledger_to_action	Sends data in ledger history to action_send			Ledsend
Usr_line_ar_upd	Bayer for line update			Lineupd
Usr_line_selection	Bayer line selection			linesel
Usr_line_selection_2	Bayer line selection			Linesel2
Usr_line_selection_3	Bayer line selection			Linesel3
Usr_lo_mat_doa_out_i	Inserts results from the material get detail			
Usr_lo_mat_dobew_out_i	Output of the material get detail			
Usr_lo_mat_doc_out_i	Output of the material get detail			

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_lo_mat_general	Selection of data for the material get list query, also check filter			
Usr_lo_mat_getlist_1st_out_i	Output of getlist			
Usr_lo_mat_status_u	Status update on material			
usr_load_all	inserts data into recipe, phase, operation, formula, material_list from the original SAP/R3 download table			Loadall
usr_loc_sel	Used to select a record from location table.			Locsel
usr_make_msg	inserts records into mshd, msel and up_tlines for a text message			Makemsg
usr_material_from_batch	Translation method for obtaining the material quantity after the batch is identified			Mttagbch
usr_mat_list_sel	Selects the materials for a given recipe from material_list			matsel
Usr_mat_list_sel2	Modified form of usr_mat_list_sel			matsel2
Usr_mat_tag_all	ADD, MODIFY, DELETE, RETRIEVE flags to do the respective operation on material tag			Matagall
usr_material_all				Mata
Usr_material_duplicate_cons	Bayer scenario 1 enhancement for material on multiple resources			
usr_material_group_all	Add, delete, modify, select, spin button selection are handled in this procedure for material_group table			Matagpa
usr_material_group_i	Inserting a record into material_group_members table			Matgpi
usr_material_group_mem_all	Add, delete, modify, select are handled in this procedure for material_group_members table			Matgpma
usr_material_group_sel	Selects record from material_group for psrgui application			Matgpsel
Usr_msic1_tag	Misc tag1 on material tag translation method			Misc1
Usr_msic2_tag	Misc tag2 on material tag translation method			Misc2
Usr_msic3_tag	Misc tag3 on material tag translation method			Misc3
Usr_msic4_tag	Misc tag4 on material tag translation method			Misc4

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_msic5_tag	Misc tag5 on material tag translation method			Misc5
Usr_mm_gm_ar_i	Goods movement action results insert mm_gm_ar			
Usr_mm_gm_ar_upd	Goods movement action results updates			
Usr_mm_gm_ar_upd2	Goods movement action results updates into mm_gm_arv table			
Usr_mm_gm_build	Scans mm_gm_arv table and prepares data in the input tables to create goods movement document			
Usr_mm_gm_build2	Scans mm_gm_arv table and prepares data in the input tables to cancel goods movement document			
Usr_mm_gm_general	For selecting data from plant_suite for goods movment			
Usr_mm_gm_sel	Mmtran application uses this procedure to select records from mm_gm_ar table			
Usr_mm_gm_status_u	Updates the status of the processed records for goods movement			
Usr_mm_gm_to_pi	Populates action_send table with material document and year data with goods movement start and end times to set the values to PI			
Usr_mm_gm_to_pi2	Populates action_send table with material document and yar data for cancelled goods movement document to set the values in PI			
Usr_mesl_u	Updates the msel table with the status of the characteristic as returned from SAP/R3			Mselu
usr_msel_i	inserts replies to specific characteristics into the SAP/R3 table from reply			Mseli
usr_msel_sel	selects characteristics from the msel table for the mshd which is requested			Mselsel
Usr_msel_sel2	Selects characteristics for msel used for transactional pmucl			Mselsel2
Usr_msel_u2	Updates MSEL, MSHD depending on the flag status			Mselu2
Usr_msg_correct	Selects records from mshd, msel error message according to flags M,L, E			Msgcrt

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_msg_hdr	takes the results in request part and formulates the values in the format SAP/R3 requires in MSHD and MSEL			Procmhdr
usr_msg_hdr22	Handles all message categories in instruction_requirements table. At the end calls another stored proc called "usr_msg_hdr23"			Msghdr22
usr_msg_hdr23	Called within the Usr_msg_hdr22 procedure. This stored proc handles all the categories of partial_result_instructions table.			Msghdr23
Usr_msg_hdr_24	Handles messages that gets populated in the tables ar_sap_tran and arv_sap_tran.			Msghdr24
usr_msg_sel	Selects data from mshd SAP/R3 messages			Msgsel
Usr_msg_sel2	Modified form of usr_msg_sel			Msgsel2
usr_msg_tlines_i	inserts records in tlines for general messages which are sent down by SAP/R3			Msgtline
Usr_msgid_new_u	Used by BAPI for message update with new message_id from SAP			msidnewu
usr_mshd_i	inserts records for the message header to replied to message requests			Mshdi
usr_mshd_sel	selects the header record from mshd for the selected message id			Mshdsel
Usr_mshd_sel2				Msgsel2
Usr_mshd_status_reset	Resets the status in MSHD for records that have not had a reply back from the BAPI message upload			
usr_mshd_sts_upd	updates the status of the MSHD when a reply is sent to SAP/R3			Mshdstsu
Usr_mshd_u	updates the mshd record with the result from SAP/R3 after the message is sent to SAP/R3			Mshdu
Usr_mt_spin	Selects a record from material_tag when the user selects spin button			Mtspin
usr_obatch_recipe	formulate the input to PID Openbatch, delivered only with this addition			Obatch
usr_open_batch_phase	Reads results from the Batchhis table constructed from Openbatch to set the start and end times for a phase, delivered only with this addition			Phasepid

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_open_batch_recipe	Read results from the Batchhis table constructed from Openbatch to set the start and end times for the recipe, delivered only with this addition			Recipid
usr_operation_display				Operation_display
usr_operation_monitor	Monitors the completion of the phases in an operation and put on action results when all phases has been completed			Oprmonit
Usr_operation_monitor_new	Monitors completion of operation but also sets tags for campaign manager on operation			oprmonitnew
usr_operation_monitor_user	Translation method for the user status on operation			oprmonur
usr_operation_sel	Selects records from operation			oprmonit
Usr_operation_sel2	Modified form of Usr_operation_sel			oprsel2
usr_operation_status	inserts records into nstru_result_values and update action_results and inserts into action_send. This is used for BES and BPI plants			Oprsel
Usr_operation_status_new	Used for campaign manager to get additional functionality			opstatusnew
usr_opst	translates the original AOPST_I instructions into message requests and request parts			Procopst
usr_opust	Translates the AOPUST_I instruction into message request and request parts			procopus
usr_osi_group_desc				Osigd
usr_pb_general				pbgen
Usr_pb_menu_all	Selects from pb_menu table			Pball
usr_pb_menu_all2	Modified form of Usr_pb_menu_all			pbmenua2
Usr_pg_all	Selects from plant_group table			Pltgall
usr_pg_insert	Used to insert, update, delete a record in point_group table.			Pgins
usr_pg_select	Used to select a record from point_group table or point_group_members table depending on the parameter passed.			Pgpgmsel
Usr_pgg_all	Selects from point_group_groups table			Pggall

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_pgm_copy	Used to copy point_group_members of one point_group to another.			Pgmcopy
usr_pgm_from_menu	This procedure is called from Process book menu. According to the menu choosen, it inserts data into equipment, point_group, point_group_members and symbol_menu tables.			Frommenu
usr_pgm_insert	Used to insert record into point_group_members table.			Pgmins
usr_pgm_sel	Used to select records from point_group_members table for the given symbol_type.			Pgmsel
usr_pgm_update	Used to update a record in point_group_members table.			Pgmupdat
usr_phact_activity	Translation method for PPPI_ACTIVITY			phactact
Usr_phact_confirmation_text	Translation method for the confirmation text in the PI_PHACT			phacttxt
Usr_phact_status_monitor	Translation method for the status in the PI_PHACT			phactmon
usr_phar	translates the original APHPAR_1 instructions into formula table			Procphar
Usr_phase	To insert a record pi_process_book table			Phase1
usr_phase_alias_monitor	Sets up action_results for the phase. Monitors the status of the phases for a recipe. It start to monitor the phase after the recipe has been started. The monitor is based on the phase alias name and this is used for BES plants			Phsamoni
usr_phase_all	Selects data from phase			Phssel
usr_phase_eqp_monitor	Translates phase status when resource is to be included in the selection			Phseqpmo
usr_phase_eqp_monitor_user	Translates phase user status when resource is to be included in the selection			phuseeqpm
usr_phase_monitor	Sets up action-results for the phse monitor. Monitors the status of the phases for a recipe. It starts to monitor the phase after the recipe has been detected as started. Used in BPI plants			phsmonit
usr_phase_monitor_user	Translates phase user status			phusmon

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_phase_resource	Translation method for PPPI_PHASE_RESOURCE			Phaseres
usr_phase_sel	Select data from phase			Phactmon
Usr_phase_sel_2	Modified form of Usr_phase_sel			phssel2
Usr_phcon	Procedure for translation of orginal recipe message APHCON			procphco
Usr_phcon_time_status	Translation method for the status in the APHCON instructions			phcomtm
usr_phst	translates the original APHST_I instructions into message requests and request parts			Procphst
usr_phust	Translation for instruction APHUST_I			procphust
Usr_pi_error_log_sel	Selects a record from error_log			Logsel
Usr_pi_function_list	Gets list of pi functions in the pimod.exe			pifnsel
usr_plant_all	Selects records from the plant table			Plantall
Usr_plant_all_cm	This is used only in campaign manager applications. It selects record from plant table based on type etc.,			plantallcm
usr_plant_loc_I	On creation of new plant id, it inserts records in to plant, location table as well as inserts data into translator, instruction_characteristic tables depending on model plant_id choosen by the user. In case of deletion of plant id, it takes care of deleting the records from the above mensioned tables.			Pltloci
Usr_plant_sel	PLANT and RETRIEVE flags for spin and retrieval of records from the plant table			Pltsel
Usr_pltgpm_all	Plant group members			Pltgpmal
Usr_pmu_check	Checks to see is there any record to be sent to SAP. Added the instructions ZI_CONS and ZI_PROD to the instructions that will be checked for 0 quantity.			Pmucheck
Usr_pmu_tid_all	TID management for process message upload			pmutidall
usr_point_group_all				Ptgpa
usr_point_group_i	Inserts a point group			Ptgrpi
usr_point_group_mem_all				Ptgpma
usr_point_group_r				Ptgpr

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_post_date	Translation method for post date in PI_PHCON			Postdt
Usr_post_date_sec	Translation method for the post date in the PI_SRCON			postdts
Usr_pp_dt_convert2	Date Time manipulation			Dtconv2
Usr_pp_general	Used for all the BAPI calls			rcgen
Usr_pp_pc_char_data_out_i	Used for BAPI version of character detail to insert data			chdtaoi
Usr_pp_pc_char_text_out_i	Used for BAPI version of character detail to insert text			pctxtoi
Usr_pp_pc_dforhelpvalues_out_i	Used for BAPI version of helpvalues to insert helpvalues??			Dfhlpvoi
Usr_pp_pc_fixvalues_out_i	Used for BAPI version of helpvalues to insert into fixValues			fixvaloi
Usr_pp_pc_helpvalues_out_i	Used for BAPI version of helpvalues to insert into helpvalues			hlpvaloi
Usr_pc_valuesforfield_out_i	Used for BAPI version of helpvalues to insert into values for field			vffldoi
Usr_pp_rc_cntlrecheader_out_i	Used for BAPI version of contrl recipe for the getlist of recipes			rchdroi
Usr_pp_rm_ar_rem_i	Inserts new point groups into pp_ar_rem and pp_arv_rem			arremi
Usr_pp_rm_bapi_general	Selects data for BAPI call			rmstsu
Usr_pp_rm_arem_2_r3	Procedure that reads from action_results and inserts into SAP like tables			Ar2r3
Usr_pp_rm_ar_rem_upd	Updates results from PI into pp_ar_rem			arremupd
Usr_pp_rm_putvalue_2_pi	Confirmation number gets written back to PI by sending to action_send			putvalue
Usr_pp_rm_status_u	Updates return status and results from BAPI call			rmstsu
Usr_pp_status_u	Used for BAPI status updates for all BAPI's			rcstsu
usr_pre_process_check	checks the recipe which comes down from SAP/R3 against the required characteristics in the Instruction_characteristics table			Phaseall
usr_prev_next				prevnxt
Usr_prn_all	Selects from plant_resource_network table			Prnall

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_prod_mpo1	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmpo1
usr_prod_mpo2	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmpo2
usr_prod_mpo3	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmp03
usr_prod_mpo4	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmp04
usr_prod_mpo5	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmp05
usr_prod_mpo6	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmpo06

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_prod_mpo7	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmp07
usr_prod_mpo8	This procedure is used to process the APROD instruction. There are 8 permutations on the data given in an APROD nstruction depending on wheter material, operation or phase are given. Thus there are 8 procedures to handle each case			Prodmp08
usr_program_executioner	selects records from action results, called by all PI routines etc to see if there is any data from them to process			Preprock
usr_pt_gp_all	Point_group information			ptgpall
usr_purge	selects purge procedures from the purge table for execution			Purgutil
Usr_purge_all	Selects from purge table			Purgeall
usr_qm_general	Profile application information			usr_general
Usr_qm_insrt_login_info	Profile application information			loginfo_insert
usr_qmsmr1	sets up the action result for QM instructions in continuous plants			Prqmsmr1
usr_qmsmr1_monitor	sets up the action_result request for QM instructions when the phase the request is in has completed. Used in BPI and BES plants			Qmsmonit
usr_qmsmr1_monitor_desc	Translation method for qm instruction the description characteristic			qmonides
usr_qmsmr1_monitor_dev	Translation method for qm instruction the deviation characteristic			qmonidev
usr_qmsmr1_monitor_no	Translation method for qm instruction the number of samples characteristic			qmonino
usr_qmsmr1_monitor_s1_v1	Translation method for qm instruction the value characteristic			Qmonis1
usr_read_and_process	Reads the original tables from SAP/R3 and translates to the SP88 model and message requestes			Readandp

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_read1_monitor	sets up the action_result request for Read1 instructions when the phase the request is in has completed. Used in BPI and BES plants			read1mon
usr_read1_read2	translates the original AREAD1 and AREAD2 instructions into message requests and request parts			Procread
usr_read2_monitor	sets up the action_result request for Read2 instructions when the phase the request is in has completed. Used in BPI and BES plants			read2mon
usr_reason_for_variance	Translation method for the reason for variance			Reason
Usr_reason_phcon	Translation method for REASON on phcon instruction			Reason_phcon
Usr_rebuild_basetable_index	Rebuilds base table index			rebldidx
usr_recipe_all	Retrieves data from the recipe table			Rcpall
Usr_recipe_all_cm	Campaign manager			rcpallcm
usr_recipe_display				Recipe_display
usr_recipe_monitor	Sets up Action _Results for the control recipe monitor. Monitors the status of the recipe by checking for a status change from the time it is first read from SAP/R3. Used in BPI and BES plants			Rcpmonit
Usr_recipe_sel2	Modified form of Usr_recipe_sel			rcpsel2
usr_recipe_upd (removed version 1.4)	updates the recipe record for status changes and timestamps			Rcpupd
usr_reply_translator	Translates the results in action_result_values back into the request part values for the original SAP/R3 request.			Reptrans
usr_reservation	Translation method for the reservation			reservat
usr_reservation_item	Translation method for the reservation item			resesrv
Usr_reset_alarm	Resets alarm tags			resetalr
usr_rs_and_rsi	Translation method for getting the reservation and reservation_item from batch			rsandrsi
usr_rs_and_rsi_app	Application for getting the reservation and reservation_item from the batch			rsrsiapp

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_rs_and_rsi_app_nobatch	Appliction for getting the reservation and reservation_item from material list without a batch			Rsrsiapp_withoutbatch
Usr_sap_msg_alias_all	Populates data in instruction_characteristic and translator table			Smaall
Usr_sap_tran_master_all	General sap transactions			saptidal
Usr_scrap	Translation method for the PPPI_SCRAP in the PI_PHCON			scrap
Usr_sec_activity	Procedure for ASRACT			procsrac
Usr_sel_for_helpvalues_sel	Helpvalues slections			sfhlpsel
Usr_server_status	Procedure for changing the status of server usr_server_status ‘U’, ‘servername’, ‘Y or N’			
Usr_set_alarm	Sets alarm tags			Setalarm
usr_set_location	Inserts a record into action_results_values table and updates the status in action_results to ‘S’			Arvsetlo
usr_set_status	sets the status of the recipe, operation and phase for a continuous plant			Arvsetst
Usr_set_status_partial	Partial status application for phase in continuous reciep			arvsetsp
Usr_show_axres_records	Lists records that have not been completed in action_results but that the recipe has completed the results are put in table axres_records			
usr_sp_all	Used to retrieve and pdate the system_parameter table.			Spall
Usr_sract_activity	Translation method for the activity in PI_SRACT			sractact
Usr_sract_confirmation_text	Translation method for the confirmation text in PI_SRACT			sracttxt
Usr_sract_monitor	Translation method for the status in PI_SRACT			sractmon
Usr_srcon	Procedure for the instruction ASRCON in the original recipe			procsrco
Usr_srst	Procedure for the instruction ASRST in the original recipe			procsrst
Usr_srst_confirmation_text	Translation method for the confirmation text in PI_SRST			srshort

Stored Procedure	***Description***	***Input values***	Output values	***Filename***
Usr_srst_monitor	Translation method for secondary resource status			srstmoni
Usr_srst_reason_for_variance	Translation method for reason for variance in PI_SRST			srvarian
Usr_stream_all	To insert, update, delete and select the record from stream table			Streamal
usr_stream_menu	This procedure takes care of inserting record into stream, symbol_menu, point_group, point_group_members table for the given stream.			Streamnu
Usr_storage	Translation method for getting storage location from material list			Stor_tm
Usr_storage_matlist_app	Application for getting storage location from material list			Stor_material
usr_sub_all	Handles spin button for subscriber tab in configuration application			suball
usr_symbol_menu_sel	Selects a record from symbol_menu table.			Symbmen
usr_time_status_sec	Translation method for the status of the PI-SRCON			srcontm
usr_tlines_all	retrieves data from tlines			Tlinesall
usr_tlines_I	inserts records into tlines. Used when processing the original download of SAP/R3 recipes			Tlinesi
usr_trans_data_sel	This general procedure selects record from instruction_characteristic, instruction_category, translation_methods, and application table depending on the parameter passed.			Transdat
usr_trans_I	Used to insert record into translator table.			Transi
Usr_trans_sel	Selects from translator table			Transel
Usr_u_all	Retrieves all units			Uall
Usr_ug_all	Unit groups			Ugall
Usr_unchg_code	Translation method for phase to get alias UNCHG_CODE			unchg
usr_unit_sel	Selects a record from unit table for the given plant_id.			Unitsel
Usr_unplanned_material	Not complete, for unplanned materials			unplnmat

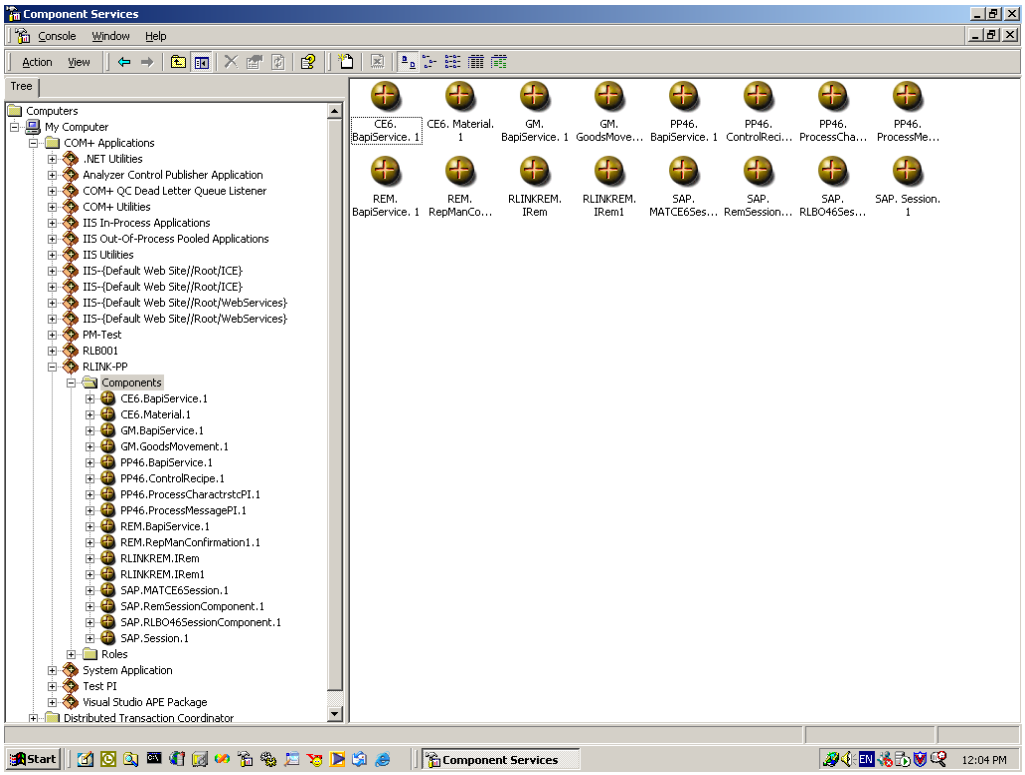
Stored Procedure	***Description***	***Input values***	Output values	***Filename***
usr_up_tlines_all	retrieves data from up_tlines			Uptlnall
usr_upd_phs_res	Updates the phase resource in phase table, action_results			updphres
usr_update_action_list_3	Inserts record into action_results_values and updates the status of action_results table to 'P' or 'S' depending on various conditions.			Updaxon3
usr_update_action_list2	Inserts record into action_results_values and updates the status of action_results table to 'P' or 'S' depending on the various phase statuses.			Updaxpn2
usr_update_complete_flag	The categories in the table instruction_requirements, may have multiple values. This procedure checks whether a category has got all values. If "YES", then it sets the complete_flag column of request_part_values table to 'C'.			Cmpflag
usr_update_statistics	Updates the statistics for tables used in the application			udstatic
usr_upd_phase_ar	Updates the phase and action results for the change in phase status			updphsar
Usr_upd_phase_ar2(removed version 1.4)	Modified form of Usr_upd_phase_ar			updphsa2
usr_upd_rcp_ar	Updates recipe table and action results for the status and timestamp information			updrcpar
Usr_upd_rcp_ar2	Modified form of Usr_upd_rcp_ar			updrcpa2
Usr_upd_srst_resource_ar	Updates secondary resource			updsecar
Usr_update_action_list	Updates action results and inserts a record into action result values, used in the status setting routines			updaxon
usr_update_action_send	Updates the action send record after it has successfully been archived to PI			upacsend
Usr_update_axres_records	Updates records in action_results that have been found not to be completed and the recipe has finished, use usr_show_axres_records to determain what records are going to be updated			
Usr_update_olddate2iso	Updates data format in action_results and action_send to iso format			

Stored Procedure	*Description*	*Input values*	Output values	*Filename*
Usr_update_server	Updates server name in tables common_name, ledger_history, point_group_members, material_tag			Upserver
usr_uptlines_sel	Selects text lines being sent to SAP/R3			Uplines
Usr_uptlines_sel2				Uplines2
Usr_values_for_field_all	Used in the Helpvalues RFC			vlfldall
Usr_ver_upd	Updates version table			
Usr_write_data_pi	Writes msg_mshd data for materials toPI			Write_to_pi
usr_yield_to_confirm	Translation method for PPPI_YIELD_TO_CONFIRM			yield
Usr_yield_to_confirm_partial	Used for continuous recipe for partial confirmations, translation method			yieldp

Components

Components	**Description**
PP46.BapiService.1	PP46 Business Object Proxy BapiService
PP46.ControlRecipe.1	PP46 Business Object Proxy ControlRecipe
PP46.ProcessCharactrstcPI.1	PP46 Business Object Proxy ProcessCharactrstcPI
PP46.ProcessMessagePI.1	PP46 Business Object Proxy ProcessMessagePI
SAP.RLBO46SessionComponent.1	SAP DCOM Connector Session Object RLBO46SessionComponent
CE6.BapiService	
CE6.Material	
GM.BapiService	
GM.GoodsMovement	
REM.BapiService	
REM.RepManConfirmation	
RLINKRem.IRem	
RLINKRem.IRem1	

SAP.MATCE6Session	
SAP.RemSessionComponent	
SAP.Session	



Index

Revision Record

Date	***Person***	***Action***
1010/96	GMS	Initial draft
1/20/97	GMS	Version 1.0
5/23/97	GMS	Draft of version 1.2
6/6/97	GMS	Version 1.2
7/25/97	GMS	Version 1.3
11/5/97	GMS	Version 1.31
1/27/98	GMS	Version 1.33
7/15/98	GMS	Version 1.34
9/5/98	GMS	Version 1.34 build 3
11/05/98	GMS	Version 1.34 build 5
12/31/98	GMS	Version 1.34 build 5 patch
1/28/99	GMS	Version 1.34 build 6
6/07/99	GMS	Version 1.34 build 7
3/26/00	GMS	Version 1.35 draft
4/12/00	GMS	Version 1.35
2/10/01	GMS	Version 1.4
3/26/01	GMS	Version 1.4 build 2
1/27/02	GMS	Version 1.5
7/20/04	Gms	Version 1.6