RESOURCE25

Data Interface Design and Implementation

For users of RESOURCE25 and SCHEDULE25

Index	Contents	Print	InfoFinder
Go to the Index, where you can click on a page number to jump to that page.	Go to the Contents, where you can click on a page number to jump to that page.	Open the Print dialog to print a page or pages.	Return to the InfoFinder page.

UNIVERSAL ALGORITHMS

Document History	Edition	Version	Date
	First Edition	1.6a	April 1998

Software Version

RESOURCE25 version 1.6a

Trademarks

Copyright

Table of Contents

4 - Building Your Bulk Interface

5 - Using Your Bulk Interface

Transaction-based Event Data Interface

6 - The Transaction-based Event Data Interface

7 - Building Your Transaction-based Interface

8 - Using Your Transaction-based Interface

Appendices and Glossary

Appendix A - vCalendar Properties

Appendix B - Publishing Events

About This Manual

Introduction

Purpose of manual

The purpose of this manual is to tell you how to create and use interfaces that allow you to share event data among RESOURCE25, SCHEDULE25, your student information system (SIS), and other systems.

It provides guidance for creating and using:

- A bulk data interface to move SIS class event data to and from RESOURCE25 and SCHEDULE25 once per term/semester.
- A transaction-based data interface to keep event data in RESOURCE25 and your SIS in sync on an ongoing basis.

Who should read this manual

This manual should be read by the technical personnel on your RESOURCE25 implementation team responsible for building system interfaces to SCHEDULE25 and your SIS.

What's in this manual

Manual contents This table summarizes the contents of this manual.

This section...	And this chapter...	Explain..
Overview	1 - Introduction to Event Data Flows	• The bulk data flow options • The transaction-based data flow
	2 - Event Data Format Options	• The class descriptor data format • The vCalendar data format
Bulk Event Data Interface	3 - Bulk Interface Options	• The four bulk interface options and their implementation requirements
	4 - Building Your Bulk Interface	• How to set up each of the bulk interface options
	5 - Using Your Bulk Interface	• How to use each of the bulk interface options
Transaction-based Event Data Interface	6 - The Transaction-based Interface	• The transaction-based interface and its implementation requirements
	7 - Building your transaction-based interface	• How to set up the transaction-based interface
	8 - Using Your Transaction-based Interface	• How to use your transaction-based interface
Appendices and Glossary	Appendix A - vCalendar Properties	• The components (properties) of vCalendar data objects
	Appendix B - Publishing Events	• The implicit and explicit publishing of event data
	Glossary	• Terms used in this manual

Instructional command conventions

This manual uses the following usage conventions for instructional commands:

This command...	Tells you how to...	Example
Choose <menu>|<menu item>	Use menus to navigate to a form	Choose Administration|System Definitions. *Meaning:* Position the cursor on the "Administration" menu title, and click the left mouse button to open the Administration menu. Then position the cursor on the "System Definitions" menu item and click the left mouse button to open the System Definitions and Defaults form.
Click <button name>	Use a function controlled by a form button	Click OK. *Meaning:* Position the cursor on the button labeled "OK" and click the left mouse button.
Double-click <item>	Open a record or form	Double-click the vCalendar icon. *Meaning*: Position the cursor on the icon labeled "vCalendar" and click the left mouse button twice rapidly.
Select <item>	Highlight an item on a form so you can subsequently act on that item	Select the Weekdays tab. *Meaning*: Position the cursor on the tab labeled "Weekdays" and click the left mouse button.

Other RESOURCE25 documentation

What to look for in print

In addition to this manual, the following RESOURCE25 information is available in print:

RESOURCE25 Installation and Database Preparation. A technical guide to installing and maintaining RESOURCE25 and the RESOURCE25 database.

RESOURCE25 System Administration. A guide to preparing RESOURCE25 for users, including setting system preferences, adding and maintaining users, and using RESOURCE25 security functionality to appropriately assign system access.

RESOURCE25 - Anatomy of an Event. Information about what RESOURCE25 event records are, how they're structured, and the information they contain.

RESOURCE25 Data Preparation 1: Space and Customers. A guide to preparing the space and customer data RESOURCE25 uses in event scheduling.

RESOURCE25 Data Preparation 2: Events. A guide to preparing the event data that must be in place before you can begin using RESOURCE25 for daily, ongoing event scheduling.

Using RESOURCE25. A guide for the RESOURCE25 user performing the daily tasks of creating, modifying, and allocating space to events. Introduces basic concepts, describes the working environment, and contains instructions for all related tasks.

Migrating From 25E to RESOURCE25. Detailed strategy planning information for migrators.

You can purchase additional hardcopy manuals by emailing *bett@unival.com* or calling 503.973.5200.

What to look for on-line

The following RESOURCE25 information is available on-line:

On-line reference and instructional help. RESOURCE25 on-line documentation is available for all forms. Simply choose Help|Current Form or press F1 from any RESOURCE25 form to access help information.

All of the manuals listed in the "in print" section on the previous page. RESOURCE25 manuals are available on-line for viewing and printing at no charge. To access them, click InfoFinder on any RESOURCE25 on-line help form, or select InfoFinder from the Help menu. Follow the on-screen instructions.

The manuals are also available on our web site (*www.unival.com*).

Customer support

If RESOURCE25 is not working as you expect

Customer support is available from 8:00 a.m. to 5 p.m. Pacific Standard (or Daylight) Time. If you telephone outside these hours, leave a voice mail message, and we'll return your call as soon as possible.

If you're located in a country whose business hours don't overlap those of Universal Algorithms, arrangements can be made for a support call outside our regular hours.

The Universal Algorithms telephone numbers are:

- Customer support – 503.973.5250
- Main office line – 503.973.5200
- Fax – 503.973.5252

You can also reach us by email at *support@unival.com*.

If you have comments

Send us your comments and suggestions

Please let us know how we're doing. Send an email comment to *margo@unival.com* if you:

- Discover an error in the documentation
- Find a concept or instruction confusing
- Have a suggestion to improve the documentation
- Have a kind word to say!

Overview

1 Introduction to Event Data Flows

Introduction

Overview

Before you begin designing interfaces to link RESOURCE25, SCHEDULE25, and your SIS, you need to understand the data flow options available to you.

There are two types of data flows you can set up in your scheduling environment:

- A **bulk** data flow
- A **transaction-based** data flow

What's in this chapter

This chapter is designed to familiarize you with these event data flows.

Chapter contents

Bulk event data flow

Description

The bulk event data flow is a process you can set up to send class records from your SIS to SCHEDULE25 for room assignment and/or to RESOURCE25 for room assignment or other changes. Updated class records can then be sent back to your SIS from SCHEDULE25 and/or RESOURCE25.

It is a bulk process because all the class records for a term or semester are processed *during a defined time period* to prepare the class schedule for that term/semester.

Bulk data flow options

There are four bulk data flow options you can choose from. Each option is illustrated and described beginning on the next page.

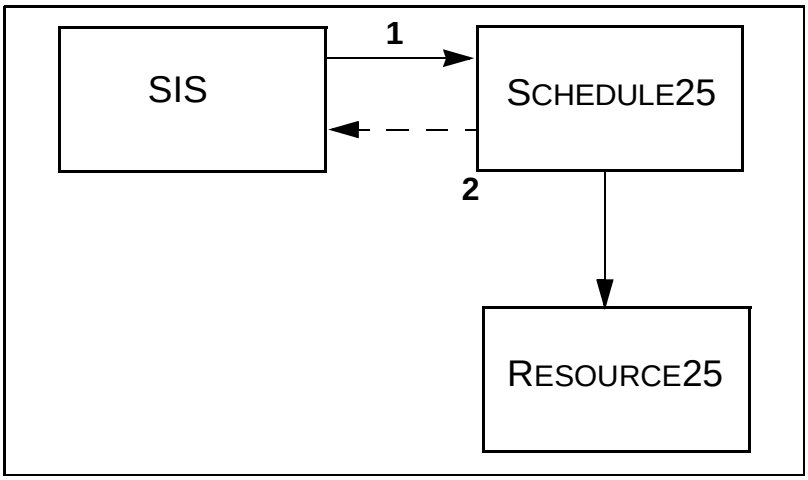
See chapter 3 for detailed information about how each of these processes would function in your scheduling environment.

Bulk event data flow options

Option 1:

Data flow diagram

This is a diagram of bulk data flow option 1:



Description of data flow

In this data flow:

1. Class event records are extracted from your SIS and sent to SCHEDULE25.
2. SCHEDULE25 is run. Its output of classes with room assignments is sent to RESOURCE25 and, optionally, may also be sent back to your SIS.

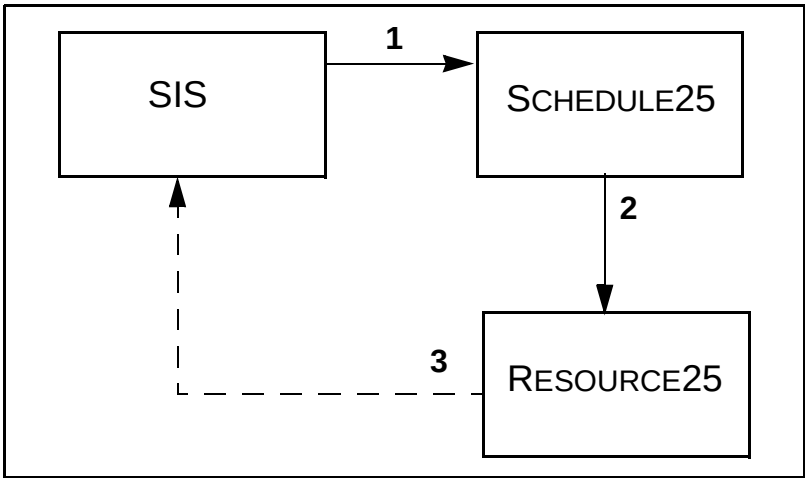
Data format

In this option, class records are put in class descriptor data format for transfer between systems. See chapter 2 for more information.

Option 2:

Data flow diagram

This is a diagram of bulk data flow option 2:



Description of data flow

In this data flow:

1. Class event records are extracted from your SIS and sent to SCHEDULE25.
2. SCHEDULE25 is run. Its output of classes with room assignments and without room assignments is sent to RESOURCE25.
3. RESOURCE25 is used to assign rooms to classes that need them, and, optionally, all the updated class records are sent back to your SIS.

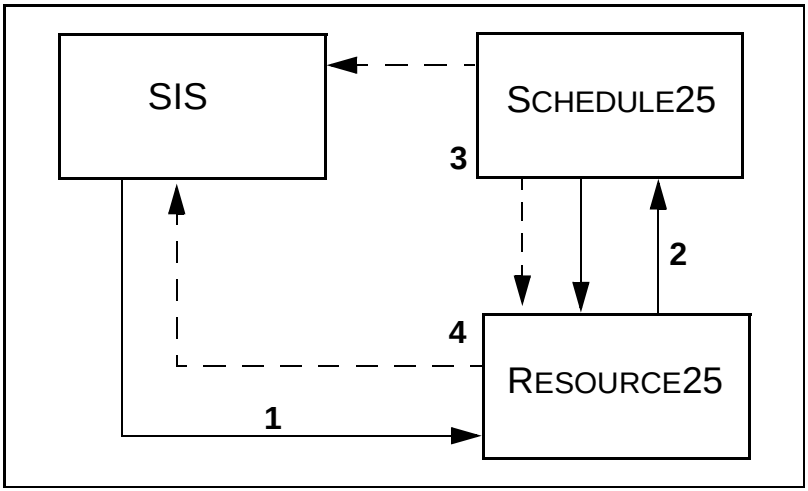
Data format

In this option, class records are put in class descriptor data format for transfer between systems. See chapter 2 for more information.

Option 3:

Data flow diagram

This is a diagram of bulk data flow option 3:



Description of data flow

In this data flow:

1. Class event records are extracted from your SIS and sent to RESOURCE25.
2. RESOURCE25 is used to add space feature and location preferences to the class records, and the updated class records are sent to SCHEDULE25.
3. SCHEDULE25 is run. Its output of classes with room assignments is sent to RESOURCE25 and, optionally, back to your SIS.
4. Class records without room assignments are sent to RESOURCE25. RESOURCE25 is used to assign rooms to those classes, and, optionally, the updated class records are sent to your SIS.

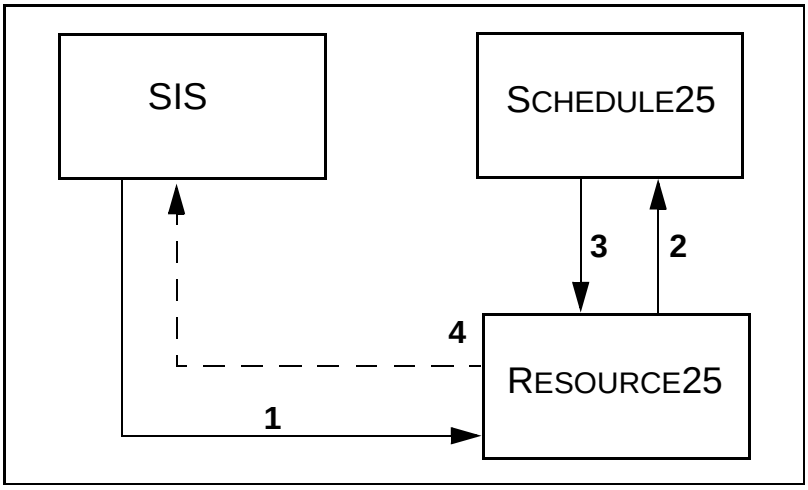
Data format

In this option, class records are put in class descriptor data format for transfer between systems. See chapter 2 for more information.

Option 4:

Data flow diagram

This is a diagram of bulk data flow option 4:



Description of data flow

In this data flow:

1. Your SIS sends a group of class events to RESOURCE25.
2. RESOURCE25 is used to add space feature and location preferences to the class records, and the updated class records are sent to SCHEDULE25.
3. SCHEDULE25 is run. Its output of classes with room assignments and without room assignments is sent to RESOURCE25.
4. RESOURCE25 is used to assign rooms to classes that need them, and, optionally, all the updated class events are sent to your SIS.

Note: *The process used to send records to RESOURCE25 from your SIS and receive updates back to your SIS from RESOURCE25 is the same process used in the transaction-based data flow described on page* 1-8.

Data format

In this option, class records are put in class descriptor data format for transfer between RESOURCE25 and SCHEDULE25, and in vCalendar data format for transfer between RESOURCE25 and your SIS. See chapter 2 for more information.

Transaction-based event data flow

Description

The transaction-based event data flow is a process you can set up to keep the class event records in your SIS and in RESOURCE25 in sync on an ongoing basis.

Important: *In this environment, your SIS is always the initiator of the transactions, never RESOURCE25.*

When a new class is added to your SIS, a change to an existing class is made, or a class is cancelled, your SIS sends a message to RESOURCE25. RESOURCE25 updates its database accordingly and, if you have specified that you want it to, sends a message back to your SIS.

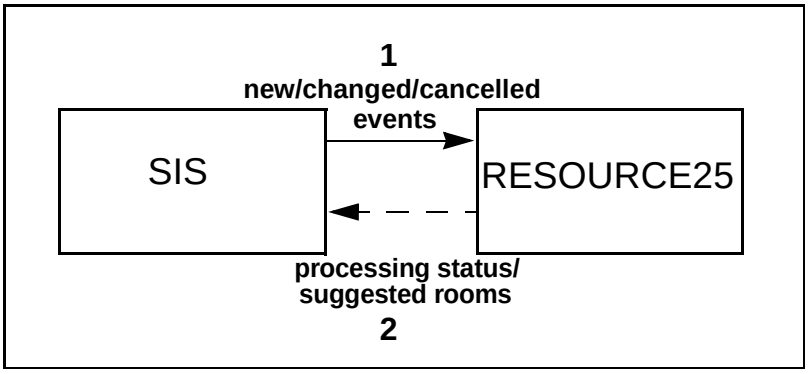
The message may include room suggestions for class events that were sent to RESOURCE25 without room assignments. If your SIS agrees with the room suggestions, the room assignments are subsequently made to the corresponding class records in your SIS.

This is a process you set up to run automatically at predetermined time intervals.

See chapter 6 for detailed information about how this process would function in your scheduling environment.

Data flow diagram

This is a diagram of the transaction-based event data flow:



Description of data flow

In this data flow:

1. Your SIS sends a message alerting RESOURCE25 of class records that have been added, changed, or cancelled in your SIS.

2. RESOURCE25 attempts to update its database with the class additions, changes, and cancellations. If you have specified that you want it to, it also sends a message back to your SIS indicating that the changes were processed successfully, or suggesting rooms for class events without room assignments.

Note: *The same process is used in bulk data flow option 4 (page 1-6) to send records to RESOURCE25 from your SIS and receive updated records back to your SIS from RESOURCE25.*

Data format

In this data flow, class records are put in vCalendar data format for transfer between RESOURCE25 and your SIS. See chapter 2 for more information.

2 Event Data Formats

Introduction

What's in this chapter

This chapter describes the two data formats you can use to share event data among RESOURCE25, SCHEDULE25, and your SIS:

- Class descriptor format
- vCalendar format

It also reviews which data format option(s) are applicable to each event data flow and provides examples of each data format.

Chapter contents

Class descriptor data format

Description

The class descriptor data format is the format used to share event data between RESOURCE25 and SCHEDULE25 and can be used to share event data between RESOURCE25 and your SIS.

If you are already using SCHEDULE25 successfully in your scheduling environment and have built your course master extract and update programs, you are probably already familiar with the class descriptor data format.

If you're not familiar with this data format, read the brief description that begins below. Detailed information about this format and how to use it to share data between SCHEDULE25 and your SIS is in the *SCHEDULE25 Data Preparation* manual available on our web site (*www.unival.com*).

Class Descriptor file

The Class Descriptor file is an ASCII file that contains the class records you want SCHEDULE25 to process. This file is prepared by your institution, and its data format, which you define in the Control file (*ctrl.dat*), is *unique to your campus*.

Each Class Descriptor record describes a class, including date information, days of the week, times, and enrollment. Physical feature requirements and location preferences can also be specified using P and L records.

Class Descriptor records must be at least 80 columns long, but no longer than 253 columns.

By defining the Class Descriptor record format, you are also implicitly defining the record format of the SCHEDULE25 *errors.dat, notposs.dat, losers.dat, sortrm.dat, sortdp.dat, sortrm.pl* and other output files.

Class Descriptor file example

Below is a portion of a Class Descriptor file. The placement and length of the required fields in this file are defined in the Control file (see below).

```
PSY 10101  MTWRF   961002 961219 1130A1220P10271 B 215        97F068704051    ASM MEDDELL
           GEN PSYCHOLOGY
PSY 10102   T R    961002 961219 0300P0430P10296 B 215        97F068804051    ASM FERGUSON
           GEN PSYCHOLOGY
PSY 11001   T R    961002 961219 0330P0500P10130 EDUC 131     97F068904051    15M ASCIONE
           HUM DEVELOP: GENERAL
PSY 12101  M W F   961002 961219 1030A1120A10060              97F069004051    NSM STAFF
           HUMAN ADJUSTMENT
PSY 12201   T R    961002 961219 0230P0345P10130              97F069104051    NSM MAN
           CAREER EXPLORATION
5.19 L
PSY 14001  MTWRF   961002 961219 0830A0920A10060              97F069304051    NSM CHENEY
           ANALYSIS BEHAVIOR:  BASIC
PSY 17301   T R    961002 961219 1030A1200P10090              97F069604051    NSM SAUNDERS
           PERSONAL STUDY EFFICIENCY
PSY 17302  M W F   961002 961219 0930A1020A10035              97F069704051    NSM CALL
           PERSONAL STUDY EFFICIENCY
3 L
PSY 17303   T R    961002 961219 1130A1245P10090              97F069804051    NSM LOPEZ
           PERSONAL STUDY EFFICIENCY
PSY 17501   T R    961002 961219 1030A1120A10020              97F070304051    NSM BERNHISEL
           COLLEGE READING/LISTENING
PSY 20001   T R    961002 961219 0900A1020A10027              97F070404051    NSM ASCIONE
           ADOLESCENT PSYCHOLOGY
PSY 22101  M W F   961002 961219 1230P0120P10080              97F070454051 HC NSM GIMPEL
           ABNORMAL PSYCHOLOGY
PSY 22102  M W F   961002 961219 1230P0120P10080              97F070464051 XX NSM GIMPEL
           ABNORMAL PSYCHOLOGY
PSY 24601  M W F   961002 961219 1130A1220P10080              97F070604051    NSM CHENEY
           PHYSIOLOGICAL PSY
```

The Control file

The Control file (*ctrl.dat)* is used to define the position and content requirements of the required fields in the Class Descriptor record. Although SCHEDULE25 requires that certain fields be in the Class Descriptor record, you determine the position of the fields. The Control file also defines the SCHEDULE25 processing settings that influence SCHEDULE25 processing. If you have already created your Control file, you can import it into RESOURCE25 and maintain it there. If you make changes to it using RESOURCE25, you simply export it from RESOURCE25 prior to the next SCHEDULE25 run.

If you have not yet created your Control file, you can enter your class descriptor format information and the SCHEDULE25 processing settings in RESOURCE25 and then export them. When you export this data, RESOURCE25 creates a Control file using the data you've entered.

See chapters 6 and 7 for information about importing, maintaining, and exporting Control file data.

Data flow options that use the class descriptor data format

All of the bulk data flow options described in chapter 1 use the class descriptor data format to transfer event data between RESOURCE25 and SCHEDULE25. This is the only event data format SCHEDULE25 understands.

Bulk data flow options 2 and 3 also use the class descriptor format to transfer event data between RESOURCE25 and the SIS. Bulk data flow option 4 and the transaction-based data flow use the vCalendar data format to transfer event data between RESOURCE25 and the SIS.

If you have already built your SCHEDULE25/SIS interface using the class descriptor data format, you can continue to use that data format for bulk event data transfer between your SIS and RESOURCE25, if you want to. However, you should use the vCalendar data format for the transaction-based interface.

Chapter 3 describes each of the bulk interfaces in detail and provides guidance in choosing the right one for your scheduling environment.

Chapter 6 describes the transaction-based interface in detail.

vCalendar data format

Description

The vCalendar data format is a format that can be used to share event data between RESOURCE25 and your SIS. The data format is based on an industry-wide specification for event information that is system-independent. The complete specification is available at these web sites:

Internet Mail Corp	**http://www.imc.org/pdi/**
IETF Scheduling Working Group	**http://www.imc.org/ietf-calendar/**

vCalendar file

A vCalendar file is an ASCII file that contains a "vCalendar object." A vCalendar object is a container for one or more "vEvent objects." Each vEvent object represents a scheduled amount of time on a calendar.

vEvent objects in the same vCalendar object can be closely related (for example, several date definitions for a single RESOURCE25 event) or unrelated. Each vEvent object must have a unique key that identifies it to RESOURCE25.

RESOURCE25 can handle vCalendar objects containing only one vEvent object and vCalendar objects containing more than one vEvent object.

"Alien" events and "native" events

RESOURCE25 regards vEvent objects it imports from other systems as "alien" events. This means that the vCalendar data in these events (the data that is shared between RESOURCE25 and your SIS) is subject to special vCalendar processing rules. These processing rules are selected from several vCalendar import processing options (see page 6-4).

RESOURCE25 regards events created in RESOURCE25 as "native" events. This means that the data in these events is not subject to the vCalendar processing rules.

vCalendar method types

Each vCalendar object has an identified "method type" that defines the purpose of the object and how it should be processed. The possible method types are:

- **Request**. The vCalendar object contains new events that should be added to the RESOURCE25 database and/or changes to existing events that should be made to the corresponding events in the RESOURCE25 database. "Request" vCalendar objects are generated by the SIS.
- **Cancel**. The vCalendar object contains events that should be cancelled in the RESOURCE25 database. "Cancel" vCalendar objects are generated by the SIS.
- **Reply**. The vCalendar object does *one* of these:

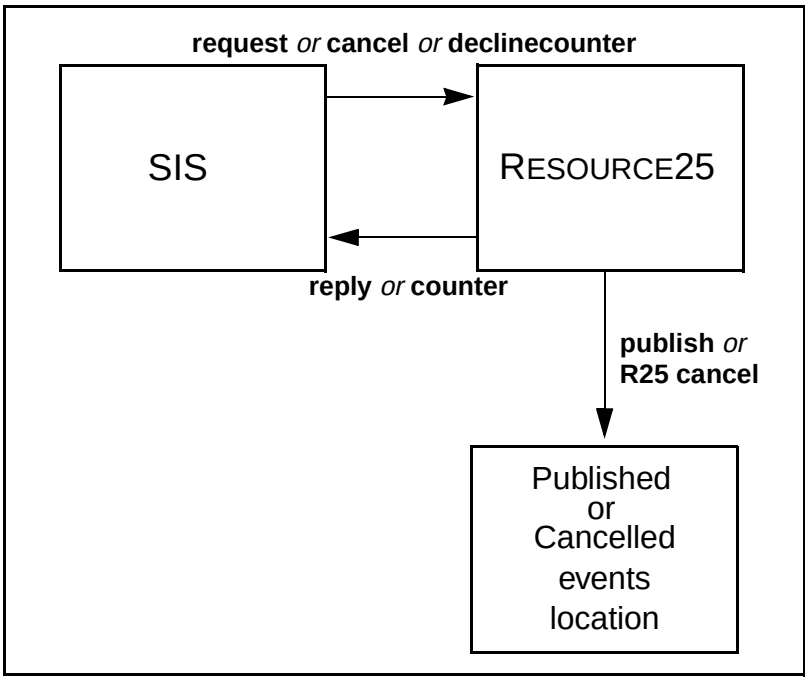
  Confirms that new and/or changed events have been successfully added or modified in the RESOURCE25 database.
  or
  Confirms that cancelled events have been successfully cancelled in the RESOURCE25 database.

  "Reply" vCalendar objects are generated by RESOURCE25 in response to a "request" object from the SIS. *There is no new event data in a reply.*
- **Counter**. The vCalendar object contains suggested room assignments for the events in the original request that did not have room assignments. "Counter" vCalendar objects are generated by RESOURCE25 in response to a "request" object from the SIS. There *is* new event data (room assignments) in a counter.
- **Decline Counter**. The vCalendar object rejects the suggested room assignment made by RESOURCE25 in the previous "counter" object. "Decline Counter" vCalendar objects are generated by the SIS in response to a counter.

vCalendar method types, continued

- **Publish**. The vCalendar object contains new events and changes to existing events that are to be posted for notification. This method is used primarily to advertise the existence of events, and represents an event "snapshot" in time. "Publish" vCalendar objects are generated by RESOURCE25.
- **R25 Cancel**. The vCalendar object contains "native" RESOURCE25 events that have been cancelled. This method is used to indicate the cancellation of published events. "R25 Cancel" vCalendar objects are generated by RESOURCE25.

This is an illustration of the method types in action:



How RESOURCE25 and your SIS process vCalendar objects based on method type is described in detail in chapter 7.

vCalendar object example

The following is a request and reply sequence of vCalendar objects passed between an SIS and RESOURCE25.

The SIS system is adding a new computer science section on Tuesdays and Thursdays from 9AM to 10AM in room LT 202. The expected head count is 50. The SIS generates this request object:

```
BEGIN:VCALENDAR
METHOD:REQUEST
VERSION:1.0
PRODID:-//SIS Vendor//NONSGML MySIS//EN
BEGIN:VEVENT
UID:MySIS-CRN:003244
SUMMARY:CS138 - Introduction to Java
SEQUENCE:1
PRIORITY:0
DCREATED:19970812T105000
LAST-MODIFIED:19970812T104000
DTSTART:19980907T090000
DTEND:19980907T100000
RRULE:W1 TU TH #10 19981231T235900
LOCATION:LT 202
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-R25-ACCOUNT:Computer Science
X-R25-TYPE:Section
END:VEVENT
END:VCALENDAR
```

Request Method

Date/time information

Location

Expected head count

Sponsoring department

vCalendar object example, continued

When RESOURCE25 receives this request, it adds the event to its database and checks on the availability of the room. If the room is available, it generates this reply object:

Reply Method

Event successfully added

```
BEGIN:VCALENDAR
METHOD:REPLY
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
 RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=935:MySIS-CRN:003244
SEQUENCE:1
X-R25-REQUEST-STATUS:200;Success
END:VEVENT
END:VCALENDAR
```

Chapter 7 explains how to issue vCalendar objects from your SIS and how to update your SIS from vCalendar objects issued by RESOURCE25.

Appendix A defines each of the data elements in the vCalendar format.

Data flow options that use the vCalendar data format

Bulk data flow option 4 and the transaction-based data flow described in chapter 1 use the vCalendar data format to transfer event data between RESOURCE25 and the SIS.

Chapter 3 describes each of the bulk interfaces in detail and provides guidance in choosing the right one for your scheduling environment.

Chapter 6 describes the transaction-based interface in detail.

Bulk Event Data Interface

3 Bulk Event Data Interface Options

Introduction

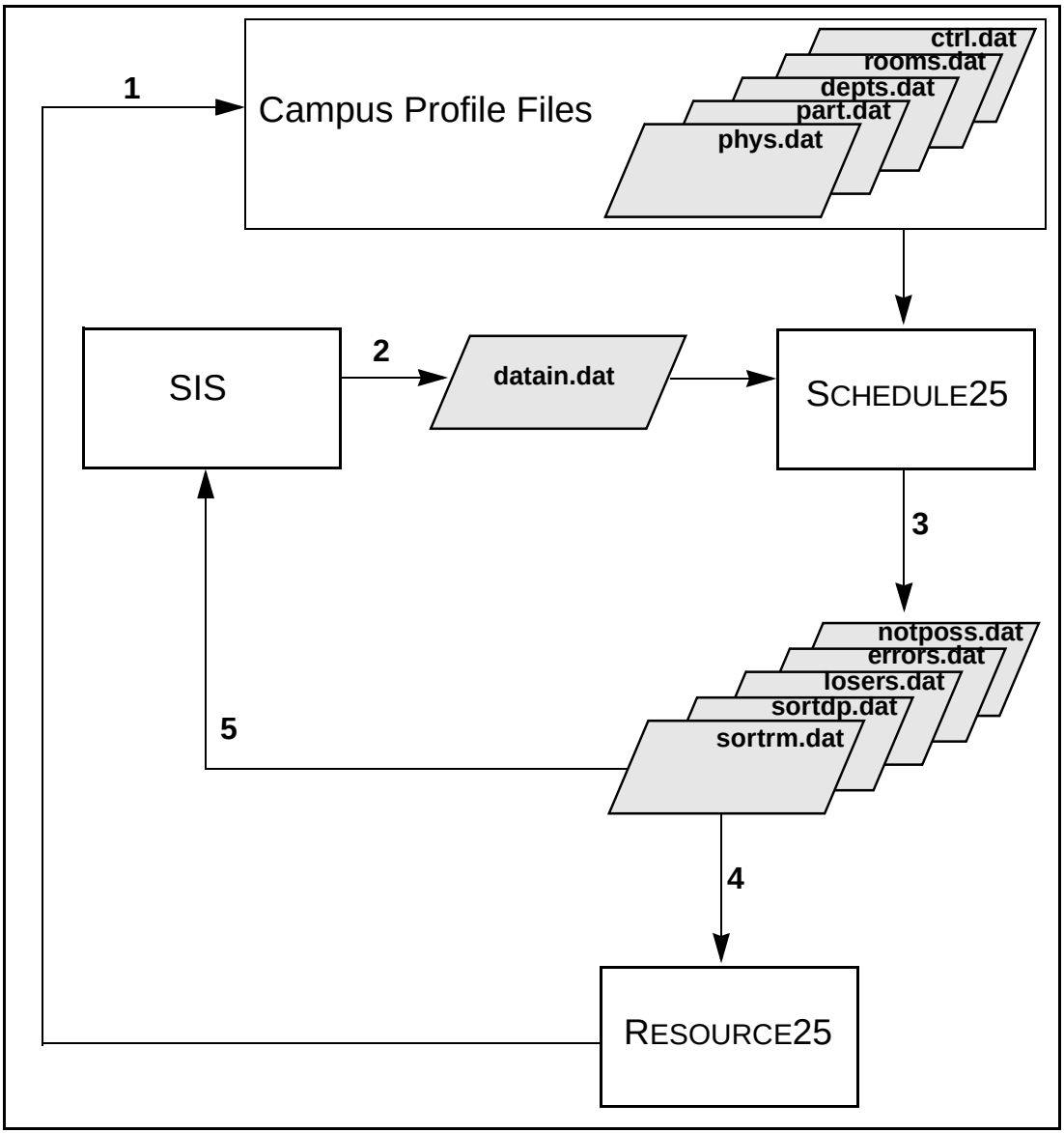
What's in this chapter

This chapter illustrates and explains in detail each of the bulk event data interface options summarized in chapter 1. It also offers guidance in choosing the right option for your scheduling environment and tells you the implementation requirements of each.

Chapter contents

Bulk interface option 1

Interface diagram This is a diagram of bulk interface option 1:



Description of interface

In this interface:

1. The RESOURCE25 export functionality is used to export your campus profile files and Control file in preparation for SCHEDULE25 processing.

2. A course master extract program written by your institution is run. This program extracts class records from your SIS and creates a Class Descriptor file (*datain.dat*).

3. SCHEDULE25 is run to assign rooms to the extracted class records. Using your campus profile files, the Control file, and the *datain.dat* file generated by your extract program, SCHEDULE25 processes the class records and generates a group of output files. One of these files is *sortrm.dat*, a list of class records with room assignments sorted by room.

4. The RESOURCE25 import functionality is used to import the *sortrm.dat* file into RESOURCE25 and place the incoming class records in the appropriate locations in your RESOURCE25 event structure. **Caution**: *The sortdp.dat file should not be imported.*

5. A course master update program written by your institution is run. This program uses the *sortrm.dat* file to update the corresponding class records in your SIS.

Caution: *In this interface, RESOURCE25 sees all the events as new. The interface should only be run once for all events in a specific time period (term, semester) to avoid duplicating events or inadvertently creating room conflicts.*

Data formats used

All of the files mentioned in the interface description above are in class descriptor data format. See chapter 2 for information on this data format.

Is this option right for you?

This option is right for you if:

- You have already created your campus profile files and Control file and are successfully running SCHEDULE25.
- You have already built your course master extract and update programs.
- You have imported your campus profile data into RESOURCE25, are maintaining it there, and know how to export the data for use by SCHEDULE25. Complete instructions for importing, maintaining, and exporting campus profile data are in the *RESOURCE25 Data Preparation 1: Space and Customers* manual.
- You don't want to use RESOURCE25 to assign rooms that SCHEDULE25 could not assign.

Implementation requirements of option 1

If you meet all the criteria listed above, option 1 has the following implementation requirement:

You need to learn how to...	For information see...
Import event records in class descriptor data format into RESOURCE25	This manual, chapter 5

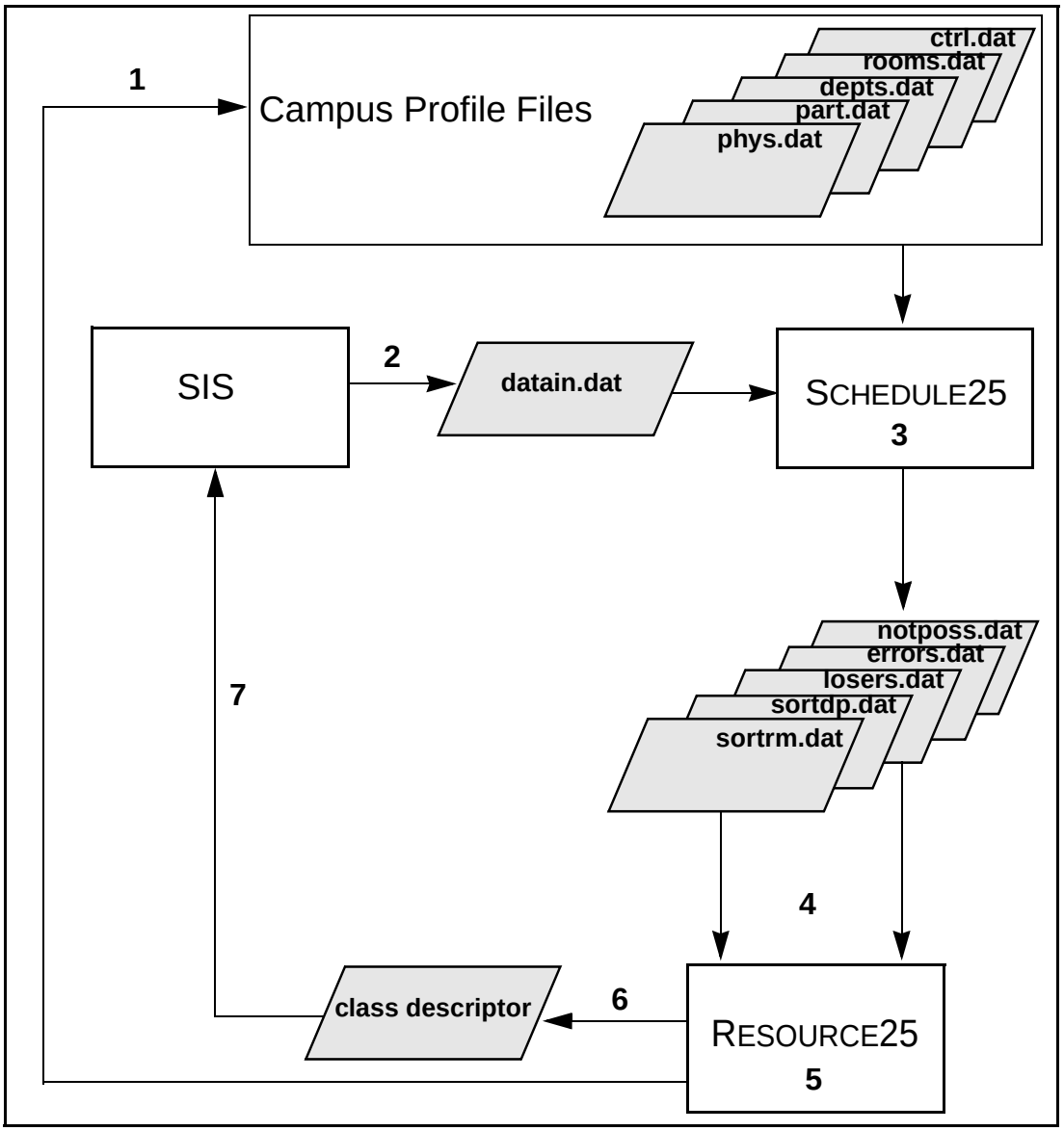
You can also import your Control file and maintain it in RESOURCE25. If you make changes to your Control file between SCHEDULE25 runs, you must export the updated file prior to running your course master extract program.

If you want to maintain your Control file using RESOURCE25, option 1 has these additional implementation requirements:

You need to learn how to...	For information see...
Import, maintain, and export your Control file using RESOURCE25	This manual, chapters 4 and 5

Bulk interface option 2

Interface diagram This is a diagram of bulk interface option 2:



Description of interface

In this interface:

1. The RESOURCE25 export functionality is used to export your campus profile files and Control file in preparation for SCHEDULE25 processing.

2. A course master extract program written by your institution is run. This program extracts class records from your SIS and creates a Class Descriptor file (*datain.dat*).

3. SCHEDULE25 is run to assign rooms to the extracted class records. Using your campus profile files, the Control file, and the *datain.dat* file generated by your extract program, SCHEDULE25 processes the class records and generates a group of output files. Among these files is *sortrm.dat*, a list of class records with room assignments sorted by room and *losers.dat*, a list of class records that could not be placed in rooms because of scheduling conflicts with other classes.

4. The RESOURCE25 import functionality is used to import the *sortrm.dat* and *losers.dat* files into RESOURCE25 and place the incoming class records in the appropriate locations in your RESOURCE25 event structure. **Caution**: *The sortdp.dat file should not be imported.*

5. RESOURCE25 is used to assign rooms to the class records without room assignments.

6. The RESOURCE25 export functionality is used to export the class records.

7. A course master update program written by your institution is run. This program uses the export file generated by RESOURCE25 to update the corresponding class records in your SIS.

Data formats used

All of the files mentioned in the interface description on the previous page are in class descriptor data format. See chapter 2 for information on this data format.

Is this option right for you?

This option is right for you if:

- You have already created your campus profile files and Control file and are successfully running SCHEDULE25.
- You have already built your course master extract and update programs.
- You have imported your campus profile data into RESOURCE25, are maintaining it there, and know how to export the data for use by SCHEDULE25. Complete instructions for importing, maintaining, and exporting campus profile data are in the *RESOURCE25 Data Preparation 1: Space and Customers* manual.
- You want to use RESOURCE25 to assign rooms that SCHEDULE25 could not assign.

Implementation requirements of option 2

If you meet all the criteria listed above, option 2 has the following implementation requirements:

You need to learn how to...	For information see...
Import event records in class descriptor data format into RESOURCE25	This manual, chapter 5
Export event records in class descriptor data format from RESOURCE25	This manual, chapter 5
Assign rooms to events using RESOURCE25	*Using RESOURCE25* (use InfoFinder to access)

Implementation requirements of option 2, continued

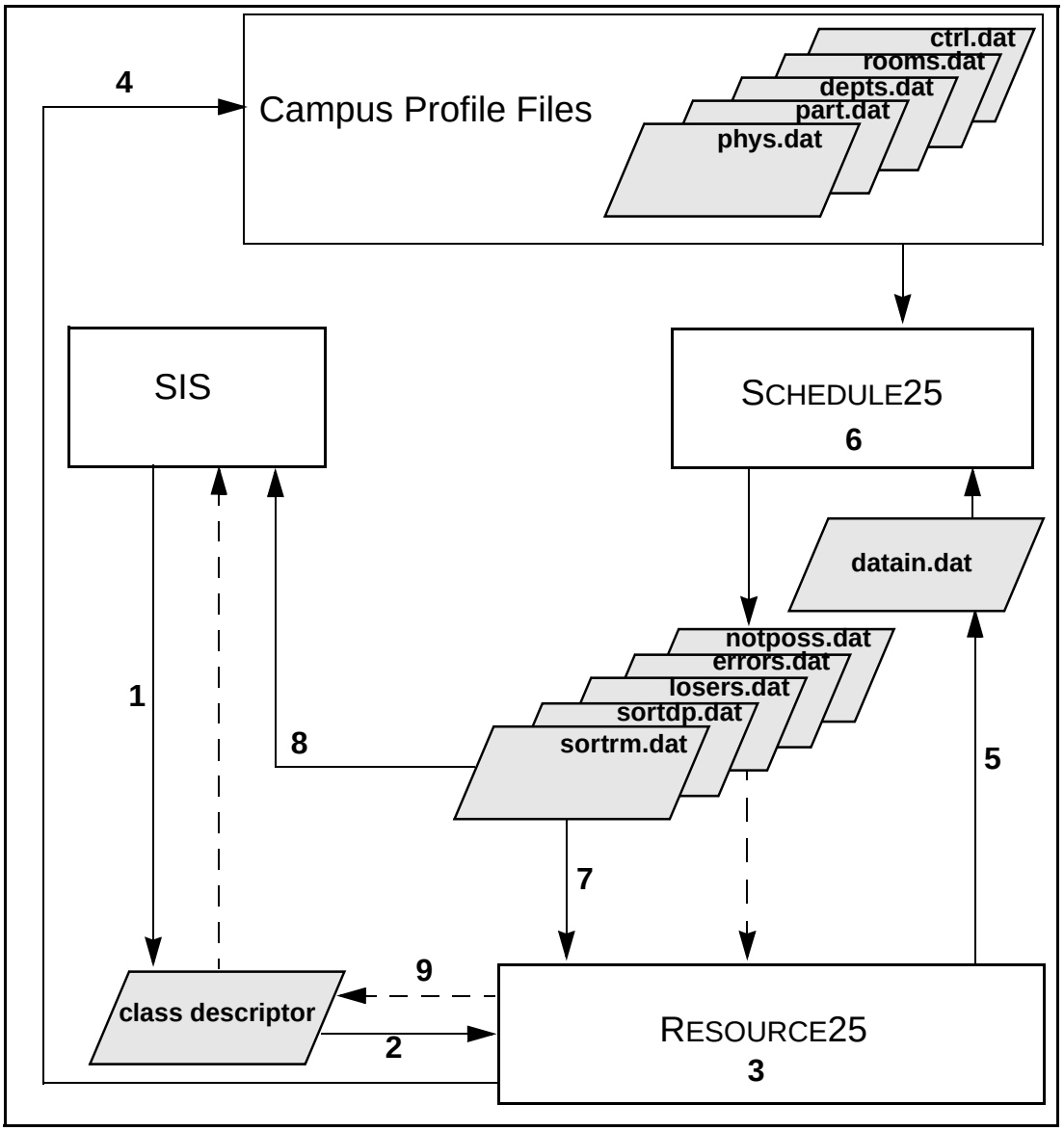
You can also import your Control file and maintain it in RESOURCE25. If you make changes to your Control file between SCHEDULE25 runs, you must export the updated file prior to running your course master extract program.

If you want to maintain your Control file using RESOURCE25, option 2 has these additional implementation requirements:

You need to learn how to...	For information see...
Import, maintain, and export your Control file using RESOURCE25	This manual, chapters 4 and 5

Bulk interface option 3

Interface diagram This is a diagram of bulk interface option 3:



Description of interface

In this interface:

1. A course master extract program written by your institution is run. This program extracts class records from your SIS and puts them in class descriptor data format.

2. The RESOURCE25 import functionality is used to import the class record file generated by your extract program into RESOURCE25 and place the incoming class records in the appropriate locations in your RESOURCE25 event structure.

3. RESOURCE25 is used to add space feature and location preferences to the class records.

4. The RESOURCE25 export functionality is used to export your campus profile files and Control file in preparation for SCHEDULE25 processing.

5. The RESOURCE25 export functionality is used to export a file of class records (*datain.dat*) to SCHEDULE25 for processing.

6. SCHEDULE25 is run to assign rooms to the class records. Using your campus profile files, the Control file, and the *datain.dat* file exported from RESOURCE25, SCHEDULE25 processes the class records and generates a group of output files. Among these files is *sortrm.dat*, a list of class records with room assignments sorted by room, and *losers.dat*, a list of class records that could not be placed in rooms because of scheduling conflicts with other classes.

7. The RESOURCE25 import functionality is used to import the *sortrm.dat* file generated by SCHEDULE25 into RESOURCE25 and update the corresponding event records. **Caution**: *The sortdp.dat file should not be imported.*

8. A course master update program written by your institution is run. This program uses the *sortrm.dat* file generated by SCHEDULE25 to update the corresponding class records in your SIS.

Description of interface, continued

9. Optionally, *losers.dat* can be imported into RESOURCE25 for room assignment and then exported in class descriptor format. The update program can then be run against the export file to update the corresponding class records in your SIS.

Data formats used

All of the files mentioned in the interface description on the previous page and above are in class descriptor data format. See chapter 2 for information on this data format.

Is this option right for you?

This option is right for you if:

- You have already created your campus profile files and Control file and are successfully running SCHEDULE25.
- You have already built your course master extract and update programs.
- You have imported your campus profile data into RESOURCE25, are maintaining it there, and know how to export the data for use by SCHEDULE25. Complete instructions for importing, maintaining, and exporting campus profile data is in the *RESOURCE25 Data Preparation 1: Space and Customers* manual.
- You want to use RESOURCE25 to add space feature and location preferences to class records.
- You may want to use RESOURCE25 to assign rooms that SCHEDULE25 could not assign.

Implementation requirements of option 3

If you meet all the criteria listed on the previous page, option 3 has the following implementation requirements:

You need to learn how to...	For information see...
Import event records in class descriptor data format into RESOURCE25	This manual, chapter 5
Export event records in class descriptor data format from RESOURCE25	This manual, chapter 5
Add space features and location preferences to events using RESOURCE25	*Using RESOURCE25* (use InfoFinder to access)
Assign rooms to events using RESOURCE25	*Using RESOURCE25* (use InfoFinder to access)

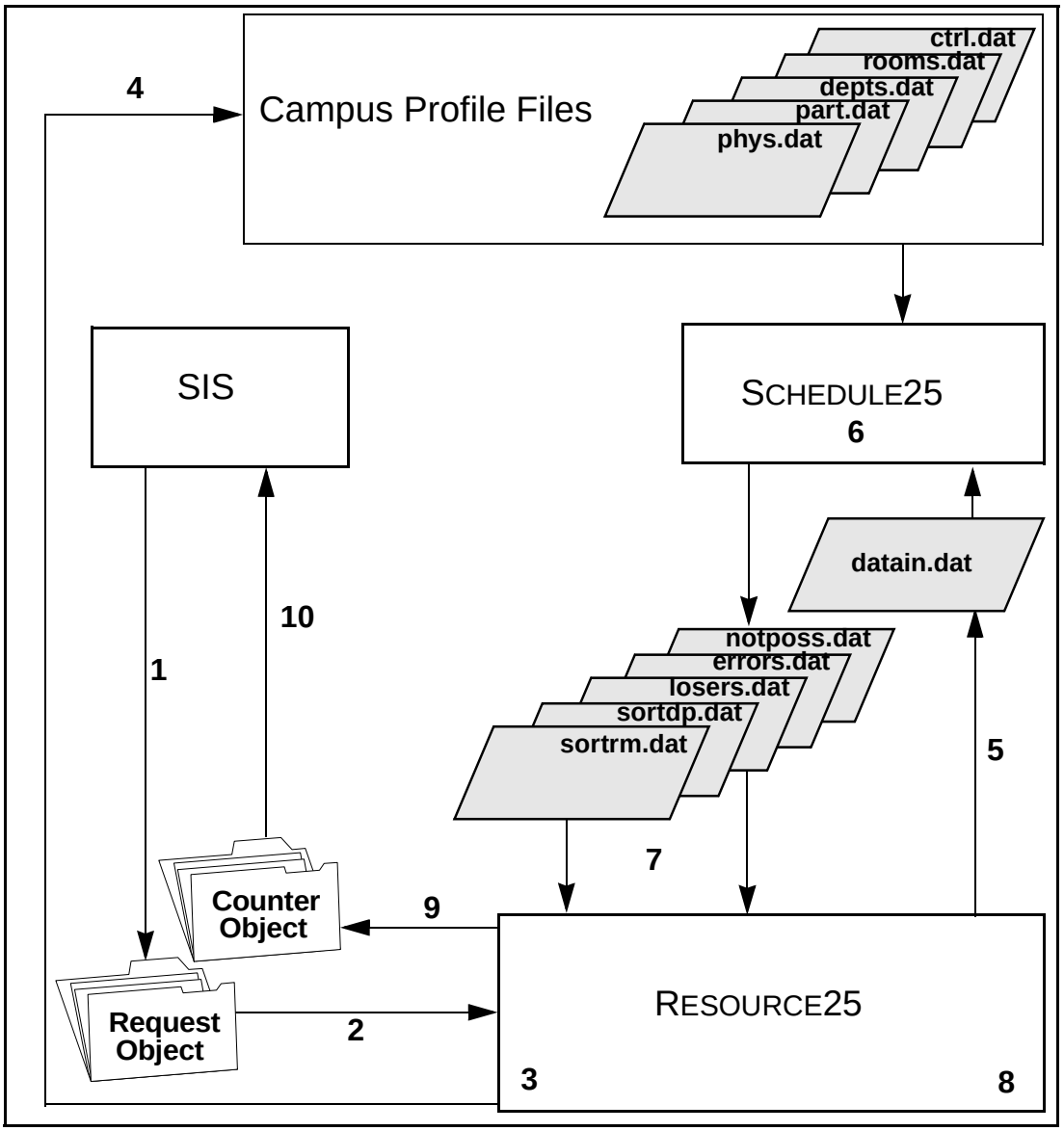
You can also import your Control file and maintain it in RESOURCE25. If you make changes to your Control file between SCHEDULE25 runs, you must export the updated file prior to running your course master extract program.

If you want to maintain your Control file using RESOURCE25, option 3 has these additional implementation requirements:

You need to learn how to...	For information see...
Import, maintain, and export your Control file using RESOURCE25	This manual, chapters 4 and 5

Bulk interface option 4

Interface diagram This is a diagram of bulk interface option 4:



Description of interface

In this interface:

1. A course master extract program written by your institution is run. This program extracts class records from your SIS, generates a vCalendar "request" object containing those events, and sends it to RESOURCE25.

2. The RESOURCE25 vCalendar import functionality is used to import the class record data generated by the extract program into RESOURCE25 and places the incoming class records in the appropriate locations in your RESOURCE25 event structure.

3. RESOURCE25 is used to add space feature and location preferences to the class records.

4. The RESOURCE25 export functionality is used to export your campus profile files and your Control file in preparation for SCHEDULE25 processing.

5. The RESOURCE25 export functionality is used to export a file of class records (*datain.dat*) to SCHEDULE25 for processing.

6. SCHEDULE25 is run to assign rooms to the class records. Using the campus profile files, the Control file, and the *datain.dat* files exported from RESOURCE25, SCHEDULE25 processes the class records and generates a group of output files. Among these files is *sortrm.dat*, a list of class records with room assignments sorted by room, and *losers.dat*, a list of class records that could not be placed in rooms because of scheduling conflicts with other classes.

7. The RESOURCE25 import functionality is used to import the *sortrm.dat* and *losers.dat* files generated by SCHEDULE25 into RESOURCE25 and update the corresponding event records. **Caution**: *The sortdp.dat file should not be imported.*

8. RESOURCE25 is used to assign rooms to class records without room assignments.

9. The RESOURCE25 vCalendar export functionality automatically exports a vCalendar "counter" object containing the suggested room assignments.

Description of interface, continued

10. A course master update program written by your institution is run. This program reads the export file and updates the class data in your SIS.

The SIS/ RESOURCE25 part of the interface

The process used to send records to RESOURCE25 from your SIS and receive record updates back to your SIS from RESOURCE25 is the same process used in the transaction-based interface described in detail in chapter 6. *Read that chapter carefully.*

Data formats used

In this option, class records are put in class descriptor data format for transfer between RESOURCE25 and SCHEDULE25, and in vCalendar data format for transfer between RESOURCE25 and your SIS. See chapter 2 for information on these data formats.

Is this option right for you?

This option is right for you if:

- You are new to both RESOURCE25 and SCHEDULE25, and have not yet created your campus profile and Control files.
- You have not yet built your course master extract and update programs.
- You want to use RESOURCE25 to add space feature and location preferences to class records.
- You want to use RESOURCE25 to assign rooms that SCHEDULE25 could not assign.
- You only want to write one interface from/to your SIS for your “bulk” and “transaction-based” interfaces.

Implementation requirements of option 4

If you meet all the criteria listed on the previous page, option 4 has the following implementation requirements:

You need to learn how to...	For information see...
Create your campus profile data in RESOURCE25 and export it	*RESOURCE25 Data Preparation 1: Space and Customers* (for information and instructions - use InfoFinder to access) *Data Preparation - SCHEDULE25* (for necessary reference information - available on the UAI web site, *www.univ al.com*)
Define your class descriptor file and SCHEDULE25 processing settings using RESOURCE25, and export them to create your Control file	This manual, chapters 4 and 5 *Data Preparation - SCHEDULE25* (for necessary reference information - available on the UAI web site, *www.univ al.com*)
Build your course master extract and update programs	This manual, chapter 4
Import event records in class descriptor data format into RESOURCE25	This manual, chapter 5
Export event records in class descriptor data format from RESOURCE25	This manual, chapter 5
Prepare RESOURCE25 for the import and export of vCalendar files	This manual, chapter 4

You need to learn how to...	For information see...
Add space features and location preferences to events using RESOURCE25	*Using RESOURCE25* (use InfoFinder to access)
Assign rooms to events using RESOURCE25	*Using RESOURCE25* (use InfoFinder to access)

4 Building Your Bulk Interface

Introduction

What's in this chapter

This chapter provides instructions and guidance for performing the tasks necessary to build each of the bulk interface options described in chapter 3.

This chapter tells you how to:

- Import and maintain the Control file in RESOURCE25
- Define your Class Descriptor file using RESOURCE25
- Define SCHEDULE25 processing settings using RESOURCE25
- Create course master extract and update programs
- Prepare RESOURCE25 for the import of Class Descriptor files
- Prepare RESOURCE25 for the import and export of vCalendar files

Chapter contents

Tasks to build bulk interface option 1, 2, or 3

If you're already using SCHEDULE25...

If you're already using SCHEDULE25 and have built your course master extract and update programs, you must perform the following tasks to build bulk interface option 1, 2, or 3:

Task	Instructions/guidance
Import and maintain your Control file in RESOURCE25	page 4-5
Prepare RESOURCE25 for the import of Class Descriptor files	page 4-22

If you're new to SCHEDULE25...

If you're new to SCHEDULE25 and have not built your course master extract and update programs, we recommend that you build bulk interface 4 rather than any of these interfaces. However, if you decide that bulk interface option 1, 2, or 3 is the best interface for your scheduling environment, you must perform the following tasks to build it:

Task	Instructions/guidance
Define your Class Descriptor file using RESOURCE25	page 4-8
Define SCHEDULE25 processing settings using RESOURCE25	page 4-15
Create course master extract and update programs	page 4-20
Prepare RESOURCE25 for the import of Class Descriptor files	page 4-22

Tasks to build bulk interface option 4

Tasks

You must perform the following tasks to build bulk interface option 4:

Task	Instructions/guidance
Define your Class Descriptor file using RESOURCE25	page 4-8
Define SCHEDULE25 processing settings using RESOURCE25	page 4-15
Create course master extract and update programs	page 4-21
Prepare RESOURCE25 for the import of Class Descriptor files	page 4-22
Prepare RESOURCE25 for the import and export of vCalendar files	page 4-38

Importing and maintaining the Control file

Introduction

If you have already created your Control file, we suggest that you import it into RESOURCE25 and maintain it there. This section tells you how to do that.

Instructions

Follow these instructions to import your Control file into RESOURCE25 and modify it.

Step	Action
1	FTP your Control file (*ctrl.dat*) to the *S25/IMPORT* subdirectory of the *R25* installation directory (use ASCII FTP file type).
2	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>3</td><td>Double-click the UAI Products icon.

Result: The UAI Product Options form opens.
  </td></tr>
<tr><td>4</td><td>Choose Actions|Import CTRL.DAT.

Result: The Control file data is imported into RESOURCE25.</td></tr>
<tr><td>5</td><td>Modify the Control file data as needed.
<table>
<tr><th>To modify...</th><th>Do this...</th></tr>
<tr><td>The starting columns of any fields</td><td>1. Select the Starting Columns tab.
2. Edit the starting column numbers as needed.</td></tr>
<tr><td>The widths of any fields</td><td>1. Select the Field Widths tab.
2. Edit the widths as needed.</td></tr>
<tr><td>Your weekday abbreviations</td><td>1. Select the Weekdays tab.
2. Edit the abbreviations as needed.</td></tr>
<tr><td>SCHEDULE25 processing settings</td><td>1. Select the SCHEDULE25 Settings tab.
2. Edit the settings as needed.</td></tr>
</table></td></tr>
</table>

Step	Action
6	Click OK. **Result**: You are returned to the System Definitions and Defaults form. Your Control file data modifications are saved.

Defining your Class Descriptor file

Introduction

If you are new to RESOURCE25 and SCHEDULE25, you must define your Class Descriptor file. This section describes each of the SCHEDULE25 fields you can define using RESOURCE25 and tells you how to define them.

Class Descriptor fields

The table beginning below lists all the Class Descriptor fields you can define using RESOURCE25 and their contents. All of the required fields are used by SCHEDULE25. The optional fields are not used by SCHEDULE25, but can be defined to RESOURCE25.

You may want to define additional fields, such as units of credit or term code. These fields are not used by RESOURCE25 or SCHEDULE25, although they will appear on SCHEDULE25 output reports. They must occupy columns unused by SCHEDULE25 fields.

The R25 Event Key field described below is used by RESOURCE25, and must be defined in the Class Descriptor record. It is not used by SCHEDULE25.

For detailed information about the Class Descriptor file and fields, see the *SCHEDULE25 Data Preparation* manual available on our web site (*www.unival.com*). It is the main reference source for information about the Class Descriptor file.

This Class Descriptor field...	Contains this data...
R25 Event Key (required)	• A unique event identifier generated and used by RESOURCE25 (20 characters fixed length)
Room Name (required)	• Nothing if a room must be assigned by SCHEDULE25 • A valid SCHEDULE25 room name if preassigned

This Class Descriptor field...	Contains this data...
Event Title (Required)	The name of the event
Event Contact (Optional)	The contact associated with the event
Phone Number (Optional)	The phone number of the event contact
Meeting Days (Required)	Day of the week abbreviations indicating the days the class is held
Start Hours (Required)	An integer between 1 and 12 (or 1 and 24 military time) indicating the hour the class begins
Start Minutes (Required)	An integer between 00 and 59 indicating the minute the class begins
Finish Hours (Required)	An integer between 1 and 12 (or 1 and 24 military time) indicating the hour the class ends
Finish Minutes (Required)	An integer between 00 and 59 indicating the minute the class ends
AM/PM Designator (Required)	• "A" if the class ends before noon • "P" if the class ends after noon • "H" to specify when the class ends in military time
Enrollment (Required)	• If known, an integer representing the estimated or actual class enrollment • If not known, the default enrollment indicator (see page 4-15)

This Class Descriptor field...	Contains this data...
Conflict Deciders (Required)	Column designations used to break ties among classes with identical requirements. Use is optional, but they must be defined in the Class Descriptor file.
Department Name (Required)	A valid SCHEDULE25 department key value
Start Month (Required)	An integer between 01 and 12 indicating the month the class begins
Start Day (Required)	An integer between 01 and 31 indicating the day of the month the class begins
Finish Month (Required)	An integer between 01 and 12 indicating the month the class ends
Finish Day (Required)	An integer between 01 and 31 indicating the day of the month the class ends
Start Week (Required)	Not used, but must be defined to satisfy SCHEDULE25 Control file requirements
Finish Week (Required)	Not used, but must be defined to satisfy SCHEDULE25 Control file requirements

Instructions

Follow these instructions to define your Class Descriptor file using RESOURCE25.

Step	Action
1	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 
2	Double-click the UAI Products icon. **Result**: The UAI Product Options form opens. 

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="8">3</td><td colspan="2">Select the Starting Columns tab, and enter the starting column of each Class Descriptor field as specified below:</td></tr>
<tr><td>This field...</td><td>Should have a starting column...</td></tr>
<tr><td>R25 event key</td><td>After column 80, and near the end of the record</td></tr>
<tr><td>Each of these fields...</td><td>Must have a starting column...</td></tr>
<tr><td>Room Name
Meeting Days
Start Hours
Start Minutes
Finish Hours
Finish Minutes
AM/PM Designator
Enrollment
Department Name
Start Week
Finish Week</td><td>Before column 78</td></tr>
<tr><td>Each of these fields...</td><td>Must have a starting column...</td></tr>
<tr><td>Event Title
Event Contact
Phone Number
Start Month
Finish Month
Start Day
Finish Day</td><td>Before column 78, or after column 80</td></tr>
</table>

<table>
<tr><th>Step</th><th colspan="2">Action</th></tr>
<tr><td rowspan="2">3,
cont.</td><td>Each of these fields...</td><td>Can...</td></tr>
<tr><td>Conflict Decider #1 - #5</td><td>• Be inserted programmatically in the file when you run your course master extract, if you’re sending records from your SIS directly to SCHEDULE25.
• Use columns already defined for another field (you can use the same or different columns for each conflict decider).
• Use blank columns, if you don’t want to use conflict deciders.</td></tr>
<tr><td rowspan="8">4</td><td colspan="2">Select the Field Widths tab, and enter the width of each of the listed fields and the entire Class Descriptor record:</td></tr>
<tr><td>Each of these fields...</td><td>Can be...</td></tr>
<tr><td>Room Name
Department Name
Event Title
Event Contact</td><td>Any length - you define

Note: When the event title is imported into RESOURCE25 from SCHEDULE25, the first 40 characters are entered in the Event Name field and the whole title is entered in the Event Title field.</td></tr>
<tr><td>This field...</td><td>Must be...</td></tr>
<tr><td>Meeting Days</td><td>7 - 14 characters</td></tr>
<tr><td>The length of the...</td><td>Must be...</td></tr>
<tr><td>Class Descriptor record</td><td>80 - 253 characters</td></tr>
</table>

Step	Action
5	Select the Weekdays tab, and enter an abbreviation for each day of the week. Make sure the total number of characters you enter for all the day abbreviations equals the length you've defined for the Meeting Days field.
6	Click OK. **Result**: You are returned to the System Definitions and Defaults form. The class descriptor information you've entered is saved.

Defining SCHEDULE25 settings

Introduction

Before you can run SCHEDULE25 in your scheduling environment, you must define several SCHEDULE25 settings that allow you to control certain aspects of the scheduling process. This section describes each of these settings and tells you how to define them.

SCHEDULE25 settings

The table beginning below describes each of the SCHEDULE25 settings. For detailed information about these settings, see the *SCHEDULE25 User's Guide* and the *SCHEDULE25 Data Preparation* manual available on our web site (*www.unival.com*). They are the main reference sources for information about these settings.

This setting...	Defines...	Example
Minimum Fill Ratio	The minimum percentage of seats in a room that must be filled by any SCHEDULE25 room assignment.	0.10 (10%)
Enrollment Adjuster	The percentage by which SCHEDULE25 is to adjust the enrollments appearing on the Class Descriptor records prior to initiating the room assignment process.	0.90 (90%)
Default Enrollment	The value that is to be substituted in the Enrollment field of a Class Descriptor record when SCHEDULE25 encounters a "Default Enrollment Indicator" in the field (see below).	0035
Default Enrollment Indicator	A four-character string that appears in the Enrollment field of a Class Descriptor record when the enrollment data for the class is unavailable in the SIS.	****
Minimum Correct Input Fraction	The minimum percentage of Class Descriptor records that must be correct before SCHEDULE25 will perform its room assignment process.	1.00 (100%)

This setting...	Defines...	Example
Number of Weeks in Term	The maximum span of weeks represented by the classes in the Class Descriptor file.	16
Mini-Classes?	Whether or not all classes in the Class Descriptor file are the same duration (as defined by the Number of Weeks in Term setting above). If set to "NO," SCHEDULE25 ignores the Start Week/Finish Week fields and any date fields, and assumes that all classes are of equal duration. If set to "DATES," SCHEDULE25 uses the Begin/End Date fields to determine the duration of room assignments.	DATES
Extra Finish Minutes	The number of minutes SCHEDULE25 must add internally to the finish time of each class when making room assignments.	10
Earliest Start Time	The earliest valid class start time. SCHEDULE25 writes an error message if a Class Descriptor record contains an earlier start time than this.	0700
Latest Finish Time	The latest valid class finish time. SCHEDULE25 writes an error message if a Class Descriptor record contains a later finish time than this.	2300
Sum Enrollments For Crosslisted Classes	How SCHEDULE25 treats enrollments for cross-listed classes. If set to "YES," SCHEDULE25 adds the enrollments of each Class Descriptor record in a cross-listed group before initiating the room assignment process and only assigns a room that is large enough to hold the combined enrollments. If set to "NO," SCHEDULE25 assigns a room large enough to hold the enrollment of the NSM record of the cross-listed group.	YES

Instructions

Follow these instructions to define the SCHEDULE25 settings.

Step	Action
1	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 
2	Double-click the UAI Products icon. **Result**: The UAI Product Options form opens. 

Step	Action	
3	Select the SCHEDULE25 Settings tab, and enter the appropriate settings for your scheduling environment:	
	This setting...	**Can be set to....**
	Minimum Fill Ratio	Any number between 0.01 and 1.00.
	Enrollment Adjuster	Any number between 0.01 and 1.00 Set to 1.00 for no adjustment.
	Default Enrollment	Any whole number
	Default Enrollment Indicator	Any 4 characters
	Minimum Correct Input Fraction	Any number between 0.00 and 1.00
	Number of Weeks in Term	Any whole number greater than 0 but less than or equal to 54
	Mini-Classes?	NO or DATES **Warning**: *Don't set to YES.*
	Extra Finish Minutes	Any whole number from 0 to 30
	Earliest Start Time	Any military time designation 0001 - 2400. Must be earlier than Latest Finish Time.
	Latest Finish Time	Any military time designation 0001 - 2400. Must be later than Earliest Start Time.
	Sum Enrollments For Crosslisted Classes	YES or NO

Step	Action
4	Click OK. **Result**: You are returned to the System Definitions and Defaults form. The SCHEDULE25 settings you've entered are saved.

Creating course master extract and update programs

If you're building bulk interface 1, 2, or 3...

If you've decided to build bulk interface option 1, 2, or 3 and are new to RESOURCE25 and SCHEDULE25, you must develop two programs:

- **A course master extract program** that extracts data from the course master file in your SIS and writes it to a Class Descriptor file.
- **A course master update program** that updates your course master file with room assignments and any other changes made using SCHEDULE25 and RESOURCE25.

Your extract program must:

1. Read class records in the course master file, determine which of them must be extracted, and extract them.
2. Write the extracted records to a Class Descriptor file in the file format defined by your institution.
3. Attach the appropriate assignment code to each record in the Class Descriptor file.

Your update program must:

1. Read records from the SCHEDULE25 *sortrm.dat* output file or a Class Descriptor file exported from RESOURCE25.
2. Find the matching records in the course master file.
3. Update the course master records.

The information you need to prepare your course master extract and update programs is found in "Extracting Course Master Data" (chapter 5) and "Updating Course Master Data" (chapter 6) of the *SCHEDULE25 Data Preparation* manual available on our web site (*www.unival.com*). Follow the guidance and instructions in that manual to prepare your extract and update programs.

If you're building bulk interface 4...

If you've decided to build bulk interface option 4, you must develop two programs:

- **A course master extract program** that extracts data from the course master file in your SIS and writes it to a vCalendar file.
- **A course master update program** that updates your course master file with room assignments made using SCHEDULE25 and RESOURCE25.

Your extract program must:

1. Read class records in the course master file, determine which of them must be extracted, and extract them.
2. Write the extracted records to a vCalendar file.
3. Put the vCalendar file in the Request directory.

Your update program must:

1. Retrieve vCalendar files generated by RESOURCE25.
2. Parse the files to identify the information for each class.
3. Find the corresponding class records in the course master file.
4. Update the course master records.

The information you need to prepare your course master extract and update programs is found in chapter 7 of this manual.

Preparing for Class Descriptor file import

Process Steps

There are two process steps you must perform to prepare to import Class Descriptor files into RESOURCE25:

1. Identify where you want imported class events to be placed in your RESOURCE25 event structure.
2. Create an import profile.

Step 1: Identify where imported events should be placed

The first thing you must do to prepare to import Class Descriptor files is determine where you want imported events to be placed in your RESOURCE25 event structure.

Placement options

You must choose one of these options:

- To have all the imported events placed in a single location in your event structure. *This is generally not recommended.*
- To have imported events placed in different locations in your event structure based on the interface tag encountered in each "destination event." Destination events are events you want imported events to be placed directly under in your event structure.

Interface tags

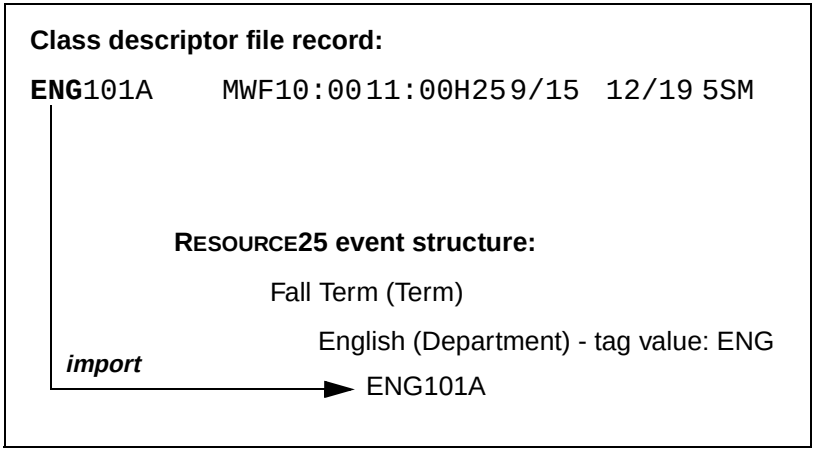
Interface tags are used to identify destination events in your RESOURCE25 event structure. They are composed of one or both of these:

- A tag value
- A tag prefix and/or suffix

Interface tag value

An interface tag value contains a character string from your Class Descriptor file records that identifies a destination event in your RESOURCE25 event structure. *All destination events must have an interface tag value.*

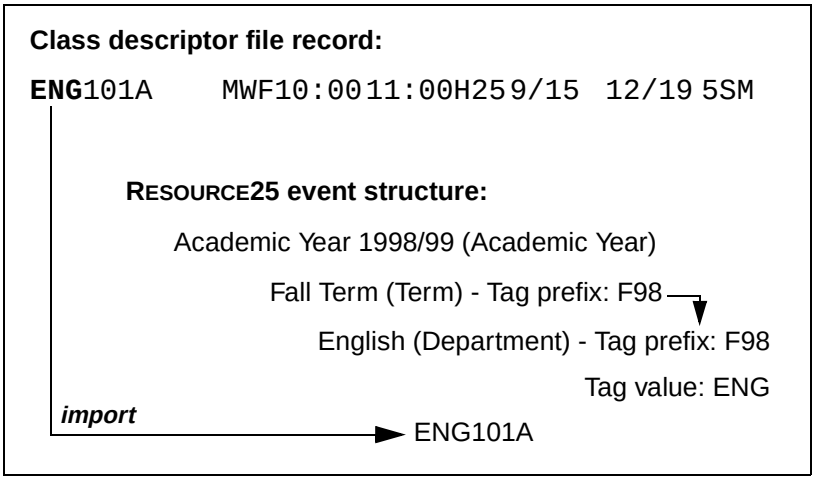
For example, if we want all English classes placed under the English department in our event structure, we would enter the character string used to identify the English department (ENG in our example) as the interface tag value for the English department event as shown here:



Tag prefixes and suffixes

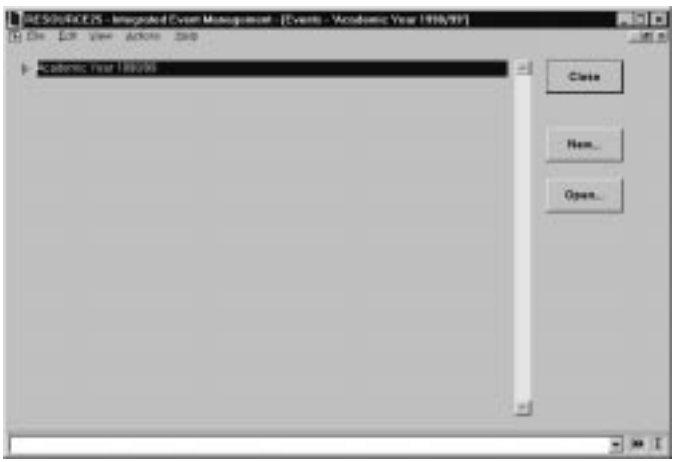
An interface tag prefix or suffix must be used if more than one destination event in your RESOURCE25 event structure contains the same interface tag value. Prefixes and suffixes are static labels that are prepended or appended to a tag value. They are defined in the import profile (see page 4-33).

For example, if both our fall and spring term English department events have a tag value of ENG, we could specify a tag prefix of "F98" for our Fall Term 1998 event and have that prefix inherited by all department events below it. This would direct our incoming ENG101A class to the fall term English department area of our event structure as shown here:



Instructions for entering event interface tags

Follow these instructions to enter interface tags for destination and other events in your event structure:

Step	Action
1	From the RESOURCE25 main menu, choose Event|Event Cabinets. **Result**: The Event Cabinets form opens. 
2	Double-click the cabinet that contains the event you want to add an interface tag for. **Result**: The selected cabinet opens on the Events form. 

Step	Action
3	Click the expand arrows to the left of the event names until you locate the event you want to add an interface tag for, and double-click the event. **Result**: The selected event opens on the Edit Event form.
4	Select the Interface tab. You may have to click the "more" arrow ▼ if you can't see the tab. **Result**: The Interface page of the Edit Event form opens.
5	Enter the interface tag value and/or prefix and/or suffix for the event. In this example, we've entered the tag prefix F98 for the Fall Term 1998 event. 
6	Click OK. **Result**: You are returned to the Events form.

Step	Action
7	Repeat steps 3 - 6 for other events as needed. In this example, we've entered ENG as the tag value for the English department event. Notice that the F98 prefix was inherited from the Fall Term 1998 event and can't be changed. It was prepended to the ENG tag value.  **Note**: *The angle brackets around the concatenated tag value are used to aid readability on this form. They do not exist in the Class Descriptor file.*

Step 2: Create an "import profile" of the incoming events

After you've created interface tags to identify destination events in your event structure, you're ready to build import profiles.

Import profiles

An import profile provides information to RESOURCE25 about the nature of the events you plan to import and how to handle the events, including:

- Where the incoming event file is located.
- The year the incoming events apply to.
- How you want to populate certain fields in the RESOURCE25 event record including: event state, enrollment, contact name, and phone number.
- Whether all of the imported events should be placed in one location in your RESOURCE25 event structure or in different locations based on interface tags. If based on interface tag, the location of the character string in the incoming file that identifies the interface tag value and any prefix and/or suffix (see below and next page).
- What to do with events that can't be routed into your RESOURCE25 event structure.
- What to do with incoming related and cross-listed events.
- Who should be assigned as the owner of each incoming event.

How interface tags are used in placing incoming events

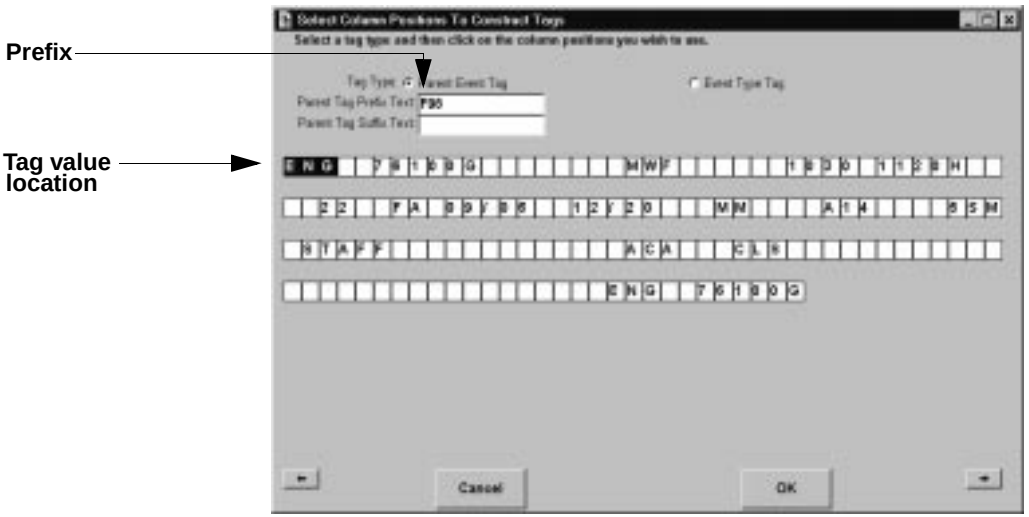
If you plan to use interface tags, you must identify where in each import file event record the interface tag value is located. This is part of building your import profile.

If you're using a prefix and/or suffix, you must also indicate that in your import profile, so RESOURCE25 places incoming events in the correct area of your event structure. The tag prefix or suffix you specify in the profile is in effect for the duration of the import.

Example

This example shows how we've specified our tag value location and prefix to RESOURCE25. This form is part of the wizard used to build the import profile.

In this example, we're telling RESOURCE25 to read the first three characters of each incoming import file record and use that as the tag value to place the incoming class event in the right department. We're also telling RESOURCE25 to place the incoming class event under the appropriate (based on tag value) fall term 1998 (F98) department event in our event structure.



This means that if RESOURCE25 reads "ENG" in the first three characters of an incoming event record and <F98>-<ENG> as the interface tag of the RESOURCE25 English department event record, it will place the incoming class event under the fall term English department event in our event structure.

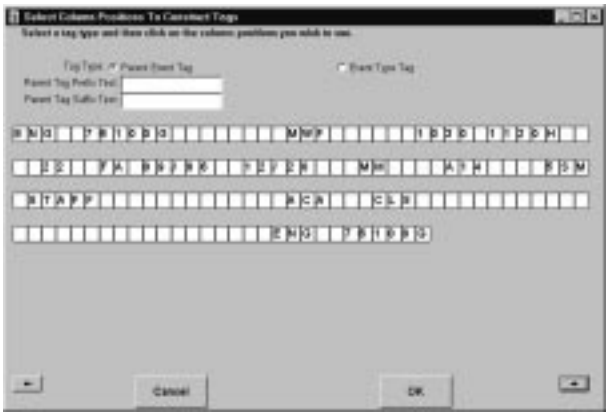
Instructions for building an import profile

Follow these instructions to build an import profile.
Note: *Once you build a profile, you can modify it and reuse it as needed.*

Step	Action
1	From the RESOURCE25 main menu, choose Administration|Import|Event Import Profiles. **Result**: The Event Import Profile Master List form opens. 
2	Enter a name for your import profile and click New. **Result**: The File Name and Data Format Specification wizard form opens. 

Step	Action
3	On this import wizard form: • Select "SCHEDULE25 class descriptor file" as the Import Data Format. • Enter the full pathname of the import file (or browse to locate and select). • Use the up/down arrows to select the year the import file applies to. • Select the RESOURCE25 event state you want assigned to the imported events from the drop down list. • Select whether you want the event enrollment copied into the RESOURCE25 expected head count field only or into the expected *and* registered head count fields.
4	Click Next. **Result**: The Event Insertion Rule wizard form opens. 
5	Select one of these import destination options: • To have all the events in the import file placed under the same RESOURCE25 event in your event structure (single destination option). If you choose this option, continue to step 6 of this procedure. • To use specific column positions in the import file to determine the destinations (interface tag option). If you choose this option, skip to step 8 of this procedure.

Step	Action
6	If you've chosen the single destination option: • Click Next. *The Specify the Destination Event wizard form opens.*  • Use the scroll bar to locate the destination event. • Select the destination event. • If the Use This Event Type drop down list is active, select the event type of the destination event.
7	• Click Next. **Result**: The Importing Related and Cross-Listed Classes wizard form opens.  • Skip to step 13 of this procedure.

Step	Action
8	If you've chosen the interface tag option: • Click Assign Column Tags. *The Select Column Positions to Construct Tags wizard form opens.*  • Select the Parent Event Tag option button. • Enter the tag prefix and/or suffix, if this is part of the interface tag of the destination events. • Select the column positions in your import file that identify the interface tag value of your destination events. • If the parent event type has more than one child event type, select the Event Type Tag option button, and select the column positions in your import file that identify that event type (the columns you use to define a class's instruction type). **Note**: *This character string must also be entered in the Interface field on the Event Type Master List form for this event type.*

Step	Action
9	Click OK. **Result**: The Handling Imported Events That Cannot be Routed wizard form opens. 
10	Select one of these options for dealing with events that can't be routed based on interface tag data: • To discard the events and write an error message. *This is the recommended option.* If you choose this option, continue to step 11 of this procedure. • To place the events under one event in your event structure. If you choose this option, skip to step 12 of this procedure.
11	If you've chosen to discard events that can't be routed, click Next. **Result**: The Importing Related and Cross-Listed Classes wizard form opens. See the form sample on page 4-32. Skip to step 13 of this procedure.

Step	Action
12	If you've chosen to place events that can't be routed under one destination event in your event structure: • Click Select Destination. *The Specify Destination Event wizard form opens. See the form sample on page* 4-32. • Use the scroll bar to locate the destination event. • Select the destination event. • If the Use This Event Type drop down list is active, select the event type of the destination event. • Click Next. *The Importing Related and Cross-Listed Classes wizard form opens. See the form sample on page* 4-32. • Continue to step 13.
13	On this import wizard form, you select how you want RESOURCE25 to treat SCHEDULE25 assignment codes for related and cross-listed classes it may encounter in the import file. Select what you want the RESOURCE25 import function to do with "related" (RSM) class records: • Treat them as additional meeting patterns of one event (*recommended option*). • Treat them as separate events. Select what you want the RESOURCE25 import function to do with cross-listed (WSM/VSM/WRM) class records: • Ignore them. • Treat them as one event and show their cross-listed names in the event title (*recommended option*). **Note**: *See chapter 5 for information on how RESOURCE25 imports events based on assignment codes.*

Step	Action
14	Click Next. **Result**: The Event Ownership Selection wizard form opens. 
15	Select who should own each imported event: • The owner of the event's parent event. *This is the recommended option.* • The person executing the import function. If you choose this, also indicate if you want object-level security to be enforced during the import. **Note**: *If you elect to have object-level security enforced during the import, some events may not be imported even if they have a valid destination, because of object-level security restrictions on those events. For example, several departments could import the same Class Descriptor file, and have only "their" events imported.*

Step	Action
16	Click Next. **Result**: The Migrating Contact Names and Phone Numbers wizard form opens. 
17	Select how you want contact information in the import file processed: • Ignore it. • Create the contacts in RESOURCE25. If you select this option, also indicate if you want the contact phone numbers in the import file to be added as the contact work phone numbers in RESOURCE25 (if there is no work phone number currently). Also, select the role you'd like the contact to have from the drop down list. **Note**: *The contact names are not parsed. The entire name is placed in the last name field of the RESOURCE25 contact record.*
18	Click Finish. **Result**: You are returned to the Event Import Profile Master List form where your new import profile is listed.

Preparing for vCalendar file import and export

Introduction

If you are building bulk interface option 4, you must also prepare RESOURCE25 for the import and export of vCalendar files by:

- Selecting vCalendar processing options using the vCalendar system definition.
- Specifying vCalendar directory and script locations using the vCalendar system definition.
- Creating vCalendar filter rules.

Read chapter 7

Information and complete instructions for performing each of these tasks is in chapter 7. *Read that chapter carefully, and follow the instructions you find there.*

5 Using Your Bulk Interface

Introduction

What's in this chapter

This chapter tells you which RESOURCE25 import and/or export tasks apply to each of the bulk interfaces described in chapter 3, and how to do the tasks. These are tasks you'll perform on an ongoing basis once you begin using your bulk interface.

The chapter tells you how to:

- Export Control file data
- Import class events in class descriptor data format
- Export class events in class descriptor data format
- Import class events in vCalendar data format

It also explains how the vCalendar export process is initiated.

.

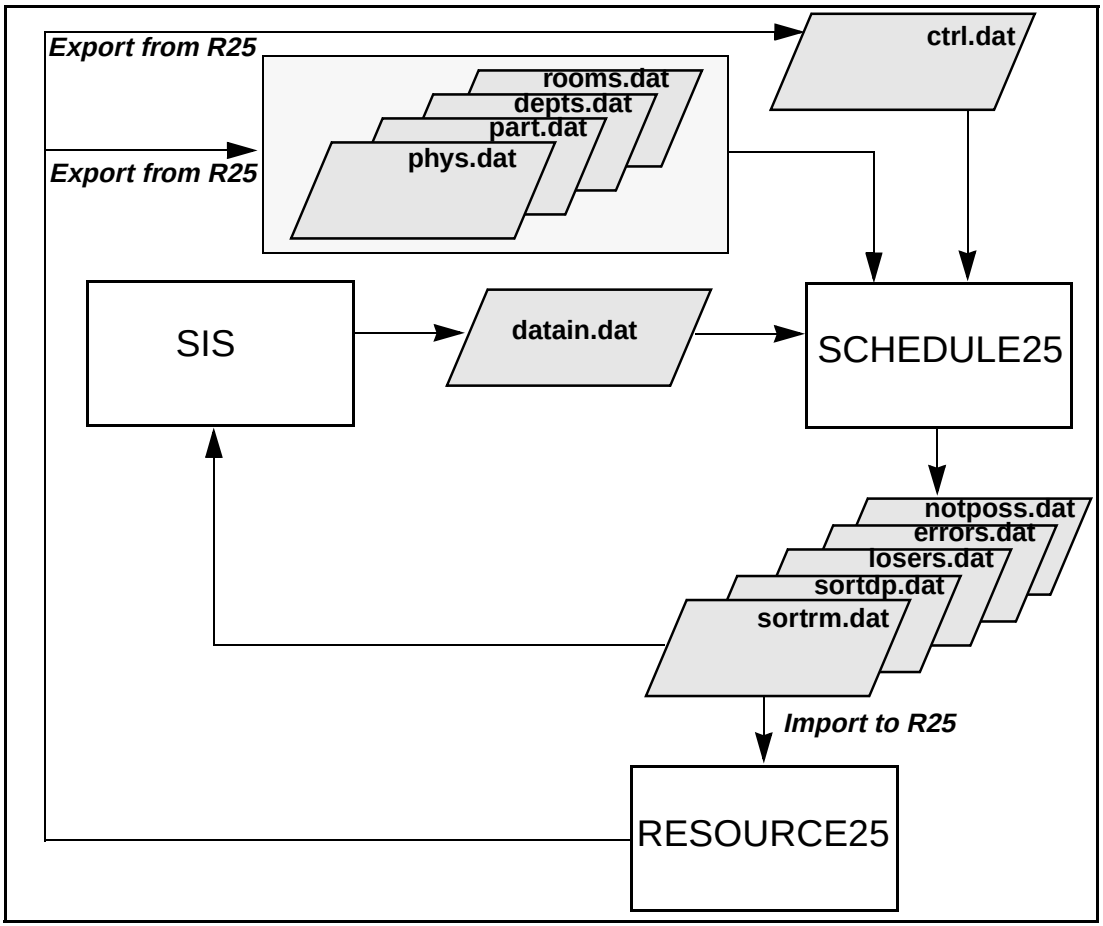
Chapter contents

Tasks required for bulk interface option 1

Task diagram

As shown in this diagram, to use bulk interface option 1, you must:

- Export your *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat* files.
- Export your Control file data.
- Import the *sortrm.dat* file.



Where to find instructions

This table tells where to find instructions and guidance in performing the tasks associated with bulk interface option 1:

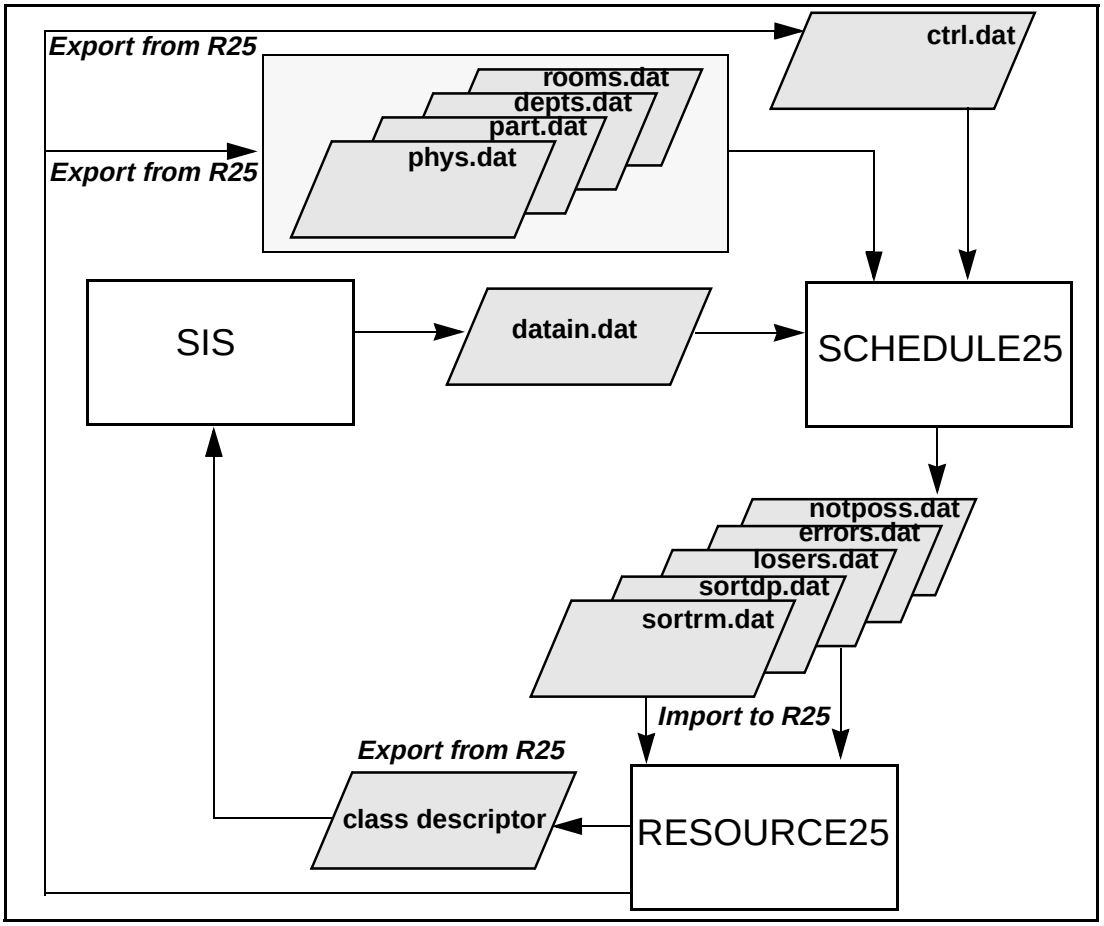
Instructions for this task...	Are here...
Exporting *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat*	*RESOURCE25 Data Preparation 1: Space and Customers*, chapter 5 (use InfoFinder to access)
Exporting Control file data	This manual, page 5-11
Importing *sortrm.dat*	This manual, page 5-13

Tasks required for bulk interface option 2

Task diagram

As shown in this diagram, to use bulk interface option 2, you must:

- Export your *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat* files.
- Export your Control file data.
- Import the *sortrm.dat* and *losers.dat* files.
- Export a Class Descriptor file.



Where to find instructions

This table tells where to find instructions and guidance in performing the tasks associated with bulk interface option 2:

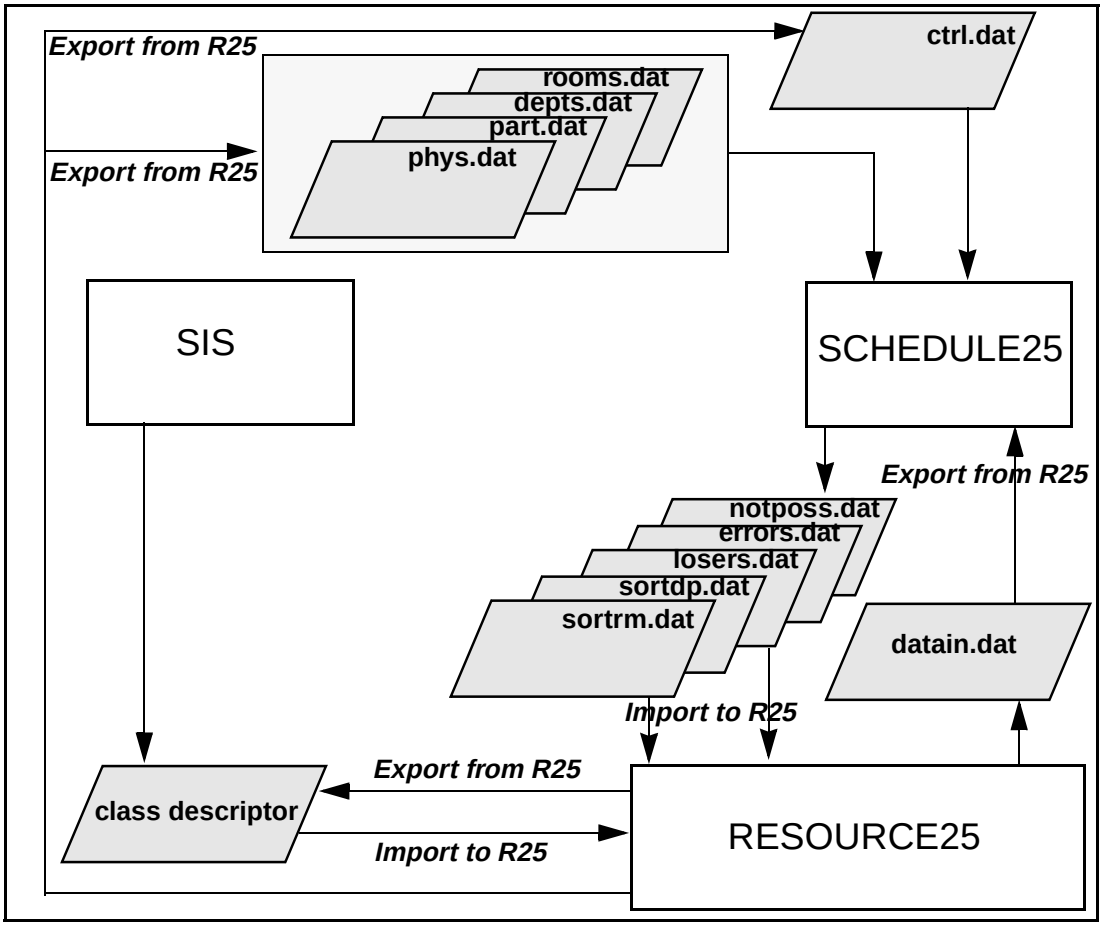
Instructions for this task...	Are here...
Exporting *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat*	*RESOURCE25 Data Preparation 1: Space and Customers*, chapter 5 (use InfoFinder to access)
Exporting Control file data	This manual, page 5-11
Importing *sortrm.dat* and *losers.dat*	This manual, page 5-13
Exporting a Class Descriptor file	This manual, page 5-17

Tasks required for bulk interface option 3

Task diagram

As shown in this diagram, to use bulk interface option 3, you must:

- Export your *phys.dat, part.dat, depts.dat,* and *rooms.dat* files.
- Export your Control file data.
- Import *sortrm.dat* and *losers.dat* files and a Class Descriptor file.
- Export the *datain.dat* file and a Class Descriptor file.



Where to find instructions

This table tells where to find instructions and guidance in performing the tasks associated with bulk interface option 3:

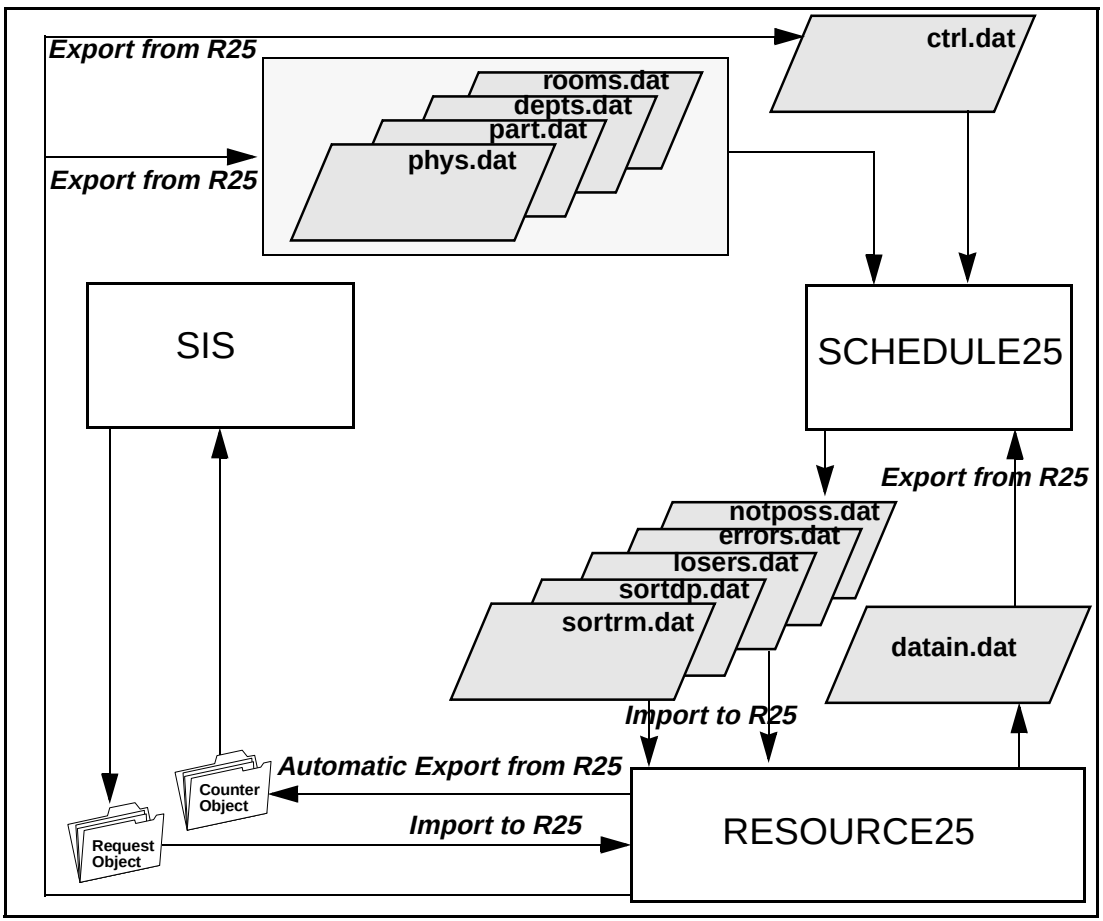
Instructions for this task...	Are here...
Exporting *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat*	*RESOURCE25 Data Preparation 1: Space and Customers*, chapter 5 (use InfoFinder to access)
Exporting Control file data	This manual, page 5-11
Importing *sortrm.dat*, *losers.dat,* and a Class Descriptor file	This manual, page 5-13
Exporting the *datain.dat* file and a Class Descriptor file	This manual, page 5-17

Tasks required for bulk interface option 4

Task diagram

As shown in this diagram, to use bulk interface option 4, you must:

- Export your *phys.dat, part.dat, depts.dat,* and *rooms.dat* files.
- Export your Control file data.
- Import a vCalendar request file.
- Import the *sortrm.dat* and *losers.dat* files
- Export the *datain.dat* file.



Where to find instructions

This table tells where to find instructions and guidance in performing the tasks associated with bulk interface option 4:

Instructions for this task...	Are here...
Exporting *phys.dat*, *part.dat*, *depts.dat*, and *rooms.dat*	*RESOURCE25 Data Preparation 1: Space and Customers*, chapter 5 (use InfoFinder to access)
Exporting Control file data	This manual, page 5-11
Importing *sortrm.dat* and *losers.dat*	This manual, page 5-13
Exporting the *datain.dat* file	This manual, page 5-17
Importing a vCalendar request file	This manual, page 5-22

Exporting Control file data

Instructions

Prior to each SCHEDULE25 run, you should export Control file data. These instructions tell you how to do that.

Step	Action
1	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 
2	Double-click the UAI Products icon. **Result**: The UAI Product Options form opens. 

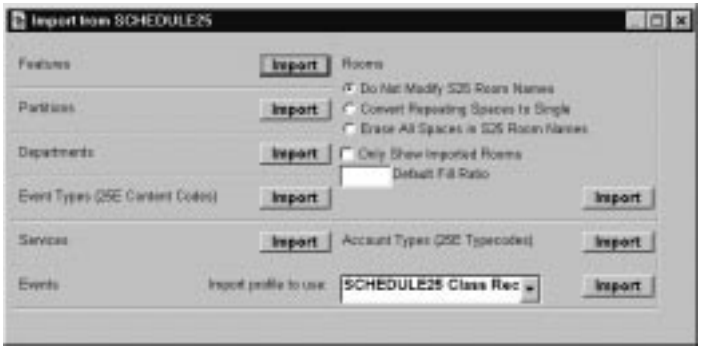
Step	Action
3	Select each tab on this form in turn, and make sure the class descriptor information and SCHEDULE25 settings are correct for this SCHEDULE25 run. Update information as needed.
4	Choose Actions|Export CTRL.DAT. **Result**: A Control file with the information you've exported is created in the RESOURCE25 Export directory.

Importing Class Descriptor files

Instructions

Follow these instructions to import a class event file in class descriptor data format.

Step	Action
1	Decide which import profile you want to use for this import and whether or not you need to modify the profile.
2	If you don't need to modify the import profile, skip to step 3. If you need to modify the import profile: • From the RESOURCE25 main menu, choose Interface|Import|Event Import Profiles *The Event Import Profile Master List form opens.* • Select the profile you want to modify and click Open. *The first event profile wizard form opens.* • Follow the instructions beginning with step 3 on page 4-31 to modify the profile as needed. • When you have completed the import wizard and are returned to the Event Import Profile Master List form, click OK.

Step	Action
3	From the RESOURCE25 main menu, choose Interface|Import|Import SCHEDULE25/25E Files. **Result**: The Import From SCHEDULE25 form opens. 
4	In the Events area of the form, select the import profile you want to use for this import from the drop-down list.
5	Click the Import button to the right of the import profile drop-down list. **Result**: The events are imported into RESOURCE25 and correctly placed in your RESOURCE25 event structure. **Note**: *Any errors appear on an errors form immediately after the import. Starting a new import clears the previous contents of the errors form.*

How RESOURCE25 handles SCHEDULE25 assignment codes

Your event import file must contain a SCHEDULE25 assignment code in columns 78-80 of each event record. If you are currently using SCHEDULE25, you are probably familiar with these codes and their meanings. If you are not, please see the *SCHEDULE25 Data Preparation* manual for detailed information about these codes.

This table describes how RESOURCE25 imports class event records based on assignment code:

If the import record has this code...	RESOURCE25...
NSM	Imports it as a new event without a room assignment.
1SM	Imports it as a new event without a room assignment. The room preference is ignored.
WSM	Ignores the record or appends its meeting pattern to the previously imported NSM event depending on what you specified in your import profile. See chapter 4.
RSM	Imports it as a new event or appends its meeting pattern to the previously imported NSM event depending on what you specified in your import profile. See chapter 4.
ASM	Imports it as a new event and attempts to assign the specified room. If it can't assign the room, reports an error on the errors form.
5SM	Imports it as a new event and attempts to assign the specified room. If it can't assign the room, reports an error on the errors form.
WRM	Ignores the record.
NXM	Imports it as a new event without a room assignment.
RXM	Imports it as a new event or appends its meeting pattern to the previously imported NXM event depending on what you specified in your import profile. See chapter 4.
1XM	Imports it as a new event without a room assignment. The room preference is ignored.

If the import record has this code...	RESOURCE25...
AXM	Imports it as a new event and attempts to assign the specified room. If it can't assign the room, reports an error on the errors form.
5XM	Imports it as a new event and attempts to assign the specified room. If it can't assign the room, reports an error on the errors form.
HSM	Imports it as a new event and attempts to assign the specified room. If it can't assign the room, reports an error on the errors form.
VSM	Ignores the record or appends its meeting pattern to the previously imported HSM event depending on what you specified in your import profile. See chapter 4.

Exporting Class Descriptor files

Process

There are two major steps you must perform to export class events in class descriptor data format:

1. Use a saved event search to identify the events you want to export.
2. Export the events.

Export rules

Each event you elect to export must meet the following criteria:

- It must have at least one reservation definition.
- If it has a room assignment, the room must be defined in RESOURCE25 as a SCHEDULE25 room in a SCHEDULE25 partition.
- The account assigned to it must be defined in RESOURCE25 as a SCHEDULE25 department.

Instructions for exporting class events

Follow these instructions to export class events in class descriptor data format:

Step	Action
1	From the RESOURCE25 main menu, choose Interface|Export|Export SCHEDULE25 files. **Result**: The Export to SCHEDULE25 form opens. 
2	In the Class Descriptor File area of the form, select Events. **Result**: The Based on Saved Search drop down list becomes active.

Step	Action
3	If you know the saved event search results you want to export and you don't want to modify the search: • Select the search name in the drop down list. • Skip to step 7. If you know the saved search results you want to export but need to modify the search: • Select the search name in the drop down list. • Click the "Details" button to the right of the list. *The Event Search form opens with the saved search criteria entered/selected.* • Continue to step 4. If you want to create a new event search: • Select (none) from the drop down list. • Click the "Details" button to the right of the list. *The Event Search form opens.* • Continue to step 4.

Step	Action
4	Enter or modify the search criteria so that the search results will list the class events you want to export. Click Search! **Result**: The results of the search are listed.  In this example, we set up our search criteria to list all events with a “Section” event type that have start and end dates between September 1, 1998 and December 31, 1998. This allows us to export our fall term 1998 class events.
5	Click Close. **Result**: A dialog opens asking you to confirm your changes to the search.
6	Click Yes. **Result**: If this is a new search, the Save Search as form opens. Enter a name for your search, select the check box if you want to see the search name on your RESOURCE25 Explorer desktop, and click OK. **Result**: You are returned to the Export to SCHEDULE25 form.

Step	Action
7	Click Export. **Result**: You are prompted for the location you want for the exported file. Enter (or browse to locate and select) the full pathname to the location.
8	Click OK. **Result**: A Class Descriptor file is created and exported to the location you specified.
9	To view the exported Class Descriptor file, double-click the file name and select the application you want to use to view the file.

Importing vCalendar files

Introduction

When you're ready to import vCalendar files, you must:

1. Specify the import "cycle time," that is, how often you want RESOURCE25 to poll the vCalendar import directories for vCalendar files coming from your SIS.
2. Specify the "default organizer" to be used for imported events that don't have an ATTENDEE property.
3. Start the process when you're ready to begin importing.

Instructions

Follow these instructions to initiate the vCalendar file import process:

Step	Action
1	From the RESOURCE25 main menu, choose Interface|Import|Import vCalendar files. **Result**: The Import vCalendar Files form opens. 

Step	Action
2	Enter or use the up/down arrows to select how often (in minutes) you want RESOURCE25 to check the vCalendar import directories for vCalendar files.
3	Select a default organizer name from the drop down list. This list includes all active RESOURCE25 users.
4	When you're ready to begin importing events, click Start! **Result**: The import function immediately checks the directories you've specified for request, cancel, and declinecounter objects. If it finds vCalendar files, it imports the associated events and displays their names in the Log area of the form as shown in this example.  If it cannot import an event, the event remains in the vCalendar file and the problem is reported in the Log area of the form. RESOURCE25 attempts to import the event again the next time the import process runs. **Note**: *While the import is running, you can't perform any other functions in RESOURCE25. You can't close this form until you have stopped the import process.*
5	When you are done importing events, click Stop! **Result**: RESOURCE25 stops polling the import directories. Importing stops.

How the vCalendar export process is initiated

What initiates the export

What triggers RESOURCE25 to initiate the export of vCalendar files containing reply or counter objects depends on which "To Do Generated" option you've selected on the vCalendar Processing Options form (see page 7-33) and requires that you've set the "Transaction Cycle" option to "Maintain."

If you've selected:

- "Always," RESOURCE25 initiates the export process after the assigned user completes the To Do item.
- "Only If Action Required" and no action is required, RESOURCE25 initiates the export process immediately after the import.
- "Only If Action Required" and action is required, RESOURCE25 initiates the export process after the assigned user completes the To Do item.
- "Never," RESOURCE25 prompts the assigned user to generate a vCalendar reply or counter when the user changes the event's state to "confirmed."

Export timing

When you process a semester or term's worth of events, keep in mind that RESOURCE25 won't send a reply back to your SIS until any required To Do's are completed. This could take several hours or even days, depending on your scheduling environment.

Transaction-based Event Data Interface

6 The Transaction-based Event Data Interface

Introduction

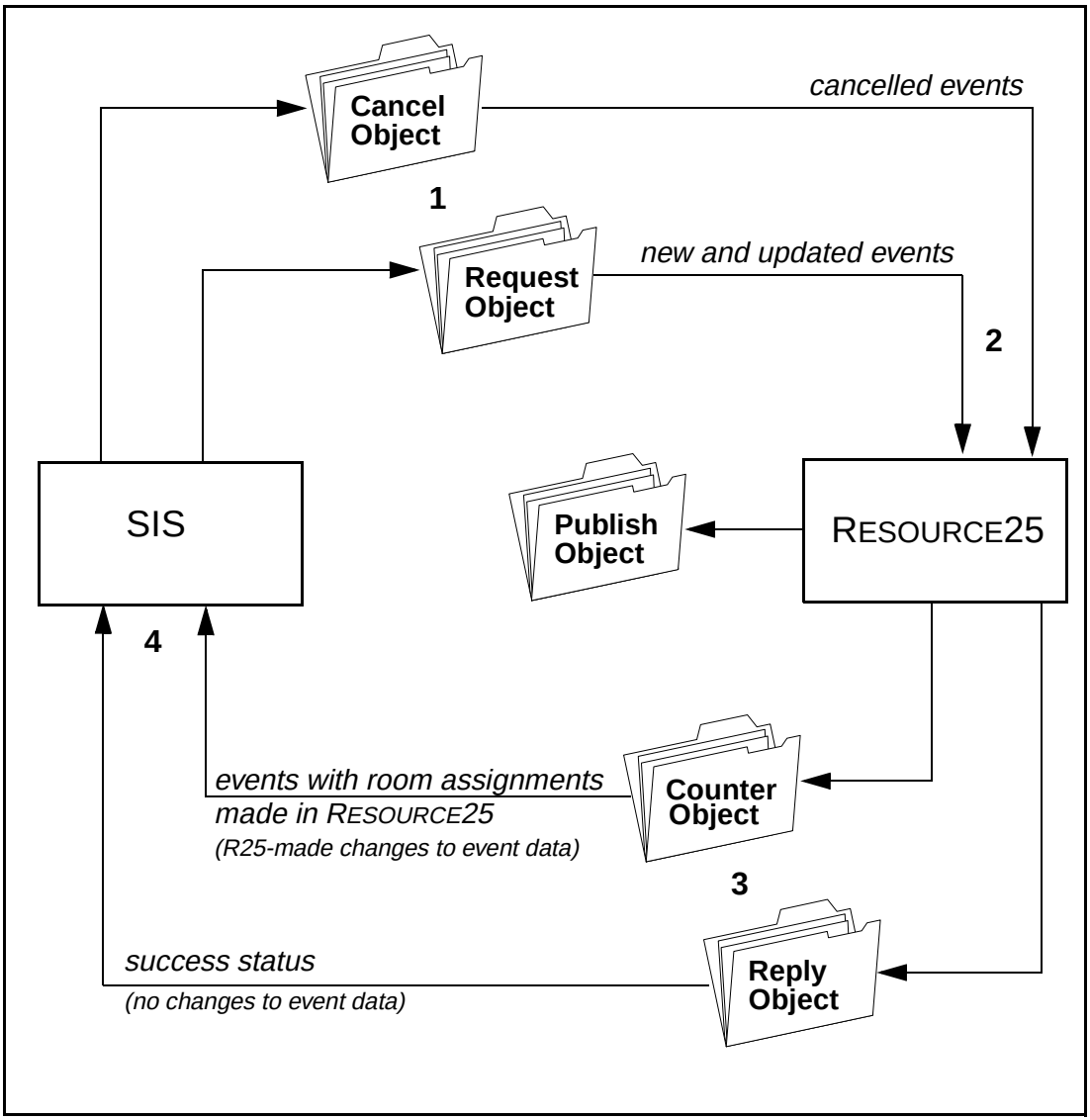
What's in this chapter

This chapter illustrates and explains in detail the transaction-based event data interface summarized in chapter 1. It also tells you the implementation requirements of this interface.

Chapter contents

Interface diagram and description

Interface Diagram This is a diagram of the transaction-based interface:



Description of interface

In this interface:

1. Your SIS sends either one or more new and/or changed events in a vCalendar "request object" or one or more cancelled events in a "cancel object" to RESOURCE25 for processing.

2. Periodically, RESOURCE25 checks for the presence of a vCalendar file coming from your SIS. If it finds one, it imports it for processing.

3. If RESOURCE25 finds data errors in the incoming vCalendar file, it does not update the RESOURCE25 database and sends the unchanged vCalendar file to an error directory.

   If RESOURCE25 doesn't find any data errors, it processes each event record and sends a vCalendar file to your SIS. If the events in the incoming file had room assignments, RESOURCE25 returns a "reply object" with a success status. If the events in the incoming file didn't have room assignments, room assignments are made in RESOURCE25, and then RESOURCE25 returns a "counter object" with the suggested assignments. The next section, "How RESOURCE25 processes imported events," explains in detail how RESOURCE25 handles each incoming event.

   RESOURCE25 also exports a "publish object" containing new and modified events. See Appendix B for information.

4. Your course master update program updates the corresponding records in your SIS with room assignments made in RESOURCE25.

How RESOURCE25 processes imported events

Processing options

The vCalendar system definition allows you to select how you want RESOURCE25 to process event records on import. You can specify:

Who can edit the vCalendar data in imported events:

- **No one** - The vCalendar data in imported events can't be edited.
- **Event Organizer only** - The contact specified as the event organizer for the import can edit the vCalendar data in imported events.
- **R25 Permissions** - The user(s) who have the appropriate security to edit an imported event can edit the vCalendar data in that event.

Note: *Non-vCalendar data is not subject to these editing restrictions.*

How cancelled events should be handled:

- **Change State** - The event's state is set to cancelled and all space reservations for the event are deleted.
- **Delete Event** - The event is deleted from RESOURCE25.

Whether or not RESOURCE25 should respond to your SIS:

- **Break** - RESOURCE25 won't respond. The transaction cycle between RESOURCE25 and the SIS is broken.
- **Maintain** - RESOURCE25 will respond. The transaction cycle between RESOURCE25 and the SIS is maintained.

Whether or not a To Do should be generated for each imported event:

- **Always** - The event organizer will receive a To Do for every imported event.
- **Only If Action Required** - The event organizer will receive a To Do for an imported event only if action is required.
- **Never** - No To Do's will be issued for imported events.

Instructions for setting these options are in chapter 7.

How imported event records are processed

This table explains how RESOURCE25 processes incoming event records when the transaction cycle is set to "maintain" (see previous page).

If the event record is...	RESOURCE25...
New and contains all shared scheduling data including a room assignment	1. Creates the event record in the RESOURCE25 database. 2. Checks the availability of the assigned room. 3. If the room is available, assigns the room to the event, even if the head count exceeds the room's capacity in the required layout. 4. Generates To Dos if required based on the vCalendar To Do processing option selected: • If "Always" is selected and the room is available, generates an informational To Do for the event organizer to alert him/her of the new event. If the head count exceeds the room's capacity, that information is also included in the To Do. When that person completes the To Do, generates a success reply. • If "Always" or "Only If Action Required" is selected and the room is not available, generates an assign room To Do for the event organizer. When that person completes the To Do, generates a counter message containing the suggested room. • If "Never" is selected and the room is available, generates a success reply. • If "Never" is selected and the room is not available, the event organizer or appropriate user must search for events without rooms, assign a room to this event, and change the event state to "confirmed." RESOURCE25 then asks if it should issue a counter message. If the person says "yes," RESOURCE25 issues a counter message containing the suggested room. If the person says "no," RESOURCE25 doesn't issue a counter message.

If the event record is...	RESOURCE25...
New and contains all shared scheduling data except a room assignment	1. Creates the event record in the RESOURCE25 database. 2. Generates To Dos based on the vCalendar To Do processing option selected: • If “Always” or “Only If Action Required” is selected, generates an assign room To Do for the event organizer. When that person completes the To Do, generates a counter message containing the suggested room. • If “Never” is selected, the event organizer or appropriate user must search for events without rooms, assign a room to this event, and change the event state to “confirmed.” RESOURCE25 then asks if it should issue a counter message. If the person says “yes,” RESOURCE25 issues a counter message containing the suggested room. If the person says “no,” RESOURCE25 doesn’t issue a counter message.
Changed and the change is to something other than the RESOURCE25 event type, room assignment, date/time, or head count	1. Updates the event record in the RESOURCE25 database. 2. Generates To Dos if required based on the vCalendar To Do processing option selected: • If “Always” is selected, generates an informational To Do for the event organizer to alert him/her of the changes to the event. When that person completes the To Do, generates a success reply. • If “Only If Action Required” or “Never” is selected, generates a success reply.
Changed and the change is to the RESOURCE25 event type	1. Does not change the event type of the event. (Changes to event type are prohibited by RESOURCE25). 2. Puts the file in the vCalendar error directory.

If the event record is...	RESOURCE25...
Changed and the change is to date/time	1. Updates the event record in the RESOURCE25 database. 2. If the event has a previous room assignment, deletes the previous room assignment in the RESOURCE25 database. 3. If the request contains a room assignment, attempts to assign that room to the event, even if the head count exceeds the new room's capacity in the required layout. 4. Generates To Dos based on the vCalendar To Do processing option selected: • If "Always" is selected and the request contains a room assignment that is available, generates an informational To Do for the event organizer to alert him/her of the event changes. If the head count exceeds the room's capacity, that information is also included in the To Do. When that person completes the To Do, generates a success reply. • If "Always" or "Only If Action Required" is selected and the request either contains a room assignment that is not available or does not contain a room assignment, generates an assign room To Do for the event organizer. When that person completes the To Do, generates a counter message containing the suggested room. • If "Never" is selected and the request contains a new room assignment that is available, generates a success reply. • If "Never" is selected and the request contains a new room assignment that is not available or does not contain a room assignment, the event organizer or appropriate user must search for events without rooms, assign a room to this event, and change the event state to "confirmed." RESOURCE25 then asks if it should issue a counter message. If the person says "yes," RESOURCE25 issues a counter message containing the suggested room. If the person says "no," RESOURCE25 doesn't issue a counter message.

If the event record is...	RESOURCE25...
Changed and the change is to the assigned room	1. Deletes the previous room assignment in the RESOURCE25 database. 2. Attempts to assign the new room to the event, even if the head count exceeds the new room's capacity in the required layout. 3. Generates To Dos based on the vCalendar To Do processing option selected: • If "Always" is selected and the new room is available, generates an informational To Do for the event organizer to alert him/her of the event changes. If the head count exceeds the room's capacity, that information is also included in the To Do. When that person completes the To Do, generates a success reply. • If "Always" or "Only If Action Required" is selected and the new room is not available, generates an assign room To Do for the event organizer. When that person completes the To Do, generates a counter message containing the suggested room. • If "Never" is selected and the new room is available, generates a success reply. • If "Never" is selected and the new room is not available, the event organizer or appropriate user must search for events without rooms, assign a room to this event, and change the event state to "confirmed." RESOURCE25 then asks if it should issue a counter message. If the person says "yes," RESOURCE25 issues a counter message containing the suggested room. If the person says "no," RESOURCE25 doesn't issue a counter message.

If the event record is...	RESOURCE25...
Changed and the change is to the head count	1. Updates the event record in the RESOURCE25 database. 2. If the event has a room assignment, checks to see if the new head count exceeds the room's capacity in the requested layout. 3. Generates To Dos based on the vCalendar To Do processing option selected: • If "Always" is selected and the event either doesn't have a room assignment or the new head count doesn't exceed the assigned room's capacity, generates an informational To Do for the event organizer to alert him/her of the head count change. When that person completes the To Do, generates a success reply. • If "Only If Action Required" or "Never" is selected and the event either doesn't have a room assignment or the new head count doesn't exceed the assigned room's capacity, generates a success reply. • If "Always" or "Only If Action Required" is selected and the new head count does exceed the assigned room's capacity, generates an assign room To Do for the event organizer. When that person completes the To Do, generates a counter message containing the new suggested room. • If "Never" is selected and the new head count does exceed the assigned room's capacity, the event organizer or appropriate user must assign a new room to this event and change the event state to "confirmed." RESOURCE25 then asks if it should issue a counter message. If the person says "yes," RESOURCE25 issues a counter message containing the suggested room. If the person says "no," RESOURCE25 doesn't issue a counter message.

If the event record is...	RESOURCE25...
Cancelled	1. Processes the record based on the vCalendar cancelled events processing option selected: If “Change State” is selected, changes the state of the event in the RESOURCE25 database to “cancelled” and deletes the event’s room assignment(s). If “Delete Event” is selected, deletes the event from the RESOURCE25 database. 2. Generates To Dos if required based on the vCalendar To Do processing option selected: • If “Always” is selected, generates an informational To Do for the event organizer to alert him/her of the state change or deletion of the event. When that person completes the To Do, generates a success reply. • If “Only If Action Required” or “Never” is selected, generates a success reply.

Implementation requirements

To implement this interface, you need to:

- Modify your SIS to create vCalendar objects for new, modified, and cancelled class sections.
- Set up a cron process that transfers the vCalendar objects to the appropriate vCalendar directory for RESOURCE25 processing on a “timely” basis.
- Write a script to retrieve vCalendar objects coming to your SIS from RESOURCE25.
- Modify your SIS to update section records from incoming vCalendar objects.
- Write a policy for data exchange initiation.
- Prepare RESOURCE25 for the import and export of vCalendar files.

Information about how to meet these implementation requirements is in chapter 7.

7 Building Your Transaction-based Interface

Introduction

What's in this chapter

This chapter provides instructions and guidance for performing the tasks necessary to build the transaction-based interface described in chapter 6.

It tells you how to:

- Modify your SIS to implement the interface
- Prepare RESOURCE25 for the import and export of vCalendar files
- Write a data exchange policy

Chapter contents

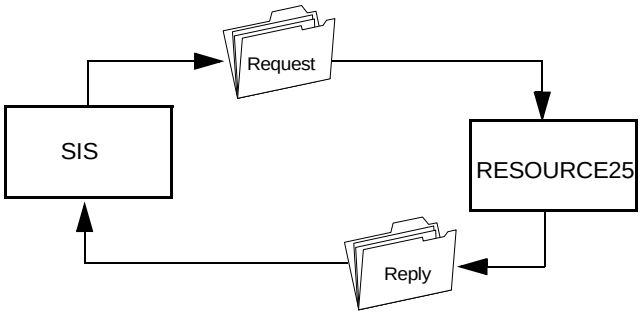
vCalendar processing scenarios

Introduction

This section describes the processing scenarios your transaction-based interface should support. Detailed information on how to modify your SIS to implement the interface begins on page 7-15.

The interface process is cyclic

The transaction-based interface process is cyclic. Your SIS makes a request and RESOURCE25 replies. The transaction is not considered complete until it has gone from your SIS to RESOURCE25 and back to your SIS (unless you have specified that you want one-way communication from your SIS to RESOURCE25 only - see page 7-33).



Note: *Shared event data can only be edited in RESOURCE25 in mid-cycle. You can't edit shared data in RESOURCE25 between cycles.*

Processing scenarios

There are two processing scenarios that you should accommodate in your transaction-based interface:

- Request/Reply
- Request/Counter

You may also want to accommodate a Request/Counter/Declinecounter scenario, although to do so requires some human and/or artificial intelligence on behalf of your SIS.

Each of these scenarios is illustrated and described beginning on the next page, followed by vCalendar object examples for each.

Processing scenarios, continued

In these scenarios, the request folder and reply folder are "local" to RESOURCE25. Your SIS is responsible for transferring files (probably via FTP) into the Request directory. RESOURCE25 writes files out to the Reply directory, and then executes a "script" written by your institution. That script should transfer (FTP) files back to your SIS file space for processing. It is very rare that your SIS and RESOURCE25 will read and write directly into the same directory.

Request/Reply scenario

In the request/reply scenario shown below:

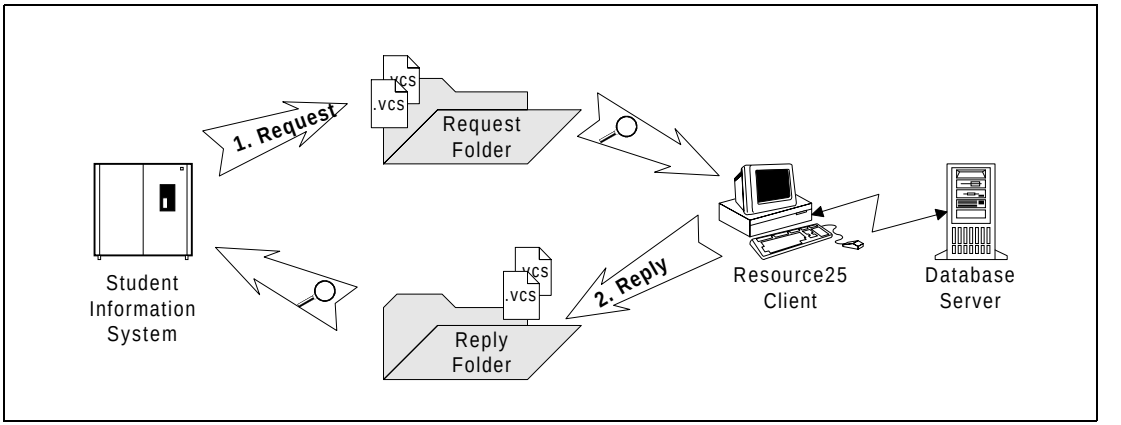
1. Your SIS generates either a vCalendar file with a request object containing new and/or modified events or a vCalendar file with a cancel object containing cancelled events. The new events have all pertinent scheduling information *including location*. Your SIS puts the file in the vCalendar Request directory.

   RESOURCE25 periodically polls the Request directory. When it finds a file, it imports it and updates its database with the event information.

2. RESOURCE25 sends a vCalendar file with a Reply object containing a "success" status message to the vCalendar Reply directory.

   Your SIS periodically polls the Reply directory. When it finds the file, it loads and processes it.

Note: *vCalendar files always have a "vcs" extension.*



Request/Counter scenario

In the request/counter scenario shown below:

1. Your SIS generates a vCalendar file with a request object containing *new events that do not have room assignments.* Your SIS puts the file in the vCalendar Request directory.

   RESOURCE25 periodically polls the Request directory. When it finds the request file, it imports it and updates its database with the new event information.

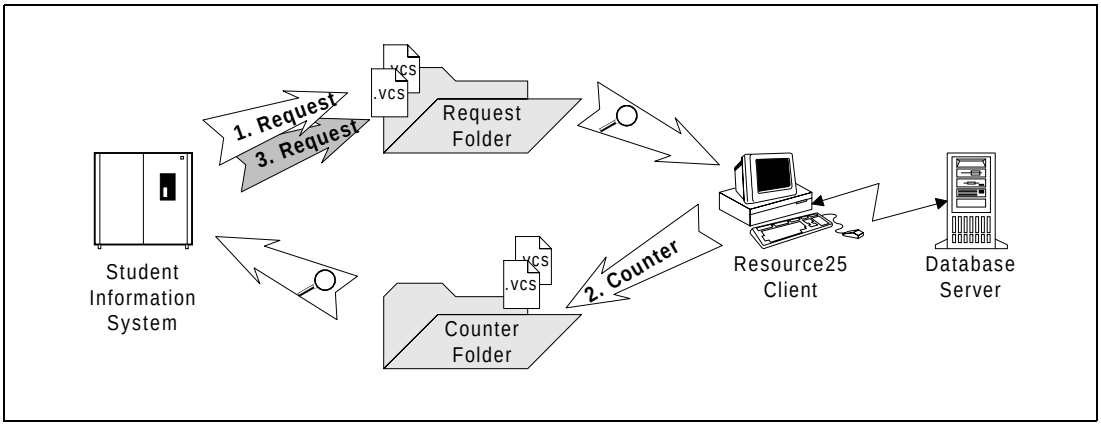
   A To Do message is generated to the appropriate scheduler to assign rooms to the events.

2. When the scheduler assigns the rooms and completes the To Do, RESOURCE25 sends a vCalendar file with a counter object to the vCalendar Counter directory.

   Your SIS periodically polls the Counter directory.

3. When your SIS finds the file, it reads it and updates the corresponding records in its database with the room assignments.

   It then generates a vCalendar file with a request object containing those class records (now with room assignments) and puts it in the vCalendar Request directory available for RESOURCE25 processing. This is a confirmation that the SIS is happy with the room assignments.



Request/Counter/ Declinecounter scenario

In the request/counter/declinecounter scenario shown on the next page:

1. Your SIS generates a vCalendar file with a request object containing new events that do not have room assignments. Your SIS puts the file in the vCalendar Request directory.

   RESOURCE25 periodically polls the Request directory. When it finds the file, it imports it and updates its database with the new event information.

   A To Do message is generated for the appropriate scheduler to assign rooms to the events.

2. When the scheduler assigns the rooms and completes the To Do, RESOURCE25 sends a vCalendar file with a counter object to the vCalendar Counter directory.

   Your SIS periodically polls the Counter directory.

3. When your SIS finds the file, it reads it, but does not accept the room assignments.

   It then generates a vCalendar file with a declinecounter object containing the class records and puts it in the vCalendar Request directory.

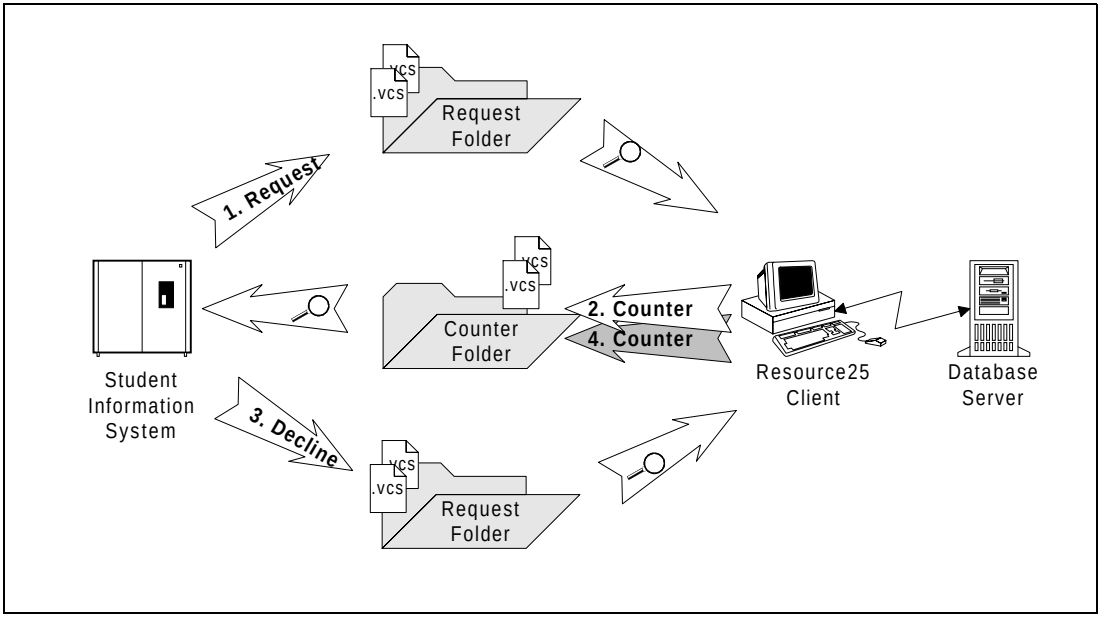
   RESOURCE25 periodically polls the Request directory. When it finds the file, it imports it.

   A To Do message is generated for the appropriate scheduler to assign rooms to the events again. The previous room assignment is *not* deleted.

4. When the scheduler assigns the rooms and completes the To Do, RESOURCE25 sends a vCalendar file with a counter object to the vCalendar Counter directory location available for SIS processing.

   This counter and declinecounter sequence can continue until the SIS is happy. It indicates it is satisfied with the room assignments by sending a request instead of a declinecounter.

Request/Counter/ Declinecounter description, continued



vCalendar object processing examples

Introduction

The following are examples of vCalendar objects generated for the processing scenarios described on the previous pages.

Request/Reply example

In this example, a new computer science section, CS138, is being added on Tuesday and Thursday from 2PM to 3PM in room LT 202. The expected head count is 50.

1. The SIS issues this vCalendar file with a request object:

```
BEGIN:VCALENDAR
METHOD:REQUEST
VERSION:1.0
PRODID:-//SIS Vendor//NONSGML MySIS//EN
BEGIN:VEVENT
UID:MySIS-CRN 003244
SUMMARY:CS138 - Introduction to Java
SEQUENCE:1
PRIORITY:0
DCREATED:19970812T105000
LAST-MODIFIED:19970812T104000
DTSTART:19980907T140000
DTEND:19980907T150000
RRULE:W1 TU TH #10 19981231T235900
LOCATION:LT 202
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-R25-ACCOUNT:Computer Science
X-R25-TYPE:Section
STATUS:Tentative
END:VEVENT
END:VCALENDAR
```

Request
Section name
Date/time information
Event location
Expected head count
Sponsoring department
Event type

Request/Reply example, continued

2. When this event is imported, RESOURCE25 checks to see if the requested room is available. If it is, RESOURCE25 returns a vCalendar file with a reply object that indicates success. Notice that RESOURCE25 returns an R25 internal identifier for this event. Using this value in subsequent requests will speed things up.

Reply

R25 internal identifier

Success status

```
BEGIN:VCALENDAR
METHOD:REPLY
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=935:MySIS-CRN:003244
SEQUENCE:1
X-R25-REQUEST-STATUS:200;Success
END:VEVENT
END:VCALENDAR
```

However, if the room is already booked, RESOURCE25 generates an "assign room" To Do for the appropriate scheduler. When the scheduler assigns a room for the event and completes the To Do, RESOURCE25 generates a vCalendar file with a counter object indicating the new room assignment (see page 7-12).

Request/Reply example, continued

If an event has several meeting patterns, multiple, related vEvent objects must be included in the request, as shown in this example:

	BEGIN:VCALENDAR
Request	**METHOD:REQUEST**
	PRODID:-//SIS Vendor//NONSGML MySIS//EN
Beginning of "group" event	**BEGIN:VEVENT**
	UID:MySIS-CRN/003172
	SUMMARY:CS101 - Introduction to Programming
	SEQUENCE:1
	PRIORITY:0
	DCREATED:19970812T105000
	LAST-MODIFIED:19970812T104000
	DTSTART:19980831T000000
	DTEND:19981220T000000
Identifies the event as a "group" event with no actual date reservations	**TRANSP:1**
	X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
	X-R25-TYPE:Section
	X-R-25-ACCOUNT:Computer Science
End of "group" event	**END:VEVENT**
Beginning of first related event - class schedule	**BEGIN:VEVENT**
	UID:MySIS-CRN/003172/1
	SUMMARY:Class Schedule
	SEQUENCE:1
	PRIORITY:0
	DCREATED:19970812T105000
	LAST-MODIFIED:19970812T104000
	DTSTART:19980907T100000
	DTEND:19980907T110000
	RRULE:W1 MO TH
Related to "group" event as one date reservation for the event	**RELATED-TO;X-R25-REL=RESERVATION:MySIS-CRN/003172**
	X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
	X-R25-TYPE:Section
	X-R25-ACCOUNT:Computer Science
End of 1st related event	**END:VEVENT**
Beginning of 2nd related event - lab schedule	**BEGIN:VEVENT**
	UID:MySIS-CRN/003172/2
	SUMMARY:Lab Schedule
	SEQUENCE:1
	PRIORITY:0
	DCREATED:19970812T1050000
	LAST-MODIFIED:19970812T104000
	DTSTART:19980908T140000
	DTEND:19980908T170000
	RRULE:W1 TU WE TH
Related to "group" event as another date reservation for the event	**RELATED-TO;X-R25-REL=RESERVATION:MySIS-CRN/003172**
	X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
	X-R25-TYPE:Section
	X-R25-ACCOUNT:Computer Science
End of 2nd related event	**END:VEVENT**
	END:VCALENDAR

Request/Counter/ Declinecounter example

In this example, the SIS is requesting a room assignment (there is no location in the request) for a new computer science section, CS137, on Monday, Wednesday and Friday from 9AM to 10AM. The expected head count is 50.

1. The SIS issues a vCalendar file with this request object:

Request

```
BEGIN:VCALENDAR
METHOD:REQUEST
VERSION:1.0
PRODID:-//SIS Vendor//NONSGML MySIS//EN
BEGIN:VEVENT
UID:MySIS-CRN:003243
SUMMARY:CS137 - Introduction to HTML
SEQUENCE:1
PRIORITY:0
```

Event organizer

```
ATTENDEE;ROLE=ORGANIZER:jdoe@myu.edu
DCREATED:19970812T104000
LAST-MODIFIED:19970812T104000
DTSTART:19980906T090000
DTEND:19980906T100000
```

No location specified

```
RRULE:W1 MO WE FR #10 19981231T235900
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-25-ACCOUNT:Computer Science
X-R25-TYPE:Section
STATUS:Tentative
END:VEVENT
END:VCALENDAR
```

Request/Counter/ Declinecounter example, continued

2. After the scheduler has found a room for the event, RESOURCE25 generates a vCalendar file with a counter object containing the location information. Note how some of RESOURCE25's internal identifiers have been filled.

Counter

RESOURCE25 internal event identifier

RESOURCE25 internal event organizer identifier

Suggested location

RESOURCE25 internal event type identifier

```
BEGIN:VCALENDAR
METHOD:COUNTER
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=928:MySIS-CRN:003243
SUMMARY:CS137 - Introduction to HTML
SEQUENCE:1
PRIORITY:0
ATTENDEE;ROLE=ORGANIZER;X-R25-ID=2:jdoe@myu.edu
DCREATED:19970812T104000
LAST-MODIFIED:19970812T130200
DTSTART:19980906T090000
DTEND:19980906T100000
RRULE:W1 MO WE FR #10 19981231T235900
LOCATION;X-R25-ID=83:LT 204
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-25-ACCOUNT:Computer Science
X-R25-TYPE;X-R25-ID=5:Section
STATUS:Tentative
END:VEVENT
END:VCALENDAR
```

Request/Counter/ Declinecounter example, continued

3. If the SIS is happy with the suggested location, it reissues the request. Note that the sequence number is incremented.

Request

Sequence number has been incremented

```
BEGIN:VCALENDAR
METHOD:REQUEST
VERSION:1.0
PRODID:-//SIS Vendor//NONSGML MySIS//EN
BEGIN:VEVENT
UID;X-R25-ID=928:MySIS-CRN:003243
SUMMARY:CS137 - Introduction to HTML
SEQUENCE:2
PRIORITY:0
ATTENDEE;ROLE=ORGANIZER;X-R25-ID=2:Smith.Tom@r25
DCREATED:19970812T104000
LAST-MODIFIED:19970812T151000
DTSTART:19980906T090000
DTEND:19980906T100000
RRULE:W1 MO WE FR #10 19981231T235900
LOCATION;X-R25-ID=83:LT 204
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-25-ACCOUNT:Computer Science
X-R25-TYPE;X-R25-ID=5:Section
STATUS:Tentative
END:VEVENT
END:VCALENDAR
```

RESOURCE25 replies with a success message.

Reply

Success status

```
BEGIN:VCALENDAR
METHOD:REPLY
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=928:MySIS-CRN:003243
SEQUENCE:2
X-R25-REQUEST-STATUS:200;Success
END:VEVENT
END:VCALENDAR
```

Request/Counter/ Declinecounter example, continued

4. However, if the SIS doesn't like the suggested location, it issues a vCalendar file with a declinecounter object. Note that the sequence number is not incremented.

Declinecounter

Sequence number is same as counter

```
BEGIN:VCALENDAR
METHOD:DECLINECOUNTER
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVET
UID;X-R25-ID=928:MySIS-CRN:003243
SEQUENCE:1
END:VEVENT
END:VCALENDAR
```

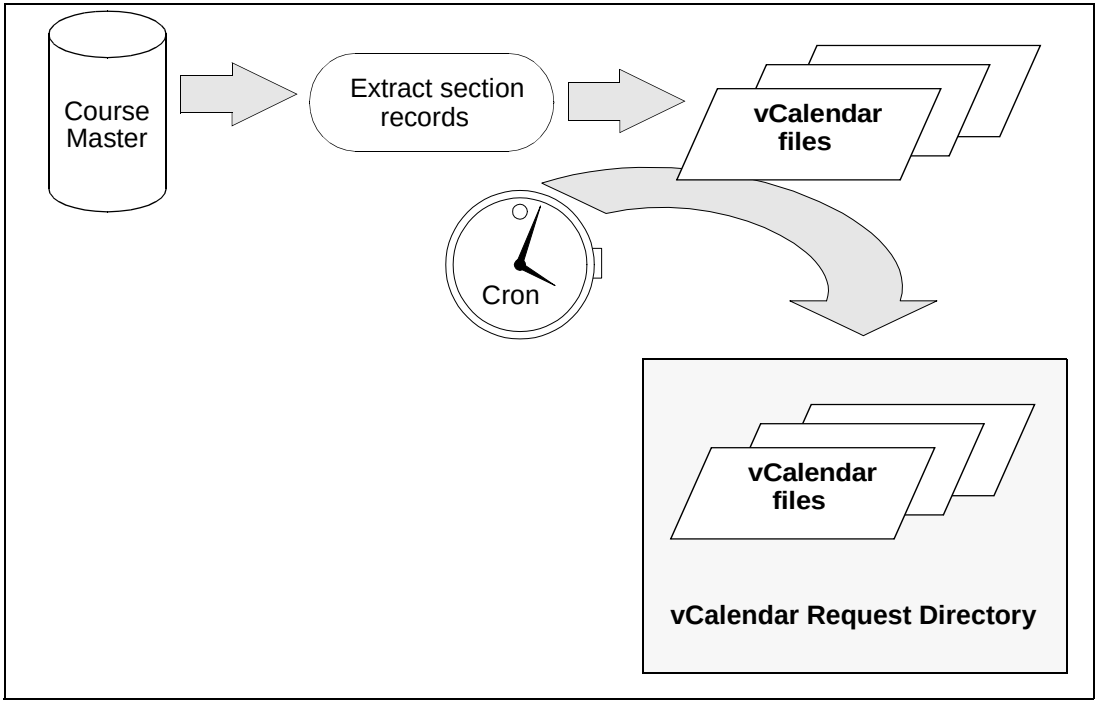
After the scheduler has found an alternative room for the event, RESOURCE25 generates another counter message with the location information.

Creating vCalendar files

Introduction

To begin building your transaction-based interface you must:

- Modify your SIS so that it is capable of extracting new, modified, and cancelled section records and creating vCalendar files with request and cancel objects containing those events. It must also be capable of keeping a sequence number.
- Set up a cron (or other timed) process to transfer vCalendar files to the vCalendar Request directory on a “timely” basis.



Extracting section records

The modifications you make to your SIS must allow you to extract new, modified, and cancelled sections for section types that are included in the RESOURCE25 database. This might include section types such as LEC, LAB, and so on.

Creating vCalendar files

The modifications you make to your SIS must allow you to create vCalendar files containing the extracted events. These files must have the correct structure, and the vCalendar and vEvent objects in them must be formatted correctly and contain the appropriate properties and data. The file type for all vCalendar files must be *.vcs*.

SIS and RESOURCE25 shared data

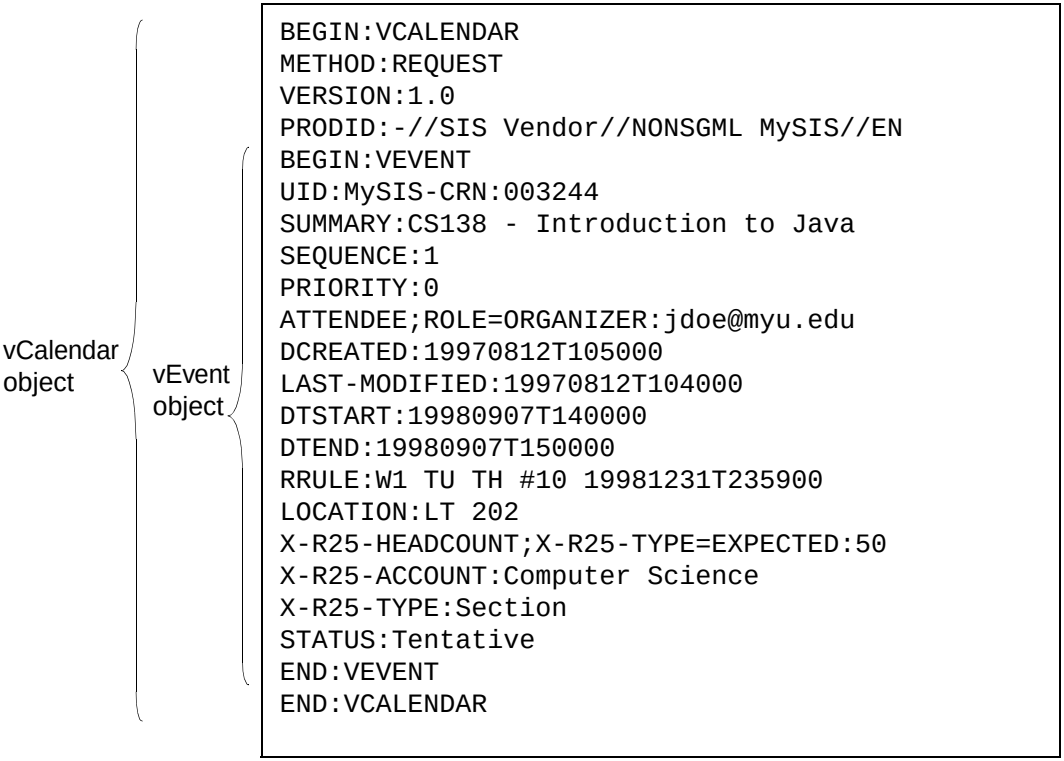
For each event, the following data can be shared between your SIS and RESOURCE25:

- Universal identifier
- Name
- Description
- Creation date
- Last modified date
- Start date and time of first event occurrence
- End date and time of first event occurrence
- Meeting pattern and/or ad hoc meeting dates
- Date exceptions/constraints
- Location
- Expected and/or actual head count
- Event type
- Organizer role and associated contacts
- Categories
- Space preferences
- Status
- Account

You may want to modify your SIS database to store additional event data based on this list.

vCalendar request example

The example below shows a vCalendar file with a request object. As mentioned in chapter 2, a vCalendar file contains a vCalendar object with an identified method (transaction) type. The vCalendar object contains vEvent objects representing scheduled time periods. This file example contains only one vEvent object, but could contain more than one. Each event is processed based on the method type of the vCalendar object in which it is contained.



vCalendar properties in the request example

Both vCalendar and vEvent objects are made up of "object properties" that define the data in the object. The table beginning below describes each of the vCalendar and vEvent properties in the request example on the previous page. It doesn't include all the properties that could be in a request.

Some of the properties listed below are found in all vCalendar objects, regardless of method type. These include all of the vCalendar properties and these vEvent properties: BEGIN, UID, SEQUENCE, and END.

Note*: See Appendix A for a complete description of all vCalendar properties including detailed usage information.*

This vCalendar property...	Is used to identify the...	Example
BEGIN	Beginning of the vCalendar object	`BEGIN:VCALENDAR`
METHOD	Processing type for all vEvent objects in the vCalendar object	`METHOD:REQUEST`
VERSION	Supported vCalendar version (currently 1.0)	`VERSION:1.0`
PRODID	Product that created the object (your SIS)	`PRODID:-//SIS Vendor// NONSGML MySIS//EN`
END	End of the vCalendar object	`END:VCALENDAR`
This vEvent property...	**Is used to identify the...**	**Example**
BEGIN	Beginning of the vEvent object	`BEGIN:VEVENT`
UID	Persistent, globally-unique identifier of the object (from your SIS)	`UID:MySIS-CRN:003244`
SUMMARY	Name of the event	`SUMMARY:CS138 - Introduction to Java`
SEQUENCE	Number of times changes have been made to the event	`SEQUENCE:1`

This vEvent property...	Is used to identify the...	Example
PRIORITY	Priority of the event	`PRIORITY:0`
ATTENDEE	The RESOURCE25 contact specified as the event organizer	`ATTENDEE; ROLE=ORGANIZER: jdoe@myu.edu`
DCREATED	Date and time the vevent object was created	`DCREATED: 19970812T105000`
LAST-MODIFIED	Date and time the event information was last modified	`LAST-MODIFIED: 19970812T104000`
DTSTART	Start date and time of the initial event meeting	`DTSTART: 19980907T140000`
DTEND	End date and time of the initial event meeting	`DTEND:19980907T150000`
RRULE	Recurring date/time pattern of the event	`RRULE:W1 TU TH #10 19981231T235900`
LOCATION	Intended location (space/room) for the event	`LOCATION:LT 202`
X-R25-HEADCOUNT	Head count for the event (used with the RESOURCE25 internal identifier for expected head count in the example)	`X-R25-HEADCOUNT; X-R25-TYPE=EXPECTED:50`
X-R25-ACCOUNT	Sponsoring account (department) of the event	`X-R25-ACCOUNT: Computer Science`
X-R25-TYPE	RESOURCE25 event type of the event	`X-R25-TYPE:Section`
STATUS	Current status of the event	`STATUS:Tentative`
END	End of the vEvent object	`END:VEVENT`

Guidelines for specifying event data in request objects

Make sure you follow these guidelines when creating vEvent objects in vCalendar request objects:

- Include a complete specification of each event, because the absence of a property implies that the data is no longer valid or applicable.
- When using this method for event changes, make sure you increment the SEQUENCE number of the event. On import, RESOURCE25 ignores modified events in a request object that have a sequence number equal to or less than the current version. There can be gaps between sequence numbers, as long as the sequence number is greater.

vCalendar cancel example

This example shows a vCalendar file containing a Cancel vCalendar object.

```
BEGIN:VCALENDAR
METHOD:CANCEL
VERSION:1.0
PRODID:-//SIS Vendor//NONSGML MySIS//EN
BEGIN:VEVENT
UID:MySIS-CRN:003244
SEQUENCE:1
STATUS:CANCELLED
END:VEVENT
END:VCALENDAR
```

vCalendar properties in the cancel example

The table below describes each of the vCalendar and vEvent properties in the cancel example on the previous page. It doesn't include all the properties that could be in a cancel.

Note*: See Appendix A for a complete description of all vCalendar properties including detailed usage information.*

This vCalendar property...	Is used to identify the...	Example
BEGIN	Beginning of the vCalendar object	`BEGIN:VCALENDAR`
METHOD	Processing type for all vEvent objects in the vCalendar object	`METHOD:CANCEL`
VERSION	Supported vCalendar version (currently 1.0)	`VERSION:1.0`
PRODID	Product that created the object (your SIS)	`PRODID:-//SIS Vendor// NONSGML MySIS//EN`
END	End of the vCalendar object	`END:VCALENDAR`
This vEvent property...	**Is used to identify the...**	**Example**
BEGIN	Beginning of the vEvent object	`BEGIN:VEVENT`
UID	Persistent, globally-unique identifier of the object	`UID:MySIS-CRN:003244`
SEQUENCE	Number of times changes have been made to the event	`SEQUENCE:1`
STATUS	Current status of the event	`STATUS:CANCELLED`
END	End of the vEvent object	`END:VEVENT`

Guidelines for specifying event data in cancel objects

Make sure you follow these guidelines when creating vEvent objects in vCalendar cancel objects:

- The UID and SEQUENCE properties are required.
- If you use the STATUS property, set it to CANCELLED.
- When you use this method, RESOURCE25 cancels the entire event. (How the event is cancelled is based on the cancel option you've chosen for import processing. See page 7-32). If you want to just cancel individual event meetings, send a request object with an EXDATE property (see Appendix A).
- An event cancelled through vCalendar can't be resurrected using a new request. You must create a new event.

Making vCalendar request files available to RESOURCE25

After you've modified your SIS to create vCalendar files, you set up a cron (or other timed) process to transfer the files to the vCalendar Request directory, ready for import by RESOURCE25.

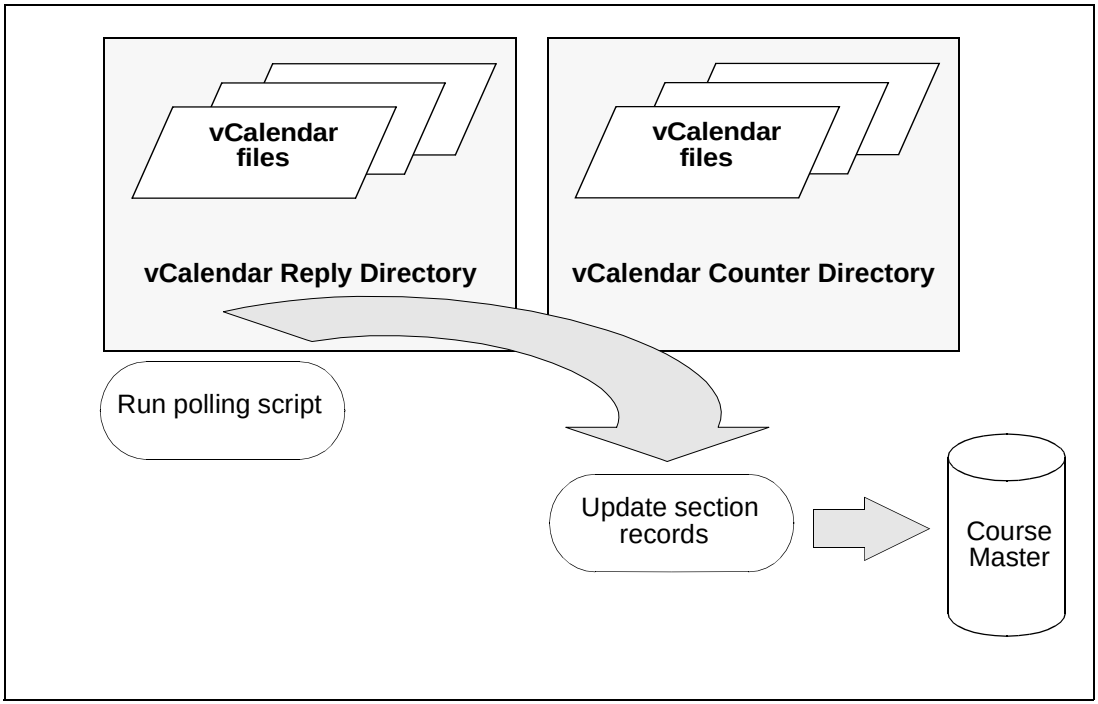
Your cron process should execute on a "timely" basis. What "timely" means will probably vary depending on what part of the semester/term you are in. Near the beginning of the semester/term, when many section changes are occurring, you may want the cron process to execute frequently. Later in the semester/term, as fewer section changes are occurring, you may want the process to execute less frequently.

Updating your SIS from vCalendar files

Introduction

To continue building your transaction-based interface you must:

- Write a file transfer script that RESOURCE25 can run each time it creates a vCalendar file and sends it to the Reply or Counter directory.
- Modify your SIS to update section records in your course master file from incoming vCalendar files.



Checking for vCalendar files

As part of the course master update process, you must write a script that RESOURCE25 can execute each time it sends a vCalendar file to the Reply or Counter directory. The script should transfer (FTP) the file to your SIS and possibly invoke an update process. See the information beginning on page 7-36.

The actual file name used for a vCalendar file is not significant. As a convention, RESOURCE25 names the files it creates after the event reference identifier. Your SIS should not place any special significance on this name.

The file type for all vCalendar files is *.vcs*.

Updating section records

The modifications you make to your SIS must allow you to read incoming vCalendar files, identify the section records, locate the corresponding records in your course master file, and update them.

vCalendar reply example

This example shows a vCalendar file containing a Reply vCalendar object.

```
BEGIN:VCALENDAR
METHOD:REPLY
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=935:MySIS-CRN:003244
SEQUENCE:1
X-R25-REQUEST-STATUS:200;Success
END:VEVENT
END:VCALENDAR
```

vCalendar properties in the reply example

The table below describes each of the vCalendar and vEvent properties in the reply example on the previous page. It doesn't include all the properties that could be in a reply.

Note*: See Appendix A for a complete description of all vCalendar properties including detailed usage information.*

This vCalendar property...	Is used to identify the...	Example
BEGIN	Beginning of the vCalendar object	`BEGIN:VCALENDAR`
METHOD	Processing type for all vEvent objects in the vCalendar object	`METHOD:REPLY`
VERSION	Supported vCalendar version (currently 1.0)	`VERSION:1.0`
PRODID	Product that created the object (RESOURCE25)	`PRODID:-//Universal Algorithms Inc// NONSGML RESOURCE25//EN`
END	End of the vCalendar object	`END:VCALENDAR`
This vEvent property...	**Is used to identify the...**	**Example**
BEGIN	Beginning of the vEvent object	`BEGIN:VEVENT`
UID	Persistent, globally-unique identifier of the object (now includes the R25 identifier)	`UID;X-R25-ID=935: MySIS-CRN:003244`
SEQUENCE	Number of times changes have been made to the event	`SEQUENCE:1`
X-R25-REQUEST-STATUS	Status of the reply to the request (status code and text)	`X-R25-REQUEST-STATUS: 200;Success`
END	End of the vEvent object	`END:VEVENT`

Guidelines for reading event data in a reply

Make sure you follow these guidelines when reading vEvent objects in vCalendar reply objects:

- If RESOURCE25 detects errors in an event, it sends the vCalendar file containing the event to an error directory. *It does not send a reply with an error status message.* Your SIS should respond with a new request with corrected property values.

vCalendar counter example

This example shows a vCalendar file containing a Counter vCalendar object.

```
BEGIN:VCALENDAR
METHOD:COUNTER
VERSION:1.0
PRODID:-//Universal Algorithms Inc//NONSGML
RESOURCE25//EN
BEGIN:VEVENT
UID;X-R25-ID=928:MySIS-CRN:003243
SUMMARY:CS137 - Introduction to HTML
SEQUENCE:1
PRIORITY:0
ATTENDEE;ROLE=ORGANIZER;X-R25-ID=2:jdoe@myu.edu
DCREATED:19970812T104000
LAST-MODIFIED:19970812T130200
DTSTART:19980906T090000
DTEND:19980906T100000
RRULE:W1 MO WE FR #10 19981231T235900
LOCATION;X-R25-ID=83:LT 204
X-R25-HEADCOUNT;X-R25-TYPE=EXPECTED:50
X-25-ACCOUNT:Computer Science
X-R25-TYPE;X-R25-ID=5:Section
END:VEVENT
END:VCALENDAR
```

vCalendar properties in the counter example

The table beginning below describes each of the vCalendar and vEvent properties in the counter example on the previous page. It doesn't include all the properties that could be in a counter.

Note*: See Appendix A for a complete description of all vCalendar properties including detailed usage information.*

This vCalendar property...	Is used to identify the...	Example
BEGIN	Beginning of the vCalendar object	`BEGIN:VCALENDAR`
METHOD	Processing type for all vEvent objects in the vCalendar object	`METHOD:COUNTER`
VERSION	Supported vCalendar version (currently 1.0)	`VERSION:1.0`
PRODID	Product that created the object (RESOURCE25)	`PRODID:-//Universal Algorithms Inc// NONSGML RESOURCE25//EN`
END	End of the vCalendar object	`END:VCALENDAR`
This vEvent property...	**Is used to identify the...**	**Example**
BEGIN	Beginning of the object	`BEGIN:VEVENT`
UID	Persistent, globally-unique identifier of the object (now includes the R25 identifier)	`UID;X-R25-ID=928: MySIS-CRN:003243`
SUMMARY	Name of the event	`SUMMARY:CS137- Introduction to HTML`
SEQUENCE	Number of times changes have been made to the event	`SEQUENCE:1`
PRIORITY	Priority of the event	`PRIORITY:0`

This vEvent property...	Is used to identify the...	Example
ATTENDEE	The RESOURCE25 contact specified as the event organizer (now includes the R25 identifier)	`ATTENDEE; ROLE=ORGANIZER; X-R25-ID=2: jdoe@myu.edu`
DCREATED	Date and time the vevent object was created	`DCREATED: 19970812T104000`
LAST-MODIFIED	Date and time the event information was last modified	`LAST-MODIFIED: 19970812T130200`
DTSTART	Start date and time of the initial event meeting	`DTSTART: 19980906T090000`
DTEND	End date and time of the initial event meeting	`DTEND:19980906T100000`
RRULE	Recurring date/time pattern of the event	`RRULE:W1 MO WE FR #10 19981231T235900`
LOCATION	Location (space/room) that has been assigned to the event in RESOURCE25 (now includes the R25 identifier)	`LOCATION; X-R25-ID=83:LT 204`
X-R25-HEADCOUNT	Head count for the event (used with the RESOURCE25 internal identifier for expected head count in the example)	`X-R25-HEADCOUNT; X-R25-TYPE=EXPECTED:50`
X-R25-ACCOUNT	Sponsoring account (department) of the event	`X-R25-ACCOUNT: Computer Science`
X-R25-TYPE	RESOURCE25 event type of the event (now includes the R25 identifier)	`X-R25-TYPE; X-R25-ID=5:Section`
END	End of the vEvent object	`END:VEVENT`

Guidelines for reading event data in a counter

Make sure you follow these guidelines when reading vEvent objects in vCalendar counter objects:

- Your SIS should check all fields for changes to shared data.
- Your SIS should respond to a Counter with a Request agreeing to the suggested room or a Declinecounter if the suggested room is not acceptable.

Preparing RESOURCE25 for file import and export

What you have to do

There are three things you need to do to prepare RESOURCE25 for the import and export of vCalendar files:

You must...	Which means...	For information see page...
1. Select vCalendar import processing options	Using the vCalendar system definition to indicate how you want RESOURCE25 to handle certain aspects of the import process.	7-31
2. Specify vCalendar directory and export script locations	Using the vCalendar system definition to enter the directories and export script locations you want to use.	7-36
3. Specify where imported events should be placed in your event structure	Creating vCalendar filter rules for import event types as needed and/or Making sure your event structure supports use of the default filter rule.	7-40 7-48

Selecting vCalendar import processing options

Introduction

As part of preparing RESOURCE25 for the import of vCalendar files, you must select from options that control certain aspects of the import process.

You set these options using the vCalendar system definition.

"Alien" events

Events sent from your SIS to RESOURCE25 for processing are considered "alien" events by RESOURCE25. This means that they are subject to the processing restrictions you impose by the import processing options you select.

You can "naturalize" an alien event if you no longer want it to be subject to these restrictions. With the Identification tab selected, choose Actions|Naturalize Event on the Edit Event form.

Processing options

The vCalendar system definition allows you to select how you want RESOURCE25 to process event records on import. You can specify:

Who can edit the vCalendar data in imported events:

- **No one** - The shared data in imported vCalendar events can't be edited. This includes all data defined by the vCalendar properties of the event. You may want to choose this if you're using RESOURCE25 as a repository of class event data and space assignments, but do not want to edit class data in RESOURCE25.
- **Event Organizer only** - The contact specified as the event organizer can edit the shared data of the event. This is the person identified by the ATTENDEE property of the event.

  Note: *In RESOURCE25, the event organizer must be an active RESOURCE25 user. This may not be the same event organizer specified in your SIS. It you want to preserve the SIS organizer information in RESOURCE25, you should create the appropriate contact records for all potential event organizers in RESOURCE25. These contacts do not have to be RESOURCE25 users. Remember, however, that any To Do items associated with an event import will be sent to the RESOURCE25 event organizer, not to the event organizer defined in your SIS, unless they are the same person.*
- **R25 Permissions** - The user(s) who have the appropriate security for an imported event can edit the shared data of that event. Select this option if you want authorized users to have access to imported events and their shared event data.

How cancelled events should be handled:

- **Change State** - The event's state is set to cancelled and all space reservations for the event are deleted. You may want to choose this option until you are confident about the SIS/RESOURCE25 interaction.
- **Delete Event** - The event is deleted from RESOURCE25. You may want to choose this option once you are confident about the SIS/RESOURCE25 interaction.

Processing options, continued

Whether or not RESOURCE25 should respond to your SIS:

- **Break** - The transaction cycle from RESOURCE25 back to your SIS is broken. RESOURCE25 won't export reply and counter objects. You will probably want to select this option if you've indicated that no one can edit imported events, if you don't want to update your SIS, or if you haven't written an SIS update program.
- **Maintain** - The transaction cycle from RESOURCE25 back to your SIS is maintained. RESOURCE25 will export reply and counter objects. You will probably want to select this option if you've indicated that the event organizer or authorized users can edit imported events and you want to update your SIS.

Whether or not a To Do should be generated for each imported event:

- **Always** - The event organizer will receive a To Do for each imported event, even if action on the shared event data is not required.
- **Only If Action Required** - The event organizer will receive a To Do for an imported event only if action on the shared event data is required.
- **Never** - No To Dos will be issued for imported events. You should select this option when you are sending a large number of events to RESOURCE25 at one time, such as when you are sending a semester's/term's worth of classes for room assignments.

The processing options you've chosen should be reflected in your data exchange policy. See page 7-53.

Instructions

Follow these instructions to select import processing options:

Step	Action
1	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 
2	Double-click the vCalendar icon. **Result**: The vCalendar Processing Options form opens. 

Step	Action
3	Select the processing option you want in each category. If you select “Always” or “Only If Action Required” in the To Do Generated category, select the To Do priority from the drop down list.
4	Click OK. **Result**: You are returned to the System Definitions and Defaults form. The processing options you selected are saved.

Specifying vCalendar directory and export script locations

Introduction

Another task you must perform to prepare RESOURCE25 for the import and export of vCalendar files is to specify the directory locations you want to use for vCalendar files and the full pathnames of any scripts you want RESOURCE25 to execute as part of the vCalendar export process.

You specify these locations and files using the vCalendar system definition.

Default locations

This table lists the default physical locations of vCalendar import and export files. If you decide to change the default locations, make sure you, at minimum, designate a different directory for import and export files. Your SIS extract and update processes must coordinate the use of the correct directories.

	Of this type...	Have this default location...
Import files...	Request	*<R25>\vcal\request*
	Declinecounter	*<R25>\vcal\request*
	Cancel	*<R25>\vcal\request*
	Errors	*<R25>\vcal\error*
Export files..	Publish	*<R25>\vcal\publish*
	Reply	*<R25>\vcal\reply*
	Counter	*<R25>\vcal\counter*
	Cancel (in R25)	*<R25>\vcal\cancel*

Note: *These locations are platform-specific. For example, you need to edit these settings from a Win3.x machine to create Win3.x names.*

Default locations, continued

Note the two different locations for canceled events - alien events canceled by your SIS that are being sent to RESOURCE25, and native events canceled inside RESOURCE25 that are being exported for notification to the outside world.

Export scripts

You can also enter the full pathnames of any export scripts you want RESOURCE25 to run in association with the export of publish, reply, counter, or R25 cancel vCalendar objects.

For example, if you want RESOURCE25 to run an FTP script that retrieves and processes replies generated by RESOURCE25, enter the script location in the REPLY area for export scripts on the vCalendar Directories form (see page 7-39).

RESOURCE25 places the name of the exported file as the first and only parameter on the command line when it executes the script.

Instructions

Follow these instructions to modify default vCalendar file locations and specify scripts you want executed during export:

Step	Action
1	From the RESOURCE25 main menu, choose Administration|System Definitions. **Result**: The System Definitions and Defaults form opens. 
2	Double-click the vCalendar icon. **Result**: The vCalendar Processing Options form opens. 

Step	Action
3	Click Directories. **Result**: The vCalendar Directories form opens. 
4	Modify the directory locations for import and export files as needed. You can type the directory pathname or use the browse button to the right of the entry to locate and select the location.
5	Enter (or use the browse button to locate and select) the pathnames of the export scripts you want to use.
6	Click OK. **Result**: You are returned to the vCalendar Processing Options form.
7	Click OK. **Result**: You are returned to the System Definitions and Defaults form. Your directory and file locations are saved.

Specifying where imported events should be placed in your event structure

Introduction

The final task you must perform to ready RESOURCE25 for the import and export of vCalendar files is to specify where you want imported events to be placed in your RESOURCE25 event structure. You do this by creating vCalendar filter rules for the event types you plan to import, or perferably, by ensuring that your event structure supports use of the "default filter rule" described beginning on page 7-48.

vCalendar filter rules

When RESOURCE25 imports a vEvent object, it refers to a set of rules you've set up that are linked to the event's event type (X-R25-TYPE property). These rules describe where to place events of that event type in your event structure.

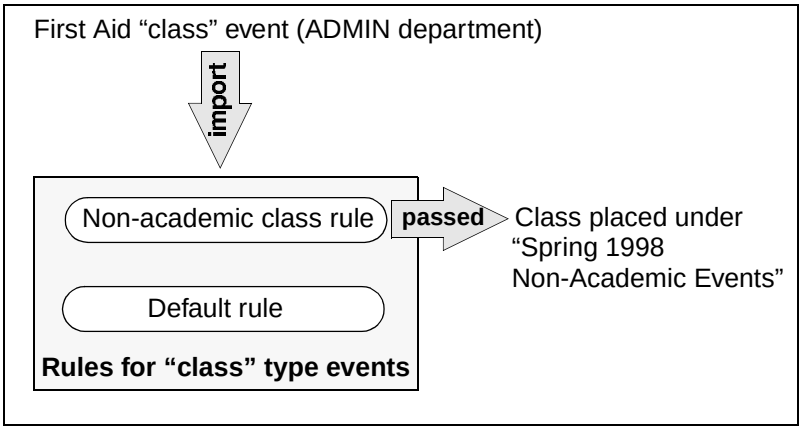
Each rule contains criteria that are matched against property values in the vEvent object. You can create multiple rules for each event type and specify the order in which you want them applied to incoming events.

The values in the SUMMARY, DSTART, DEND, and X-R25-ACCOUNT vEvent properties can be used as criteria checks.

If an incoming event doesn't match any of the criteria you've set up for its event type, RESOURCE25 tries to use the "default filter rule" described beginning on page 7-48 to place the event.

Example

For example, we could create a filter rule for non-academic classes that specifies that incoming events with a “class” event type and an “Admin” account (department) should be placed under the “Spring 1998 Non-Academic Events” event in our event structure.

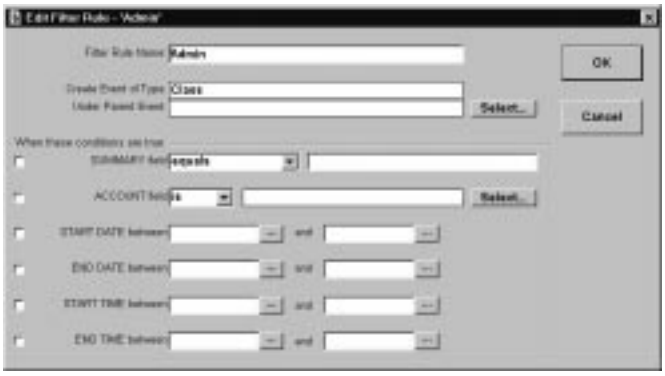


Instructions

Follow these instructions to create vCalendar filter rules.

Note: *It is highly recommended that you build your academic event structure to take full advantage of the default filter rule described beginning on page* 7-48. ***If you build your academic event structure so that the default filter rule correctly places all imported events, you don't have to create any vCalendar filter rules.***

Step	Action
1	From the RESOURCE25 main menu, choose Interface|Import|vCalendar Filter Rules. **Result**: The vCalendar Filter Rules Master List form opens. 
2	Select the event type you want to create a filter rule for from the drop down list.
3	Enter a descriptive name for your new rule in the New vCalendar Filter Rule field.

Step	Action
4	Click New. **Result**: The Edit Filter Rule form opens. 
5	Click the Select button to the right of the Under Parent Event field. **Result**: The Select Destination Event form opens. 

<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>6</td><td>Locate and select the parent event. You may have to move up or down the cabinet event structure to find it, using the up directory button to move up and double-clicking event names to move down.
Note: Appropriate parents have a check mark next to their event name. These are the only events you can select.

You can select whether you want to see potential parents of all possible event types or just a specific event type from the Event of Type drop down list. Changing the selection changes which events are “check-marked.”</td></tr>
<tr><td>7</td><td>Click OK.

Result: You are returned to the Edit Filter Rule form. The parent event you’ve selected is displayed in the Under Parent Event field.</td></tr>
<tr><td>8</td><td>Specify one or more filter conditions for the rule:
<table>
<tr><th>To add this kind of condition...</th><th>Do this...</th></tr>
<tr><td>Summary</td><td>1. Click the Summary option box.
2. Select the appropriate operator from the drop down list.
3. Type the event name in the blank field.</td></tr>
<tr><td>Account</td><td>1. Click the Account option box.
2. Select “is” or “is not” from the drop down list.
3. Click the Select button.
The Account List form opens.
4. Select an account and click OK.
You are returned to the Edit Filter Rule form. The account you selected is displayed.</td></tr>
</table></td></tr>
</table>

Step	Action	
8, cont.	**To add this kind of condition...**	**Do this...**
	Start Date	1. Click the Start Date option box. 2. Enter the first date of a DTSTART date range or click the Details button to open an Edit Date form where you can select a date. 3. If you want to, enter the last date of a DTSTART date range or click the Details button to open an Edit Date form where you can select a date.
	End Date	1. Click the End Date option box. 2. Enter the first date of a DTEND date range or click the Details button to open an Edit Date form where you can select a date. 3. If you want to, enter the last date of a DTEND date range or click the Details button to open an Edit Date form where you can select a date.
	Start Time	1. Click the Start Time option box. 2. Enter the earliest time of a DTSTART time range or click the Details button to open an Edit Time form where you can select a time. 3. If you want to, enter the latest time of a DTSTART time range or click the Details button to open an Edit time form where you can select a time.

Step	Action	
8, cont.	**To add this kind of condition...**	**Do this...**
	End Time	1. Click the End Time option box. 2. Enter the earliest time of a DTEND time range or click the Details button to open an Edit Time form where you can select a time. 3. If you want to, enter the latest time of a DTEND time range or click the Details button to open an Edit Time form where you can select a time.
9	Click OK. **Result**: You are returned to the vCalendar Filter Rules Master List form. Your new rule name is in the list and the Rule Contents are summarized. 	
10	To create another rule for the same event type, repeat steps 3 - 9. To create a new rule for another event type, repeat steps 2 - 9.	

Ordering filter rules

If you have created more than one filter rule for an event type, you must put the rules in the order you want them applied to incoming events of that event type.

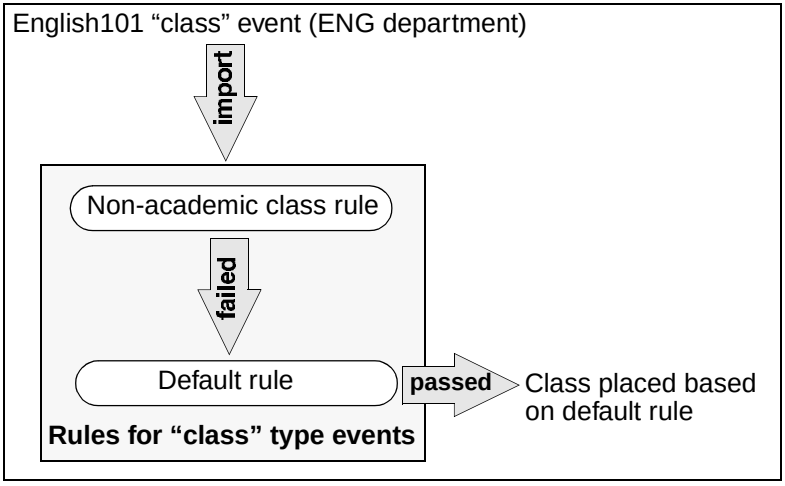
To do this, use the up/down arrows on the vCalendar Filter Rules Master List form to move rule names up or down in the list until you have positioned the rules in the order you want.

The “default filter rule”

Introduction

RESOURCE25 provides a “default filter rule” that is used when you haven’t set up any filter rules for an event type you’re importing or when an imported event fails all the other criteria checks you’ve set up for that event type.

In our previous example on page 7-41, if a “class” event is imported into RESOURCE25 with an account (department) other than “Admin,” it is placed based on the default filter rule. This rule is applied only after the incoming event has failed all other criteria checks for its event type.



Default filter rule criteria

The default filter rule uses the following criteria checks to determine the destination parent event of an incoming event:

- The event type of the parent must be a valid parent event type for the incoming child (based on the X-R25-TYPE property value).
- The date boundaries of the parent must be broad enough to contain the initial meeting of the incoming child (based on the DTSTART and DTEND property values).
- The account associated with the parent must be the same as the account of the incoming child (based on the X-R25-ACCOUNT property value).

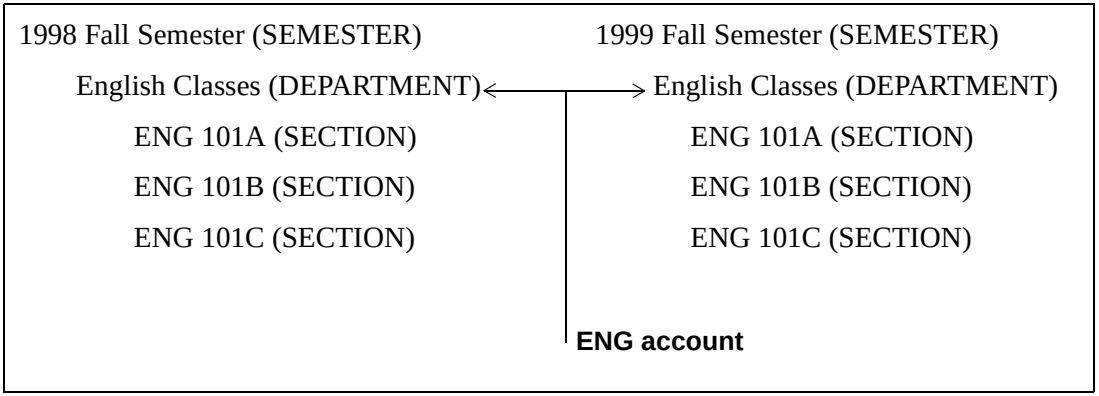
To use the default filter rule, RESOURCE25 must find only *one* event in your event structure that meets the above criteria.

Building your event structure to use the default filter rule

Depending on how your academic event structure is organized, you may be able to use the default filter rule for most or all the events RESOURCE25 will be importing from your SIS. This can eliminate or greatly reduce the need to create filter rules by event type.

Note*: It is recommended that you build your academic event structure to make maximum use of the default filter rule.*

The following example shows an academic event structure that can use the default filter rule:

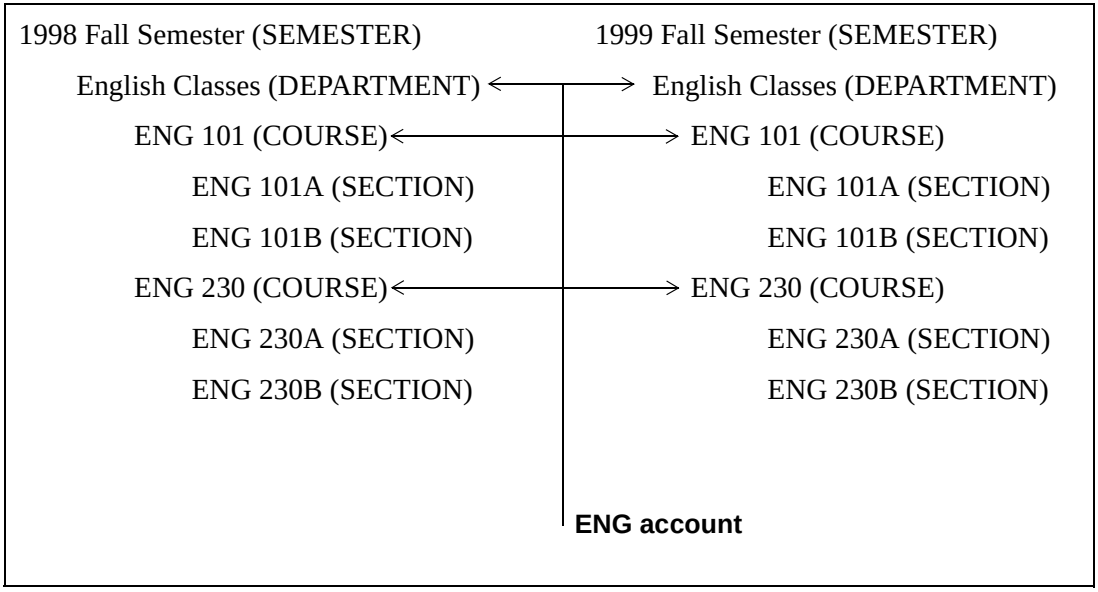


Fall 1999 “section” events imported into this event structure would be able to use the default filter rule because:

- The date boundaries of the 1999 Fall Semester cabinet are broad enough to include the incoming Fall 1999 English classes, and the 1998 Fall Semester cabinet boundaries are not.
- The account associated with the English Classes event (ENG) is only associated with English “department” events.
- There is only one event that meets these criteria - the English Classes event in the 1999 Fall Semester cabinet.

An event structure that can’t use the default filter rule

The following example shows an academic event structure that can’t use the default filter rule:



Fall 1999 events imported into this event structure would not be able to use the default filter rule because:

- The account designation for the English Classes event (ENG) is the same as the account designation for the ENG 101 and ENG 230 events. How does the import choose the right parent? It can’t.
- There are multiple events that meet the criteria for being a parent event.

If an event fails the default filter rule

If an imported event fails the default filter rule (and by implication all the other rules that have been set up for events of its event type), it is not imported into RESOURCE25. It remains in the vCalendar file and RESOURCE25 attempts to import it the next time the import process runs. The import log (see chapter 8) identifies events that could not be imported.

The assumption behind this process is that there may be some missing (out of order) events that would help place this event. It also gives you a chance to correct your event structure and/or your filter rules.

An event might fail the default filter rule for any of these reasons:

- The filter rules you’ve established for its event type are incorrect based on your event structure, and you must change them to ensure proper placement of the event.
- Your event structure does not support use of the default filter rule, and you must either change your event structure or create filter rules for the event type that ensure proper placement of the event.
- The appropriate parent event is not yet in your RESOURCE25 database, and you must re-import the event after it is.

Writing a data exchange policy

Why you need a data exchange policy

Before you begin using your transaction-based interface, it's important that you write a data exchange policy. This assures that everyone involved in academic scheduling on your campus knows that changes to shared event data are initiated by your SIS and who can edit the shared data.

Guidelines for writing your policy

Make sure your data exchange policy specifies:

- **That your SIS "owns" the shared event data and initiates updates of that data.**
  This means that the creation, modification, and cancellation of section events is always initiated by your SIS, never by RESOURCE25. RESOURCE25 simply responds to event changes your SIS initiates.
- **Who can modify shared data.**
  You must determine who can modify shared data in your SIS and who can modify shared data in RESOURCE25. You may have already specified who can modify shared data in RESOURCE25 when you selected the "Who Can Edit" import processing option (see page 7-31).

8 Using Your Transaction-based Interface

Introduction

What's in this chapter

This chapter tells you how to use your transaction-based interface.

It tells you:

- How to initiate the import process
- How the export process is initiated

All vCalendar actions for an event are recorded in the vCalendar transaction log. You can view this log data by selecting the History tab on the Edit Event form to view the History page of an event. (See *Using RESOURCE25* for more information about the History page.)

Chapter contents

Initiating the import process

Introduction

When you're ready to begin importing vCalendar files, you must:

- Specify the import "cycle time," that is, how often you want RESOURCE25 to poll the vCalendar import directories for vCalendar files coming from your SIS.
- Specify the "default organizer" for imported events that don't have an ATTENDEE property.
- Start the process when you're ready to begin importing.

Instructions

Follow these instructions to initiate the vCalendar file import process:

Step	Action
1	From the RESOURCE25 main menu, choose Interface|Import|Import vCalendar files. **Result**: The Import vCalendar Files form opens. 
2	Enter or use the up/down arrows to select how often (in minutes) you want RESOURCE25 to check the vCalendar import directories for vCalendar files.

Step	Action
3	Select a default organizer name from the drop down list. This list includes all active RESOURCE25 users.
4	When you're ready to begin importing events, click Start! **Result**: The import function immediately checks the directories you've specified for request, cancel, and declinecounter objects. If it finds vCalendar files, it imports the associated events and displays their names in the Log area of the form as shown in this example. This log is echoed to the *vcalimp.log* file.  If it cannot import an event, the event remains in the vCalendar file and the problem is reported in the Log area of the form. RESOURCE25 attempts to import the event again the next time the import process runs. **Note**: *While the import is running, you can't perform any other functions in RESOURCE25. You can't close this form until you have stopped the import process.*
5	When you are done importing events, click Stop! **Result**: RESOURCE25 stops polling the import directories. Importing stops.

How the export process is initiated

What initiates the export

What triggers RESOURCE25 to initiate the export of vCalendar files containing reply or counter objects depends on which "To Do Generated" option you've selected on the vCalendar Processing Options form (see page 7-33) and that you've set the "Transaction Cycle" option to "Maintain."

If you've selected:

- "Always," RESOURCE25 initiates the export process after the assigned user completes the To Do item.
- "Only If Action Required" and no action is required, RESOURCE25 initiates the export process immediately after the import.
- "Only If Action Required" and action is required, RESOURCE25 initiates the export process after the assigned user completes the To Do item.
- "Never," RESOURCE25 prompts the assigned user to generate a vCalendar reply or counter when the user changes the event's state to "confirmed."

Appendices and Glossary

Appendix A vCalendar Properties

Introduction

What's in this appendix

This appendix contains technical descriptions and usage information for all supported vCalendar properties including:

- General information
- Definition, usage information, and examples of all the supported vCalendar properties
- Definitions, usage information, and examples of all the supported vEvent properties

The appendix describes Universal Algorithm's implementation of vCalendar for RESOURCE25. It does not represent the complete vCalendar specification, and the complete specification should be read as an accompaniment to it. Additional vCalendar information is available at these world wide web locations:

Internet Mail Corp	**http://www.imc.org/pdi/**
IETF Scheduling Working Group	**http://www.imc.org/ietf-calendar/**

Contents

Overview of vCalendar

RESOURCE25 supports the use of vCalendar - an industry wide data specification for the exchange of event information.

A vCalendar is a data stream consisting of one or more vCalendar components. The individual vCalendar definitions can be identified and parsed within the data stream. RESOURCE25 supports a vCalendar data stream when it exists as a persistent form in a file system. RESOURCE25 can handle vCalendar files containing one event only and vCalendar files containing more than one event.

The vCalendar specification provides for a clear-text encoding for ease of human reading and verification. It also includes a formal grammar for the clear-text encoding to aid in the implementation of parsers and to serve as the definitive reference when ambiguities or questions arise in interpreting the descriptive prose definition of the specification.

Internally, RESOURCE25 uses technical keys to identify every object within the database. These keys are not normally exposed to the user, but form part of the published export stream to allow other systems to respond with references to the exact RESOURCE25 object. The “natural” key information is specified as well to allow the data to be used in external applications.

General information about the supported properties

General information

This is a list of general information about the properties supported in this release:

- Only the default ASCII character set is supported.
- Only the default US English language is supported.
- The RESOURCE25 database works in local time, so all dates and times are exported in local time. On import, dates and times are converted to local time, unless the TZ and DAYLIGHT vCalendar object properties are defined in the import file (see pages A-8 and A-9). You use the Date/Time System Definition to define your local time zone.
- The default placement for all vCalendar property values is inline in the data stream. In certain cases, a URL reference to a local file (file://) is allowed.
- For large text values, RESOURCE25 supports the QUOTED-PRINTABLE format. For simple text values (no hard line breaks), the simple line folding technique is used.
- A non-standard property parameter, `X-R25-ID`, defines the internal RESOURCE25 key value of the corresponding object. Example: LOCATION;X-R25-ID=35:Carnegie Hall

  We recommend that your SIS store these identifiers and use them when returning data to RESOURCE25. The identifiers used by RESOURCE25 are site specific, so care must be taken when importing a file created from another database.

  Some property values are references to items created and stored in RESOURCE25 in either a master list or as a main data object. If on import RESOURCE25 encounters property values with no match in the corresponding RESOURCE25 table, the import process generates an error. If the property includes the X-R25-ID parameter, RESOURCE25 assumes the ID is valid and ignores the text value.

Object property descriptions

Both vCalendar and vEvent objects are made up of “object properties” that define the data in the object.

The tables beginning on the next page describe each of the vCalendar and vEvent properties applicable to vCalendar objects generated by your SIS and to vCalendar objects generated by RESOURCE25. Each property description tells you:

- The name of the property
- The purpose of the property
- The property usage syntax
- Which objects (based on method type) use the property
- What RESOURCE25 data field the property maps to, if any
- How to use the property

The properties are listed in alphabetical order in this appendix. There is no required order of the properties in vCalendar and vEvent objects, except that they must have a BEGIN property at the beginning of the object and an END property at the end of the object.

Properties in SIS-generated vCalendar files

Summary of properties

This table lists each of the vCalendar and vEvent properties that can be present in SIS-generated vCalendar files with page references to their associated descriptions and usage information.

	Property	Purpose	Page
vCalendar	BEGIN	Identifies the beginning of the vCalendar object	A-8
	DAYLIGHT	Defines the dates and subsequent offsets for daylight savings time	A-8
	END	Identifies the end of the vCalendar object	A-8
	METHOD	Identifies the processing method for all events in the vCalendar object	A-9
	PRODID	Identifies the product that created the vCalendar object	A-9
	TZ	Defines the time zone offset for all events in the vCalendar object	A-9
	VERSION	Identifies the supported vCalendar version	A-10
vEvent	ATTENDEE	Identifies the RESOURCE25 contact specified as the event organizer	A-11
	BEGIN	Identifies the beginning of the vEvent object	A-12
	CATEGORIES	Identifies the event categories the event belongs to	A-12
	DCREATED	Identifies the date/time the vEvent object was created	A-12
	DESCRIPTION	Identifies miscellaneous text associated with the event	A-13

	Property	Purpose	Page
vEvent continued	DTEND	Identifies the end date/time of the initial event meeting	A-13
	DTSTART	Identifies the start date/time of the initial event meeting	A-14
	END	Identifies the end of the vEvent object	A-14
	EXDATE	Identifies an exception date	A-15
	EXRULE	Identifies an exception rule	A-16
	LAST-MODIFIED	Identifies the date/time the event information was last modified	A-16
	LOCATION	Identifies the intended event location (space)	A-17
	PRIORITY	Identifies the priority of the event	A-18
	RDATE	Identifies the recurring dates of the event	A-18
	RELATED-TO (parent)	Relates subevents to their associated parent event	A-19
	RELATED-TO (reservation)	Relates multiple reservation definitions to their associated event	A-20
	RESOURCES	Identifies the space preferences of the event	A-21
	RRULE	Identifies a recurring date/time meeting pattern of the event	A-22
	SEQUENCE	Identifies the number of times changes have been made to the vEvent object	A-23
	STATUS	Defines the current status of the event	A-23
	SUMMARY	Specifies the event name and/or a reservation definition name	A-24
	TRANSP	Indicates if the event is “transparent” or not	A-25

	Property	Purpose	Page
vEvent continued	UID	Identifies the event's persistent, globally-unique identifier	A-26
	X-R25-ACCOUNT	Identifies the account sponsor(s) of the event	A-27
	X-R25-HEADCOUNT	Specifies the head count for the event	A-27
	X-R25-TYPE	Identifies the event type of the event	A-28

vCalendar object property descriptions

The table beginning below describes each of the properties that must/can be included in vCalendar objects generated by your SIS - Requests, Cancels, and Declinecounters. The properties are listed in alphabetical order. See the tables of RESOURCE25-generated properties beginning on page A-29 for additional information.

Note: *Required vCalendar properties are shown in bold in the table. If the syntax of a property is shown without examples, you must enter the property and property value exactly as shown.*

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples**
BEGIN	Identifies the beginning of the vCalendar object.	N/A	**Syntax:** BEGIN:VCALENDAR
DAYLIGHT	Defines the dates and subsequent offsets for daylight savings time	N/A	**Syntax:** DAYLIGHT:TRUE;offset; date/time daylight savings time begins; date/time daylight savings time ends; standard time designation; daylight savings time designation DAYLIGHT:FALSE **Examples:** DAYLIGHT:TRUE;07;19980405T020000; 19981025T020000;PST;PDT DAYLIGHT:FALSE
	How to use:	When you define this property with the TZ property (see next page), it overrides any time zone information stored in RESOURCE25 for this particular vCalendar object.	
END	Identifies the end of the vCalendar object.	N/A	**Syntax:** END:VCALENDAR

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples
METHOD	Identifies the processing method for all events in the vCalendar object	N/A	**Syntax:** METHOD:method type **Example:** METHOD:REQUEST
	How to use:	Possible values for objects generated by your SIS: **REQUEST** - The object defines a set of new or modified events. **CANCEL** - The object defines a set of cancelled events. **DECLINECOUNTER** - The object is used to decline proposed event data changes (room assignments) made in RESOURCE25. The method type specifies how all events in the vCalendar object should be processed. So, for example, all events in a vCalendar object with a "cancel" method type are cancelled when imported into RESOURCE25.	
PRODID	Identifies the product that created the vCalendar object (your SIS)	N/A	**Syntax:** PRODID:ISO 9070 value **Example:** PRODID:- / /SIS vendor/ /NONSGML My SIS/ /EN
	How to use:	The value should follow the ISO 0970 Formal Public Identifier standard	
TZ	Defines the time zone offset for all events in the vCalendar object	N/A	**Syntax:** TZ:offset **Example:** TZ:-08
	How to use:	When you specify this property with the DAYLIGHT property, it overrides any time zone information stored in RESOURCE25 for this particular vCalendar object. When RESOURCE25 imports the object, it converts any date/time in local time format (that is, without an explicit time zone offset or "Z" suffix) to the local time zone of the RESOURCE25 database by using the TZ offset and any DAYLIGHT rules.	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples**
VERSION	Identifies the supported vCalendar version	N/A	**Syntax:** VERSION:1.0
	How to use:	The value must be 1.0.	

vEvent object property descriptions

The table beginning below describes each of the properties that must/can be included in vEvent objects in Request, Cancel, and Declinecounter vCalendar objects generated by your SIS. The properties are listed in alphabetical order. See the tables of RESOURCE25-generated properties beginning on page A-29 for additional information.

Note: *Required vEvent properties are shown in bold in the table. Optional parameters are shown in square brackets in the syntax. If the syntax of a property is shown without examples, you must enter the property and property value exactly as shown.*

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
ATTENDEE	Identifies the RESOURCE25 contact specified as the event organizer	Event Organizer	**Syntax:** ATTENDEE;ROLE=ORGANIZER [;X-R25-ID=R25 identifier]: RFC822 email address **Example:** ATTENDEE;ROLE=ORGANIZER:jdoe@myu.edu ATTENDEE;ROLE=ORGANIZER; X-R25-ID=2:tsmith@myu.edu
	Used in:	Request objects	
	How to use:	You use this property to identify the event organizer, the scheduler currently in charge of processing the event within RESOURCE25. The person specified by this property must be a RESOURCE25 user. The property value you specify is his/her email address. You must always specify the ROLE=ORGANIZER property value. It is recommended that you also include the X-R25-ID parameter to uniquely identify the contact. If you don't include the X-R25-ID parameter, on import the property value is validated against the contact's email address in the RESOURCE25 contacts database. This means that the contact's email address must be present and up-to-date in RESOURCE25. If you don't include this property or the person you specify is not a RESOURCE25 user, the "default organizer" specified using the RESOURCE25 Import vCalendar Files form prior to import will be designated as the event organizer.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
BEGIN	Identifies the beginning of the vEvent object	N/A	**Syntax:** BEGIN:VEVENT
	Used in:	Request, Cancel, and Declinecounter objects. *Required.*	
CATEGORIES	Identifies the event categories the event belongs to	Event Categories	**Syntax:** CATEGORIES:category name [;category name] CATEGORIES[;X-R25-ID=R25 identifier]: category name **Examples:** CATEGORIES:FACULTY ONLY CATEGORIES:FEE PAYING;FACULTY ONLY CATEGORIES;X-R25-ID=10:OPEN TO PUBLIC
	Used in:	Request objects	
	How to use:	You can include several category properties in a vEvent object, each specifying one event category, or you can include one property and separate each category name with a semi-colon. The vCalendar specification defines a set of standard category phrases. To include values other than those, you must define a X-R25-ID parameter. If you use the X-R25-ID parameter, put each event category on a separate line.	
DCREATED	Identifies the date/time the vEvent object was created	Creation Date	**Syntax:** DCREATED:yyyymmddThhmmss **Example:** DCREATED:19980314T153200
	Used in:	Request objects	
	How to use:	If you don't include this property, RESOURCE25 uses the date/time of the import.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
DESCRIPTION	Identifies miscellaneous text associated with the event	Description (text in event record)	**Syntax:** DESCRIPTION:text DESCRIPTION;VALUE=URL:file://pathname **Examples:** DESCRIPTION:This event discusses the role of the author in shaping modern society. DESCRIPTION;VALUE=URL:file://C:\evdec1.txt
	Used in:	Request objects	
	How to use:	You can enter a text string or a file location using the VALUE parameter. RESOURCE25 imports the text description or the contents of the specified file as event text (the text that appears when you select the Text tab on the Edit Event form).	
DTEND	Identifies the end date/time of the initial event meeting	End Date and Time of first event occurrence	**Syntax:** DTEND[;X-R-25-TAKEDOWN=PTnumber of hours and/or minutes]:yyyymmddThhmmss **Examples:** DTEND:19980314T090000 DTEND;X-R25-TAKEDOWN=PT15M: 19980314T090000
	Used in:	Request objects	
	How to use:	For this property, enter the end date/time of the actual event, or, for a recurring event (one that has either an RDATE or RRULE property), enter the end date/time of the first occurrence in the series. If the event has takedown time, specify it with the extension parameter X-R25-TAKEDOWN and a time period. See also DTSTART.	

Property **(bold=required)**	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
DTSTART	Identifies the start date/time of the initial event meeting	Start Date and Time of first event occurrence	**Syntax:** DTSTART[;X-R-25-SETUP=PTnumber of hours and/or minutes]:yyyymmddThhmmss **Examples:** DTSTART:19980314T080000 DTSTART;X-R25-SETUP=PT1H30M:19980314T080000
	Used in:	Request objects	
	How to use:	For this property, enter the start date/time of the actual event, or, for a recurring event (one that has either an RDATE or RRULE property), enter the start date/time of the first occurrence in the series. If the event has setup time, specify it with the extension parameter X-R25-SETUP and a time period. The DTSTART and DTEND properties relate to the actual meeting times of an event. RESOURCE25 also uses the concept of date boundaries to define the first and last possible dates within which an event must take place. These boundary dates are inherited from the parent event.	
END	Identifies the end of the vEvent object	N/A	**Syntax:** END:VEVENT
	Used in:	Request, Cancel, and Declinecounter objects. *Required.*	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
EXDATE	Identifies an exception date	Exclusion Constraint (specific date/time)	**Syntax:** EXDATE:yyyymmddThhmmss [;yyyymmddThhmmss] **Example:** EXDATE:19980904T080000;19980911T080000
	Used in:	Request objects	
	How to use:	This property is used to define individual exceptions to a pattern of dates specified in a recurrence rule. Multiple exception dates must be separated by semi-colons. The date/times defined by this property must correspond to the start date/times of the meeting pattern. This means that there is not necessarily an exact match to data defined in RESOURCE25 where exceptions are defined as a range of dates and times, and any meeting falling within that range is excluded. The import process ignores this property for an existing event when there would have been no reservation for the excluded date. Otherwise, if a reservation exists, a specific date exclusion constraint is created to cover the duration of the meeting. No attempt is made to recreate some pattern out of the series of dates. Any assigned space is released for the duration of the exclusion period. If date constraints are being inherited by events being imported, you don't have to include an EXDATE property for those events.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
EXRULE	Identifies an exception rule	Exclusion Constraint (whole day, repeating)	**Syntax:** EXRULE:interval frequency [days,months affected] [number of occurrences] **Examples:** EXRULE:W2 TH //except every other week on Thursday EXRULE:D1 #3 //except daily for 3 occurrences
	Used in:	Request objects	
	How to use:	This property is equivalent to a RESOURCE25 exclusion constraint (repeating pattern of date exclusions) with the added restriction that it applies to a complete day. Therefore, when RESOURCE25 imports a new rule, it sets the time constraint to start at midnight and end at 11:59 PM. The rule is also assumed to be in effect for the entire period of the event, unless number of occurrences is specified. If the start and end times of a constraint are a subset of a day, use the EXDATE property instead.	
LAST-MODIFIED	Identifies the date/time the event information was last modified	Last Modified Date	**Syntax:** LAST-MODIFIED:yyyymmddThhmmss **Example:** LAST-MODIFIED:19980204T142700
	Used in:	Request objects	
	How to use:	This property is used to indicate the date and time the event information was last modified. When imported into RESOURCE25, it appears in the Last Modified field on the Identification page of the Edit Event form.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
LOCATION	Identifies the intended event location (space)	Assigned Space	**Syntax:** LOCATION[;X-R25-ID=R25 identifier] [;X-R25-LAYOUT=layout name]:location name **Examples:** LOCATION:MM 202 LOCATION;X-R25-ID=35:LT 450 LOCATION;X-R5-LAYOUT=Circular:MM 350
	Used in:	Request objects	
	How to use:	You use this property to specify the space indicated for the event in the SIS database. You can use multiple properties, one per space, to specify several spaces for the same event. It is recommended that you also use the X-R25-ID property parameter to explicitly identify the space to RESOURCE25, instead of relying on the text value. If you include this property, on import RESOURCE25 attempts to assign the designated space(s) to the event. If it can assign the space, it sends a success Reply object to your SIS. If the space in unavailable, RESOURCE25 generates an assign room To Do for the event organizer. When the event organizer completes the To Do, RESOURCE25 sends a Counter object with the new space assignment to your SIS for update. The absence of this property in a Request alerts RESOURCE25 that you want to use RESOURCE25 to assign space to the event. On import, RESOURCE25 generates an assign room To Do for the event organizer. When the event organizer completes the To Do, RESOURCE25 sends a Counter object with the space assignment to your SIS for update. RESOURCE25 allows a room's setup configuration to be stored as part of its space assignment. If you want to send this information to RESOURCE25, use the X-R25-LAYOUT property parameter. The layout name must match a valid layout for the space. If it doesn't or you don't specify the parameter, the space's default layout is used.	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
PRIORITY	Identifies the priority of the event	Event Priority	**Syntax:** PRIORITY:number **Example:** PRIORITY:1
	Used in:	Request objects	
	How to use:	This property is used to specify the priority of an event. *It must be included in all events in request objects.*	
RDATE	Identifies the recurring dates of the event	Reservation Definitions (repeating ad hoc events)	**Syntax:** RDATE:yyyymmddThhmmss; yyyymmddThhmmss **Example:** RDATE:19980509T080000;19980523T080000
	Used in:	Request objects	
	How to use:	This property is used to identify an ad-hoc series of dates/times when event meetings take place. The start and end times of the meetings are defined by the DTSTART and DTEND properties. The time portion of the recurring date values must match the start time of the DTSTART property. Each date/time specified by this property is stored as a reservation definition in RESOURCE25.	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
RELATED-TO (Parent)	Relates subevents to their associated parent event	Parent relationship	**Syntax:** RELATED-TO;X-R25-REL=PARENT: UID of parent event **Example:** RELATED-TO;X-R25-REL=PARENT: MySIS-CRN/003172
	Used in:	Request objects	
	How to use:	This property is used to relate subevents to their parent event. By using this property, you can send a parent event and its subevents to RESOURCE25 in one vCalendar object and ensure that the parent/child event relationships will be properly maintained when the events are imported into your RESOURCE25 event structure. The associated event types of the events must conform to your RESOURCE25 event type relationships hierarchy. The parent vEvent object must include a TRANSP property (see page A-25). For example, event A is the parent event of events A1, A2, and A3. These will be imported into RESOURCE25 as a parent event and subevents if the vEvent objects A1, A2, and A3 each include a parent RELATED-TO property to vEvent object A. You can use the RELATED-TO property with the PARENT parameter to avoid having to set up or use vCalendar filter rules (described in chapter 7).	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
RELATED-TO (Reservation)	Relates multiple reservation definitions to their associated event	Reservation relationship	**Syntax:** RELATED-TO;X-R25-REL=RESERVATION: UID of "group" event **Example:** RELATED-TO;X-R25-REL=RESERVATION: MySIS-CRN/004556
	Used in:	Request objects	
	How to use:	This property is used to relate multiple reservation definitions to their associated event. If an event has multiple reservation definitions, the vCalendar object should contain a "group event" that becomes the event when imported into RESOURCE25, and several "reservation events," one for each of the event's reservation definitions. These "reservation events" are imported into RESOURCE25 as the reservation definitions of the event. The "group event" vEvent object must include a TRANSP property (see page A-25). For example, event B has meeting patterns B1, B2, and B3. These will be imported into RESOURCE25 as a single event if the vEvent objects B1, B2, and B3 each include a reservation RELATED-TO property to vEvent object B. All the properties of B1, B2, and B3 will be ignored (because they are defined in B) except the date related values and the UID.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
RESOURCES	Identifies the space preferences of the event	Space Preferences	**Syntax:** RESOURCES; X-R25-TYPE=SPACE; X-R25-USE=PREF; X-R25-SUBTYPE=space preference type [;X-R25-PRIORITY=priority of preference] [;X-R25-ID=R25 identifier]: **Example:** RESOURCES; X-R25-TYPE=SPACE;X-R25-USE=PREF; X-R25-SUBTYPE=CATEGORY; X-R25-PRIORITY=1; X-R25-ID=4:NORTH QUAD RESOURCES; X-R25-TYPE=SPACE;X-R25-USE=PREF; X-R25-SUBTYPE=FEATURE; X-R25-QUANTITY=20:CHAIRS
	Used in:	Request objects	
	How to use:	This property is used to indicate an event's space preferences. Valid property parameters are: **X-R25-TYPE** - must be "SPACE" **X-R25-USE** - must be "PREF" **X-R25-SUBTYPE** - identifies the type of space preference. Valid values are "CATEGORY," "FEATURE," and "LAYOUT." There is no default value for subtype and it must be present. **X-R25-PRIORITY** - identifies the priority of the preference. Default is 1. **X-R25-QUANTITY** - identifies the desired quantity of a space feature preference. The default is 1. This parameter is ignored for space category and space layout preferences. The property value you enter is the name of the corresponding space category, space feature, or space layout. It is recommended that you also include an X-R25-ID parameter to uniquely identify the item to RESOURCE25. Multiple property values can be specified on the same resource record if they all have the same set of parameter values. Separate the values with a semi-colon.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
RRULE	Identifies a recurring date/time meeting pattern of the event	Reservation Definitions (recurring date/ time meeting pattern)	**Syntax:** RRULE:interval frequency [days,months affected] [number of occurrences or event end date] **Examples:** RRULE:W1 MO WE FR 19981219T180000 //Once a week on Mondays, Wednesdays, and Fridays until 12/19/98 RRULE:W1 TH #10 //Once a week on Thursdays for 10 occurrences
	Used in:	Request objects	
	How to use:	This property is used to define a pattern for a series of event meetings. It requires a DTSTART and DTEND property to define the start and end of the initial meeting. The start and end values must be valid for the rule. Currently, vCalendar supports only one rule per vEvent object, so if an event has multiple recurring reservation definition patterns, include one vEvent object for each and use the RELATED-TO property to link them to the event. Include either the event end date or number of occurrences in the definition to indicate the last date of the rule, or the event's end boundary date (typically inherited from the parent event) is used.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
SEQUENCE	Identifies the number of times changes have been made to the vEvent object	Version Number	**Syntax:** SEQUENCE:sequence number **Example:** SEQUENCE:17
	Used in:	Request, Cancel, and Declinecounter objects	
	How to use:	This property is used to identify the revision number of the vEvent object. *It must be included in all events in request, cancel, and declinecounter objects.* On import, RESOURCE25 ignores vEvent objects with a sequence number that is less than or equal to the value in the current RESOURCE25 event record. A value of 0 (zero) indicates a new event. If the property is missing, 0 (zero) is assumed. If you have several related vEvent objects for one event (several date definitions), use the same sequence number in each vEvent object.	
STATUS	Defines the current status of the event	Event State	**Syntax:** STATUS:event state **Example:** STATUS:TENTATIVE
	Used in:	Request and Cancel objects	
	How to use:	This property is used to indicate the event's status. *It must be included in all events in request and cancel objects.* Each of the possible values corresponds to a valid event state in RESOURCE25. When a vEvent object is imported to an existing event in RESOURCE25, a new event history record is created if the status values do not match.	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
SUMMARY	Specifies the event name and/or a reservation definition name	Event Name Reservation Definition Name	**Syntax:** SUMMARY:event name SUMMARY:definition name SUMMARY:event name;definition name **Example:** SUMMARY:CS102A SUMMARY:Lectures SUMMARY:CS101D;Class Schedule
	Used in:	Request objects	
	How to use:	An event may have several different meeting patterns associated with it. To be able to easily identify them in a vCalendar object where they are linked using the reservation RELATED-TO property, specify the event name as the SUMMARY value for the “group event” and the appropriate reservation definition name as the SUMMARY value for each “reservation event.” If there is only one meeting pattern for an event, include both the event name and the definition name in the property value, separated by a semi-colon. If you don’t include both, RESOURCE25 uses the event name as the definition name too.	

Property (bold=required)	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
TRANSP	Indicates if the event is "transparent" or not	N/A	**Syntax:** TRANSP:number **Example:** TRANSP:1
	Used in:	Request and Cancel objects	
	How to use:	This property is used in conjunction with the RELATED-TO property described on pages A-19 and A-20. To use the property, specify a value equal to or greater than 1 (one). When used in a parent or group event, this property indicates that the vEvent object is "transparent," that is, that the start and end dates of the parent or group event should not be treated as reservation dates and a date definition should not be created for them. The expectation is that the DTSTART and DTEND dates in the subsequent vEvent objects linked to the parent or group event using the RELATED-TO property contain actual reservation definitions. The absence of the property is equivalent to TRANSP:0 and means that the dates defined in the vEvent object are actual meeting patterns of the event. See the vCalendar object example on page 7-10.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
UID	Identifies the event's persistent, globally-unique identifier	Used internally	**Syntax:** UID[;X-R25-ID=R25 identifier]:SIS identifier **Examples:** UID:MySIS/CRN/234780 UID;X-R25-ID=2542:MySIS/CRN/234780 UID;X-R25-ID=2542:MySIS/CRN/234783/1
	Used in:	Request, Cancel, and Declinecounter objects	
	How to use:	This property must contain an identifier that uniquely identifies the event in your SIS and will uniquely identify the event to RESOURCE25. *It must be included in all events in request, cancel, and declinecounter objects.* The identifier must be unique across all database instances. For changes to an event made after an the event has already been imported into RESOURCE25 and a reply or counter generated, you should also use the X-R25-ID parameter to specify the RESOURCE25 internal identifier of the event. In order to provide a globally unique value, this property must use more than just the internal identifier of the RESOURCE25 event. In cases where several vEvent objects may be required to represent a single RESOURCE25 event, each vEvent must have a unique UID. RESOURCE25 only stores the first 240 characters of any UID created by an external system. It uses the UID to determine if it has seen the event object before, so external systems should ensure that the first 240 characters are unique.	

Property (bold=required)	Purpose	R25 Data Field	Syntax and Examples [ ] = optional parameter
X-R25-ACCOUNT	Identifies the account sponsor(s) of the event	Account	**Syntax:** X-R25-ACCOUNT[;X-R25-ID=R25 identifier]: account name **Examples:** X-R25-ACCOUNT:ENGLISH X-R25-ACCOUNT;X-R25-ID=254:ENGLISH
	Used in:	Request objects	
	How to use:	This property identifies the sponsoring organization(s) of the event. It should be the same for all vEvent objects that relate to the same RESOURCE25 event. The RESOURCE25 specific property parameter X-R25-ID should be used to explicitly identify the account, instead of relying on the text value. To take advantage of the vCalendar import default filter rule, (see page 7-48) each vEvent object should contain this property.	
X-R25-HEADCOUNT	Specifies the head count for the event	Expected or Registered Head Count	**Syntax:** X-R25-HEADCOUNT [;X-R25-TYPE=headcount type]: head count number **Examples:** X-R25-HEADCOUNT:35 X-R25-HEADCOUNT; X-R25-TYPE=REGISTERED:50
	Used in:	Request objects	
	How to use:	This property specifies the expected or registered head count for the event. It should be the same for all vEvent objects that relate to the same RESOURCE25 event. An optional parameter, X-R25-TYPE, can be used to indicate the type of head count. Supported values are “EXPECTED” and “REGISTERED.” The default value is “EXPECTED.” If both values are known, enter two separate head count properties, one for the expected head count and one for the registered head count.	

Property **(bold=required)**	**Purpose**	**R25 Data Field**	**Syntax and Examples** [ ] = optional parameter
X-R25-TYPE	Identifies the event type of the event	Event Type	**Syntax:** X-R25-TYPE[;X-R25-ID=R25 identifier]: event type **Example:** X-R25-TYPE:Section X-R25-TYPE;X-R25-ID=5:Section
	Used in:	Request objects	
	How to use:	This property identifies the RESOURCE25 event type of the event. *The property must be specified for new events that are unknown to RESOURCE25*, and the value must conform to the structure of your event type relationships hierarchy. It is also recommended that you include the X-R25-ID parameter to identify the event type to RESOURCE25 without relying on the text value.	

Properties in RESOURCE25-generated vCalendar files

Summary of properties

This table lists each of the vCalendar and vEvent properties that can be present in RESOURCE25-generated vCalendar files with page references to their associated descriptions and usage information.

	Property	Purpose	Page
vCalendar	BEGIN	Identifies the beginning of the vCalendar object	A-32
	DAYLIGHT	Defines the dates and subsequent offsets for daylight savings time	A-32
	END	Identifies the end of the vCalendar object	A-32
	METHOD	Identifies the processing method for all events in the vCalendar object	A-33
	PRODID	Identifies the product that created the vCalendar object	A-33
	TZ	Defines the time zone offset for all events in the vCalendar object	A-33
	VERSION	Identifies the supported vCalendar version	A-33
vEvent	ATTENDEE	Identifies the RESOURCE25 contact specified as the event organizer	A-34
	BEGIN	Identifies the beginning of the vEvent object	A-34
	CATEGORIES	Identifies the event categories the event belongs to	A-34
	DCREATED	Identifies the date/time the vEvent object was created	A-34

	Property	Purpose	Page
vEvent continued	DESCRIPTION	Identifies miscellaneous text associated with the event	A-35
	DTEND	Identifies the end date/time of the initial event meeting	A-35
	DTSTART	Identifies the start date/time of the initial event meeting	A-35
	END	Identifies the end of the vEvent object	A-35
	EXDATE	Identifies an exception date	A-36
	EXRULE	Identifies an exception rule	A-36
	LAST-MODIFIED	Identifies the date/time the event information was last modified	A-36
	LOCATION	Identifies the intended event location (space)	A-37
	PRIORITY	Identifies the priority of the event	A-37
	RDATE	Identifies the recurring dates of the event	A-37
	RELATED-TO (parent)	Relates subevents to their associated parent event	A-37
	RELATED-TO (reservation)	Relates multiple reservation definitions to their associated event	A-38
	RESOURCES	Identifies the space preferences of the event	A-38
	RRULE	Identifies a recurring date/time meeting pattern of the event	A-39
	SEQUENCE	Identifies the number of times changes have been made to the vEvent object	A-39
	STATUS	Defines the current status of the event	A-39
	SUMMARY	Specifies the event name and/or a reservation definition name	A-39

	Property	Purpose	Page
vEvent continued	TRANSP	Indicates if the event is “transparent” or not	A-40
	UID	Identifies the event’s persistent, globally-unique identifier	A-40
	X-R25-ACCOUNT	Identifies the account sponsor(s) of the event	A-40
	X-R25-HEADCOUNT	Specifies the head count for the event	A-41
	X-R25-REQUEST-STATUS	Specifies the status of a reply to a request	A-41
	X-R25-TYPE	Identifies the event type of the event	A-41

vCalendar object property descriptions

The table below and on the next page describes each of the properties that must/can be included in vCalendar objects generated by RESOURCE25 - Replies, Counters, and Publishes. The properties are listed in alphabetical order. See the tables of SIS-generated properties beginning on page A-5 for additional information.

.

Property	Purpose	R25 Data Field	Syntax and Examples
BEGIN	Identifies the beginning of the vCalendar object.	N/A	**Syntax:** BEGIN:VCALENDAR
DAYLIGHT	Defines the dates and subsequent offsets for daylight savings time	N/A	**Syntax:** DAYLIGHT:TRUE;offset; date/time daylight savings time begins; date/time daylight savings time ends; standard time designation; daylight savings time designation DAYLIGHT:FALSE **Examples:** DAYLIGHT:TRUE;07;19980405T020000; 19981025T020000;PST;PDT DAYLIGHT:FALSE
END	Identifies the end of the vCalendar object.	N/A	**Syntax:** END:VCALENDAR

Property	Purpose	R25 Data Field	Syntax and Examples
METHOD	Identifies the processing method for all events in the vCalendar object	N/A	**Syntax:** METHOD:method type **Example:** METHOD:REPLY
	How used:	Possible values for objects generated by RESOURCE25: **REPLY** - The object is a confirmation response to a request. **COUNTER** - The object is a response to a request that contains new or changed information. The SIS should respond with a new request or a declinecounter. When RESOURCE25 publishes events, it does not include the METHOD property in the vCalendar object. Publish is the default and is assumed.	
PRODID	Identifies the product that created the vCalendar object (RESOURCE25)	N/A	**Syntax:** PRODID:ISO 9070 value **Example:** PRODID:- / /Universal Algorithms Inc// NONSGML Resource25//EN
TZ	Defines the time zone offset for all events in the vCalendar object	N/A	**Syntax:** TZ:offset **Example:** TZ:-08
VERSION	Identifies the supported vCalendar version	N/A	**Syntax:** VERSION:1.0
	How used:	Is always 1.0.	

vEvent object property descriptions

The table beginning below describes each of the properties that must/can be included in vEvent objects in Reply, Counter, and Publish vCalendar objects generated by RESOURCE25. The properties are listed in alphabetical order. See the tables of SIS-generated properties beginning on page A-5 for additional information.

Property	Purpose	R25 Data Field	Syntax and Examples
ATTENDEE	Identifies the RESOURCE25 contact specified as the event organizer	Event Organizer	**Syntax:** ATTENDEE;ROLE=ORGANIZER; X-R25-ID=R25 identifier: RFC822 email address **Example:** ATTENDEE;ROLE=ORGANIZER; X-R25-ID=2:tsmith@myu.edu
	Used in:	Counter and Publish objects	
BEGIN	Identifies the beginning of the vEvent object	N/A	**Syntax:** BEGIN:VEVENT
	Used in:	Reply, Counter, and Publish objects	
CATEGORIES	Identifies the event categories the event belongs to	Event Categories	**Syntax:** CATEGORIES; X-R25-ID=R25 identifier:category name **Examples:** CATEGORIES;X-R25-ID=10:OPEN TO PUBLIC
	Used in:	Counter and Publish objects	
DCREATED	Identifies the date/time the vEvent object was created	Creation Date	**Syntax:** DCREATED:yyyymmddThhmmss **Example:** DCREATED:19980314T153200
	Used in:	Counter and Publish objects	
	How used:	If this property was absent from the request, RESOURCE25 uses the date/time of the import.	

Property	Purpose	R25 Data Field	Syntax and Examples
DESCRIPTION	Identifies miscellaneous text associated with the event	Description (text in event record)	**Syntax:** DESCRIPTION:text DESCRIPTION;VALUE=URL:file://<pathname> **Examples:** DESCRIPTION:This event discusses the role of the author in shaping modern society. DESCRIPTION;VALUE=URL:file://C:\evdec1.txt
	Used in:	Counter and Publish objects	
	How used:	RESOURCE25 defaults to line folding for inline values on export with a QUOTED-PRINTABLE encoding.	
DTEND	Identifies the end date/time of the initial event meeting	End Date and Time of first event occurrence	**Syntax:** DTEND:yyyymmddThhmmss DTEND;X-R-25-TAKEDOWN=PTnumber of hours and/or minutes:yyyymmddThhmmss **Examples:** DTEND:19980314T090000 DTEND;X-R25-TAKEDOWN=PT15M: 19980314T090000
	Used in:	Counter and Publish objects	
DTSTART	Identifies the start date/time of the initial event meeting	Start Date and Time of first event occurrence	**Syntax:** DTSTART:yyyymmddThhmmss DTSTART;X-R-25-SETUP=PTnumber of hours and/or minutes:yyyymmddThhmmss **Examples:** DTSTART:19980314T080000 DTSTART;X-R25-SETUP=PT1H30M: 19980314T080000
	Used in:	Counter and Publish objects	
END	Identifies the end of the vEvent object	N/A	**Syntax:** END:VEVENT
	Used in:	Reply, Counter, and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
EXDATE	Identifies an exception date	Exclusion Constraint (specific date/time)	**Syntax:** EXDATE:yyyymmddThhmmss EXDATE:yyyymmddThhmmss; yyyymmddThhmmss **Example:** EXDATE:19980904T080000;19980911T080000
	Used in:	Counter and Publish objects	
EXRULE	Identifies an exception rule	Exclusion Constraint (whole day, repeating)	**Syntax:** EXRULE:interval frequency days,months affected number of occurrences **Examples:** EXRULE:W2 TH //except every other week on Thursday EXRULE:D1 #3 //except daily for 3 occurrences
	Used in:	Counter and Publish objects	
LAST-MODIFIED	Identifies the date/time the event information was last modified	Last Modified Date	**Syntax:** LAST-MODIFIED:yyyymmddThhmmss **Example:** LAST-MODIFIED:19980204T142700
	Used in:	Counter and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
LOCATION	Identifies the event location (space) assigned in RESOURCE25	Assigned Space	**Syntax:** LOCATION;X-R25-ID=R25 identifier; X-R25-LAYOUT=layout name: location name **Examples:** LOCATION;X-R25-ID=35; X-R25-LAYOUT=Circular:LT 450
	Used in:	Counter and Publish objects	
	How used:	If this property was absent from the request, RESOURCE25 assumes that you want to use RESOURCE25 to assign space to the event. After a scheduler has assigned a space and completed the associated To Do, RESOURCE25 issues a counter object with the space assignment.	
PRIORITY	Identifies the priority of the event	Event Priority	**Syntax:** PRIORITY:number **Example:** PRIORITY:0
	Used in:	Counter and Publish objects	
RDATE	Identifies the recurring dates of the event	Reservation Definitions (repeating ad hoc events)	**Syntax:** RDATE:yyyymmddThhmmss; yyyymmddThhmmss **Example:** RDATE:19980509T080000;19980523T080000
	Used in:	Counter and Publish objects	
RELATED-TO (Parent)	Relates subevents to their associated parent event	Parent relationship	**Syntax:** RELATED-TO;X-R25-REL=PARENT: UID of parent event **Example:** RELATED-TO;X-R25-REL=PARENT: MySIS-CRN/003172
	Used in:	Counter and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
RELATED-TO (Reservation)	Relates multiple reservation definitions to their associated event	Reservation relationship	**Syntax:** RELATED-TO;X-R25-REL=RESERVATION: UID of "group" event **Example:** RELATED-TO;X-R25-REL=RESERVATION: MySIS-CRN/004556
	Used in:	Counter and Publish objects	
RESOURCES	Identifies the space preferences of the event	Space Preferences	**Syntax:** RESOURCES; X-R25-TYPE=SPACE; X-R25-USE=PREF; X-R25-SUBTYPE=preference type; X-R25-QUANTITY=feature quantity; X-R25-ID=R25 identifier **Example:** RESOURCES; X-R25-TYPE=SPACE;X-R25-USE=PREF; X-R25-SUBTYPE=CATEGORY; X-R25-PRIORITY=1; X-R25-ID=4:NORTH QUAD
	Used in:	Counter and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
RRULE	Identifies a recurring date/time meeting pattern of the event	Reservation Definitions (recurring date/ time meeting pattern)	**Syntax:** RRULE:interval frequency days,months affected number of occurrences or event end date **Examples:** RRULE:W1 MO WE FR #19981219T180000 //Once a week on Mondays, Wednesdays, and Fridays until 12/19/98 RRULE:W1 TH #10 //Once a week on Thursdays for 10 occurrences
	Used in:	Counter and Publish objects	
	How used:	The current vCalendar specification only supports one rule per vEvent object, so if there are multiple reservation definitions in RESOURCE25, they are exported as multiple vEvent objects.	
SEQUENCE	Identifies the number of times changes have been made to the vEvent object	Version Number	**Syntax:** SEQUENCE:sequence number **Example:** SEQUENCE:17
	Used in:	Reply, Counter, and Publish objects	
STATUS	Defines the current status of the event	Event State	**Syntax:** STATUS:event state **Example:** STATUS:TENTATIVE
	Used in:	Counter and Publish objects	
SUMMARY	Specifies the event name and/or a reservation definition name	Event Name Reservation Definition Name	**Syntax:** SUMMARY:event name SUMMARY:definition name SUMMARY:event name;definition name **Example:** SUMMARY:CS102A SUMMARY:Lectures SUMMARY:CS101D;Class Schedule
	Used in:	Counter and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
TRANSP	Indicates if the event is "transparent" or not	N/A	**Syntax:** TRANSP:number **Example:** TRANSP:1
	Used in:	Counter and Publish objects	
UID	Identifies the event's persistent, globally-unique identifier	Used internally	**Syntax:** UID;X-R25-ID=R25 identifier:UAI/R25/ R25 license number/event locator/ date definition ID **Example:** UID;X-R25-ID=99:UAI/R25/1/1998-AAAADY/1
	Used in:	Reply, Counter, and Publish objects	
	How used:	The structure of the event identifiers created by RESOURCE25 is: *UAI/R25/license number/event locator/date definition id* RESOURCE25 includes the internal event identifier in the X-R25-ID parameter to allow external systems to easily reference a RESOURCE25 event. The date definition sequence is optional when referencing an event in general. Your SIS should not base any processing on this format or assume any hidden meaning in the UID value.	
X-R25-ACCOUNT	Identifies the account sponsor(s) of the event	Account	**Syntax:** X-R25-ACCOUNT;X-R25-ID=R25 identifier: account name **Examples:** X-R25-ACCOUNT;X-R25-ID=254:ENGLISH
	Used in:	Counter and Publish objects	

Property	Purpose	R25 Data Field	Syntax and Examples
X-R25-HEADCOUNT	Specifies the head count for the event	Expected or Registered Head Count	**Syntax:** X-R25-HEADCOUNT; X-R25-TYPE=REGISTERED: head count number **Examples:** X-R25-HEADCOUNT:35 X-R25-HEADCOUNT; X-R25-TYPE=REGISTERED:50
	Used in:	Counter and Publish objects	
X-R25-REQUEST-STATUS	Specifies the status of a reply to a request	N/A	**Syntax:** X-R25-REQUEST-STATUS:200;Success
	Used in:	Reply objects	
	How used:	This property indicates the status code returned after an event request. Only the 200 “success” status is supported in this release. The property is only used by RESOURCE25 in replying to a request. The value consists of a numeric status and a text description, separated by a semi-colon. **Note**: In this release, if a vCalendar object generated by your SIS has syntax errors, RESOURCE25 moves the file to an error directory. *The request status is not used to return error information.*	
X-R25-TYPE	Identifies the event type of the event	Event Type	**Syntax:** X-R25-TYPE;X-R25-ID=R25 identifier: event type **Example:** X-R25-TYPE;X-R25-ID=25:Section
	Used in:	Request objects	
	How used:	RESOURCE25 includes the X-R25-ID parameter on export.	

Appendix B Publishing Events

Introduction

What's in this appendix

This appendix contains information about event publishing from RESOURCE25.

It describes two kinds of publishing:

- **Implicit** - as part of your transaction-based interface
- **Explicit** - manually

It also provides instructions for explicit event publishing and suggests how you might use published events.

Contents

Implicit event publishing

Publish method

RESOURCE25 automatically exports a vCalendar object with a Publish method type when it processes a new or changed event as part of the transaction-based interface process.

For an event to be published in this way, it must belong to a special event category called “Publish to vCalendar.”

RESOURCE25 does not expect any response from your SIS to a “publish” action.

Using scripts with automatic publish

As with other vCalendar export files, you can specify that RESOURCE25 run a specific script each time it exports events to the Publish directory. See page 7-36 for information and instructions.

Explicit event publishing

Two ways

You can manually publish events from RESOURCE25 at any time. You can publish:

- Individual events from your event structure.
- Groups of selected events based on event search results.

Publishing individual events

Follow these instructions to publish individual events:

Step	Action
1	From the RESOURCE25 main menu, choose Event|Event Cabinets. **Result**: The Event Cabinets form opens.
2	Double-click the cabinet that contains the event you want to publish. **Result**: The Events form opens to the selected cabinet.
3	Click the expand arrows to the left of the event names until you locate the event you want to publish, and select it.
4	Choose Actions|Publish in vCalendar Format. If you've selected an event with subevents, a dialog asks you whether you want to publish the selected event only or the selected event and all its subevents. Click the selection button you want. **Result**: RESOURCE25 creates a vCalendar file with a Publish object containing the event(s) you selected, and sends it to the vCalendar Publish directory.

Publishing groups of events

Follow these instructions to publish a group of events:

Step	Action
1	From the RESOURCE25 main menu, choose Interface|Export|Publish vCalendar files. **Result**: The Publish Events form opens. 
2	If you know the saved event search results you want to publish and you don't want to modify the search: • Select the search name in the drop down list. • Click Publish! *RESOURCE25 creates a vCalendar file with a Publish object containing the events in the search results, and sends it to the vCalendar Publish directory.* If you know the saved search results you want to export but need to modify the search: • Select the search name in the drop down list. • Click the "Details" button to the right of the list. *The Event Search form opens with the saved search criteria entered/selected.* • Continue to step 3. If you want to create a new event search: • Select (none) from the drop down list. • Click the "Details" button to the right of the list. *The Event Search form opens.* • Continue to step 3.

Step	Action
3	Enter or modify the search criteria so that the search results will list the events you want to publish. Click Search! **Result**: The results of the search are listed.
4	Choose Actions|Export all to vCalendar Format. **Result**: RESOURCE25 creates a vCalendar file with a Publish object containing the events in the search results, and sends it to the vCalendar Publish directory.

Using published events

There are many ways to use published event data. Here are but two:

- You could make the event data available to students, faculty, and/or the public via the world wide web.
- Faculty and students could search for events in RESOURCE25, publish the events in vCalendar format, and import that file into a personal digital assistant.

Glossary

Active user

An individual who currently has logon access to RESOURCE25 and the RESOURCE25 database via a unique user ID and password. Each user's membership in one or more security groups determines the scope of that user's security rights and permissions.

Alien event

An event that was created in another system, but shares event data with RESOURCE25. On import into RESOURCE25, the shared data in alien events is controlled by the import processing options set in RESOURCE25. See "Import processing options."

Assignment code

A code that must be included in columns 78-80 of each record in a Class Descriptor file that specifies how the record should be processed by SCHEDULE25 and/or imported into RESOURCE25.

Bulk interface

An event data interface used to send a large number of class events to SCHEDULE25 and RESOURCE25 for processing once per term/semester to prepare the class schedule for that term/semester.

Campus profile data

Space and customer data used in SCHEDULE25 processing. Campus profile data can be created and maintained in RESOURCE25, and exported for use by SCHEDULE25.

Class descriptor data format

A data format used to share event data among a student information system, SCHEDULE25 and RESOURCE25. The data is contained in a flat ASCII file whose format is defined by the Control file. This is the only event data format that SCHEDULE25 "understands." See "Control file."

Control file

An ASCII file used by SCHEDULE25 that defines the placement and length of the data elements in a Class Descriptor file and certain SCHEDULE25 processing parameters.

Course master extract

A program developed by your institution that extracts class records from your course master file for processing by SCHEDULE25 and/or RESOURCE25.

Course master update

A program developed by your institution that updates class records in your course master file with changes made in SCHEDULE25 and/or RESOURCE25.

Cycle time

The frequency (in minutes) with which RESOURCE25 polls the vCalendar import directories to see if files have been sent from the student information system for processing.

Data exchange policy

A written statement that defines the system that controls shared event data and who can modify shared data.

Destination event

An event that has been specified as a potential parent event for some imported class descriptor class events. Destination events are identified in the import profile. See "Import profile."

Event structure The hierarchy of events within an event cabinet, including the cabinet event itself. Every event in the structure has a parent event, and may or may not have subevents.

Export The act of sending event-related data from RESOURCE25.

Filter rules Placement criteria defined in RESOURCE25 that is used to appropriately place incoming vCalendar events in the RESOURCE25 event structure. Filter rules can be defined for any event types that are likely to be imported. RESOURCE25 comes with a defined "default filter rule" that is used for incoming events that fail all other filter rules for events of their event type or have no filter rules defined for events of their event type. See "Event structure."

Import The act of bringing event-related data into RESOURCE25.

Import processing options Options specified in the RESOURCE25 vCalendar system definition that control certain aspects of how vCalendar event records are processed when imported into RESOURCE25.

Import profile A description created in RESOURCE25 of the class descriptor events included in an import.

Interface tag A data "tag" used to mark certain events in the event structure as destination events during class descriptor event data import. See "Destination event."

Method Type A vCalendar data property that specifies the transaction requirements of the events controlled by it. Valid method types are: Request, Cancel, Declinecounter, Reply, Counter, Publish, and R25 Cancel.

Native event — An event created and maintained in RESOURCE25.

Naturalized event — A formerly alien event whose shared event data is no longer controlled by RESOURCE25 import processing options. Authorized users can "naturalize" alien events when appropriate. See "Alien event" and "Import processing options."

To Do item — An action item created by RESOURCE25 when a specific action must be taken on an imported vCalendar event. The issuing of To Do items is controlled by the To Do import processing option. See "Import processing options."

Transaction-based interface — An event data interface used to keep the shared event data in a student information system and in RESOURCE25 in sync on an ongoing basis.

vCalendar data format — An industry-wide, system-independent data format used to share event data between a student information system and RESOURCE25. The data is contained in an ASCII file whose format and data elements are defined by a data specifications standard.

vCalendar object — A component of a vCalendar file that contains one or more events. Each vCalendar object defines the general characteristics of the events it contains.

vCalendar property — A data element in a vCalendar file. RESOURCE25 supports the use of a defined set of these properties. See Appendix A.

vEvent object A component of a vCalendar file that defines a single event or one date/time pattern of an event. vEvent objects of the same method type are contained in the same vCalendar object. See “vCalendar object” and “Method type.”

Index

D

E

F

H

I

P

R

S

T

U

V

W

X