



REV SCHEDULER for AS/400

IBM Server *Proven*TM

Powerful Scheduling made easy

Run scheduled jobs in a unattended environment to increase:

- Throughput,
- Accuracy,
- Efficiency.

ANY Job that can be:

- Submitted to Batch,
 - Run Interactively
- on the AS/400 can be executed unattended by REV SCHEDULER for AS/400 as a scheduled Job Event.

View and Hear the status of Jobs as they are running

The Windows interfaces allow to immediately see and hear the status of Job Events on:

- Local AS/400,
- Networked AS/400's,
- Other Networked platforms running REV SCHEDULER such as -
 - AIX,
 - LINUX,
 - WINDOWS.

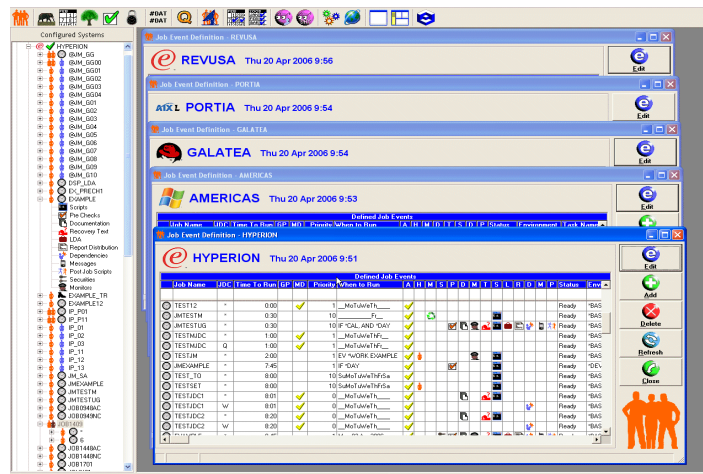
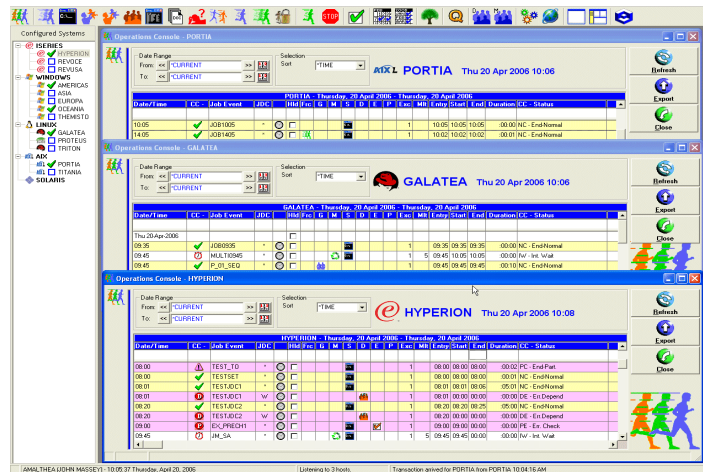
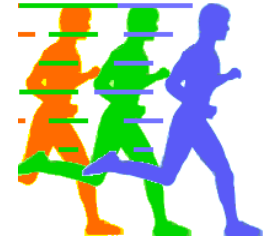
As the Operations panel uses push technology the information is pushed from the Host(s) directly to the Windows interface and the Job Events change color and status immediately.

Sounds can be configured to reflect the completion status of Job Events.

Control & Manage multiple AS/400's & other platforms

The Windows interfaces allow you to control and manage Job Events for:

- Local AS/400,
- Networked AS/400's,
- Other Networked platforms running REV SCHEDULER such as -
 - AIX,
 - LINUX,
 - WINDOWS.



Major Features

- Environments,
- Job Day Codes,
- Job Event Security,
- Module Security,
- LDAP Interface,
- Group Jobs,
- Group job Intra dependencies,
- Operations Panels,
- Networking,
- Dependencies,
- Monitor for JobQ and Active times,
- Date Variables,
- Screen Capture & Run,
- Audit and Undo/Undelete,
- Hyperlinks,
- Components,
- High Availability,



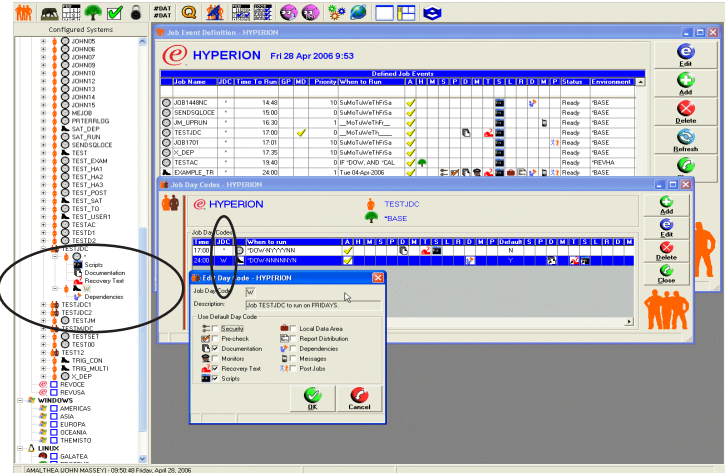
Multiple Job Day Codes.

Each Job Event, in REV SCHEDULER, can be defined to have different Job Day Codes (JDC) or 'flavors' of the same Job Event.

Each JDC can be seen as a different flavor of the same Job Event e.g. Chocolate, Strawberry etc.,

A simple example is as follows:

- Job Day Code * runs on Monday through Thursday by Time at 17:00 with no dependencies and Job Scripts defined,
- Job Day Code W runs on Friday by Trigger when a Dependency Roster is satisfied and use the Documentation, Recovery Text and Job Scripts defined for the default Job Day Code (*).

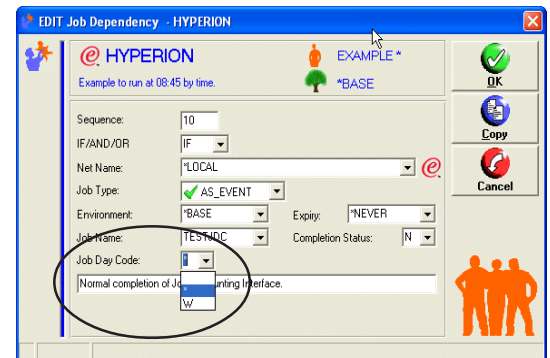


This is all still only 1 Job Event with the same Job Name, Environment etc., but has varying Job Day Codes or flavors.

In other Scheduling software you would need to define 2 separate Job Events with different names.

Dependency Rosters can also use the JDC as the dependency sequence can be dependent upon:

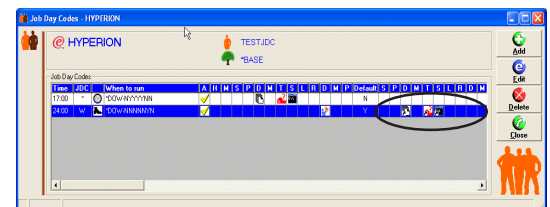
- Completion status of a designated JDC of the Job Event - by referencing the specific Job Day Code,
- Completion status of any JDC's of the Job Event - by referencing a blank Job Day Code.



Inheriting default JDC components

Any additional JDC can inherit any of the components from the default JDC (*).

In this example we have defined the Dependencies, for the new Job Day Code W, but will inherit the Job Scripts, from the default Job Day Code (*).



By inheriting JDC components you do not have to define the components in the additional JDC definition and you can use the default JDC (*) component definition at execution time.

The components that can be inherited are:

- Security,
- Pre Check,
- Job Documentation,
- Monitors,
- Job Recovery Text,
- Scripts,
- Local Data Area,
- Report Distribution,
- Dependencies,
- Messages,
- Post Job Scripts.



Security

The security function in REV SCHEDULER can be implemented in up to 4 levels:

- Module level.
- Environment level.
- Job Event level.
- Job Script override level.

Module Level

Using Module Security you can define Users who are authorized to:

- Command or Menu option,
- Selection options from panels.

All security is defined by:

- User Profile,
- Group Profile,
- *PUBLIC,
- User defined Authorization groups.

Job Event Level

Using the Job Event security you can control if the Job Event appears within Definitions and Operations panels.

Utilizing the security you can give the control of some or all Job Events back to the Users as:

- Only authorized users will be able to view the authorized Job Events,
- Only authorized users will be able to perform selection options (such as Hold, Release or Force run) on the authorized Job Events that are displayed.

Environment Level

By setting the security at Environment level any:

- Existing Job Events,
 - New Job Events,
- for the specific Environment are automatically secured.

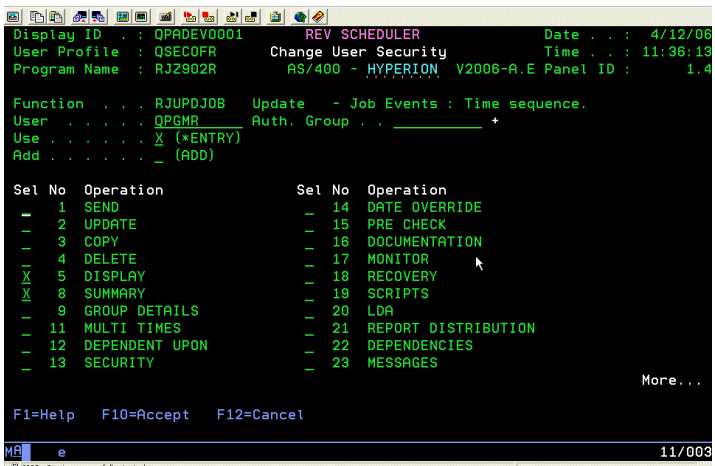
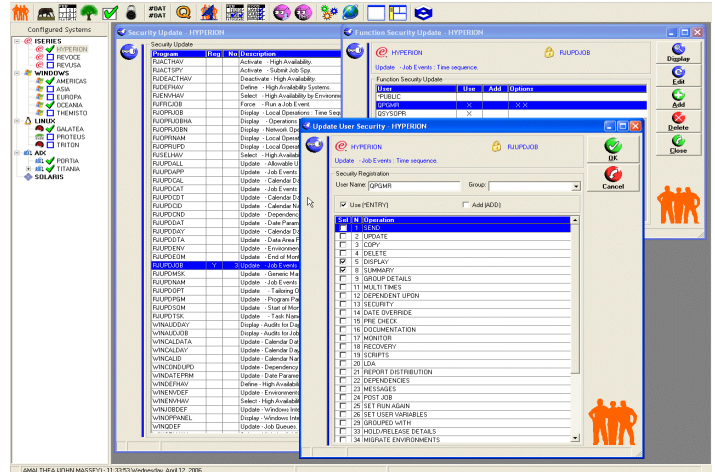
Setting the security at Environment level allows for the security to be:

- Controlled,
 - Managed,
- in one central location.

Job Script override Level

Using the Job Script override security you can control if a User or Group of Users can:

- View,
 - Update,
- areas of a Job Script value.





REV SCHEDULER for AS/400

IBM Server Proven™

LDAP Interface Security

REV SCHEDULER now has full support for LDAP (Lightweight Directory Access Protocol) which is an Open Systems protocol to allow programs to look up information from a central server.

In this way LDAP can be a central security repository and contain the security definitions for some of all of the servers where REV SCHEDULER is operational.

The LDAP Interface can support security checking for both of the:

- 5250 Interface or terminal sessions,
- Windows Interface.

The LDAP security also supports keyword values for:

- *PUBLIC_SYSTEMS - these are Systems other than the Systems specifically defined,
- *PUBLIC_USERS - these are Users other than the Users specifically defined.

Security Policies

There are two security policies available:

- *SYSTEM,
- *LDAP.

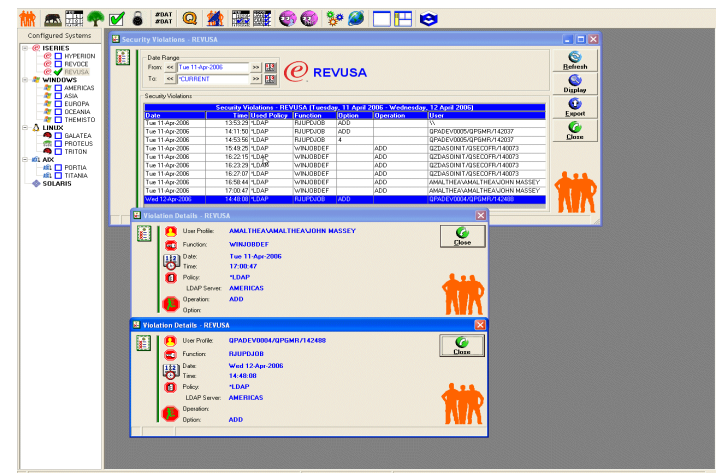
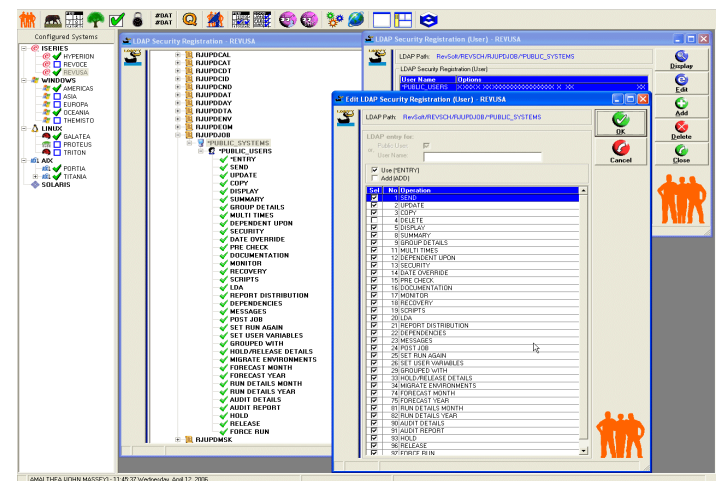
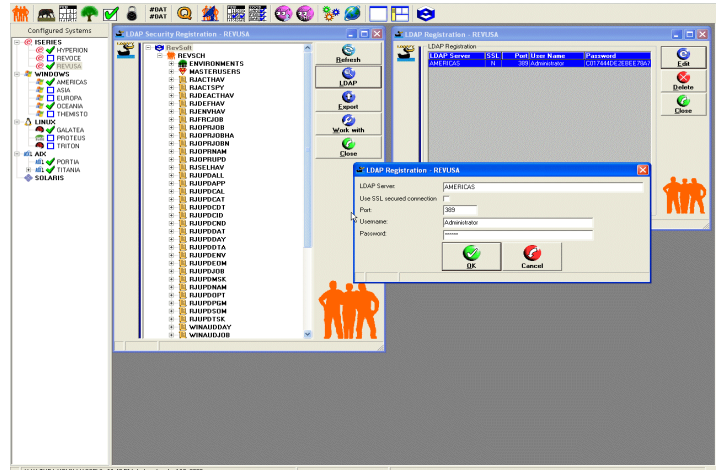
There is also a Tailoring Option that can allow the System policy to override the LDAP policy, and this can be set on a system by system basis.

Security violations

All Security violations are logged and can be displayed by date or date range.

The security violation logs:

- Domain, System, User - if from the Window application,
- Job/User/Number - if from the 5250 interface,
- Function that was attempted to be accessed,
- Operation that was attempted,
- Date and Time of violation,
- Policy type - if this value is *LDAP the LDAP server name will also be displayed.





Group Jobs.

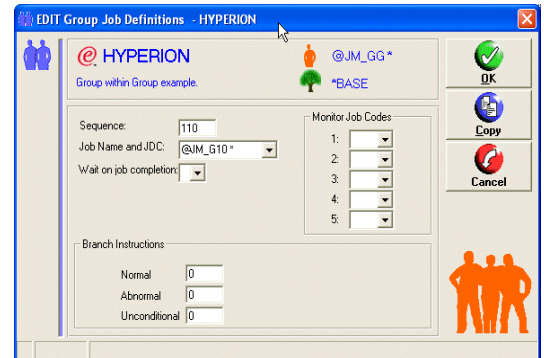
In some circumstances computer tasks are run as 'streams' where a group of jobs are executed in a sequence or are grouped to run together.

This is referred to as a Group Job where the initial or controlling Job Event is referred to as the Parent Job and all the subsequent stream jobs are referred to as the Child or Children Jobs.

Child or Children Job Events are controlled by a Parent Job.

Each Child Job can also be defined to:

- Monitor for messages,
- Submit and continue or wait until the completion of the Child Job,
- Be able to Branch to another Group sequence based on the completion status of the Child Job.



Group Job terminology

The terminology for Group Jobs are as follows:

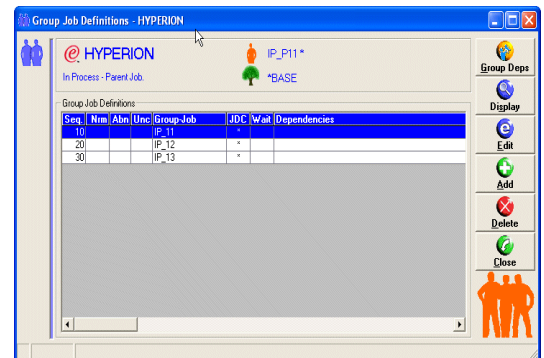
- Parent Job - This is the Group Control job or the main Job Event that executes all the chained Job Events.
- Child Jobs - These are the Group Member jobs or the Child Job Events that are being controlled by the Parent Job.
- Intra Dependency - This is a dependency of a Child Job on one or more other Children Jobs within the same Group.

Group Job - Concurrent

A Concurrent Group Job is a group execution where all the Child jobs will be submitted at the same time.

In this example the Parent Job IP_P11 will start and it will submit the Child Jobs:

- IP_11, • IP_12, • IP_13,
- at the same time.

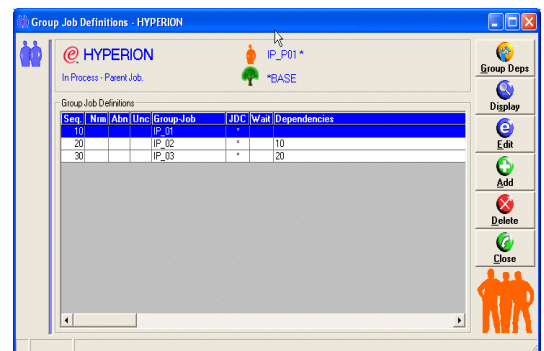


Group Job - Sequential

A Sequential Group Job is a group execution where all the Child Jobs wait until the successful completion of the prior Child Job.

In this example the Parent Job IP_P01 will start and it will submit the first Child Job - IP_11,

- If IP_11 completes normally then Child Job IP_12 will be submitted,
- If IP_12 completes normally then Child Job IP_13 will be submitted.





Group Job - Mixed

A Mixed Group Job is a group execution with Concurrent and Sequential processing.

In this example the Parent Job @JM_GG will start and it will submit the Child Jobs:

- @JM_G01 and @JM_G02 at the same time,
- If @JM_G02 completes normally then Child Job @JM_G03 will be submitted,
- If @JM_G03 completes normally then Child Job @JM_G04 will be submitted as will @JM_G05,
- If @JM_G04 and @JM_G05 complete normally then Child Job @JM_G06 will be submitted as will @JM_G07, @JM_G08, @JM_G09 and @JM_GG00,
- If @JM_G07, @JM_G08, @JM_G09 and @JM_GG00 complete normally then Child Job @JM_G10 will be submitted.

Seq	Nm	Abn	Unc	Group-Job	JDC	Wait	Dependencies
10				@JM_G01	*		
20				@JM_G02	*		
30				@JM_G03	*	20	
40				@JM_G04	*	30	
50				@JM_G05	*		
60				@JM_G06	*		40 AND 50
70				@JM_G07	*		
80				@JM_G08	*		
90				@JM_G09	*		
100				@JM_GG00	*		
110				@JM_G10	*		70 AND 80 AND 90 AND 100

Intra Dependencies

For the Sequential and Mixed processing the conditioning of the Children Jobs is based on the completion of a list of previously executed Children Jobs which are referred to as Intra Dependencies.

Groups within Groups

As Group Jobs basically are a representation of Job Streams within a business work flow we may encounter Job Streams within Job Streams.

Within REV SCHEDULER for AS/400 the concept of Group Jobs within Group Jobs is supported and can be simply implemented.

Group Explosion execution Panels

Parent Jobs can be easily identified on the Operations Panels by the image displayed.



Double clicking on the Parent Job image will launch a Group Explosion view - which is a subset of the Operations Panel for that Group Job only.

In this example we have two Group Explosion panels as we are executing a Group within a Group.

The Group Explosion panel uses exactly the same push technology as the Operations Panels and they are immediately updated when actions occur on the Host - and this is communicated by color and sound.

Job Name	JDC	Grp	Abn	Unc	Grp	Job	S	P	D	M	E	S	L	B	D	M	P	Ex	Mh	En	St	End	CC	CC	
@JM_GG00	*																					14:45	14:45	14:45	End OK
@JM_G01	*																					14:45	14:45	14:45	End OK
@JM_G02	*																					14:45	14:45	14:45	End OK
@JM_G03	*																					14:45	14:45	14:45	End OK
@JM_G04	*																					14:45	14:45	14:45	End OK
@JM_G05	*																					14:45	14:45	14:45	End OK
@JM_G06	*																					14:45	14:45	14:45	End OK
@JM_G07	*																					14:45	14:45	14:45	End OK
@JM_G08	*																					14:45	14:45	14:45	End OK
@JM_G09	*																					14:45	14:45	14:45	End OK
@JM_G10	*																					14:45	14:45	14:45	End OK



Operations Panels

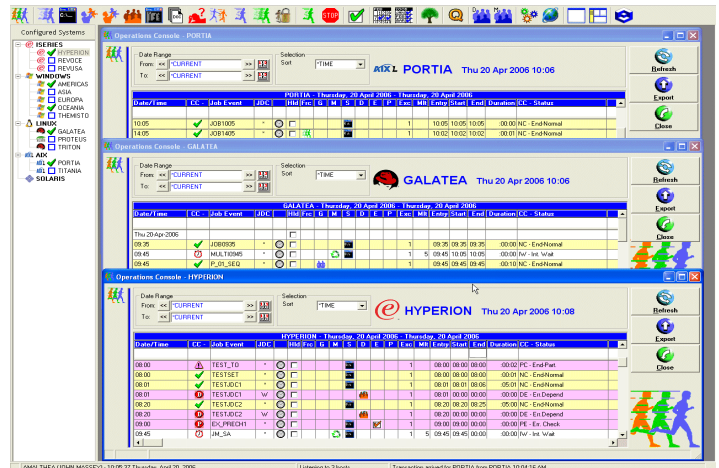
When Job Events are due to run a mechanism is required to allow you to control and manage the entire 'Schedule' of Job Events and this is performed by the Operations Panels.

There is also a Network Operations Panel to allow you to control and manage Job Events on networked machines - from Focal Point(s).

The Operations Panel will be the panel most used by the operations personnel within your corporation to control the Job Events under the control of REV SCHEDULER for AS/400.

There are various Operations Panels to provide varied views and functionality for the Local and Network schedule of Job Events.

To make the Operations panel more informative the Job Events status is displayed in color and the colors will change based upon the status of the Job Event(s).



Types of Operations Panels

The Operations panels are available for:

- All Job Events to be executed on the directly connected AS/ 400 - this is the Local Operations panel.
- All Job Events to be executed on any networked machines (regardless) of Operating system that are running REV SCHEDULER and this system is a Focal Point - this is the Network Operations panel.

Operations Panel software

The Operations panels can be executed on:

- 5250 interface.
- WINDOWS interface.

Environments started

The Operations panels display Job Events that are available to be executed on the date for:

- Environments that are currently started.
- Environments that you are authorized to see.

If you are logged on as a Master User you will be able to see all Job Events for the Environments.

Authorization for options

The security for options can also restrict what options the user(s) can execute on the Operations panels.

If you are logged on as a Master User you will be able to perform all options.



REV SCHEDULER for AS/400

IBM Server **Proven™**

Fully Networked Solution

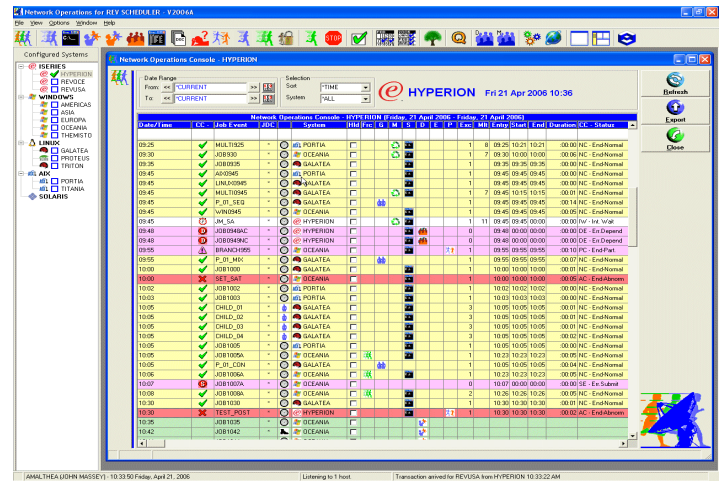
REV SCHEDULER for AS/400 is a fully networked solutions not only to Local and Remote LPAR's or AS/400's but also to other platforms.

REV SCHEDULER currently has native scheduling software that runs on:

- AS/400,
- WINDOWS,
- LINUX,
- AIX.

Push Technology

The Local and Network Operations panels use Push Technology - where the information is pushed from the Host(s) directly to the Windows application software.



Using Push technology :

- Eliminates unnecessary polling of machines,
- Reduces network traffic,
- Display the status immediately the Host(s) register it,
- Eliminates manual intervention to detect the status of a Job Event.

Multiple Focal Points

All RevSoft products have been designed and constructed to have:

- Multiple Focal Points,
- Multiple Focal Points on multiple platforms.

Using this technology you can have -

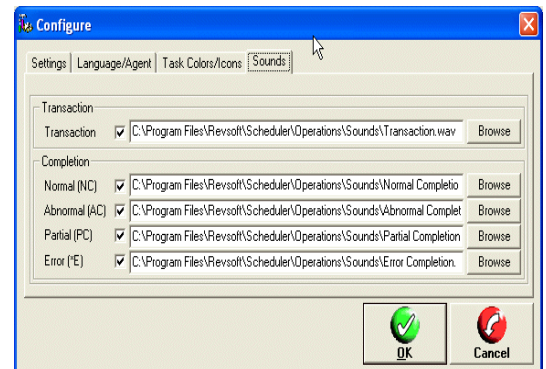
- Multiple major Focal Points on AS/400 for all AS/400, WINDOWS, LINUX and AIX machines,
- Single or Multiple WINDOWS Focal Point only for the WINDOWS machines,
- Single or Multiple LINUX Focal Point only for the LINUX machines,
- Single or Multiple AIX Focal Point only for the AIX machines.

Configurable Sounds

The Local and Network Operations panels can be configured for sounds to be played based on the completion status of Job Events.

The Operations panels can be configured to:

- Automatically update,
 - Play sounds,
 - Change colors,
- whenever the status of a Job Event changes in your network.





Dependencies

Not all Job Events to be executed by the REV SCHEDULER run strictly by time and a Dependency Roster can:

- Pass a Job for execution,
- Trigger a Job to be executed immediately.
- Only accept Dependency Roster completion records between time ranges.

What is a Dependency Roster ?

A Dependency Roster is a defined list of dependency criteria which can contain:

- Job Events - These are scheduled Jobs executed in REV SCHEDULER on any platform,
- Conditions - These are conditions set in REV SCHEDULER on any platform,
- User Jobs- These are non scheduled batch jobs executed on AS/400 platforms only.

Dependency Expiry

Each Dependency Roster element can have a 'use by date' or expiration for the dependency record which can be:

- *NEVER - This value will never expire and the dependency record can be used on any date,
- *TODAY - Only dependencies created on the same date can be used,
- *YESTERDAY - Only dependencies created on the same date or the day before can be used.

Platforms for Job Events

Dependencies can be executed within REV SCHEDULER on any available platforms. e.g.

- On AS/400 use REV SCHEDULER for AS/400,
- On LINUX use REV SCHEDULER for LINUX.
- On WINDOWS use REV SCHEDULER for WINDOWS,
- On AIX use REV SCHEDULER for AIX.

Local, Network & Cross Platform dependencies

Dependencies can be on the:

- Local machine,
- Networked AS/400 machines.
- Non AS/400 machines running REV SCHEDULER on other platforms such as WINDOWS, LINUX, AIX etc.,.

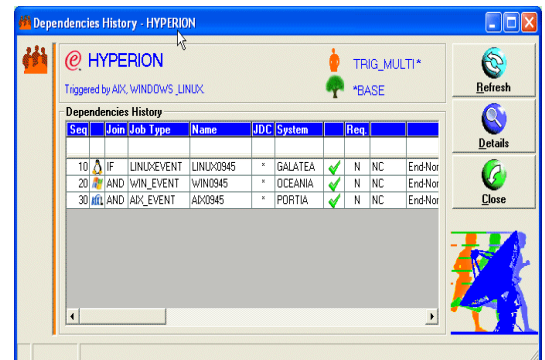
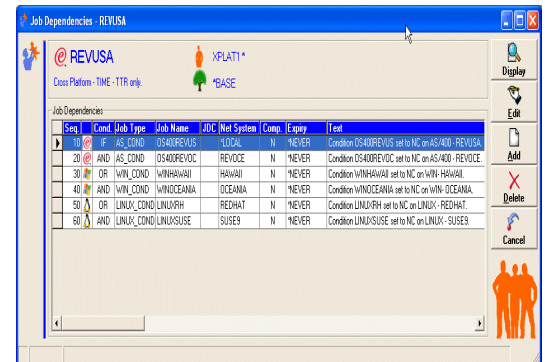
All Network and Cross Platform dependencies can be simply:

- Defined,
 - Monitored,
 - Managed,
- from the REV SCHEDULER WINDOWS application.

Dependency History

On the Operations Panels you can view the Dependency:

- Roster status - This displays the current status of all the elements in the Dependency Roster,
- History - This displays status of all the elements of the Dependency Roster when the Job Event started or failed the dependency check.





Monitor for Job Queue and Active times

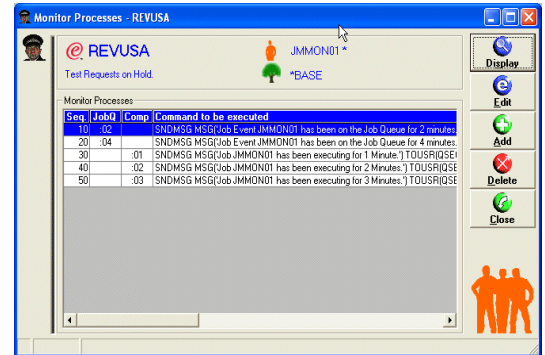
After a Job Event is submitted you can monitor for times the Job Event is:

- On the Job Queue for more than a pre defined times e.g. 5 minutes, 10 minutes etc.,
- Executing for more than a pre defined times e.g. 5 minutes, 10 minutes etc.,

There can be multiple sequences defined and actioned for both the Job Queue wait time and the Job Active duration time.

In this example we have monitors defined for -

- Job Queue wait times of:
 - 2 minutes,
 - 4 minutes,
- Job Active duration times of:
 - 1 minute,
 - 2 minutes,
 - 3 minutes.



Date Variables

Using Date Variables allow you to define a variable and the calculation for the date to be transposed whenever that date variable is located in the:

- Job Scripts, • Local Data Area,
- Post Job Scripts.

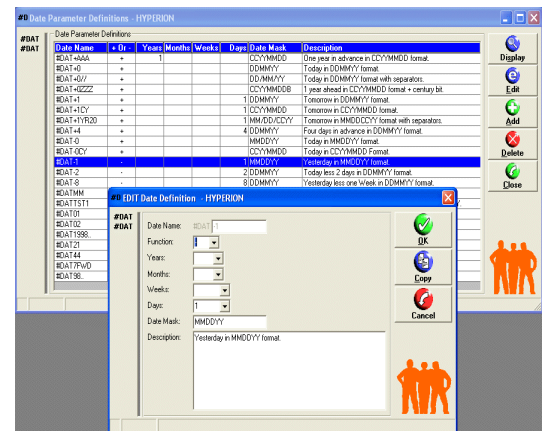
Each Date Variable can calculate the required date based on combination of:

- Years, • Months, • Weeks and • Days,
- and the required Date Format mask is defined.

Date Variables can be dragged and dropped across platforms as they are available in REV SCHEDULER on all platforms.

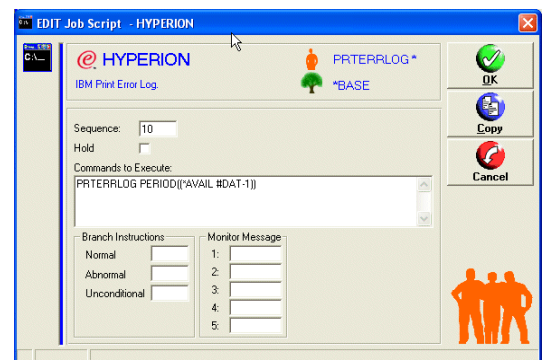
Other User Defined Variables are also available to define:

- Start of Month values, • End of Month values,
- Data Area values, • Program Call values,
- Calendar date values, • Generic mask values.



In this example the Job Script is executing the OS/400 command PRTERLOG with the beginning date set to #DAT-1.

#DAT-1 is defined as today's date - 1 day (yesterday) and this is transposed when the Job Script is executed.





Screen Capture & Run

Within REV SCHEDULER for AS/400 you can Capture ANY Interactive session (the entire session from Signon through to Signoff) and Run this as a scheduled Job Event.

The captured Screen details can then be executed (in Batch) as a scheduled Job Event where REV SCHEDULER for AS/400 actually communicates directly with the 5250 Data Stream to execute the Captured session.

What is captured

When an interactive Job is captured it is the complete Job including:

- Log on information (User ID and password),
 - Every screen that is sent,
 - Every keystroke that is made,
- through to the log off process - and the subsequent return to the sign on screen.

Why is the Log On captured ?

The Log On information is captured so:

- The entire session from Log On through to Log Off is executed,
- The correct authorization is executed during the Screen Run process.

User Passwords

All User Passwords are stored in encrypted state and cannot be viewed at any time.

Screen Masking

Once the Screens have been captured you can mask dates over the existing values that were captured on the screens.

Screen Comparison

You can define Screen comparisons where the actual data on the screen will be compared to pre defined values with the added ability to:

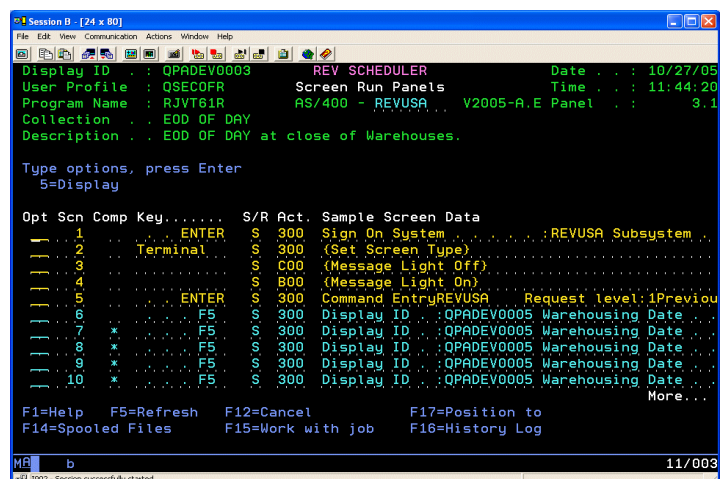
- Press a Function key if the comparison fails i.e. F5 to Refresh,
- Check the screen a defined number of times waiting the defined time between retries.

Screens Executed can be viewed

As REV SCHEDULER executes the screens, within the Screen Capture & Run, they are being registered in the REV SCHEDULER history database.

These can be reviewed after the Screen Capture & Run process has completed.

These are the actual panels as they were displayed during the Screen Capture & Run process.





REV SCHEDULER for AS/400

IBM Server **Proven™**

Audit

The Audit function in REV SCHEDULER logs every change that is made to a Job Event or any of its components.

These details are logged in the REV SCHEDULER database and can be reviewed.

The details are simple to follow with 6 possible images:

- Add new,
- Update (Before),
- Update (After)
- Delete,
- Undo Update,
- Undo Delete.

The audit details will even log the -

- Domain\System\User if the change was made in the Windows applications,
- Job/User/Number if the change was made via the 5250 interface,
- How the change was made by:
 - Keyboard - Manual Entry,
 - REV VIEW - Send from another machine,
 - Drag & Drop - Drag and Drop in the Windows interface.

Undo Changes

When updates are performed to the Job Event or any of the components the:

- Before image - Image of the data prior to the change,
 - After image - Image of the data after the change,
- are captured and this does allow for the component to be 'rolled back' or 'Undo' the change.

The Undo action is also registered in exactly the same manner you can actually 'Undo the Undo' should the need arise.

Undelete Job Events or Components

When a Job Event or any of the components are deleted all the details are registered.

The Undo functionality allows you to:

- Undelete a complete Job Event all its original:
 - Job Day Codes,
 - Components,
- Undelete a Job Event component or component sequence.

Date	Time	Job Name	JDC	Component	Op.	Origin	Entry Type	Who
Fri 14-Apr-2006	11:46:48			JSCRIPT		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:47:07	JMTESTUG		EVENT_ID		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:47:48	JMTESTUG		JDOCUMENT		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:48:00	JMTESTUG		JDOCUMENT		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:48:15	JMTESTUG		JDOCUMENT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:48:31	JMTESTUG		JDOCUMENT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:48:45	JMTESTUG		JDOCUMENT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:49:11	JMTESTUG		JDOCUMENT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:49:29	JMTESTUG		JDOCUMENT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:52:49	JMTESTUG		JRECOVERY		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:52:54	JMTESTUG		JRECOVERY		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:52:56	JMTESTUG		JRECOVERY		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:52:58	JMTESTUG		JRECOVERY		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:53:11	JMTESTUG		JRECOVERY		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:53:40	JMTESTUG		JRECOVERY		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:53:55	JMTESTUG		JRECOVERY		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:54:31	JMTESTUG		JRECOVERY		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:54:39	JMTESTUG		JRECOVERY		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:55:12	JMTESTUG		JSCRIPT		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:55:21	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:55:36	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:55:59	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:56:11	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:56:21	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Fri 14-Apr-2006	11:56:30	JMTESTUG		JSCRIPT		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:42:09	JMTESTUG		JPOST_JOB		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:42:14	JMTESTUG		JPOST_JOB		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:42:36	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:43:30	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:43:42	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:43:59	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:44:05	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:44:12	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:44:20	JMTESTUG		JPOST_JOB		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:45:18	JMTESTUG		JMESSAGE		WIN_APP	DRAG&DROP	AMALTHEA\AMALTHEA\JOHN MASSEY
Sat 15-Apr-2006	08:46:14	JMTESTUG		JMESSAGE		WIN_APP	KEYBOARD	AMALTHEA\AMALTHEA\JOHN MASSEY

Audit Log Details - HYPERION

Internal ID: 785000785 JDC: *
 Component: JRECOVERY Job Recovery Text (F2T>X0)
 Function: UPDATE Existing Record Updated

Job Event Details:
 Job Name: JMTESTUG
 Environment: *BASE
 Description: Test job for JM UG 2006A.

Audit Details:
 Date: Fri 14-Apr-2006
 Time: 11:53:13
 Domain: AMALTHEA
 System: AMALTHEA
 User: JOHN MASSEY

Entry Type:
 Origin: WIN_APP WINDOWS Application
 Source: FORM User Entry/WINDOWS Application
 Entry: KEYBOARD Manual Keyboard Entry

Seq. Recovery Text
 40 Recovery Text Line 4 - Sequence 40
 40 Recovery Text Line 4 - Sequence 40 - changed

Audit Log Details - HYPERION

Internal ID: 785000785 JDC: *
 Component: JSCRIPT Job Script (F2FN00)
 Function: DELETE Delete Existing Record

Job Event Details:
 Job Name: JMTESTUG
 Environment: *BASE
 Description: Test job for JM UG 2006A.

Audit Details:
 Date: Fri 14-Apr-2006
 Time: 11:55:59
 Domain: AMALTHEA
 System: AMALTHEA
 User: JOHN MASSEY

Entry Type:
 Origin: WIN_APP WINDOWS Application
 Source: FORM User Entry/WINDOWS Application
 Entry: KEYBOARD Manual Keyboard Entry

Seq. (Hold) Nrm|Abn|Unc [Command Function to be Executed]
 20 N SNMSG MSG(REVJOBNAM is running) TDUSRIDSECOF



Hyperlinks

When defining the:

- Documentation,
 - Recovery Text,
- of a Job Event you can enter a URL anywhere in the text.

Using a URL (that contains documentation) will allow you to enable Recovery instructions to be actioned in a short period of time and this will decrease times for problem resolution.

The decreasing of problem resolution time will give greater throughput and accuracy.

By having your:

- Operational Procedures,
 - Soft copy manuals,
- on the Intranet or a departmental server will allow them to be accessed via the URL.

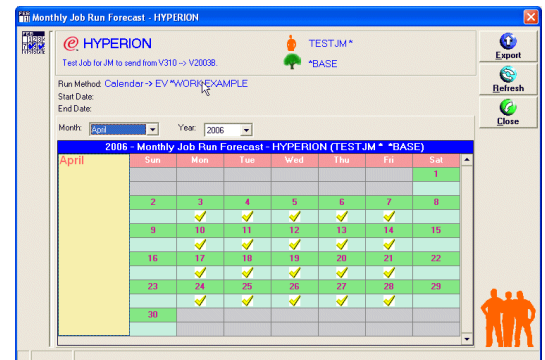


Forecasting

As well as being able to view dates in the future from the Local and Network Operations panel to see Job Events that will be available to be execute din future dates - there is also Forecasting by:

- Month,
 - Year,
- in the Windows application and is available in the:
- Definitions,
 - and
 - Operations Panels, Applications.

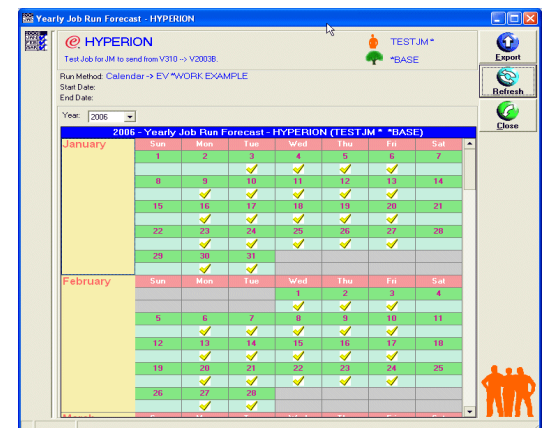
Any types of Job Holds (Operator or Future Holds) are immediately visible.



Export

Both the Monthly and Yearly forecast panels can be exported for:

- Reporting,
- Posting on the Intranet,
- E-mailed to departmental heads so the Job run rosters are notified and visible.





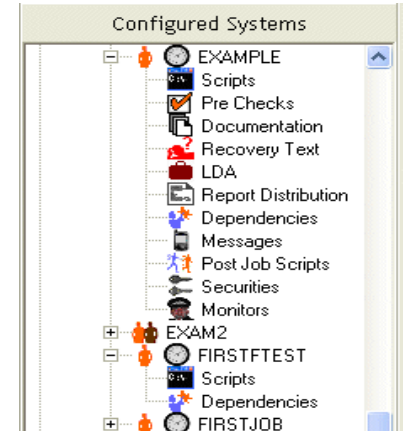
Job Event components

A Job Event definition can contain some/all of the available components.

The defined components can be viewed on the Job Events definition Tree in the Windows application.

The Base Model components are available in REV SCHEDULER on all platforms are:

- Job Execution - This controls when, where and how a Job Event will be executed.
- Job Documentation- This contains detailed instructions about the Job Event.
- Job Scripts- This contains the scripts (Commands or Program calls) that are executed during the processing of the Job Event.
- Dependencies - This contains a list of prerequisite dependency conditions that must be satisfied to allow the Job Event to be available to run.
- Job Recovery Text- This contains detailed recovery instructions about to the Job Event.
- Post Job processing - This contains the scripts (Commands or Program calls) that are executed at the completion of the Job Event and can be based on the completion status.



Additional components that are specific to REV SCHEDULER for AS/400 are:

- Security details - This contains individual security for a Job Event.
- Pre Check details - This contains checks for Tape Drives, Object Locks etc., prior to starting the processing for the Job Event.
- Local Data Area details - This contains the 1,024 bytes available for the *LDA of the Job Event.
- Monitor details - This contains definitions of scripts to be executed if the Job Event is:
 - On Job Queue, or
 - Active for longer than pre defined durations.
- Report Distribution details - This contains details of spooled files, scan instructions and where you would like to distribute the reports to.
- Message details - This contains details of where you would like the completion message to be distributed to.

Drag and Drop - AS/400 and Cross Platform

As well as being able to Drag and Drop Job Events across: AS/400's, or LPAR's, you can also Drag and Drop some: Job Event components, and REV SCHEDULER objects, across Platforms.

Some of these REV SCHEDULER objects are::

- Tasks.
- Calendars.
- Environments.
- #DAT Date variables etc.,



REV SCHEDULER for AS/400

IBM Server Proven™

High Availability

Some companies have multiple High Availability (HA) machines as their applications are mission critical to the continuity of their business.

The HA function with REV SCHEDULER for AS/400 operates totally independently of the HA solution implemented for the ERP application.

The HA Functionality in REV SCHEDULER is available on every platform (e.g. AS/400, LINUX, AIX and WINDOWS).

Why use REV SCHEDULER HA ?

Even though the ERP applications and data may be replicated to achieve business continuity it may not be wise to replicate all the Scheduler Job Events as some will be specific to the machine.

A simple example of this is the execution of scheduled Backups as you would not have them both executing at the same time on both machines as they would then be both unavailable.

The Scheduled Job Events can then be broken down into the areas of:

- Machine specific,
- HA applicable.

Within REV SCHEDULER for AS/400 you can ensure the HA applicable Job Events are shadowed and both Job Schedules are kept in sync.

Shadowing

Shadowing is the term used for the process synchronize the Job Events between the machines.

Selecting for HA

Job Events can be selected for HA by:

- Individual Job Event name,
- Environment name.

Once the Job Events are selected the shadowing process begins immediately and the Job Event is sent to the HA computer(s).

Activate for HA

The RJACTHAV (Activate HS) command can be executed as part of the cutover process to:

- Start defined H.A. Environments,
- and
- End NON H.A. defined Environments.

De Activate for HA

The RJDEACTHAV (De Active HA) command can be executed as part of the cut back process to:

- End defined H.A. Environments,
- and
- Start NON H.A. defined Environments.

