

HP Service Manager

适用于 Windows® 和 Linux® 操作系统

软件版本：9.30

文档引擎指南

文档发布日期：2011 年 7 月

软件发布日期：2011 年 7 月



法律声明

担保

HP 产品和服务的唯一担保由相应产品和服务随附的明示担保声明加以规定。此处的任何内容均不构成附加担保条款。对于本文档中出现的技术或编辑上的错误或遗漏，HP 不承担任何责任。

本文档中的信息如有更改，恕不另行通知。

有限权利声明

这是机密的计算机软件。拥有、使用或复制本软件均需要 HP 有效许可。遵照 FAR 12.211 和 FAR 12.212，商业计算机软件、计算机软件文档和商业项目技术数据依据供应商标准商业许可授权美国政府使用。

版权声明

© 版权所有 1994–2011 Hewlett-Packard Development Company, L.P.

商标声明

Adobe™ 是 Adobe Systems Incorporated 的商标。

Java 是 Oracle 和/或其分公司的注册商标。

Microsoft® 和 Windows® 是 Microsoft Corporation 在美国的注册商标。

Oracle® 是加州红木城 Oracle Corporation 在美国的注册商标。

UNIX® 是 The Open Group 的注册商标。

文档更新

本文档的标题页包含以下标识信息：

- 软件版本号，表示软件版本。
- 文档发行日期，会随每次文档的更新而更改。
- 软件发行日期，表示此版本软件的发行日期。

要检查最近是否有更新或要验证使用的文档是否为最新版本，请转到：

<http://h20230.www2.hp.com/selfsolve/manuals>

此网站要求您注册获取 HP Passport，然后才能登录。要注册以获取 HP passport ID，请转到：

<http://h20229.www2.hp.com/passport-registration.html>

或在 HP Passport 登录页上单击 **New users - please register**（新用户 - 请注册）链接。

如果您订阅了相应的产品支持服务，还将接收到全新或更新的版本。有关详细信息，请联系 HP 销售代表。

支持

访问 HP Software Support Online 网站，网址是：

<http://www.hp.com/go/hpsoftwaresupport>

此网站提供联系信息和有关 HP 软件提供的产品、服务和支持的详细信息。

HP Software Online Support 提供客户自解决功能。用户借助相关功能，可快捷高效地访问管理业务所需的交互式技术支持工具。作为尊贵的支持客户，您可以通过使用支持网站受益：

- 搜索感兴趣的知识文档
- 提交和跟踪支持案例和增强功能请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人
- 检查有关可用服务的信息
- 参加与其他软件客户的讨论
- 研究并注册软件培训

大多数支持区域要求您注册为 HP Passport 用户并登录。很多区域还要求提供支持合同。要注册 HP Passport ID，请转至：

<http://h20229.www2.hp.com/passport-registration.html>

要查找有关访问级别的详细信息，请转到：

http://h20230.www2.hp.com/new_access_levels.jsp

目录

文档引擎指南.....	1
目录.....	5
文档引擎概述.....	8
什么是文档引擎?	8
模块化方法的优点.....	8
一致性.....	8
开发时间减少.....	8
灵活性.....	8
访问文档引擎.....	8
对象.....	10
创建和更新对象.....	10
“对象定义”窗体和字段.....	11
“对象信息”选项卡字段描述.....	11
“锁定”选项卡字段描述.....	13
“修订”选项卡字段描述.....	14
“变量和全局列表”选项卡字段描述.....	14
“活动”选项卡字段描述.....	14
“警报”选项卡字段描述.....	15
“批准”选项卡字段描述.....	16
“管理队列”选项卡字段描述.....	18
“视图和模板”选项卡字段描述.....	19
“通知”选项卡字段描述.....	19
“搜索配置”选项卡字段描述.....	20
“已定义查询”选项卡.....	20
“范围”选项卡.....	21
状态记录.....	22
搜索.....	22
记录列表.....	22

查看单个记录.....	22
浏览记录.....	22
显示应用程序的集成提示.....	22
创建和更新状态.....	23
“状态定义” 字段描述.....	23
进程.....	26
创建和修改进程.....	26
“进程定义” 窗体和字段描述.....	26
“初始表达式” 选项卡.....	27
“初始 Javascript” 选项卡.....	27
RAD 选项卡.....	27
“最终表达式” 选项卡.....	28
“最终 Javascript” 选项卡.....	28
“下一进程” 选项卡.....	28
文档引擎资源.....	30
DEFAULT 对象.....	30
RAD 应用程序.....	30
se. search. engine.....	30
se. list. engine.....	30
se. view. engine.....	31
RAD 应用程序流程.....	31
显示应用程序的集成提示.....	32
se. search. engine 中的基本功能.....	32
se. view. engine 中的基本功能.....	33
se. list. engine 中的基本功能.....	35
局部变量.....	35
故障排除概述.....	38
检查通过文档引擎的应用程序路径.....	38
查找使用的数据库字典或对象.....	38
查找记录的状态.....	38
查找进程的名称.....	38
检查应用程序错误.....	39

打印变量的值或表达式的结果.....	40
工作单示例概述.....	42
创建表.....	42
向表中添加键字段.....	44
创建窗体.....	44
创建窗体的副本.....	45
创建工作单窗体的链接.....	46
创建序号文件.....	47
创建对象定义.....	47
创建初始化进程定义.....	48
创建显示应用程序屏幕定义.....	49
创建显示应用程序选项定义.....	51
创建状态定义.....	56
添加关闭工作单按钮.....	58
创建工作单向导.....	59
添加进程定义记录.....	62
为向导创建输入窗体.....	63
修改 close 和 update 突发事件窗体.....	64
在 Probsummary 表中创建链接的别名.....	67
修改 im.set.close 进程定义.....	68
测试工作单示例.....	70

第 1 章

文档引擎概述

文档引擎是一种定制工具，它可以对系统进行自定义而不需要更改 RAD。它提供一种集中方法来设置列出、查看和搜索等标准操作的特权和行为，从而增强了模块之间的一致性。文档引擎的三个主要组件是对象、状态和进程。进程可以在多个模块化编程中重复使用，集成是无缝的，可以减少开发时间。

什么是文档引擎？

文档引擎包含一系列工具和方法，用于开发和修改 Service Manager 工作流程。文档引擎最初的任务是开发一种基本功能集合，这些功能可以支持 Service Manager 内的多个模块，提高模块间用户界面的一致性并减少任何新模块所需的代码数量。

文档引擎扩展了显示应用程序功能，使其具备简化且更为广泛的操作，尤其是那些涉及多个应用程序调用的操作。此外，文档引擎还支持使用联接表和主窗体控制调用。旨在满足大多数客户的需求，同时保持灵活性。对象、状态和进程之间是层级式的关系。

文档引擎通过对象控制行为。无论何时打开窗体都会引用对象，对象会确定窗体状态的行为（如打开、列出、搜索等）。对象定义整体表行为。在对象中，状态描述记录所处的生命周期阶段，如打开、列出、搜索等。在状态中，根据用户在记录上启动的不同操作而执行不同进程。状态还定义系统如何显示记录，以及在特定时间或情况下可用的选项（操作）。例如，状态可以确定某个操作（例如“保存”）的用户访问权限。

进程是基于用户的操作从状态调用的。进程使用 RAD 表达式 JavaScript，并调用到现有 RAD 应用程序以针对当前记录执行操作。

模块化方法的优点

模块化设计的优点包括保持开发一致性、减少开发时间以及提高灵活性。

一致性

引擎允许所有应用程序通过使用相同的基础 RAD 应用程序进行运行，从而可确保 Service Manager 应用程序套件的一致性。由于核心功能（如锁定、警报、批准等）和记录列表功能均使用相同的代码库，因而这些功能对于任何模块都是相同的。

开发时间减少

文档引擎的模块化允许重复使用现有代码和进程。

灵活性

文档引擎使用进程记录作为修改 Service Manager 应用程序内部模块行为的一种机制。可以创建与基本系统行为不同的新进程，而不需要更改或删除系统中的原始进程。此外，在定制 Service Manager 以满足组织的特定要求时，可以使用您自己的进程来覆盖系统的基本进程，从而为系统开发人员提供了更大灵活性。

访问文档引擎

要访问文档引擎，请执行下列操作：

1. 启动 Service Manager 客户端并以管理员身份登录。
2. 在 Service Manager 系统导航栏中，单击**定制**。
3. 单击**文档引擎**。您可以在此处访问主要的三个区域以定义对象、进程或状态。还可以设置从对象调用的警报和批准，或者创建由对象使用的搜索配置记录。

第 2 章

对象

对象是定义的基本集合，用于确定记录的行为并设置定义和管理规则。对象与 Service Manager 中的数据库字典 (dbdict) 记录一一对应。如果某个表没有专用对象记录，则文档引擎将使用 DEFAULT 对象中的设置。所有对象都应该定义列表和默认状态。如果未定义，则默认状态为 db.browse、db.list、db.search 和 db.view。

备注：请勿修改或删除 DEFAULT 对象记录，因为这会导致不可预测的结果。

对象记录设置文档引擎内表的行为的定义和管理规则。例如，内容可能包括：

- 用于在对象内创建用户配置文件的应用程序，用户配置文件确定用户针对表的任何记录可采取的操作。
- 特殊情况下使用的状态记录（有关详细信息，请参阅“状态记录”一节）。
- 对象的类别、阶段和分页文件名称。
- 对象使用的数字记录名称。
- 对象如何使用锁定。
- 表中记录的修订的设置。
- 对于针对对象执行的处理，哪些变量应该可用。
- 使用对象时，哪些全局列表应始终可用。
- 活动记录的使用。
- 如何针对对象处理警报。
- 如何处理对象的批准。
- 工作队列的设置。
- 能够设置个人或全局视图和默认模板。
- 添加/更新/删除对象中记录的通知。
- 能够针对对象配置其他搜索选择。

要查看预置对象的列表，请在“对象定义”窗体中单击**搜索**。

要查看对象定义中字段和字段描述的列表，请参见[“对象定义”窗体和字段（第 11 页）](#)。

创建和更新对象

要创建对象，请执行下列操作：

1. 导航至文档引擎。请参见[访问文档引擎（第 8 页）](#)，以了解相关步骤。
2. 双击**对象**。“对象”窗体随即打开。
3. 使用“对象”窗体上的各个选项卡，填写必填字段以创建将执行所需功能的对象。有关更多信息，请参见字段描述。

要更新对象定义，请执行下列操作：

1. 访问文档引擎。请参见[访问文档引擎\(第 8 页\)](#)，以了解相关步骤。
2. 在“对象”字段中输入要更新的对象名称，或单击[搜索](#)来搜索对象。

“对象定义”窗体和字段

“对象定义”窗体的字段描述如下：

字段名	描述
文件名 <i>file.name</i>	输入对象的数据库字典名称，即使用与此对象名称对应的数据库字典名称。（必填字段）
公用名称 <i>message</i>	系统从“数据策略”填写此信息。它是对象的公用名称。公用名称可以是简单名称，例如“工作单”。
唯一键 <i>unique.field</i>	系统从数据库字典填写此信息。这是对象的唯一键。

此窗体还包括以下选项卡。在相应选项卡的字段描述表中对这些选项卡上的字段进行了描述。

- [“对象信息”选项卡字段描述\(第 11 页\)](#) - 指定对象的常规属性和行为。
- [“锁定”选项卡字段描述\(第 13 页\)](#) - 确定对象的锁定行为。
- [“修订”选项卡字段描述\(第 14 页\)](#) - 跟踪对象的修订。
- [“变量和全局列表”选项卡字段描述\(第 14 页\)](#) - 描述对象使用的局部变量和全局变量。
- [“活动”选项卡字段描述\(第 14 页\)](#) - 定义日志记录（活动）。
- [“警报”选项卡字段描述\(第 15 页\)](#) - 定义设置警报的位置以及生成警报的条件。
- [“批准”选项卡字段描述\(第 16 页\)](#) - 设置对象的批准选项。
- [“管理队列”选项卡字段描述\(第 18 页\)](#) - 控制队列显示和线程，以及能够创建视图（收藏箱）的人员。
- [“视图和模板”选项卡字段描述\(第 19 页\)](#) - 定义用户是否能够在对象上创建全局和个人视图及模板。
- [“通知”选项卡字段描述\(第 19 页\)](#) - 标识针对添加、删除或更新对象中任何记录的活动而自动发送的通知。
- [“搜索配置”选项卡字段描述\(第 20 页\)](#) - 控制搜索屏幕的“更多选择”选项卡上的可用选项。

“对象信息”选项卡字段描述

此选项卡指定对象的常规属性和行为。

“对象信息”选项卡的字段描述如下：

字段名	描述
描述字段 <i>desc. field</i>	指定对象的简短描述。
配置文件应用程序 <i>profile. appl</i>	指定创建配置文件的 RAD 应用程序，此配置文件可确定用户是否能够执行类似添加和删除的某些功能。例如，db.environment。(必填字段)
配置文件变量 <i>profile. variable</i>	指定在不访问环境记录的情况下调用此对象时可随时访问的变量。例如，\$L.env。(必填字段)
编号记录名称 <i>number. record</i>	定义对象的编号类别，此对象可以通过调用 getnumb 应用程序(通过进程、窗体控制或 RAD 调用)来访问。可用于检索作为记录的唯一键的序号。例如，EXWorkOrder。
类别表名称 <i>category. file. name</i>	指定与此对象的类别值关联的类别表相链接的表名称。显示这种类型的记录时，如果存在名为“类别”的字段，那么对象将搜索类别表名称和具有对应名称的记录。如果找到，系统会将类别文件名存储为变量：\$L.category。
阶段表名称 <i>phase. file. name</i>	指定与此对象关联的阶段表相链接的表名称(如果适用)。系统显示这种类型的记录时，如果存在名为“阶段”的字段，则该对象将转到阶段表名称并选择具有对应名称的记录。如果找到，系统会将阶段表名称存储为变量：\$L.phase。
分页表名称 <i>paging. file</i>	指定用于存储页面的表的名称。在应用最新更新之前将创建页面作为当前记录的完整副本。每次更新记录时都会执行该操作，且该操作会创建详细的审核跟踪。
主窗体控制 <i>master. fc</i>	如果记录对象存在“主窗体控制”记录，则指定记录的名称。“主窗体控制”允许在一个记录中定义“窗体控制”语句，这些语句会应用于区域中所有阶段，例如变更管理请求阶段。通常，主窗体控制的名称是“突发事件管理”、“问题管理”或“服务台”中数据库字典或者所有类别的名称。
Joindef <i>joindef</i>	指定用于连接多个表的 joindef 记录的名称，例如 joincomputer 或计算结果为有效 joindef 名称的表达式(例如 joindef in \$L.category)
状态字段 <i>statusField</i>	指定包含记录的状态信息的字段名称。
分配目标字段 <i>assignedToFields</i>	指定包含此对象的代理人名称字段的字段名称。文件夹权利验证是否已将记录分配给已登录的操作员时，将引用此字段。
工作组字段 <i>workgroupFields</i>	指定包含此对象的分配组字段的字段名称。文件夹权利验证是否已将记录分配给任何已登录操作员的工作组时，将引用此字段。
打开状态	指定打开新记录时要使用的状态定义记录。

字段名	描述
<i>open.state</i>	
关闭状态 <i>close.state</i>	指定用于关闭现有记录处理的状态定义记录。
列表状态 <i>list.state</i>	指定用于显示记录列表的状态定义记录。
默认状态 <i>default.state</i>	指定用作对象默认值的状态名称。若要编辑对象中的记录，则使用默认状态。
搜索状态 <i>search.state</i>	指定用于搜索的状态定义记录。
浏览状态 <i>browse.state</i>	定义记录使用锁定时应使用的状态定义记录。实质上，此字段定义某个记录当前由其他用户锁定时的只读状态。
手动状态 <i>manual.states</i>	指定可用于此对象的状态的数组，但不包括生命周期中的打开、关闭、列出、查看、搜索或浏览状态。

“锁定”选项卡字段描述

“锁定”选项卡确定对象的锁定行为。这意味着某人正在更新记录时，系统不允许其他用户更新同一记录。

“锁定”选项卡的字段描述如下：

字段	描述
使用锁定 <i>use.locking</i>	选中此复选框可启用锁定。
显示时锁定 <i>lock.on.display</i>	如果为 true 或条件计算结果为 true，则 Service Manager 会尝试在记录显示时立刻将其锁定。必须启用“使用锁定”，此字段才能生效。
锁定父记录 <i>lock.parent</i>	锁定当前记录及其父记录。例如，选中此字段并且某人正在更新变更任务时，该任务的变更请求也将锁定。
父记录 ID 字段 <i>parent.id</i>	在包含要锁定父记录的 ID 的当前记录中输入字段名称。
父文件名/对象 <i>parent.object</i>	包含父记录的表的名称。

字段	描述
监视变量 <i>watch.variables</i>	文档引擎在检查记录是否已更改时将使用监视变量。记录首次显示时监视变量必须为空。

“修订”选项卡字段描述

“修订”选项卡确定对象的修订行为。

该选项卡的字段描述如下：

字段名	描述
修订表名称 <i>revision.file</i>	指定表的名称，此对象记录的修订内容将存储在该表中。
最大修订数 <i>max.revisions</i>	指定允许对此对象进行的修订总数。如果保留空白，则表示没有上限。

“变量和全局列表”选项卡字段描述

“变量/全局列表”选项卡描述对象使用的局部变量和全局变量。

备注：全局变量在内存中构建和存储。

“变量/全局列表”选项卡的字段描述如下：

字段名	描述
局部变量 <i>local.variables</i>	输入可以在进程中使用的局部变量的列表。局部变量在应用程序中定义。局部变量将指定给您要创建的对象，并且可用于与此对象相关联的所有进程和状态。局部变量不能用于其他对象。
全局列表 <i>global.lists</i>	全局列表在创建后即可用于所有进程。如果“启动”列表中列出了全局列表，则可以在登录时构建全局列表。您每次访问对象时，全局列表均可用。

“活动”选项卡字段描述

“活动”选项卡定义了对象的更新（活动）日志记录。

“活动”选项卡的字段描述如下：

字段名	描述
活动日志表 <i>activitylog.file.name</i>	用于存储对象的活动日志条目的表的名称。

字段名	描述
选择列表变量 <i>activity.selection.var</i>	在更新窗体上使用的变量，用于显示操作员更新特定对象的记录时可以选择的类型。
发布链接 <i>activity.post.link</i>	用于发布信息的链接的名称。
如果不生成活动记录，则需要更新？ <i>activity.mandatory</i>	选中该复选框时表示需要活动更新。
更新字段 <i>update.field.var</i>	窗体上包含活动更新的字段或变量。仅当显示 如果不生成活动记录，则需要更新？ 时，才会出现此字段。
显示消息 <i>activity.mandatory.msg</i>	指示需要活动更新的消息。仅当显示 如果不生成活动记录，则需要更新？ 时，才会出现此字段。

“警报”选项卡字段描述

“警报”选项卡定义了设置警报的位置以及生成警报的条件。任何具有唯一键的对象都可以使用这些警报。

“警报”选项卡的字段描述如下：

字段名	描述
警报位置 <i>alert.location</i>	警报位置定义了可查找或存储要执行的“警报定义”的名称的位置。指定“警报定义”的存储位置： 记录：将警报存储在记录本身中。 类别：将警报存储在“对象信息”选项卡上定义的类别文件中。 阶段：将警报存储在“对象信息”选项卡上定义的阶段记录中。 对象：将警报存储在对象记录中。选择此选项后，将显示警报数组表，您可以填充要使用的警报。 备注：对象表需要有唯一键才能与警报相关。包含非空键但不包含唯一键的表不能使用警报。
警报条件 <i>alert.condition</i>	指定一个条件以确定是否要处理警报。例如，open in \$L.file~=false。
警报字段名称 <i>alert.field.name</i>	如警报位置所定义，指定包含实际警报名称的字段名称。
警报状态字段 <i>alert.status.field</i>	指定在处理警报之后当前记录中放置警报状态的字段。

字段名	描述
警报更新处理 <i>alert.update.process</i>	指定警报运行之后系统所执行的其他功能的处理记录的名称。
记录警报? <i>log.alerts</i>	如果选择该选项，则在处理警报之后会将其移动到 Alertlog 文件以保留警报历史记录。
处理父记录上的警报? <i>alerts.against.parent</i>	如果已在“锁定”/“修订”选项卡上选择了“锁定父记录”字段，并且选中此复选框，那么警报激活时还会向父记录注册。
重新计算警报的条件 <i>alert.recalc</i>	指定条件以确定是否重新计算关于现有警报的条件。
重置警报的条件 <i>alert.reset</i>	确定删除现有警报的时间，并重新计算所有条件。

“批准”选项卡字段描述

“批准”选项卡为对象设置了批准选项及其关联的通知。批准在 ApprovalDef 文件中定义。

“批准”选项卡的字段描述如下：

字段名	描述
批准条件 <i>approval.condition</i>	如果批准条件计算结果为 true，则在对象记录上使用批准。
批准位置 <i>approval.location</i>	指示存储批准信息的位置：记录、阶段、对象或类别。
批准字段名 <i>approval.field.name</i>	包含在批准位置中定义的表中实际批准名称的字段名称。
批准状态字段 <i>approval.status.field</i>	当前记录中用于存储批准状态的字段。
批准组 <i>approval.groups</i>	存储一个变量，包含当前用户必须属于的组，以便对此对象执行批准操作。
批准类型 <i>appr.cond.type</i>	有四种预定义的批准类型： 须经所有成员批准：在系统将记录的状态设置为“已批准”之前，“批准定义”中定义的所有组/操作员都必须执行批准操作。如果只获得一个或一些（但并非所有）组/操作员的批准，则状态将设置为“待定”。

字段名	描述
	<p>须经任一成员批准：此记录只需获得批准组/操作员中任一成员的批准即可处于批准状态。</p> <p>须经法定多数批准：此记录需要获得批准组的多数成员的批准才能处于批准状态。</p> <p>须经所有成员批准 - 一票即刻否决：所有组/操作员都必须批准该记录。只要有一个成员进行了否决，则该记录的状态即更改为“拒绝”。此种情况下所有其他批准人都无需执行任何操作。</p>
批准通知 <i>single.notify.approval</i>	选择批准请求的情况下要运行的通知。
拒绝通知 <i>single.notify.denial</i>	选择有一个批准人拒绝请求的情况下要运行的通知。
撤销通知 <i>single.notify.retraction</i>	选择撤销上一操作时要运行的通知。
最终批准通知 <i>final.notify.approval</i>	选择最终获得批准后要发送的通知。
最终拒绝通知 <i>final.notify.denial</i>	选择拒绝请求时要发送的通知。
批准 FC <i>appr.fc</i>	指定获得批准后要运行的“窗体控制”记录的名称。
批准进程 <i>approval.process</i>	选择批准记录时要运行的进程。
拒绝进程 <i>denial.process</i>	选择拒绝记录时运行的进程。
打开时执行预批准 <i>preapprove.cond</i>	<p>确定是否应当自动批准记录。</p> <p>如果该条件成立，并且该用户属于某一个待定批准组，则自动进行批准。如果该用户不属于某一个待定批准组，则不会自动执行批准操作，并且必须通过常规批准进程。默认值为 true。</p>
记录批准? <i>log.approvals</i>	选中此复选框以将批准历史记录记录在 ApprovalLog 表中。

字段名	描述
需要批准注释 <i>approval.comments</i>	如果选中，则要求批准人给出批准注释。
汇总批准？ <i>aggregate.approvals</i>	如果选中，则批准将累积。
重新计算批准的条件 <i>approval.recalc</i>	指定条件以确定是否重新计算关于现有批准的条件。
重置批准的条件 <i>approval.reset</i>	确定删除现有批准的时间，并重新计算所有条件。

“管理队列”选项卡字段描述

“管理队列”选项卡控制队列、视图的显示方式以及线程。没有关联对象记录的文件的“数据策略”中提供了这些相同字段，或者这些字段从对象记录（如果存在）虚拟链接到数据库字典记录。

“管理队列”选项卡的字段描述如下：

字段名	描述
管理条件 <i>scm.condition</i>	指定条件以仅允许某些用户查看显示此对象记录的队列。例如，在 <code>\$G.pm.environment</code> 中浏览。
管理显示窗体 <i>scm.manage.screen</i>	选择要用于显示视图的窗体。预置 Service Manager 有默认的显示窗体：如果未选择任何其他窗体，则会使用 <code>sc.manage.generic</code> 。HP 建议您不要更改 <code>sc.manage.generic</code> 窗体。
管理默认视图 <i>scm.inbox</i>	选择此队列的默认视图。通过为特定用户指定用户视图，可以为 HP Service Manager 管理队列设置特定的视图列表。如果某个用户没有定义特定的视图，则会使用默认用户视图。
管理默认查询 <i>scm.query</i>	指定在未选择默认视图时要运行的默认查询。
默认查询说明 <i>scm.query.name</i>	为上面的字段指定名称。可以将消息与此字段相关联。例如， <code>scmsg(491, "us")</code> 。
线程视图 -> 搜索？ <i>scm.thread.list.edit</i>	指定 <code>true</code> 或指定计算结果为 <code>true</code> 的表达式以在执行搜索时打开新的线程。
搜索窗体（如果需要） <i>scm.search.format</i>	选择默认搜索窗体。

字段名	描述
线程搜索 -> 列表? <i>scm.thread.search.list</i>	指定 true 或计算结果为 true 的表达式，以便在用户发现有可查看记录时获取新线程。
线程列表 -> 编辑? <i>scm.thread.list.edit</i>	指定 true 或计算结果为 true 的表达式，以便在用户所选查看记录不属于记录列表时获取新线程。
线程视图 -> 编辑? <i>scm.thread.inbox.edit</i>	指定 true 或计算结果为 true 的表达式，以便在用户查看不属于队列的现有记录时打开新线程。
允许添加条件 <i>scm.add.condition</i>	指定表达式，评估操作员添加记录的能力。
添加/打开应用程序 <i>scm.add.appl</i>	指定添加或打开记录时要调用的应用程序的名称。
参数名称 <i>scm.add.names</i>	指定要传递到“添加/打开应用程序”字段中指定应用程序的参数名称。
参数值 <i>scm.add.values</i>	指定要传递到“添加/打开应用程序”字段中指定的应用程序的参数值。

“视图和模板”选项卡字段描述

“视图/模板”选项卡定义用户是否可以创建全局和个人视图以及模板支持。

该选项卡的字段描述如下：

字段名	描述
可以创建个人视图 <i>personal.inbox</i>	指定计算结果为 true 或 false 的条件。若为 true 则允许用户创建个人视图。
可以创建系统视图 <i>global.inbox</i>	指定计算结果为 true 或 false 的条件以确定用户是否可以创建全局视图。
默认模板 <i>default.template</i>	指定将用于此表中记录的默认模板的名称。
支持模板? <i>supportTemplates</i>	选中此复选框可为对象启用模板支持。

“通知”选项卡字段描述

“通知”选项卡标识针对添加、更新或删除对象的活动自动发送的通知。

“通知”选项卡的字段描述如下：

字段名	描述
添加 <i>notification.add</i>	选择将记录添加到表中时自动发送的通知。
更新 <i>notification.update</i>	选择更新表中记录时自动发送的通知。
删除 <i>notification.delete</i>	选择从表中删除记录时自动发送的通知。

“搜索配置”选项卡字段描述

“搜索配置”选项卡控制搜索屏幕上“更多选择”选项卡中的可用选项。

“搜索配置”选项卡的字段描述如下：

字段名	描述
表名称 <i>tablename</i>	将查询的表的名称。
搜索窗体 <i>searchFormat</i>	用于“更多选择”选项卡的子窗体的名称。
初始化进程 <i>init.process</i>	显示搜索窗体之前将运行的进程的名称。
允许高级查找 <i>allowAdvAccess</i>	定义条件以确定是否允许高级查找。

“已定义查询”选项卡

“已定义查询”选项卡定义了要在搜索屏幕的“更多选择”选项卡（例如，“搜索突发事件”窗体上的“更多选择”）上使用的查询语句和标签。相应字段在 SearchConfig 表中定义。

“已定义查询”选项卡的字段描述如下：

字段名	描述
ID <i>id</i>	提供查询的唯一 ID，并且不能包含特殊字符（包括空格）。

字段名	描述
查询 <i>query</i>	使用系统语言语法的查询表达式。例如， <code>assignee.name=operator()</code> 。
描述 <i>description</i>	为“更多选择”选项卡上的复选框提供标签。

“范围”选项卡

使用“范围”选项卡可以轻松设置搜索来搜索开始和结束日期范围。若要执行此操作，请单击[修改配置](#)链接以定义开始和结束日期变量，从而用作窗体中的输入。在“变量 1”列中输入此变量，并使用“字段”和“运算符 1”列定义将执行的查询。修改配置与在“已定义查询”选项卡上进行修改的方式相同。

“范围”选项卡的字段描述如下：

字段名	描述
字段 <i>fieldName</i>	指定要在查询中使用的表中的字段。
运算符 1 <i>operator1</i>	查询中的比较运算符。（例如：>=）
变量 1 <i>variable1</i>	指定用作窗体上的输入的变量。
特殊类型 <i>specialType</i>	当前未使用。

第 3 章

状态记录

状态记录 (State) 由对象调用，由进程来定义。状态记录包含有关记录的显示外观以及在特定时刻的行为的信息。

状态记录定义可以包括以下内容：

- 状态名称。
- 用于显示一条或多条记录的显示屏幕。
- 当列表首次显示时使用的初始化进程。
- 用于显示数据的窗体。
- 指示用户是否可以修改记录的输入条件。
- 当用户触发特定的显示选项（显示操作）时运行的进程。

使用的状态记录取决于用户查看的记录数量。有用于搜索记录、查看列表、查看单个记录和浏览记录的预置状态记录。

搜索

如果当前文件变量中没有记录，则将假设用户处于“搜索”模式中。所使用的状态是在表的对象记录中定义的“搜索状态”。默认搜索状态为“db.search”。

记录列表

查看记录列表时，使用在表的对象记录中定义的“列表状态”。默认列表状态为“db.list”。

备注：在某些更早版本的 Service Manager 中，记录列表称为 QBE 列表。

查看单个记录

在查看单个记录时，将会首先检查该记录在数据库字典中是否包含了“状态”字段。如果该字段存在并已填写，则该字段的内容将用作记录的当前状态。如果此字段不存在或是 NULL，则将使用在表的对象记录中定义的“默认状态”。默认状态的默认值为“db.view”。

浏览记录

以只读模式浏览记录时，使用在表的对象记录中定义的“浏览状态”。注意，只有使用锁定的文件需要浏览状态。虽然在没有更新权限的情况下查看记录似乎是在浏览屏幕，但是它所使用的状态为默认状态，而不是浏览状态。没有默认浏览状态。

显示应用程序的集成提示

如果显示屏幕上没有任何相关联的显示事件，则在发生“OnFormModified”时，系统将自动使用执行 se.lock.object 的 se.default 显示事件。如果显示选项记录的“修改记录”复选框已激活，则在按下按钮时将锁定该记录。

创建和更新状态

要创建新状态，请执行下列操作：

1. 访问文档引擎。请参见[访问文档引擎\(第 8 页\)](#)。
2. 在“状态定义”窗体上，填写必填字段，创建一个用于执行所需功能的状态。请参见[“状态定义”字段描述\(第 23 页\)](#)中的字段描述。

要修改现有状态，请执行下列操作：

1. 访问文档引擎。请参见[访问文档引擎\(第 8 页\)](#)。
2. 在“状态”字段中输入要修改的状态的名称，或单击[搜索](#)来搜索该状态。

“状态定义”字段描述

“状态”定义记录定义了记录在特定时间段内的行为和显示规范。

在“状态”记录中，可以定义非基础方法，也可以修改基础方法的行为。以下各节中描述了基础方法：“se.search.engine 中的基础功能”、“se.view.engine 中的基础功能”和“se.list.engine 中的基础功能”。它们包括了诸如保存、查找、填充、确定、取消和搜索之类的功能。

“状态定义”窗体的字段描述如下：

字段名	描述
状态 <i>state</i>	指定状态的名称。(必填字段)
显示屏幕 <i>display.screen</i>	与状态相关联的显示屏幕。
初始化进程 <i>init.process</i>	在输入状态之前应运行的进程的名称。
窗体 <i>format.name</i>	指定打开“状态”记录的窗体。此窗体存储在记录中，可以是变量，也可以是硬编码。窗体的名称可以硬编码，可以包含在变量中，也可以通过系统语言表达式从记录中检索。
输入条件 <i>input.condition</i>	输入条件的计算结果用于查看状态，并确定记录是否只读。指定 false 表示只读，指定 true 表示可编辑。也可以输入计算结果为 true 或 false 的表达式。否则，输入 true 或者输入计算结果为 true 或 false 的表达式。
非基础方法	
显示操作 <i>process.label</i>	指定用户输入后，显示操作的操作参数。在用户单击按钮或菜单项显示选项后，其中的操作字段会发出显示操作。

字段名	描述
进程名称 <i>valid.process</i>	指定由于执行操作而运行的进程的名称。
条件 <i>process.condition</i>	为在要调用的进程名称中指定的进程指定计算结果为 true 的表达式。
先保存 <i>run.save.before</i>	选择在运行此进程之前先运行保存进程。键入 true 可先保存；否则，请输入 false。默认设置为 false。

第 4 章

进程

进程是文档引擎可以使用的最小可分离作业单位，数据操作即在此级别上实现。用户可以创建自己的进程，也可以使用 Service Manager 附带的 700 多个进程中的任意一个。进程定义由初始表达式、RAD 和最终表达式组成，在“初始表达式”、“初始 Javascript”、RAD、“最终表达式”、“最终 Javascript”、“下一进程”选项卡以及下一进程的可选调用上输入。表达式是用标准 RAD 表达式和/或 JavaScript 编写的。

创建和修改进程

要创建进程，请执行下列操作：

1. 访问文档引擎。请参见[访问文档引擎\(第 8 页\)](#)。
2. 使用“进程”面板上的各个选项卡，填写必填字段以创建将执行所需功能的进程。请参见[“进程定义”窗体和字段描述\(第 26 页\)](#)，以了解更多相关信息。

要修改进程，请执行下列操作：

1. 访问文档引擎。请参见[访问文档引擎\(第 8 页\)](#)。
2. 在“进程名称”字段中输入要修改的进程的名称，或单击**搜索**来搜索进程。

“进程定义”窗体和字段描述

在“进程定义”窗体上，可以定义新进程或编辑现有进程。进程运行代码或表达式以执行用户选择的的操作。

“进程定义”窗体的字段描述如下：

字段名	描述
进程名称 <i>process</i>	指定进程的名称。(必填字段)
保存光标位置 <i>save.cursor.position</i>	如果要在操作(例如填充)之后返回到同一光标位置，则选中此框。
完成后运行标准进程? <i>run.standard</i>	如果选中，则系统将在完成当前操作或进程之后运行标准进程。例如，像“保存”这样的标准进程。如果已经创建了保存进程，并要在运行您定义的保存进程之后运行“文档引擎”附带的保存进程，则请选中此框。标准进程在此文档的基础功能部分定义。
在窗口中运行? <i>run.in.window</i>	如果选中，则进程在单独的窗口中运行。

字段名	描述
窗口标题 <i>window.name</i>	如果选中 在窗口中运行? 复选框，则请指定窗口标题。scmsg 表达式可用于本地化后的窗口标题，例如 scmsg(1980, "us")。

除了上面描述的字段以外，此窗体上还有一些选项卡，可用于进一步定义进程。这些选项卡包括：

- [“初始表达式”选项卡\(第 27 页\)](#)
- [“初始 Javascript”选项卡\(第 27 页\)](#)
- [RAD 选项卡\(第 27 页\)](#)
- [“最终表达式”选项卡\(第 28 页\)](#)
- [“最终 Javascript”选项卡\(第 28 页\)](#)
- [“下一进程”选项卡\(第 28 页\)](#)

“初始表达式”选项卡

“初始表达式”选项卡定义了初始 Javascript 之前以及 RAD 选项卡上定义的 RAD 调用之前运行的初始表达式。初始表达式是使用标准 Service Manager 表达式编写的。

“初始 Javascript”选项卡

“初始 Javascript”选项卡可定义在 RAD 选项卡上定义的 RAD 之前运行的初始 JavaScript 表达式。

RAD 选项卡

RAD 选项卡定义了 RAD 前表达式、RAD 调用和 RAD 后表达式，它们均作为进程的一部分运行。

RAD 选项卡的字段描述如下：

备注：对于此选项卡上定义的每个 RAD 应用程序，这些字段都是重复的。

字段名	描述
在 RAD 调用之前计算的表达式 <i>pre.rad.expressions</i>	指定在“RAD 应用程序”字段中定义的 RAD 应用程序之前运行的表达式。传递到 RAD 应用程序的所有参数值必须是变量或表达式形式。必须在 RAD 前表达式中指定这些变量的值，例如 \$L.value.name="Test"。
RAD 应用程序 <i>application</i>	指定要运行的 RAD 应用程序的名称。

字段名	描述
条件 <i>rad.condition</i>	指定应当执行(条件计算结果为 true) 或跳过(条件计算结果为 false) RAD 应用程序的条件。
参数名称 <i>names</i>	指定传递到 RAD 应用程序的参数名称。
参数值 <i>values</i>	指定传递到 RAD 应用程序的参数值。这些值必须是变量或表达式的形式。将字符串值包含在双引号中 (“Wizard Name”) 时可以传递。
RAD 后表达式 <i>post.rad.expressions</i>	指定在 RAD 应用程序完成之后将运行的任何 RAD 表达式。

“最终表达式” 选项卡

“最终表达式” 选项卡定义了 RAD 选项卡处理完成之后运行的最终表达式。最终表达式是使用标准 Service Manager 表达式编写的。

“最终 Javascript” 选项卡

此选项卡上定义的 Javascript 在最终表达式以及 RAD 选项卡上的 RAD 应用程序之后运行。

“下一进程” 选项卡

此选项卡指定当前进程完成后要运行的进程。

“下一进程” 选项卡的字段描述如下：

字段名	描述
下一进程 <i>next.process</i>	指定要运行的下一个进程的名称。
条件 <i>process.condition</i>	指定与“下一进程” 字段中的进程关联的条件，该条件计算结果为 true 或 false。例如，如果为 true，则下一进程为 sm.close。

第 5 章

文档引擎资源

文档引擎包括可以由任何对象使用的本地变量和基本功能。基本功能是预定义的，用于执行用户可用的标准操作。例如，在用户完成更新记录后单击“保存”按钮时，文档引擎将处理此请求并将更改写入到数据库。可以在状态记录中覆盖基本功能以运行不同的进程。

DEFAULT 对象

数据库管理器访问过的任何文件将自动使用为相应对象定义的规则和进程。如果没有为文件定义对象，则将使用 DEFAULT 对象。DEFAULT 对象会复制数据库管理器早期版本的功能。

系统管理员还可以通过 DEFAULT 对象访问拥有相应对象的文件，方法是选中“管理模式”复选框，然后通过数据库管理器访问该文件。此复选框仅对系统管理员可用。具有功能字“AlwaysAdmin”的用户在访问 Service Manager 内的信息时将始终使用 DEFAULT 对象。不建议向用户分配功能字。

文档引擎使用环境配置文件。在以前版本的系统中，系统管理员拥有所有权限（添加、更新、删除），而不考虑窗体控制设置。使用文档引擎，授予的权限将与窗体控制记录中定义的权限进行匹配。希望采用旧授权方法的管理员只需在 DEFAULT 对象记录中将“配置文件应用程序”设置从“db.environment”更改为“db.environment.sysadmin”即可。

RAD 应用程序

使用数据库驱动的应用程序，用户可以使用以下三个基本记录集中的任何一个：

- 零个记录 - 搜索信息时。
- 多个记录 - 查看信息列表时。
- 一个记录 - 对单个记录进行更改时。

使用数据时，透过用户透视图可以看到：

- 您正在查看可以在其中输入搜索信息（或创建新记录）的空白窗体；
- 或正在更新/查看单个记录；
- 或已执行搜索，且现在拥有一组以列表形式显示的包含多个记录的记录集。

文档引擎使用三个主要 RAD 应用程序来实现此构想。所使用的 RAD 例程由任意时刻显示的记录的数量来确定。

se. search. engine

没有要查看的记录时，使用搜索引擎。此例程的主要用途是建立查询公式并从正确的表中选择记录。此例程也可用于将信息初始输入到空白记录中以便向数据库中添加新记录。默认状态为 db. search。

se. list. engine

列表引擎显示多个记录。使用此列表引擎，用户可以从列表中选择特定记录，或者对整个记录列表执行操作。默认状态为 db. list。

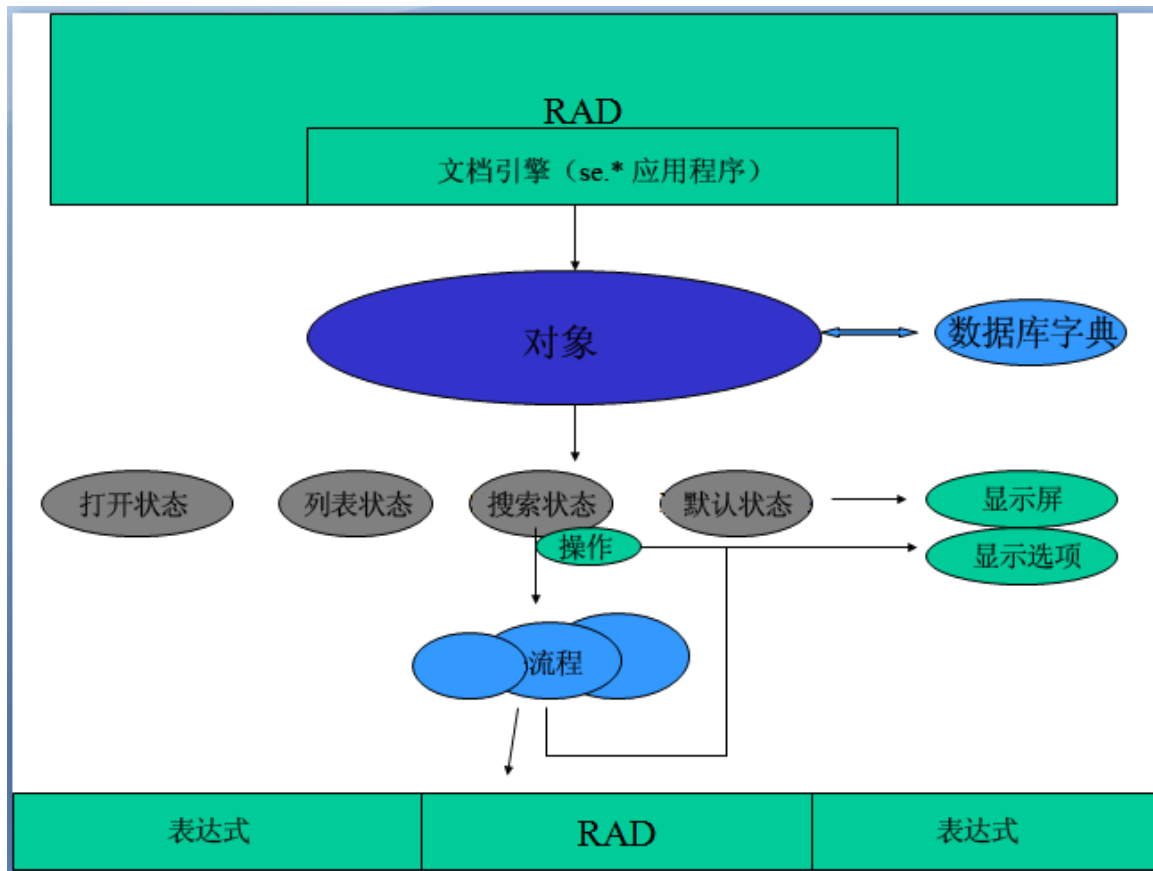
se.view.engine

查看引擎显示单个记录。此应用程序用于执行针对特定记录的操作，例如更新或删除。默认状态为 db.view。

备注：使用 Service Manager 的记录列表功能时，查看引擎可同时用于列表和单记录信息。

RAD 应用程序流程

下图概述了其他定制工具与文档引擎的交互方式。



备注：与文档引擎相关的所有 RAD 应用程序均以“se”开头，如 se.view.engine。文档引擎将每个记录视为用户可用的文档。

在使用文档引擎显示文件时，它将始终使用三个应用程序中之前在文档中指定的那个应用程序。显示屏、显示选项和窗体由当前记录的状态确定。当触发某个显示选项时，在执行了标准显示功能后，引擎会根据当前状态记录中定义的可用进程来检查该选项的“操作”。如果所定义进程的条件计算结果为 true，则引擎将针对当前记录（或记录集）执行该进程。如果未在状态记录中定义操作，则引擎将会检查该操作是否在当前应用程序中定义为“基本进程”。如果是，则系统将针对记录执行该基本进程。如果不是，将不采取任何操作。

显示应用程序的集成提示

如果显示屏幕上没有任何相关联的显示事件，则在发生“OnFormModified”时，系统将自动使用执行 `se.lock.object` 的 `se.default` 显示事件。

如果显示选项记录的“修改记录”复选框已激活，则在按下按钮时将锁定该记录。

se.search.engine 中的基本功能

显示操作	描述
back	退出记录。
find	显示链接到字段的记录中的详细信息。
fill	将链接记录中的信息填入目标记录。将源记录字段中的数据放入目标字段。
advanced	启动“高级搜索”进程，它可以向用户提供更多搜索选项。通常，此操作仅对系统管理员可用，但可以将此选项设置为对其他用户可用。
clear	清除当前屏幕。
openinbox / inbox	提示用户打开与当前文件相关联的视图。这是将结果显示为视图的预定义查询。
search	通过使用提供的信息创建搜索查询并在记录列表中显示结果来执行标准搜索。
add	执行“添加”窗体控制并尝试向数据库添加记录。
restore	还原屏幕内容（在执行“清除”操作后）。
irquery	运行“IR 查询”文本搜索。
validitylookup	执行 Service Manager 有效性查找。有效性查找通过有效性表中的条目来验证某特定值是否对定义的字段有效。使用有效性查找，系统可以验证用户输入的值是否符合验证规则。
expandarray	扩展数组提供了用于编辑数组的其他功能，如插入行和删除行。
reset	重置当前文件。此操作将删除表数据库中的 所有 记录。
regen	重新生成当前文件的索引。此操作仅适用于拥有 IR 键的表，并将仅执行一次 IR 重新生成。
export/unload	启动“无记录”导出/卸载进程。仅用于卸载没有数据的空 DBDICT 定义。
views	选择该选项后，系统将会向用户呈现显示当前记录的替代窗体的列表。

显示操作	描述
findrevision	显示此对象中记录的修订。
initrevision	创建此对象中记录的修订。
newsite	
newview	
newTable	
addFilter	由客户在标准搜索屏幕中执行“高级筛选”搜索时触发。
editFilter	由客户在标准搜索屏幕中执行“高级筛选”搜索时触发。
addCompound	由客户在标准搜索屏幕中执行“高级筛选”搜索时触发。
removeSelection	由客户在标准搜索屏幕中执行“高级筛选”搜索时触发。

se.view.engine 中的基本功能

显示操作	描述
save	在“窗体控制”中执行更新，如果更新有效，则将更新数据库中的记录。
add	在“窗体控制”中执行添加，如果新记录有效，则将新记录添加到数据库。
ok	如果记录已经更改，则执行保存，否则退出。
reselect	从数据库中重新选择当前记录（如果其已被更改）。
fill	将链接记录中的信息填入目标记录。将源字段中的信息放入目标字段。
find	显示链接到字段的记录中的详细信息。
next	移动到列表中下一记录（检查更改之后）。
previous	移动到列表中上一记录（检查更改之后）。
back	退出列表或搜索窗体。
menu	退出到调用菜单。
delete	删除“窗体控制”并在验证请求后尝试从数据库中删除记录。
views	选择该选项后，系统将会向用户呈现显示当前记录的替代窗体的列表。

显示操作	描述
print	打印当前记录。
printlist*	打印记录的当前列表。
validitylookup	执行 Service Manager 有效性查找。有效性查找通过有效性表中的条目来验证某特定值是否对定义的字段有效。使用有效性查找，系统可以验证用户输入的值是否符合验证规则。
export/unload	标准 Service Manager 导出/卸载功能(多记录)。
massunload*	为记录列表提供卸载功能。
massadd*	添加完全复制了现有记录集合的新集合，除非新记录集合将拥有随新数据一同更新的唯一键值。在处理结束时表中记录将是原有记录的两倍。
massupdate*	向记录列表中的指定字段集合提供相同的更新。
massdelete*	从表中删除特定的记录集合。
irquery	运行“IR 查询”文本搜索。
expandarray	扩展数组提供了用于编辑数组的其他功能，如插入行和删除行。
count*	对记录列表中的记录进行计数，并且显示总计数。
audithistory	调用标准审核历史记录例程。审核历史记录由待审核字段确定，如 auditdef 表中所定义。
inbox.save / inbox*	将当前查询保存为视图。
approval.log	查看当前记录的批准历史记录。
alert.log	查看当前记录的警报历史记录。
alerts	查看当前记录的当前警报和计划警报。
pagelist / listpages	页面列表显示记录的完整历史记录。启用分页后，每次执行更新时，记录的副本都将被写入到页面文件。页面列表基于分页文件显示记录的完整历史记录。
clocks	显示与此记录关联的时钟记录。
xmlfill	处理 XML 字段，如服务目录中的用户选项。
createTemplate	在当前记录之外创建模板记录。
applyTemplate	将现有模板应用到当前记录。

备注： * 这些命令只有在使用了记录列表功能时才会应用。

se.list.engine 中的基本功能

显示操作	描述
exit(或 back)	返回到搜索(或调用) 窗体。
inbox.save / inbox	提示用户将当前查询保存为新视图。
count	执行标准计数功能，并显示总记录数。
refresh	刷新使用当前查询的列表。
big.green	完全退出当前模块。“绿色大箭头”
print	打印显示的记录列表。
views	选择该选项后，系统将会向用户呈现显示当前记录的替代窗体的列表。
export/unload	标准 Service Manager 导出/卸载功能(多记录) 。
massadd	添加完全复制了现有记录集合的新集合，除非新记录集合将拥有随新数据一同更新的唯一键值。在处理结束时表中记录将是原有记录的两倍。
massupdate	向记录列表中的指定字段集合提供相同的更新。
massdelete	从表中删除特定的记录集合。

局部变量

局部变量以 \$L. 开头，仅在当前执行的 RAD 应用程序内存在。当服务器退出 RAD 应用程序时，将自动清除局部变量。

以下是在文档引擎中使用的“标准变量”列表：

\$L.action - 显示选项的显示操作值

\$L.bg - 后台标志

\$L.category - 类别记录(如果可用)

\$L.env - 当前环境记录

\$L.exit - 内部退出参数

\$L.file - 当前文件变量

\$L.file.save - 处于原始状态的记录的副本

\$L.format - 用于显示记录的窗体名称

\$Irspread - 确定 IR 发现选项：0=浅度搜索，2=深度搜索，4=完全匹配

\$L.mode - 查看的记录所处的模式，通常“添加”表示创建新记录，“更新”表示修改现有记录，“关闭”表示完成现有记录的处理

\$L.mult - 一种标志，\$L.file 变量中有多个记录时为 True

`$L.object` - 对象记录

`$L.phase` - 阶段记录(如果可用)

`$L.sql` 或 `$L.query` - 当前查询

`$L.sort` - 当前排序顺序

`$L.state` - 系统正在使用的状态记录(即记录所处的状态) 。

“查看”模式中可用的变量(查看单个记录时)

`$L.fc` - 详细窗体控制记录的副本

`$L.fc` - 主窗体控制记录的副本

第 6 章

故障排除概述

为了成功排除“文档引擎”故障，您需要收集以下信息：

- 正在使用的是哪个数据库字典和对象
- 记录处于什么状态
- 正在调用的是什么进程
- 再现（STR）问题需遵循的步骤

检查通过文档引擎的应用程序路径

在排除文档引擎故障的过程中，与排除任何 Service Manager 应用程序故障一样，在 Service Manager `sm.ini` 文件中输入 **RTM:3** 和 **debugdbquery:999**，然后启动新客户端连接。除非此用户进程是调用文档引擎进程调试的第一个进程，否则在进行此跟踪时它可能不会在 `sm.log` 文件中显示所选状态或进程记录，但是会给出关于正在调用的进程的有用提示。

查找使用的数据库字典或对象

若要确定正在使用的数据库字典或对象，请搜索日志文件。

日志示例：

```
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object get.object select CPU( 0 1411 )
1320 07/18/2006 11:00:36 (0x0129AC08) DBACCESS - Cache Find against file Object found 1
record, query:file.name="pcsoftware"
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object set.access process CPU( 0 1411 )
```

查找记录的状态

下一个要查看的问题是记录处于何种状态。若要查找该信息，请在跟踪 `sm.log` 中搜索以下信息：

日志示例：

```
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state select.state select CPU( 0 1491 )
1320 07/18/2006 11:00:48 (0x01292FB0) DBACCESS - Cache Find against file States found 1
record, query:state="pcs.list"
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state exit.normal process CPU( 0 1491 )
```

查找进程的名称

还可以查找进程的名称，方法是在包含下面示例中所显示跟踪的 `sm.log` 中进行搜索。

日志示例：

```
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process select.process select CPU( 0
1542 )

1320 07/18/2006 11:00:50 (0x00B56810) DBACCESS - Cache Find against file Process found 1
record, query:process="upgrade.pcs"

1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process run.pre.exp process CPU( 0 1542
)
```

检查应用程序错误

进程可调用大量 RAD 应用程序和执行大量表达式，并且随后可能会调用更多进程。如果任何应用程序或表达式由于错误语法或错误逻辑导致错误退出，则可以在 `sm.log` 文件中找到此信息。

日志示例：

```
Process panel run.pre.exp in RAD se.call.process encountered error in line 1
(se.call.process,run.pre.exp)

Cannot evaluate expression (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Bad arg(2) oper = (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Bad arg(2) oper = (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Bad arg(2) oper nullsub (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Bad arg(2) oper in (se.call.process,run.pre.exp)
Cannot evaluate expression (se.call.process,run.pre.exp)
Unrecoverable error in application:se.search.objects on panel call.list.engine
Unrecoverable error in application:se.list.engine on panel call.process.1
Unrecoverable error in application:se.call.process on panel run.pre.exp
```

在此非预置的示例中，`se.call.process` 和 `run.pre.exp` 中发生了错误，换句话说，在计算进程的初始表达式时发生错误。若要找出导致问题的进程，请完成上述步骤，并注意下面行中的进程：

```
DBACCESS - Cache Find against file Process found 1 record, query:process="upgrade.pcs"
```

根据名称 `upgrade.pcs` 转到进程记录，并在“初始表达式”选项卡上检查所有语句。在此特定情况下，表达式将包含单词 `nullsub`。例如，对于此测试有问题的表达式可能是：

```
$.licount=nullsub($.licount, anynumberIwant)
```

变量 `anynumberIwant` 不是有效字段、常量或变量，因此必须对其进行更改才能防止此问题发生。

打印变量的值或表达式的结果

在文档引擎中，通过 workflows 的路径通常是由指定给字段或变量的值确定的。为了确定指定给某个影响 workflows 的字段或变量的值，请使用 JavaScript `print()` 函数，或者在 RAD 表达式中使用 `log rtecall($L,void=rtecall("log", $L.rc, "message"))`。消息可以是连接字符串，例如：

```
$L.message="The value of $L.test is " + $L.test
```

其中 `$L.test` 是分配了字符值的变量。

第 7 章

工作单示例概述

什么是工作单？工作单是分配给某位工程师的特定任务，用以确定解决突发事件或可能存在的任何其他模块（如果工作单系统已扩展为包含更改、问题或已知错误）所必需的某些活动。以下示例基于 Service Manager 版本 7.11，旨在使用户能够创建、更新和关闭工作单以用于“突发事件管理”。可以轻松对其进行修改以适用于其他 Service Manager 应用程序。

此示例将创建工作单系统以演示如何使用文档引擎。此工作单系统允许用户为突发事件创建工作单，并查看这些工作单的状态。它还允许用户从“突发事件管理”查看并更新工作单。所有的工作单均与系统中特定突发事件相关联，且该突发事件只有在其所有工作单都关闭之后才能关闭。

此示例适用于定制或自定义系统的用户。您应当全面了解以下定制功能：

- 数据库词典，用于创建新表。
- 窗体设计器，用于修改预置窗体和创建新窗体。
- 向导创建工具。

工作单示例将引导您完成以下步骤：

- 使用数据库词典实用程序创建新数据库词典（dbdict）。
- 指定表的关键字段。
- 使用“窗体设计器”为 EXWorkOrder 表创建 EXWorkOrder 窗体。
- 创建序号文件。
- 修改窗体以使其包含下拉列表。
- 为 EXWorkOrder 窗体创建链接。
- 为向导创建窗体以用于收集工作单的信息。
- 修改 close 和 update 突发事件窗体。
- 创建别名以将工作单链接到突发事件。
- 创建状态定义。
- 创建 open、close 和 view 的显示应用程序屏幕定义。
- 为 open、close 和 view 创建 Add、Cancel、Fill 和 Find 的显示应用程序选项定义。
- 修改进程定义记录 im.set.close。
- 测试工作单示例。

创建表

使用数据库字典实用程序可以创建新表。在此示例中，将创建名为 EXWorkOrder 的表。在创建此表之前，需要知道所需的字段以及字段的属性。在此示例中，表存储与突发事件相关联的工作单的数据。

要创建表，请执行下列操作：

1. 在“系统导航栏”单击**定制** > **数据库字典**。
此时将打开“数据库字典”窗体。
2. 在“文件名”中键入 EXWorkOrder。
3. 单击**新建**。
4. 在“字段”选项卡上，单击**新的字段/键**后输入以下信息。

字段	描述
ID	将通过数字文件填充。 类型：字符
RelatedID	将按相关记录的唯一 ID 进行填写，示例：突发事件编号。 类型：字符
status	用于存储工作单的状态。 类型：字符
initiator	打开工作单的操作员。 类型：字符
assignee.name	分配到工作订单的操作员。 类型：字符
description	用于保存工作单要执行的工作的描述。 类型：字符数组
category	通过相关记录的类别填充。 类型：字符
RelatedCIs	用于与工作单相关联的配置项（CI）的列表。 类型：字符数组
impact	类型：字符
urgency	类型：字符
priority	类型：字符
closure.code	类型：字符
deadline	工作单打开后必须填写完成日期。 类型：日期/时间
est.finish	代理人预计的工作单完成时间。 类型：日期/时间
update.action	类型：字符数组
closure.comments	类型：字符数组

5. 在将字段添加到表之后需要将键添加到表。在对 **EXWorkOrder** 表完成此操作之前不能退出数据库字典实用程序。请参见[向表添加键字段](#)以获取详细信息。

向表中添加键字段

使用数据库字典实用程序可向新表添加键。在创建名为 EXWorkOrder 的表之后，需要向表中添加键。键可以执行索引化的搜索并确保数据一致性。

要向表中添加键，请执行以下操作：

1. 在“数据库词典”中单击**键**选项卡。
2. 选择新键字段的第一个可用条目。
3. 单击**新的字段/键**。
4. 为您创建或编辑的每个键输入以下信息。

字段	描述
ID	类型：唯一
RelatedID	类型：非 Null 值
RelatedCIs	类型：允许 Null 值和重复值
assignee.name	类型：允许 Null 值和重复值

5. 单击**确定**。

创建窗体

转到“窗体设计器”，为 EXWorkOrder 表创建窗体。利用向导创建窗体，然后选择单个记录的详细信息并继续。此操作可以创建包含所有字段的基本窗体，并可根据需要修改此窗体。以下是 EXWorkOrder 窗体的示例：

要创建窗体，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 窗体设计器**。
此时将打开“窗体设计器”窗体。
2. 在“窗体名称”中键入 EXWorkOrder。
3. 单击**新建**。
4. 单击**是**以使用“窗体向导”。
5. 键入 EXWorkOrder 作为要为其创建窗体的表的名称。
6. 选择**单个记录的详细信息**作为要创建的窗体的类型。
7. 单击**确定**。
8. 单击**继续**以接受要包含在窗体上的字段的默认值。
9. 使用“窗体设计器”工具修改窗体布局。以下是 **EXWorkOrder** 窗体的示例。

10. “状态” 的值包括：
 - New (default status) (新建(默认状态))
 - Open (打开)
 - Ready (就绪)
 - Closed (已关闭)
11. “关闭代码” 的值包括：
 - Implemented (已实现)
 - Canceled (已取消)
 - Rollback (回滚)

以下是 EXWorkOrder 窗体示例。

The screenshot shows a web browser window titled "WorkOrder". The browser's address bar and toolbar are visible at the top. The page content includes a header with the HP logo and a navigation bar with buttons for "确定" (OK), "取消" (Cancel), "上一个" (Previous), "下一个" (Next), and "删除" (Delete). The main form area is titled "WorkOrder" and contains several input fields and a dropdown menu. The fields are labeled "ID:", "相关ID:", "类别:", "状态:", "启动人:", "代理人姓名:", "最后期限:", "预期完成时间:", "影响:", "紧急程度:", and "优先级:". The "启动人:" and "代理人姓名:" fields have search icons. The "最后期限:" and "预期完成时间:" fields are dropdown menus. The "影响:", "紧急程度:", and "优先级:" fields are also dropdown menus. A large text area for "描述:" is located at the bottom of the form. The form is organized into tabs: "常规" (General), "资产信息" (Asset Information), and "更新/关闭" (Update/Close). The "常规" tab is currently selected.

创建窗体的副本

在“窗体设计器”中，可通过创建 `sc.manage.problem` 的副本来创建 `sc.manage.WorkOrder` 窗体。修改表的列输入可使用 EXWorkOrder 表中的字段。

要通过窗体副本创建新窗体，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **窗体设计器**。
此时将打开“窗体设计器”窗体。
2. 在窗体中键入 `sc.manage.problem`。

3. 单击**搜索**。
4. 选择 `sc.manage.problem.g`。
5. 在详细信息“选项”菜单中单击**复制/重命名**。
6. 在“新名称”中键入 `sc.manage.WorkOrder`。
7. 单击**确定**。
8. 在“窗体设计器”中，用 `EXWorkOrder` 表更新列的输入字段。
9.
 - 事件 ID - ID
 - 类别 - category
 - 相关 ID - RelatedID
 - 状态 - status
 - 代理人 - assignee.name
 - 描述 - description,1
 - 优先级 - priority
 - 影响 - impact
 - 紧急程度 - urgency
10. 单击**确定**。

创建工作单窗体的链接

关系型数据库的优势之一是能够清除冗余信息。实现方法是有关某个主题的信息存储在一个位置或表格中，而这里又包含指向其他主题的链接。链接是数据和链接定义，与包含链接信息之间关系的一系列条件的组合。

要创建工作单窗体的链接，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 定制工具 > 链接**。
此时将打开“链接文件”窗体。
2. 在“名称”中键入 `EXWorkOrder`。
3. 在“描述”中添加描述。
4. 单击**新建**。
5. 输入以下信息。

源字段名称	目标文件名	目标字段名
initiator	operator	name
assignee.name	operator	name
RelatedCIs	device	logical.name

6. 选择 `initiator` 行（突出显示），然后右键单击 `initiator` 并选择**选择行**。

- 在 link.structure.g 窗体中输入启动人的以下信息。然后，对 `assignee.name` 和 `RelatedCIs` 重复步骤 6，并分别使用这些字段的以下信息。

源字段(填充到/发布自)	目标字段名称(填充到/发布自)
initiator	name
assignee.name	name
RelatedCIs	logical.name

- 单击**保存**。
- 单击**确定**。

创建序号文件

创建序号文件以为 EXWorkOrder 表中的记录生成序号。

要创建序号文件，请执行下列操作：

- 在“系统导航栏”中，单击**定制 > 定制工具 > 序列号**。
此时将打开“序列号”窗体。
- 在类中键入 `EXWorkOrder`。
- 在“最后一个编号”字段中键入 1。
- 在“描述”中键入工作单的编号。
- 在“长度”字段中键入 5。
- 在“前缀”中键入 `wo`。
- 单击**添加**。

创建对象定义

此对象用于定义 EXWorkOrder 对象的特征和行为，确定哪些数据需要包括在工作单记录中以及系统如何处理工作单。

备注：请参见[“对象定义”窗体和字段\(第 11 页\)](#)，以了解关于“对象定义”的字段描述。

要创建对象定义，请执行下列操作：

- 在“系统导航栏”中，单击**定制 > 文档引擎 > 对象**。
此时将打开“对象定义”窗体。
- 在“文件名”中键入 `EXWorkOrder`。
- 单击**添加**创建对象记录。

4. 在“对象信息”选项卡上，输入以下信息：

字段	值
唯一键	自动填充 (ID)
公用名称	自动填充 (EXWorkOrder)
描述字段	保存能够关联任何模块的工作单的表
配置文件应用程序	db.environment
配置文件变量	\$L.env
编号记录名称	EXWorkOrder
类别表名称	category
主窗体控制	EXWorkOrder
状态字段	status
分配目标字段	assignee.name
打开状态	EXWorkOrder.open
关闭状态	EXWorkOrder.close
默认状态	EXWorkOrder.view
搜索状态	EXWorkOrder.search

5. 单击**保存**。
 6. 选择**管理队列**选项卡。
 7. 在“管理队列”选项卡上，输入以下信息。

字段	值
管理条件	true
管理显示窗体	sc.manage.WorkOrder
管理默认查询	assignee.name=operator()
允许添加条件	false
默认查询说明	My WorkOrders

8. 单击**保存**。
 9. 单击**确定**。

创建初始化进程定义

工作单示例的这些进程定义指定当用户打开工作单记录时使用的初始表达式和 RAD 应用程序。

要创建初始化进程定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **文档引擎** > **进程**。
此时将打开“进程定义”窗体。
2. 在“进程名称”中键入 `EXWorkOrder.open.initial`。
3. 单击**添加**。
4. 在“初始表达式”选项卡中键入 `$L.format="EXWorkOrder"`。
5. 单击**保存**。
6. 在 RAD 选项卡中输入以下信息。

字段	值
在 RAD 调用之前计算的表达式	<code>\$L.number.record="EXWorkOrder";\$L.number.type="string"</code>
RAD 应用程序	<code>getnumb</code>
参数名称	参数值
<code>name</code>	<code>\$L.number.record</code>
<code>index</code>	<code>ID in \$L.file</code>
<code>text</code>	<code>\$L.number.type</code>

7. 单击**保存**。

创建显示应用程序屏幕定义

可以为工作单窗体的 `open`、`close` 和 `view` 屏幕创建显示应用程序屏幕定义，以允许用户打开、关闭和查看工作单记录。

要创建 `open`、`close` 和 `view` 的屏幕应用程序定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **定制工具** > **显示屏幕**。
此时将打开“显示应用程序屏幕定义”窗体。
2. 在“屏幕 ID”中键入 `EXWorkOrder.open`。
3. 输入以下信息：

字段	值
屏幕 ID	<code>EXWorkOrder.open</code>
标题	打开新工作单
窗体	<code>\$L.format</code>
I/O（如果代替 RIO 面板）	<code>true</code>

按选项 0 时的操作:	刷新屏幕
语言	ENG

- 单击**添加**，然后单击**确定**。

对于此示例，还需要为 close 屏幕创建显示应用程序屏幕定义。

- 在“系统导航栏”中，单击**定制** > **定制工具** > **显示屏幕**。
此时将打开“显示应用程序屏幕定义”窗体。
- 使用以下信息：

字段	值
屏幕 ID	EXWorkOrder.close
标题	关闭工作单
窗体	\$L.format
I/O (如果代替 RIO 面板)	true
按选项 0 时的操作:	redraw screen
语言	ENG

- 单击**添加**，然后单击**确定**。

对于此示例，还需要创建 view 屏幕的显示应用程序屏幕定义。

- 在“系统导航栏”中，单击**定制** > **定制工具** > **显示屏幕**。
此时将打开“显示应用程序屏幕定义”窗体。
- 使用以下信息：

字段	值
屏幕 ID	EXWorkOrder.view
标题	查看新工作单
窗体	\$L.format
I/O (如果是 RIO 面板)	true
按选项 0 时的操作:	redraw screen
语言	ENG

- 单击**添加**，然后单击**确定**。

创建显示应用程序选项定义

此操作步骤逐步介绍了如何为下列工作单定义创建 add、cancel、fill 和 find 显示应用程序选项定义：

- open
- close
- view

在此操作步骤中，将对每个工作单屏幕定义(open、close、view) 重复 4 次相同的步骤。但是，每次在“显示应用程序选项定义”窗体的字段中输入的值不同。步骤下面的表提供了配置显示应用程序选项所需的值。

有关显示应用程序选项的其他信息，请参阅 HP Service Manager 联机帮助服务器上关于显示应用程序的联机帮助主题。

在为 open、close 和 view 创建显示应用程序屏幕定义后，需要为 open、close 和 view 屏幕创建显示应用程序选项。

要创建显示应用程序选项定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **定制工具** > **显示选项**。
此时将打开“显示应用程序屏幕定义”窗体。
2. 在“屏幕 ID”中键入 EXWorkOrder.open。
3. 输入以下信息。

字段	值
唯一 ID	自动生成 (EXWorkOrder.open_add)
操作	add
GUI 选项	4
文本选项	4
库	1
条件	true
默认标签	Add

4. 单击**添加**。
5. 单击**确定**。
6. 使用表中提供的值对以下显示选项定义重复步骤 1 - 6。

备注：在某些情况下，“操作”字段的下拉列表中没有提供所需的值，这时您可以将适用的值键入字段中。

Open - Cancel

字段	值
屏幕 ID	EXWorkOrder.open
唯一 ID	由系统生成。(EXWorkOrder.open_cancel)
操作	back
GUI 选项	3
文本选项	3
库	1
条件	true
默认标签	Cancel

Open - Fill

字段	值
屏幕 ID	EXWorkOrder.open
唯一 ID	由系统生成。(EXWorkOrder.open_fill)
操作	fill
GUI 选项	9
文本选项	9
库	1
条件	true
默认标签	Fill

Open - Find

字段	值
屏幕 ID	EXWorkOrder.open
唯一 ID	由系统生成。(EXWorkOrder.open_find)
操作	find

GUI 选项	8
文本选项	8
库	1
条件	true
默认标签	Find

Close - Cancel

字段	值
屏幕 ID	EXWorkOrder.close
唯一 ID	由系统生成。(EXWorkOrder.close_cancel)
操作	back
GUI 选项	3
文本选项	3
库	1
条件	true
默认标签	Cancel

Close - Close

字段	值
屏幕 ID	EXWorkOrder.close
唯一 ID	由系统生成。(EXWorkOrder.close_close)
操作	close
GUI 选项	5
文本选项	5
库	1
条件	true
默认标签	Close

Close - Fill

字段	值
屏幕 ID	EXWorkOrder.close
唯一 ID	由系统生成。(EXWorkOrder.close_fill)
操作	fill
GUI 选项	9
文本选项	9
库	1
条件	true
默认标签	Fill

Close - Find

字段	值
屏幕 ID	EXWorkOrder.close
唯一 ID	由系统生成。(EXWorkOrder.close_find)
操作	find
GUI 选项	8
文本选项	8
库	1
条件	true
默认标签	Find

View - Cancel

字段	值
屏幕 ID	EXWorkOrder.view
唯一 ID	由系统生成。(EXWorkOrder.view_cancel)
操作	back

GUI 选项	3
文本选项	3
库	1
条件	true
默认标签	取消

View - Fill

字段	值
屏幕 ID	EXWorkOrder.view
唯一 ID	由系统生成。(EXWorkOrder.view_fill)
操作	fill
GUI 选项	9
文本选项	9
库	1
条件	true
默认标签	填充

View - Find

字段	值
屏幕 ID	EXWorkOrder.view
唯一 ID	由系统生成。(EXWorkOrder.view_find)
操作	find
GUI 选项	8
文本选项	8
库	1
条件	true
默认标签	查找

View - Save

字段	值
屏幕 ID	EXWorkOrder.view
唯一 ID	由系统生成。(EXWorkOrder.view_save)
操作	save
GUI 选项	4
文本选项	4
库	1
条件	true
默认标签	Save

创建状态定义

工作单示例中的这些状态定义指定要使用的进程以及当用户打开、关闭或查看工作单记录时所允许的操作。

备注：请参见 [“状态定义” 字段描述\(第 23 页\)](#)，以了解“状态定义”的字段描述。

要创建 open 的状态定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 状态**。
此时将打开“进程定义”窗体。
2. 输入以下信息：

字段	值
状态	EXWorkOrder.open
显示屏幕	EXWorkOrder.open
初始化进程	EXWorkOrder.open.initial
窗体	\$L.format
输入条件	true

3. 单击**添加**。

要创建 close 的状态定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 状态**。
此时将打开“状态定义”窗体。

2. 输入以下信息：

字段	值
状态	EXWorkOrder.close
显示屏幕	EXWorkOrder.close
初始化进程	EXWorkOrder.close.initial
窗体	\$L.format
输入条件	true

3. 单击**添加**。

要创建 view 的状态定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 状态**。
此时将打开“状态定义”窗体。
2. 输入以下信息：

字段	值
状态	EXWorkOrder.view
显示屏幕	EXWorkOrder.view
初始化进程	EXWorkOrder.view.initial
窗体	\$L.format
输入条件	true

3. 单击**添加**。

要创建 search 的状态定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 状态**。
此时将打开“状态定义”窗体。
2. 输入以下信息：

字段	值
状态	EXWorkOrder.search
显示屏幕	db.search

字段	值
初始化进程	EXWorkOrder.view.search
窗体	\$L.format
输入条件	true

3. 单击**确定**。

添加关闭工作单按钮

当用户完成一个或多个任务后，应当关闭工作单。此操作步骤介绍如何在 EXWorkOrder 窗体上放置关闭按钮并将工作单的状态更新为关闭。此操作步骤会添加显示应用程序选项定义、EXWorkOrder.view 的状态定义以及 EXWorkOrder.close 的进程定义。

备注：此操作步骤不会说明如何包括此活动的验证。您可以使用“窗体控制”来添加验证。

要创建 EXWorkOrder.view 的显示应用程序选项定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 定制工具 > 显示选项**。
此时将打开“显示应用程序屏幕定义”窗体。
2. 在“屏幕 ID”中键入 EXWorkOrder.view。
3. 输入以下信息。

字段	值
唯一 ID	自动生成 (EXWorkOrder.view_close)
操作	close
GUI 选项	5
文本选项	5
库	1
条件	true
默认标签	Close

4. 单击**添加**。
5. 单击**确定**。

要更新 EXWorkOrder.view 状态定义，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 状态**。
此时将打开“状态定义”窗体。
2. 在“状态”字段中，键入 **EXWorkOrder.view**，然后单击**搜索**。此时将打开 EXWorkOrder.view 状态定义窗体。
3. 添加以下信息以更新状态定义窗体。

字段	值
显示操作	close
进程名称	EXWorkOrder.close
条件	true

4. 单击**保存**，然后单击**确定**。

要添加 EXWorkOrder.close 的进程定义记录，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 进程**。
此时将打开“进程定义”窗体。
2. 在“进程名称”中键入 EXWorkOrder.close。
3. 单击**添加**。
4. 在“初始表达式”选项卡上，键入以下表达式：
 - status in \$L.file="Closed"
 - \$L.mode="closed"
5. 单击**保存**。
6. 在 RAD 选项卡上，输入以下信息：

字段	值
在 RAD 调用之前计算的表达式	\$L.EXaction="update"
RAD 应用程序	se.base.method
条件	true
参数名称	参数值
文件	\$L.file
提示	\$L.EXaction
第二个文件	\$L.file.save
记录	\$L.fc
second.record	\$L.object
boolean1	false

7. 单击**保存**，然后单击**确定**。

创建工作单向导

此示例将使用向导创建工作单，因为向导可为用户简化打开工作单的任务，并且系统中已提供该功能。

此示例将使用向导创建工具创建一个向导，允许用户通过“突发事件管理”模块创建工作单。当用户从突发事件创建工作单时，向导会提示用户输入相关信息，并自动填写工作单中的某些字段。

要为工作单窗体创建向导，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **向导**。
此时将打开“向导信息”窗体。
2. 在“向导名称”中键入创建工作单 - 1。
3. 单击**添加**创建向导记录。
4. 在“向导信息”选项卡上，输入以下信息：

字段	值
简要描述	创建新工作单。
窗口标题	创建工作单
标题	创建工作单
启动节点	选中此项(设置为 True) 表示在有一系列向导记录时，这是其中的第一个向导。

5. 单击**保存**。
6. 在“向导信息”窗体的“文件选择”选项卡中，选择 **Select \$L. file by** 选项卡。
7. 在“Select \$L. file by”选项卡上，输入以下信息：

字段	值
创建记录	true
类型	EXWorkOrder

8. 单击**保存**。
9. 在“行为设置”选项卡上，输入以下信息：

字段	值
向导行为	在“向导行为”部分中单击 请求用户输入 。
要显示的子窗体	键入 createW0. assigneeAndCIs
显示屏幕	键入 wizard. display
激活“完成”选项	选中此项(设置为 True) 可在向导窗体上添加“完成”按钮。

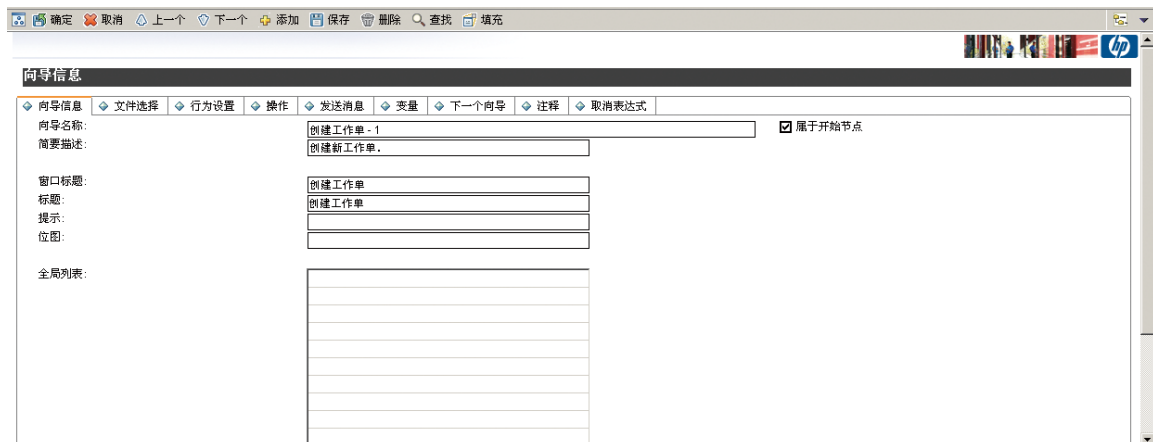
10. 单击**保存**。
11. 在“操作”选项卡上，输入以下信息：

字段	值
执行操作的对象	选择当前文件 (\$L.file)。
要执行的操作 > 表达式	<pre> initiator in \$L.file=operator() status in \$L.file="New" RelatedID in \$L.file=number in \$relatedRec category in \$L.file=category in \$relatedRec impact in \$L.file=initial.impact in \$relatedRec urgency in \$Lfile=severity in \$relatedRec priority in \$L.file=priority.code in \$relatedRec </pre>
完成时是否显示记录?	单击 完成时是否显示记录?
模式	选择“添加”。

- 单击**保存**。
- 在“取消表达式”选项卡上，输入以下信息：

字段	值
对“取消”执行的表达式	cleanup(\$relatedRec)

- 单击**保存**，然后单击**确定**。



添加进程定义记录

进程定义记录将定义系统响应用户操作的方式。进程定义使用 RAD 表达式、JavaScript，并调用现有 RAD 应用程序对当前记录（本例中为工作单记录）执行操作。

要添加进程定义记录，请执行下列操作：

1. 在“系统导航栏”中，单击**定制 > 文档引擎 > 进程**。
此时将打开“进程定义”窗体。
2. 在“进程名称”中键入 `create.WorkOrder`。
3. 单击**添加**。
4. 在“初始表达式”选项卡上，键入以下表达式：
 - `$L.void=fduplicate($relatedRec, $L.file)`
 - `$relatedCIs={}`
5. 单击**保存**。
6. 在“初始 Javascript”选项卡上，键入以下表达式：

```
system.vars.$relatedCIs=system.library.BSGFunctions.getMembers(system.vars.$L_
file.affected_item, false, 3)
```

备注：必须在一行上输入该表达式。

7. 单击**保存**。
8. 在 RAD 选项卡上，输入以下信息：

字段	值
在 RAD 调用之前计算的表达式	键入以下两个表达式： <code>\$L.wiz.name="Create Workorder-1"</code> <code>if (not null(logical.name in \$L.file)) then (\$relatedCIs=insert (\$RelatedCIs, 1, 1, logical.name in \$L.file))</code> 备注： 确保在一行上输入该表达式。还需注意，输入该表达式时，单词 'insert' 后没有空格。例如，上面的表达式紧接单词 'insert'，如下： <code>insert(\$RelatedCIs, 1, 1, logical.name in \$L.file))</code>
RAD 应用程序	wizard.run
参数名称	name

参数值	\$L.wiz.name
参数名称	text
参数值	\$L.exit
条件	true

9. 单击**保存**。
10. 在“最终表达式”选项卡上，键入 `cleanup($relatedRec)`。
11. 单击**保存**，然后单击**确定**。

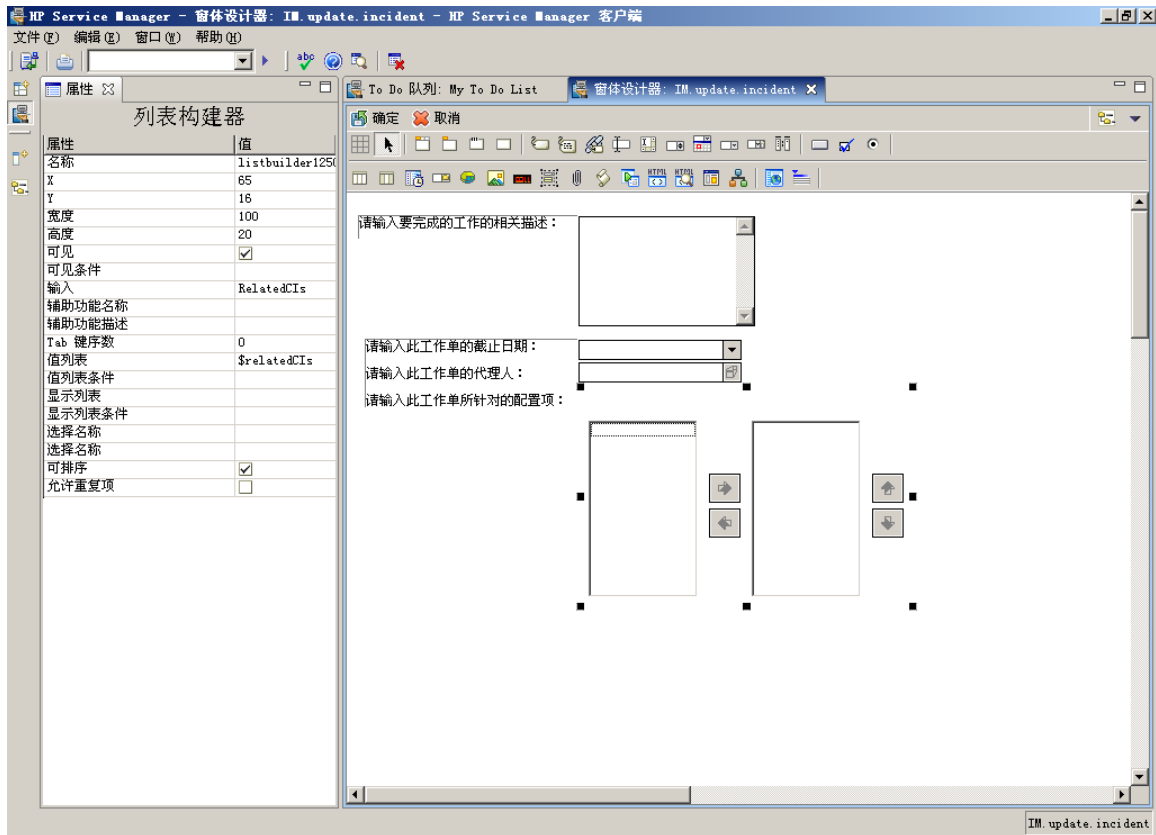
为向导创建输入窗体

此窗体是在工作单向导中显示的第一个窗体。用户输入其需要的信息，以供向导创建工作单记录。使用“窗体设计器”创建此窗体。对于此示例，窗体名称为 `createW0.assigneeAndCIs`。

要为向导创建输入窗体，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **窗体设计器**。
此时将打开“窗体设计器”的“搜索/新建”窗体。
2. 在窗体的窗体名称处键入 `createW0.assigneeAndCIs`。
3. 单击**新建**。不需要对此窗体使用“窗体设计器”向导。
4. 在窗体上创建以下输入字段：
 - 描述(请输入要完成的工作的相关描述)
 - 截止日期(请输入此工作单的截止日期)
 - 代理人(请输入此工作单的代理人)
 - 配置项(请输入此工作单所针对的配置项)
5. 此窗体的“属性”应包括：
 - 输入: **RelatedCIs**
 - 值列表: **\$relatedCIs**
 - 可排序: 选中
6. 单击**保存**。

下图显示了 EXWorkorder 向导输入窗体的 `createW0.assigneeAndCIs` 窗体示例。



修改 close 和 update 突发事件窗体

在此工作单示例中，您需要修改“关闭突发事件”以及“更新突发事件”窗体，以便查看分配到突发事件的工作单，并且可通过双击来读取或编辑突发事件中的工作单。使用“窗体设计器”可以更新 *IM.update.incident* 和 *IM.close.incident* 窗体。要执行此操作，必须在 *IM.update.incident* 和 *IM.close.incident* 窗体上添加一个选项卡，然后在选项卡上添加一个窗体。

要修改 update 突发事件窗体，请执行下列操作：

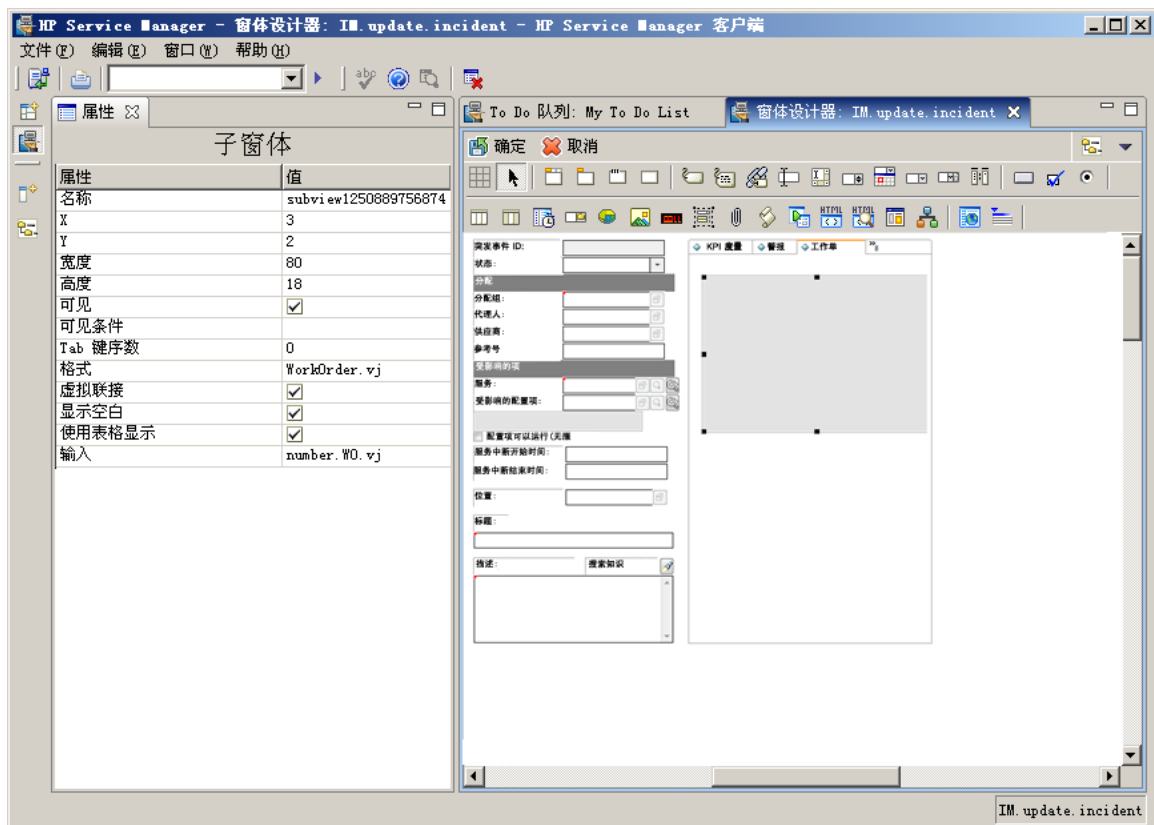
备注：请使用相同的常规步骤来更新 close 突发事件窗体。

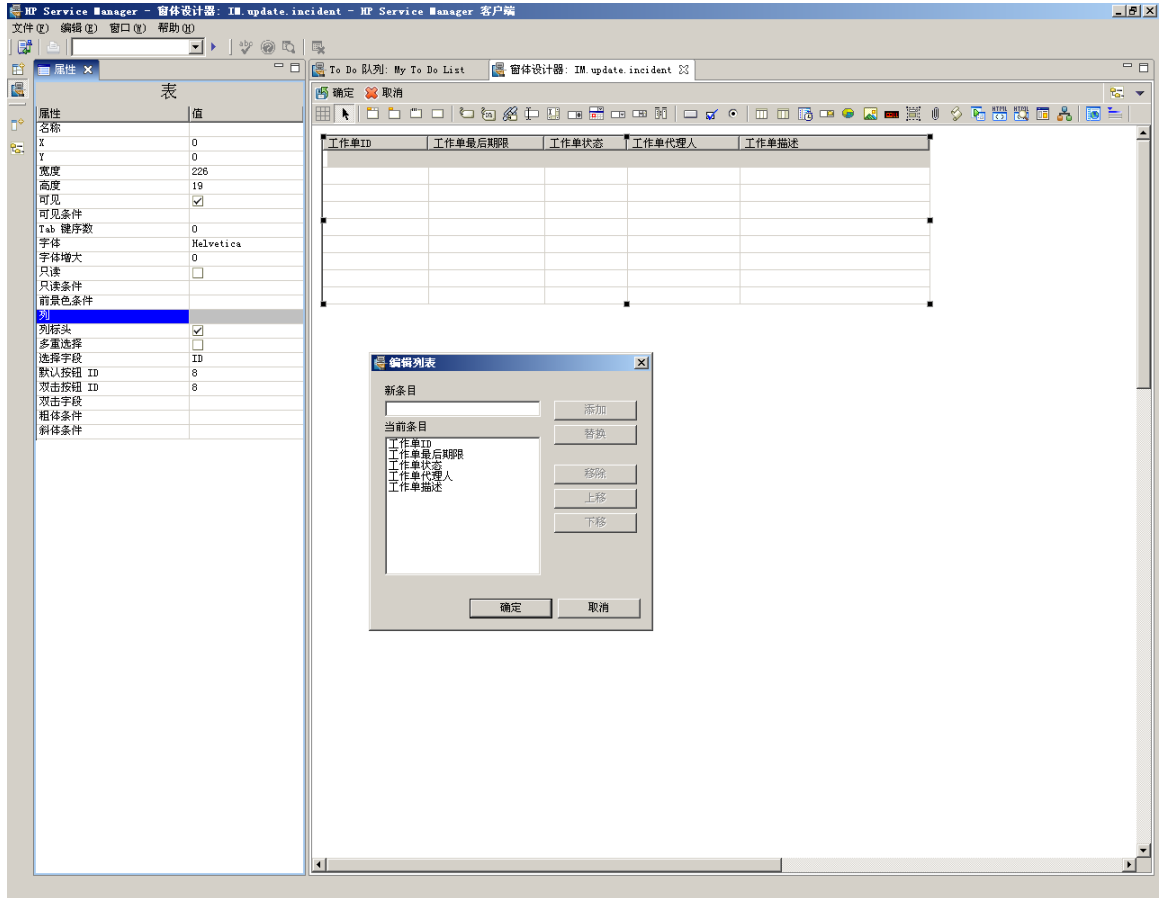
1. 在“系统导航栏”中，单击**定制 > 窗体设计器**。
此时将打开“窗体设计器”的搜索窗体。
2. 在窗体中的窗体名称处键入 **IM.update.incident**。
3. 单击**搜索**。
4. 单击**设计**。
5. 将新选项卡添加到现有笔记本。
6. 将选项卡的标题设置为“工作单”。

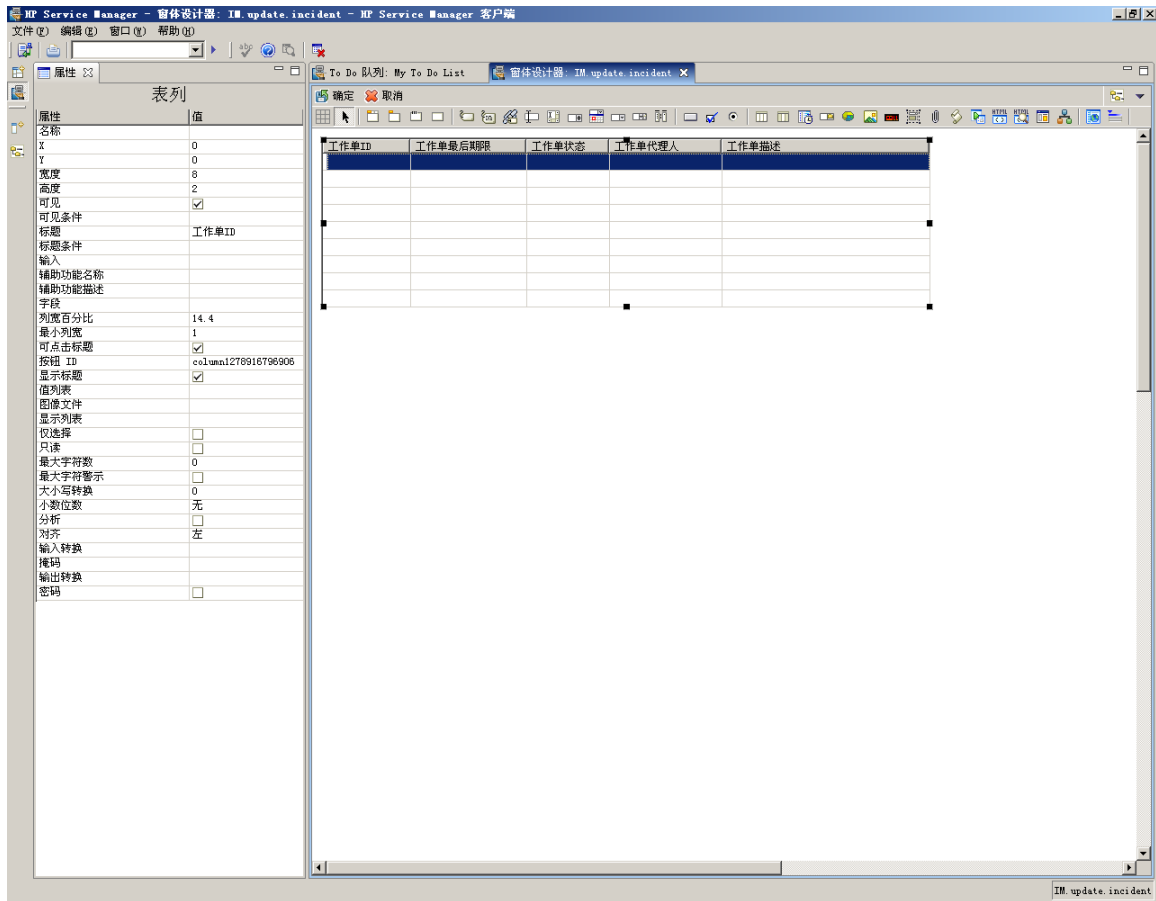
7. 添加具有以下属性的子窗体：
 - 可见：选中
 - 窗体：WorkOrder.vj
 - 虚拟联接：选中
 - 显示空白：选中
 - 显示使用表：选中
 - 输入：number.WO.vj
8. 单击**保存**。
9. 使用“窗体设计器”工具创建 WorkOrder.vj 窗体：
 - 工作单 ID：ID
 - 工作单最后期限：deadline
 - 工作单状态：status
 - 工作单代理人：assign.name
 - 工作单描述：description
10. 对 **IM.close.incident** 重复步骤 1 - 8。

备注：您可以使用在虚拟联接的步骤 9 中创建的 WorkOrder.vj 窗体。

下图显示了要添加的选项卡的示例。







在 Probsummary 表中创建链接的别名

在工作单示例中，如果突发事件有与其关联的工作单，则用户需要能够从突发事件记录访问工作单。要实现此目的，必须在 Probsummary 表中创建别名，然后使用别名在 Probsummary 表和 EXWorkOrder 表之间创建链接。

要向 Probsummary 表中添加别名，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **数据库字典**。
此时将打开“数据库字典”窗体。
2. 在“文件名”中键入 Probsummary。
3. 单击**搜索**。
4. 在“字段”选项卡上，选择数值字段，然后单击**编辑字段/键**。
5. 单击**创建别名**，然后在名称处键入 number.WO .vj，在类型处键入字符。
6. 单击**确定**。

要链接 EXWorkOrder 表和 probsummary 表，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **定制工具** > **链接**。
此时将打开“链接文件”窗体。
2. 在“名称”中键入 Probsummary。

3. 单击**搜索**。
4. 在最后一个条目之后单击，为新条目创建空白行。
5. 在“来源”字段中，键入 `number.WO.vj`。
6. 选择整个新行，然后单击**选择行**。
7. 输入以下信息：

字段	值
字段(出处/源)	<code>number.WO.vj</code>
文件(到/目标)	<code>EXWorkOrder</code>
字段(到/目标)	<code>RelatedID</code>
查询	<code>\$query</code>
表达式	<code>\$query="RelatedID=\""+number in \$File+"\\""</code>

8. 单击**保存**，然后单击**返回**。
9. 重复这些步骤，为 ID 字段创建链接行。
10. 输入以下信息。

字段	值
字段(出处/源)	<code>ID</code>
文件(到/目标)	<code>EXWorkOrder</code>
字段(到/目标)	<code>ID</code>
查询	<code>\$query</code>
表达式	<code>\$query="ID=\""+nullsub(cursor.field.contents(), "xxx")+"\\""</code>

11. 单击**保存**，然后单击**返回**。

修改 `im.set.close` 进程定义

要防止用户在有突发事件的工作单处于打开状态时关闭突发事件，需要对此进程进行修改。

要修改进程定义记录，请执行下列操作：

1. 在“系统导航栏”中，单击**定制** > **文档引擎** > **进程**。
此时将打开“进程定义”窗体。
2. 在“进程名称”中键入 `im.set.close`。
3. 单击**搜索**。

4. 在“初始 Javascript”选项卡上，键入以下 JavaScript：


```
var WO=new SCFile ("EXWorkOrder")

var FoundOpenWO=WO.doSelect ("RelatedID=\""+system.vars.$L_
file.number + "\""+ " and status ~=\"\" + "Closed" + "\"")

if (FoundOpenWO == RC_SUCCESS)
{
system.vars.$openWO=true;
}
else
{
system .vars.$openWO=false;
}
}
```
5. 单击**保存**。
6. 在 RAD 选项卡上，输入以下信息：

字段	值
备注：RAD 选项卡上的某些字段是预填写的，不需要对这些值进行修改	
RAD 应用程序 - 选择调用 RAD 应用程序 us.consume.wrapper 部分下的空白部分，然后输入以下信息：	
在 RAD 调用之前计算的表达式	\$L.text="There are still open work orders.This incident cannot be closed yet."
RAD 应用程序	apm.mb.ok
条件	\$openWO=true
参数名称	text
参数值	\$L.text

7. 单击**保存**。
8. 在“最终表达式”选项卡上，输入以下信息：
 - if (\$openWO=true) then (\$L.exit="badval")
 - \$L.exit="closestate"
9. 单击**保存**，然后单击**确定**。

测试工作单示例

完成创建工作单系统的任务之后，需要验证其是否可以正常工作。要验证工作单示例的基本功能，请执行下列操作：

- 在突发事件管理器中，查找或创建 open 突发事件。
- 使用工具栏上的“选项”菜单或按钮创建工作单。“选项”菜单上应显示“创建工作单”。
- 在工作单向导中，输入数据以为突发事件创建工作单。
- 将工作单添加到突发事件，并保存对突发事件的更改。
- 打开待编辑的突发事件，并从“工作单”选项卡编辑工作单。
- 保存所做的更改。
- 再次打开突发事件，对其进行编辑，然后关闭。
- 现在应可以关闭突发事件。
- 重复上述步骤，但此次为突发事件创建两个工作单。
- 仅关闭其中一个工作单，然后尝试关闭突发事件。系统应会生成一则错误消息，指示突发事件仍有工作单处于打开状态，因此无法将其关闭。
- 关闭突发事件的所有工作单。现在应可以关闭突发事件。

