

FRAMEWORK SW

Framework SW

Conceitos

Versão 1.3

Framework SW - Overview

Framework SW - Overview

Sumário

1. INTRODUÇÃO	4
2. CONCEITOS GERAIS	5
3. VARIÁVEIS DE AMBIENTE	12
4. AMBIENTE DE EXECUÇÃO.....	19
5. APLICAÇÕES ONLINE	22
6. APLICAÇÕES BATCH.....	29
7. ACESSOS EXTERNOS	33
8. CLASSES SW	36
9. SWADM	40
10. OUTROS RECURSOS.....	44

Framework SW - Overview

1. INTRODUÇÃO

O framework SW foi construído de forma a possibilitar a execução de sistemas originados de ambiente mainframe Unisys na plataforma baixa sem alteração de especificação e codificação original. Devido a essa característica, o framework possui uma série de recursos e funcionalidades que se assemelham ao funcionamento do mainframe em seus vários níveis, utilizando vários conceitos existentes nessa plataforma.

Os sistemas desenvolvidos em ambiente mainframe Unisys têm como característica o uso de programas Batch e/ou Online escritos em linguagem Cobol ou em linguagem 4GL, que acabam por gerar fontes finais em Cobol, acessando banco de dados hierárquico DMSII.

Para suportar os fluxos de execução Batch, existem scripts em linguagem WFL, que possibilitam a construção de lógicas para sequenciamento de execução dos programas, bem como validação de seus requisitos de execução e operações de pós-processamento desses programas.

Os objetos de negócio foram transformados em classes java, com uso de banco relacional para o acesso às informações armazenadas em banco de dados. Os fluxos de execução são disponibilizados através de shell script, onde os vários processos batch e seus controles são processados.

Existe uma ferramenta online de configuração e monitoração do ambiente, chamado SWADM, bem como daemons para executar funções específicas relacionadas às características do framework.

Nesse documento serão apresentados os principais conceitos existentes no framework, bem como sua relação com o ambiente do qual esses conceitos derivam.

Framework SW - Overview

2. CONCEITOS GERAIS

Alguns conceitos importantes que são usados no framework SW serão descritos a seguir. Eles permitem que as execuções dos objetos na plataforma baixa possuam a mesma característica existente na plataforma alta.

A utilização desses conceitos na forma de informações configuradas é feita na aplicação SWADM. Essa aplicação utiliza interface WEB, é baseada em banco de dados Oracle, e armazena todas as configurações realizadas que permitem que as aplicações baseadas em framework SW possam ser executadas.

Além de ser um repositório para a configuração relacionada às aplicações, o SWADM também é utilizado para a operação e monitoração do ambiente.

USERNAME

É o código de acesso principal ao ambiente, sendo utilizado para autenticação, podendo ou não exigir senha. Todo USERNAME, também chamado USERCODE, definido possibilita a criação de arquivos físicos nos discos do sistema, que estarão contidos em um subdiretório formado pelo nome do username.

Na prática existem duas categorias de USERNAME:

- Username de conexão: Nesse caso, o Usercode pode ser utilizado apenas para permitir a entrada do usuário no ambiente e definir quais sistemas existentes podem ser utilizados em sua sessão
- Username de execução: Essa categoria está associada aos processos em execução e aos arquivos gerados nos vários discos do sistema. Esse Username é o dono desses processos e arquivos e segue a regra de privilégio para acesso aos arquivos de outros usuários.

ACCESSCODE

É o código de acesso secundário, sendo utilizado para identificar a pessoa dentro de um grupo pertencente a um mesmo USERNAME, tendo uma senha associada. Um Accesscode pode fazer parte do grupo de vários Usernames. Um Username pode possuir ou não Accesscodes atribuídos. Existe um atributo chamado ACCESSCODENEDED que indica a obrigatoriedade de autenticação usando a sequencia Username/senha(se for definida) e, posteriormente, Accesscode/senha.

Framework SW - Overview

FAMÍLIA

Família é a denominação de um conjunto de bytes disponíveis para armazenamento, que pode ser formado por espaço em vários discos físicos. Arquivos são armazenados em famílias e podem existir arquivos de diferentes Usernames e tipo em uma mesma família.

Todo o Username definido no ambiente possui o que é chamado de família primária, uma área de armazenamento default para os arquivos lidos ou gravados.

MIXLIMIT

O número máximo de processos batch que podem estar em execução no ambiente em um determinado momento é chamado MIXLIMIT. Quando esse número for atingido no ambiente, as novas solicitações de execução de processos batch ficarão em fila, aguardando a finalização de algum dos processos em execução para permitir um novo processo ser inicializado.

FILA

Existe a possibilidade de criar várias filas para serem associadas à execução de Jobs. Um job só pode ser executado se estiver definido em seu fonte qual a fila na qual o job será alocado. Além do MIXLIMIT total do ambiente, as filas também possuem MIXLIMIT, indicando quantos Jobs simultâneos podem ser executados a partir dela. A regra do MIXLIMIT global ainda é válida, isto é, mesmo que o MIXLIMIT da fila não tenha sido ultrapassado, um job pode não ser iniciado porquê o MIXLIMIT global foi atingido.

Quando se define MIXLIMIT=1 para uma fila, isso significa que somente um job por vez será executado. Isso é importante quando existem jobs com elevado consumo de recursos que dever ter execução controlada ou quando dois Jobs não podem estar em execução ao mesmo instante.

PIDs

Todo processo executado no ambiente, jobs ou classes de negócio(tasks), recebe da framework um número de PID único e sequencial. No caso de processo de job, esse número de PID recebe o nome de JPID(Job PID). Para classes de negócio(tasks), esse número tem o nome de PPID(Program PID).

Além desses dois PIDs, que são atribuídos pelo framework, temos ainda um terceiro, que é o PID que o processo recebeu do sistema operacional do servidor onde está sendo executado. Esse PID é chamado de SPID(Server PID).

Framework SW - Overview

LOCALIZAÇÃO ARQUIVOS ESPECIAIS

O armazenamento de alguns arquivos de tipos específicos é feito em famílias definidas explicitamente. Esses tipos de arquivo são:

- Arquivos de log de jobs
- Arquivos de log de programa
- Arquivos intermediários dos processos de sort
- Arquivos de relatórios gerados pelos programas

ARQUIVO DE DADOS

Os arquivos existentes no ambiente podem ser armazenados nas várias famílias definidas. Eles possuem atributos que são utilizados para definição de ações que serão realizadas com eles durante a execução dos programas que os utilizam.

Os atributos são:

FILETITLE

Um arquivo sempre possui um nome e uma localização. Essa combinação determina o seu FILETITLE, que inclui o Usercode sob o qual o arquivo é armazenado. Pode ser especificado da forma:

`(([usercode])[nome] on [família])`

O resultado de um FILETITLE como esse é que o arquivo será procurado com a definição:

`$SWDIR_[família]/[usercode]/[nome]`

Como alternativa de acesso e armazenamento, existe a possibilidade de um arquivo estar sob nenhum Usercode. O FILETITLE utilizado para definição desse arquivo será:

`*[nome] on [família]`

O resultado de um FILETITLE como esse é que o arquivo será procurado com a definição:

`$SWDIR_[família]/ [nome]`

Framework SW - Overview

FILENAME

O nome de arquivo é chamado de FILENAME. Ele é a parte <nome> do FILETITLE.

LASTRECORD

Esse atributo representa o número de registros existentes no arquivo, menos 1.

MAXRECSIZE

Os arquivos de dados gerados pela execução das classes possuem tamanho de registro fixo. O atributo MAXRECSIZE é a quantidade de bytes existente no registro de um arquivo.

SECURITY TYPE

O privilégio de um arquivo está representado no atributo SECURITY TYPE. Os valores desse atributo podem ser:

- PUBLIC IN
- PUBLIC OUT
- PUBLIC IO
- PRIVATE IN
- PRIVATE OUT
- PRIVATE IO

OPTIONAL

O atributo de arquivo OPTIONAL define se um arquivo é opcional ou não no processamento executado por uma classe que o utiliza. O default é não opcional. Quando um arquivo tem o OPTIONAL ativado, os comandos de abertura, fechamento e leitura associado a esse arquivo não apresentaram problemas durante a execução caso o arquivo não exista fisicamente. Esse arquivo será tratado como se fosse um arquivo existente no diretório e com zero registro.

ARQUIVO DE RELATÓRIOS

Os arquivos de relatórios são sempre armazenados em uma única família, especificada através da variável de ambiente SWDIR_PRINT. Cada arquivo de impressão fechado durante o

Framework SW - Overview

processamento de uma classe de negócio grava um arquivo no diretório definido em SWDIR_PRINT e associa um número identificador, chamado request de impressão. Através desse número(id) é possível utilizar a ferramenta SWADM para executar alterações em atributos de impressão ou gerar novos requests a partir do original alterando atributos como impressora destino, cópias, etc. O banco SWADM possui um registro para cada request gerada, com os atributos relacionados.

Arquivos de relatório possuem grande parte dos atributos de um arquivo de dados, mas por ser um tipo especial de arquivo, apresenta alguns atributos distintos.

Esses atributos são:

DESTINATION

O valor movido para esse atributo, durante a execução de um processo, definirá a impressora na qual o relatório deve ser impresso automaticamente, após o fechamento do arquivo. O componente do framework responsável pelo gerenciamento de impressão será o responsável por identificar um novo relatório com a destinação para uma impressora específica e executará as ações necessárias.

MAXPRINTLINES

O valor contido nesse atributo indicará o número máximo de linhas que podem ser geradas para o arquivo. A tentativa de gravar uma quantidade de linhas superior ao limite especificado, provocará o encerramento anormal da execução da classe.

FORMID

O valor movido para esse atributo fará com que a execução do processo de impressão do arquivo gerado só ocorrerá quando um comando de operação for executado para a requisição gerada, indicando que a impressora está com a formatação requerida pelo relatório.

NUMCOPIES

O valor movido para esse atributo pela classe em execução define o número de cópias que serão geradas pela execução do processo de impressão. O valor default para esse atributo é 1.

TITLE

Framework SW - Overview

É um atributo que armazena uma string que fará parte do nome do arquivo de impressão gerado, caso o atributo USERBACKUPNAME não esteja ativado. Caso USERBACKUPNAME tenha sido setado, o conteúdo de TITLE será exatamente o nome físico com que o arquivo vai ser criado.

USERBACKUPNAME

É um atributo do tipo TRUE ou FALSE, sendo esse o valor default. Define como o nome físico do arquivo de relatório será gerado. Se FALSE, o nome físico será montado como mostrado a seguir. Se TRUE, o nome físico será formado pelo conteúdo do atributo TITLE.

BDNAME

O atributo BDNAME é utilizado para agrupar relatórios de mesmo módulo ou sistema. O conteúdo desse atributo é utilizado para formar a parte inicial do nome do arquivo de impressão como mostrado a seguir. O valor default desse atributo é "BD".

O FILETITLE de um arquivo de relatório possui um padrão específico. Se USERBACKUPNAME estiver ativo, o nome físico do arquivo será o conteúdo do atributo TITLE. Caso contrário, ele é formado por:

```
<bdname>_000<pid job>_000<pid task>_000<title>
```

EXECUÇÃO DE PROCESSOS

A execução de processos online, batch e jobs possui alguns conceitos. Quando um objeto de um desses três tipos está em execução, com um determinado PPID para classes online e batch, ou um JPID para job, situações durante o processamento podem ocorrer e são tratadas pelo framework. Em alguns casos, existe a possibilidade de troca de mensagens entre o software de administração e monitoração do ambiente(SWADM) e os objetos.

Algumas situações que podem ocorrer durante o processamento fazem com que os objetos tenham sua execução suspensa até que a situação seja resolvida.

A falta de espaço em disco durante a execução de uma classe faz com que a classe entre em status de WAIT, aguardando o problema ser resolvido. A mensagem gerada e status da execução pode ser visualizada pelo SWADM e a ação adequada pode ser tomada.

A ausência de determinado arquivo em uma família durante o processamento quando esse arquivo teria uma operação de abertura para leitura sendo executada pela classe de negócio também é uma situação de suspensão de execução. Nesse caso, a aplicação SWADM permite, se o arquivo está residente mas tem um nome diferente do que está sendo solicitado pela

Framework SW - Overview

aplicação, que uma mensagem seja enviada pelo operador à classe suspensa indicando que o arquivo que deve ser usado tem outro FILETITLE. É uma operação chamada FA, ou FILE ASSOCIATION.

Quando um script tem em sua lógica a execução de uma classe java que não é encontrada, esse script tem sua execução suspensa e é possível visualizar a mensagem correspondente na console do SWADM. Pode ser realizada uma operação para disponibilizar a classe no ambiente, bastando o operador enviar uma mensagem de OK para o script executar uma nova tentativa de execução.

Qualquer classe ou job pode, em qualquer instante, ter sua execução abortada por comando de operação, enviada pelo SWADM. Essa situação é chamada de DS, DESCONTINUED. O processo que recebeu mensagem de DS se encerra de forma anormal, e o status no SWADM e logs de execução fica como ABORT.

É possível ainda passar mensagens para classes de negócio que, se tiverem código particular incluído para tratar essas mensagens, pode alterar o fluxo de execução. Essa situação é possível através do HI, ou EXECPTION EVENT.

Além disso, é possível para classes Online e Batch ter alterado seu nível de debug durante a execução. Essa situação é possível com utilização de um HI com código específico, 89. O nível de debug desejado deve ser informado após o 89 do código, formando um texto de 3 números.

Outra situação de execução, baseada em mensagem enviada pelo SWADM, possibilita que um dump das variáveis de trabalho (SWFields) possa ser gerado no log de execução do programa. O nome da variável e seu conteúdo ficam disponíveis no log de execução para visualização. Essa situação é possível através do uso de outro HI específico, com código 87.

Estatísticas de leituras e gravações em arquivos de dados, arquivos indexados e tabelas de banco de dados podem ser solicitados para gravação no log de execução das classes com a posição acumulada no instante da solicitação através de outro HI especial, com código 88.

A identificação de algum HI enviado pelo SWADM à classe de negócio ocorre na execução do método PERFORM. Programas suspensos, portanto, não conseguem receber HI, pois o fluxo de execução da classe não passa por nenhum PERFORM para que o HI enviado possa ser recebido.

Framework SW - Overview

3. VARIÁVEIS DE AMBIENTE

Para que o ambiente de execução de jobs, classes online e batch possa funcionar corretamente, é necessário a definição de variáveis de ambiente para serem utilizadas como parâmetros de armazenamento de arquivos e de execução de processos.

Essas informações devem ficar residentes no arquivo `profile_scripts_definitions`, existente no diretório definido na variável de ambiente **SWDIR_LIB**. Esse arquivo é utilizado como script de login, portanto deve ser incluído no `.bash_profile` dos usuários definidos no Linux, e de execução dos processos do ambiente da framework.

A descrição dessas variáveis é mostrada a seguir.

SWMOUNT_POINT

Essa variável define o diretório base da área compartilhada a ser utilizada pelo sistema “mainframe” onde está configurada. Todos os servidores que compõem esse sistema devem ter acesso à esse diretório.

SWDIR_HOME

Essa variável define o diretório base de armazenamento de todas as classes e arquivos que compõem o ambiente de execução de um determinado sistema “mainframe”. Os vários subdiretórios existentes contém arquivos relacionados à framework e sistemas migrados.

SWDIR_FAMILY

Essa variável define a localização física de arquivos de dados para serem lidos ou gravados e que não tiverem referência específica a uma família. O objetivo dessa variável é ser a família primária do usuário.

SWDIR_<família>

Essa variável define a localização física de arquivos de dados para serem lidos ou gravados associados a uma família específica existente no mainframe. Cada família existente na plataforma alta que possuem arquivos para armazenamento na plataforma baixa deve ter sua

Framework SW - Overview

variável correspondente definida. As referências para essas famílias dentro de Jobs utilizarão essa variável.

SWDIR_PRINT/SWDIR_BACKUP

Essa variável define a localização física dos arquivos de backup criados durante a execução do programa migrado. O nome do backup gerado é semelhante ao mainframe. Caso seja executado através de script, o arquivo é gerado como:

BD/<mix do programa>/<número seqüencial do arquivo><intname>

Caso seja gerado através de execução originada de um shell script, o nome do arquivo gerado será:

BD/<mix do job>/<mix do programa>/<número seqüencial do arquivo><intname>

No caso de execução por shell, é possível ainda definir o atributo USERBACKUPNAME como true, fazendo com que o nome do arquivo de impressão seja montado dinamicamente pelo programa. Caso contrário, o <intname> será utilizado.

Caso a variável SWBDNAME tenha sido definida, a string "BD" do prefixo do backup será substituída pelos caracteres definidos nessa variável.

SWDIR_TEMP

Essa variável define a localização física dos arquivos de temporários gerados na execução dos processos Online e Batch. Caso essa variável não seja definida, os arquivos temporários serão gerados no diretório apontado pela variável SWDIR_FAMILY.

SWDIR_LOGPROGAM

Essa variável define a localização do arquivo de log gerado na execução do programa. Esse arquivo contém informações gravadas automaticamente pela framework, tais como início de execução, término de execução, abertura e fechamento de arquivos, bem como todos os DISPLAYs e exceções geradas durante o processamento do programa.

SWDIR_SETUP

A variável de ambiente SWDIR_SETUP indica a localização física de todos os arquivos de configuração das conexões de banco utilizadas, além do SETUP.properties. Existe um arquivo de configuração para cada banco de dados migrado.

Framework SW - Overview

SWDIR_SORT

A variável de ambiente SWDIR_SORT indica o diretório que será utilizado pelo processo de sort como área temporária de armazenamento.

SWDIR_SORT_PROGRAM

Essa variável indica diretório onde o utilitário de sort está localizado.

SWBDNAME

Essa variável define qual será o BDNAME que deve compor o nome dos arquivos de impressão criados. Caso essa variável não tenha sido definida, a string "BD" será utilizada como valor default.

SWDIR_SCRIPTS

Essa variável define o armazenamento dos fluxos de execução de podem ser executados por um determinado usuário do sistema. Cada usuário deve ter sua área de armazenamento distinta.

SWRUN_PREFIX

Essa variável define parte da package onde os programas batch estão definidos. É utilizada na execução dos jobs. Somente a última parte do nome da package não é incluído, já que para cada projeto ou sistema essa parte é distinta.

SWRUN_PREFIX_LIB

Essa variável define parte da package onde as libraries utilizada pelos programas batch estão definidos. É utilizada na execução dos jobs. Somente a última parte do nome da package não é incluído, já que para cada conjunto de libraries normalmente é definido um identificador único.

Framework SW - Overview

SWENV_DEBUG

Essa variável define o nível de debug a ser utilizado por qualquer programa que seja executado no ambiente. Essa situação pode ser alterada por um comando setSW_DEBUG executado dentro do programa ou pelo uso de -DDEBUG=<nível> no argumento da JVM na execução de um programa batch migrado. O valor associado deve ser um valor numérico, de 0(sem mensagens de debug) a 5(nível máximo de mensagens de debug)

SWENV_DEBUG_MANAGER

Essa variável define o nível de debug a ser utilizado pelo gerenciador de ambientes Online. OS valores possíveis são "N" ou "S".

SWFILE_SEPARATOR

Essa variável define o caracter separados dos nomes de arquivos de dados e backup. O valor default é "_". Para compatibilidade com outros sistemas da plataforma baixa, nos ambientes Linux essa variável tem o conteúdo ".".

SWDB_REBLOCK

Quando setada indica que os acessos seriais a tabelas de banco de dados identificados resultam em um fetch de um número maior de registros que o normal(reblock). Dessa forma, o tempo de execução dos acessos a todos os registros necessários será reduzido. OS valores possíveis são "N" ou "S".

SW_MAINFRAME

Essa variável define o IP do mainframe que será utilizado nas comunicações das aplicações migradas na execução de rotinas no mainframe. A porta a ser utilizada estará definida na variável SW_<rotina>_PORT correspondente. Cada rotina específica terá sua porta de comunicação mas utilizará o IP do mainframe definido nessa variável. O mainframe em questão deve ter o serviço em execução.

SW_<serviço>_PORT

Essa variável define a porta que será utilizada para execução de um serviço no mainframe. Esse serviço é um programa no mainframe que disponibiliza essa porta para conexão e contém uma ou mais rotinas disponíveis para execução. Como exemplo, SW_FONETICA_PORT define a porta que será utilizada pelo serviço FONETICA criado na plataforma baixa.

Framework SW - Overview

SWBACKUP_SERVER

Essa variável define o servidor que será usado nos comandos de cópia e restauração de arquivos salvos no ambiente de backup instalado. Deve ser definido obrigatoriamente nos casos onde softwares de backup precisam ser utilizados para complementar o ambiente de execução de Jobs.

SW_SAA

Essa variável permite que o SAA possa ser simulado no ambiente Eclipse. Com o valor “S” as execuções de autorização passam a ocorrer, utilizando os parâmetros de SETUP.properties USERCODE, ACCESSCODE e ID.

SW_URL_MESSAGE

Essa variável contém a URL que foi configurada para o sistema de mensagens, que é responsável pelo envio de mensagens assíncronas. Essas mensagens podem ser originadas de programas Online ou Batch e são enviadas ao browser do usuário associado à solicitação.

SWDISABLE_DEBUG

Essa variável define se o nível de debug de um programa pode ser ativado através do método setSW_DEBUG() por lógica de programação. Se for definido com o valor “S”, os debugs utilizando esse método não serão ativados em nenhum programa. Ficam valendo alterar o nível de debug através do comando HI 89n, onde n é o nível de debug desejado. O default desse parâmetro é FALSE.

SWTIME_LIMIT

Essa variável define o limite em segundos máximo que uma transação online pode ser executada. Ultrapassando esse tempo, o programa é abortado, gerando uma exceção de tempo de execução elevado. Essa situação pode evitar programas que ficam em loop por qualquer problema encontrado durante a sua execução.

SWPRINT_LOG

Essa variável define se as linhas de log geradas pelos programas Online serão armazenadas no arquivo catalina.out, além do arquivo de log da execução. Definindo essa variável com “S”, as informações são geradas tanto no arquivo de log, localizado em SWDIR_LOGPROGRAM, como

Framework SW - Overview

no catalina.out. Definindo essa variável com o valor "N", somente o log do programa conterá as informações geradas pelo programa. O valor default é "S".

SW_LOGFILE

Essa variável define como serão gerados os logs das libraries chamadas por algum programa. Ativando essa variável de ambiente para o valor "A" faz com que todos os logs de execução gerados pelo processamento de uma classe e de todas as libraries chamadas por ela fique registrada no arquivo de log da classe Online ou Batch. Não serão gerados, nesse caso, nenhum log particular de uma classe do tipo "library".

Portanto, com a utilização desse parâmetro, todas as informações geradas na execução de um determinado processamento ficam em um único arquivo de log.

Qualquer valor diferente de "A" faz com que seja gerado um arquivo para cada programa ou library executada.

SWONLINE_MONITOR

Essa variável define a ativação de conexão do programa Online com o SWADM, permitindo que o programa possa receber um DS. Sempre que um programa Online iniciar a execução, ele verificará se essa variável está definida no sistema para possibilitar a monitoração do ambiente. Qualquer informação válida passada pelo SWADM para um programa Online só será processada quando o programa está em execução. Portanto, a solicitação fica em fila para processamento até que o programa possa receber a informação, quando retorna para processamento.

SWPRINT_SAMPLE

Essa variável define se a função de marca d'água (WaterMark) estará ativa, invalidando todos os relatórios emitidos em PDF por todos os sistemas. Os valores possíveis são "N" ou "S". Serão exibidos um texto e uma imagem a serem definidos nas variáveis SWPRINT_SAMPLE_TEXT e SWWATERMARK_IMAGE respectivamente.

SWPRINT_SAMPLE_TEXT

Com a variável SWPRINT_SAMPLE ativa um texto será exibido no rodapé do arquivo impresso. Essa variável define o texto que será exibido, no máximo 30 caracteres. Caso esse valor não seja preenchido será utilizado o valor default: "Esse é um documento inválido!!!!".

Framework SW - Overview

SWWATERMARK_IMAGE

Essa variável define o caminho para a imagem utilizada como marca d'água. Um caminho válido deve ser passado e a imagem deve estar no formato jpg. O caminho indicado para armazenar essas imagens é a pasta /mainframe/sys/swadm/SWSCRIPTS/images.

Framework SW - Overview

4. AMBIENTE DE EXECUÇÃO

O ambiente e execução de aplicações baseadas em framework SW utilizam o conceito de SYSTEM, que é o agrupamento de recursos de hardware e software que possibilita a execução de aplicações Online e Batch.

Um SYSTEM pode ser formado por um ou mais SERVERs, que podem ser divididos de acordo com o tipo de serviço que executa. Os tipos de SERVERs possíveis são:

- SCRIPT SERVER

Tipo de servidor que tem como função básica a execução de scripts representando os jobs dos sistemas. Nesse servidor é instalada também a aplicação SWADM, de administração e monitoração do ambiente

- ONLINE SERVER

Tipo de servidor que tem como função a execução dos tomcats com as aplicações online disponibilizadas

- BATCH SERVER

Tipo de servidor que executará as classes batch dos sistemas. Essas classes são executadas a partir dos jobs em execução no servidor de scripts.

- DATABASE SERVER

Tipo de servidor relacionado aos bancos de dados relacionais utilizados nos sistemas

- PRT SERVER

Tipo de servidor utilizado exclusivamente como um dos servidores de impressão do ambiente. Podem existir vários servidores de impressão ativos.

- MIXED SERVER

Tipo de servidor que deve ser utilizado para servidores online onde o período noturno seja de baixa utilização, aproveitando esse intervalo de maior ociosidade para execução de processos batch. Define-se um horário de início e término de processamento batch possível e, com isso, pode-se utilizar um servidor com uma função primária(Online), sempre ativa, e outra secundária(batch), controlada por um intervalo determinado.

Framework SW - Overview

- ALL SERVER

Tipo de servidor onde todos os tipos de execuções podem ser feitas: scripts, online e batch. Essa definição é utilizada em ambientes de teste/desenvolvimento que não necessitam de estrutura completa e segregada, como é o caso do ambiente de produção.

Como restrição, um ambiente pode ter somente um SCRIPT SERVER, pois o processamento do fluxo de execução de jobs não exige recursos que necessitem de mais de um servidor para processamento.

A quantidade de servidores do tipo ONLINE, BATCH e DATABASE devem ser definidos de acordo com o perfil de aplicações, volume de transações Online, quantidade e característica de processos batch que precisam ser executados diariamente.

A execução de programas batch, online e scripts têm como requisito básico a utilização de uma área compartilhada de armazenamento. Essa área faz com que todos os arquivos possam ser acessíveis através de todos os servidores existentes no ambiente.

Isso significa que um arquivo gerado por um processo executado em um determinado servidor batch, aplicação ou script possa ser acessado pela execução de outro processo batch, online ou script qualquer.

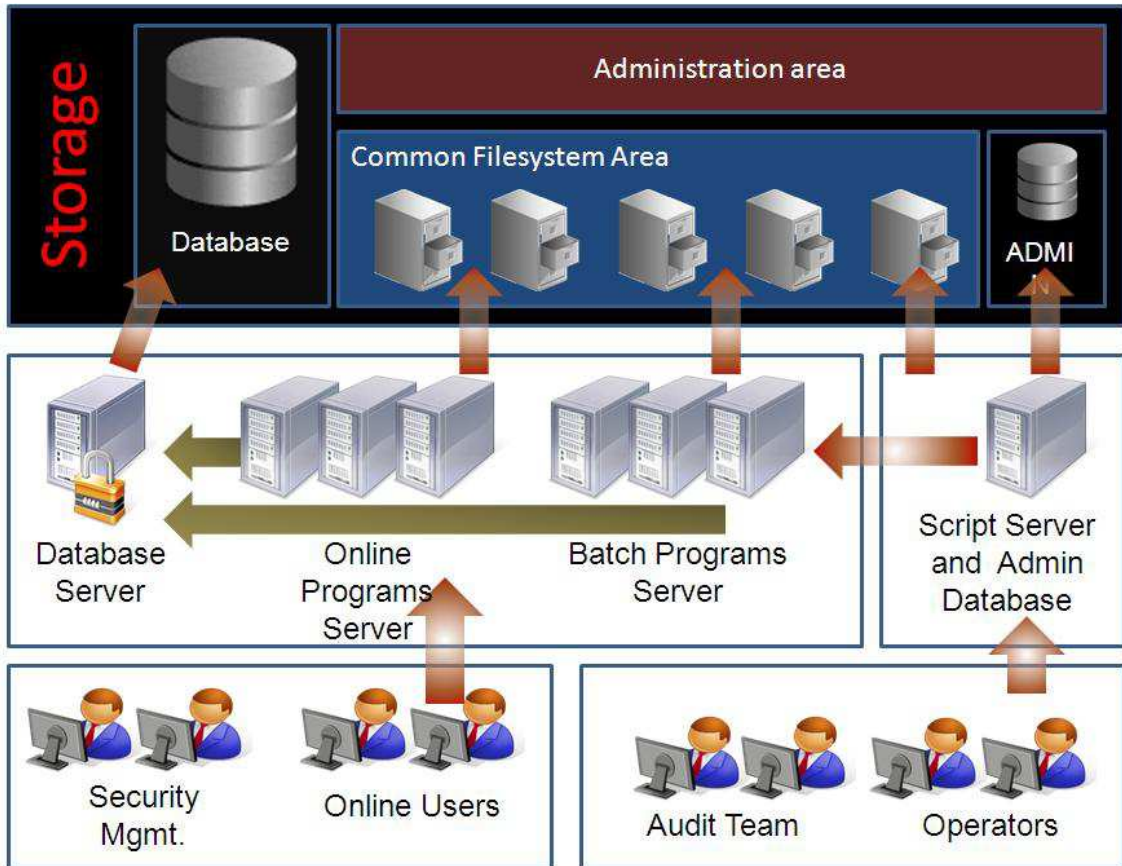
Para manter-se a correspondência com a localização física de um arquivo do mainframe para a plataforma baixa, a área compartilhada deve ser formada por, no mínimo, tantos filesystems quanto as famílias de armazenamento de informações de aplicação do mainframe e com os mesmos nomes.

O tamanho em gigabytes e a forma como o acesso dos vários servidores à essa área compartilhada pode ser diferente para o ambiente de testes e homologação/produção. Porém, tais filesystems deveram estar sempre disponíveis para utilização em um servidor sempre que este for reiniciado.

Deve existir uma variável de ambiente SWDIR_<família> para cada família do mainframe utilizada no processamento do ambiente, para programas batch, online e jobs. Essa variável deve apontar para um diretório físico e deve existir um subdiretório com o nome dos logins executores utilizados. Todos os arquivos criados pela execução de um programa sob um login específico em uma dada família será armazenado em \$SWDIR_[família]/[login].

A figura seguinte apresenta um desenho da configuração de um SYSTEM, com todos os seus componentes.

Framework SW - Overview



Framework SW - Overview

5. APLICAÇÕES ONLINE

O ambiente Online de sistemas desenvolvidos utilizando framework SW utiliza o tomcat para processamento das transações. Servidores apache podem ser definidos como infraestrutura de entrada e balanceamento de carga. Os sistemas podem ser configurados em uma ou mais instâncias tomcat, porém, recomenda-se a adoção de múltiplas instâncias para possibilitar uma administração mais eficiente.

Aplicações Online construídas a partir do uso do framework SW utilizam gerenciadores específicos por tipo de tecnologia utilizada no desenvolvimento das aplicações. As transações podem ser enviadas através de browsers, utilizando páginas jsp, ou por sequencia de caracteres, vindas de outras classes em servidores distintos, por exemplo.

Esses gerenciadores são especificados no nível de contexto, que é o agrupamento de funções e classes de negócio acessadas a partir de uma URL. A grande diferença entre esses vários gerenciadores é a forma como manipulam a entrada de dados feita pelo usuário, possuindo regras específicas para validação automática dos dados digitados, além de detalhes relacionados à forma de apresentação das informações nas páginas, manipulados a partir de lógica existente nas classes de regra de negócio.

Para processamento das transações Online, classes de negócio e classes de interface(screens) são desenvolvidas. Para aplicações que permitem entradas através de browsers, é necessário desenvolver páginas jsp.

O gerenciador possui várias funções dentro do fluxo de execução de uma transação Online, sendo que uma delas é a de ser um típico gerenciador de transações. As funções executadas pelos gerenciadores são:

- Receber transações enviadas vindas de browsers ou de outros programas
- Transformar as informações recebidas em textos, de acordo com a definição feita na classe Screen a ser utilizada
- Executar as validações de entrada definidas na classe Screen e, em caso de erro, retorna mensagens correspondentes, encerrando o fluxo
- Quando a transação enviada está relacionada à geração de PDF, o gerenciador é responsável pela geração física do PDF a partir do arquivo de relatório correspondente e retornar o PDF para o browser
- Identificar o transaction code relacionado aos dados recebidos e verificar qual a classe que está associada a esse transaction code
- Colocar a transação recebida na fila da classe associada ao transaction code
- Gerenciar as cópias de cada classe de regra de negócio
- Receber o resultado da transação processada pela classe de negócio e gerar o retorno de acordo com a origem da transação

Framework SW - Overview

Existe um gerenciador específico para cada tipo de programa Online migrado, que contém tratamento específico para as particularidades existentes na tecnologia Online suportada:

- Gerenciador COMS: utilizado para execução de programas Online que não possuem telas
- Gerenciador SADS: utilizado para execução de programas Online de sistemas SADS
- Gerenciador SDF/SDFPlus: utilizado para execução de programas com telas definidos através do SDF ou SDFPlus.
- Gerenciador LINC/EAE: utilizado para execução de Specs existentes em um sistema LINC/EAE
- Gerenciador MPACT: utilizado para execução de programas associados a sistemas MPACT

Um contexto pode possuir uma ou mais transações(transaction) associados. Essa associação é feita atribuindo à transação uma classe de regra de negócio. Pode-se limitar a utilização de um contexto, associando lista de usuários que tem permissão de acesso.

Uma ou mais transações podem ser processadas por uma mesma classe de regra de negócio Online, sendo identificadas unicamente através do transaction code. Isto significa que um ou mais transaction codes podem ser processados pela mesma classe de negócio. Um transaction code pode estar associado a apenas uma classe de negócio para um determinado contexto.

O processamento das transações pelas classes de negócio utiliza o sistema de filas, onde podem existir várias cópias das classes de negócio tratando transações existentes em sua fila. Cada vez que uma transação chega para ser processada e é validada inicialmente pelo gerenciador, a transação é colocada na fila de execução da classe de regra de negócio e é selecionada quando sua posição é a primeira da fila. Isso significa que o tempo de execução de uma transação recebida é composto pelo tempo de fila adicionada do tempo de processamento efetivo dessa transação na classe de regra de negócio.

A vantagem de utilizar a arquitetura de filas faz com que o consumo de recursos de hardware utilizado no processamento de um contexto seja controlado. Isso porque o número de cópias de classes é limitada de acordo com os parâmetros de configuração definidos, e independente da carga de transações recebidas pelo ambiente em um dado período de tempo.

Pode-se perceber que, no caso do uso de filas, quando o volume de transação de um determinado tipo é demasiadamente elevado em um dado período do dia, o tempo perdido na fila da classe pode ser significativo, muito maior que o tempo médio de processamento das transações pela classe de negócio. A quantidade de cópias é utilizada justamente para deixar uniforme o tempo de execução de uma transação. É tarefa do ajuste do ambiente determinar

Framework SW - Overview

qual a quantidade de cópias adequada a uma determinada classe de negócio para fazer com que o tempo de execução total de uma transação seja adequado na grande parte do tempo.

Devido à quantidade de transações de um determinado transaction code, durante as 24 horas do dia, apresentar-se muito distinta, pode-se configurar número mínimo de máximo de cópias para uma classe de regra de negócio. Com isso, o número de cópias em execução varia de um mínimo a um máximo dependendo da quantidade de transações de um determinado tipo recebidas para processamento. É de responsabilidade do gerenciador controlar o número de cópias de acordo os parâmetros configurados.

Na prática, a configuração de transações nos contextos não é feita associando diretamente transaction code a classes. As transações não na verdade associada à Agendas, que são estruturas de definição que podem receber lógica de pré e pós processamento. As Agendas são o elo entre os transaction codes e as classes de execução(programas).

A definição de parâmetros para que um contexto possa funcionar normalmente é feita pela aplicação SWADM, uma interface web de configuração e monitoração do ambiente. Contexto, agenda, transaction code, programas são definidos através do SWADM.

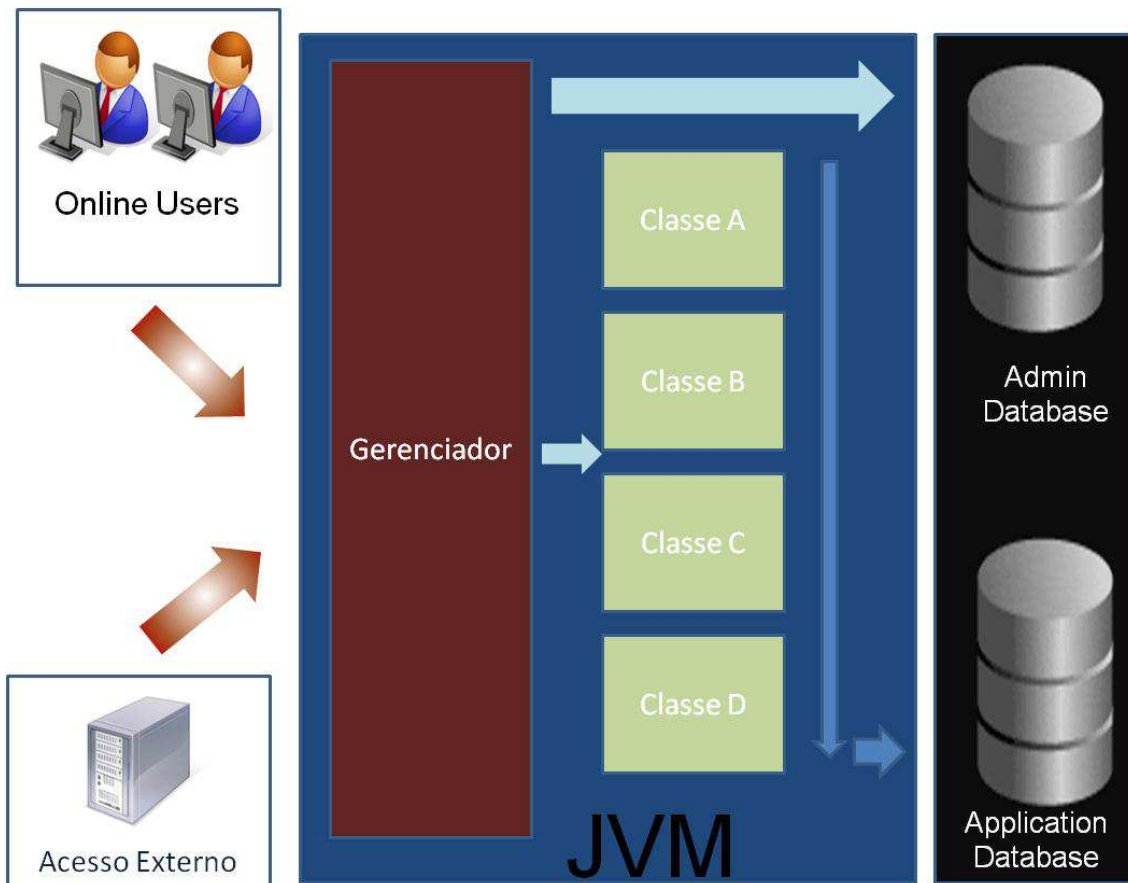
Além do SWADM, existe um daemon (SWAdmEventDaemon) que verifica as instâncias ativas e se essas instâncias ainda estão disponíveis. Caso qualquer instância tomcat seja abortada, todas as classes online em execução nessa instância ficarão com status de ABORT no SWADM.

Existem parâmetros de execução, através de variáveis de ambiente, que são utilizados pelos gerenciadores e classes de negócio online para geração de informações adicionais sobre as transações recebidas e o resultado do processamento. Além disso, níveis de debug podem fazer com que mensagens adicionais sejam gravadas no catalina.out, possibilitando informações mais detalhadas a cerca do processamento das transações, permitindo analisar situações de exceção.

O utilitário SWClient pode ser utilizado para listar o status de execução de cada classe de uma determinada URL, com informações como número de cópias no momento, número de transações em fila no momento, tempo médio de execução das transações e número de transações executadas desde a inicialização da instância do tomcat. Além disso, possibilita a geração de um relatório com detalhamento de todas as transações em fila em um determinado instante, com informações de IP de origem, session id, hora de chegada da transação e classe que deve processar a requisição.

A figura seguinte apresenta os componentes básicos utilizados em um ambiente de processamento Online.

Framework SW - Overview



A seguir estão relacionados definições e atributos utilizados em um ambiente Online, incluindo sua respectiva descrição.

CONTEXT

CONTEXT é uma definição relacionada a uma aplicação Online que agrupa agendas, transaction codes e programs, indentificada através de uma URL específica em um dado servidor.

Para ter acesso a um contexto, um usuário deve ter esse contexto associado à sua lista de contextos permitidos. Um contexto, através de comandos de operação, pode ser desabilitado, fazendo com que os usuários que tentem acessar esse contexto recebam mensagem de erro correspondente.

Um contexto pode ser baseado em transaction codes ser um contexto onde não existe código de transação no texto recebido pelo gerenciador. No primeiro tipo, podem ser definidas várias classes de negócio e transaction codes e a definição de transaction code/agenda/program definirá qual classe de regra de negócio possui a lógica para tratamento de um determinado transaction code. No segundo, teremos um contexto de classe de regra de negócio única.

Framework SW - Overview

CONTEXT LIST

CONTEXT LIST é uma definição para agrupamento de um ou mais contextos. A criação dessa lista é utilizada para autorizar um determinado usuário a acessar os contextos incluídos. Quando um usuário tenta acessar um context que não está incluído na lista de contextos atribuída a ele, uma mensagem de erro será enviada e o acesso ao contexto será negado.

AGENDA

As AGENDAs são mecanismos para que um pré e/ou pós-processamento possa ocorrer em uma determinada transação enviada. As transações, ou transaction codes, são associados à agendas. Quando as transações são recebidas pelo gerenciador, este determina para um dado contexto, qual agenda está associada. Uma agenda está diretamente relacionada a uma classe de negócio(program) e, com a identificação da agenda associada a um transaction code, automaticamente está definido qual classe de negócio receberá essa transação.

Uma única agenda em um contexto pode ter o atributo "DEFAULT AGENDA" ativado. Uma DEFAULT AGENDA é a agenda que será associada a qualquer texto recebido pelo gerenciador de um determinado contexto que não tem um transaction code válido. Essa característica é utilizada de duas formas:

- a. Para tratar transações que não possuem um transaction code válido, em um contexto baseado em transaction code,
- b. Para tratar transações em um contexto que não é baseado em transaction code, sendo formado por apenas uma classe de regra de negócio

TRANSACTION CODE

Os transaction codes são sequencia de caracteres de até 17 posições que identificam unicamente um texto recebido pelo gerenciador. Ele ocupa os primeiros bytes do texto gerado através do mapeamento dos campos do formulário jsp com a Screen correspondente e a quantidade de bytes é especificada na definição do contexto.

Sempre que uma transação chega ao gerenciador, após a transformação da entrada em um texto correspondente à Screen associada, o transaction code é identificado e é pesquisada a agenda associada. Caso essa agenda não exista, devido a um transaction code não cadastrado, a DEFAULT AGENDA será utilizada para direcionar o texto recebido. É responsabilidade da classe de negócio associada à DEFAULT AGENDA realizar o tratamento de textos que não possuem transaction code válido para um determinado contexto.

Framework SW - Overview

PROGRAM

As classes de negócio são definidas no SWADM como programs. Para cada program é possível definir número máximo e mínimo de cópias que serão controladas pelo gerenciador correspondente, dentre outros atributos.

Um program é associado a um contexto através de uma agenda. Toda a agenda deve possuir uma regra de negócio associada(program). Um program pode ter lógica para tratar vários transaction codes. Esses transaction codes precisam estar associados à agenda correspondente à regra de negócio que deve tratá-la.

MIN COPIES

Esse atributo define qual o número mínimo de cópias que uma determinada classe de negócio terá em execução quando o gerenciador correspondente realizar operação de redução de cópias que foram acrescidas em um período de pico de transações.

Quando a quantidade de transações recebidas de transaction codes associados a uma classe de negócio(program) aumenta em um determinado período, gerando uma fila de transações aguardado recurso disponível de programs para processamento, cópias adicionais de uma classe de negócio são colocadas em execução, até o limite definido no atributo MAX COPIES.

Quando a atividade de transações diminui, a quantidade de cópias atual pode não ser mais necessária e o gerenciador passa a encerrar cópias devido à inatividade. Essa redução é feita até o limite definido em MIN COPIES.

A regra para encerrar uma cópia em execução, até o MIN COPIES definido, depende do atributo TERMINATION TIME LIMIT.

MAX COPIES

Esse atributo define qual o número máximo de cópias de uma determinada classe de negócio que o gerenciador pode colocar em execução dependendo da quantidade de transações recebidas de transaction codes associados ao program em questão.

A regra para colocar uma nova cópia em execução depende dos atributos QUEUE DEPTH e INITIATION TIME LIMIT.

QUEUE DEPTH

Esse atributo define o número de transações na fila de um determinado program a partir da qual será contado o tempo definido em INITIATION TIME LIMIT. Se a fila ficar igual ou maior

Framework SW - Overview

que o valor especificado em QUEUE DEPTH e em um tempo superior ao definido em INITIATION TIME LIMIT, uma nova cópia será colocada em execução.

INITIATION TIME LIMIT

Esse atributo define o tempo, em segundos, para avaliar a fila de um determinado program e possibilitar ao gerenciador colocar ou não uma nova cópia em execução. Se a fila for igual ou superior ao valor definido em QUEUE DEPTH pelo tempo definido nesse atributo, o gerenciador irá colocar uma nova cópia do program em execução, até o limite definido em MAX COPIES.

TITLE

Esse atributo define o nome da classe de regra de negócio associada à definição de um program.

DESTINATION

No cadastramento ou alteração de uma agenda, esse atributo define o nome do program que será associado a essa agenda.

Framework SW - Overview

6. APLICAÇÕES BATCH

Aplicações batch criadas com a utilização do framework SW utilizam dois tipos de objetos para processamento: classes de regra de negócio e jobs com fluxos de execução das classes existentes.

Um programa batch é uma classe executada a partir da linha de comando ou de um script shell e tem como característica a interface através de parâmetros de execução passados externamente ou definido por acesso a dados(banco ou arquivos). Sua execução é feita em uma JVM exclusiva e o tempo de processamento e recursos utilizados, via de regra, é mais significativo em comparação a uma transação Online.

Apesar do script ser executado no servidor de scripts, uma classe batch será executada em um servidor do tipo BATCH, MIXED ou ALL. A seleção de qual servidor será utilizado para processamento da classe de regra de negócio batch depende da quantidade de processos batch em execução nos servidores no instante em que a classe tem que ser iniciada.

Como componentes tradicionais de um programa batch temos arquivos de dados, entrada e saída, arquivos de impressão, acesso a tabelas de banco de dados, para leitura e/ou atualização.

As operações realizadas com esses objetos são registradas no log de execução da classe de negócio, que é um arquivo que contém informações geradas durante o processamento, com estatísticas relacionadas a arquivos e tabelas de banco de dados gravadas ao final do processamento.

Um programa batch estende a classe SWCobol da framework SW. Essa classe contém toda a infraestrutura necessária para que uma classe desenvolvida possa utilizar os recursos para regra de negócio definidos no framework SW. É a classe base para qualquer classe de negócio desenvolvida.

Uma classe batch deve possuir um método main, que possibilita a execução direta através de linha de comando. Esse método é utilizado para identificação de parâmetros de execução, quando necessários, que devem ser passados ao método EXECUTE.

A classe de negócio batch, assim como a online, durante a execução grava informações de processamento em seu log de execução, de acordo com o nível de debug do processamento. Esses arquivos estão localizados no diretório apontado pela variável de ambiente SWDIR_LOGPROGRAM.

A aplicação SWADM permite monitorar a execução dos processos e interagir com os programas batch e jobs, além de classes online em execução, quando necessário. Algumas opções estão disponíveis durante a execução de um programa batch em execução, tais como AX, DS, HI, etc.

Framework SW - Overview

Para que qualquer mensagem seja enviada, é necessário selecionar o processo desejado no SWADM e receber a página de status do processo. Nessa página existe um combo com as opções disponíveis na situação atual de execução.

Quando um job ou classe batch executa um comando ACCEPT, o processo fica suspenso até que um valor seja enviado por usuário conectado no SWADM e com privilégio para envio de mensagens para jobs e classes. Essa operação no SWADM é feita através da seleção do AX como opção de envio.

No caso do HI, dois valores específicos (88 e 89) permitem gravação das estatísticas parciais e linhas de debug do arquivo de log de execução. Existe ainda mais um valor especial de HI, 87, que grava no log variáveis SWField da classe e os valores correspondentes no instante do processamento do HI recebido.

Uma classe batch pode ter sua execução suspensa se executar um comando de abertura de arquivo para leitura e o arquivo não existir no diretório indicado. Nesse caso, o processo fica com status de WAIT, em situação de NO FILE. Três são as situações possíveis nesse caso: disponibiliza-se o arquivo no diretório solicitado, descontinua-se a classe batch(DS) ou utiliza-se o FA(FILE EQUATION) pelo SWADM, informando um novo nome de arquivo que deve ser utilizado pelo processo em sua sequência de execução.

As classes de negócio são executadas pelos jobs, que contém o fluxo de execução de um determinado processamento sistêmico, refletindo a sequência de eventos que devem ocorrer para que a rotina possa ser completada corretamente.

A execução de jobs pode ser feita através da aplicação SWADM, onde jobs e parâmetros são previamente cadastrados, ou através do comando START do prompt do servidor de scripts.

A solicitação de execução de um job segue regras para efetivar a execução. Essas regras são tratadas por um daemon, SWAdmEventDaemon. Esse daemon é o responsável pela seleção de jobs solicitados para execução, assim como tratar todos os processos ativos, como jobs, instâncias online e classes batch.

A execução dos jobs é através da especificação de uma fila. Quando solicita-se a execução de um job esse é colocado na fila correspondente e será selecionado de acordo com a regra "First In, First Out".

Os jobs podem ser solicitados utilizando um parâmetro STARTTIME. Quando esse parâmetro é especificado, define-se hora e quantidade de dias a partir da data corrente que deve ser utilizado como base para seleção do job para execução. Enquanto essa data base não é atingida, os jobs com STARTTIME não são tratados pelo algoritmo de seleção. Essa situação permite programar a execução de um job para uma data futura. Um job sem STARTTIME indica que a execução deve ser feita imediatamente, se o algoritmo de seleção indicar a possibilidade de execução imediata.

Framework SW - Overview

Cada fila contém um parâmetro chamado MIXLIMIT, que determina o número máximo de jobs simultâneos que podem estar sendo executados por ela. Esse parâmetro é usado pelo algoritmo de seleção de jobs para determinar se a execução pode ocorrer ou não.

Além do parâmetro MIXLIMIT de fila, um parâmetro global de ambiente (MIXLIMIT) contém o valor máximo de jobs em execução no ambiente, independente da fila. Dessa forma, mesmo que o MIXLIMIT de uma determinada fila não tenha sido alcançado e existam jobs para execução, nenhum é processado. O daemon colocará um job para execução efetiva quando as condições de MIXLIMIT da fila e do sistema permitirem.

O daemon armazena informações de sua execução em um log, com nome "execuções.log", armazenado no diretório definido pela variável de ambiente SWDIR_LOG. A cada ciclo de execução, as mensagens geradas são armazenadas nesse log, possibilitando visualização das atividades executadas pelo daemon, bem como localizar situações de exceção e mensagens correspondentes.

Nos jobs, além da execução de classes, existe a transferência de arquivos entre servidores, cópia, renomeação e testes de residência de arquivos, por exemplo. COPYs e CHANGEs de arquivos possuem uma característica de execução um pouco diferente do Linux, permitindo que possa ser especificada uma máscara mais abrangente.

Como resultado de uma execução de classe de negócio ou uma subrotina do job existe a chamada variável de task. Essa variável, usada para controle do status do processo, indica o resultado da execução, podendo ser utilizada no fluxo do job para verificar términos anormais de processamento de regras de negócio, o que inviabilizaria a execução das etapas seguintes. A variável SW_TASK pode ser utilizada após a execução de uma classe para consultar o status final de execução ou para ser armazenada em outra variável que possa ser utilizada mais adiante.

Dependendo da característica dos processos que são executados dentro de um job, existe a possibilidade de execução síncrona ou assíncrona desses processos.

No caso da execução síncrona, o fluxo do job segue um caminho único, sendo que a próxima etapa só é iniciada ao final da etapa anterior.

No caso em que partes do processo definido em um job possam ser executadas em paralelo até que um ponto de sincronismo seja necessário, é possível solicitar a execução assíncrona desses processos(função PROCESS), possibilitando ao fluxo do job seguir em outras atividades, até que encontre uma função solicitando sincronismo dos processos assíncronos. Nessa situação, o job para seu processamento e aguarda que as rotinas paralelas especificadas sejam finalizadas para que a sequencia de execução possa ocorrer.

A execução assíncrona, para ser sincronizada com o fluxo do job mais adiante no processamento, deve possuir uma variável de task deve ser passada como parâmetro. O status de execução do processamento assíncrono pode ser consultado posteriormente utilizando essa variável definida(função WAIT_PROCESS).

Framework SW - Overview

É possível passar informações entre jobs e classes batch, além de parâmetros de execução. O uso de TASKVALUE e TASKSTRING fazem com que valores numéricos e alfanuméricos sejam passados do job para classe batch de negócio e a última movimentação da classe de negócio para essas duas variáveis podem ser retornadas ao job. No job utilizam-se as variáveis SW_TASKVALUE e SW_TASKSTRING enquanto que na classe de negócio utiliza-se os métodos setTASKVALUE() e setTASKSTRING() do atributo Myself da classe básica SWCobol.

A situação mais comum de uso desse recurso ocorre quando a classe de negócio precisa passar algum retorno relacionado à informação lida do banco ou de arquivo para que o fluxo de execução do job, após a finalização da classe de negócio, possa seguir de acordo com o resultado do processamento finalizado.

Nos sistemas baseados em framework SW, as classes Online também podem solicitar execução de jobs, utilizando os métodos SYSTEM_WFL ou START. A transação Online pode ser finalizada sem que o job solicitado entre efetivamente em execução, dependendo das condições de MIXLIMIT atual do ambiente.

É possível que uma classe batch, executada por um job disparado através de uma classe Online, retorne informações para o browser do originador do START, com informações sobre o andamento de sua requisição. Para isso, basta que o job executado pela transação Online passe o session id da sessão como parâmetro. A aplicação online SEND_MESSAGE, que deve estar configurada e ativa, se encarregará de efetuar o direcionamento das mensagens externas aos usuários correspondentes.

Framework SW - Overview

7. ACESSOS EXTERNOS

Existem algumas formas de execução de transações no ambiente Online provenientes de entradas de outros sistemas, e não de transações enviadas por páginas de usuários conectados, utilizando uma URL.

Uma das alternativas para aplicações externas que precisam executar transações em quaisquer contextos existentes em um ambiente passa pela necessidade de configuração para possibilitar acesso controlado. Essa configuração é feita através da definição de PORTs no SWADM e utilizará o daemon AplCcf para gerenciar as transações entre os ambientes.

Cada PORT definido é associado a um único contexto, representado pela nome do contexti e string correspondente à URL de processamento.

Existem alguns tipos de PORTs disponíveis, chamado de service type. Cada service type trata a conexão do meio externo de forma diferente, portanto é necessário conhecer como o ambiente externo envia requisições para possibilitar a configuração adequada desse atributo.

É possível executar um “auto login” no contexto destino, informando usuário e senha que possuam acesso ao contexto destino. Existe a possibilidade de limitar a origem dos textos a um IP pré-determinado.

A operação desse tipo de comunicação entre aplicações externas e aplicações baseada em framework SW é feita através do módulo AplCcf.

Os principais atributos são:

PORT NUMBER

Esse atributo define a porta que será utilizada na comunicação entre a aplicação externa e o contexto framework.

SERVICE

Define o nome do serviço referenciado pelo AplCcf.

Framework SW - Overview

YOUR IP ADDRESS

Quando informado, esse atributo define o IP de origem que terá textos aceitos para ser repassado ao contexto associado. Se alguma mensagem for originada de um IP diferente do cadastrado, um mensagem será gerada e o texto não seguirá seu fluxo até o contexto.

SERVICE TYPE

Esse atributo define o tipo de serviço atribuído ao PORT definido. Cada tipo de serviço envolve uma regra de tratamento específica:

- 0: ADMIN
- 1: APL06
- 2: APL09
- 3: CRLF
- 4: SIZE
- 5: ECHO

MAX OUTPUT

Atributo que define o tamanho máximo de texto de saída do contexto que será enviado à aplicação externa

CONTEXT

Esse atributo define qual contexto está associado ao PORT em questão. Textos recebidos pelo AplCcf serão direcionados ao URL do contexto em questão. Todas as regras envolvendo contexto utilizarão essa informação como referência.

URL

Nesse atributo é especificada a URL correspondente ao contexto associado ao PORT.

USER NAME

Esse atributo define o Username que deve ser utilizado para o processo de “auto login” no contexto associado, quando essa situação for necessária. Esse será o Username do originador da transação que pode ser pesquisado dentro da classe de negócio.

Framework SW - Overview

PASSWORD

É o complemento para possibilitar o “auto login”. A password informada é usada em conjunto com o Username para executar o procedimento de “auto logon”, permitindo que uma transação externa possa ser executada em um contexto que exige autenticação válida.

Além do AplCcf, um sistema migrado pode ser acessado através do Web Service(WS), um recurso de execução baseado execução via browser, com um único input, processamento múltiplas transações e uma página de retorno formada por informações selecionadas dos processamentos executados.

Para o caso de comunicação de um texto de entrada e um texto de retorno, pode-se utilizar execução por URL. Isso pode ser feito para sistemas originalmente desenvolvidos em COMS, LDL, SADS ou SDF.

Para execução de transações para programa COMS, a transação deve ser disparada através da montagem da URL correspondente à janela desejada, passando como parâmetros para o programa o campo RECORD, com o layout requerido pelo programa. Se houver necessidade de TRANCODE, esse deve ser informado nas primeiras posições do registro.

O retorno dessa transação é uma sequencia de caracteres enviados pelo programa, no layout definido pelo item de grupo existente na WORKING-STORAGE(WS) do programa. Dessa forma, o retorno deve ser particionado de acordo com esse layout para obtenção dos vários campos relacionados.

O programa chamador precisa ter codificação para envio de URLs como essa para o sistema migrado, possibilitando o processamento das requisições externas.

No caso de programas SADS, a transação deve ser disparada também através da montagem da URL mas com um parâmetro adicional, CWS=S. Esse recurso irá utilizar o sw-ws, pacote que atende a característica CWS no ambiente. O atributo RECORD também precisa ser informado e contém o texto que será passado para o sistema SADS na mesma sequencia como está definido na classe SCREEN correspondente ao layout da transação que se deseja executar.

No caso de programas SDF, a transação deve ser disparada também através da montagem da URL e com um parâmetro adicional, CWS=S. O atributo RECORD também precisa ser informado e contém o texto que será passado para o sistema SDF na mesma sequencia como está definido na classe screen correspondente à transação que se deseja executar. Esse layout sempre começa com o trancode da transação, seguido dos campos necessários para processamento da regra de negócio.

O processamento é similar ao que é executado para transações SADS, com exceção do retorno, que possuem parâmetros diferentes.

Framework SW - Overview

8. CLASSES SW

O arquivo sw.jar contém os recursos necessários ao desenvolvimento das aplicações online e batch, através da disponibilização de várias classes. Em conjunto com vários daemons e aplicação SWADM, compõe o framework SW.

A principal classe do framework de referência para o desenvolvimento de classes de negócio é a SWCobol. Ela contém os principais métodos básicos utilizados por qualquer regra de negócio.

As classes batch devem estender a classe SWCobol para utilização dos recursos do framework, além de utilizar um esqueleto padrão para efetivar esses recursos.

Já as classes Online devem estender a classe correspondente ao gerenciador utilizado na execução do ambiente: SWComs, SWLinc, SWMpact, SWSads ou SWSdf. A classe SWComs é a classe básica para aplicações Online e estende a classe SWCobol. Todas as demais classes online estendem SWComs, tendo, portanto, os mesmos recursos básicos do framework disponíveis. Além disso, recursos específicos para cada tecnologia estão disponíveis para utilização.

As principais classes existentes no sw.jar são relacionadas a seguir, incluindo uma descrição de recursos relacionados.

SWAdmLibrary

Essa classe é utilizada para interface entre outras classes e o banco SWADM de administração e monitoração. É utilizada por classes Online, batch, jobs e daemons do framework.

SWApICcfInvoker

Essa classe é utilizada envio de transações online a partir de uma classe batch de negócio, executando por intermédio de um job.

SWCobol

Classe responsável pelas funções básicas do framework, como rotinas de inicialização e finalização das classes Online e Batch, além de recursos diversos para apoio à codificação.

Framework SW - Overview

SWComs

Classe básica para definição de classes Online de regra de negócio, estendendo SWCobol. Contém os métodos RECEIVE() e SEND() que estão relacionados à recebimento e envio de retorno de transações Online. As demais classes Online estendem SWComs.

SWConn

É a classe responsável pela criação de atributo de conexão de um banco de dados. Qualquer classe que queira efetuar operações com uma base de dados deve definir um atributo do tipo SWConn, que utilizará parâmetros de conexão armazenados em diretório apontado por variável de ambiente SWDIR_SETUP.

SWExtract

Classe responsável pela manipulação de arquivos de dados. Uma classe para cada arquivo de dados deve ser criada, estendendo SWExtract, contendo atributos SWField e construtores de acordo com a necessidade da classe de regra de negócio que utilizará o arquivo.

SWField/SWFieldAlpha/SWFieldNumeric/SWFieldGroup e Redefines

Classes utilizadas em classes de tabelas de banco, arquivos de dados, arquivos indexados, Screens, regras de negócio online e batch para criação de atributos de tipos específicos. Cada tipo de Field contém atributos particulares e são os atributos mais utilizados na codificação das regras de negócio.

SWIndex

Classe responsável pela definição de índices incondicionais e condicionais utilizadas na definição de classe de tabela ou arquivo indexado. Contém a lógica para geração dos comandos SQL que representam os métodos de acesso utilizados nas classes de negócio.

SWIndexFile

Classe responsável pela definição de arquivo indexado. Um arquivo indexado possui uma tabela base e várias tabelas de dados, cada uma com um nome distinto. Por serem estruturas de tabela relacional, essa classe estende SWTable, que é a classe utilizada na criação das classes de tabelas relacionais.

Framework SW - Overview

SWMpact

Classe responsável pelas funções relacionadas a aplicações online que utilizam gerenciador Mpact.

SWMyself

Classe com atributos disponíveis para identificação de uma classe online ou batch em uma dada execução. Atributos como USERCODE, JOBNUMBER, MIXNUMBER, STATION, SESSIONID podem ser pesquisadas pela classe de negócio.

SWOccurs

Classe utilizada nos construtores de SWFields para definir que o atributo criado possui ocorrências de armazenamento. A indexação dessas ocorrências inicia-se sempre com 1 e no construtor de SWOccurs define-se a quantidade máxima de ocorrências permitida.

SWParagraph

Classe que define um atributo parágrafo, semelhante ao método. Essa definição permite que a classe de regra de negócio utilize o método PERFORM, contido em SWCobol, para execução desse método específico, fazendo com que o framework tenha um maior controle sobre a execução, possibilitando, por exemplo, a conversação com a aplicação SWADM através de HI e DS.

SWPdf

Classe contendo métodos que permitem a geração de arquivos PDF a partir de arquivos de relatório gerado pelas aplicações.

SWPrint

Classe responsável manipulação de arquivos de impressão. Arquivos de impressão que precisam ser criado na execução de uma classe devem ter um atributo do tipo SWPrint definido, para que os métodos existentes em SWPrint possam ser utilizados.

Framework SW - Overview

SWRotinas

Classe que possui uma série de funções de uso geral para utilização por classes online e batch.

SWSads

Classe responsável pelas funções relacionadas a aplicações online que utilizam gerenciador Sads.

SWScreen

Classe utilizada em definição de classes de interface entre páginas e classes online, contendo atributos e métodos de validação. Essas classes de interface devem estender SWScreen e utilizar SWFields na definição de seus atributos.

SWSdf

Classe responsável pelas funções relacionadas a aplicações online que utilizam gerenciador Sdf.

SWTable

Classe utilizada na definição de classes de tabelas de banco de dados. Os métodos existentes em SWTable ficam disponíveis na classe de tabela, que estende SWTable em sua definição. Esses métodos são utilizados nas classes Online e Batch para manipulação da tabela relacionada.

SWView

Classe utilizada para definição de um atributo em classe online ou batch associado a uma view existente em um banco de dados. As operações de leitura dos dados e retorno dos campos relacionados fazem parte dessa classe.

Framework SW - Overview

9. SWADM

O SWAdm é uma aplicação Online que compõe o framework SW e destina-se à operação e à administração dos sistemas, dos recursos e dos serviços executados para o ambiente desenvolvido utilizando recursos do framework.

As atividades básicas existentes estão disponíveis em ambiente WEB onde as funções são disponibilizadas aos vários da usuários da aplicação através de configuração das autorizações correspondentes. Isso possibilita acesso limitado dependendo do perfil do usuário conectado.

Através da definição de sistemas, que são unidades de processamento formadas por vários servidores e definições de processamento, pode-se configurar um ambiente onde os sistemas desenvolvidos podem ter uma administração e operação segura.

Suas várias funções estão agrupadas de acordo com o tipo do recurso ou da tarefa a ser realizada, a saber:

- Administration
- Backup
- Security
- System Management (SM)
- Statistics
- WFL

Todas as definições e informações geradas são armazenadas em banco de dados relacional. A retenção das informações históricas pode ser feita para garantir que todo tipo de necessidade relacionado à auditoria do ambiente pode ser atendido diretamente da base de dados.

Os parâmetros de segurança definidos através do SWAdm possibilitam a autenticação de usuários e a autorização de execução de sistemas e transações existentes.

A autenticação segue o padrão que inclui a seleção de usercode e password para validação primária e accesscode e password para validação secundária, sempre que o usercode tiver sido especificado como atributo “accesscode required” ativado.

Um usercode pode ser do tipo “accesscode indicator” ou não. Quando é definido com esse tipo ativado, esse usuário é chamado usuário de conexão. Esse usuário pode conectar na aplicação SWADM normalmente.

Um usuário com “accesscode indicator” desativado é chamado usuário de execução. Normalmente haverá um usuário com mesmo nome disponível no ambiente Linux. Os processos batch e online deverão estar associados a esses usercodes em sua execução.

De forma geral, as opções selecionadas apresentam uma página de consulta aos itens existentes no formato de um grid. Nessa mesma página é possível executar uma consulta mais restritiva, através das opções disponíveis na parte superior de página, além de gerar um relatório em formato PDF do conteúdo do grid.

Framework SW - Overview

A seleção de um item do grid faz com que a tela de manutenção desse item seja apresentada, possibilitando alteração e exclusão. Para operações desse tipo uma janela de confirmação sempre será apresentada.

Parte da definição existente é utilizada na documentação do ambiente. Para isso, campos de descrição breve e longa foram incluídos em várias das páginas de manutenção. A classe para geração de documentação extrai a informação existente e deixa disponíveis em forma de páginas HTML estáticas.

Um resumo dos módulos existentes e sua característica são apresentados a seguir.

ADMINISTRAÇÃO

Contém as funções de cadastramento dos recursos lógicos que permitem a configuração de um sistema. São elas:

- Aplicações (Applications);
- Email List
- Email List Detail
- Filas de execução (Queues)
- Servidores (Servers);
- Sistemas (Systems);

BACKUPS

Contém as funções de cadastramento e de controle dos arquivos de impressão controlados pelo sistema. São elas:

- Relatórios (Backups)
- Impressoras (Printers)
- Filtros de Impressão (Filters)

SECURITY

Contém as funções de cadastramento dos recursos objeto de controles de segurança, envolvendo autenticação de usuários de conexão e usuários de execução e autorização de execução de contextos existentes.

Framework SW - Overview

É através dessa função que são feitas manutenções nas definições de processamento Online tais como contextos, agendas, programas, transações, etc.

Além disso, os parâmetros para conexão externa também são especificados através do módulo SECURITY.

A lista de opções disponível inclui:

- Agenda
- Context
- Context List
- Context List Detail
- Local Data
- Ports
- Program
- Security Type
- Security Type List
- Station List
- Station List Detail
- Transaction
- Users

STATISTICS

Esse módulo exibe uma janela com possibilidade de consulta de estatísticas de todos os jobs executados no sistema, da atividade Online executada, caso a opção de gravação esteja habilitada, e comandos SQL que ultrapassam o limite de tempo definido.

SYSTEM MANAGEMENT(SM)

Contém as funções diretamente associadas à operação do Sistema. As opções disponíveis são:

- Jobs/Programa Events
- System Events
- Process
- Queues
- Start Job
- Instances

Framework SW - Overview

WFL

Esta opção está relacionada com definições para processamento batch, envolvendo tanto definição de scripts, com seus parâmetros de execução e programas associados, como também a definição dos programas disponíveis no ambiente, por sistema.

As opções disponíveis são:

- Query
- Program

Framework SW - Overview

10. OUTROS RECURSOS

O framework SW é formado por componentes adicionais que complementam o ambiente de execução.

swlpdaemon.jar

Esse pacote é o responsável pelo processamento das requests de impressão que são associadas com alguma impressora. Um daemon fica permanentemente em execução para identificar requests que devem ser tratadas.

Utilizando o SWADM, é possível definir o comando que deve ser utilizado para tratar arquivos de impressão de uma determinada impressora, podendo ainda possuir filtro de pré-processamento. As definições gravadas no SWADM são utilizadas pelo daemon de impressão para processamento das requests.

sw-scripts.jar

Esse pacote contém classes que são utilizadas no ambiente de execução batch, envolvendo solicitação de impressão e execução de jobs.

sw-ws.jar

Esse pacote contém classes que são utilizadas em transações externas que devem ser processadas por aplicações migradas no formato CWS. Sistemas Online que utilizam gerenciadorSads e gerenciadorSdf podem utilizar esse tipo de recurso. As transações recebidas do ambiente externo podem possuir lógica para conversação com a aplicação Sdf ou Sads, podendo enviar e receber "n" transações com essas aplicações e formatando uma única página de saída.

AplCcf.jar

AplCcf é o pacote responsável pela comunicação de sistemas externos com sistemas desenvolvidos utilizando framework SW. Seu funcionamento básico depende de configurações feitas através da aplicação SWADM, onde as portas e tipo de serviço são

Framework SW - Overview

definidos. Basicamente, quando iniciado, o AplCCF busca todos os parâmetros para os serviços cadastrados no SWADM e passa a gerenciar esses serviços, aguardando conexões de ambientes externos. Quando uma conexão é recebida, o AplCCF repassa o texto para o gerenciador correspondente ao da URL associada, que processa a transação e retorna o resultado para o ambiente externo.

messagesManager

A aplicação Online messagesManager é a responsável pelo gerenciamento de mensagens enviadas por processamento batch originados de disparos de jobs através de transações Online.

Através de parâmetro contendo sessionid passado ao job, é possível enviar mensagens que possam ser visualizadas na página do browser do usuário que solicitou a execução, permitindo manter o usuário informado do progresso da execução dos processos batch que solicita.

sw-indexed-operations.jar

Esse pacote é responsável por operações de dump e load relacionado a arquivos indexados. Esses arquivos, que residem em tabelas de banco de dados, tem uma característica particular onde tabelas com nomes diferentes, mas mesma definição básica, são armazenadas, como se fossem arquivos de dados em um diretório.

copybook.jar

Esse pacote é utilizado para atualização em massa de books utilizados nas classes de negócio batch ou online. Books são arquivos com parte de código java e que são inseridos em classes que necessitam dessa sequencia de lógica. Um book pode ser utilizado por "n" classes e, caso exista necessidade de atualização de lógica desse book, o copybook executa a atualização em todas as classes que utilizam referencia para o arquivo.

Além de atualizar, o copybook possibilita extrair para um ou vários arquivos isolados a lógica embutida nas classes relativas a um ou vários books.