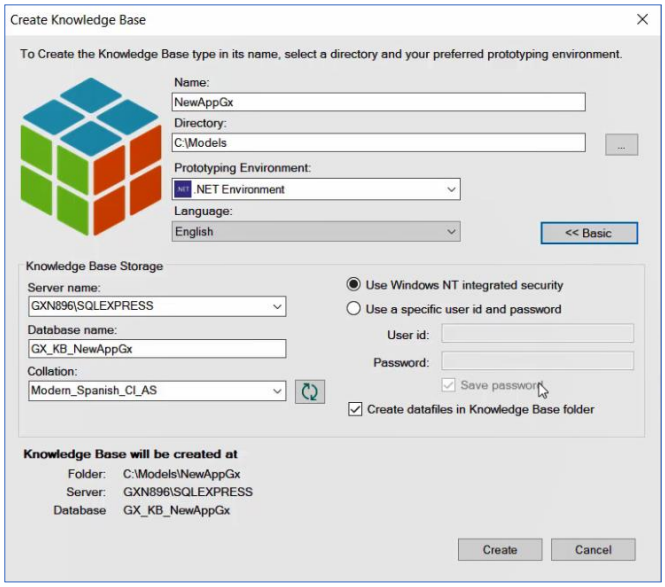
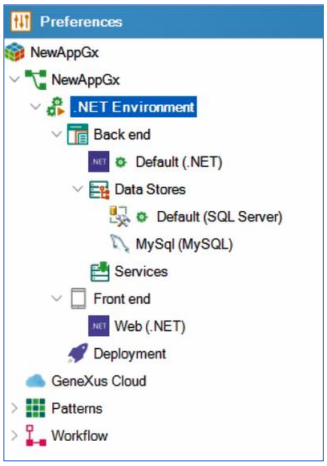
Setting some properties

GeneXus™





Quando criamos uma nova KB, como sabemos, a partir das opções básicas podemos configurar: seu nome, localização, linguagem de programação e idioma utilizado para os rótulos, botões, mensagens, etc. E em opções avançadas será possível alterar o nome e a localização do servidor de base de dados onde se encontrará a base de dados da KB, o nome dessa base de dados, a collation e se será utilizado o usuário do Windows ou um usuário específico para o acesso à base de dados.

Uma vez criada nossa KB, vejamos a partir da janela de “preferences” as opções que nos são apresentadas.
Sob o nó versão de nossa base de conhecimento, vemos o ambiente criado automaticamente a partir das definições iniciais.
Dentro se encontram os seguintes nós:

- **Backend**, onde estará tudo relacionado ao servidor de aplicações, os acessos à base de dados e os serviços.
No backend são definidas as linguagens de programação que serão utilizadas para gerar o código correspondente ao backend da aplicação, isto através dos geradores.
Em seguida, em Data Stores estará a informação para acessar a base de dados associada.
Também é possível gerar novos Data Stores com diferentes DBMS, por exemplo, para obter dados a partir de outras bases de dados externas.
A partir do nó Serviços, podemos configurar o manuseio dos serviços, configurações

de armazenamento, notificações, etc.

- E então temos o nó **Frontend**, onde se encontra tudo relacionado à interface do usuário e se conecta ao backend.
São mostrados aqui os geradores disponíveis para criar o Frontend da aplicação.
Por padrão, aparecerá a linguagem que selecionamos no environment, .Net, .Net Framework ou Java, e dependendo do tipo de aplicação que desenvolvemos, podem aparecer como geradores: Android, Apple e Angular.

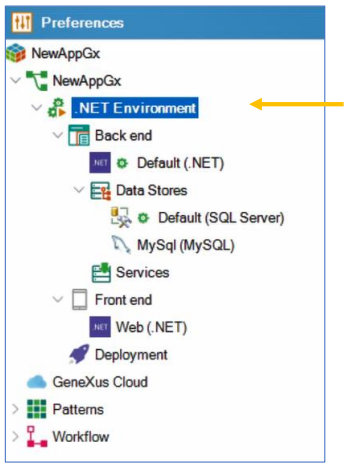
E por último temos o nó Deployment, que está relacionado aos Deployment Unit objects definidos na KB Version.
Temos a possibilidade de criar novos ambientes. Por exemplo, em desenvolvimentos onde temos um ambiente de testes e outro de produção, temos que ter, nesses casos, mais de um ambiente.

Suponhamos que criamos um novo com gerador Java e base de dados Oracle, e queremos que seja nosso gerador padrão para nossa aplicação. Podemos diretamente marcar que será nosso environment padrão quando criamos o ambiente, ou podemos fazê-lo posteriormente clicando com o botão direito do mouse sobre o ambiente desejado e selecionando “Set as current Environment”, o mesmo para mudar de um ambiente para outro.
Para executar uma aplicação Java Web, devemos previamente ter o Java Development Kit (JDK) instalado, o qual podemos baixar diretamente da página oficial da Oracle.
E também será necessário um servidor web onde será executada nossa aplicação, neste caso utilizaremos o Apache Tomcat, um software de código aberto, que pode ser baixado diretamente de seu site oficial.

Uma vez instalado, podemos verificar a partir da app do Windows “Serviços” se está executando corretamente, caso contrário, podemos iniciá-lo a partir dali. Aqui vemos o Apache Tomcat 9, que é o que temos instalado neste PC, funcionando corretamente.

GeneXus

Some properties of the KB Environment node



Vejamos agora algumas propriedades configuráveis dos nós que comentamos há pouco
No nó versão da KB, uma das propriedades que vemos é chamada Enable Integrated Security. Esta propriedade estabelece se a aplicação gerada irá gerenciar ou não a segurança através do GAM (GeneXus Access Manager), por padrão aparece como false.

Também encontramos propriedades que têm impacto na interface de nossa aplicação.
Podemos, por exemplo, especificar o objeto theme ou design system que queremos que seja aplicado por padrão aos diferentes objetos transação e/ou Web Panel que criamos, embora posteriormente a partir de cada um desses objetos possamos especificar qual tema ou design queremos que seja aplicado.

Também podemos definir a Master Page que será aplicada por padrão, que por padrão aponta para o objeto RwdMasterPage. Assim como na propriedade anterior, cada Transação e Web Panel, poderá configurar esta propriedade de forma independente.

Também podemos configurar algumas propriedades referentes às validações que são realizadas no lado do cliente, como a propriedade stop on error: Por padrão está em Não, mas se for alterada para Sim, quando em um formulário são validadas as regras

que definimos e encontre alguma entrada incorreta, será mantido o foco nesse campo até que seja corrigido, não permite continuar avançando para o próximo até que seja corrigido. Assim como também podemos definir a posição das mensagens de validação, entre outras configurações.

Dentro da seção Defaults temos a propriedade automatic refresh, que está por padrão em sim, e isso nos permite que quando tenhamos um grid e sejam realizadas alterações em qualquer variável que esteja sendo utilizada pelo evento Load, evento Refresh ou condições do grid, será realizada uma atualização automática do grid sem necessidade de que o usuário faça isso manualmente. O exemplo mais comum deste caso é quando temos filtros que impactam o grid. Se não for desejado este comportamento, pode ser alterada para Não.

No nó do ambiente da KB, na seção Environment temos a possibilidade de modificar a plataforma para a qual será gerada a aplicação, Web ou Windows. Podemos também a partir daqui modificar a linguagem de programação escolhida.
Também podemos alterar o DBMS atribuído.

A partir da propriedade Startup Object podemos atribuir um objeto de início para que seja aquele que é executado quando é iniciada a aplicação, ou seja, quando pressionamos F5.
Também podemos a partir desta propriedade modificar o nome do environment.
Vemos aqui a propriedade commit on exit, a partir dela definimos o valor que terá por padrão para cada objeto transação ou procedimento que criamos. Por padrão está em Yes, o que significa que o programa gerado executa uma confirmação, ou seja, um commit, ao final da unidade lógica de trabalho (LUW). Se a alteramos para No, não será realizada esta confirmação.

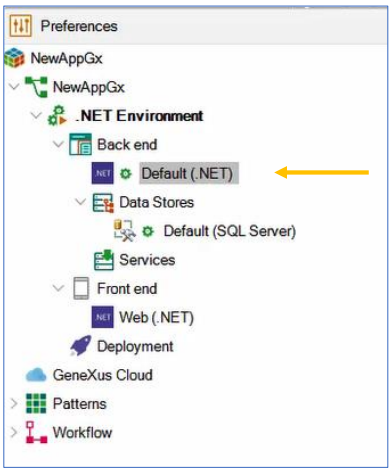
Também temos a propriedade chamada Encrypt Url Parameters. Com esta propriedade podemos permitir ou negar a encriptação dos parâmetros enviados para uma URL, e estabelecer níveis de segurança quando se utiliza a encriptação dos parâmetros entre Objetos Web. Por padrão, está definida como No, ou seja, que os parâmetros na URL dos objetos web não serão criptografados.
A opção session key indica que os parâmetros da URL serão criptografados com uma chave diferente para cada sessão. A criptografia é realizada usando cookies locais. Este valor oferece um maior nível de segurança, mas não permite URL compartilhada. Isto significa que um usuário não poderá enviar uma URL com parâmetros para outro usuário porque a URL não funcionará, pois é necessário o cookie correspondente para descriptografia.

A última opção é Site Key, se selecionarmos esta, os parâmetros na URL dos objetos Web estarão criptografados, mas a chave de criptografia será a mesma para todo o

site. Neste caso, não são utilizados cookies. Isto implica em um menor nível de segurança, mas facilita a transferência de links.
A propriedade protocol specification nos permite especificar sob qual protocolo será executada a aplicação ou os serviços Web. Por padrão, será acessível por HTTP, mas podemos alterar para HTTPS. Se o deixamos sem especificar, a aplicação não aplica nenhum protocolo, podendo acessar através de HTTP ou HTTPS indistintamente.

GeneXus

Some properties of the back-end generator



Em cada gerador, também temos várias propriedades configuráveis e estas dependem um pouco da linguagem escolhida, embora várias são compartilhadas. Vejamos por exemplo algumas se escolhemos .Net.
Podemos especificar a plataforma na qual será gerada a aplicação, web ou Windows.
Através destas propriedades (Log Level / User Log Level) podemos configurar o detalhe da informação que será salva no log com uma escala de 0 a 6, sendo 0 a opção off, ou seja, que não serão feitos registros, e a opção 6 “All” a que registrará toda a informação possível.

A propriedade reorganize server tables é utilizada para indicar se as tabelas do modelo serão reorganizadas ou não pelo gerador que atribuímos como reorganizador do modelo. Por padrão, vem definida como Yes. Se for deixada em No, quando o Relatório de Análise de Impacto da base de dados indicar que as tabelas precisam ser reorganizadas, será ignorado pelo gerador.

A propriedade transactional integrity, por padrão definida como Yes, permite que todas as Transações e Procedimentos sejam gerados com integridade transacional, caso se deseje o contrário pode ser alterada para No.

Se selecionamos o gerador Java, temos por exemplo a propriedade Compiler Options, aqui será possível incluir qualquer propriedade válida para o compilador de Java, conforme sua versão.

Para obter mais informações sobre as propriedades válidas, recomendamos acessar a informação no site de documentação da Oracle.
https://docs.oracle.com/javase/9/tools/javac.htm#JSWOR627 .

Estas outras propriedades (Reorganization Option / Create Database Option) determinam se o processo de criação/reorganização da base de dados terá uma intervenção do usuário ou tomará outras ações na reorganização.

Os possíveis valores para inserir são os seguintes:

-nogui, é o que vem por padrão, e determina que não é permitida a intervenção do usuário nestes processos. Por exemplo, no caso de executar em alguma plataforma diferente do Windows e que não conte com uma interface gráfica, pode ocorrer um erro ao reorganizar, e desta forma, com o comando -nogui evitamos a necessidade de uso da interface gráfica e solucionar aquele erro.
-force. No GeneXus temos arquivos com a extensão .Gen e .Exp, os .Gen são utilizados para executar a reorganização e os .Exp para controlar se a reorganização foi executada ou não. Com este comando, o controle sobre estes arquivos não é realizado, e a criação/reorganização da base de dados é forçada de qualquer maneira a ser realizada. Seu uso não é recomendado.
-noverifydatabaseschema, não verifica o esquema da base de dados no processo de criação/reorganização da mesma. Pode ser útil, por exemplo, em algum caso em que ocorre um erro de reorganização no processo de verificação do esquema.
-recordcount, este parâmetro irá contar os registros que serão criados/reorganizados em todas as tabelas, mostrará os resultados e irá parar. Em vez do botão reorganizar, aparecerá um continuar. Isto pode ser usado para estimar quanto tempo pode levar a tarefa.
-ignoreresume. Suponhamos que foi executada uma reorganização e falhou. Em seguida, são realizadas algumas alterações em transações que implicam atualizar a base de dados. Ao executar é necessária uma nova reorganização e ocorre outro erro. Isso porque GeneXus realiza as reorganizações consecutivamente. Se falha uma reorganização e tenta realizar outra, ainda faltará a reorganização anterior.
GeneXus oferece rotinas para continuar a partir do último ponto executado pela reorganização antes de uma falha.
Para evitar a execução destas rotinas de validação e executar o processo de reorganização desde o início. é inserido o comando -ignoreresume.
-donotexecute, é útil para ambientes em que precisa que a criação/reorganização seja gerada, mas não seja executada e, uma vez realizado o impacto, deseja realizar uma compilação completa e implementar a criação ou reorganização e os programas de aplicação em outro lugar.

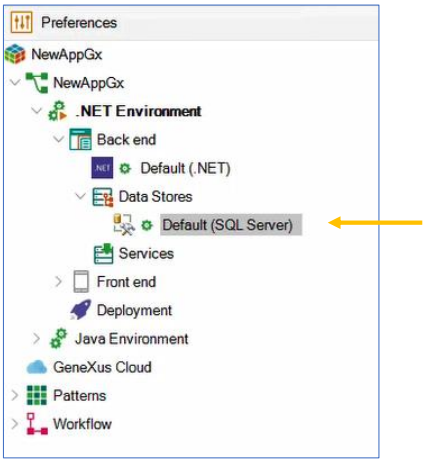
Vale esclarecer que os comandos podem ser combinados separando cada opção com

um espaço em branco. Por exemplo, se quiser combinar as opções nogui e force, ficaria desta maneira.
Outras propriedades configuráveis são sobre a execução de nossa aplicação. Por exemplo, definir se a prototipagem será na nuvem ou não, e no caso de selecionar que sim, qual será a URL do servidor.

Podemos também modificar o servidor Web que será utilizado, entre outras opções.

GeneXus

Some properties of the database generator



Para os geradores de base de dados, as propriedades dependem também do gerador escolhido, embora algumas são compartilhadas. Por exemplo para o caso de SQL Server:
Temos a propriedade Access technology to set, que nos permite configurar as possíveis tecnologias para o acesso à base de dados. Por exemplo, ADO.Net, usada no gerador .NET ou .NET Framework, ou JDBC para o gerador Java. Dependendo de qual selecionamos, são as propriedades que serão habilitadas para configurar em Connection Information.

Dentro desta seção (Connection Information), temos por exemplo a propriedade Database name, que é utilizada quando os ambientes necessitam nomes de bases de dados. Indica o nome da base de dados que conterá as tabelas e índices da aplicação.

Temos também a propriedade Server Name, que especifica o nome padrão que identificará o servidor de base de dados. Se não for especificado este parâmetro, os programas gerados exibirão uma mensagem na tela solicitando que seja inserido o nome do sistema quando for o momento de estabelecer a conexão.
Podemos também especificar a porta que utilizará a instância para se comunicar e trocar dados com as aplicações.

A propriedade Connect to server permite controlar em qual momento os programas gerados estabelecem conexão com a base de dados. A opção padrão é “At first

request", o que significa que a aplicação tentará estabelecer a conexão imediatamente após enviar a primeira solicitação ao servidor de base de dados.

A outra opção é "At application startup", neste caso a aplicação tentará estabelecer a conexão como parte do processo de inicialização, antes que seja exibido o objeto principal.
Um exemplo para entender a diferença poderia ser quando temos um procedimento que realiza cálculos sem conexão com a base de dados, e se for cumprida determinada condição, é chamado outro procedimento que realiza conexão com a base de dados. Com "At application startup" será estabelecida conexão com ela ao inicializar a aplicação, sem saber se fará uso ou não do segundo procedimento. E com "At first request" estabelece conexão com a base de dados apenas no caso em que se cumpra a condição e seja chamado o segundo procedimento, que é aquele que requer conexão com o servidor.

A propriedade Database schema permite apontar para um esquema de base de dados específico.
Um esquema de base de dados é quem descreve sua estrutura, é um modelo ou arquitetura de como serão vistos nossos dados. Em uma base de dados relacional, o esquema define suas tabelas, seus campos em cada tabela e as relações entre cada campo e cada tabela.

Estas que vimos são algumas das propriedades configuráveis de nossa KB, mas como vemos existem muitas outras que não mencionamos sua funcionalidade.

Se nos posicionamos em alguma propriedade que nos interessa e pressionamos a tecla "F1", nos leva para a Wiki de GeneXus e especificamente à descrição e uso da propriedade escolhida. Também podemos entrar diretamente na wiki e pesquisar pelo nome da propriedade. Levar em consideração que o atalho com a tecla "F1" pode, em algumas propriedades, ainda não estar disponível.

GeneXus™

training.genexus.com
wiki.genexus.com