

SMS Broker System
Software Integration

Our system uses
WCF
(Windows Communication Foundation)
&
.NET

Annex. Participating Government Agencies classes and interfaces.

(ver. from 5/26/2026)

Table of Contents

1. List of PGA templates.	1
2. Sample. Template usage. Adding of a new PGA document.	6
3. Sample. Template usage. Updating of existed PGA document.	8
4. Sample. How to queue a PGA Data Corrections message.	9

1. List of PGA templates.

Agency	Template code	Description	Program codes	Tariff flag codes
Food and Drug Administration (FDA)	FDADEV001	Medical Devices	DEV	FD1, FD2, FD3
	FDACOS001	Cosmetic commodities	COS	FD1, FD2, FD3
	FDABIO001	Biologics commodities	BIO	FD1, FD2, FD3
	FDADRU001	Drug commodities	DRU	FD1, FD2, FD3
	FDAFOOPN001	PN food commodity combined entry submission	F00	FD1, FD2, FD3, FD4
	FDAFO0001	Non-PN food commodity or PN requirements previously met	F00	FD1, FD2, FD3, FD4
	FDATOB001	Tobacco commodities	TOB	FD1, FD2
	FDARAD001	Radiation Emitting Products Commodity	RAD	FD1, FD2

	FDAVME001	Animal Drugs and Devices Commodity	VME	FD1, FD2, FD3
	FDAFDA001	Disclaimer	FDA	FD1, FD3
National Highway Safety Administration (NHTSA)	NHTXXX001	NHTSA Regulations	MVS	DT1, DT2
	NHTNHT001	Disclaimer	MVS	DT1
Animal and Plant Health Inspection Service (APHIS), Lacey Act Declaration	APHAPL001	Lacey Act	APL	AL1, AL2
	APHAPLD001	Lacey Act – Disclaimer	APL	AL1
Animal and Plant Health Inspection Service (APHIS), CORE	APHAPH001	Animal care – Disclaimer	AAC	AQ1
	APHAPH001	Biotechnology Regulatory Services – Disclaimer	ABS	AQ1
	APHAPH001	Veterinary Services – Disclaimer	AVS	AQ1
	APHAPH001	Plant Protection and Quarantine – Disclaimer	APQ	AQ1
	APHXXX001	Veterinary services	AVS	AQ1, AQ2
	APHXXX001	Plant protection and quarantine	APQ	AQ1, AQ2
	APHXXX001	Biotechnology regulatory services (aka BRS)	ABS	AQ1, AQ2
	APHXXX001	Animal Care	AAC	AQ1, AQ2
	APHXXX001	Veterinary services (data may be required, no disclaim required)	AVS	AQX
	APHXXX001	Animal Care (data may be required, no disclaim required) [1]	AAC	AQX
	APHXXX001	Plant protection and quarantine (data may be required, no disclaim required)	APQ	AQX
	APHXXX001	Biotechnology regulatory services (aka BRS) (data may be required, no disclaim required)	ABS	AQX
Environmental Protection Agency (EPA)	EPAV1E001	Vehicles and engines, form 3520-1	VNE	EP3, EP4
	EPAV2E001	Vehicles and engines, form 3520- 21	VNE	EP3, EP4
	EPAPS1001	Registered pesticides, form 3540- 1	PS1	EP5, EP6

	EPAPS2001	Pesticides devices, form 3540-1	PS2	EP5, EP6
	EPAPS3001	Pesticides other, form 3540-1	PS3	EP5, EP6
	EPAXX001	Toxic substance control act	TS1	EP1, EP2, EP7, EP8
	EPAXX001	Ozone depleting substances	ODS	EP1, EP2, EP7, EP8
	EPAEPA001	Vehicles and engines - Disclaimer	VNE	EP1, EP3, EP5, EP7
	EPAEPA001	Registered pesticides, form 3540-1 - Disclaimer	PS1	EP1, EP3, EP5, EP7
	EPAEPA001	Pesticides devices, form 3540-1 - Disclaimer	PS2	EP1, EP3, EP5, EP7
	EPAEPA001	Pesticides other, form 3540-1 - Disclaimer	PS3	EP1, EP3, EP5, EP7
	EPAEPA001	Toxic substance control act - Disclaimer	TS1	EP1, EP3, EP5, EP7
	EPAEPA001	Ozone depleting substances - Disclaimer	ODS	EP1, EP3, EP5, EP7
	EPAEPA001	Hydrofluorocarbons - Disclaimer	HFC	EH1
	EPAHFC001	Hydrofluorocarbons - CAS number(s)	HFC	EH1, EH2
	EPAHF1001	Hydrofluorocarbons - ASHRAE number	HFC	EH1, EH2
Alcohol and Tobacco Tax and Trade Bureau (TTB)	TTBT01001	Permit number - Tobacco	TOB	TB1, TB2, TB3
	TTBT01001	Permit number - Beer/Malt Beverages	BER	TB1, TB2, TB3
	TTBT01001	Permit number - Distilled Spirits	DSP	TB1, TB2, TB3
	TTBT01001	Permit number - Wine	WIN	TB1, TB2, TB3
	TTBT02001	COLA ID Number - Wine	WIN	TB1, TB2, TB3
	TTBT02001	COLA ID Number - Beer/Malt Beverages	BER	TB1, TB2, TB3
	TTBT02001	COLA ID Number - Distilled Spirits	DSP	TB1, TB2, TB3
	TTBT03001	Foreign Certificates - Distilled	DSP	TB1, TB2,

		Spirits		TB3
	TTBT03001	Foreign Certificates - Wine	WIN	TB1, TB2, TB3
	TTBT04001	IRC Registry Number - Beer/Malt Beverages	BER	TB1, TB2, TB3
	TTBT04001	IRC Registry Number - Distilled Spirits	DSP	TB1, TB2, TB3
	TTBT04001	IRC Registry Number - Wine	WIN	TB1, TB2, TB3
	TTBT05001	Permit Number for Tobacco Products, Cigarette Papers, or Cigarette Tubes Released Without Payment of Tax	TOB	TB1, TB2, TB3
	TTBT06001	Large Cigar Sales Price	TOB	TB1, TB2, TB3
	TTBT07001	Quantities of Cigarette Papers and Cigarette Tubes	TOB	TB1, TB2, TB3
	TTBXXX001	General form - Beer/Malt beverages	BER	TB1, TB2, TB3
	TTBXXX001	General form - Distilled spirits	DSP	TB1, TB2, TB3
	TTBXXX001	General form - Tobacco	TOB	TB1, TB2, TB3
	TTBXXX001	General form - Wine	WIN	TB1, TB2, TB3
	TTBTTB001	Disclaimer - Beer/Malt beverages	BER	TB1, TB3
	TTBTTB001	Disclaimer - Distilled spirits	DSP	TB1, TB3
	TTBTTB001	Disclaimer - Tobacco	TOB	TB1, TB3
	TTBTTB001	Disclaimer - Wine	WIN	TB1, TB3
Department of State Bureau of Political-Military Affairs Directorate of Defense Trade Controls (DDTC)	DTCXXX001	ITAR-related commodities	DTC	Tariff flags are not provided by agency.
	DTCDDTC001	Disclaimer	DTC	Tariff flags are not provided by agency.
Department of Justice Drug Enforcement Administration (DEA)	DEAXXX001	Drug enforcement administration	DEA	DE1
	DEAMAC001	Drug enforcement administration,	MAC	DE3

		Form DEA-452		
	DEADEA001	Disclaimer	DEA	DE1, DE3
Department of Agriculture (USDA), Food Safety Inspection Service (FSIS)	FSIFSIN001	For non-eCert Country	FSI	FS3, FS4
	FSIFSIS001	For eCert Country (Currently Australia, New Zealand, and the Netherlands)	FSI	FS3, FS4
	FSIFSI001	Disclaimer	FSI	FS3
Bureau of Alcohol, Tobacco, Firearms and Explosives (ATF)	ATFATF001	U.S. Department of Justice, Bureau of Alcohol, Tobacco, Firearms and Explosives	ATF	Tariff flags are not provided by agency.
National Marine Fisheries Service NMFS	NMF370001	Tuna Tracking and Verification Program (TTVP/NOAA 370)Requirements(ImportsOnly)	370	NM1, NM2
	NMFHMS001	Highly Migratory Species(HMS) Program Import and Export Requirements	HMS	NM5, NM6
	NMFAMR001	Antarctic Marine Living Resources(AMR) Program Requirements	AMR	EP3, EP4
	NMFNMF001	370 - Disclaimer	370	NM1
	NMFSIMA001	If an aquaculture scenario	SIM	NM8
	NMFSIMB001	If wild caught harvest (including small vessel harvests transferred to a larger vessel at sea) scenario	SIM	NM8
	NMFSIMC001	If wild caught - aggregated harvest, offloaded from multiple small vessels	SIM	NM8
USDA Agricultural Marketing Service (AMS)	AMSMO1001	Request for marketing order inspection (processing code - 1)	MO	AM3, AM4
	AMSMO2001	Reporting products previously inspected (processing code - 2)	MO	AM3, AM4
	AMSMO3001	Exempt per existing FV6 (processing code - 3)	MO	AM3, AM4
	AMSMO4001	Request an inspection exemption (processing code - 4)	MO	AM3, AM4
	AMSMO5001	Request pistachio aflatoxin testing (processing code - 5)	MO	AM3, AM4
	AMSMO6001	Report exempted type or variety	MO	AM3, AM4

		(processing code - 6)		
	AMSM07001	Report seasonally exempt product (processing code - 7)	MO	AM3, AM4
	AMSM08001	report minimum weight exempt product (processing code - 8)	MO	AM3, AM4
	AMSEG1001	Request for egg inspection (processing code - 1)	EG	AM1, AM2
	AMSEG2001	LPS-222 previously approved (processing code - 2)	EG	AM1, AM2
	AMSPN1001	Request for peanut inspection (processing code - 1)	PN	AM6
	AMSOR1001	Certification of organic product (processing code - 1)	OR	AM7, AM8
	AMSOR2001	Certification of organic product, electronic import certificate (processing code - 2)	OR	AM7, AM8
	AMSAMS001	Disclaimer	AMS	AM1, AM7
U.S. Fish and Wildlife Service (FWS)	FWSEDS001	EDS (entire data set)	FWS	FW1, FW2, FW3
	FWSLDS001	LDS (limited data set)	FWS	FW1, FW2, FW3
	FWSFWS001	Disclaimer	FWS	FW1, FW3
Office of Marine Conservation (OMC)	OMCXX001	Office of Marine Conservation	OMC	OM2
	OMCOMC001	Disclaimer	OMC	OM1

2. Sample. Template usage. Adding of a new PGA document.

```

public static void Test1()
{

    ServicesFactory sf = new ServicesFactory(@"https://localhost:8130/brokerservice", "UserName@sandbox", "YourPassword");

    IPGAManager PGAManager = sf.GetManager<IPGAManager>();
    IHarmonizedTariffManager mngrHTS = sf.GetManager<IHarmonizedTariffManager>();
    IContactManager mngrContact = sf.GetManager<SMS.Broker.ServiceContracts.Directories.IContactManager>();
    ICustomsPortManager mngrPort = sf.GetManager<SMS.Broker.ServiceContracts.Directories.ICustomsPortManager>();

    // In a real case it can be a tariff object with HTS number from the Customs entry:
    // For example:
    // We have CustomsEntry ce with some PGA ShipmentArticleTariff tariff.
    // Then we can take PGA tariff flags directly from tariff.HarmonizedTariff.PGA.
    // But to simplify the sample code and its usage we look here at hardcoded HTSnumber = "0801220000"
    // as to initial condition for our task.
    string HTSnumber = "0801220000"; // SHELLED BRAZIL NUTS
    HarmonizedTariff HTS = mngrHTS.GetByCode(HTSnumber, "");
    var pgaCodes = HTS.PGA.Trim().Split(3); // This HTS has 3 flags: AM7, AQ1, FD4.

```

```

// It means that final PGA document have three PG01 records.
// In the sample we will fill only FDA branch.

// Having PGA tariff flags we can limit the list to 13 templates and suggest it to user for template selection.
List< PGATemplate > templates = PGATemplate.Items.Where(x => x.Active)
    .Where(x => x.PGATariffCodes.Any(p => pgaCodes.Contains(p))).OrderBy(x => x.ProgramCode).ToList();

// This line immitate that user selects from the list a "PN food commodity combined entry submission" template.
string templateCode = templates.Where(x => x.Name == "PN food commodity combined entry submission").FirstOrDefault().Code;

// Create a new PGA document and save it to database. Its header and PGOI record are created by default.
Broker.DataContracts.Documents.PGA.PGA pga = PGAManager.New(null);
pga = PGAManager.Save(pga);

// PGA document tree needs PG01 record per each agency to work with templates.
// To simplify the sample code here we will add PGA data only for one agency.
PGABranch branchPG01 = new PGABranch();
branchPG01.Type = "PG01";
branchPG01.PGA = pga;
pga.PGOI.AddBranch(branchPG01);

if (templateCode == "FDAFOOPN001")
{
    FDAFOOPN001 template = new FDAFOOPN001();

    // Real template properties can be got from user interface
    // Hardcoded input is for this sample only.

    template.PGOI_CommercialDescription = "SOME GOODS";

    template.PG01_GovernmentAgencyProcessingCode = "PRO"; // Processed Food

    template.PG02_FDAProductCode = "1234567";

    template.PG06_CountryOfProductionGrowth = "BR"; // BRAZIL
    template.PG06_CountryOfShipment = "BR"; // BRAZIL

    template.PG10_CommodityCharacteristicDescription = "SOME CHARACTERISTICS"; // pga.PGOI.PGOI_CommercialDescription;
    template.PG26_BaseQuantity = 10;
    template.PG26_BaseUnitOfMeasure = "KG";

    // Commercial entities:
    template.PG19_PNS = mngrContact.GetByCode("TESPLA", ""); // PN Submitter
    template.PG19_PNT = mngrContact.GetByCode("TESPLA", ""); // PN Transmitter
    template.PG19_FD1 = mngrContact.GetByCode("TESPLA", ""); // FDA Importer
    template.PG19_DFP = mngrContact.GetByCode("TESPLA", ""); // Owner
    template.PG19_UC = mngrContact.GetByCode("TESPLA", ""); // Ultimate Consignee
    template.PG19_FSV = mngrContact.GetByCode("TESPLA", ""); // Foreign Supplier Verification Program Importer
    template.PG19_DEQ = mngrContact.GetByCode("TESSEL", ""); // Shipper
    template.PG19_MF = mngrContact.GetByCode("XYZMAN", ""); // Shipper

    // Points of Contact:
    template.PG19_PNS_PG21_PNS = mngrContact.GetByCode("TESPLA", ""); // PN Submitter
    template.PG19_PNT_PG21_PNT = mngrContact.GetByCode("TESPLA", ""); // PN Transmitter
    template.PG19_FD1_PG21_FD1 = mngrContact.GetByCode("TESPLA", ""); // FDA Importer
    template.PG19_FSV_PG21_FSV = mngrContact.GetByCode("TESPLA", ""); // Foreign Supplier Verification Program Importer

    template.PG19_FSVIdentificationCode = "16"; // D & B - assigned(DUNS number)

    template.PG23_VES_VesselName = "COSCO";
    template.PG23_VFT_VoyageTripFlightNumber = "333a";
    template.PG23_FME_FoodProcessingFacilityRegistrationExemption = "K"; // Unable to determine the registration
    // number of the manufacturer

    template.PG25_PGALineValue = 1000;

    template.PG26_BaseUnitOfMeasure = "KG"; // Kilograms(Weight)
    template.PG26_BaseQuantity = 100; // Kilograms(Weight)

    FDAFOOPN001.PG27Item equipment1 = new FDAFOOPN001.PG27Item() { PG27_ContainerNumber = "CBHU5465487888";
    template.PG27Items.Add(equipment1);
    FDAFOOPN001.PG27Item equipment2 = new FDAFOOPN001.PG27Item() { PG27_ContainerNumber = "YMLU654878788";
    template.PG27Items.Add(equipment2);
    FDAFOOPN001.PG27Item equipment3 = new FDAFOOPN001.PG27Item() { PG27_ContainerNumber = "MAUR5487555";
    template.PG27Items.Add(equipment3);

    template.PG30_RequestedOrScheduledTimeOfInspection = new DateTime(2025, 12, 19, 22, 54, 0); // military time
    // of 12/19/2025 10:54:15 PM
    template.PG30_PortOfArrivalOrEntryPort = mngrPort.GetByCode("4601", ""); // NEW YORKNEWARK AREA

    // Save template to tree.

```

```

template.FillTo(branchPG01);

// In a real case if you want to link this stand-alone PGA document to Customs entry tariff
// it is a possible point to do it.
//pga.ShipmentArticleTariff = tariff; // link a PGA document to tariff
//tariff.PGA = pga; // link a tariff to PGA document
//mngnCustomsEntry.Save(ce); // where mngnCustomsEntry = SMS.Broker.ClientContext
// // ServicesFactory.GetManager<ICustomsEntryManager>();
// // ce must be saved because we changed its tariff object.

// Save ALL PGA document (header and tree) to database.
PGAManager.SaveSubTree1(branchPG01);

}
if (templateCode == "FDADRU001")
{
    // .....
}
if (templateCode == "FDABIO001")
{
    // .....
}
if (templateCode == "FDAFOO001")
{
    // .....
}
}
}

```

3. Sample. Template usage. Updating of existed PGA document.

```

public static void Test2()
{
    ServicesFactory sf = new ServicesFactory(@"https://localhost:8130/brokerservice", "UserName@sandbox", "YourPassword");

    IPGAManager PGAManager = sf.GetManager<IPGAManager>();
    IHarmonizedTariffManager mngnHTS = sf.GetManager<IHarmonizedTariffManager>();
    IContactManager mngnContact = sf.GetManager<SMS.Broker.ServiceContracts.Directories.IContactManager>();
    ICustomsPortManager mngnPort = sf.GetManager<SMS.Broker.ServiceContracts.Directories.ICustomsPortManager>();
    ICustomsEntryManager mngnCustomsEntry = sf.GetManager<SMS.Broker.ServiceContracts.Documents.ICustomsEntryManager>();

    CustomsEntry ce = mngnCustomsEntry.Get(292, "Shipments.Invoices.Articles.Tariffs.HarmonizedTariff,Shipments.Invoices.Articles.Tariffs.PGA");
    ShipmentArticleTariff tariff = ce.Shipments
        .SelectMany(a => a.Invoices.SelectMany(b => b.Articles.SelectMany(c => c.Tariffs.Where(d => d.PGA != null))))
        .FirstOrDefault();

    var pga = PGAManager.Get(tariff.PGA.Id, "Branches,Branches.PG19_Contact,Branches.PG21_Contact");

    // Suppose user selects (APHIS) - Animal and Plant Health Inspection Service CORE.
    PGABranch BranchAPH_PG01 = pga.Banches.Where(x => x.Type == "PG01").Where(x => x.PG01_GovernmentAgencyCode ==
"APH").FirstOrDefault();

    if (BranchAPH_PG01 == null)
        return;

    string templateCode = BranchAPH_PG01.Template;

    if (templateCode == "FDAFOOPN001")
    {
        // ...
    }
    if (templateCode == "APHAPH001")
    {
        APHAPH001 template = new APHAPH001();

        // Fill template from PGA-tree:
        template.FillFrom(BranchAPH_PG01);

        // User inputs Plant Protection and Quarantine - Disclaimer:
        template.PG01_GovernmentAgencyProgramCode = "APQ";
        template.PG01_Disclaimer = "A"; // Product is not regulated by this agency
    }
}

```



```

        template.FillTo(BranchAPH_PG01);

        // Save ALL PGA document (header and tree) to database.
        PGAManager.SaveSubTree1(BranchAPH_PG01);

    }
    if (templateCode == "APHXXX001")
    {
        // ...
    }
    if (templateCode == "EPAV1E001")
    {
        // ...
    }
}
}

```

4. Sample. How to queue a PGA Data Corrections message.

```

public static void Test3()
{
    ServicesFactory sf = new ServicesFactory(@"https://localhost:8130/brokerservice", "UserName@sandbox", "YourPassword");

    // We need to prepare these managers first.
    SMS.Broker.ServiceContracts.ISmsABIManager mngrSmsABI = sf.GetManager<SMS.Broker.ServiceContracts.ISmsABIManager>();

    string SourceLink = "CustomsEntry=291"; // Customs entry link
    string Action = "R"; // Replace
    string Articles = "325;405"; // Articles Id-s

    // There is no a simple query method, we need to use a general PutToQueue() with parameters:
    List<SMS.Broker.DataContracts.Parameter> parameters = new List<SMS.Broker.DataContracts.Parameter>();
    parameters.Add(new SMS.Broker.DataContracts.Parameter() { Name = "SourceLink", Value = SourceLink });
    parameters.Add(new SMS.Broker.DataContracts.Parameter() { Name = "Action", Value = Action });
    parameters.Add(new SMS.Broker.DataContracts.Parameter() { Name = "Articles", Value = Articles });

    long ABITransmissionId = mngrSmsABI.PutToQueue("CA", parameters);
}

```